

# **Efficient User Authentication and Key Establishment Protocols for Wireless Sensor Networks & Internet of Things**

A thesis submitted to University of Hyderabad in partial fulfillment

for the degree of

**Doctor of Philosophy**

by

**Anup Kumar Maurya**

Reg.No. 13MCPC18



**SCHOOL OF COMPUTER & INFORMATION SCIENCES  
UNIVERSITY OF HYDERABAD  
HYDERABAD - 500046**

**Telangana, India**

**September, 2018**



This is to certify that the thesis entitled “**Efficient User Authentication and Key Establishment Protocols for Wireless Sensor Networks & Internet of Things**” submitted by **Mr. Anup Kumar Maurya** bearing registration number **13MCPC18** in partial fulfillment of the requirements for the award of the Degree of Doctor of Philosophy in Computer Science at the School of Computer & Information Sciences is a bonafide work carried out by him under my supervision and guidance at the Centre for Mobile Banking (CMB), IDRBT, Hyderabad.

This thesis is free from plagiarism and has not been submitted previously in part or in full to this or any other University or Institution for award of any degree or diploma.

Parts of this thesis have been published in the following publications:

1. Proceedings of SSCC 2015, pp. 134 –145, Springer (Chapter 3).
2. Information 2017, Vol. 8, 136. pp. 1 –38, MDPI (Chapter 4).
3. Proceedings of SSCC 2017, pp. 39 –53, Springer (Chapter 5).
4. Proceedings of SSCC 2017, pp. 79 –94, Springer (Chapter 6).
5. Proceedings of CIC 2017, pp.173 –180, IEEE (Chapter 7)
6. Proceedings of ANTS 2017, pp. 1 –6, IEEE (Chapter 8).

Further, the student has passed the following courses towards fulfillment of coursework requirement for Ph.D.:

	Course Code	Name	Credits	Pass/Fail
1	BT701	Data Structure and Algorithms	4	Pass
2	BT702	Operating System and Programming	4	Pass
3	BT708	Secure Computing	4	Pass
4	BT716	Mobile Computing and Secure Mobile Banking	4	Pass

Supervisor	Director	Dean
Prof. V. N. Sastry	Dr. A. S. Ramasastry	Prof. K. Narayana Murthy
IDRBT, Hyderabad-500 057	IDRBT, Hyderabad-500 057	SCIS, UOH
India	India	Hyderabad -500 046, India

# DECLARATION

I, **Anup Kumar Maurya**, hereby declare that this thesis entitled “**Efficient User Authentication and Key Establishment Protocols for Wireless Sensor Networks & Internet of Things**” submitted by me under the guidance and supervision of **Prof. V.N.Sastry**, IDRBT, is a bonafide research work of mine and is free from plagiarism. I also declare that it has not been submitted previously in part or in full to this University or any other University or Institution for the award of any degree or diploma. I hereby agree that my thesis can be deposited in Shodganga/INFLIBNET.

**A report on plagiarism statistics from the University Librarian is enclosed.**

**Signature of the Student**

**Date:**

**(Anup Kumar Maurya)**

**Reg. No.: 13MCPC18**

*//Countersigned//*

**Signature of the Supervisor:**

***Dedicated to My Parents and Teachers for their  
support and encouragement***

## Acknowledgements

I say my first “thank you” to my advisor **Prof. V. N. Sastry** for his guidance and support throughout my PhD duration at IDRBT. I am appreciative of the time and effort he has devoted to advising me over the past years, and his consistent encouragement and positive reinforcement have made it a very rewarding experience. I would also like to thank him for his encouragement and support that helped me build confidence when I felt stuck in my research.

It is my privilege to thank **Dr. A. S. Ramasastry**, Director, IDRBT for reviewing the work and timely suggestions throughout my research. I also thank **Mr. B. Sambamurthy**, Former Director, IDRBT for extending his cooperation at the preliminary stage of my research. It is an honor for me to thank **Prof. K. Narayana Murthy** (Dean, SCIS) and **Prof. Arun Agarwal** (Pro-Vice-Chancellor), UoH, Hyderabad for their academic support throughout research work. Thank you so much!

I would like to thank my doctoral review committee members, **Dr. V. Radha** and **Prof. S.K. Udgata** for providing invaluable inputs and suggestions.

I am thankful to and fortunate enough to get constant encouragement, support and guidance from **Prof. B. L. Deekshatulu**, **Prof. D. K. Subramanian**, **Prof. Venu Govindaraju**, **Prof. Rajkumar Buyya**, **Prof. B. Yegnanarayana**, **Prof. C. R. Rao**, **Prof. Arun K Pujari** and all the distinguished fellow, teaching staff of IDRBT & University of Hyderabad which helped me in successfully completing my thesis work. Also, I would like to extend my sincere regards to all the non-teaching staff of IDRBT, Hyderabad for their timely support.

I am significantly indebted to my parents, not only for their understanding and patience in every part of my educational endeavors, but also for allowing me to take my own decisions since childhood. I would not be where I am in the absence of their attachment, sacrifice and support.

I also want to thank many more people I was lucky to find on my way: Dr. Ilaiyah Kavati, Dr. Pradeep Kumar Dadabada, Dr. Gopal Narayan Rai, Mr. B. Shravan Kumar, Dr. Jatoth Chandra Sekhar, Dr. Deepnarayan Tiwari, Dr. Ghanshyam Bopche, Dr. M. Sandhya, Kumar Ravi, Mr. B. Sriramulu, Mr. V. Dinesh, Mr. Mallikarjun Reddy, Mr. G. Jaya Krishna, Mr. S. K. Kamaruddin, Mr. Ravi Uyyala, Mr. N. D. Patel, Mr. Shadab Ahmad, Ms. Malvika Singh, Mr. Mahesh Kumar, Mr. Nisanth Kartheek, Mr. John Paul, Mr. P. Avinash, Mr. Satya Krishna, Ms. Charmila, Ms. Haritha, Mr. Kiran, Ms. Pavani, Mr. V. Manoj Kumar, Mr. Srikanth, Mr. Srinath among others. Their friendship filled gaps of a lonely period and made me feel at home in the Hyderabad. Thank you all!

Last but not the least, I would like to thank my brothers, sisters and fellow labmates of Centre for Mobile Banking in IDRBT for the stimulating research discussions and for all the fun we had in the last years.

**(Anup Kumar Maurya)**

## **Abstract**

A user who wants to avail the services of Wireless Sensor Networks (WSNs) & Internet of Things (IoT) must get authenticated by the WSNs & IoT. The gateway nodes, present in the WSNs & IoT, collect the real-time information from sensor nodes and store it in their memory which the legitimate users can query. However, since the Gateway Node (GWN) collects the data from sensor nodes in regular intervals, the data present at the GWN may not be the real-time data which can mainly be used for statistical purpose or for analytics only. Hence, it is also required by the users to communicate with the sensor nodes directly to collect the real-time information. To communicate with the sensor nodes directly, the user needs to authenticate against the gateway nodes as well as the sensor nodes and it should establish a secure session key to ensure the secure communication. Given the resource-crunchy nature of sensor nodes, it is essential to design efficient user authentication and key establishment protocols with no compromise on security. Efficient user authentication and key establishment protocol helps to stop intruders from injecting packets and carrying malicious activities in WSNs & IoT. Based on the comprehensive literature survey and cryptanalysis of existing user authentication and key establishment protocols of WSNs & IoT, we have observed that most of the existing protocols cannot achieve complete security requirement.

Therefore, we have proposed six novel protocols. The first proposed protocol based on smart card and ECC is efficient for two-factor user authentication. Second protocol based on Fuzzy Extractor and ECDH is suitable for efficient multi-factor authentication. The third protocol based on LU Decomposition is suitable for light-weight session key establishment between the user and the sensor node. The fourth protocol based on Chinese

Remainder Theorem is ideal for quickly authenticating the user without the help of the gateway node. The fifth protocol based on Bloom Filter is used for the WSNs & IoT of large hop count (i.e., the large number of intermediate sensor nodes through which data must pass between source sensor node and gateway node). The novelty of this protocol is, it eliminates unauthorised querying message transmission at the initial level (i.e., at the sensor node itself) to avoid bogus message flooding from the sensor nodes to the gateway node (which exhausts the resources of WSNs). The sixth protocol based on Symmetric Hash Functions is significant for authenticating the users without pre-alignment between the test and the registered minutia points of the fingerprint for multi-hop WSNs & IoT.

The formal and informal security analysis indicates that our proposed protocols withstand the various security vulnerabilities involved in WSNs & IoT. The automated validation using AVISPA and Scyther tool ensures that the proposed protocols resist various major security attacks of WSNs & IoT. The logical authentication using the Burrows-Abadi-Needham (BAN) logic establishes the truth or correctness of our proposed protocols. The computational analysis shows that our proposed protocols are suitable for resource constrained sensor nodes like TelosB and MicaZ. Finally, the comparative analysis based on computational overhead and security features of other existing protocol indicate that the proposed user authentication systems are secure and efficient. This research work significantly improves the security of WSNs & IoT in an efficient manner.



# Contents

<b>Acknowledgments</b>	<b>v</b>
<b>Abstract</b>	<b>vii</b>
<b>List of Figures</b>	<b>xvi</b>
<b>List of Tables</b>	<b>xviii</b>
<b>List of Notations</b>	<b>xx</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.1.1 Wireless Sensor Networks and Internet of Things . . . . .	2
1.1.1.1 Hardware constraints of WSNs . . . . .	3
1.1.1.2 Sensor network limitations . . . . .	5
1.1.1.3 WSNs Topology . . . . .	6
1.1.2 Applications of WSNs and IoT . . . . .	7
1.1.2.1 Military and battlefield surveillance . . . . .	7
1.1.2.2 Smart homes . . . . .	7
1.1.2.3 Other applications . . . . .	8
1.1.3 General security requirements of WSNs and IoT . . . . .	8
1.2 Authentication in WSN and IoT . . . . .	10
1.2.1 Key Establishment in WSNs and IoT . . . . .	14
1.2.2 Assumptions of the Threat Model for WSNs & IoT . . . . .	16
1.2.3 Security specifications for User Authentication in WSNs and IoT	16

1.2.4	Functionality requirements of user authentication protocols for WSNs & IoT . . . . .	19
1.3	Technologies, Cryptography Concepts, and Formal Security Verification Methods used in the thesis . . . . .	20
1.3.1	Secure Hash Function . . . . .	20
1.3.2	Symmetric Hash Function . . . . .	21
1.3.3	Secure Encryption Algorithm . . . . .	21
1.3.4	ECDH Key Establishment . . . . .	21
1.3.5	Fuzzy Extractor . . . . .	22
1.3.6	Bloom Filter . . . . .	23
1.3.7	Chinese Remainder Theorem . . . . .	23
1.3.8	AVISPA Tool . . . . .	23
1.3.9	Syther Tool . . . . .	24
1.3.10	BAN Logic . . . . .	24
1.3.11	Random Oracle Model . . . . .	25
1.4	Motivation of the work . . . . .	25
1.5	Objective of the thesis . . . . .	26
1.6	Thesis Contributions . . . . .	27
1.7	Organization of the thesis . . . . .	28
<b>2</b>	<b>Literature Review</b>	<b>33</b>
2.1	Classification of security protocols in WSNs & IoT . . . . .	33
2.2	User authentications protocols for WSNs and IoT . . . . .	33
2.2.1	Literature review and problems identified in user authentication for WSNs and IoT . . . . .	35
2.3	Key Establishment Protocols for Wireless Sensor Networks and Internet of Things . . . . .	41
2.3.1	Probabilistic Key Distribution Methods . . . . .	41
2.3.2	Deterministic Key Distribution Methods . . . . .	46
2.3.3	Hybrid Key Distribution Methods . . . . .	48
2.4	Summary . . . . .	49

<b>3</b>	<b>Two Factor User Authentication and Key Establishment protocol for WSNs &amp; IoT</b>	<b>50</b>
3.1	Introduction and Problem Definition . . . . .	50
3.2	Our Contributions . . . . .	51
3.3	Review of Choi et al.'s [1] Protocol . . . . .	53
3.3.1	User Registration . . . . .	53
3.3.2	User Login and Authentication Phase . . . . .	53
3.3.3	User's Password Update Phase . . . . .	54
3.4	Cryptanalysis of Choi et al.'s [1] Scheme . . . . .	55
3.4.1	Assumptions . . . . .	55
3.4.2	Attacks on Choi et al.'s protocol . . . . .	55
3.4.2.1	Stolen Smart Card Attacks: . . . . .	55
3.4.2.2	Resilience against Sensor Node Capture Attack : . .	56
3.4.2.3	Sensor Node Energy Exhausting Attack: . . . . .	57
3.5	Proposed Protocol . . . . .	57
3.5.1	User Registration Phase: . . . . .	58
3.5.2	Login, authentication and session key establishment Phase: .	59
3.5.3	User's Password Update Phase: . . . . .	63
3.6	Security Analysis and Comparison . . . . .	63
3.6.1	Stolen Smart Card Attacks . . . . .	63
3.6.2	Resilience Against Sensor Node Capture Attack and Energy Exhausting Attack . . . . .	63
3.6.3	Other Security Attacks . . . . .	64
3.6.4	Formal Security verification of the proposed protocol using AVISPA Tool . . . . .	64
3.6.4.1	Experimental Setup and the Size of the Entities Involved in WSNs/IoT for the Simulation of Proposed Protocol Using AVISPA Tool . . . . .	65
3.6.5	Implementation of the Proposed Protocol Using HLPSL . . .	65
3.6.5.1	Analysis of the result obtained using AVISPA tool .	67
3.7	Performance Comparison . . . . .	67
3.8	Summary . . . . .	73

<b>4</b>	<b>Three Factor User Authentication and Key Establishment protocol for WSNs &amp; IoT</b>	<b>74</b>
4.1	Introduction and Problem Definition . . . . .	74
4.2	Our Contributions . . . . .	76
4.3	Review and Cryptanalysis of Various Recent Protocols of User Authentication for WSNs . . . . .	77
4.3.1	Review of A.K.Das's Protocol . . . . .	77
4.3.2	Cryptanalysis of A.K.Das's Protocol . . . . .	80
4.3.2.1	Stolen Smart Card Attacks . . . . .	80
4.3.3	Review of Choi et al.'s protocol . . . . .	82
4.3.4	Cryptanalysis of Choi et al.'s protocol . . . . .	83
4.3.4.1	Attack on the basis of Authorized or Legal User . . . . .	83
4.3.4.2	User Impersonation . . . . .	85
4.3.5	Review of Park et al.'s protocol . . . . .	85
4.3.6	Cryptanalysis of Park et al.'s protocol . . . . .	86
4.3.6.1	Sensor Node Impersonation Attack . . . . .	86
4.3.6.2	User Impersonation Attack . . . . .	88
4.4	Proposed Protocol . . . . .	89
4.4.1	Set-Up before the Deployment of WSNs & IoT . . . . .	89
4.4.2	Registration of $\mathcal{U}_i$ by the $GWN$ Using Secure Communication Channel . . . . .	90
4.4.3	User Authentication and Session Key Establishment Phase . . . . .	91
4.4.4	User's Credential Update Phase . . . . .	92
4.5	Security Analysis: . . . . .	92
4.5.1	Informal Analysis . . . . .	92
4.5.2	Formal Security Analysis . . . . .	97
4.5.2.1	Formal Security Verification Using Random Oracle Model . . . . .	97
4.5.2.2	Verification Using Scyther tool . . . . .	100
4.5.3	Verification Using AVISPA Tool . . . . .	104
4.5.3.1	Experimental Setup and the Size of the Entities Involved in WSNs & IoT for the Simulation of Proposed Protocol Using AVISPA Tool . . . . .	104

4.5.3.2	Basic Features of AVISPA Tool . . . . .	104
4.5.4	Implementation of the Proposed Protocol Using HLPSL . . . .	105
4.5.5	Description of the Output Format Generated by AVISPA Tool . .	107
4.5.5.1	Logical Verification Using BAN Logic . . . . .	111
4.6	Comparative Analysis based on Security Features and Computational Overhead . . . . .	117
4.6.1	Relative Security Analysis . . . . .	117
4.6.2	Relative Performance based on Computational Cost . . . . .	117
4.7	Overall Analysis and Lessons Learned . . . . .	120
4.8	Summary . . . . .	121
<b>5</b>	<b>LU Decomposition based User Authentication and Key Establishment pro- tocol for WSNs &amp; IoT</b>	<b>122</b>
5.1	Introduction and Problem Definition . . . . .	122
5.2	Our Contributions . . . . .	123
5.3	Discussions and Proposal . . . . .	124
5.3.1	LU Decomposition of <i>Mat</i> and secret sharing: . . . . .	124
5.3.1.1	Storage Analysis . . . . .	126
5.3.2	Pre-deployment Phase: . . . . .	126
5.3.3	User Registration Phase: . . . . .	127
5.3.4	User Authentication and Session Key Establishment Phase: . .	127
5.3.5	User's Credential Update Phase: . . . . .	127
5.4	Security Analysis: . . . . .	128
5.4.1	Informal Security Analysis: . . . . .	128
5.4.1.1	Attack based on stolen smart card: . . . . .	130
5.4.1.2	Replay Attack: . . . . .	130
5.4.1.3	User impersonation attack: . . . . .	130
5.4.1.4	Sensor node impersonation attack: . . . . .	131
5.4.1.5	Man-in-the-middle attack(MITM) : . . . . .	131
5.4.2	Security verification using Scyther and AVISPA tool : . . . . .	131
5.4.2.1	Logical verification using BAN logic . . . . .	131
5.5	Performance Comparison . . . . .	136
5.6	Summary . . . . .	137

<b>6</b>	<b>Chinese Remainder Theorem based User Authentication and Key Establishment Protocol for WSNs &amp; IoT</b>	<b>138</b>
6.1	Introduction and Problem Definition . . . . .	138
6.2	Our Contributions . . . . .	139
6.3	Proposed protocol . . . . .	140
6.3.1	Pre-Deployment Phase . . . . .	140
6.3.2	Registration Phase . . . . .	140
6.3.3	Authenticated Key Establishment Phase . . . . .	140
6.3.4	User's Credential Update Phase . . . . .	142
6.4	Security Analysis . . . . .	142
6.4.1	Informal Security Analysis . . . . .	142
6.4.2	Security verification using Scyther and AVISPA tool : . . . .	146
6.4.3	Logical verification using BAN logic . . . . .	149
6.5	Performance Comparison . . . . .	150
6.6	Summary . . . . .	151
<b>7</b>	<b>Bloom Filter based User Authentication and Key Establishment Protocol for WSNs &amp; IoT</b>	<b>152</b>
7.1	Introduction and Problem Definition . . . . .	152
7.2	Our Contributions . . . . .	153
7.3	Proposed Protocol . . . . .	155
7.3.1	Pre-Deployment Phase: . . . . .	155
7.3.2	User Registration Phase: . . . . .	156
7.3.3	Authenticated Key Exchange Phase: . . . . .	157
7.4	Performance and Security Analysis: . . . . .	159
7.4.1	Security Analysis . . . . .	163
7.4.1.1	Formal security analysis using Random Oracle Model	163
7.4.1.2	Informal security analysis . . . . .	165
7.5	Performance Comparison . . . . .	166
7.6	Summary . . . . .	169

<b>8</b>	<b>Symmetric Hash Function based User Authentication and Key Establishment Protocol for WSNs &amp; IoT</b>	<b>170</b>
8.1	Introduction and Problem Definition . . . . .	170
8.2	Our Contributions . . . . .	172
8.3	Proposed Protocol . . . . .	173
8.3.1	Pre-deployment Phase: . . . . .	174
8.3.2	User registration based on Symmetric Hash Function: . . . . .	174
8.3.3	Authenticated Key Exchange Phase: . . . . .	175
8.4	Performance and Security Analysis: . . . . .	177
8.4.1	Security Analysis . . . . .	180
8.4.1.1	Informal security analysis . . . . .	181
8.4.1.2	Formal security analysis . . . . .	182
8.5	Comparison of Security Features and Computational Overhead . . . . .	184
8.6	Summary . . . . .	185
<b>9</b>	<b>Conclusions and Future Work</b>	<b>186</b>
	<b>References</b>	<b>190</b>
	<b>List of Publications</b>	<b>206</b>
	<b>Appendix</b>	<b>207</b>
<b>A</b>	<b>Annexure (Publications Online)</b>	<b>208</b>

# List of Figures

1.1	Wireless body area network (WBAN). . . . .	4
1.2	Components of a sensor node [2] . . . . .	4
1.3	Applications of WSNs and IoT [3] . . . . .	9
1.4	Mutual Authentication and Session Key Establishment [4] . . . . .	12
1.5	Authentication Factors [5] . . . . .	13
1.6	Key Establishment in WSNs and IoT . . . . .	15
1.7	Differential power analysis [6] . . . . .	17
2.1	User Authentication and Key Establishment Protocol for WSN & IoT	34
2.2	Security Attacks on User Authentication Protocols of WSNs and IoT .	39
2.3	Basic Security Features of User Authentication Protocols of WSNs and IoT . . . . .	40
3.1	Sequence Diagram 1 for Registration, Authentication and Key Estab- lishment . . . . .	62
4.1	Sequence Diagram 2 for Registration, Authentication and Key Estab- lishment . . . . .	94
4.2	Security validation result obtained using Scyther tool. . . . .	103
4.3	AVISPA Architecture [7]. . . . .	105
5.1	Sequence Diagram 3 for Registration, Authentication and Key Estab- lishment . . . . .	128
5.2	Result obtained using Scyther tool. . . . .	133
5.3	Result obtained using AVISPA tool. . . . .	134



## LIST OF FIGURES

---

6.1	Sequence Diagram 4 for Registration, Authentication and Key Establishment . . . . .	144
6.2	Result obtained using Scyther tool. . . . .	148
6.3	Result obtained using AVISPA tool. . . . .	148
7.1	Pre-Deployment and Registration Phase . . . . .	158
7.2	Sequence Diagram 5 for Registration, Authentication and Key Establishment . . . . .	161
7.3	$f_{positive}$ with respect to $\alpha$ and $\beta$ [8] . . . . .	163
8.1	Sequence Diagram 6 for Registration, Authentication and Key Establishment . . . . .	178
A.1	Screenshot of SSCC-2015 Publication [2] . . . . .	208
A.2	Screenshot of Information-2017, MDPI Publication [1] . . . . .	209
A.3	Screenshot of CIC-2017 Publication [3] . . . . .	210
A.4	Screenshot of SSCC-2017 Publication [4] . . . . .	211
A.5	Screenshot of SSCC-2017 Publication [5] . . . . .	212
A.6	Screenshot of ANTS-2017 Publication [6] . . . . .	213

# List of Tables

1	List of notations used in this thesis . . . . .	xxi
2.1	Comparison of communication & computation costs among EG, q-composite, polynomial-pool and MPDKE methods . . . . .	46
3.1	User Registration Phase: . . . . .	59
3.2	User Authentication and Key Establishment Phase: . . . . .	61
3.3	Comparison of protocols based on security features. . . . .	64
3.4	Specification of $\mathcal{U}_i$ 's role in HLPSL. . . . .	68
3.5	Specification of $GWN$ 's role in HLPSL. . . . .	69
3.6	Specification of $SN_j$ 's role in HLPSL. . . . .	70
3.7	Specification of proposed protocol's session in HLPSL. . . . .	70
3.8	Specification of proposed protocol's environment in HLPSL. . . . .	71
3.9	Specification of proposed protocol's goal in HLPSL. . . . .	71
3.10	Security verification result of proposed protocol -1 obtained using AVISPA tool. . . . .	72
3.11	Computational Overhead Comparisons . . . . .	73
4.1	User registration phase of A.K.Das's protocol. . . . .	78
4.2	Login, authentication and key sharing phase of A.K.Das's protocol. . . . .	81
4.3	Authentication and session key establishment phase of Choi et al. protocol. . . . .	84
4.4	User registration phase of Park et al.'s protocol. . . . .	86
4.5	$\mathcal{U}_i$ 's authentication and session key sharing phase of Park et al. protocol. . . . .	87
4.6	User registration phase of proposed protocol. . . . .	90

## LIST OF TABLES

4.7	User authentication and session key establishment phase of the proposed protocol. . . . .	93
4.8	User's credential update phase of proposed protocol. . . . .	95
4.9	SPDL specification of the proposed ECC & Fuzzy Extractor based protocol. . . . .	101
4.10	Specification of the user's role in SPDL. . . . .	102
4.11	Specification of the gateway node's role in SPDL. . . . .	102
4.12	Specification of the sensor's role in SPDL. . . . .	103
4.13	Specification of $\mathcal{U}_i$ 's role in HLPSL. . . . .	108
4.14	Specification of $GW_N$ 's role in HLPSL. . . . .	109
4.15	Specification of $SN_j$ 's role in HLPSL. . . . .	110
4.16	Specification of session of the proposed protocol in HLPSL. . . . .	110
4.17	Specification of environment of the proposed protocol in HLPSL. . . . .	111
4.18	Specification of goal of the proposed protocol in HLPSL. . . . .	111
4.19	Security verification result obtained using AVISPA tool. . . . .	112
4.20	Goals: The goals made to analyze the proposed protocol. . . . .	112
4.21	Hypotheses: The assumptions made to analyze the proposed protocol. . . . .	113
4.22	Comparison of protocols on the basis of security features. . . . .	117
4.23	Execution time on computer system for cryptographic operation. . . . .	118
4.24	Execution time and energy consumption on MicaZ sensor node for cryptographic operations. . . . .	118
4.25	Comparison of protocols on the basis of computational cost. . . . .	119
4.26	Comparison of protocols on the basis of energy consumption on sensor node $SN_j$ . . . . .	120
4.27	Comparison of protocols on the basis of communication overhead. . . . .	120
5.1	User Registration Phase . . . . .	127
5.2	Authenticated Key Exchange Phase . . . . .	129
5.3	User's Credential Update Phase . . . . .	130
5.4	The spdl specification of the proposed protocol . . . . .	132
5.5	Comparisons of Protocols based on Security Features . . . . .	136
5.6	Comparison of Protocols based on Computational Performance . . . . .	136
6.1	User Registration Phase . . . . .	141

## LIST OF TABLES

---

6.2	Authenticated Key Exchange Phase . . . . .	143
6.3	User's Credential Update Phase . . . . .	145
6.4	The spdl specification of the proposed protocol . . . . .	147
6.5	Comparisons of Security Features . . . . .	150
6.6	Computational Cost Comparison . . . . .	151
7.1	Summary of User Registration Phase . . . . .	157
7.2	Authenticated Key Exchange Phase . . . . .	160
7.3	Comparisons of Security Features . . . . .	167
7.4	Computational Cost Comparison . . . . .	167
7.5	Security Verification Result of AVISPA tool . . . . .	168
8.1	User Registration Phase . . . . .	176
8.2	$\mathcal{U}_i$ 's authentication and Key Exchange Phase . . . . .	179
8.3	Security Verification Output by AVISPA tool . . . . .	183
8.4	Comparisons of Security Features . . . . .	184
8.5	Time and Energy consumed by MicaZ sensor node . . . . .	184
8.6	Comparison of Protocols based on Computational Performance . . . . .	185
A.1	Fact sheet of SSCC-2015 Publication [2] . . . . .	208
A.2	Fact sheet of Information-2017, MDPI Publication [1] . . . . .	209
A.3	Fact sheet of CIC-2017 Publication [3] . . . . .	210
A.4	Fact sheet of SSCC-2017 Publication [4] . . . . .	211
A.5	Fact sheet of SSCC-2017 Publication [5] . . . . .	212
A.6	Fact sheet of ANTS-2017 Publication [6] . . . . .	213

# List of Notations

**Table 1:** List of notations used in this thesis

Notation	Description
$p, q$	: Appropriate large prime numbers
$F_p$	: Finite field of characteristic $p$
$E$	: Elliptic curve over $F_p$
$G$	: Group of points on $E$
$P$	: Generator or base point on $E$ with order $q$
$n$	: Maximum numbers of Users and Sensor Nodes in WSNs & IoT
$\mathcal{U}_i$	: $i^{th}$ User of WSNs & IoT
$ID_{\mathcal{U}_i}$	: Identity of $\mathcal{U}_i$
$PW_{\mathcal{U}_i}$	: Password of $\mathcal{U}_i$
$BIO_{\mathcal{U}_i}$	: Bio-metric information of $\mathcal{U}_i$
$SC_{\mathcal{U}_i}$	: $\mathcal{U}_i$ 's Smart card
$SN_j$	: $n^{th}$ sensor node
$ID_{SN_j}$	: The identity of $SN_j$
$GWN$	: The gateway node
$Gen(.)$	: The probabilistic generator function of Fuzzy Extractor
$Rep(.)$	: The deterministic reproduction function of Fuzzy Extractor
$h(.)$	: Secure hash function
$Enc_k[s]$	: Symmetric encryption of message $s$ using key $k$
$Dec_k[E_k[s]]$	: Symmetric decryption of $E_k[s]$ using key $k$
$\mathcal{T}$	: The error tolerance limit
$T', T'', T'''$	: Current time at $GWN, SN_j$ and $\mathcal{U}_i$
$\Delta T$	: Maximum transmission delay
$LO$	: $n \times n$ Lower triangular matrix
$UP$	: $n \times n$ Upper triangular matrix
$Mat$	: $n \times n$ Symmetric matrix such that $Mat = LO \times UP$
$LO_{ij}$	: Element of $LO$ matrix at $i^{th}$ row and $j^{th}$ column

## LIST OF TABLES

---

$Mat_{ij}$	: Element of $Mat$ at row $i$ and column $j$
$LO_r(\mathcal{U}_i)$	: Row matrix securely assigned to $\mathcal{U}_i$
$UP_c(\mathcal{U}_i)$	: Column matrix assigned to $\mathcal{U}_i$
$LO_r(SN_j)$	: Row matrix securely assigned to $SN_j$
$UP_c(SN_j)$	: Column matrix assigned to $SN_j$
$\mathbb{Z}_q^*$	: Set of positive integers less than $q$
$x$	: Random number
$\times$	: Point multiplication operator of $E$
$\parallel$	: Concatenation operator
$\oplus$	: Bitwise XOR operator
$P_{BAN}, Q_{BAN}$	: Principals like $\mathcal{U}_i, GWN$ , and $SN_j$
$S$	: Statements like $T_{\mathcal{U}_i}, T_{GWN}, \alpha, \beta$ etc.
$K$	: Secret information or secret key like $K_{GSN_j}, X'_{\mathcal{U}_i}$ etc.
$\#(S)$	: $S$ is fresh and it has not been transferred before.
$P_{BAN} \mid \equiv S$	: $P_{BAN}$ believes $S$ , or $P_{BAN}$ believes $S$ is true.
$P_{BAN} \triangleleft S$	: $P_{BAN}$ has obtained a information having $S$ and it can repeat or read $S$
$P_{BAN} \mid \sim S$	: $P_{BAN}$ once said $S$ . $P_{BAN}$ sent a information containing $S$ and it could be a old or fresh information.
$P_{BAN} \Rightarrow S$	: $P_{BAN}$ has jurisdiction over $S$ . That is $P_{BAN}$ 's beliefs about $S$ should be trusted
$P_{BAN} \stackrel{S}{\rightleftharpoons} Q_{BAN}$	: $S$ is a secret data which is known only to $P_{BAN}$ or $Q_{BAN}$ and perhaps to the trusted principals
$\langle S \rangle_{S1}$	: $S1$ is a secret value which provides the identity of whoever produces $\langle S \rangle_{S1}$

# Chapter 1

## Introduction

“The key to growth is the introduction of higher dimensions of consciousness into our awareness.” Lao Tzn

This chapter introduces Wireless Sensor Networks (WSNs) & Internet of Things (IoT), its hardware constraints, limitations, topology and applications. Later, this chapter describes the general security requirements, functionality requirements, technologies, cryptography concepts and tools used for user authentication and key establishment protocols of WSNs & IoT. The chapter presents the research motivation, objective, contributions and organization of the thesis.

### 1.1 Introduction

In a wireless sensor network (WSN), a large quantity of small-scale computing physical devices, called sensor nodes or motes, are distributed in a target area. These sensor nodes are used for sensing relevant information and communicating this sensing information to the nearby gateway node for additional processing. Sensor nodes are generally deployed densely in close proximity to the event to be observed. A sensor node is a node in a WSN & IoT that is suitable for data processing, collecting sensory data and communicating with different connected sensor nodes. Sensor nodes communicate with each other through short-range radio interfaces. The gateway node is a computationally well-equipped device in the WSNs & IoT, whereas the sensor nodes are

resource-starved. The sensor nodes are usually scattered in a terrain area (i.e., deployment region or target field), and all of the distributed nodes has the abilities to receive data and route data back to the gateway node through a multi-hop infrastructure-less interaction with other sensor nodes.

In this chapter, we present the motivation factor for the research activity carried out in the thesis. The chapter begins with the introduction of user authentication for WSNs & IoT based on various challenges. Then, we demonstrate the desired security requirements for authentication, storing, and sharing data. Finally, we discuss the problem statement, objectives, and our contributions for designing a secure authenticated key establishment mechanism.

### 1.1.1 Wireless Sensor Networks and Internet of Things

Up-to-date improvements in the micro-electro-mechanical operation facilitate the reproduction of low-cost sensor nodes including small-scale sensing module, a radio frequency transceiver, a small processing module for inadequate computation, small-scale memory and a temporary power unit. For example, a sensor node can have a light, pressure, temperature and humidity sensors with 7.7 MHz 8-bit ATmega 128 processor, 128 K byte ROM, 512 K byte EEPROM, 4 K byte RAM, and 2 AA battery. The sensing unit may consist of some sensors with analog to digital converters (ADCs). Those sensors can estimate the variation in environmental parameters such as light, temperature, pressure and humidity. The analog signals generated by the sensor node based on the evaluated ecological or physical parameters can be converted into the digital signal applying ADC. Later, the digital signals can be supplied into the processing component to complete the important estimation on fresh data, and the transceiver unit communicates with its neighbouring sensor nodes. Nowadays, we discover sensors in our watches, smartphones, vehicles, cities, and gadgets in offices, homes, and industries which unite our world stronger than we ever thought feasible. The traditional specializations of WSNs, control systems, embedded systems and automation (including industry and building automation, smart home, smart city) contribute to facilitating the IoT. The advancements in IoT technology facilitate wearable gadgets which broadly incorporate health, convenience and recreation requirements.



A WSN [2] or IoT [9] may consist of a large number of distributed sensor nodes proficient of accumulating data of their surroundings for particular users, interacting with the neighbouring sensor nodes utilising wireless communication and routing the data to the gateway node having trusted high-performance computing devices. Remarkable essential features of WSNs & IoT are as follows:

- The sensor nodes of WSNs & IoT suffer by energy limitations, memory constraints, unstable communications, higher latency in communication and unattended operation of networks.
- The topology of IoT & WSNs can change very frequently.
- The sensor node can be disposed of densely in WSNs & IoT field.

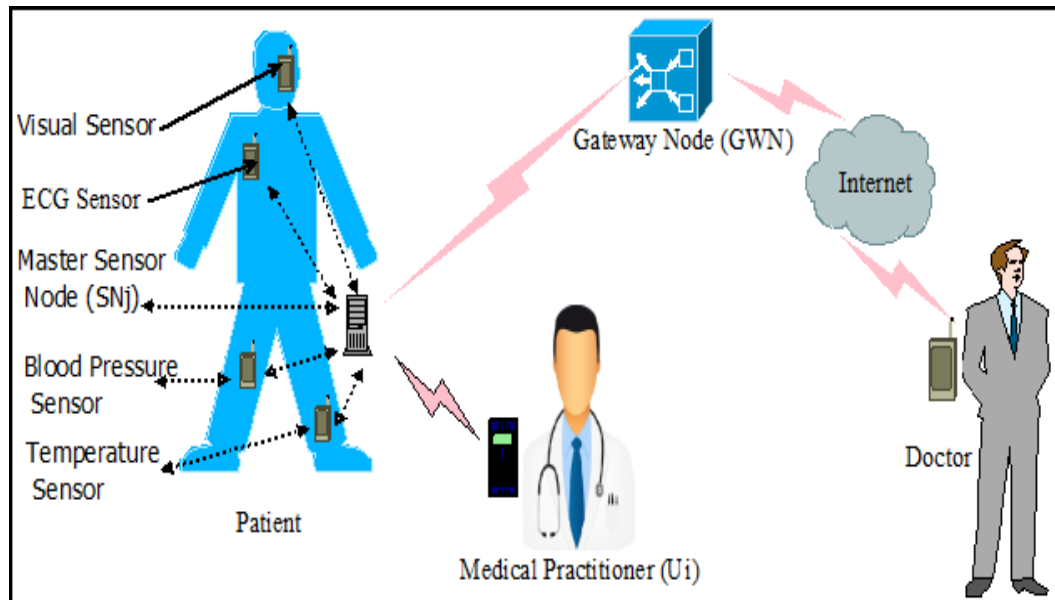
The IoT intends to overcome the gap within the real world and its characterization in the digital world. The word things refer to an object that has sensors connected to it and can transfer data to the internet, where it can be processed, evaluated and utilised to make decisions, one such example is medical healthcare protocol [10].

An instance of the medical healthcare practice for monitoring patient's disease and improvement by authentic medical practitioners and doctors utilising a wireless body area network (WBAN) is shown in Figure 1.1. The sensor nodes are attached or located in the patient's body for measuring various parameters like ECG, blood pressure, temperature, visual straight, etc. The data of these measured parameters from multiple sensor nodes are communicated to a master sensor node. The master sensor node processes the data regionally and sends to the gateway node. Only the genuine medical practitioners and doctors are permitted to access the confidential and real-time data of high-profile patients from the master sensor node and the gateway respectively.

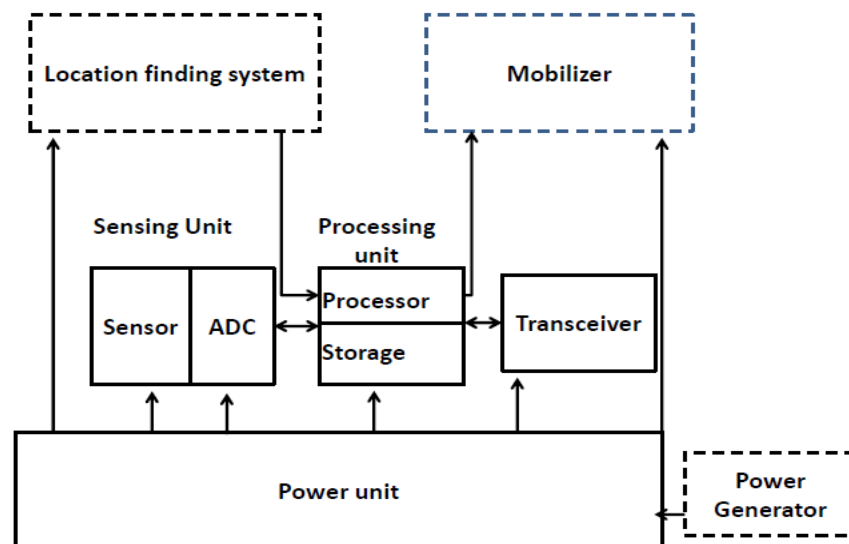
### 1.1.1.1 Hardware constraints of WSNs

A sensor node consists of four basic components: (i) a sensing unit, (ii) a processing unit, (iii) a transceiver unit, and (iv) a power unit, as shown in Figure 1.2.

Sensor node may also have additional components like external memory, a location finding system, power generator, and mobilizer. Sensing units are basically of two types: (i) passive sensors like camera and (ii) active sensors like radar. Sensing units are usually composed of two sub-units: sensor and ADC (analog to digital converter).



**Figure 1.1:** Wireless body area network (WBAN).



**Figure 1.2:** Components of a sensor node [2]

After sensing, the analog data are converted by ADC to digital data, and then fed to the processing unit for further processing. Transmission unit helps to connect the sensor node to the network. Power unit is another important component of a sensor node, which is in general battery powered. In most of the sensor networks, routing techniques and sensing tasks require knowledge of location with high accuracy. Thus, it is common that a sensor node will have a location finding system. A mobilizer may sometimes be required in order to move a sensor node when it is required to carry out the assigned tasks.

### 1.1.1.2 Sensor network limitations

In this section, we discuss the limitations of WSNs & IoT that affect its basic operations such as routing and deployment of sensor nodes along with the security features which are required for secure communication. It is required to overcome these challenges to build an efficient WSNs & IoT.

**Limited resources:** Sensor nodes are equipped with limited memory for storage and low computing capabilities which make it difficult to perform memory intensive operations and computationally costly operations.

**Limited communication capabilities:** Sensor nodes have short-range communication capability using radio transmission waves to transmit the sensed data to neighbor sensor nodes or to the nearby base station. Also, sensor nodes have less bandwidth which makes it difficult to transmit large amount of data.

**Limited lifetime:** Sensor nodes operate on battery and may go offline due to battery draining out. Hence, the security mechanisms implemented for the sensor nodes must be efficient to use less power for computation and transmission.

**Node capture:** Sensor nodes operate in unattended fashion and when deployed in mission critical environments like battle fields which are prone to be captured by the adversary. The data stored in the sensor node's memory can be read by the adversary if sensor nodes do not have tamper-resistant hardware.

**Lack of knowledge on post-deployment configuration:** Sensor nodes are generally deployed randomly in the target network (e.g., using drones to deploy sensor nodes in a large crop field). This makes the WSNs & IoT difficult to know

the deployment configuration by knowing the neighbors of each sensor node. Though the sensor nodes are deployed manually in the target field, it becomes expensive to pre-determine the location of each sensor node in a large WSNs & IoT containing large number of nodes.

### 1.1.1.3 WSNs Topology

The lifetime of wireless sensor network depends on three phases: (i) pre-deployment phase, (ii) post-deployment phase, and (iii) redeployment of additional nodes phase. Topology of WSN is dynamic in nature and may change phase-wise. Sensor nodes may expire due to battery-energy consumption and also new sensor nodes may be needed to deploy to the network in order to replace battery-exhausted nodes and malicious nodes. The details of three phases are given below:

**Pre-deployment phase:** The sensor nodes can be placed all over the target terrain area using one of the following procedures:

- Randomly by throwing sensor nodes from an airplane or vehicle.
- Through an artillery shell, rocket or missile.
- Planned way like grid-based deployment by human or robot.

**Post-deployment phase:** After the deployment, the topology of the sensor network may change due to the following reasons:

- Coverage problems of sensor nodes due to jamming, noise, etc.
- Irregularities in the sensor field like obstacles.
- Sensor node's battery-energy constraints.
- Sensor node's malfunctioning.

**Redeployment of additional nodes phase:** Redeployment and replacement of sensor nodes are inevitable due to the following reasons:

- Sensor node can be physically captured or compromised by an attacker from the target field.
- Due to battery-energy constraints, some nodes can expire.

### 1.1.2 Applications of WSNs and IoT

In this section, we discuss some of the popular applications of WSNs & IoT. WSNs & IoT has gained significant attention in the recent years in real-time applications ranging from critical applications like battlefield surveillance and border monitoring, health-care applications like remote medical diagnosis, smart homes to add intelligence to the home equipment for better comfort and security, etc. As the sensor nodes operate in the unattended environment and communicate using wireless medium, it is essential to protect the sensed information and ensure secure communication to transmit the data to the gateway nodes for further processing. An adversary in a wireless medium not only can eavesdrop but also can intercept and modify the legitimate traffic. Hence, security becomes the primary concern in WSNs & IoT, and many of the security protocols do not merely work given the resource-constrained nature of sensor nodes concerning computing capabilities and storage resources. In the following section, we discuss some prominent WSNs & IoT applications where the security is essential.

#### 1.1.2.1 Military and battlefield surveillance

Significant areas like geographical borders of territory and battlefields need to be monitored for any suspicious activity. It is risky to deploy army or human patrolling as such regions are prone to enemy attacks. Hence, it is beneficial to use sensor networks in such critical areas by employing sensors for surveillance. Sensor nodes can be deployed using military trucks or low flying planes, such as drones, to detect various kinds of information such as tracking military vehicles, identifying the trajectory of missiles and locating snipers, thereby checking for any violation of territory laws. Upon sensing any such information, the sensor nodes transmit the data to the gateway node, and the gateway node then can further analyse more on the received data. Upon receiving similar information from multiple nodes, appropriate decisions can be taken to neutralise the situation.

#### 1.1.2.2 Smart homes

The value of sensors in home appliances has been overgrowing over the past few years. The purpose of sensors is expanded to control electronic devices like air-conditioner, geyser, switching on/off lights, locking doors and raising alarms in case of trespassing

remotely using mobile (smart) phone. The use of sensors in-home devices supports people to enrich the lifestyle, taking care of old people and engaging kids by interacting with them. The ZigBee technology is used for monitoring and collecting data from various sensors in the automation of home appliances. The data is further processed by a microprocessor and is displayed in a convenient way to control them.

### 1.1.2.3 Other applications

There are various other useful applications of WSNs & IoT in the practical world. WSNs & IoT is remarkably beneficial in environmental monitoring which involves tracking the movement of birds or animals, forest fire detection and detecting foods. Precision agriculture is another application where WSNs & IoT are useful to monitor the pests and soil strength at different places of the crop for better production. The gaming field has taken a great leap with the attachment of sensors to improve interacting capabilities of toys with humans by responding to their touch, speech and gestures. Some of the other commercial applications include interactive museums wherein the famous personalities communicate with visitors by responding to their questions, monitoring product quality, inventory management and monitoring and detecting crimes. Some important applications of WSNs & IoT are shown in following Figure 1.3.

### 1.1.3 General security requirements of WSNs and IoT

Security requirements in WSNs & IoT are very similar to those of ad-hoc networks. WSNs and IoT have the following general security requirements

- **Authentication:** It is required to validate a communicating node who it claims to be. A proper authentication mechanism is necessary to validate gateway nodes, cluster heads, sensor nodes and registered users in the WSNs & IoT before granting a resource or before sending any sensitive information.
- **Integrity:** Any message from an authorized sender to an intended recipient must not be altered during the transit.
- **Confidentiality:** It must be ensured that the data owing in the network is understood by intended recipients only. In WSNs & IoT, the confidentiality or privacy



- **Non-repudiation:** Preventing malicious sensor nodes of WSNs to hide their important activities.
- **Authorization:** In WSNs & IoT where access control mechanisms are employed, it is required to authorize an authenticated user to check if he/she has required privileges to access the requested resource. Unauthorized access leads to an under-privileged user accessing an elevated resource. For example, a normal user must not abandon a sensor node from the WSNs & IoT which the system administrator can only perform.
- **Information Freshness:** Freshness is an important feature in WSNs & IoT to ensure that the received data is freshly generated by the genuine participant and is not a replay message by an adversary. A periodic counter, current timestamp or a random nonce can be embedded in the message to verify the freshness of a message.
- **Secure localization:** At times, it is required to locate the accurate location of a sensor node in the target field. For example, a WSN deployed to locate the faults in the target field is required to identify the location of the sensor node reporting a fault. It is possible that an adversary can manipulate the location of a sensor node by modifying the signal strength or by replaying of messages.

Apart from these security requirements, the forward and backward secrecy need to be considered as new sensors can be deployed in the network and old sensors may fail due to energy problems.

- **Forward secrecy:** It implies if a node or an entity leaves the WSNs & IoT, it must not be able to read any communication owing in the network after its departure.
- **Backward secrecy:** It implies if a new node joins the network, it must not be able to decrypt or read the data communication that is owed before its introduction.

## 1.2 Authentication in WSN and IoT

A user who wants to avail the services of a WSNs & IoT must get authenticated by the WSNs & IoT. The gateway nodes, present in the WSN & IoT, collect the real-time information from sensor nodes and stores it in their memory which the legitimate



users can query. However, since the *GWN* collect the data from sensor nodes in regular intervals, the data present at the *GWN* may not be the real-time data and such data can majorly be useful for statistical purposes or for analytics only. Hence, it is also required for the users to communicate with the sensor nodes directly to collect the real-time information. To communicate with the sensor nodes directly, the user needs to authenticate against the gateway nodes as well as the sensor nodes to ensure the secure communication. Given the resource-crunchy nature of sensor nodes, it is essential to design efficient authentication protocols with no compromise in security.

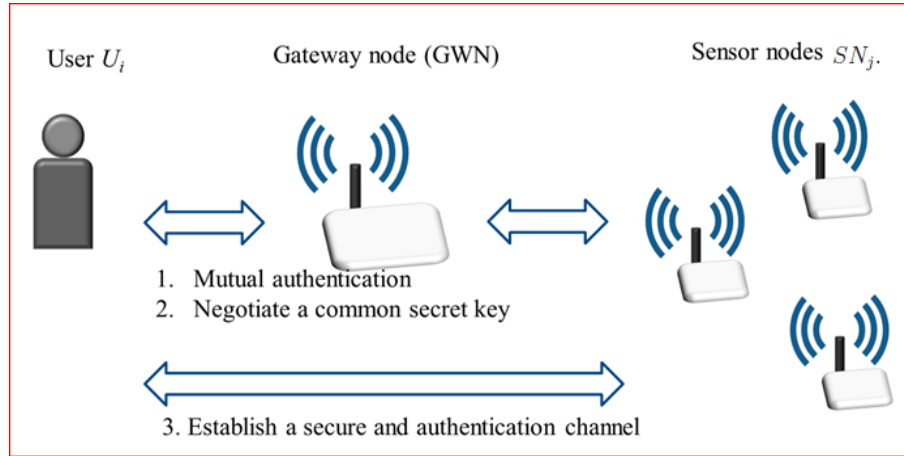
Beneson et al. [11] illustrated the inside and the outside security requirements in WSNs & IoT as:

- Inside security represents secure data communication between the sensor node  $SN_j$  and secure data communication between the sensor node  $SN_j$  and the gateway node *GWN* of WSNs & IoT.
- Outside security represents secure data communication between outside user  $\mathcal{U}_i$  and sensor nodes  $SN_j$  of WSNs & IoT.

Authentication in WSNs & IoT [12] is a remarkable specification which is a function of both outside and inside security of WSNs & IoT. If a secure user authentication technique is not considered, then an adversary  $\mathcal{A}$  can often produce forge data packets and influence sensor node to process those data packets to make sensor nodes resources exhausted. A forge data can cause sensor nodes to receive and transfer faulty messages which in turn makes sensor nodes of WSNs & IoT vulnerable to various major security attacks. Authentication in WSNs & IoT can be categorize into following manner:

- Authentication of the gateway node *GWN* to the sensor node  $SN_j$  or in between sensor nodes.
- Authentication of user  $\mathcal{U}_i$  to the sensor nodes  $SN_j$  or to the gateway node *GWN*.

There are various techniques proposed to address the gateway to sensor nodes authentication and sensor nodes to other sensor nodes authentication. In this thesis, we emphasize on the authentication of user  $\mathcal{U}_i$  to the sensor nodes  $SN_j$ . In this model of authentication, outside user  $\mathcal{U}_i$  are authenticated to a sensor node  $SN_j$  with the help of gateway node *GWN* (as shown in Figure 1.4).



**Figure 1.4:** Mutual Authentication and Session Key Establishment [4]

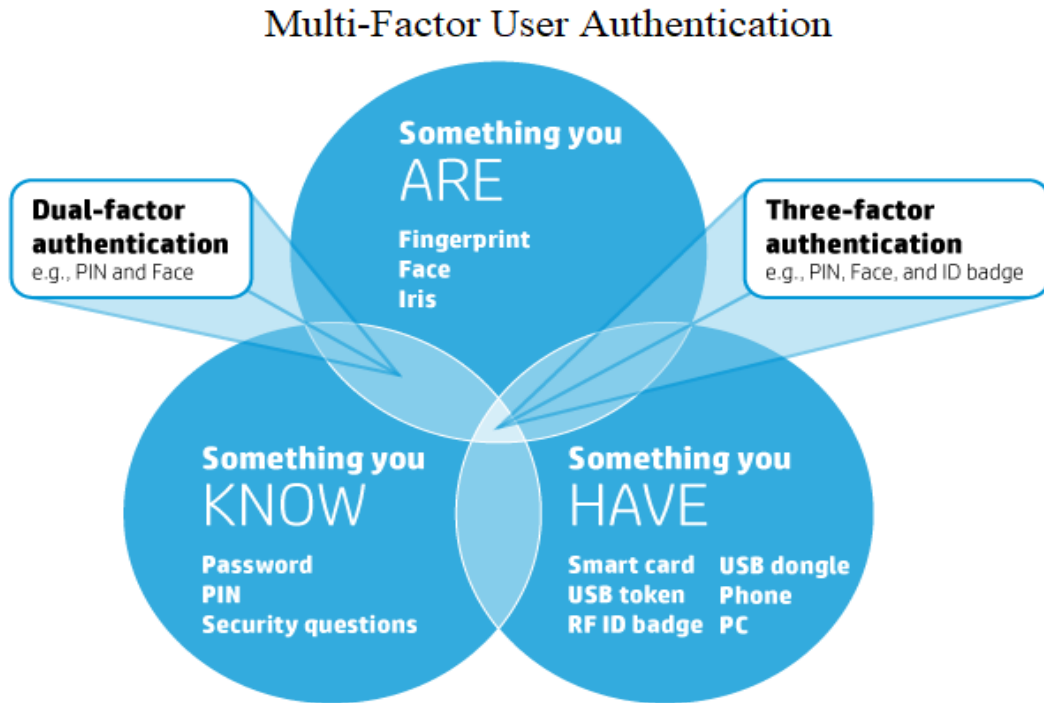
Formerly, just one factor was utilised to authenticate the user. By that moment, Single-Factor Authentication (SFA) was often used by the society due to its uniformity and user-friendliness. As an illustration, the use of a password (or a PIN) to verify the claim of the user ID could be considered. This is the weakest level of authentication.

By distributing the password, one can compromise the account instantly. Further, an illegal user can also try to get access by utilising the dictionary attack [16], rainbow table, or social engineering procedures. Generally, the minimum password complexity condition is to be considered while using this type of authentication.

Moreover, it was recognized that authentication with merely a single factor is not reliable to provide sufficient protection due to many security threats. As a fundamental step forward, Two-Factor Authentication (2FA) was proposed that pairs the typical data (user-name and password incorporation) with the factor of personal possession, such as a smart card or a phone. Now, three types of factor combinations are convenient to connect a person with the authorized credentials:

- Knowledge factor - something that the user remembers, such as a password or a secret;
- Ownership factor - something that user holds, such as smart cards, smartphones, or additional tokens;
- Biometric factor - something that user is, i.e., biometric information or behaviour pattern.

Subsequently, Multi-Factor Authentication (MFA) [13] was proposed to provide a higher level of safety and facilitate continuous protection of computing devices as well as other critical services from unauthorized access by using more than two categories of credentials. For the most part, MFA is based on biometrics (as shown in Figure 1.5), which is automated recognition of individuals based on their behavioral and biological characteristics. This step offered an improved level of security as the users were required to present the evidence of their identity, which relies on two or more different factors.



**Figure 1.5:** Authentication Factors [5]

Our proposed user authentication protocols for WSNs & in IoT has the following phases:

**Pre-Deployment Phase** This phase allows the WSNs and IoT parameters to be chosen by the *GWN*. Before the sensing nodes are deployed or installed, they need to be registered with the *GWN*. The *GWN* then loads the necessary secret credentials before deployment.

**User Registration Phase** To access information (service) from certain sensing nodes, a user  $\mathcal{U}_i$  needs to register with the *GW**N*.  $\mathcal{U}_i$  first provides his/her credentials (e.g., identity, password and bio-metric information) secretly to the *GW**N* and the *GW**N* issues a smart card securely to  $\mathcal{U}_i$ .

**Login, Authentication and Key Establishment:** In this phase,  $\mathcal{U}_i$  enters his/her credentials, and these are validated by smart card, a login request message is formed and sent to the *GW**N* via open channel. After receiving the login request message, the *GW**N* first validates it and if the validation passes, the *GW**N* sends an authentication request message to the sensing node being accessed, say  $SN_j$ .  $SN_j$  then validates the received message and dispatches the authentication reply to  $\mathcal{U}_i$ .  $\mathcal{U}_i$  also validates the received message from  $SN_j$ . Only after mutual authentication between  $\mathcal{U}_i$  and  $SN_j$  a session key  $SK_{ij}$  is established between them. Both later use  $SK_{ij}$  secure communication.

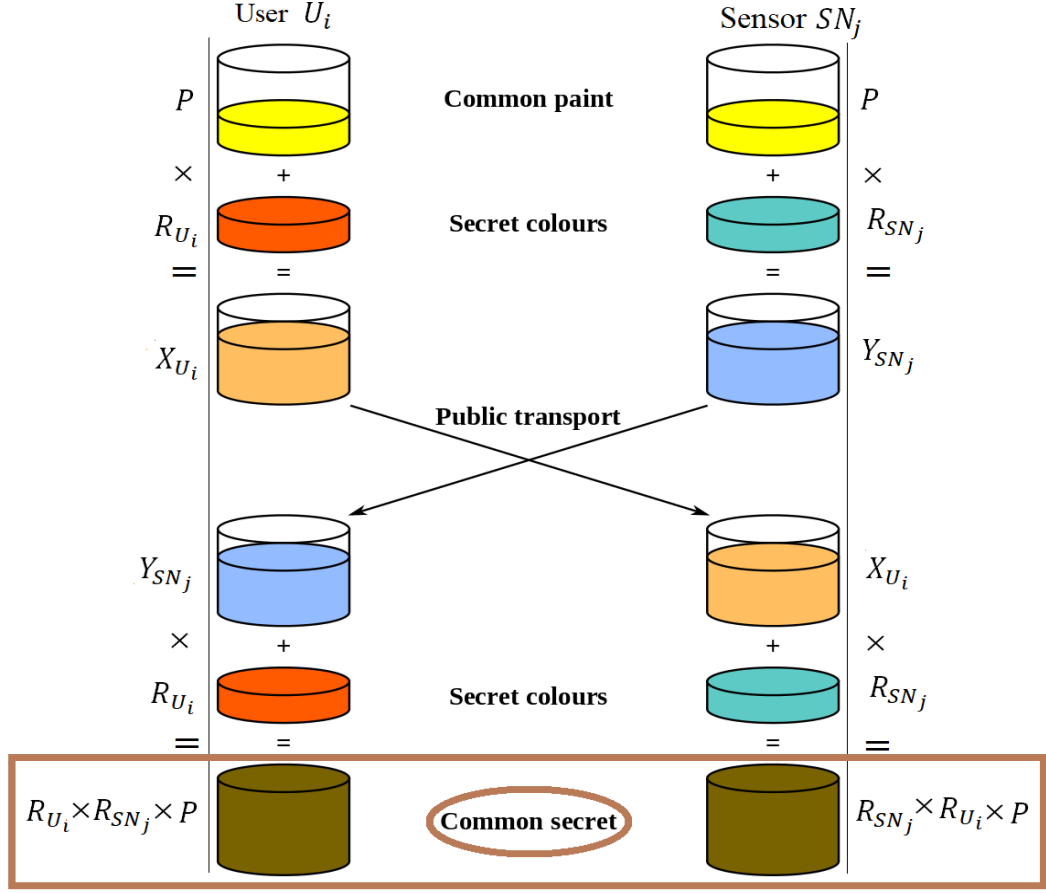
**User's Credential Update Phase** This phase is needed only when a legitimate user  $\mathcal{U}_i$  wishes to update his/her password and biometrics. It is desirable that  $\mathcal{U}_i$  should not involve the *GW**N* for this activity and hence, this phase can be entirely executed locally without the involvement of the *GW**N* by  $\mathcal{U}_i$ .

### 1.2.1 Key Establishment in WSNs and IoT

Key establishment is a technique in cryptography through which cryptography keys are exchanged among two people, enabling the usage of a cryptography algorithm. For instance, DiffieHellman key exchange establishes a shared secret between two parties that can be used for securely transferring data over a public network. The following conceptual diagram demonstrates the general idea of the key establishment by utilizing colours instead of very big numbers.

For key establishment which is a significant building block for every security mechanisms, the following objectives are necessary:

- Resource consumption efficiency: energy, memory space, and processing capabilities of sensor nodes must be considered.



**Figure 1.6:** Key Establishment in WSNs and IoT

- Scalability: the network may contain thousands of sensor nodes, the key management solution has to be adaptive. So, post-deployment of nodes must also be considered.
- Backward and Forward secrecy: when a sensor node joins the network (i.e., post-deployment) or leaves it (i.e., consumes the totality of its energy), the cryptographic keys of this sensor node cannot be used to read any previously exchanged (for backward secrecy) or future exchanged message (for forward secrecy).

### 1.2.2 Assumptions of the Threat Model for WSNs & IoT

- Sensor node may not fix up with the tamper-resistant device, and if a node is captured by an adversary, all the prominent and confidential information stored in its memory can be accessed by the adversary. If the sensor nodes are tamper-resistant, the adversary can know the information saved in the memory by measuring the energy consumption of the captured sensor nodes.
- The gateway node is trusted, and it works both as an authentication as well as a key distribution centre.
- The adversary  $\mathcal{A}$  can intercept the public data communication channel, replay the previously forwarded packets and insert packets.
- The adversary  $\mathcal{A}$  can take the smart card  $SC_{\mathcal{U}_i}$  of user  $\mathcal{U}_i$  and it can obtain the security information stored in the smart card by simple and differential power analysis procedures [14].
- We believe that the WSNs & IoT consist of several users (with the smart card which can be captured or stolen by the adversary  $\mathcal{A}$ ), hundreds of sensor nodes ( $\mathcal{A}$  can capture it) and the trusted gateway node.
- The processed data of the sensor nodes are accumulated periodically at the gateway node  $GWN$ . The collected data may not always be real-time and fresh at  $GWN$ . Consequently, the genuine user should be permitted to access the data instantly from the sensor node  $SN_j$  to make prompt decision for secure and real-time applications of WSNs and IoT.

### 1.2.3 Security specifications for User Authentication in WSNs and IoT

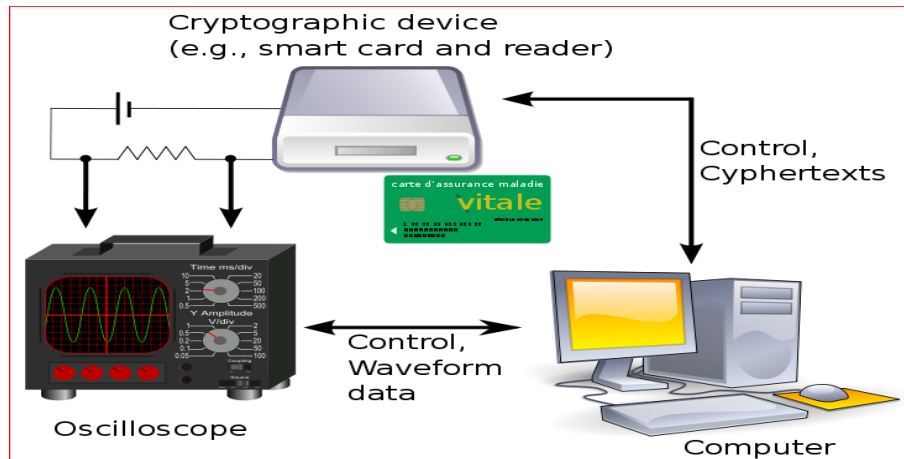
In this section, we introduce various security attacks which a user authentication protocols for WSNs & IoT should resist and different useful security characteristics that the protocol should provide a friendly and secure authentication mechanism.

Before reviewing the attacks, it is assumed that an adversary may hold complete command over the network with the following abilities.

- An adversary  $\mathcal{A}$  can intercept, remove or change, and insert any message across the public communication channels.
- An adversary  $\mathcal{A}$  can either hack passwords or steal users smart-card and utilises secrets saved in the smart card, however not entirely at the same time.
- An adversary  $\mathcal{A}$  may intercept all the messages at any time.
- An adversary  $\mathcal{A}$  may either hack passwords or steal users smart-card and utilize secrets stored in smartcard, but not all at the same time.
- An adversary  $\mathcal{A}$  can compromise the sensor node and derive all parameters saved in it.

According to the security specifications for user authentication in WSNs and IoT, the following attacks must be prevented:

**Stolen smart card attack:** Though smart-cards are built with tamper-resistant hardware, the contents of the smart card can be read using power analysis techniques [14] as shown in following figure 1.7. The authentication protocols, which use smart cards for authentication, must be careful to protect sensitive information though the information stored in the smart card is revealed to a malicious user.



**Figure 1.7:** Differential power analysis [6]

**Sensor node capture attack:** Sensor nodes often operate in hostile and unattended environment and they are prone to physical node capture by an attacker. The

effect of the node capture attack refers to the portion of the network that is affected when a node or a set of nodes in the network are captured. In other words, we define the probability of an adversary to decrypt the secure communication between a non-compromised sensor node  $SN_j$  and a gateway node  $GW_N$  when  $m$  nodes are captured in the network as  $Pr(m)$ . If  $Pr(m) = 0$ , such an authentication protocol is said to be unconditionally secure against sensor node capture attack.

**Impersonation attack:** During this attack, a malicious user, being an adversary, attempts to act as a legitimate participant in the secure communication. This attack is feasible only when a malicious user can create an original message with the data known to him/her. The malicious user can try to impersonate a gateway node, a sensor node and a legitimate user, which corresponds to gateway node impersonation attack, sensor node impersonation attack and user impersonation attack.

**Man-in-the-middle attack:** Under this attack, an adversary  $\mathcal{A}$  may secretly transfer and modifies the message communication between two individuals who assume that they are talking with each other. The attacker can alter a valid message to generate another original message of his/her own choice or can entirely make a new valid message. These kinds of attacks are severe as the original parties do not know that they are being victimized. Authentication protocols must ensure that they are not prone to man-in-the-middle attacks by guaranteeing mutual authentication.

**Denial-of-Service attack:** This attack indicates flooding a participant or a network beyond its capacity so that it can not perform its standard functions. The main purpose of this attack is to impact the availability of service so that it cannot serve legitimate requests. Sometimes, the denial of service can also be generated by hardware failure, change in environmental situations and software bugs.

**Privileged insider attack:** This type of attack is performed by a system administrator or an insider of a gateway node by elevated privileges. In general, it is assumed that the messages transmitted while the registration process is likely to be accessible to a privileged insider. A user authentication system must take precise measures to be resilient against privileged insider attack.



**Replay attack:** Toward this attack, an attacker captures one or more packets over the network from a legitimate participant and tries to re-send the packets to the destined party. In this process, the attacker attempts to deceive the recipient by reusing the information during the run of the protocol. The attacker does not need any additional knowledge to originate this attack.

**Password guessing attack:** Toward this attack, a malicious user attempts to guess the password of a legitimate user by using online or offline guessing methods. An online password guessing attack is performed when no information of password is available, and it is carried out by trying passwords. This attack is noisy, slow and infeasible for most of the times. An online password guessing attack is carried out when password hashes are available. This attack, as the name indicates, is performed online on the attacker site, for example, by computing hash of a random password and verifying whether it matches with any of the password hashes available.

**Gateway Bypass Attack:** In protocols where users transmit their login information to the sensor node, the sensor node demands the gateway to confirm the legitimacy of the user during the authentications phase. If some legitimate but malicious user or an adversary can successfully pass the authentication method without allowing the gateway to play its role, then a gateway bypass attack occurs.

### 1.2.4 Functionality requirements of user authentication protocols for WSNs & IoT

The primitive functionality requirements of a user authentication protocol for WSNs & IoT are as follows.

- The authentication protocol must be efficient in terms of computation, communication and storage as the sensor nodes are resource-starved devices.
- The sensor node's registration process should be done in offline mode by the *GW* due to resource limitations of sensor nodes.
- The authentication protocol must be designed in such a way that capturing a sensor node must not compromise the security of the entire WSN. This is an important feature as sensor nodes often operate in hostile environments.

- The authentication protocol must support dynamic addition of sensor nodes to the WSNs & IoT. This is an essential property as sensor nodes may run out of battery or may fail due to a hardware failure, and it also requires to deploy new nodes in the target field.
- A genuine user must be able to change his/her password locally without contacting the gateway nodes in the WSNs & IoT.
- The authentication protocol must be scalable to support large number of sensor nodes in the target network.

### 1.3 Technologies, Cryptography Concepts, and Formal Security Verification Methods used in the thesis

In this section, we discuss some mathematical preliminaries in order to design and analyze our proposed protocols. We first discuss the security properties of one-way hash function. We discuss the elliptic curve and its properties, the rules for adding points on an elliptic curve, the rules for scalar point multiplication in the Elliptic-curve DiffieHellman (ECDH) problem. Then we discuss the concepts of Fuzzy Extractor, Bloom Filter, Chinese Remainder Theorem and Symmetric Hash Function. We finally describe the formal security analysis techniques based on AVISPA Tool, Syther Tool, BAN Logic and Random Oracle Model.

The remarkable elementary cryptography methods applied in the security analysis of current protocols and also in our proposed protocols are described as follows:

#### 1.3.1 Secure Hash Function

**Definition 1.3.1.** *Secure Hash Function [15]: A function  $h : In \rightarrow Out$ , with a binary string  $s \in In \{0, 1\}^*$  of inconsistent range as input and a binary string  $d \in Out \{0, 1\}^m$  of fixed range  $m$  as an output, is a secure hash function if the following stipulations exist:*

- *Adversary  $\mathcal{A}$ 's advantage to determine the collision is negligible i.e,  $Adv_{\mathcal{A}}^h(t_1) = Pr[(s, s') \leftarrow_R \mathcal{A} : s \neq s', h(s) = h(s')] \text{ and}$*

- $Adv_A^h(t_1) \leq \tau$ , for any adequately small  $\tau > 0$ .

where  $(s, s') \leftarrow_R$  symbolizes that the pair  $(s, s')$  is randomly decided by  $\mathcal{A}$  and  $Pr$  represents the probability of the event  $(s, s') \leftarrow_R \mathcal{A}$  including performance time  $t_1$ .

### 1.3.2 Symmetric Hash Function

**Definition 1.3.2.** *Symmetric Hash Function [16]: A hash function which is invariant to the order of the given input can be considered as a symmetric hash function. For example : The following function (1) is not a symmetric hash function, whereas function (2) is symmetric hash function.*

$$h_{asymmetric}(c_{1j}, c_{2j}, \dots, c_{nj}) = k_1 c_{1j} + k_2 c_{2j} + \dots + k_n c_{nj}, \quad (1)$$

Where  $k_1 \neq k_2 \dots \neq k_n$  and

$$h_{mj}(c_{1j}, c_{2j}, \dots, c_{nj}) = c_{1j}^m + c_{2j}^m + \dots + c_{nj}^m \quad (2)$$

If the order of the input  $(c_{1j}, c_{2j}, \dots, c_{nj})$  is changed to  $(c_{2j}, c_{1j}, \dots, c_{nj})$ , the function  $h_{asymmetric}(c_{1j}, c_{2j}, \dots, c_{nj})$  gives a different value where as  $h_{mj}(c_{1j}, c_{2j}, \dots, c_{nj})$  remains unchanged. Therefore,  $h_{mj}(c_{1j}, c_{2j}, \dots, c_{nj})$  is a symmetric hash function.

### 1.3.3 Secure Encryption Algorithm

**Definition 1.3.3.** *Secure Encryption Algorithm [15]: For each probabilistic, polynomial time adversary  $\mathcal{A}$ , an encryption protocol  $Enc$  is supposed to be IND-CPA (indistinguishability of encryption and chosen plaintext attack) secure if  $Adv_{Enc, \mathcal{A}}^{IND-CPA}$  is negligible. Where  $Adv_{Enc, \mathcal{A}}^{IND-CPA}(t_2) = 2Pr[\mathcal{A} \leftarrow O_k; (b_0, b_1 \leftarrow \mathcal{A}); \tau \leftarrow_R \{0, 1\}; \gamma \leftarrow_R O_k(b_\tau) : \mathcal{A}(\gamma) = \tau] - 1$  denotes the advantage function of  $\mathcal{A}$ . Where  $\tau \leftarrow_R \{0, 1\}$  represents that the bit  $\tau$  is randomly chosen from  $\{0, 1\}$  and  $t_2$  represents the execution time.*

### 1.3.4 ECDH Key Establishment

**Definition 1.3.4.** *Elliptic Curve Diffie-Hellman [17]: Consider a prime number  $p > 3$ , the elliptic curve  $E_p(a, b)$  over the finite field  $\mathbb{Z}_p^*$  that is interpreted through the solutions  $(x, y) \in \mathbb{Z}_p^* \times \mathbb{Z}_p^*$  of the equation  $y^2 = x^3 + ax + b$ , including a point  $\mathcal{O}$*

of infinity, where  $4a^3 + 27b^2 \neq 0 \pmod{p}$ . If  $P$  be a generator or a base point of a cyclic subgroup  $G$  of the elliptic curve  $E_p(a, b)$  estimated over the finite field  $\mathbb{F}_p^*$ , i.e.,  $G = \langle P \rangle$ , the elliptic curve Diffie-Hellman (ECDH) key exchange can be expressed as follows:

First,  $\mathcal{U}_i$  and  $SN_j$  agree on a generator point  $P$  and choose their private key as  $R_{\mathcal{U}_i}$  and  $R_{SN_j}$  respectively. Then, they create and exchange their public keys as  $X_{\mathcal{U}_i} = R_{\mathcal{U}_i} \times P$  and  $Y_{SN_j} = R_{SN_j} \times P$ . Finally,  $\mathcal{U}_i$  and  $SN_j$  calculate the common secret key as  $R_{\mathcal{U}_i} \times (R_{SN_j} \times P)$  and  $R_{SN_j} \times (R_{\mathcal{U}_i} \times P)$  respectively. Where  $R_{\mathcal{U}_i} \times (R_{SN_j} \times P) = R_{SN_j} \times (R_{\mathcal{U}_i} \times P)$  and it is intractable to obtain  $R_{\mathcal{U}_i}$  and  $R_{SN_j}$  for an adversary  $\mathcal{A}$  having  $X_{\mathcal{U}_i}$  and  $Y_{SN_j}$ . i.e.,

The advantage in determining  $R_{\mathcal{U}_i}$  is represented by  $\text{Adv}_{\mathcal{A}}^{\text{ECDH}}(t_3) = \Pr[(R_{\mathcal{U}_i}, P) \leftarrow_R \mathcal{A} : X_{\mathcal{U}_i} = R_{\mathcal{U}_i} \times P]$ . Where  $\text{Adv}_{\mathcal{A}}^{\text{ECDH}}(t_3) \leq \tau$ , for any sufficient small  $\tau > 0$  and  $(R_{\mathcal{U}_i}, P) \leftarrow_R \mathcal{A}$  represents the pair  $(R_{\mathcal{U}_i}, P)$  is randomly chosen by  $\mathcal{A}$  including execution time  $t_3$ , such that  $X_{\mathcal{U}_i} = R_{\mathcal{U}_i} \times P$ .

### 1.3.5 Fuzzy Extractor

**Definition 1.3.5.** Fuzzy extractor [18] is technique based on cryptography for securely authenticating a user  $\mathcal{U}_i$  utilizing bio-metric credentials  $BIO_{\mathcal{U}_i}$ . Assume a finite set  $M$  is a metric space. Consider a distance function  $dis$  including an error tolerance limit  $\mathcal{T}$  based on error correction codes for a specific distance metric (set difference metric, hamming distance, edit distance etc.) such that:

- $dis : M \times M \rightarrow R^* = [0, \infty)$ .
- $dis(BIO_{\mathcal{U}_i}, BIO'_{\mathcal{U}_i}) = 0$  iff  $BIO_{\mathcal{U}_i} = BIO'_{\mathcal{U}_i}$ ,
- $dis(BIO_{\mathcal{U}_i}, BIO'_{\mathcal{U}_i}) = dis(BIO'_{\mathcal{U}_i}, BIO_{\mathcal{U}_i})$ ,
- $dis(BIO_{\mathcal{U}_i}, BIO''_{\mathcal{U}_i}) \leq (dis(BIO_{\mathcal{U}_i}, BIO'_{\mathcal{U}_i}) + dis(BIO'_{\mathcal{U}_i}, BIO''_{\mathcal{U}_i}))$ , where  $BIO_{\mathcal{U}_i}, BIO'_{\mathcal{U}_i}, BIO''_{\mathcal{U}_i} \in M$ .

The fuzzy extractor consists of a pair of randomized procedures, i.e., Generator ( $Gen$ ) and Reproduction ( $Rep$ ) with the following properties:

- The  $Gen()$  procedure uses a bio-metric credential  $BIO_{\mathcal{U}_i} \in M$  of user  $\mathcal{U}_i$  as an input and provides outputs—a secret string  $\sigma_i \in \{0, 1\}^l$  and a public accessory string  $\tau_i \in \{0, 1\}^*$ , i.e.,  $Gen(BIO_{\mathcal{U}_i}) = (\sigma_i, \tau_i)$

- The  $Rep()$  procedure uses a noisy bio-metric credential  $BIO'_{\mathcal{U}_i} \in M$  of user  $\mathcal{U}_i$  and the public accessory string  $\tau_i$  as an input and reproduces the secret string  $\sigma_i \in \{0, 1\}^l$  as an output i.e.,  $Rep(BIO'_{\mathcal{U}_i}, \tau_i) = \sigma_i$  if and only if  $dis(BIO_{\mathcal{U}_i}, BIO'_{\mathcal{U}_i}) \leq \mathcal{T}$ .

### 1.3.6 Bloom Filter

A Bloom Filter [8] is an  $m$  bit vector  $V$  which denotes a set  $S$  of  $n$  elements applying  $k$  independent hash functions. It supports set membership queries utilising an operative and probabilistic data structure including a false positive  $f = (1 - e^{-\frac{kn}{m}})^k$ .

### 1.3.7 Chinese Remainder Theorem

**Definition 1.3.6.** *Chinese Remainder Theorem:* If  $n_1, n_2, \dots, n_k$  are pair-wise relatively prime integer numbers, then the operation of simultaneous congruence:

$$x \equiv r_1 \pmod{n_1}$$

$$x \equiv r_2 \pmod{n_2}$$

$$\vdots$$

$$x \equiv r_k \pmod{n_k}$$

has a unique solution:  $\sum_{i=1}^k r_i N_i^{-1} N$  where;

$$N = \prod_{i=1}^k n_i$$

$$N_i = \frac{N}{n_i}$$

$$N_i^{-1} N = 1 \pmod{n_i}$$

### 1.3.8 AVISPA Tool

The AVISPA Tool [7] is a push-button software tool in support of the Automated Validation of Internet Security Protocols and Applications. We have used this tool for rigorous security analysis of existing as well as our proposed user authentication protocols of WSNs & IoT by either finding flaws or establishing their correctness. AVISPA

Tool gives a modular and descriptive formal language for describing security protocols and their security properties, and consolidates different back-ends that implement a variety of automatic protocol analysis procedures ranging from protocol falsification (by detecting an security attack on the input security protocol) to abstraction-based verification techniques for both finite and infinite numbers of sessions.

### **1.3.9 Syther Tool**

Scyther [19] is an automated security protocol verification tool which we have used for the analysis of our proposed user authentication protocols. Novel features of Scyther tool incorporate the likelihood of unrestricted verification with guaranteed termination, security testing of infinite sets of traces with regard to the patterns, and support for multi-protocol security analysis. Scyther is established on a pattern refinement technique, providing brief representations of (infinite) sets of traces. This enables the tool to support in the analysis of various major classes of security attacks and possible protocol behaviours, or to prove accuracy for an unrestricted number of protocol sessions.

### **1.3.10 BAN Logic**

Authentication protocols lay foundation for secure communication in the information security field. Hence, it is very crucial to ensure the correctness of authentication protocols. Many protocols proposed in the literature have been poorly designed, and they also contain flaws which can be exploited to weaken the systems. The prime objective of authentication is to ensure that a principal is really communicating with other principal with whom he/she intends to communicate. The Burrows-Abadi-Needham (BAN) logic [20] offers a mechanism to validate the correctness of an authentication protocol using the assumptions or beliefs made in the definition of an authentication protocol, there by employing reasoning or logic to deduce conclusions. In recent years, the mutual authentication proof between two communicating parties has been performed in several user authentication protocols using the widely-accepted BAN logic. The important notations which we have used for the logical security verification is represented in Table 1. The basic rules of BAN logic which we have used for the security verification of the protocol are as follows:

**Rule 1** Message meaning rule:  $\frac{P_{BAN} \models P_{BAN} \xleftarrow{K} Q_{BAN}, P_{BAN} \triangleleft \{S\}_K}{P_{BAN} \models Q_{BAN} \sim S}$ . That is, if  $P_{BAN}$  believes that she shared the key  $K$  with  $Q_{BAN}$ , and  $P_{BAN}$  sees the message  $\{S\}$  encrypted with key  $K$ ,  $P_{BAN}$  believes that  $Q_{BAN}$  once said  $S$ .

**Rule 2** Nonce verification rule:  $\frac{P_{BAN} \models \#(S), P_{BAN} \models Q_{BAN} \sim S}{P_{BAN} \models Q_{BAN} \equiv S}$ . That is, if  $P_{BAN}$  believes  $S$  is fresh and  $Q_{BAN}$  once said  $S$ ,  $P_{BAN}$  believes  $Q_{BAN}$  believes  $S$ .

**Rule 3** Jurisdiction rule:  $\frac{P_{BAN} \models Q_{BAN} \Rightarrow S, P_{BAN} \models Q_{BAN} \equiv S}{P_{BAN} \models S}$ . That is, if  $P_{BAN}$  believes that  $Q_{BAN}$  had jurisdiction right to  $S$  and believes  $Q_{BAN}$  believes  $S$ ,  $P_{BAN}$  believes  $S$ .

### 1.3.11 Random Oracle Model

A Random Oracle is based on a hypothetical black box that responds to each query with a (truly) random responses taken consistently from its output element, besides that for any specific query, it answers the similar way each time it takes that query i.e. a random oracle is a precise function mapping every possible query of the protocol to a random response from its output region.

Bellare and Rogaway first formalised the Random Oracle Model (ROM) [21] in 1993 . In the random oracle model, the individual believes that some hash function is replaced by a publicly available random function (the random oracle). This indicates that the adversary cannot calculate the result of the hash function by itself, he must query the random oracle.

## 1.4 Motivation of the work

Wireless Sensor Networks (WSNs) & Internet of Things (IoT) have attracted a large number of researchers due to its ubiquitous nature, easy deployment and a wide range of applications. In general, most of the queries in WSN & IoT applications are issued at the points of base stations or Gateway ( $GWN$ ) node of the network. However, one can foresee that there are greater needs to access the real-time data inside WSNs & IoT . The user must be able to access the real-time data from sensor nodes when required. For some applications, the collected data is valuable and confidential. In many applications, integrity and confidentiality of collected data as well as the user privacy are critical. Security measures should be incorporated to protect the access to

critical data and to restrict non- authorized users from acquisition of critical data. If the data is made available to the user on demand then user authentication must be ensured before allowing the data access.

Over the years, password based user authentication and two factor user authentications have been proposed by a number of researchers. Two-factor user authentication allows for a separation of roles at the expense of adding a multitude of implementation and deployment issues, which make them expensive. The two-factor user authentication assumes that WSNs are deployed in a confined area. The basic idea of the protocol is that during registration phase, a user receives a smart card from *GW* node. Then during login-authentication phase, the user can login to the sensor and access data with the aid of the users password and smart card. In this case, the user must insert the smart card into the specific terminal to be able to login. However in many applications ad hoc topology of WSN is deployed in uncontrolled areas where using a specific terminal and smart card may not be feasible and will restrict the user mobility and utility of WSN. Password-based authentication is easy to integrate and at least it does not involve some incompressible extra costs. Therefore, we are motivated to develop secure and efficient authentication protocols that eliminates various well known security attacks and allows only the legitimate users of the WSNs & IoT.

## 1.5 Objective of the thesis

Sensor nodes in a WSNs & IoT are battery-powered and have limited communication, computation and storage capabilities. This requires that the security design must be lightweight and efficient regarding both communication and computation overheads. Due to wireless communication between nodes in a WSN & IoT, an adversary can eavesdrop the communication messages and launch different types of attacks. The unattended nature of some WSN & IoT makes it vulnerable to node compromise attack. The resource and network constraints together with different attacks impose many challenging requirements for the security design in WSNs. A sophisticated security or authentication protocol requires a balance among the requirements and its design must be robust against sensor compromise and different attacks. All the password-based authentication protocols proposed so far have security weaknesses that make them unsuitable for wireless network. Moreover, all password-based protocols require strict



time synchronization, which increases network overhead and makes the protocol vulnerable to replay attack within a certain time interval. The past protocols also suffer from many logged in users with same ID threat. Therefore, our main objective is to develop a robust user authentication method that inherits all the advantages of previous password-based protocols and improves security by resisting gateway bypass attack, replay attack and many logged in user with the same ID threat. In this thesis, we propose a robust user authentication method which ensures that only the legitimate user can get the access to the sensor data. The objectives of this thesis are as follows:

1. Designing light- weight cryptography mechanisms of authenticated key exchange for the resource constrained sensor devices.
2. To propose protocols for eliminating unauthorized query information circulation at the initial level (i.e., at the sensor node itself) in order to prevent bogus information flooding from the sensor nodes to the gateway node.
3. Authenticating the users without pre-alignment between the test and the registered minutia points of the fingerprint.
4. Providing various known security features such as secure users password update, efficient session key exchange, data confidentiality, message integrity, availability, non-repudiation, data freshness and mutual authentication of the user, sensor node, gateway node.
5. Resisting various well-known security attacks such as sensor node and users identity impersonation attack, replay attack, denial of service and man-in-the middle attack, stolen smart card attack, energy and constrained resources exhausting attack.

## 1.6 Thesis Contributions

The contributions of the thesis are summarized in the following subsections. The significant contributions of our work are as follows:

- In this thesis, we first discuss various security issues involved in authenticating the users of WSNs and IoT.

- We perform the comprehensive literature survey and security analysis of various existing protocols of user authentication and key establishment for WSNs and IoT. Through security analysis, we show that the existing protocols are vulnerable to various attacks like user impersonation attack, sensor node impersonation attack, attacks based on legitimate users.
- We propose various secure and efficient protocols for authenticating the users of WSNs and IoT considering mutual authentication, session key establishment, data freshness, and confidentiality.
- Through informal security analysis, we confirm that our proposed protocols resist the stolen smart card, sensor node compromise, gateway node compromise, man-in-the-middle and replay attacks.
- We complete proof of security applying the random oracle model to assure the correctness of multiple security features associated in our proposed protocols.
- Afterwards, we perform security testing of the proposed protocols on popular and robust security verification tool such as AVISPA and Scyther.
- We apply BAN logic to ascertain whether exchanged messages of the proposed protocols are trustworthy and secure against eavesdropping.
- Finally, we perform the comparative analysis of our proposed protocols with other existing protocols based on security features and computational overhead.

## 1.7 Organization of the thesis

This thesis consists of 9 chapters including an introductory chapter and a concluding chapter. The content of each of these chapters is summarized below:

### Chapter 1: Introduction

In chapter 1, we provide the motivation for the work carried out in the thesis. The chapter begins with the introduction of user authentication for WSNs and IoT with challenges. The desired security requirements for authentication, storing, and sharing data. We discuss the problem statement, objectives, and our contributions for designing a secure authenticated key establishment mechanism.

### **Chapter 2: Literature Survey**

Chapter 2 reviews the related works and highlights the novel aspects of this thesis. In this chapter, the related work is categorized into two categories: (i) User Authentication for WSNs and IoT (ii) Key establishment for WSNs and IoT.

### **Chapter 3: ECC based User Authentication and Key Establishment protocol for WSNs & IoT**

In 2014, Choi et al. proposed an elliptic curve cryptography based user authentication protocol with enhanced security for wireless sensor networks and after security analysis of their protocol we find that their protocol has some security drawbacks such as (1) not resilient against node capture attack, (2) insecure against stolen smart card attack (3) vulnerable to sensor node energy exhausting attacks. Based on the security analysis we propose a protocol to withstand the various security weaknesses of WSNs. Furthermore, the comparative security and computational performance analysis indicate that our proposed protocol is relatively more secure and efficient.

### **Chapter 4: Fuzzy Extractor and ECC based User Authentication and Key Establishment protocol for WSNs and IoT**

In Chapter 4, we perform the security analysis of A.K.Das's user authentication protocol (given in 2015), Choi et al.'s protocol (given in 2016), and Park et al.'s protocol (given in 2016). The security analysis shows that their protocols are vulnerable to various attacks like user impersonation attack, sensor node impersonation attack and attacks based on legitimate users. Based on the cryptanalysis of these existing protocols, we propose a secure and efficient authenticated session key establishment protocol which ensures various security features and overcomes the drawbacks of existing protocols. The formal and informal security analysis indicates that the proposed protocol withstands the various security vulnerabilities involved in WSNs & IoT. The automated validation using AVISPA and Scyther tool ensures the absence of security attacks in our protocol. The logical verification using the Burrows-Abadi-Needham (BAN) logic confirms the correctness of the proposed protocol. Finally, the comparative analysis based on computational overhead and security features of other existing protocols indicate that the proposed user authentication system is secure and efficient.

### **Chapter 5: LU Decomposition based User Authentication and Key Establishment protocol for WSNs and IoT**

In Chapter 5, we propose a lightweight mechanism for authenticating users of a sensor network using fuzzy extractor along with a novel matrix based session key establishment protocol. After that, we perform the security analysis of our protocol using widely accepted automated verification tools such as AVISPA and Scyther. Then, we perform logical verification using BAN Logic. Finally, we do the computational analysis, and demonstrate by comparative efficiency analysis in terms of computational overhead and security features.

### **Chapter 6: Chinese Remainder Theorem based User Authentication and Key Establishment protocol for WSNs and IoT**

In Chapter 6, we propose an effective authenticated key exchange mechanism using the theories of the fuzzy extractor and Chinese Remainder Theorem. Subsequently, we present the security analysis of our proposed protocol using universally trusted automated security verification tools such as AVISPA and Scyther. Later, we do logical verification applying BAN Logic. Eventually, we do the computational study and illustrate the relative performance analysis in respect of computational cost and security traits.

### **Chapter 7: Bloom Filter based User Authentication and Key Establishment Protocol for WSNs & IoT**

In Chapter 7, we introduce Bloom filter based authentication protocol applicable for the WSNs and IoT of large hop count (i.e., the vast number of intermediate sensor nodes by which information must transfer between origin sensor node and terminal gateway node). The originality of our recommended protocol is, it discards illegal querying information communication at the beginning level (i.e., at the sensor node itself) to avoid false information flooding from the sensor nodes to the gateway node (which consumes the resources of WSNs & IoT). We conduct the formal and informal security interpretation of the proposed protocol utilising a universally trusted AVISPA tool plus random oracle model. The significant computational study confirms that our proposed protocol is fit for resource-constrained sensor nodes like TelosB and MicaZ. The comparative

security and performance analysis outcomes show that our protocol is further secure, cost-effective and strong in comparison to other existing protocols.

### **Chapter 8: Symmetric Hash Function based User Authentication and Key Establishment Protocol for WSNs & IoT**

In Chapter 8, we propose symmetric hash function and bloom filter based secure authenticated key exchange (AKE) protocol for WSNs & IoT. The proposed protocol is appropriate for authentication of the users without pre-alignment between the test and the registered minutia points of the fingerprint for multi-hop WSNs & IoT. The proposed protocol eliminates unauthorized query information circulation at the initial level (i.e., at the sensor node itself) to prevent bogus information flooding from the sensor nodes to the gateway node. We show that the proposed protocol resists the resource exhaustion attacks associated with WSNs & IoT of large hop count (i.e., a large number of intermediary sensor nodes through which information must pass between source sensor node and the trusted gateway node). We present both the formal and informal security analysis of the proposed protocol using AVISPA tool and basic cryptography concepts. The analysis of computational overhead demonstrates that our proposed protocol is preferable for resource-constrained sensor motes like MicaZ. The security analysis and performance evaluation reveal that the proposed protocol is more secure, effective and resilient in comparison to other existing protocols.

### **Chapter 9: Conclusions and Future Directions**

Chapter 9 summarizes the contributions of the thesis and outlines the future directions. The security analysis of existing user authentication protocols of the literature demonstrates that they are vulnerable to various attacks like user impersonation attack, sensor node impersonation attack and attacks based on legitimate users. The performance analysis illustrates that the existing protocols are inefficient considering the computational cost. Whereas, the comparative security and performance analysis indicate that our proposed protocols are secure against stolen smart card attack, user impersonation attack, sensor node impersonation attack, sensor node capture attack, replay attack, man-in-the-middle attack. The proposed authentication protocols provide various security features such as mutual authentication, three-factor authentication, secure

password and biometric information update, confidentiality, integrity, freshness. The proposed protocols are efficient concerning the computational cost of the resource-constrained sensor nodes, save communication energy and bandwidth. As a result, the protocol is appropriate for applications of resource-constrained ubiquitous sensor devices. Therefore, the proposed protocols can be used in various real-world applications consisting of resource constraint sensor devices of WSNs and IoT where bio-metric based secure user authentication and efficient session key establishment is required. The proposed protocols can be used for the implementation of bio-metric based secure authentic banking and financial transactions using the smart card, automated teller machines (ATM), point-of-sale (POS) machines. In this thesis, we have discussed the security issues involved with the sensor nodes of WSNs & IoT and performed the security analysis of various existing protocols of user authentication for WSNs. We have proposed various efficient user authentication, session key establishment protocols for WSNs and IoT using the Smart card, Fuzzy Extractor, ECDH techniques, Bloom filter, Chinese Remainder Theorem, LU Decomposition, Symmetric Hash Functions. We have presented security proof using random oracle model and BAN logic to ensure the correctness of various security features involved in the proposed protocols. Afterwards, we have performed the security analysis and verification using well-known and robust tools such as AVISPA and Scyther. Through the precise security analysis using mathematical functions and simulation tools, we have demonstrated that the proposed protocols fulfill the desirable security requirements and withstand the security drawbacks found in existing protocols of user authentication for WSNs. Finally, we have presented the comparative analysis of our protocols with other existing protocols based on security features and computational overhead which justify that our proposed protocols are secure, efficient and suitable for WSNs & IoT. In future, we would like to propose Hyper-Elliptic Curve Cryptography, Blockchain, Artificial Neural Network based authenticated key exchange protocols suitable for WSNs, IoT and IoT based Cloud Services.

# Chapter 2

## Literature Review

“Research is formalized curiosity. It is poking and prying with a purpose.” Zora Neale Hurston

In this chapter, we first provide a classification of various user authentication and key establishment protocols proposed for WSNs & IoT. We then present an overview of the related works on user authentication and key establishment protocols in WSNs & IoT.

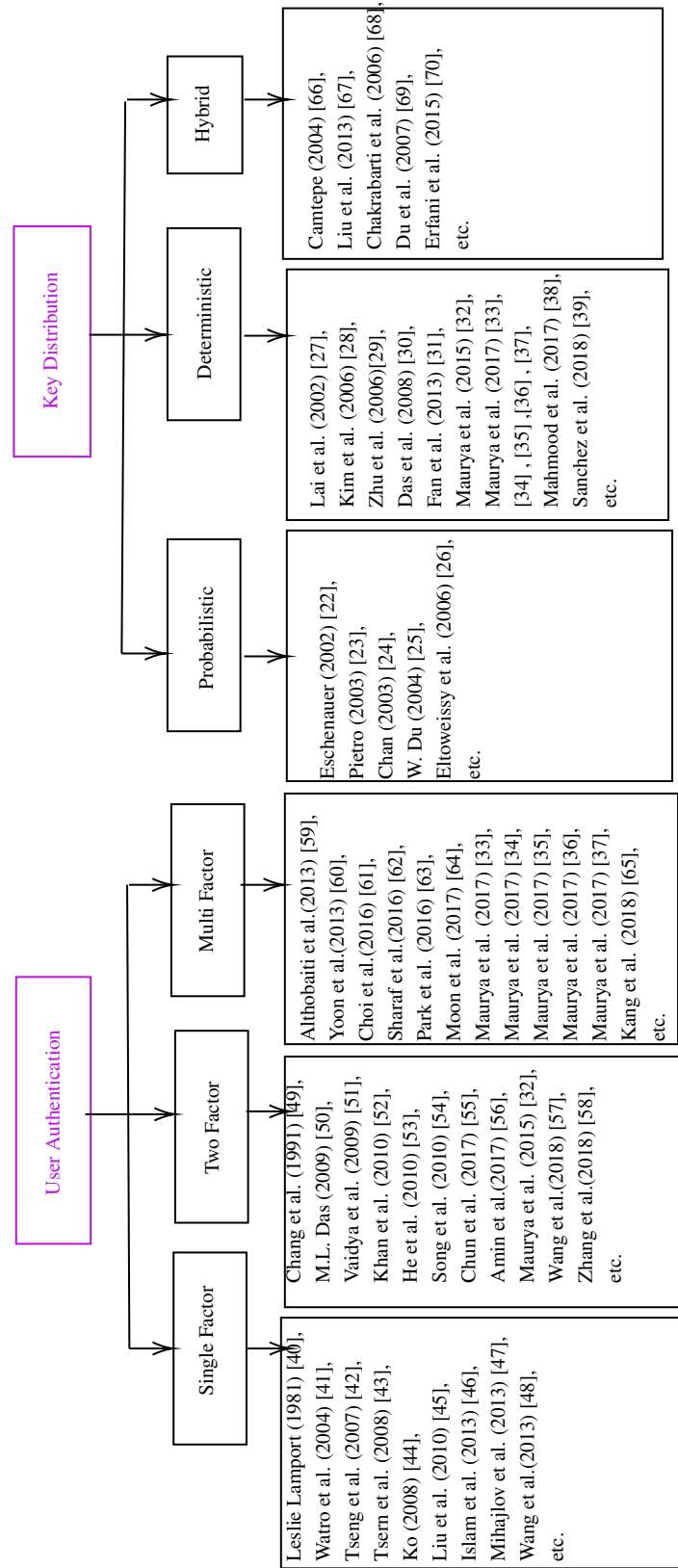
### 2.1 Classification of security protocols in WSNs & IoT

The user authentication and session key establishment are the main security issues in WSNs & IoT. Figure 2.1 shows a classification of security protocols related to user authentication and key establishment for WSNs & IoT.

### 2.2 User authentications protocols for WSNs and IoT

The existing user authentications protocols proposed in the literature for WSNs & IoT usually fall into three categories: (i) Single factor user authentication protocol, (ii) Two factor user authentication protocol and (iii) Multi-factor user authentication protocol. Some basic features and major security attacks on the existing protocols are illustrated in figure 2.2 and 2.3 respectively. In 1981, Lamport [40] first proposed a

**Figure 2.1:** User Authentication and Key Establishment Protocol for WSN & IoT





## 2.2 User authentications protocols for WSNs and IoT

---

remote password authentication protocol for insecure channels, and since then, many authentication protocols have been studied in order to enhance security and efficiency.

The user authentication protocols proposed by Watro et al. [41], Wong et al. [71], Tseng et al. [42], Tsern et al. (2008) [43], Ko [44] and Liu et al. (2010) [45] etc. come under single-factor password based authentication.

The user authentication methods proposed by M.L. Das (2009) [50], Vaidya et al. (2009) [51], Khan et al. (2010) [52], He et al. (2010) [53], Song et al. (2010) [54], Amin et al.(2017) [56], Wang et al.(2018) [57], Zhang et al.(2018) [58] etc. fall under two-factor password based authentication.

Finally, the protocols proposed by Althobaiti et al.(2013) [59], Yoon et al.(2013) [60], Choi et al.(2016) [61], Sharaf et al.(2016) [62], Park et al. (2016) [63], Moon et al. (2017) [64], Kang et al. (2018) [65], Sutrala et al. (2018) [72] etc. fall under bio-metric (multi factor) based user authentication.

### 2.2.1 Literature review and problems identified in user authentication for WSNs and IoT

In 2002, Akyildiz et al. [2] explored many significant aspects of WSNs and discussed critical open research issues of WSNs. Afterwards, several user authentications and session key agreement mechanism for WSNs have been proposed. Unfortunately, many of them still suffer from various security vulnerability. In 2004, Benenson et al. [11] proposed a user authentication and access control mechanism for WSNs. Consequently, Watro et al. [41] (in 2004) designed a user authentication method for WSNs based on public-key cryptographic protocols, called TinyPK protocol. The RSA algorithm [73] and Diffie-Hellman protocol [74] are applied in TinyPK which provides mutual authentication and withstand sensor node impersonation attack. As outlined in [50], [75], the TinyPK is prone to the following attack. After receiving the user's public key, an adversary, say A can encrypt a session key along with other parameters and transmit the encrypted message to that user. When the encrypted message is received, the user believes that the message has come from the legitimate sensor node. This forces the user to decrypt the receiving encrypted message using his/her private key. In addition, the user applies the session key for subsequent operations that the attacker intends to perform.

## 2.2 User authentications protocols for WSNs and IoT

---

Subsequently (in 2005), Benenson et al. [76] designed an elliptic curve cryptography based user authentication protocol. In 2006, Wong et al. [71] declared that Benenson et al.'s [76] method is resistless to denial of service and impersonation attacks. Wong et al. [71] then designed a user authentication method based on user's password using hash function which is a lightweight operation. However, it was shown in [50], [75] that their method does not resist many logged in users with the same login-id attack in which if an attacker has an authorized user's password, he/she can easily login to the network. In addition, their method is also vulnerable to stolen-verifier attack as both the gateway node and login sensor node need to maintain the lookup table of the registered user's secret information.

However, in 2007, Tseng et al. [42] specified that Watro et al.'s [41] and Wong et al.'s [71] methods exhibit replay and forgery attack. Further, Tseng et al. improved Wong et al.'s method and recommended password update mechanism. In 2008, Lee [43] revealed that Wong et al. [71] method exhibit more computational overhead on sensor node compared to gateway node and proposed an improved authentication method by fixing the security drawbacks of Wong et al. method with less computation overhead of sensor node. Later, L.C. Ko [44] indicated that Tseng et al.'s method does not provide mutual authentication. Then, L.C. Ko [44] proposed mutual authenticity and time-stamp based user authentication method in 2008.

In 2009, Vaidya et al. [51] elaborated mutual authentication method with formal verification. In 2009, Das [50] developed a secure mechanism to provide authenticity using smart card and user's password (two-factor) but it does not offer session key between the user and sensor node. Afterwards, Ren et al (2009) [77] proposed a protocol of multi-user authentication using bloom filter to store multiple user IDs and Public keys. The disadvantage is that the Bloom filter can be forged and cannot prevent the DOS attack. In 2010, Khan and Alghathbar [52] detected that Das [50] protocol is vulnerable to the gateway node bypassing attack and privileged-insider attack, it does not provide methods to change users passwords, and it does not achieve mutual authentication between the gateway node and the sensor node. To rectify these deficiency, they proposed security patches and enhancements.

The introduced two-factor authentication mechanism based on the user's identity and password is usually not secure because the user aims to choose a low-entropy password that can be easily decoded by implementing simple dictionary attacks. To

## 2.2 User authentications protocols for WSNs and IoT

---

enhance the security point of two-factor user authentication mechanism that is vulnerable to password guessing attacks and subject to incompetent password update method in WSNs & IoT, biometric-based user authentication mechanism, followed with user passwords and smart cards, have drawn considerable attention.

In 2010, Yuan et al. [75] provided a bio-metric based method but it is unprotected from node capture and denial of service attack. In 2012, Yoo et al. [78] designed a method that provides secure session key and mutual authentication. In 2013, Xue et al. [79] designed a mutual authentication method based on temporal information. However, in 2015, Jiang et al. [80] revealed that Xue et al.'s method is susceptible to stolen smart card and privilege insider attack. In 2015, A.K. Das [81] proposed fuzzy extractor based authentication method which resists well known security attacks of WSNs and have more security features compared to Althobaiti et al. (2013) [59] method. Sharaf et al. [62] proposed (in 2016) an object authentication protocol in order to exploit device-specific data, known as fingerprints, to authenticate the objects associated with the IoT. In 2016, Alizadeh et al. [82] presented a comprehensive survey of authentication methods of mobile cloud computing (MCC) to explain MCC authentication and differentiate it with that of cloud computing methods.

In 2016, Chang and Le [83] proposed a user authentication method in WSNs using password and smart card. They designed two protocols, namely P1 and P2. While P1 is based on bitwise XOR operations and one-way cryptographic hash functions, P2 uses ECC technique in addition to the two functions used in P1. Though their protocols were efficient, Das et al. [84] pointed out that both P1 and P2 are prone to session specific temporary information and offline password guessing attack, while P1 is also insecure against session key breach attack. In addition, they noticed that both the protocols P1 and P2 are inefficient in authentication as well as password change phases. To withstand these security weaknesses and limitations, they proposed a new authentication key agreement method using ECC.

Afterwards, in 2016, Amin and Biswas [85] proposed a two factor user authentication method using the multigateway based hierarchical WSNs. However, Wu et al. [86] showed that their method is insecure with respect to sensor capture attack, user forgery attack, gateway forgery attack, sensor forgery attack and offline guessing attack. In addition, Wu et al. [86] demonstrated that the user in Amin-Biswas's method can be

## 2.2 User authentications protocols for WSNs and IoT

---

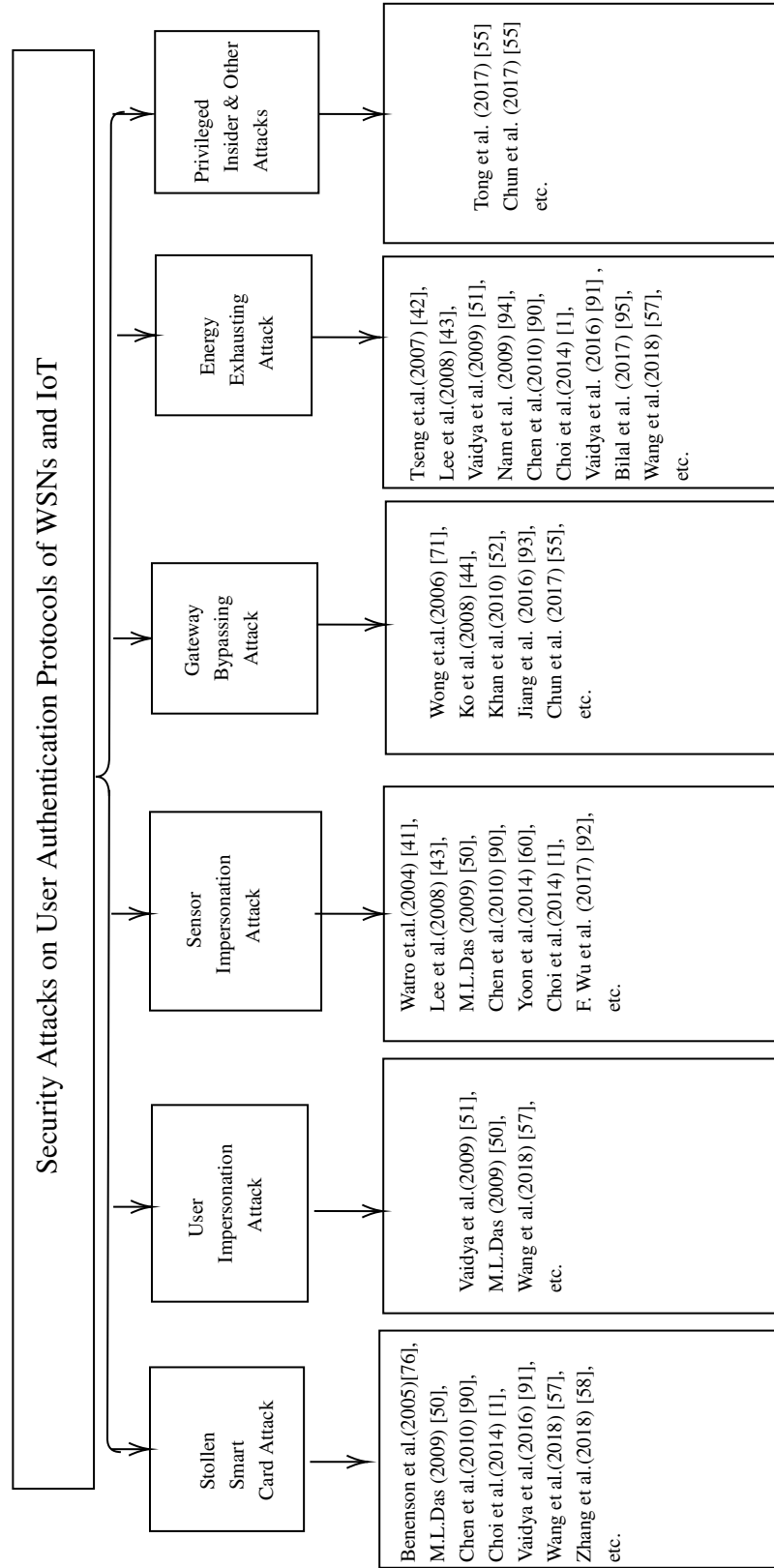
tracked due to the use of a constant pseudo-identity and also previously established session keys can be calculated by an attacker. In addition, Amin-Biswas's method does not provide efficient online sensor node registration and password change phases, and also their method contains design flaws in authentication and key agreement phase. Wu et al. then designed an efficient authenticated key agreement method for multi-gateway WSNs.

In 2017, Srinivas et al. [87] analyzed the security of Amin-Biswas's method [85] and pointed out that their method is insecure against several attacks, including leakage of sensors secret keys, gateway spoofing attack, stolen smart card attack, password guessing attack, impersonation attack and also identity guessing attack. To withstand these security drawbacks, they presented another efficient and more secure user authentication method. Their method is a three-factor method based on smart card, user password and personal bio-metrics. However, Srinivas et al.'s method still has the following drawbacks:

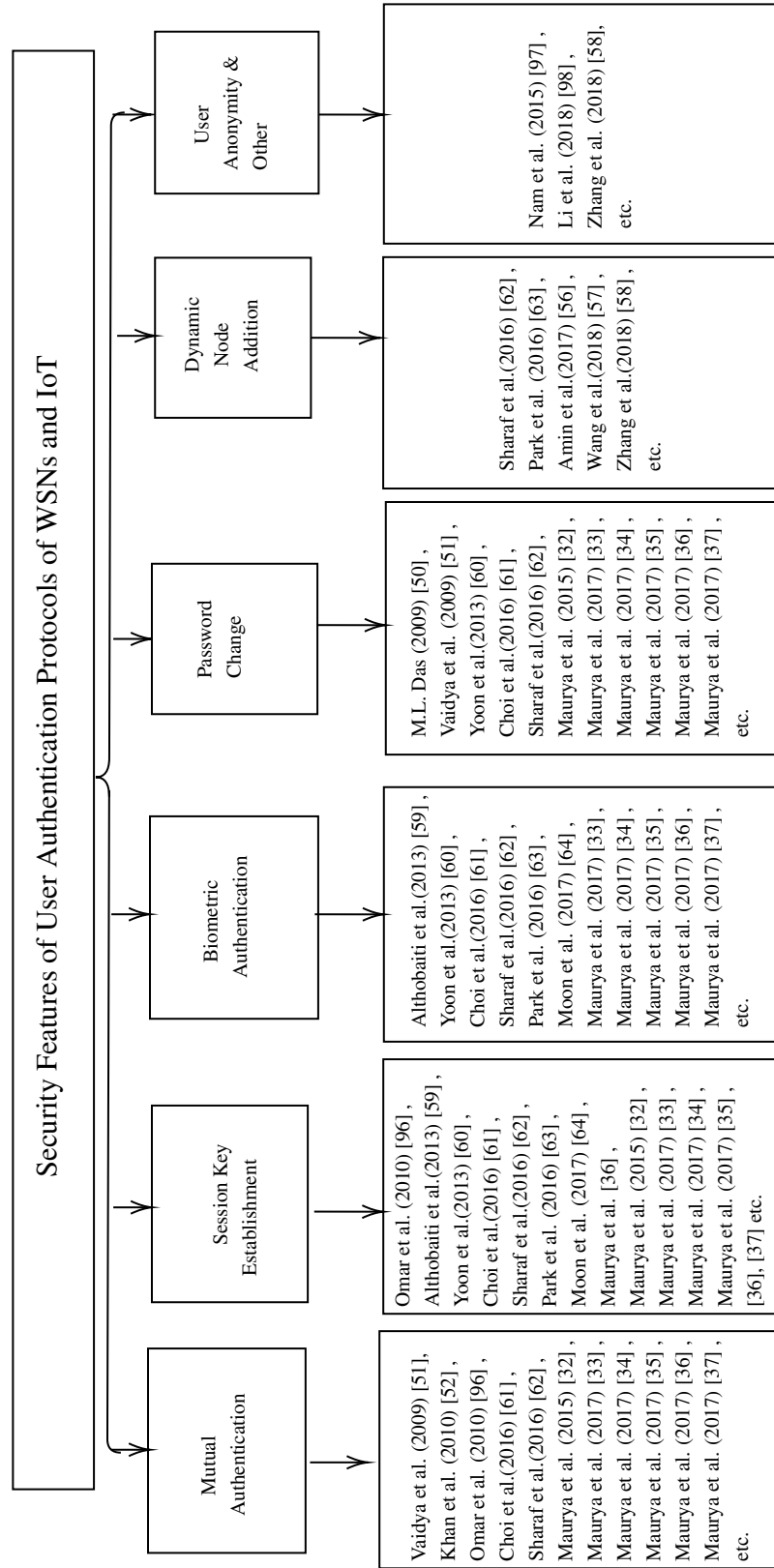
- The sensor node registration takes place in online mode as in Amin-Biswas's method. Hence, Srinivas et al.'s [87] method does not provide efficient online sensor node registration.
- The bio-hashing function is not good for bio-metric verification in authentication methods as it was shown for Mishra et al.'s method [88] by Chang et al. [89]. Since Srinivas et al.'s method applies the bio-hashing function for bio-metric verification, it leads to denial-of-service attack.

However, in this thesis we performed the crypt-analysis of A.K.Das [81] method and found that it is susceptible to stolen smart card attack. Similarly, we found that Choi et al. [61] (proposed in 2016), Park et al. [63] (introduced in 2016), and Moon et al.'s [64] (proposed in 2017) methods are also insecure against various security attacks as we have illustrated in chapter-4 of this thesis. The security attacks and features found in existing user authentication protocols for WSNs & IoT are described in following figure 2.2 and 2.3 respectively.

**Figure 2.2:** Security Attacks on User Authentication Protocols of WSNs and IoT



**Figure 2.3:** Basic Security Features of User Authentication Protocols of WSNs and IoT



## **2.3 Key Establishment Protocols for Wireless Sensor Networks and Internet of Things**

Key establishment practices in the WSNs & IoT are divided into three categories (as shown in Figure 2.1) : (a) Probabilistic (b) Deterministic and (c) Hybrid. In an instance of probabilistic distribution, key ring for each node is chosen randomly from a large pool of keys. Deterministic distribution allocates keys to nodes in a deterministic way to guarantee reliable connectivity. Hybrid protocols apply probabilistic techniques on deterministic resolutions to improve scalability and resilience.

### **2.3.1 Probabilistic Key Distribution Methods**

In 2002, Eschenauer and Ghorog were the first to give a key management protocol based on the random possibility (the so-called basic random key distribution protocol, or E-G protocol [22]) for WSNs, which is the basis of the other key management protocols. This process consists of the following three phases:

- **Key pre-distribution phase:** In this phase the (key) setup server generates a key pool  $K$  of  $M_K$  randomly chosen symmetric keys. Each key is then associated with a unique identifier in  $K$ . For each sensor node  $SN_i$  to be deployed in a target field, the setup server selects a random subset  $K_{SN_i}$  of size  $m_K$  from the pool  $K$ , and loads  $K_{SN_i}$  into its memory.  $K_{SN_i}$  denotes the key ring of the sensor node  $SN_i$ .
- **Direct key establishment phase:** This phase is executed by each sensor node once they are deployed in a target field. To establish a secret pairwise key between two neighbor sensor nodes, say  $SN_i$  and  $SN_j$ , they can exchange the key ids from their key rings  $K_{SN_i}$  and  $K_{SN_j}$ , respectively. If there is a common key id between their key rings, the corresponding key is taken as the secret key between them and they apply this key for their secure communication in future. Therefore, the EG method demands that at least one common key must be common between the key rings of two neighbor sensor nodes.
- **Path key establishment phase:** This phase is an optional phase, and if it is applied, adds to the connectivity of the network. If two neighbor nodes  $SN_i$  and  $SN_j$  are

## 2.3 Key Establishment Protocols for Wireless Sensor Networks and Internet of Things

---

unable to establish a secret pairwise key between them in the direct key establishment phase due to lack of any common keys in their key rings and there exists a secure path between them,  $SN_i$  can generate a new random pairwise key  $k$  shared with  $SN_j$  and securely transmits it along the secure discovered path to the desired destination node  $SN_j$ . Thus,  $SN_i$  and  $SN_j$  can now communicate secretly and directly using the key  $k$ . However, the communication overhead increases significantly with the number of hops. To tackle this issue, in practice, the number of hops is restricted to a small value, say 2 or 3.

The network connectivity, that is, the probability of establishing a direct pairwise key between two sensor nodes in the EG method is given by:

$$\begin{aligned} Pr_{EG} &= 1 - \frac{\binom{M_K - m_K}{m_K}}{\binom{M_K}{m_K}} \\ &= 1 - \prod_{i=0}^{m_K-1} \frac{M_K - m_K - i}{M_K - i} \end{aligned} \quad (2.1)$$

where  $M_K$  is the key pool size and  $m_K$  the key ring size of a sensor node. If  $c$  sensor nodes are physically captured by an attacker, the resilience against sensor node capture attack is estimated by

$$Pr_e(c) = 1 - \left(1 - \frac{m_K}{M_K}\right)^c \quad (2.2)$$

Blundo et al. [99] designed the polynomial-based key pre-distribution method. In the key pre-distribution phase, the key setup server selects unique identities to all deployed nodes in a target field. A  $t$ -degree symmetric bivariate polynomial  $f(x, y) = \sum_{i,j=0}^t a_{ij}x^i y^j$  is then generated with the coefficients  $a_{ij}$  ( $0 \leq i, j \leq t$ ) randomly chosen from a finite field  $GF(q)$ ,  $q$  is a prime that is large enough to accommodate a symmetric cryptographic key. Since  $f(x, y)$  is bivariate, it follows that  $f(x, y) = f(y, x)$ . The setup server computes a polynomial share  $f(ID_{SN_i}, y)$  for each deployed sensing node  $SN_i$  and loads the coefficients of  $y^j$  of  $f(ID_{SN_i}, y)$  and the identity  $ID_{SN_i}$  in the memory of  $SN_i$ . For establishing a pairwise key between



## 2.3 Key Establishment Protocols for Wireless Sensor Networks and Internet of Things

---

two neighbor nodes  $SN_i$  and  $SN_j$ , they need to exchange their ids only.  $SN_i$  computes the secret key shared with its neighbor  $SN_j$  as  $sk_{SN_i,SN_j} = f(ID_{SN_i}, ID_{SN_j})$  and  $SN_j$  also derives the same pairwise secret key shared with its neighbor  $SN_i$  as  $sk_{SN_j,SN_i} = f(ID_{SN_j}, ID_{SN_i})$ . Since  $f(ID_{SN_i}, ID_{SN_j}) = f(ID_{SN_j}, ID_{SN_i})$ , we have  $sk_{SN_i,SN_j} = sk_{SN_j,SN_i}$ . Therefore, both  $SN_i$  and  $SN_j$  share the common key  $sk_{SN_i,SN_j}$  for their secure communication in future. The advantage of this method is it provides 100% network connectivity with minimal communication overhead. However, if more than  $t$  nodes are compromised by an adversary, he/she can easily reconstruct the original polynomial back using Lagrange interpolation [100] which results in deriving all the pairwise keys between any two non-compromised nodes in  $WSN$ . As a result, this method is known as unconditionally secure and  $t$ -collusion resistant.

Using the Blundo et al.'s polynomial-based method [99], Liu and Ning [101] designed the polynomial-pool based method. Let  $f(x, y)$  be a  $t$ -degree symmetric bi-variate polynomial i.e.,  $f(x, y) = f(y, x)$ , whose coefficients are from a finite field  $GF(q)$ . The key setup server generates a pool  $K$  of  $s$  symmetric bi-variate polynomials randomly in  $GF_q[x, y]$  each of degree  $t$  in  $x$  and  $y$ . Some  $ids(ID_{SN_1}, ID_{SN_2}, \dots, ID_{SN_n} \in GF_q)$  are also generated for the sensor nodes to be deployed in a target field, where  $n$  is the network size. For each deployed sensor node  $SN_i$ ,  $s'$  polynomials, say,  $f_1(x, y), f_2(x, y), \dots, f_{s'}(x, y)$  are randomly selected from the polynomial pool  $K$  and the polynomial shares  $f_1(ID_{SN_i}, y), f_2(ID_{SN_i}, y), \dots, f_{s'}(ID_{SN_i}, y)$  are loaded in the key ring  $K_{SN_i}$  of node  $SN_i$ . After nodes deployment, each sensor node  $SN_i$  exchanges the ids of the polynomial shares residing in its key ring  $K_{SN_i}$ . Two physical neighbors  $SN_i$  and  $SN_j$  having shares of some common polynomial(s) will be able to establish a pairwise key between them. For a given polynomial-pool size  $s$  and the number  $s'$  of polynomial shares loaded in each sensor node's memory, the probability of establishing a direct pairwise key between any two neighbor sensor nodes (network connectivity) is given by [102]:

## 2.3 Key Establishment Protocols for Wireless Sensor Networks and Internet of Things

---

$$\begin{aligned}
 Pr_{poly-pool} &= 1 - \frac{\binom{s-s'}{s'}}{\binom{s}{s'}} \\
 &= 1 - \prod_{i=0}^{s'-1} \frac{s-s'-i}{s-i}
 \end{aligned} \tag{2.3}$$

If  $c$  sensor nodes are captured, the probability of a particular bi-variate polynomial  $f$  being compromised is given by [102]:

$$Pr_c = 1 - \sum_{i=0}^t Pr(i) \tag{2.4}$$

Where

$$Pr(i) = 1 - \frac{c!}{(c-i)!i!} \left(\frac{s'}{s}\right)^i \left(1 - \frac{s'}{s}\right)^{c-i} \tag{2.5}$$

Here,  $s$  and  $s'$  are the polynomial pool size and the number of polynomial shares given to each sensor node, and  $t$  is the degree of the bivariate polynomial. Since  $f$  is any polynomial in the polynomial pool  $K$ , the fraction of compromised secure links between non-compromised nodes can be estimated as  $Pr_c$  when  $c$  sensor nodes are already captured by an attacker in  $WSN$ .

Chan et al. [24] proposed several modifications of the EG method and they proposed several methods, such as random pairwise keys method,  $q$ -composite method and multipath key reinforcement method. In the  $q$ -composite method, two neighbor sensor nodes need at least  $q$  common keys ( $q > 1$ ) instead of one for establishing a secret pairwise key between them as compared to the EG method. If  $SN_i$  and  $SN_j$  be two neighbor sensor nodes sharing  $q'$  common keys, say,  $k_1, k_2, \dots, k_{q'} (q' \geq q)$  in their key rings, they can come up with computing the same common secret key as  $sk_{SN_i, SN_j} = h(k_1 || k_2 || \dots || k_{q'})$ , where  $h(.)$  is a one-way cryptographic hash function (for example, SHA-1 [103]). The probability of establishing a direct pairwise key between two neighbor sensor nodes is 1- (probability that the two sensor nodes share less than  $q$  common keys), which is given by [24]:

## 2.3 Key Establishment Protocols for Wireless Sensor Networks and Internet of Things

---

$$Pr_q(c) = 1 - \sum_{i=0}^{(q-1)} Pr(i) \quad (2.6)$$

Where the probability that the sensor nodes will share exactly  $i$  keys from their key rings is given by

$$Pr_e(c) = 1 - \frac{\binom{M_K}{i} \binom{M_K-i}{2(m_K-i)} \binom{2(m_K-i)}{m_K-i}}{\binom{M_K}{m_K}^2} \quad (2.7)$$

Where  $M_K$  and  $m_K$  are the key pool size and the key ring size respectively. The probability that any secure link setup in the direct key establishment phase between two non-compromised nodes is compromised when  $c$  sensor nodes are already captured is given by [24]:

$$Pr_i = \frac{1}{Pr_q(c)} \sum_{i=q}^m Pr(i) \times \left[ 1 - \left( 1 - \frac{m_K}{M_K} \right)^c \right]^i \quad (2.8)$$

In 2017, Gandino et al. [104] proposed a new key distribution method in WSNs, called q-s-composite, which is based on random pre-distribution of the secret material. The q-s-composite method improves the network performance as compared to the methods [99], [24], [22], [101]. The sensor nodes in the random methods [99], [24], [22], [101] do not regenerate the keys in their key rings after the first deployment in a target terrain, and the keys persist static during the entire lifetime of the sensor nodes. A dynamic key pre-distribution protocol handles this problem in which the sensor nodes in each deployment phase refresh their keys in key rings. Das [105] gave the multi-phase deployment key establishment protocol (MPDKE) based on hash chain. Toward this protocol, the sensor nodes are systematically deployed in the target area in many deployment stages. It was shown that this protocol obtains significantly better resilience against sensor node capture attack as compared to that for other existing random key distribution protocols without damaging the network connectivity.

The following Table 2.1 demonstrates the correlation of the storage, communication and computation costs, and also network connectivity amongst the random key

## 2.3 Key Establishment Protocols for Wireless Sensor Networks and Internet of Things

**Table 2.1:** Comparison of communication & computation costs among EG, q-composite, polynomial-pool and MPDKE methods

Protocols	Communication Cost	Computation Cost	Network Connectivity
EG [22]	$m_K \log M_K$ bits	$\frac{2m_K + Pr_{EG} - Pr_{EG} \cdot m_K}{2} \log m_K$ comparisons	Poor when key pool size is large
Polynomial-pool [101]	$s' \log s$ bits	$\frac{2s' + Pr_{poly} - Pr_{poly} \cdot s'}{2} \log s'$ comparisons plus one t-degree polynomial evaluation	Poor when polynomial pool size is large Poor when polynomial pool size is large
q-composite [24]	$m_K \log M_K$ bits $m_K \log M_K$ bits	$m_K \log m$ comparisons plus one hash operation	Poor when key pool size is large Poor when key pool size is large
MPDKE [105]	$m_K \log M_K$ bits	$m \log m$ comparisons plus one $PRF$ operation	Much better even if second key pool size is large

Note:  $M_K$  = key pool size;  $m_K$  = key ring size;  $s'$  = number of polynomial shares;  $s$  = polynomial-pool size;  $Pr_{EG}$ ,  $Pr_{poly}$  are secure (direct) network connectivity in EG method and polynomial-pool method, respectively;  $PRF$  is pseudo-random function

pre-distribution protocol, such as  $EG$ ,  $q$ -composite, polynomial-pool and MPDKE. It is worth mentioning that MPDKE gives better network connectivity even if the key pool size is large as compared to other protocol. But, it needs more computation cost for updating the keys in all sensing node's memory in all deployment phase applying the hash chain.

### 2.3.2 Deterministic Key Distribution Methods

This approach has the following characteristics:

- Every sensor node is pre-loaded by just a unique network-wide key prior to deployment. Therefore, a single key is required. As a consequence, the storage cost for each node is very low.
- This approach does not need completion of the direct key establishment phase.

But, the major shortcoming of this protocol is its very weak resilience against sensor node capture attack. It is because the physical capture of even an individual sensor node in WSNs & IoT would reveal the secret network-wide key, and therefore, it enables decryption of all network traffic. To improve the resilience against node capture attack, a different method is to apply a single shared network-wide key to establish pairwise keys among any two neighbour sensor nodes and then remove the network-wide key. But, the significant disadvantage of such an alternative is that it does not

## 2.3 Key Establishment Protocols for Wireless Sensor Networks and Internet of Things

---

allow new sensor node addition after the first deployment which is a meaningful evaluation metric of a key pre-distribution method applied in WSNs & IoT. Remarkable alternatives to such a strategy based on a single network-wide key cover the following methods:

**Localized Encryption and Authentication Protocol (LEAP):** Zhu et al. designed LEAP [29] that utilizes four types of keys for each sensor node. The keys are applied for various purposes and they range from the individual key that is shared with the base station, up to a group key that is shared with all the sensor nodes in the network. In LEAP, all sensor nodes are pre-loaded with a single network-wide key in their memory prior to their deployment. After deployment, each sensor node establishes keys using the pre-loaded initial network-wide key with its all neighbor nodes during a short initial phase after deployment, assuming that no nodes are compromised during this phase. Once it is done, all the sensor node need to subsequently erase the network-wide key. This method, however, is vulnerable to the behavior of a single node that misses the key setup period, and does not delete its initial network-wide key. The advantage of this method is that it supports new sensor node addition phase after initial deployment for establishing secret keys between new nodes and their old neighbor nodes.

**Enhanced LEAP:** Kim et al. [28] pointed out that in LEAP, the entire WSN may suffer a severe loss if the initial network-wide key is exposed to an attacker during the key setup phase. To strengthen the security of LEAP, early key establishment needs to be completed quickly by each sensor node. They presented a modified version of LEAP in order to improve its security as well as efficiency. In their method, during the direct key establishment phase, each sensor node needs a single broadcast to establish secret pairwise keys with its all neighbor nodes. After establishing pairwise keys with its all neighbor nodes, each node subsequently deletes the network-wide key. Single broadcast reduces significantly the amount of communication and computation overheads as compared to those for LEAP. This makes less probability of exposing the initial master key during key setup in Kim et al.'s method. Moreover, their method also supports addition of new nodes after initial deployment.

## 2.3 Key Establishment Protocols for Wireless Sensor Networks and Internet of Things

---

**BROadcast Session Key (BROSK) negotiation protocol:** In BROSK negotiation protocol [27], each node can negotiate a session key with its neighbour nodes by broadcasting the key negotiation message. The key setup server stores a network-wide master key in memory of each node prior to deployment in a target terrain. The network-wide master key is shared by all the sensor nodes as well as the gateway node. Following deployment, each node broadcasts a request message that contains its identifier and a random nonce generated by that node to all its neighbours. After receiving the message from the neighbour nodes, they calculate the secret keys among them accepting their random nonces. It is believed that in BROSK, the master key will not be removed from a captured sensor node's memory. This needs that the sensor nodes must be provided with tamper-resistant hardware. But, in practical applications, this assumption is ineffective due to cost constraints of the sensor nodes. Thus, compromise of a single sensor node proceeds to compromise the entire WSN, because the network-wide master key will be revealed to the adversary from the captured node's memory. But, this design gives 100% network connectivity, and it is further very scalable, that is, it supports a tremendous number of sensor nodes in WSN.

In 2017, Mahmood et al. [38] proposed a deterministic (polynomial based) key establishment protocol which is efficient regarding storage, computation, and communication overhead and can protect WSN-based IoT infrastructure.

### 2.3.3 Hybrid Key Distribution Methods

There are many strategies in the hybrid class which merge the deterministic and probabilistic protocols to produce a consistent level of metrics. In 2003, Liu et al. [67] introduced a hybrid method which applies a threshold method and gives a trade-off between the security against node capture and the act of setting pairwise keys. In 2006, Chakrabarti et al. [68] recommended a protocol which is tunable to user specifications, and it also correlates conveniently with state of the art design approaches. A significant speciality of this design is the appearance of the more substantial number of shared keys among any two sensor nodes. In 2007, Du et al. [69] introduced a hybrid protocol which gets the advantage of the robust high-end sensors in heterogeneous sensor networks and affords greater security with moderate complexity based on the notable

reduction on storage requirement, related to existing key management protocols. Then, Erfani et al. (in 2015) [70] introduced a dynamic key management protocol which is adequate concerning confidentiality, flexibility, memory usage, energy depletion and cost.

Chan et al. [24] recommended an upgraded random key distribution protocol, called q-composite protocol, which enhances the difficulty exponentially for an adversary in damaging the security link, although it degrades the network connectivity. Choi et al. [106] explained a new robust key predistribution protocol by applying keys allocation based on the concept of eigenvalues and eigenvectors of a square matrix of a keys pool. Zhou et al. [107] introduced a key predistribution protocol consolidating the LU matrix with Chinese Remainder Theorem (CRT). Liu et al. proposed a revised method in [101] based on the Blundo et al. protocol [99], specifically a key predistribution protocol based on the polynomial pool, and two flexible instantiation protocols were introduced. In their protocol, the server randomly produces  $S$  bivariate  $t$ -degree polynomials.

## 2.4 Summary

In this chapter, we have discussed various security protocols proposed for WSNs & IoT based on the taxonomy presented in Figure 2.1. We have first surveyed various user authentication protocols proposed in WSNs & IoT environment based on 1, 2 and 3 factors. Then, We have surveyed various key establishment mechanisms proposed in WSNs & IoT environment. Under this we have analyzed deterministic, probabilistic as well as hybrid key distributions methods in WSNs & IoT. Finally, we have done literature survey on user authentication and key agreement problem for both two-factor as well as three-factor methods. In this regard, we have analyzed various user authentication methods for distributed and hierarchical WSNs & IoT. Based on the survey, the identified problems have been considered and solved in the subsequent chapters.

## **Chapter 3**

# **Two Factor User Authentication and Key Establishment protocol for WSNs & IoT**

“Count what is countable, measure what is measurable, and what is not measurable, make measurable” Galileo Galilei

This chapter presents an efficient two-factor user authentication and key establishment protocol for WSNs & IoT based on smart card and elliptic curve cryptography. It briefly describes the problem, the motivation behind the work, challenges and contributions made thereof. Later, it elaborates the proposed protocol, and then, the security and computational overhead are analyzed.

### **3.1 Introduction and Problem Definition**

One of the top priorities in the world of information security is to protect the sensitive and confidential information from being accessed in an unauthorized manner. Such access is verified by letting users to prove, who they claim to be, using some authentication mechanism. User authentication and secret session key exchange between a user and a sensor node are important security requirements of wireless sensor networks for retrieving the important, confidential and real time information from the sensor nodes.



A two-factor user authentication for WSNs & IoT is a concept used to describe an authentication procedure, where more than one factor (e.g., password and smart card) is required to authenticate the users. In this chapter our aim is to propose a two-factor user authentication protocol for WSNs & IoT, which provides strong authentication, session key establishment, and achieves computational efficiency.

In 2014, Choi et al. [1] proposed an elliptic curve cryptography based user authentication protocol with enhanced security for wireless sensor networks and after security analysis of their protocol we find that their protocol has some security drawbacks such as (1) no resilient against node capture attack, (2) insecure against stolen smart card attack (3) vulnerable to sensor node energy exhausting attacks. Based on the security analysis we propose a protocol to withstand the various security weaknesses of WSNs & IoT. Moreover, formal security analysis and simulations are also conducted using AVISPA(Automated Validation of Internet Security Protocols and Applications) to show that our protocol is secure against active and passive attacks. The comparative security and computational performance analysis indicate that our proposed protocol is relatively more secure and efficient.

**Problem Definition:** The problem definition of this chapter is as follows:

Design and analysis of secure and efficient two-factor user authentication and session key establishment protocol for resource constraints WSNs& IoT which provides the major security features (such as mutual authentication, secure session key establishment, confidentiality, integrity, freshness etc. ) and prevents the major security attacks (such as stolen smart card, user impersonation, sensor node impersonation, etc.) with lesser computational overhead.

## 3.2 Our Contributions

In this chapter, we propose a two factor user authentication protocol based on smart card and elliptic curve cryptography to provide user access to the real-time data by authorizing him/her directly at node level and also making it possible for users to communicate with the nodes in order to have responses to their queries. Our protocol has the following attractive properties:

### 3.2 Our Contributions

---

- Our protocol is secure against different attacks. The resilience against node compromise attack of our protocol is much higher than other existing protocols.
- Our protocol requires less communication, computation, and storage overheads as compared to other existing protocols.
- Higher security along with lower communication, computation and storage overheads make our protocol much suitable for practical applications in WSN & IoT.
- The formal security analysis and simulations conducted using AVISPA(Automated Validation of Internet Security Protocols and Applications) represent that our protocol is secure against active and passive attacks.
- It provides better security as compared with the other related protocols, since it supports mutual authentication between the user and the sensor node, resists denial-of-service attack, privileged-insider attack, smart card breach attack and node capture attack.
- It supports dynamic node addition after initial deployment of nodes in the WSNs & IoT. The proposed protocol does not require to update information for new nodes addition in the user's smart card.
- It supports changing the user's password locally without the help of the gateway node *GW*.
- It provides unconditional security against node capture attacks. That is, compromise of a sensor nodes does not reveal any secret information of other sensor nodes and it does not lead to compromise any other secure communication between the user and the non-compromised nodes in the network.
- It establishes a secret session key between the user and a sensor node for future secret communication of the real-time data inside WSN & IoT between them using the established session key.
- In addition, we have compared the functionality provided by our protocol with other protocols. Overall, our proposed protocol has better performance than other existing protocol.

### 3.3 Review of Choi et al.'s [1] Protocol

In this section, we reproduce the Choi et al.'s [1] protocol for clear understanding of our security analysis. This protocol assumes that the trusted gateway node generates two random number  $x$  and  $y$  and shares a secret key  $SK_{GS} = h(ID_{SN_j} || y)$  with  $j^{th}$  sensor node. The protocol consists of three phases as discussed in the following subsection based on the notations of Table 1.

#### 3.3.1 User Registration

- At this stage,  $\mathcal{U}_i$  chooses  $ID_{\mathcal{U}_i}$ ,  $PW_{\mathcal{U}_i}$ , generates random number  $b_{\mathcal{U}_i}$ , computes  $\overline{PW_{\mathcal{U}_i}} = h(PW_{\mathcal{U}_i} \oplus b_{\mathcal{U}_i})$  and then sends  $ID_{\mathcal{U}_i}$  and  $PW_{\mathcal{U}_i}$  to  $GW_N$  through secure channel.
- $GW_N$  computes  $K_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i} || x) \times P$ ,  $A_{\mathcal{U}_i} = \overline{PW_{\mathcal{U}_i}} \oplus h(x \oplus y)$ ,  $B_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i} || \overline{PW_{\mathcal{U}_i}} || h(x \oplus y))$ ,  $W_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i} || \overline{PW_{\mathcal{U}_i}}) \oplus K_{\mathcal{U}_i}$  and generates a smart card  $SC_{\mathcal{U}_i}$  for  $\mathcal{U}_i$  which stores  $\langle A_{\mathcal{U}_i}, B_{\mathcal{U}_i}, W_{\mathcal{U}_i}, h(.) \rangle$
- $\mathcal{U}_i$  Stores  $b_{\mathcal{U}_i}$  into  $SC_{\mathcal{U}_i}$ .

#### 3.3.2 User Login and Authentication Phase

The description of login and authentication phase involves following five steps:

1.  $\mathcal{U}_i$  puts  $SC_{\mathcal{U}_i}$  into the card reader and gives  $ID_{\mathcal{U}_i}$  and  $PW_{\mathcal{U}_i}$ . Then  $SC_{\mathcal{U}_i}$  computes :  $\overline{PW_{\mathcal{U}_i}} = h(PW_{\mathcal{U}_i} \oplus b_{\mathcal{U}_i})$ ,  $h(x \oplus y) = \overline{PW_{\mathcal{U}_i}} \oplus A_{\mathcal{U}_i}$ . If  $B'_{\mathcal{U}_i} \neq h(ID_{\mathcal{U}_i} || \overline{PW_{\mathcal{U}_i}} || h(x \oplus y))$ , aborts the protocol. Otherwise,  $SC_{\mathcal{U}_i}$  computes  $K_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i} || \overline{PW_{\mathcal{U}_i}}) \oplus W_{\mathcal{U}_i}$ , generates  $r_{\mathcal{U}_i} \in Z_q^*$ , finds  $\mathcal{U}_i$ 's timestamp  $T_{\mathcal{U}_i}$ , computes  $X = r_{\mathcal{U}_i} \times P$ ,  $X' = r_{\mathcal{U}_i} \times K_{\mathcal{U}_i}$ ,  $\omega = h(ID_{\mathcal{U}_i} || h(ID_{SN_j} || h(x \oplus y)) || T_{\mathcal{U}_i})$ ,  $\alpha = h(ID_{\mathcal{U}_i} || ID_{SN_j} || X || X' || T_{\mathcal{U}_i} || \omega)$ . Then,  $\mathcal{U}_i$  sends message  $M_1 = (ID_{\mathcal{U}_i}, ID_{SN_j}, X, T_{\mathcal{U}_i}, \alpha, \omega)$  to  $SN_j$
2.  $SN_j$  retrieves current timestamp  $T'$ . If  $T' - T_{\mathcal{U}_i} > \Delta T$  or  $\omega \neq h(ID_{\mathcal{U}_i} || h(ID_{SN_j} || h(x \oplus y)) || T_{\mathcal{U}_i})$ ,  $SN_j$  abort the protocol (where  $\Delta T$  is the maximum transmission delay). Otherwise,  $SN_j$  generates  $r_{SN_j} \in Z_q^*$ , finds  $SN_j$ 's timestamp  $T_{SN_j}$ , computes  $Y = r_{SN_j} \times P$ ,  $\beta = h(SK_{GS} || \alpha || \omega || ID_{\mathcal{U}_i} || ID_{SN_j} || X || Y || T_{\mathcal{U}_i} || T_{SN_j})$

- .  $SN_j$  sends  $M_2 = \langle X, \alpha, \omega, ID_{\mathcal{U}_i}, ID_{SN_j}, Y, T_{\mathcal{U}_i}, T_{SN_j}, \beta \rangle$  to  $GW_N$ .
3.  $GW_N$  retrieves current timestamp  $T''$ . Checks if  $T'' - T_{SN_j} > \Delta T$ ,  $GW_N$  abort this phase, otherwise calculates  $X' = h(ID_{\mathcal{U}_i} || x) \times X$ . If  $\beta \neq h(SK_{GS} || ID_{\mathcal{U}_i} || X || \alpha || \omega || ID_{SN_j} || Y || T_{\mathcal{U}_i} || T_{SN_j})$  or if  $\alpha \neq h(ID_{\mathcal{U}_i} || ID_{SN_j} || X || X' || T_{\mathcal{U}_i} || \omega)$ ,  $GW_N$  abort this phase. Otherwise, find  $GW_N$ 's timestamp  $T_{GW_N}$  and compute  $\gamma = h(SK_{GS} || ID_{\mathcal{U}_i} || X || \alpha || ID_{SN_j} || Y || T_{\mathcal{U}_i} || T_{SN_j} || T_{GW_N})$ ,  $\delta = h(ID_{\mathcal{U}_i} || X || X' || T_{\mathcal{U}_i} || ID_{SN_j} || Y || T_{SN_j})$ . Then,  $GW_N$  sends  $M_3 = \langle T_{GW_N}, \gamma, \delta \rangle$  to  $SN_j$ .
  4.  $SN_j$  retrieves current timestamp  $T'''$ . If  $T''' - T_{GW_N} > \Delta T$  or if  $\gamma \neq h(SK_{GS} || ID_{\mathcal{U}_i} || \alpha || ID_{SN_j} || Y || X || T_{\mathcal{U}_i} || T_{SN_j} || T_{GW_N})$ , abort the protocol. Otherwise,  $SN_j$  computes  $K_{SU} = r_{SN_j} \times X$  and retrieve  $r_{SN_j}$ 's timestamp  $T_{SN_j}'$ . Then computes  $\tau = h(Y || \delta || K_{SU} || T_{SN_j}')$  and generate a session key  $sk = h(X || Y || K_{SU})$ .  $SN_j$  sends  $M_4 \langle Y, T_{SN_j}, T_{SN_j}', \delta, \tau \rangle$  to  $\mathcal{U}_i$ .
  5.  $\mathcal{U}_i$  retrieves current timestamp  $T''''$ , computes  $K_{US} = r_{\mathcal{U}_i} \times Y$ . If  $T'''' - T_{SN_j}' > \Delta T$  or  $\delta \neq h(ID_{\mathcal{U}_i} || X || X' || T_{\mathcal{U}_i} || ID_{SN_j} || Y || T_{SN_j})$  or  $\tau \neq h(Y || T_{SN_j}' || \delta || K_{US})$ ,  $\mathcal{U}_i$  abort the protocol. Otherwise,  $\mathcal{U}_i$  establishes session key  $sk = h(Y || X || K_{US})$  with  $SN_j$ .

### 3.3.3 User's Password Update Phase

In this stage,  $\mathcal{U}_i$  puts its  $SC_{\mathcal{U}_i}$  into card reader, gives  $ID_{\mathcal{U}_i}, PW_{\mathcal{U}_i}$  and then new password  $PW'_{\mathcal{U}_i}$ .  $SC_{\mathcal{U}_i}$  calculates  $\overline{PW}_{\mathcal{U}_i} = h(PW_{\mathcal{U}_i} \oplus b_{\mathcal{U}_i}), h(x \oplus y) = A_{\mathcal{U}_i} \oplus \overline{PW}_{\mathcal{U}_i}$ . If  $B_{\mathcal{U}_i} \neq (B'_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i} || \overline{PW}_{\mathcal{U}_i} || h(x \oplus y)))$ , abort this phase. Otherwise,  $SC_{\mathcal{U}_i}$  computes:  $K_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i} || \overline{PW}_{\mathcal{U}_i}) \oplus W_{\mathcal{U}_i}, \overline{PW'}_{\mathcal{U}_i} = h(PW'_{\mathcal{U}_i} \oplus b_{\mathcal{U}_i}), A_{\mathcal{U}_i}' = h(x \oplus y) \oplus \overline{PW'}_{\mathcal{U}_i}, B'_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i} || \overline{PW'}_{\mathcal{U}_i} || h(x \oplus y)), W'_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i} || \overline{PW'}_{\mathcal{U}_i}) \oplus K_{\mathcal{U}_i}$ . Replace  $A_{\mathcal{U}_i}, B_{\mathcal{U}_i}$  and  $W_{\mathcal{U}_i}$  of  $SC_{\mathcal{U}_i}$  with  $A'_{\mathcal{U}_i}, B'_{\mathcal{U}_i}$  and  $W'_{\mathcal{U}_i}$  respectively.

### 3.4 Cryptanalysis of Choi et al.'s [1] Scheme

In this section, we first consider some assumptions under which the authentication protocol for WSN is analyzed. Then we show that Choi et.al.'s [1] protocol is insecure against various known security attacks.

#### 3.4.1 Assumptions

- Sensor node may not fix up with tamper - resistant hardware and if a node is captured by an adversary  $\mathcal{A}$ , all the prominent and confidential information saved in its memory can be accessed by the adversary. If the sensor nodes are tamper resistant,  $\mathcal{A}$  can retrieve the information stored in the memory by measuring the power consumption of the captured sensor nodes.
- Base station or gateway cannot be compromised, by the adversary.
- Adversary can intercept the public communication channel, inject packets and reply the already transmitted information.
- Adversary can capture the smart card of user and it can extract the sensitive information like user identity and password stored in the card through a power analysis attack.

#### 3.4.2 Attacks on Choi et al.'s protocol

In this section, we elaborate that Choi et al.'s [1] protocol has following security pitfalls:

##### 3.4.2.1 Stolen Smart Card Attacks:

Confidential information saved in stolen smart card can be retrieved by measuring its power consumption as described in Kocher et al.'s [14] protocol. The adversary  $\mathcal{A}$  can capture two smart cards  $SC_{\mathcal{U}_i}$  and  $SC_V$  of user  $\mathcal{U}_i$  and  $V$  respectively and then using power consumption attacks such as differential power analysis (DPA) and simple power analysis (SPA),  $\mathcal{A}$  can find out the value of  $\{A_{\mathcal{U}_i}, B_{\mathcal{U}_i}, W_{\mathcal{U}_i}, h(\cdot), b_{\mathcal{U}_i}\}$  and  $\{A_V, B_V, W_V, h(\cdot), b_V\}$  from  $SC_{\mathcal{U}_i}$  and  $SC_V$  respectively. Since,  $A_{\mathcal{U}_i} = \overline{PW_{\mathcal{U}_i}} \oplus$

### 3.4 Cryptanalysis of Choi et al.'s [1] Scheme

$h(x \oplus y)$ ,  $A_V = \overline{pw_V} \oplus h(x \oplus y)$ , applying frequency analysis attacks on  $A_{U_i}$ ,  $A_V$  and  $A_{U_i} \oplus A_V = \overline{PW_{U_i}} \oplus h(x \oplus y) \oplus \overline{PW_V} \oplus h(x \oplus y) = \overline{PW_{U_i}} \oplus \overline{PW_V}$ , the adversary  $\mathcal{A}$  can find out the value of  $h(x \oplus y)$ .

Therefore,  $\mathcal{A}$  can compute :  $\overline{PW_{U_i}} = A_{U_i} \oplus h(x \oplus y)$ ,  $B_{U_i} = h(ID_{U_i} || \overline{PW_{U_i}} || h(x \oplus y))$ , [ $ID_{U_i}$  is known by eavesdropping  $M_1$ ] and  $K_{U_i} = W_{U_i} \oplus h(ID_{U_i} || \overline{PW_{U_i}})$ . Now,  $\overline{PW_{U_i}} = h(PW_{U_i} \oplus b_{U_i})$  and  $b_{U_i}$  are known to  $\mathcal{A}$ ,  $PW_{U_i}$  have low entropy and easy to break by dictionary attack. Therefore, with the help of off - line password attacks on  $\overline{PW_{U_i}}$ ,  $\mathcal{A}$  can figure out user's password  $PW_{U_i}$ .

#### 3.4.2.2 Resilience against Sensor Node Capture Attack :

It is evaluated by finding the fraction of total communication that are compromised by capturing  $m$  number of sensor node excluding the communication in which the captured  $m$  sensor nodes are involved. It measures the effect of  $m$  compromised sensor node on the rest of the network. The probability of decrypting the encrypted communication between  $SN_j$  and user  $U_i$  can be denoted as  $P_e(m)$ . If  $P_e(m) = 0$ , the user authentication protocol will be secure against node capture attack.

Suppose  $m = 1$  (i.e., one sensor node  $S_i$  is captured by  $\mathcal{A}$ ). Then,  $\mathcal{A}$  can find out  $h(x \oplus y)$  which is store in the memory of  $SN_i$  for computing the value of  $\omega$  and also for verifying it with the  $\omega$  of  $M_1$ . If  $\mathcal{A}$  have smart card of  $U_i$ ,  $\mathcal{A}$  can extract  $\{A_{U_i}, B_{U_i}, W_{U_i}, h(\cdot), b_{U_i}\}$ . Therefore,  $\mathcal{A}$  can find out :  $\overline{PW_{U_i}} = A_{U_i} \oplus h(x \oplus y)$ ,  $B_{U_i} = h(ID_{U_i} || \overline{PW_{U_i}} || h(x \oplus y))$ , [ $ID_{U_i}$  is known by eavesdropping  $M_1$ ] and  $K_{U_i} = W_{U_i} \oplus h(ID_{U_i} || \overline{PW_{U_i}})$ . Now,  $\overline{PW_{U_i}} = h(PW_{U_i} \oplus b_{U_i})$  and  $b_{U_i}$  are known,  $PW_{U_i}$  have low entropy and easy to break by dictionary attack, so after executing off - line password attacks on  $\overline{PW_{U_i}}$ ,  $\mathcal{A}$  can figure out user's password  $PW_{U_i}$ . Then,  $\mathcal{A}$  can get authenticated with sensor node  $SN_j (i \neq j)$  by sending  $M_1 = \langle ID_{U_i}, ID_{SN_j}, X, T_{U_i}, \alpha, \omega \rangle$  to  $SN_j$  and receiving  $M_4 = \langle Y, T_{SN_j}, T_{SN_j'}, \gamma, \tau \rangle$  from  $SN_j$ . Therefore,  $\mathcal{A}$  can establish a session key  $sk_j = h(X || Y || K_{US_j})$  with  $SN_j$ . It is clear that  $\mathcal{A}$  can get authenticated and establish the session key between  $U_i$  and any other uncompromised sensor node  $SN_j$  using a compromised sensor node  $SN_i$  and a stolen smart card. Thus,  $P_e(m) \neq 0$ . So, we can say that Choi et al.'s [1] protocol is not resilient against node capture attack.

#### 3.4.2.3 Sensor Node Energy Exhausting Attack:

Sensor node has constrained resources, therefore the computational cost of sensor node is an important consideration. To increase the lifetime of sensor node we need to eliminate unnecessary computation on sensor node. Sometimes adversary's intention is to exhaust the energy of the sensor node in order to slow down or interrupt the network. This type of attack is also known as denial of service attack. In Choi et al. protocol  $SN_j$  performs various operations such as one way hashing, random number generation and scalar point multiplication of ECC. The computational cost of point multiplication is more than performing one way hashing and generating random number.

Suppose  $\mathcal{A}$  eavesdrops  $M_1 = \langle ID_{\mathcal{U}_i}, ID_{SN_i}, X, T_{\mathcal{U}_i}, \alpha, \omega \rangle$ , extracts  $h(x \oplus y)$  from captured sensor node  $SN_i$  and  $\mathcal{A}$  computes  $\omega' = h(ID_{\mathcal{U}_i} || h(ID_{SN_j} || h(x \oplus y) || T_{\mathcal{U}_i}))$ . Then,  $\mathcal{A}$  can send  $M'_1 = \langle ID_{\mathcal{U}_i}, ID_{SN_j}, X, T_{\mathcal{U}_i}, \alpha, \omega' \rangle$  to  $SN_j$ . Sensor node  $SN_j$  computes  $\omega'' = h(ID_{\mathcal{U}_i} || h(ID_{SN_j} || h(x \oplus y) || T_{\mathcal{U}_i}))$ , checks if  $\omega' = \omega''$ , generates  $r_s \in Z_q^*$ , retrieves  $SN_j$ 's timestamp  $T_{SN_j}$ , computes  $Y = r_{SN_j} \times P$ ,  $\beta = h(SK_{GS} || ID_{\mathcal{U}_i} || \alpha || \omega' || ID_{SN_j} || X || Y || T_{\mathcal{U}_i} || T_{SN_j})$  and sends  $M_2 = (ID_{\mathcal{U}_i}, \alpha, \omega', ID_{SN_j}, X, Y, T_{\mathcal{U}_i}, T_{SN_j}, \beta)$  to  $GWN$ . In this way attacker can send more fake messages  $M''_1, M'''_1, M''''_1$  etc. to  $SN_j$  and make  $SN_j$  to perform unnecessary hashing, generating random number and scalar point multiplication in order to exhaust the battery power of  $SN_j$ .

## 3.5 Proposed Protocol

In general, whenever a sensor node  $SN_j$  process a query, it should be able to verify that the query comes from a legitimate user  $\mathcal{U}_i$ . We call this problem authenticated querying. More formally, a WSN & IoT enables authenticated querying if it satisfies the following properties (perhaps, with some large probability):

**Safety:** If a sensor node  $SN_j$  processes the query  $Q_{\mathcal{U}_i}$ , then  $Q_{\mathcal{U}_i}$  was posted by a legitimate user  $\mathcal{U}_i$ .

**Liveness:** Any query  $Q_{\mathcal{U}_i}$  posted by a legitimate user  $\mathcal{U}_i$  is processed by at least all sensors  $SN_k \in SN$ , where  $SN$  is the set of sensors which must process the query in order to give the required answer to the user.

Before issuing any queries to or access data from sensor devices of WSNs& IoT, the user  $\mathcal{U}_i$  has to register with the  $GW N$  node of the network. Upon successful registration, the user can submit query to the WSNs & IoT at any time within a predefined or administrative configurable period. The basic idea of the protocol is that a user will receive a personalized smart card from the  $GW N$ -node at the time of the registration process and then, with the help of users password and smart card the user can login to the sensor node and access data from the network. The security analysis of Choi et al.'s [1] protocol clears that their protocol is vulnerable because of storing  $h(x \oplus y)$  in sensor node (which helps in node capture attacks), storing  $A_{\mathcal{U}_i} = \overline{PW_{\mathcal{U}_i}} \oplus h(x \oplus y)$  in smart card (it helps in stolen smart card attacks because the value of  $h(x \oplus y)$  is same for different smart card which helps in frequency analysis attacks) and verification of  $\omega$  on sensor node (it helps in sensor energy exhausting attacks). To overcome these security pitfalls, we propose a protocol involving three phases (1) User Registration (2) Login, authentication and session key establishment (3) Password Update Phase. We assume that (i) the trusted gateway node generates two 1024 bit secret key  $x, y$  and establishes a long term secret key  $K_{GS_n} = h(ID_{SN_j} \oplus y)$  with  $SN_j$  (ii)  $P \in E(F_p)$  is shared with  $\mathcal{U}_i, SN_j$  and  $GW N$ . The three phase of our proposed protocol (based on the notations of Table 1) are as follows:

#### 3.5.1 User Registration Phase:

$\mathcal{U}_i$  needs to register with  $GW N$  for retrieving real time and confidential data from the sensor node. This phase has three steps (Step R1, R2 and R3) as elaborated in following Table 3.1:

**Step R1:** In this step, the user  $\mathcal{U}_i$  chooses  $ID_{\mathcal{U}_i}, PW_{\mathcal{U}_i}$ , generates a 1024 bit random number  $b_{\mathcal{U}_i}$  and computes  $\overline{PW_{\mathcal{U}_i}} = h(PW_{\mathcal{U}_i} \oplus b_{\mathcal{U}_i})$ . Then  $\mathcal{U}_i$  send  $\langle ID_{\mathcal{U}_i}, \overline{PW_{\mathcal{U}_i}} \rangle$  to  $GW N$

**Step R2:** After receiving  $\langle ID_{\mathcal{U}_i}, \overline{PW_{\mathcal{U}_i}} \rangle$ , the gateway node  $GW N$  computes  $K_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i} || X) \times P$ ,  $A_{\mathcal{U}_i} = \overline{PW_{\mathcal{U}_i}} \oplus h(ID_{\mathcal{U}_i} \oplus y)$ ,  $B_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i} || \overline{PW_{\mathcal{U}_i}} || h(ID_{\mathcal{U}_i} \oplus y))$ ,  $W_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i} || \overline{PW_{\mathcal{U}_i}}) \oplus K_{\mathcal{U}_i}$ . Then,  $GW N$  stores  $\langle P, A_{\mathcal{U}_i}, B_{\mathcal{U}_i}, W_{\mathcal{U}_i}, h(\cdot) \rangle$  into  $SC_{\mathcal{U}_i}$ . Finally,  $GW N$  issues  $SC_{\mathcal{U}_i}$  to  $\mathcal{U}_i$



### 3.5 Proposed Protocol

**Step R3:** After receiving the message  $\langle h(), A_{u_i}, B_{u_i}, W_{u_i} \rangle$  from  $GW N$ , the user  $u_i$  store  $b_{u_i}$  into  $SC_{u_i}$

**Table 3.1:** User Registration Phase:

User ( $u_i$ )	Gateway Node $GW N$
<p>Step 1. <math>u_i</math> chooses <math>ID_{u_i}, PW_{u_i}</math>. Then, generates a 1024 bit random number <math>RN_{u_i}</math> and computes <math>\overline{PW_{u_i}} = h(PW_{u_i} \oplus RN_{u_i})</math>.</p> <p>Finally, <math>u_i</math> send <math>\langle ID_{u_i}, \overline{PW_{u_i}} \rangle</math> to <math>GW N</math></p> <p style="text-align: center;"><math>\xrightarrow{\text{SecureChannel}}</math></p>	<p>Step 2: <math>GW N</math> computes</p> $K_{u_i} = h(ID_{u_i}    x) \times P,$ $A_{u_i} = \overline{PW_{u_i}} \oplus h(ID_{u_i} \oplus y),$ $B_{u_i} = h(ID_{u_i}    \overline{PW_{u_i}}    h(ID_{u_i} \oplus y)),$ $W_{u_i} = h(ID_{u_i}    \overline{PW_{u_i}}) \oplus K_{u_i}$ <p>Then, <math>GW N</math> stores <math>\langle P, A_{u_i}, B_{u_i}, W_{u_i}, h(.) \rangle</math> into <math>SC_{u_i}</math>.</p> <p style="text-align: center;"><math>\xleftarrow{\text{SecureChannel}}</math></p> <p>Finally, <math>GW N</math> issue <math>SC_{u_i}</math> to <math>u_i</math></p>
<p>Step 3: <math>u_i</math> store <math>RN_{u_i}</math> into <math>SC_{u_i}</math></p>	

#### 3.5.2 Login, authentication and session key establishment Phase:

This stage involves five steps in order to get login to  $GW N$ , mutually authenticate  $u_i, SN_j, GW N$  and establish  $SK_{ij}$  (a secret session key) between  $u_i$  and  $SN_j$ . Each step checks validity of few conditions. If the condition is true the entity ( $u_i, SN_j, GW N$ ) performs the next computation, otherwise it aborts the protocol.  $u_i$  performs step A1, A5 and  $SN_j$  executes step A2, A4 and step A3 is performed by  $GW N$ . The details of these steps are shown in following Table-3.2:

**Step A1:** In this step, the user  $u_i$  inserts the smart card  $SC_{u_i}$  into the card reader. Subsequently,  $u_i$  enters his/her unique identity  $ID_{u_i}$ , the secret password  $PW_{u_i}$ . Then,  $u_i$  computes  $\overline{PW_{u_i}} = h(PW_{u_i} \oplus RN_{u_i})$ ,  $h(ID_{u_i} \oplus y) = \overline{PW_{u_i}} \oplus A_{u_i}$

and  $B'_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i} || \overline{PW_{\mathcal{U}_i}} || h(ID_{\mathcal{U}_i} \oplus y))$ . Then, it verifies the correctness of  $B'_{\mathcal{U}_i} = B_{\mathcal{U}_i}$ , computes  $K_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i} || \overline{PW_{\mathcal{U}_i}}) \oplus W_{\mathcal{U}_i}$  and generates a random number  $R_{\mathcal{U}_i} \in Z_q^*$ . Afterwards, the user  $\mathcal{U}_i$  finds the current timestamp  $T_{\mathcal{U}_i}$  and computes  $X_{\mathcal{U}_i} = R_{\mathcal{U}_i} \times P$ ,  $X'_{\mathcal{U}_i} = R_{\mathcal{U}_i} \times K_{\mathcal{U}_i}$ ,  $T'_{\mathcal{U}_i} = P \oplus T_{\mathcal{U}_i}$  and  $\alpha = h(ID_{\mathcal{U}_i} || ID_{SN_j} || X_{\mathcal{U}_i} || X'_{\mathcal{U}_i} || T_{\mathcal{U}_i})$ . Finally,  $\mathcal{U}_i$  send  $M_3 = \langle ID_{\mathcal{U}_i}, ID_{SN_j}, X_{\mathcal{U}_i}, D, T'_{\mathcal{U}_i}, \alpha \rangle$  to  $SN_j$

**Step A2:** In this step, after receiving the message  $M_1$ , the sensor node  $SN_j$  first computes  $T_{\mathcal{U}_i} = P \oplus T'$  and examines the satisfactory condition of the time-stamp  $T' - T_{\mathcal{U}_i} \leq \Delta T$ . Then, generates  $R_{SN_j} \in Z_q^*$  and finds the current timestamp  $T_{SN_j}$ . Afterwards, it computes  $Y_{SN_j} = R_{SN_j} \times P$ , and  $\beta = h(K_{GS_j} || T_{\mathcal{U}_i} || Y_{SN_j} || T_{SN_j})$ . Finally,  $SN_j$  computes  $M_4 = \langle ID_{\mathcal{U}_i}, X_{\mathcal{U}_i}, T'_{\mathcal{U}_i}, \alpha, ID_{SN_j}, Y_{SN_j}, T_{SN_j}, \beta \rangle$  and sends  $M_4$  to  $GWN$

**Step A3:** After getting the message  $M_4$  from the  $SN_j$ , the  $GWN$  tests the legitimacy of the timestamp  $T_{SN_j}$  by the condition  $T'' - T_{SN_j} \leq \Delta T$ . Afterwards, it computes  $F_{\mathcal{U}_i} = P \oplus T'$ ,  $X'_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i} || x) \times X_{\mathcal{U}_i}$  and verifies the correctness of  $\alpha = h(ID_{\mathcal{U}_i} || ID_{SN_j} || X_{\mathcal{U}_i} || X'_{\mathcal{U}_i} || T_{\mathcal{U}_i})$  and  $\beta = h(K_{GS_j} || T_{\mathcal{U}_i} || Y_{SN_j} || T_{SN_j})$ . Then, finds the current timestamp  $T_{GWN}$  and computes  $\gamma = h(K_{GS_j} || \alpha || Y_{SN_j} || T_{SN_j} || T_{GWN})$ ,  $\delta = h(X'_{\mathcal{U}_i} || T_{\mathcal{U}_i} || ID_{SN_j} || Y_{SN_j} || T_{SN_j})$ . Finally,  $GWN$  Sends  $M_5 = \langle T_{GWN}, \gamma, \delta \rangle$  to  $SN_j$

**Step A4:** After getting the message  $M_5$  from the  $GWN$ ,  $SN_j$  tests the validity of the timestamp  $T_{GWN}$  by the condition  $T''' - T_{GWN} \leq \Delta T$ , checks the condition  $\gamma = h(K_{GS_j} || \alpha || Y_{SN_j} || T_{SN_j} || T_{GWN})$ . Afterwards, find  $K_{ij} = R_{SN_j} \times X_{\mathcal{U}_i}$ ,  $\tau = h(T'_{SN_j} || \delta || K_{ij})$ ,  $SK_{ij} = h(K_{ij})$ . Then,  $SN_j$  Send  $M_6 = \langle Y_{SN_j}, T_{SN_j}, T'_{SN_j}, \delta, \tau \rangle$  to  $\mathcal{U}_i$

**Step A5:** Finally, after getting message  $M_4$  from  $SN_j$ ,  $\mathcal{U}_i$  checks the validity of the timestamp  $T_{SN_j}$  by the condition  $T'''' - T'_{SN_j} \leq \Delta T$ . Then, it computes  $\delta = h(X'_{\mathcal{U}_i} || T_{\mathcal{U}_i} || ID_{SN_j} || Y_{SN_j} || T_{SN_j})$ , and find  $K_{ij} = R_{\mathcal{U}_i} \times Y_{SN_j}$ . Afterwards, checks the condition  $\tau = h(T'_{SN_j} || \delta || K_{ij})$ . Finally, computes and establishes the session key  $SK_{ij} = h(K_{ij})$  with  $SN_j$

The sequence diagram of the message transmission for the user registration, authentication and key establishment phase is shown in following Figure 3.1.

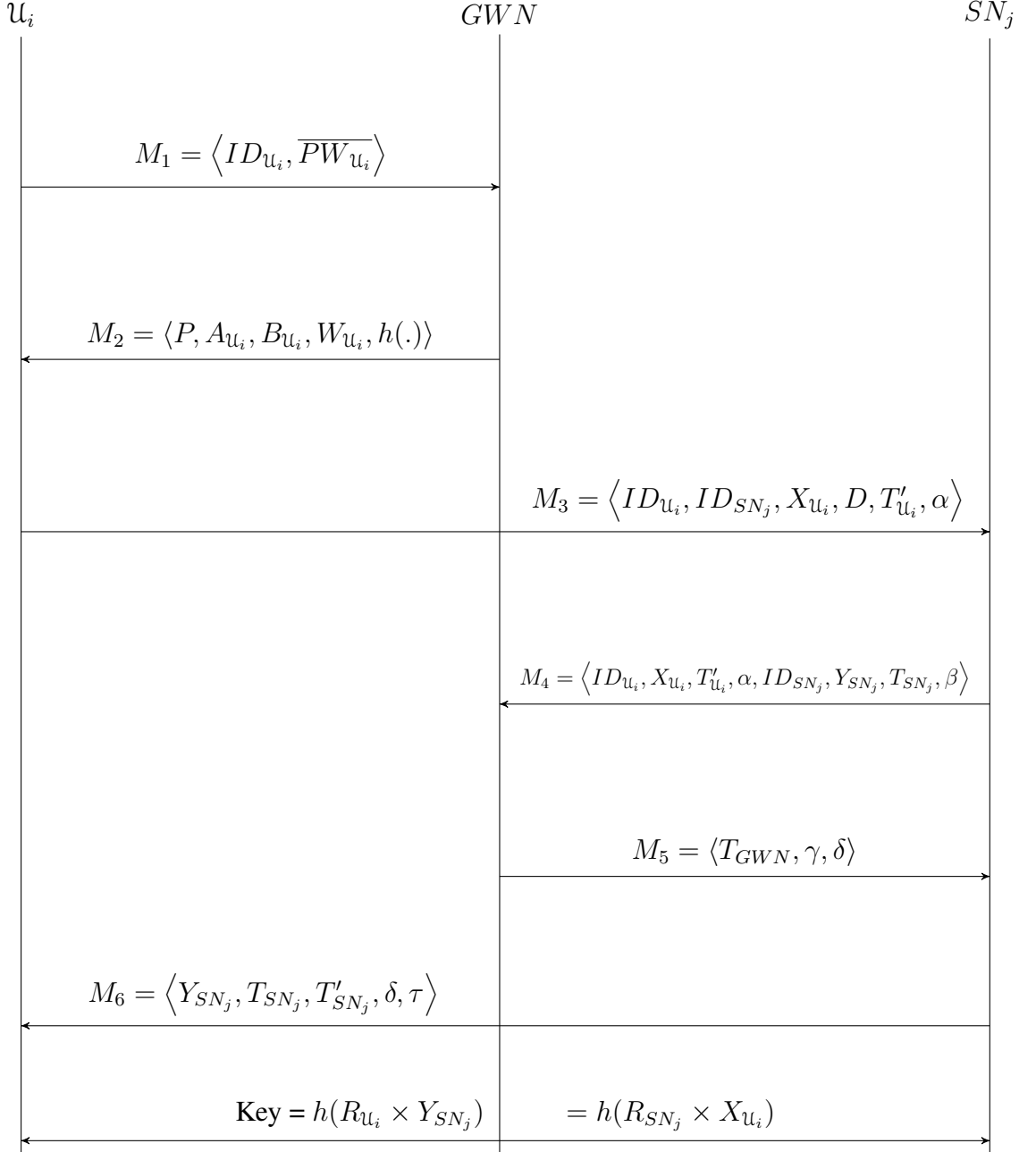
### 3.5 Proposed Protocol

**Table 3.2:** User Authentication and Key Establishment Phase:

<p><b>Step 4:</b> for <math>\mathcal{U}_i</math></p> <p><math>\mathcal{U}_i</math> first inputs the smart card, <math>ID_{\mathcal{U}_i}</math>, <math>PW_{\mathcal{U}_i}</math> and computes <math>\overline{PW}_{\mathcal{U}_i} = h(PW_{\mathcal{U}_i} \oplus RN_{\mathcal{U}_i})</math>.  <math>h(ID_{\mathcal{U}_i} \oplus y) = \overline{PW}_{\mathcal{U}_i} \oplus A_{\mathcal{U}_i}</math>,  <math>B'_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i}    \overline{PW}_{\mathcal{U}_i}    h(ID_{\mathcal{U}_i} \oplus y))</math>,  Checks <math>B'_{\mathcal{U}_i} = B_{\mathcal{U}_i}</math>  <b>if</b> <math>B'_{\mathcal{U}_i} = B_{\mathcal{U}_i}</math> <b>then</b>      Find <math>K_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i}    \overline{PW}_{\mathcal{U}_i}) \oplus W_{\mathcal{U}_i}</math>,      Generate <math>R_{\mathcal{U}_i} \in Z_q^*</math>      Find current time-stamp <math>T_{\mathcal{U}_i}</math> and compute  <math>X_{\mathcal{U}_i} = R_{\mathcal{U}_i} \times P</math>, <math>X'_{\mathcal{U}_i} = R_{\mathcal{U}_i} \times K_{\mathcal{U}_i}</math>, <math>T'_{\mathcal{U}_i} = P \oplus T_{\mathcal{U}_i}</math>,  <math>\alpha = h(ID_{\mathcal{U}_i}    ID_{SN_j}    X_{\mathcal{U}_i}    X'_{\mathcal{U}_i}    T_{\mathcal{U}_i})</math>.      Send <math>M_3 = \langle ID_{\mathcal{U}_i}, ID_{SN_j}, X_{\mathcal{U}_i}, D, T'_{\mathcal{U}_i}, \alpha \rangle</math> to <math>SN_j</math>      ViaPublicChannel  <b>end</b>  <b>else</b>      Reject <math>\mathcal{U}_i</math>  <b>end</b></p>	<p><b>Step 5:</b> for <math>SN_j</math></p> <p><math>SN_j</math> computes <math>T_{\mathcal{U}_i} = P \oplus T'</math></p> <p><b>if</b> <math>T' - T_{\mathcal{U}_i} \leq \Delta T</math> <b>then</b>      Generates <math>R_{SN_j} \in Z_q^*</math>      Find Current timestamp <math>T_{SN_j}</math>,  <math>Y_{SN_j} = R_{SN_j} \times P</math>,  <math>\beta = h(K_{GS_j}    T_{\mathcal{U}_i}    Y_{SN_j}    T_{SN_j})</math>. Then,  <math>SN_j</math> computes  <math>M_4 = \langle ID_{\mathcal{U}_i}, X_{\mathcal{U}_i}, T'_{\mathcal{U}_i}, \alpha, ID_{SN_j}, Y_{SN_j}, T_{SN_j}, \beta \rangle</math>      Send <math>M_4</math> to <math>GW_N</math>      ViaPublicChannel  <b>end</b>  <b>else</b>      Reject <math>\mathcal{U}_i</math>  <b>end</b></p>
<p><b>Step 6:</b> for <math>GW_N</math></p> <p><b>if</b> <math>T'' - T_{SN_j} \leq \Delta T</math> <b>then</b>      Find <math>F_{\mathcal{U}_i} = P \oplus T'</math>,  <math>X'_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i}    x) \times X_{\mathcal{U}_i}</math>        <b>if</b> <math>\alpha = h(ID_{\mathcal{U}_i}    ID_{SN_j}    X_{\mathcal{U}_i}    X'_{\mathcal{U}_i}    T_{\mathcal{U}_i})</math> <b>then</b>          <b>if</b> <math>\beta = h(K_{GS_j}    T_{\mathcal{U}_i}    Y_{SN_j}    T_{SN_j})</math> <b>then</b>              Find current timestamp <math>T_{GW_N}</math>              <math>\gamma = h(K_{GS_j}    \alpha    Y_{SN_j}    T_{SN_j}    T_{GW_N})</math>,              <math>\delta = h(X'_{\mathcal{U}_i}    T_{\mathcal{U}_i}    ID_{SN_j}    Y_{SN_j}    T_{SN_j})</math>.              Send <math>M_5 = \langle T_{GW_N}, \gamma, \delta \rangle</math> to <math>SN_j</math>              ViaPublicChannel              <b>end</b>          <b>else</b>              Reject <math>\mathcal{U}_i</math>          <b>end</b>      <b>end</b>      <b>else</b>          Reject <math>\mathcal{U}_i</math>      <b>end</b>  <b>end</b>  <b>else</b>      Reject <math>\mathcal{U}_i</math>  <b>end</b></p>	<p><b>Step 7:</b> for <math>SN_j</math></p> <p><math>SN_j</math> Checks <b>if</b> <math>T''' - T_{GW_N} \leq \Delta T</math> <b>then</b>      Checks      <b>if</b> <math>\gamma = h(K_{GS_j}    \alpha    Y_{SN_j}    T_{SN_j}    T_{GW_N})</math> <b>then</b>          Find <math>K_{ij} = R_{SN_j} \times X_{\mathcal{U}_i}</math>,  <math>\tau = h(T'_{SN_j}    \delta    K_{ij})</math>,  <math>SK_{ij} = h(K_{ij})</math>          Send <math>M_6 = \langle Y_{SN_j}, T_{SN_j}, T'_{SN_j}, \delta, \tau \rangle</math> to <math>\mathcal{U}_i</math>          ViaPublicChannel      <b>end</b>      <b>else</b>          Reject <math>\mathcal{U}_i</math>      <b>end</b>  <b>end</b>  <b>else</b>      Reject <math>\mathcal{U}_i</math>  <b>end</b></p>
<p><b>Step 8:</b> for <math>\mathcal{U}_i</math></p> <p><math>\mathcal{U}</math> Checks <math>T'''' - T'_{SN_j} \leq \Delta T</math>,  <math>\delta = h(X'_{\mathcal{U}_i}    T_{\mathcal{U}_i}    ID_{SN_j}    Y_{SN_j}    T_{SN_j})</math>,  Find <math>K_{ij} = R_{\mathcal{U}_i} \times Y_{SN_j}</math>  Checks <math>\tau = h(T'_{SN_j}    \delta    K_{ij})</math>  <math>SK_{ij} = h(K_{ij})</math> [<math>\mathcal{U}_i</math> Establishes session key <math>SK_{ij}</math> with <math>SN_j</math>]</p>	

### 3.5 Proposed Protocol

**Figure 3.1:** Sequence Diagram 1 for Registration, Authentication and Key Establishment



### 3.5.3 User's Password Update Phase:

In this stage, the user  $\mathcal{U}_i$  puts  $SC_{\mathcal{U}_i}$  into the card reader, gives  $ID_{\mathcal{U}_i}, PW_{\mathcal{U}_i}$  and then new password  $PW'_{\mathcal{U}_i}$ .  $SC_{\mathcal{U}_i}$  computes  $\overline{PW}_{\mathcal{U}_i} = h(PW_{\mathcal{U}_i} \oplus b_{\mathcal{U}_i})$ ,  $h(ID_{\mathcal{U}_i} \oplus y) = \overline{PW}_{\mathcal{U}_i} \oplus A_{\mathcal{U}_i}$ ,  $B'_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i} || \overline{PW}_{\mathcal{U}_i} || h(ID_{\mathcal{U}_i} \oplus y))$ , Checks  $B'_{\mathcal{U}_i} = B_{\mathcal{U}_i}$ , Computes  $K_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i} || \overline{PW}_{\mathcal{U}_i}) \oplus W_{\mathcal{U}_i}$ ,  $\overline{PW}_{\mathcal{U}_i} = h(PW'_{\mathcal{U}_i} || b_{\mathcal{U}_i})$ ,  $A'_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i} \oplus y) \oplus \overline{PW'_{\mathcal{U}_i}}$ ,  $B'_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i} || \overline{PW'_{\mathcal{U}_i}} || h(ID_{\mathcal{U}_i} \oplus y))$ ,  $W'_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i} || \overline{PW'_{\mathcal{U}_i}}) \oplus K_{\mathcal{U}_i}$ . Replace  $A_{\mathcal{U}_i}, B_{\mathcal{U}_i}, W_{\mathcal{U}_i}$  with  $A'_{\mathcal{U}_i}, B'_{\mathcal{U}_i}, W'_{\mathcal{U}_i}$ .

## 3.6 Security Analysis and Comparison

In this section, we present that our protocol can resist various well known attacks of WSNs. The attacks which we resist through our protocol are as follow:

### 3.6.1 Stolen Smart Card Attacks

In case of Choi et al.'s[1] protocol the value  $A_{\mathcal{U}_i} = \overline{PW}_{\mathcal{U}_i} \oplus h(x \oplus y)$  of smart card of user  $\mathcal{U}_i$  and the value  $A_V = \overline{PW}_V \oplus h(x \oplus y)$  of smart card of user  $V$  helps in frequency analysis attack because  $A_{\mathcal{U}_i}$  and  $A_V$  consist of same parameter  $h(x \oplus y)$ . In case of our protocol  $A_{\mathcal{U}_i}$  and  $A_V$  are not build with the same parameter. Therefore, frequency analysis attack is not easy in our protocol. So, even if the adversary  $\mathcal{A}$  have smart card of  $\mathcal{U}_i$  and  $V$ ,  $\mathcal{A}$  can not find out the password.

### 3.6.2 Resilience Against Sensor Node Capture Attack and Energy Exhausting Attack

Choi et al. [1] protocol suggest to store  $h(x \oplus y)$  in sensor node's memory in order to resist energy exhausting attacks. But it is creating more insecurity because if  $\mathcal{A}$  finds  $h(x \oplus y)$  from a captured sensor node,  $\mathcal{A}$  can find out the password associated with a stolen smart card and latter it can get authenticated and establish session key with any sensor node. In this way  $\mathcal{A}$  succeeds in node capture attack. In order to resist energy exhausting attack we compute  $T_{\mathcal{U}_i} = P \oplus T'_{\mathcal{U}_i}$  and checks the validity of  $T' - T_{\mathcal{U}_i} \leq \Delta T$ . If  $T' - T_{\mathcal{U}_i} > \Delta T$ ,  $SN_j$  aborts the protocol and avoid unnecessary computation to be safe from energy exhausting attacks. Since, we are not storing any secret parameter

### 3.6 Security Analysis and Comparison

into the sensor node which helps in finding the user's password therefore our protocol is resilient against node capture attacks.

#### 3.6.3 Other Security Attacks

Our protocol is secure against replay attack because each message which has been delivered through insecure channel contains a valid timestamp. We resist man-in-the-middle attack by including authenticating parameter  $\alpha, \beta, \gamma, \delta, \tau$  and unique long term secret key for each sensor node. User impersonation attack is not possible because attacker does not know the value of  $X'_{U_i}$  to find out the value of  $\alpha$ . Since, we are using unique long term secret key for each sensor node therefore sensor impersonation attack is not possible. Our protocol yields mutual authentication and session key establishment which can help in resisting few more possible security attacks.

**Table 3.3:** Comparison of protocols based on security features.

Security Feature	Sun [108]	Xue et al. [79]	Shi et al. [109]	Choi et al. [1]	Li et al. [98]	Zhang et al. [58]	Proposed Protocol
Resists stolen smart card attack	No	No	No	No	No	Yes	Yes
Resists sensor node capture attack	Yes	Yes	No	No	Yes	No	Yes
Resists energy exhausting attack	No	No	No	No	No	No	Yes
Resists user impersonation attack	No	No	No	Yes	Yes	Yes	Yes
Resists sensor impersonation attack	No	No	No	Yes	No	No	Yes
Resists insider attack	Yes	Yes	Yes	Yes	No	Yes	Yes
Offers mutual authentication	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Offers secure password updating	No	No	No	Yes	Yes	Yes	Yes
Offers formal security analysis	Yes	Yes	Yes	Yes	Yes	Yes	Yes

#### 3.6.4 Formal Security verification of the proposed protocol using AVISPA Tool

In this section, we first explain the setup procedure and some basic features of AVISPA tool which we use for the formal security analysis of our proposed protocol. Afterwards, we describe the implementation of our protocol using High- Level Protocol Specification Language (HLPSL). Finally, we discuss about the results obtained.

### 3.6.4.1 Experimental Setup and the Size of the Entities Involved in WSNs/IoT for the Simulation of Proposed Protocol Using AVISPA Tool

In order to simulate the proposed protocol on AVISPA v1.1, we use a Security Protocol ANimator (SPAN) Version 1.6 on a computer system having ubuntu 16.04 LTS operating system (64 bit), Intel (R) core (TM) i7-6500U CPU @ 2.50 GHz x4 processor, and 8 GB RAM. We extract the archive avispa-package-1.1\_Linux-i686.tgz, set up the environment variable AVISPA\_PACKAGE and keep the script of the avispa protocol in the execution path. We implement our protocol considering minimal number of entities involved in WSNs/IoT (i.e, one user  $\mathcal{U}_i$ , one sensor node  $SN_j$  and one gateway node  $GW_N$ ) using Dolev-Yao model [110] with a bounded number of sessions, specified goal, On-the-Fly Model-Checker(OFMC) and Constraint-Logic based Attack Searcher (CL-AtSe) backend.

AVISPA involves HLPSSL to specify the protocol in a file with.hlpssl extension. It performs a static analysis to verify the executability of the protocol. A HLPSSL2IF translator is used to translate the HLPSSL specification into an Intermediate Formate (IF) specification, which is tool-independent language and compatible for automated deduction. The IF specifications are provided as an input to one of the four back-ends. The back-ends are as follows:

1. On-the-fly model-checker (OFMC)
2. Constraint-logic based attack searcher (CL-AtSe)
3. SAT-based model-checker (SATMC)
4. Tree automata based on automatic approximation for the analysis of security protocols (TA4SP).

### 3.6.5 Implementation of the Proposed Protocol Using HLPSSL

The HLPSSL specification of the protocol consist of some important section as follow:

1. **Basic Role:** Basic role explains the activity of the entities (e.g., User  $U_i$ , Gateway  $GW_N$  and Sensor node  $SN_j$ ) involve in the protocol.

### 3.6 Security Analysis and Comparison

---

- Each role may have some parameter like  $\mathcal{U}_i$ ,  $GW N$ ,  $SN_j$  of type agent and  $K_{ui1}$ ,  $K_{gsnj}$  of type symmetric\_key.
- The parameter RCV and SND denotes the agent's communication channels for receiving and sending the information.
- The parameter (dy) represents the Dolev-Yao intruder model for the channel.
- The function H, EccMul and XOR corresponding to the hash function, elliptic curve scalar multiplication and logical XOR operations respectively.
- The term hash\_func represents all the functions which are not easily invertible because the random non-invertible arithmetic operators are not supportable in HLP SL.
- The term "played\_by  $U_i$ " denotes that the role User is played by  $U_i$ .

The HLP SL specification of roles of  $\mathcal{U}_i$ ,  $GW N$  and  $SN_j$  are shown in Tables 3.4, 3.5 and 3.6 respectively.

2. **Transitions:** The transitions are declared in steps. It consist of trigger which fires when an event occurs. For any States in a transition if a message received on channel RCV, then transition fires and allocates a new value to the State.
3. **Composed Roles:** It makes one or more basic roles to execute together and represent the sessions involve in the protocol. The operator  $\wedge$  represents the parallel execution of the roles.

The HLP SL specification of proposed protocol's session is shown in Table 3.7.

4. **Environment:** It consist of global constant and session composition, where the adversary may execute some role as a authorized user.

The HLP SL specification of proposed protocol's environment is shown in Table 3.8.

5. **Security Goal:** The HLP SL specification of proposed protocol's GOAL is shown in Table 3.9. This module specifies the security Goal of the protocol. Some important predicates used in this module are as follows:



- $\text{secret}(\{PW_{ui}, RN_{ui}\}, \text{sec1}, U_i)$ : It represents that the information  $\{PW_{ui}, RN_{ui}\}$  is secretly shared to  $U_i$  and it can be recognize with a constant identity  $\text{sec1}$  in goal section.
- $\text{witness}(U_i, GWN, \text{gateway\_user\_gu}, \text{Tui}, \text{Alpha}')$ : It represents the weak authenticity of  $U_i$  by  $GWN$  and  $U_i$  is the witness for the data  $\{\text{Tui}', \text{Alpha}'\}$ . The identity of this goal is represented as  $\text{gateway\_user\_gu}$  in goal section.
- $\text{request}(U_i, SN_j, \text{user\_sensor\_us}, \text{Skey}')$ : It represents the strong authenticity of  $U_i$  by  $SN_j$  on  $\text{Skey}$  with an identity  $\text{user\_sensor\_us}$ .
- Symbols: Concatenation (.) is used for message composition (e.g.,  $\text{SND}(\text{IDi.PBi}')$ ) and Commas (,) is used in case of multiple arguments of events or functions (e.g.,  $\text{secret}(PW_{ui}, RN_{ui}, \text{sec1}, U_i)$ ).

#### 3.6.5.1 Analysis of the result obtained using AVISPA tool

We have selected the OFMC back-end for the execution test and model checking based on bounded number of sessions. For the replay attack checking, the back-end checks whether the legitimate agents can execute the specified protocol by performing a search of a passive intruder. After that the back-end gives the intruder the knowledge of some normal sessions between the legitimate agents. For the Dolev-Yao model check, the back-end checks whether there is any man-in-the-middle attack possible by the intruder. It is assumed that the intruder has knowledge of all public parameters. We have then simulated our protocol using AVISPA for OFMC and CL-AtSe model checkers. The results of the analysis using OFMC and CL-AtSe of our proposed protocol are shown in Table 3.10, and both simulation results ensure that our protocol is secure against active attacks including replay and man-in-the-middle attacks.

## 3.7 Performance Comparison

We compare our protocol with Shi et al.'s [109] and Choi et al.'s [1] protocol on the basis of computational overhead of performing hashing, random number generation and elliptic curve scalar multiplication on smart card, sensor node and gateway node. Table 3.11 shows that the computational overhead of our protocol is more than Shi et

**Table 3.4:** Specification of  $\mathcal{U}_i$ 's role in HLPSL.

```

role user(Ui, GWN, SNj: agent,
Kgsj, Xui1: symmetric_key,
H, XOR, EccMul: hash_func,
RCV, SND: channel(dy))
played_by Ui def=
local
State: nat,
IDui, PWui, IDsnj, Rui, P, Kui1, Aui, Bui, Wui, Alpha, Beta, Gamma, Delta, Ysnj,
Ysnj1, Xui, Tui, Tgwn, X, Beta1, Kui, Rsnj, Gamma1, Skey, Skey1: text
const sec1, sec2, sec3, sec4, sec5, sec6, sec7, sec8,
gateway_sensor_gs, gateway_user_gu, user_sensor_us: protocol_id
init
State: = 0
transition
0. State = 0  $\wedge$  RCV (start) =  $\triangleright$ 
State': = 2  $\wedge$  PWui': = H(PWui.Rui')
 $\wedge$  secret(PWui, RNui, sec1, Ui)
 $\wedge$  SND (IDui.PWui')
2. State = 2  $\wedge$  RCV (P.Aui'.Bui'.Wui') =  $\triangleright$ 
State': = 3  $\wedge$  Rui': = new()
 $\wedge$  Tui': = new()
 $\wedge$  secret(Rui', sec2, Ui)
 $\wedge$  PWui1': = H(PWui.RNui1')
 $\wedge$  Kui1': = XOR(Wui, H(IDui.RNui1'))
 $\wedge$  Xui': = EccMul(Rui'.P)
 $\wedge$  Xui1': = EccMul(Rui'.Kui1')
 $\wedge$  secret(Xui1', sec3, Ui, GWN)
 $\wedge$  Alpha': = H(IDui.IDsnj.Xui.Xui'.Tui)
 $\wedge$  SND(IDui.Xui.D.Tui'.Alpha')
 $\wedge$  witness(Ui, GWN, gateway_user_gu, Tui, Alpha')
6. State = 6  $\wedge$  RCV(Beta1') =  $\triangleright$ 
State': = 7  $\wedge$  Delta1': = H(Xui'.Tui.IDsnj.Ysnj.Tsnj)
 $\wedge$  Skey': = EccMul(Rui'.Ysnj1')
 $\wedge$  request(Ui, SNj, user_sensor_us, Skey')
end role

```

### 3.7 Performance Comparison

**Table 3.5:** Specification of *GWN*'s role in HLPSP.

```

role gateway(Ui, GWN, SNj: agent,
Xui1, Kgsj: symmetric_key,
H, EccMul, XOR: hash_func,
SND, RCV: channel(dy))
played_by GWN def=
local
State: nat,
IDui, IDsnj, PWui, P, Kui1, Rui, Aui, Bui, Wui, Alpha, Beta, Gamma, Delta, Ysnj, Tui,
Tgwn, Xui, X, Beta1, Kui, Rsnj, Gamma1, Skey, Skey1: text
const sec1, sec2, sec3, sec4, sec5, sec6, sec7, sec8,
gateway_sensor_gs, gateway_user_gu, user_sensor_us: protocol_id
init
State: = 1
transition
1. State = 4 ∧ RCV (IDui. Xui.Tui'.Alpha.IDsnj.Ysnj.Tsnj.Beta)= ▷
State': = 5 ∧ X': = new() ∧ Kui': = EccMul(H(IDui.X').P)
    ∧ Aui': = XOR(PWui'.H(XOR(IDui.Y'))))
    ∧ Bui': = H(IDui.PWui'.XOR(IDui.Y')) ∧ secret(X',sec4, GWN)
    ∧ Wui': = XOR(H(IDui.PWui).Kui')
    ∧ secret(Kui', sec5, GWN,Ui)
    ∧ SND(P.Aui'.Bui'.Wui')
3. State = 3 ∧ RCV (IDui. Xui.Tui'.Alpha.IDsnj.Ysnj.Tsnj.Beta)= ▷
State': = 4 ∧ Tgwn': =new()
    ∧ request(GWN, Ui, gateway_user_gu, Alpha')
    ∧ Fui': = XOR(P'.T') ∧ Xui': = EccMul(H(IDui.X').P)
    ∧ Alpha': = H(IDui.IDsnj.Xui.Xui'.Tui))
    ∧ secret(X',sec4, GWN)
    ∧ Beta': = H(Kgsj.Tui.Ysnj.Tsnj)
    ∧ secret(Kui', sec5, GWN,Ui)
    ∧ secret(Kgsj, sec6, GWN,SNj)
    ∧ Gamma': = H(Kgsj. Alpha.Ysnj.Tsnj.Tgwn)
    ∧ Delta': = H(Xui'.Tui.IDsnj.Ysnj.Tsnj)
    ∧ SND(Tgwn'.Gamma'.Delta')
    ∧ witness(GWN, Ui, gateway_user_gu, Tgwn')
end role

```

### 3.7 Performance Comparison

**Table 3.6:** Specification of  $SN_j$ 's role in HLPSSL.

```

role sensor(Ui, GWN, SNj: agent,
Xui1, Kgsnj: symmetric_key,
H, Gen, Rep, EccMul, Enc, Dec, XOR: hash_func,
SND, RCV: channel(dy))
played_by SNj def=
local
State: nat,
IDui, IDsnj, PWui, PWui, PWui1, P, Kui1, Rui, Aui, Bui, Wui, Alpha, Beta, Gamma, Delta, Ysnj, Tui, Tgwn, Xui, X, Beta1, Tau,
Kui, Rsnj, Gamma1, Kij, SKij: text
const sec1, sec2, sec3, sec4, sec5, sec6, sec7, sec8,
gateway_sensor_gs, gateway_user_gu, user_sensor_us: protocol_id
init
State: = 2
transition
3. State = 3  $\wedge$  RCV (IDui'.Xui'.Tui'.Alpha'.IDsnj'.Ysnj'.Tsnj'.Beta') =  $\triangleright$ 
State': = 4  $\wedge$  Tsnj': =new()
     $\wedge$  Ysnj': = EccMul(Rsnj.P)
     $\wedge$  Beta': = H(Kgsj.Tui.Ysnj.Tsnj)
     $\wedge$  SND(IDui'.Xui'.Tui'.Alpha'.IDsnj'.Ysnj'.Tsnj'.Beta1')
5. State = 5  $\wedge$  RCV (Tgwn'.Gamma'.Delta') =  $\triangleright$ 
State': = 6  $\wedge$  Tsnj1': =new()
     $\wedge$  Gamma': = H(Kgsj'.Alpha'.Ysnj'.Tsnj'.Tgwn')
     $\wedge$  Kij': = EccMul(Rui.Ysnj)
     $\wedge$  secret(Kij', sec7, SNj)
     $\wedge$  request(SNj, Ui, gateway_user_gu, SKij')
     $\wedge$  witness(SNj, Ui, gateway_user_gu, Tsnj')
     $\wedge$  Tau': = H(Tsnj1'.Delta'.Kij')
     $\wedge$  SKij': = H(Kij')
     $\wedge$  secret(SKij', sec8, SNj)
     $\wedge$  SND(Ysnj'.Tsnj'.Tsnj1'.Delta'.Tau')
end role

```

**Table 3.7:** Specification of proposed protocol's session in HLPSSL.

```

role session(Ui, GWN, SNj: agent,
Xui1, Kgsj: symmetric_key,
H, EccMul, XOR: hash_func)
def=
local GWNUi, Rui, GWNSNJ, RSNj, GWNGWN, RGWN: channel(dy)
composition
    user(Ui, GWN, SNj, Xui1, Kgsnj, H, EccMul, XOR, GWNUi, Rui)
     $\wedge$  sensor(Ui, GWN, SNj, Xui1, Kgsnj, H, EccMul, XOR, GWNSNJ, RSNj)
     $\wedge$  gateway(Ui, GWN, SNj, Xui1, Kgsnj, H, EccMul, XOR, GWNGWN, RGWN)
end role

```

### 3.7 Performance Comparison

**Table 3.8:** Specification of proposed protocol's environment in HLPSL.

```

role environment()
def=
const ui, gwn, snj: agent,
xui1,kgsj,kig: symmetric_key,
h, eccMul, xOR: hash_func,
sec1, sec2, sec3, sec4, sec5, sec6, sec7, sec8,
gateway_sensor_gs, gateway_user_gu, user_sensor_us: protocol_id
intruder_knowledge = ui,gwn,snj,kig
composition
session(ui,snj,gwn,xui1,kig,h, eccMul, xOR)
    ∧ session(ui,snj,gwn,kgsj,kig,h, eccMul, xOR)
    ∧ session(ui,snj,gwn,kig,kgsj,h, eccMul, xOR)
end role

```

**Table 3.9:** Specification of proposed protocol's goal in HLPSL.

```

goal
secrecy_of sec1, sec2, sec3, sec4, sec5, sec6, sec7, sec8
authentication_on gateway_sensor_gs, gateway_user_gu, user_sensor_us
end goal
environment()

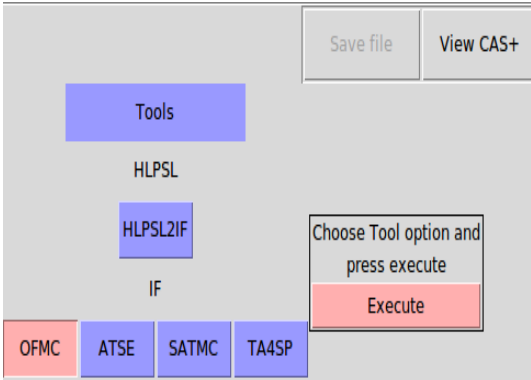
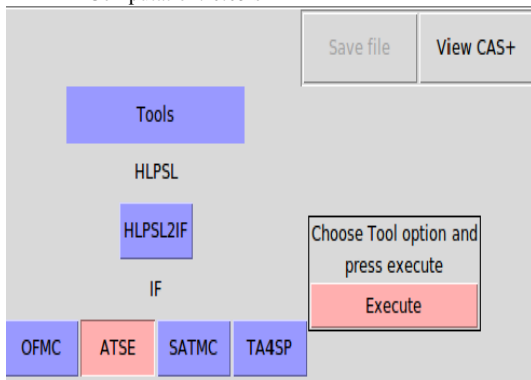
```

al.'s protocol and less than Choi et al.'s protocol but through the security analysis, as described in formal and informal security analysis, we find that our protocol is more secure than Choi et al.'s protocol. Our proposed protocol corrects the flaws of Choi et al.'s protocol, such as smart card loss attack, energy exhausting attack and sensor node capture attack. Also, even though our protocol requires a little more communication cost than some of the other protocols, we consider this acceptable because our proposed protocol assures security and provides additional functionality, as Table 3.3 shows.

Where,  $T_M$  and  $T_H$  are cost of performing elliptic curve scalar multiplication and hashing respectively.  $T_M = 50.3$  millisecond and  $T_H = 0.5$  millisecond based on the execution time as considered in Maurya et al.'s [33] protocol for the different cryptographic operation (performed by user  $U_i$  and the gateway node  $GW_N$  with a computer system having windows 7 operating system, Intel (R) core (TM) 2 Quad CPU Q8300, @2.50 Hz processor, and 2 GB RAM).  $TS_M = 370$  millisecond and  $TS_H = 3.63$  millisecond based on The computational time (as considered in Maurya et al.'s [33] protocol) of various cryptographic operations (performed by MicaZ sensor node  $SN_j$  with 8-bit ATmega128L Atmel processor, 4 K bytes ROM, 128 K bytes ROM, 512

### 3.7 Performance Comparison

**Table 3.10:** Security verification result of proposed protocol -1 obtained using AVISPA tool.

Using OFMC BACKEND	Using CL-AtSe BACKEND
<p>SUMMARY SAFE</p> <p>DETAILS BOUNDED_NUMBER_OF_SESSIONS</p> <p>PROTOCOL /home/cmb-lab-22/Desktop/Proto1.if</p> <p>GOAL as_specified</p> <p>BACKEND OFMC</p> <p>STATISTICS Time: 886 ms parseTime: 0 ms visitedNodes: 524 nodes depth: 8 piles</p> 	<p>SUMMARY SAFE</p> <p>DETAILS BOUNDED_NUMBER_OF_SESSIONS TYPED_MODEL</p> <p>PROTOCOL /home/cmb-lab-22/Desktop/Proto1.if</p> <p>GOAL as_specified</p> <p>BACKEND CL-AtSe</p> <p>STATISTICS Analysed: 1564 states Reachable: 1564 states Translation: 0.08 s Computation: 0.03 s</p> 

K bytes EEPROM, 2 AA battery with TinyOS [111] and nesC [112] programming language)

**Table 3.11:** Computational Overhead Comparisons

Scheme	Computational Overhead on $\mathcal{U}_i, \mathcal{SN}_j, \mathcal{GWN}$		
	$\mathcal{U}_i$	$\mathcal{SN}_j$	$\mathcal{GWN}$
Xue et al.'s [79]	$12T_H$	$9TS_H$	$17T_H$
Sun et al.'s [108]	$6T_H$	$8TS_H$	$7T_H$
Shi et al.'s [109]	$3T_M + 5T_H$	$2TS_M + 3TS_H$	$T_M + 4T_H$
Choi et al.'s [1]	$3T_M + 7T_H$	$2TS_M + 4TS_H$	$T_M + 4T_H$
Li et al.'s [98]	$2T_M + 8T_H$	$4TS_H$	$T_M + 9T_H$
Zhang et al.'s [58]	$4T_M + 4T_H$	$2TS_M + 3TS_H$	$4T_M + 5T_H$
Proposed Protocol	$3T_M + 6T_H$	$2TS_M + 3TS_H$	$T_M + 4T_H$

## 3.8 Summary

Because of the ongoing development in WSN & IoT advancements, we have watched a colossal change in perspective in sensor network applications. The confirmation and security objectives of a sensor network have turned out to be more pivotal and testing. The vast majority of the user and sensor node authentication plans for WSNs have been created without considering the prerequisites of coordinating WSNs with developing advances, for example, IoT.

In this chapter, we have analysed the security features provided by Choi et al.'s user authentication protocol. We have discovered that the existing protocols have some security shortcoming, for example, (1) unreliable against stolen smart card attack, (2) no resilient against node capture attack (3) defenceless against sensor node energy exhausting attacks. At that point, we have proposed an enhanced user authentication protocol based on smart card and elliptic curve cryptography to accomplish attractive security features with viable computational cost. Our convention withstands the security traps of Choi et al. 's. Protocol and enhanced the computational execution too. To demonstrate the ease of use of the proposed protocol, we made complete security and performance examination and reenacted the proposed thought in a computerised convention verifier tool, AVISPA.

## Chapter 4

# Three Factor User Authentication and Key Establishment protocol for WSNs & IoT

“Cryptography has generated number theory, algebraic geometry over finite fields, algebra, combinatorics and computers.” Vladimir Arnold

This chapter presents an efficient three-factor user authentication and key establishment protocol for WSNs & IoT based on Fuzzy Extractor and Elliptic-Curve Diffie Hellman (ECDH) algorithm. It concisely illustrates the problem, security challenges, security analysis of various existing user authentication protocols and contributions made thereof. Subsequently, it elaborates the proposed protocol, and then, the security and computational overhead are analyzed.

### 4.1 Introduction and Problem Definition

It is worth mentioning that the two-factor user authentication protocol proposed in Chapter 3 uses ECC technique, and the proposed two-factor authentication techniques on the basis of user's identity  $ID_{u_i}$  and password  $PW_{u_i}$  is predominantly weak because the user desires to keep a low-entropy password that can be efficiently decoded by applying simple dictionary attacks. To enhance the security characteristics of two-factor user authentication mechanism that is resistless to password guessing attacks and



subject to inadequate password update method in WSNs & IoT, biometric-based user authentication procedure, accompanied with user's smart card and passwords, have drawn significant attention.

Three-factor authentication is mostly applied in businesses, medical and government related implementations of WSNs & IoT that involves high degrees of security.

In recent times, various user authentication techniques for WSNs & IoT have been introduced in the literature and it has been recognized that most of the protocols cannot attain comprehensive security specifications. To enhance the quality of service and reduce the probability of security attacks, a reliable and valid user authentication system is required for WSNs & IoT. Session key establishment among  $SN_j$  and  $\mathcal{U}_i$  is too crucial for secure data communication.

In this chapter, we perform the security analysis of A.K.Das's protocol (specified in 2015), Choi et al.'s protocol (specified in 2016), including Park et al.'s protocol (specified in 2016). The security analysis reveals that their user authentication protocols are resistless to various security attacks like sensor node impersonation, user impersonation and attacks by legitimate users.

Moving from the cryptanalysis of these existing protocols, we propose a secure and efficient authenticated session key establishment protocol which facilitates multiple security specialities and overwhelms the drawbacks of existing user authentication protocols. Our precise and simple security analysis demonstrates that the proposed protocol resists the numerous security vulnerabilities associated with WSNs & IoT.

The automated validation applying AVISPA and Scyther tool guarantees the inadequacy of security attacks in our protocol. The logical verification utilising the Burrows-Abadi-Needham (BAN) logic validates the accuracy of our proposed protocol. Subsequently, the comparative analysis based on computational burden and security characteristics of different existing protocol indicates that the proposed user authentication method is safe and effective. In the future, we aim to implement the proposed protocol in the real-world utilization of WSNs & IoT.

**Problem Definition:** The problem definition of this chapter is as follows:

Design and analysis of secure and efficient multi-factor user authentication and session key establishment protocol for resource constraints WSNs& IoT which provides the major security features (such as mutual authentication, secure session key establishment, confidentiality, integrity, freshness etc. ) and prevents the major security attacks

(such as stolen smart card, user impersonation, sensor node impersonation, etc.) with lesser computational overhead.

## 4.2 Our Contributions

The contributions of this chapter are as follows:

- In this chapter, we first address several security concerns associated with verifying the users of WSNs & IoT.
- We present the security analysis of numerous modern protocols of user authentication for WSNs & IoT. By security analysis, we expose that the current protocols are defenceless to several attacks like sensor node impersonation, user impersonation and attacks by legitimate users.
- We propose a protected and efficient protocol for verifying the users of WSNs and IoT considering about shared confirmation, session key foundation, information freshness, and confidentiality.
- Through casual security examination, we demonstrate that our proposed protocol opposes the sensor node compromise, stolen smart card, gateway node compromise, man-in-the-middle and replay attacks.
- We prove using random oracle model the correctness of various security features involved in our proposed protocol.
- Subsequently, we verify the proposed protocol on popular and robust security verification tool such as AVISPA and Scyther.
- We use BAN logic to determine whether exchanged messages of the proposed protocol are trustworthy and secure against eavesdropping.
- Finally, we present the comparative analysis of our proposed protocol with other existing protocols on the basis of security and computational overhead.

## 4.3 Review and Cryptanalysis of Various Recent Protocols of User Authentication for WSNs

In this section, we concisely review and present the security analysis of the various recently proposed user authentication protocols of WSNs. The security analysis performed in this section illustrates that the existing protocols have various security vulnerability based on the logical proofs and the assumptions considered in the Section 1.2.2 of this thesis. This section provides an awareness of what needs to be fixed and how the user authentication protocol should be design to withstand the miscellaneous attacks incorporated into the WSNs & IoT.

### 4.3.1 Review of A.K.Das's Protocol

A.K.Das [81] did the security study of Althobaiti et al.'s [59] protocol and recommended an enhanced protocol concerning user authentication utilising the fuzzy extractor to resist user impersonation attack, node capture attack, man-in-the-middle attack. A.K.Das [81] suggested a novel procedure (analysing the resource limitations of the sensor node) for bio-metric based user authentication applying the fuzzy extractor.

Toward evaluating the security characteristics of A.K.Das's protocol, the user registration phase of A.K. Das's protocol is illustrated in the subsequent Step DR1, Step DR2, Step DR3 and the authentication-key agreement state is reviewed in the Steps DA1, Step DA2, Step DA3 on the basis of the notations of Table 1. We interpret the user registration, authentication and key agreement phase of A.K.Das's protocol in Tables 4.1 and 4.2 respectively.

**Step DR1:** The user  $\mathcal{U}_i$  inputs  $ID_{\mathcal{U}_i}$ ,  $PW_{\mathcal{U}_i}$  and  $BIO_{\mathcal{U}_i}$  and generates 1024 bit random number  $K$ . Subsequently,  $\mathcal{U}_i$  calculates  $RPW_i = h(ID_{\mathcal{U}_i} || K || PW_{\mathcal{U}_i})$  and selects a key  $ek_i$ . Then,  $\mathcal{U}_i$  transmits  $\langle ID_{\mathcal{U}_i}, RPW_i, ek_i \rangle$  to  $GWN$  using secure communication channel.

**Step DR2:** After receiving the message  $\langle ID_{\mathcal{U}_i}, RPW_i, ek_i \rangle$ , the gateway node  $GWN$  generates 1024 bit key  $X_s$ , evaluates  $f_i = h(ID_{\mathcal{U}_i} \oplus h(X_s))$  and stores  $(h(\cdot), Gen(\cdot), Rep(\cdot), f_i, \mathcal{T})$  into  $SC_{\mathcal{U}_i}$ . Then,  $GWN$  sends  $\langle SC_{\mathcal{U}_i} \rangle$  to  $\mathcal{U}_i$  using secure communication channel.

### 4.3 Review and Cryptanalysis of Various Recent Protocols of User Authentication for WSNs

---

**Table 4.1:** User registration phase of A.K.Das's protocol.

Step 1: For User ( $\mathcal{U}_i$ )	Step 2: For Gateway ( $GWN$ )
<p>The user <math>\mathcal{U}_i</math> inputs <math>ID_{\mathcal{U}_i}</math>, <math>PW_{\mathcal{U}_i}</math> and <math>BIO_{\mathcal{U}_i}</math>.  Generates 1024 bit random number <math>K</math>.  Subsequently, <math>\mathcal{U}_i</math> calculates <math>RPW_i = h(ID_{\mathcal{U}_i}    K    PW_{\mathcal{U}_i})</math> and selects a key <math>ek_i</math>.</p> <p>Then, sends <math>\langle ID_{\mathcal{U}_i}, RPW_i, ek_i \rangle</math> to <math>GWN</math>  <span style="display: block; text-align: center;"><math>\xrightarrow{\text{ViaSecureChannel}}</math></span></p>	<p>After receiving the message <math>\langle ID_{\mathcal{U}_i}, RPW_i, ek_i \rangle</math>, the gateway node <math>GWN</math> generates 1024 bit key <math>X_s</math>, evaluates <math>f_i = h(ID_{\mathcal{U}_i} \oplus h(X_s))</math>, and stores <math>(h(), Gen(), Rep(), f_i, \mathcal{T})</math> into <math>SC_{\mathcal{U}_i}</math>.</p> <p>Finally, <math>GWN</math> sends <math>\langle SC_{\mathcal{U}_i} \rangle</math> to <math>\mathcal{U}_i</math>  <span style="display: block; text-align: center;"><math>\xleftarrow{\text{ViaSecureChannel}}</math></span></p> <p>Then, <math>GWN</math> stores <math>ek_i</math> related to <math>ID_{\mathcal{U}_i}</math></p>
Step 3: For User ( $\mathcal{U}_i$ )	
<p><math>\mathcal{U}_i</math> evaluates <math>Gen(BIO_{\mathcal{U}_i}) = (\sigma_i, \tau_i)</math>,  <math>f_i^* = f_i \oplus h(ID_{\mathcal{U}_i}    \sigma_i    K)</math>,  <math>r_i = h(ID_{\mathcal{U}_i}    \sigma_i) \oplus K</math>,  <math>e_i = h(ID_{\mathcal{U}_i}    RPW_i    \sigma_i)</math>, and  <math>BE_i = h(ID_{\mathcal{U}_i}    \sigma_i) \oplus ek_i</math>.  Then, <math>\mathcal{U}_i</math> replaces <math>f_i</math> with <math>f_i^*</math> in <math>SC_{\mathcal{U}_i}</math>.  Finally, <math>\mathcal{U}_i</math> stores <math>e_i, \tau_i, BE_i, r_i</math> into <math>SC_{\mathcal{U}_i}</math>.</p>	

### 4.3 Review and Cryptanalysis of Various Recent Protocols of User Authentication for WSNs

---

**Step DR3:** After receiving  $SC_{\mathcal{U}_i}$ , the user  $\mathcal{U}_i$  evaluates  $Gen(BIO_{\mathcal{U}_i}) = (\sigma_i, \tau_i)$ ,  $f_i^* = f_i \oplus h(ID_{\mathcal{U}_i} || \sigma_i || K)$ ,  $r_i = h(ID_{\mathcal{U}_i} || \sigma_i) \oplus K$ ,  $e_i = h(ID_{\mathcal{U}_i} || RPW_i || \sigma_i)$ , and  $BE_i = h(ID_{\mathcal{U}_i} || \sigma_i) \oplus ek_i$ . Finally,  $\mathcal{U}_i$  replaces  $f_i$  with  $f_i^*$  in  $SC_{\mathcal{U}_i}$  and stores  $e_i, \tau_i, BE_i, r_i$  into  $SC_{\mathcal{U}_i}$ .

**Step DA1:** The registered user  $\mathcal{U}_i$  inserts his/her smart card  $SC_{\mathcal{U}_i}$  into the card reader device and provides the  $ID_{\mathcal{U}_i}$ , secret  $PW_{\mathcal{U}_i}$ , bio-metric information  $BIO_{\mathcal{U}_i}$ . Then, evaluates  $\sigma'_i = Rep(BIO_{\mathcal{U}_i}, \tau_i)$ ,  $K' = r_i \oplus h(ID_{\mathcal{U}_i} || \sigma'_i)$ ,  $RPW'_i = h(ID_{\mathcal{U}_i} || PW_{\mathcal{U}_i} || K')$ ,  $e'_i = h(ID_{\mathcal{U}_i} || RPW'_i || \sigma'_i)$ . If  $e'_i = e_i$ ,  $\mathcal{U}_i$  transmits  $\langle ID_{\mathcal{U}_i}, req \rangle$  to  $GWN$  via public communication channel. Otherwise,  $\mathcal{U}_i$  aborts this phase.

**Step DA2:** After receiving the message  $\langle ID_{\mathcal{U}_i}, req \rangle$ ,  $GWN$  verifies the message. If  $ID_{\mathcal{U}_i}$  is valid,  $GWN$  sends a Random challenge  $R$  to  $\mathcal{U}_i$  via public communication channel. Otherwise,  $GWN$  aborts this phase.

**Step DA3:** After receiving the Random challenge  $R$ ,  $\mathcal{U}_i$  evaluates  $ek_i = BE_i \oplus h(ID_{\mathcal{U}_i} || \sigma'_i)$ . Finds the current time-stamp  $T_1$ . Then,  $\mathcal{U}_i$  transmits  $\langle Enc_{ek_i}(R, T_1, ID_{SN_j}) \rangle$  to  $GWN$  via public communication channel.

**Step DA4:**  $GWN$  evaluates  $R, T_1, ID_{SN_j}$  using decryption operation on the basis ofkey  $ek_i$ . If  $T_1$  is fresh and  $R$  is valid,  $GWN$  computes  $f_i^* = h(ID_{\mathcal{U}_i} \oplus h(X_s))$ ,  $f_i^{**} = h(ID_{SN_j} || f_i^*)$ , finds the current time-stamp  $T_2$  and computes  $Y_j = Enc_{K_j}[ID_{\mathcal{U}_i}, ID_{SN_j}, T_1, T_2, f_i^{**}]$ . Finally,  $GWN$  transmits  $\langle ID_{\mathcal{U}_i}, Y_j \rangle$  to  $SN_j$  via public communication channel. Otherwise,  $GWN$  aborts this phase immediately.

**Step DA5:**  $SN_j$  retrieves  $(ID_{\mathcal{U}_i}, ID_{SN_j}, T_1, T_2, f_i^{**})$  as  $(ID''_{\mathcal{U}_i}, ID''_{SN_j}, T''_1, T''_2, f''_i)$  using decryption operation on  $\langle ID_{\mathcal{U}_i}, Y_j \rangle$  on the basis ofkey  $K_j$ . If  $T''_2$  is fresh and  $ID_{\mathcal{U}_i}$  is valid,  $SN_j$  finds the current time-stamp  $T_3$  and evaluates the session key  $SK_{ij} = h(f''_i || ID_{\mathcal{U}_i} || ID_{SN_j} || T''_1, T_3)$ . Then,  $SN_j$  sends  $h(SK_{ij}), T_3$  to  $\mathcal{U}_i$  via public communication channel and stores  $SK_{ij}$  in its memory. Otherwise,  $SN_j$  aborts this phase immediately. Finally,  $SN_j$  stores  $SK_{ij}$  in its memory.

**Step DA6:** If  $T_3$  is fresh, the user  $\mathcal{U}_i$  computes  $f'_i = f_i^* \oplus h(\sigma'_i || ID_{\mathcal{U}_i} || K')$ ,  $f''_i = h(ID_{SN_j} || f'_i)$ ,  $SK'_{ij} = h(f''_i || ID_{\mathcal{U}_i} || ID_{SN_j} || T_1 || T_3)$ . If  $h(SK'_{ij}) = h(SK_{ij})$ ,  $\mathcal{U}_i$

establishes the session key  $SK'_{ij}$  with the sensor node  $SN_j$ . Otherwise,  $\mathcal{U}_i$  aborts this phase immediately.

### 4.3.2 Cryptanalysis of A.K.Das's Protocol

In this section, we perform the cryptanalysis of the A.K.Dass protocol and found that A.K.Dass protocol is also vulnerable. The vulnerabilities involve in A.K.Das's protocol are elaborated in the following subsection:

#### 4.3.2.1 Stolen Smart Card Attacks

The adversary  $\mathcal{A}$  ascertains the value of  $\{\tau_i, e_i, r_i, BE_i, f^*, h(\cdot), Gen(\cdot), Rep(\cdot), \mathcal{T}\}$  from stolen  $SC_{\mathcal{U}_i}$  by measuring the power consumption of smart card [14]. Then,  $\mathcal{A}$  computes:  $BE_i \oplus r_i = [h(ID_{\mathcal{U}_i} \parallel \sigma_i) \oplus K] \oplus [h(ID_{\mathcal{U}_i} \parallel \sigma_i) \oplus ek_i] = K \oplus ek_i$ .

Afterwards, the adversary  $\mathcal{A}$  find out the value of  $K$  and  $ek_i$  by implementing one of the following three mechanism:

1. Derives the value of  $K$  and  $ek_i$  using the frequency analysis of stream cipher  $BE_i, r_i$  and  $BE_i \oplus r_i$ .
2. Eavesdrops  $R$  and  $E_{ek_i}(R, T, ID_{SN_j})$  and implements the known plain text attack to find out the value of  $ek_i$ . Thereafter,  $\mathcal{A}$  find out the value of  $K = ek_i \oplus (K \oplus ek_i)$ .
3. Steals the bio-metric information  $BIO'_{\mathcal{U}_i}$  of  $\mathcal{U}_i$  (where  $d(BIO_{\mathcal{U}_i}, BIO'_{\mathcal{U}_i}) \leq \mathcal{T}$ ) and find out the value of  $\sigma_i = Rep(BIO'_{\mathcal{U}_i}, \tau_i)$ . Eavesdrops the value of  $ID_{\mathcal{U}_i}$  from public communication channel and then evaluates the value of  $ek_i = BE_i \oplus h(ID_{\mathcal{U}_i} \parallel \sigma_i)$ ,  $K = r_i \oplus h(ID_{\mathcal{U}_i} \parallel \sigma_i)$ . It is possible, because  $ek_i$  is not password  $PW_{\mathcal{U}_i}$  protected.

Subsequently,  $\mathcal{A}$  chooses its own identity  $ID_A$ , password  $PW_A$ , biometric information  $B_A$  and computes:

$$RPW_A = h(ID_A \parallel K \parallel PW_A), Gen(B_A) = (\sigma_A, \tau_A), e_A = h(ID_A \parallel RPW_A \parallel \sigma_A), r_A = h(ID_A \parallel \sigma_A) \oplus K \text{ and } BE_A = h(ID_A \parallel \sigma_A) \oplus ek_i.$$

Finally,  $\mathcal{A}$  replaces the information  $\{\tau_i, e_i, r_i, BE_i, f^*, h(\cdot), Gen(\cdot), Rep(\cdot), \mathcal{T}\}$  of  $SC_{\mathcal{U}_i}$  with  $\{\tau_A, e_A, r_A, BE_A, f^*, h(\cdot), Gen(\cdot), Rep(\cdot), \mathcal{T}\}$  respectively.

### 4.3 Review and Cryptanalysis of Various Recent Protocols of User Authentication for WSNs

**Table 4.2:** Login, authentication and key sharing phase of A.K.Das's protocol.

Step 1: For User ( $\mathcal{U}_i$ )	Step 2: For Gateway ( $GWN$ )
<p>The enrolled user <math>\mathcal{U}_i</math> injects the smart card <math>SC_{\mathcal{U}_i}</math> into card reader and gives the <math>ID_{\mathcal{U}_i}</math>, secret <math>PW_{\mathcal{U}_i}</math>, <math>BIO_{\mathcal{U}_i}</math>. Then, evaluates <math>\sigma'_i = Rep(BIO_{\mathcal{U}_i}, \tau_i)</math>, <math>K' = r_i \oplus h(ID_{\mathcal{U}_i}    \sigma'_i)</math>, <math>RPW'_i = h(ID_{\mathcal{U}_i}    PW_{\mathcal{U}_i}    K')</math>, <math>e'_i = h(ID_{\mathcal{U}_i}    RPW'_i    \sigma'_i)</math></p> <p><b>if</b> <math>e'_i = e_i</math> <b>then</b></p> <p style="padding-left: 40px;"><math>\mathcal{U}_i</math> transmits <math>\langle ID_{\mathcal{U}_i}, req \rangle</math> to <math>GWN</math></p> <p style="text-align: center;"><math>\xrightarrow{\text{ViaPublicChannel}}</math></p> <p><b>end</b></p> <p><b>else</b></p> <p style="padding-left: 40px;">The user <math>\mathcal{U}_i</math> terminates this phase</p> <p><b>end</b></p>	<p>After receiving the message <math>\langle ID_{\mathcal{U}_i}, req \rangle</math>, <b>if</b> <math>ID_{\mathcal{U}_i}</math> is valid <b>then</b></p> <p style="padding-left: 40px;"><math>\xleftarrow{\text{ViaPublicChannel}} GWN \text{ sends a Random challenge } R \text{ to } \mathcal{U}_i</math></p> <p><b>end</b></p> <p><b>else</b></p> <p style="padding-left: 40px;"><math>GWN</math> aborts this phase</p> <p><b>end</b></p>
Step 3: For User ( $\mathcal{U}_i$ )	Step 4: For Gateway ( $GWN$ )
<p>After receiving the Random challenge <math>R</math>, <math>\mathcal{U}_i</math> evaluates <math>ek_i = BE_i \oplus h(ID_{\mathcal{U}_i}    \sigma'_i)</math>. Then,</p> <p style="padding-left: 40px;"><math>\mathcal{U}_i</math> transmits <math>\langle Enc_{ek_i}(R, T_1, ID_{SN_j}) \rangle</math> to <math>GWN</math></p> <p style="text-align: center;"><math>\xrightarrow{\text{ViaPublicChannel}}</math></p>	<p><math>GWN</math> evaluates <math>R, T_1, ID_{SN_j}</math> using <math>ek_i</math>.</p> <p><b>if</b> <math>T_1</math> and <math>R</math> are valid <b>then</b></p> <p style="padding-left: 40px;"><math>GWN</math> computes</p> <p style="padding-left: 80px;"><math>f_i^* = h(ID_{\mathcal{U}_i} \oplus h(X_s)), f_i^{**} = h(ID_{SN_j}    f_i^*)</math> and</p> <p style="padding-left: 80px;"><math>Y_j = Enc_{K_j}[ID_{\mathcal{U}_i}, ID_{SN_j}, T_1, T_2, f_i^{**}]</math>.</p> <p style="padding-left: 80px;">Then, <math>GWN</math> transmits <math>\langle ID_{\mathcal{U}_i}, Y_j \rangle</math> to <math>SN_j</math></p> <p style="text-align: center;"><math>\xrightarrow{\text{ViaPublicChannel}}</math></p> <p><b>end</b></p> <p><b>else</b></p> <p style="padding-left: 40px;">Reject <math>\mathcal{U}_i</math></p> <p><b>end</b></p>
Step 5: For Sensor Node ( $SN_j$ )	Step 6: For User ( $\mathcal{U}_i$ )
<p><math>SN_j</math> Retrieves <math>(ID_{\mathcal{U}_i}, ID_{SN_j}, T_1, T_2, f_i^{**})</math> as <math>(ID_{\mathcal{U}_i}'', ID_{SN_j}'', T_1'', T_2'', f_i'')</math>.</p> <p><b>if</b> <math>T_2</math> and <math>ID_{\mathcal{U}_i}</math> are valid <b>then</b></p> <p style="padding-left: 40px;"><math>SN_j</math> Evaluate the session key</p> <p style="padding-left: 80px;"><math>SK_{ij} = h(f_i''    ID_{\mathcal{U}_i}    ID_{SN_j}    T_1'', T_3)</math></p> <p style="padding-left: 80px;"><math>SN_j</math> sends <math>h(SK_{ij}), T_3</math> to <math>\mathcal{U}_i</math></p> <p style="text-align: center;"><math>\xrightarrow{\text{ViaPublicChannel}}</math></p> <p><b>end</b></p> <p><b>else</b></p> <p style="padding-left: 40px;">Reject <math>\mathcal{U}_i</math></p> <p><b>end</b></p> <p>Store <math>SK_{ij}</math></p>	<p><b>if</b> <math>T_3</math> is valid <b>then</b></p> <p style="padding-left: 40px;">Computes <math>f'_i = f_i^* \oplus h(\sigma'_i    ID_{\mathcal{U}_i}    K')</math>,</p> <p style="padding-left: 80px;"><math>f_i'' = h(ID_{SN_j}    f'_i)</math>,</p> <p style="padding-left: 80px;"><math>SK'_{ij} = h(f_i''    ID_{\mathcal{U}_i}    ID_{SN_j}    T_1    T_3)</math></p> <p style="padding-left: 80px;"><b>if</b> <math>h(SK'_{ij}) = h(SK_{ij})</math> <b>then</b></p> <p style="padding-left: 120px;"><math>\mathcal{U}_i</math> Stores <math>SK'_{ij}</math></p> <p style="padding-left: 80px;"><b>end</b></p> <p style="padding-left: 40px;"><b>end</b></p> <p><b>else</b></p> <p style="padding-left: 40px;">Reject <math>\mathcal{U}_i</math></p> <p><b>end</b></p> <p><b>end</b></p> <p><b>else</b></p> <p style="padding-left: 40px;">Reject <math>\mathcal{U}_i</math></p> <p><b>end</b></p>

### 4.3 Review and Cryptanalysis of Various Recent Protocols of User Authentication for WSNs

---

The login phase of the adversary  $\mathcal{A}$  is as follows:

- $\mathcal{A}$  insert  $SC_{\mathcal{U}_i}$  and inputs  $ID_A, PW_A$  and imprints  $B_A$ .
- $\mathcal{A}$  computes  $\sigma'_A = Rep(B_A, \tau_A)$ ,  $K' = r_A \oplus h(ID_A \parallel \sigma'_A)$ ,  $RPW'_A = h(ID_A \parallel PW_A \parallel K')$  and  $e'_A = h(ID_A \parallel RPW'_A \parallel \sigma'_A)$ . Then, it verifies if  $e'_A = e_A$ . It would be true i.e., both the password and bio-metric validation would be correct.
- Afterwards,  $\mathcal{U}_i$  sends the login message  $\langle ID_A, req \rangle$  to  $GWN$  via a public channel. However, the adversary  $\mathcal{A}$  intercepts the message  $\langle ID_A, req \rangle$  and replaces  $\langle ID_A, req \rangle$  with  $\langle ID_{\mathcal{U}_i}, req \rangle$ .

Authentication and key agreement phase for the adversary  $\mathcal{A}$  is illustrated as follows:

- Since  $ID_{\mathcal{U}_i}$  is valid, therefore  $GWN$  generates a random challenge  $R$  and send it to  $\mathcal{A}$ .
- $\mathcal{A}$  select the login sensor node  $SN_j$  and sends  $\langle E_{ek_i}(R, T_1, ID_{SN_j}) \rangle$  to  $GWN$ .
- After receiving  $\langle E_{ek_i}(R, T_1, ID_{SN_j}) \rangle$ ,  $GWN$  decrypt it using  $ek_i$  and verifies the validity of  $T_1$  and  $R$ . Subsequently,  $GWN$  computes  $f_i^* = h(ID_{\mathcal{U}_i} \oplus h(X_s))$ ,  $f_i^{**} = h(ID_{SN_j} \parallel f_i^*)$ ,  $Y_j = E_{K_j}[ID_{\mathcal{U}_i}, ID_{SN_j}, T_1, T_2, f_i^{**}]$  and finally sends  $\langle ID_{\mathcal{U}_i}, Y_j \rangle$  to the sensor node  $SN_j$ .
- After receiving  $\langle ID_{\mathcal{U}_i}, Y_j \rangle$ ,  $SN_j$  computes  $SK_{ij} = h(f_i'' \parallel ID_{\mathcal{U}_i} \parallel ID_{SN_j} \parallel T_1'' \parallel T_3)$  and sends  $h(SK_{ij}), T_3$  to  $\mathcal{A}$ .
- Then,  $\mathcal{A}$  computes  $f_i' = f_i^* \oplus h(\sigma_i' \parallel ID_{\mathcal{U}_i} \parallel K')$  using  $ID_{\mathcal{U}_i}$ , stolen bio-metric and evaluated  $K$ . It is possible because  $f_i'$  has no password protection.
- Finally,  $\mathcal{A}$  computes  $f_i'' = h(ID_{SN_j} \parallel f_i')$  and the session key  $SK_{ij} = h(f_i'' \parallel ID_{\mathcal{U}_i} \parallel ID_{SN_j} \parallel T_1'' \parallel T_3)$  shared with  $SN_j$ .

#### 4.3.3 Review of Choi et al.'s protocol

Choi et al. [61] made the security study of Yoon and Kim's [60] protocol and introduced an enhanced protocol (estimating the resource restrictions of the sensor node of WSNs & IoT) of user authentication utilising the fuzzy extractor and biometric data.



### 4.3 Review and Cryptanalysis of Various Recent Protocols of User Authentication for WSNs

---

The Choi et al.'s protocol resolves the difficulties of user validation problem, biometric recognition error, the absence of anonymity, session key disclosure by the GWN, perfect forward secrecy, DoS attack, and a cancellation problem. In this protocol, the gateway node  $GWN$  finds master keys,  $x$  and  $y$ , and allots  $h(ID_{SN_j}||y)$  to the sensor node  $SN_j$ . The registration phase of this protocol is summarized in Step CR1, Step CR2 and Step CR3. The authentication and session key establishment phase is reviewed in Table 4.3 on the basis of the notations described in Table 1.

**Step CR1:** The user  $\mathcal{U}_i$  inputs his/her identity  $ID_{\mathcal{U}_i}$ , biometric information  $BIO_{\mathcal{U}_i}$  and computes:  $(\sigma_i, \tau_i) = Gen(BIO_{\mathcal{U}_i})$ ,  $A_i = h(\sigma_i)$ . Then,  $\mathcal{U}_i$  transmits  $\langle ID_{\mathcal{U}_i}, A_i \rangle$  to  $GWN$  via secure communication channel.

**Step CR2:** After receiving the message  $\langle ID_{\mathcal{U}_i}, A_i \rangle$ , the gateway node  $GWN$  generates 1024 bit secret key  $x$  and computes  $M_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i}||x) \oplus A_i$ ,  $N_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i} \oplus x) \oplus A_i$ ,  $V_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i}||A_i)$ . Then,  $GWN$  stores  $\langle ID_{\mathcal{U}_i}, M_{\mathcal{U}_i}, N_{\mathcal{U}_i}, V_{\mathcal{U}_i}, h(\cdot) \rangle$  into smart card  $SC_{\mathcal{U}_i}$ . Finally,  $GWN$  sends the smart card  $SC_{\mathcal{U}_i}$  to the user  $\mathcal{U}_i$ .

**Step CR3:** After receiving the smart card  $SC_{\mathcal{U}_i}$ , the user  $\mathcal{U}_i$  stores  $\tau_i$  into  $SC_{\mathcal{U}_i}$ .

#### 4.3.4 Cryptanalysis of Choi et al.'s protocol

In this module, we perform the cryptanalysis of the Choi et al.'s protocol and found that Choi et al.'s protocol is also vulnerable. The vulnerabilities involve in this protocol are elaborated in the following subsection:

##### 4.3.4.1 Attack on the basis of Authorized or Legal User

In their protocol, an authorized or legal user  $U_L$  could be an adversary  $U_A$ , because  $U_L$  can get  $h(x||y)$  and then it can obtain the secret credentials of the user  $\mathcal{U}_i$  as follows:

- $U_A$  inputs  $ID_{U_A}$ , provides the bio-metric information  $B'_A$ , computes  $\sigma'_A = Rep(B'_A, \tau_A)$ ,  $A'_A = h(\sigma'_A)$ ,  $V'_A = h(ID_A||A'_A)$  and finally validates  $V_A = V'_A$ ,
- If the validation succeeds,  $U_A$  generates random number  $r_A$ , and evaluates  $X_A = r_A \times P$ ,  $D_A = M_A \oplus A'_A$ ,  $h(x||y) = N_A \oplus A'_A$

### 4.3 Review and Cryptanalysis of Various Recent Protocols of User Authentication for WSNs

**Table 4.3:** Authentication and session key establishment phase of Choi et al. protocol.

Step 1: For User ( $\mathcal{U}_i$ )	Step 2: For Gateway ( $\mathcal{GWN}$ )
<p>The registered user <math>\mathcal{U}_i</math> inputs <math>ID_{\mathcal{U}_i}</math>, <math>BIO'_{\mathcal{U}_i}</math> and computes <math>\sigma'_i = Rep(BIO'_{\mathcal{U}_i}, \tau_i)</math>, <math>A'_i = h(\sigma'_i)</math>, <math>V'_i = h(ID_{\mathcal{U}_i}    A'_i)</math>,  <b>if</b> <math>V_i = V'_i</math> <b>then</b>              <math>\mathcal{U}_i</math> generates random number <math>r_i</math>              <math>X_i = r_i \times P</math>, <math>D_i = M_i \oplus A'_i</math>, <math>h(x  y) = N_i \oplus A'_i</math>              Finds out current time-stamp <math>T_i</math> and              computes <math>k_i = h(D_i    T_i)</math>, <math>C_i = E_{k_i}(ID_{\mathcal{U}_i}    X_i)</math>,              <math>AID_i = ID_{\mathcal{U}_i} \oplus h(h(x  y)    T_i)</math>,              <math>W_i = h(h(x  y)    AID_i    X_i    C_i    T_i)</math>              Then, <math>\mathcal{U}_i</math> constructs a message              <math>M_1 = \langle AID_i, X_i, C_i, T_i, W_i \rangle</math>              Finally, <math>\mathcal{U}_i</math> transmits <math>M_1</math> to <math>\mathcal{GWN}</math>              ————— <math>\xrightarrow{\text{ViaPublicChannel}}</math> —————  <b>end</b>  <b>else</b>                Abort this phase.  <b>end</b></p>	<p><b>if</b> <math>(T' - T_i) \leq \Delta T</math> <b>then</b>              <b>if</b> <math>W_i = h(h(x  y)    AID_i    X    C_i    T_i)</math> <b>then</b>                  <math>\mathcal{GWN}</math> computes                  <math>ID'_{\mathcal{U}_i} = AID_i \oplus h(h(x  y)    T_i)</math>,                  <math>D'_i = h(ID'_{\mathcal{U}_i}    x)</math>, <math>k'_i = h(D'_i    T_i)</math>,                  <math>ID''_{\mathcal{U}_i}    X'_i = D_{k'_i}(C_i)</math>                  <b>if</b> <math>ID'_{\mathcal{U}_i} = ID''_{\mathcal{U}_i}</math> <b>then</b>                      <math>\mathcal{GWN}</math> finds its current time-stamp <math>T_g</math> and                      computes <math>k_g = h(h(SID_j    y)    T_g)</math>,                      <math>C_g = Enc_{k_g}(AID_i    X'_i)</math>,                      <math>W_g = h(h(SID_j    y)    AID_i    C_g    T_g)</math>.                      Then, <math>\mathcal{GWN}</math> construct the message                      <math>M_2 = \langle AID_i, C_g, T_g, W_g \rangle</math>. Finally,                      <math>\mathcal{GWN}</math> transmits <math>M_2</math> to <math>\mathcal{SN}_j</math>                      ————— <math>\xrightarrow{\text{ViaPublicChannel}}</math> —————                  <b>end</b>              <b>else</b>                    <math>\mathcal{GWN}</math> aborts this phase.              <b>end</b>  <b>end</b>  <b>else</b>                <math>\mathcal{GWN}</math> abort this phase.  <b>end</b></p>
Step 3: For Sensor Node ( $\mathcal{SN}_j$ )	Step 4: For User ( $\mathcal{U}_i$ )
<p><b>if</b> <math>(T'' - T_g) \leq \Delta T</math> <b>then</b>              <b>if</b> <math>W_g = h(h(SID_j    y)    AID_i    C_g    T_g)</math> <b>then</b>                  <math>\mathcal{SN}_j</math> computes <math>k'_g = h(h(SID_j    y)    T_g)</math>,                  <math>AID'_i    X' = Dec_{k'_g}(C_g)</math>,                  <b>if</b> <math>AID_i = AID'_i</math> <b>then</b>                      Generates random number <math>r_s</math>                      <math>K_{SU} = r_s \times X'_i</math>, <math>Y_i = r_s \times P</math> Computes                      <math>sk = h(AID_i    K_{SU}    T_s)</math> Find the current                      time-stamp <math>T_s</math> and computes <math>RM = \text{Query}</math>                      response, <math>V_s = h(AID'_i    X'_i    Y_i    RM    T_s)</math>,                      <math>M_3 = \langle RM, Y_i, V_s, T_s \rangle</math>. Finally,                      <math>\mathcal{SN}_j</math> transmits <math>M_3</math> to <math>\mathcal{U}_i</math>                      ————— <math>\xrightarrow{\text{ViaPublicChannel}}</math> —————                  <b>end</b>              <b>else</b>                    Abort this phase              <b>end</b>  <b>end</b></p>	<p><b>if</b> <math>(T''' - T_s) \leq \Delta T</math> <b>then</b>              <b>if</b> <math>V_s = h(AID_i    X_i    Y_i    RM    T_s)</math> <b>then</b>                  Then <math>\mathcal{U}_i</math> computes <math>K_{US} = r_u \times Y_i</math>,                  <math>sk = h(AID_i    K_{US}    T_s)</math>,                  Accept <math>PM</math>. Where                  <math>sk = h(AID_i    r_i \times r_s \times P    T_s)</math>                  {based on ECDH}              <b>end</b>              <b>else</b>                    Abort this phase              <b>end</b>  <b>end</b>  <b>else</b>                Abort this phase  <b>end</b></p>

### 4.3 Review and Cryptanalysis of Various Recent Protocols of User Authentication for WSNs

---

- $\mathcal{A}$  intercepts the message  $M_1 = \langle AID_i, X_i, C_i, T_i, W_i \rangle$  of  $\mathcal{U}_i$  and find out:  $ID_{\mathcal{U}_i} = AID_i \oplus h(h(x||y)||T_i)$ .
- Thus, we observe that Choi et al. protocol does not present user anonymity i.e., an adversary  $\mathcal{A}$  can evaluate user  $\mathcal{U}_i$ 's identity  $ID_{\mathcal{U}_i}$ . Although, Choi et al. asserted that their protocol supports user anonymity.
- Moreover  $\mathcal{A}$  intercepts the cipher text  $C_i = E_{k_i}(ID_{\mathcal{U}_i}||X_i)$  and derives the plain-text  $(ID_{\mathcal{U}_i}||X_i)$ , accordingly Choi et al. protocol is resistless to known plain-text attack.

#### 4.3.4.2 User Impersonation

In this case, an adversary  $\mathcal{A}$  having a stolen smart card  $SC_{\mathcal{U}_i}$  can impersonate a legitimate user  $\mathcal{U}_i$  of WSNs & IoT as follows:

- $\mathcal{A}$  extracts  $\langle ID_{\mathcal{U}_i}, M_{\mathcal{U}_i}, N_{\mathcal{U}_i}, V_{\mathcal{U}_i}, h(\cdot), \tau_i \rangle$  from the smart card  $SC_{\mathcal{U}_i}$  of the user  $\mathcal{U}_i$  and computes  $A_i^* = N_i \oplus h(x||y)$ ,  $V_i^* = h(ID_{\mathcal{U}_i}||A_i^*)$  and verify the computed  $V_i^*$  with the stored  $V_i$ .
- $\mathcal{A}$  generates a random number  $r_A$ , calculates  $X_A = r_A \times P$ ,  $D_A^* = M_i \oplus A_i^*$ . Find out the current timestamps  $T_A$ , computes  $k_A = h(D_i^*||T_A)$ ,  $C_A = E_{k_A}(ID_{\mathcal{U}_i}||X_A)$ ,  $AID_A = ID_{\mathcal{U}_i} \oplus h(h(x||y)||T_A)$ ,  $W_A = h(h(x||y)||AID_A||X_A||C_A, T_A)$ .
- $\mathcal{A}$  sends the message  $M_1^A = \langle AID_A, X_A, C_A, T_A, W_A \rangle$  to  $GWN$ . Subsequently,  $\mathcal{A}$  establishes the session key  $sk = h(AID_A||r_A \times r_s \times P)$  with  $SN_j$  using Steps 2–4 of authentication and session key establishment phase of Choi et al. protocol.

#### 4.3.5 Review of Park et al.'s protocol

Park et al. [63] did the security analysis of Chang et al.'s [4] protocol and proposed a revised protocol of user authentication using the fuzzy extractor and biometric knowledge to resist off-line password guessing attacks and implement forward secrecy, final password update phase. In that protocol the gateway node  $GWN$  introduces master

### 4.3 Review and Cryptanalysis of Various Recent Protocols of User Authentication for WSNs

keys,  $x$  and  $y$ , and provides a key  $h(ID_{SN_j}||y)$  to the sensor node  $SN_j$ . Then, the protocol supports the registration, login and authentication phase as presented in Tables 4.4 and 4.5.

**Table 4.4:** User registration phase of Park et al.'s protocol.

Step 1: For User ( $\mathcal{U}_i$ )	Step 2: For Gateway ( $GW_N$ )
$\mathcal{U}_i$ selects the identity $ID_{\mathcal{U}_i}$ , imprints bio-metric information $BIO_{\mathcal{U}_i}$ and computes: $(\sigma_{\mathcal{U}_i}, \tau_{\mathcal{U}_i}) = Gen(BIO_{\mathcal{U}_i}), A_{\mathcal{U}_i} = h(\sigma_{\mathcal{U}_i})$ $\xrightarrow[\text{Via Secure Channel}]{\mathcal{U}_i \text{ transmits } \langle ID_{\mathcal{U}_i}, A_{\mathcal{U}_i} \rangle \text{ to } GW_N}$	$GW_N$ computes 1024 bit secret key $x$ and Computes: $M_{\mathcal{U}_i} = h(x  y  A_i),$ $N_{\mathcal{U}_i} = M_{\mathcal{U}_i} \oplus A_{\mathcal{U}_i},$ $V_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i}  A_i),$ $C_{\mathcal{U}_i} = Enc_x(A_{\mathcal{U}_i}  up_{\mathcal{U}_i})$ Store $\langle V_{\mathcal{U}_i}, C_{\mathcal{U}_i}, N_{\mathcal{U}_i}, h(.) \rangle$ into smart card $SC_{\mathcal{U}_i}$ . $\xleftarrow{GW_N \text{ sends smart card } SC_{\mathcal{U}_i} \text{ to } \mathcal{U}_i}$
<b>Step 3: For User <math>\mathcal{U}_i</math></b> $\mathcal{U}_i$ Inputs $\tau_{\mathcal{U}_i}$ into the smart card $SC_{\mathcal{U}_i}$	

#### 4.3.6 Cryptanalysis of Park et al.'s protocol

In this section, we present the cryptanalysis of the Park et al.'s protocol and have observed that Park et al.'s protocol is also unsafe and it possesses the following security vulnerabilities:

##### 4.3.6.1 Sensor Node Impersonation Attack

According to Park et al., to impersonate a sensor node  $SN_j$ , an adversary  $\mathcal{A}$  need to have the key  $k_{GW_N} = h(h(ID_{SN_j}||y)||T_{GW_N})$ . Although, an adversary  $\mathcal{A}$  can impersonate the sensor node  $SN_j$  without having  $k_{GW_N}$  with the help of following steps:

### 4.3 Review and Cryptanalysis of Various Recent Protocols of User Authentication for WSNs

**Table 4.5:**  $\mathcal{U}_i$ 's authentication and session key sharing phase of Park et al. protocol.

Step 1: For User ( $\mathcal{U}_i$ )	Step 2: For Gateway ( $\mathcal{GWN}$ )
<p><math>\mathcal{U}_i</math> inserts the smart card <math>SC_{\mathcal{U}_i}</math>, inputs <math>ID_{\mathcal{U}_i}</math> and imprints <math>BIO'_{\mathcal{U}_i}</math>. Then, computes <math>\sigma'_{\mathcal{U}_i} = Rep(BIO'_{\mathcal{U}_i}, \tau_{\mathcal{U}_i})</math>,  <math>A'_{\mathcal{U}_i} = h(\sigma'_{\mathcal{U}_i})</math>, <math>V'_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i}    A'_{\mathcal{U}_i})</math>,  <b>if</b> <math>V_{\mathcal{U}_i} = V'_{\mathcal{U}_i}</math> <b>then</b>              Generate random number <math>r_{\mathcal{U}_i}</math>, and computes              <math>X_{\mathcal{U}_i} = r_{\mathcal{U}_i} \times P</math>, <math>M_{\mathcal{U}_i} = N_{\mathcal{U}_i} \oplus A'_{\mathcal{U}_i}</math>,              Find out current time-stamp <math>T_{\mathcal{U}_i}</math> and computes              <math>AID_{\mathcal{U}_i} = ID_{\mathcal{U}_i} \oplus h(M_{\mathcal{U}_i}    T_{\mathcal{U}_i})</math>,              <math>W_{\mathcal{U}_i} = h(M_{\mathcal{U}_i}    ID_{\mathcal{U}_i}    X_{\mathcal{U}_i}    T_{\mathcal{U}_i})</math>.              Then, <math>\mathcal{U}_i</math> constructs a message              <math>M_1 = \langle AID_{\mathcal{U}_i}, X_{\mathcal{U}_i}, C_{\mathcal{U}_i}, T_{\mathcal{U}_i}, W_{\mathcal{U}_i} \rangle</math>              Finally, <math>\mathcal{U}_i</math> transmits <math>M_1</math> to <math>\mathcal{GWN}</math>              ViaPublicChannel  <b>end</b>  <b>else</b>              Abort this phase.  <b>end</b></p>	<p><b>if</b> <math>(T' - T_{\mathcal{U}_i}) \leq \Delta T</math> <b>then</b>              <math>A_i^* \leftarrow Dec_x(C_{\mathcal{U}_i})</math>, <math>M_{\mathcal{U}_i}^* = h(x    y    A_{\mathcal{U}_i}^*)</math>,              <math>ID'_{\mathcal{U}_i} = AID_{\mathcal{U}_i} \oplus h(M_{\mathcal{U}_i}^*    T_{\mathcal{U}_i})</math>,              <math>W'_{\mathcal{U}_i} = h(M'_{\mathcal{U}_i}    ID'_{\mathcal{U}_i}    X'_{\mathcal{U}_i}    T_{\mathcal{U}_i})</math> <b>if</b> <math>W_{\mathcal{U}_i} = W'_{\mathcal{U}_i}</math>              <b>then</b>                  Find the current time-stamp <math>T_{\mathcal{GWN}}</math> and computes                  <math>k_{\mathcal{GWN}} = h(h(ID_{SN_j}    y)    T_{\mathcal{GWN}})</math>,                  <math>C_{\mathcal{GWN}} = Enc_{k_{\mathcal{GWN}}}(AID_i    X'_{\mathcal{U}_i})</math>, <math>W_{\mathcal{GWN}} =</math>                  <math>h(h(ID_{SN_j}    y)    AID_{\mathcal{U}_i}    C_{\mathcal{GWN}}    T_{\mathcal{GWN}})</math>.                  Then, <math>\mathcal{GWN}</math> constructs the message                  <math>M_2 = \langle AID_{\mathcal{GWN}}, C_{\mathcal{GWN}}, T_{\mathcal{GWN}}, W_{\mathcal{GWN}} \rangle</math>                  Finally, <math>\mathcal{GWN}</math> transmits <math>M_2</math> to <math>SN_j</math>                  PublicChannel              <b>end</b>              <b>else</b>                  Abort this phase.              <b>end</b>  <b>end</b>  <b>else</b>              Abort this phase.  <b>end</b></p>
Step 3: For Sensor Node ( $SN_j$ )	Step 4: For User ( $\mathcal{U}_i$ )
<p><b>if</b> <math>(T'' - T_{\mathcal{GWN}}) \leq \Delta T</math> <b>then</b>              <math>k'_{\mathcal{GWN}} = h(h(ID_{SN_j}    y)    T_{\mathcal{GWN}})</math>              <math>AID'_{\mathcal{U}_i}    X'_{\mathcal{U}_i} = Dec_{k'_{\mathcal{GWN}}}(C_{\mathcal{GWN}})</math>              <b>if</b> <math>W_{\mathcal{GWN}} =</math>              <math>h(h(ID_{SN_j}    y)    AID_{\mathcal{U}_i}    C_{\mathcal{GWN}}    T_{\mathcal{GWN}})</math>              <b>and</b> <math>(AID_{\mathcal{U}_i} = AID'_{\mathcal{U}_i})</math> <b>then</b>                  <math>SN_j</math> generates random number <math>r_{SN_j}</math>, computes                  <math>K_{SU} = r_{SN_j} \times X'_{\mathcal{U}_i}</math>, <math>Y_{\mathcal{U}_i} = r_{SN_j} \times P</math>                  and <math>sk = h(AID_{\mathcal{U}_i}    K_{SU}    T_{SN_j})</math>. Then,                  <math>SN_j</math> finds the current time-stamp <math>T_{SN_j}</math>, computes                  <math>RM = \text{Query response}</math>,                  <math>V_{SN_j} = h(AID'_{\mathcal{U}_i}    X'_{\mathcal{U}_i}    Y_{\mathcal{U}_i}    RM    T_{SN_j})</math>,                  <math>M_3 = \langle RM, Y_{\mathcal{U}_i}, V_{SN_j}, T_{SN_j} \rangle</math>.                  Finally, <math>SN_j</math> transmits <math>M_3</math> to <math>\mathcal{U}_i</math>                  PublicChannel              <b>end</b>              <b>else</b>                  Abort this phase.              <b>end</b>  <b>end</b>  <b>else</b>              Abort this phase.  <b>end</b></p>	<p><b>if</b> <math>(T''' - T_{SN_j}) \leq \Delta T</math> <b>then</b>              <b>if</b> <math>V_{SN_j} = h(AID_{\mathcal{U}_i}    X_{\mathcal{U}_i}    Y_{\mathcal{U}_i}    RM    T_{SN_j})</math> <b>then</b>                  <math>\mathcal{U}_i</math> computes <math>K_{US} = r_{\mathcal{U}_i} \times Y_{\mathcal{U}_i}</math>,                  <math>sk = h(AID_{\mathcal{U}_i}    K_{US}    T_{SN_j})</math>, accepts <math>RM</math> and                  establishes the session key                  <math>sk = h(AID_{\mathcal{U}_i}    r_{\mathcal{U}_i} \times r_{SN_j} \times P    T_{SN_j})</math> with                  <math>SN_j</math>.              <b>end</b>              <b>else</b>                  Abort this phase.              <b>end</b>  <b>end</b>  <b>else</b>              Abort this phase.  <b>end</b></p>

### 4.3 Review and Cryptanalysis of Various Recent Protocols of User Authentication for WSNs

- The adversary  $\mathcal{A}$  intercepts the message  $M_1 = \langle AID_{\mathcal{U}_i}, X_{\mathcal{U}_i}, C_{\mathcal{U}_i}, T_{\mathcal{U}_i}, W_{\mathcal{U}_i} \rangle, M$  and  $M_2 = \langle AID_{GWN}, C_{GWN}, T_{GWN}, W_{GWN} \rangle$ .
- Then,  $\mathcal{A}$  generates a random number  $r_{\mathcal{A}}$ , finds current timestamp  $T_{\mathcal{A}}$  and computes:  $K_{AU} = r_{\mathcal{A}} \times X_{\mathcal{U}_i}$ ,  $Y_{\mathcal{A}} = r_{\mathcal{A}} \times P$ ,  $sk = h(AID_{\mathcal{U}_i} || K_{AU} || T_{\mathcal{A}})$ ,  $RM = \text{Query response}$  and  $V_{\mathcal{A}} = h(AID_{\mathcal{U}_i} || X_{\mathcal{U}_i} || Y_{\mathcal{A}} || RM || T_{\mathcal{A}})$ .
- Afterwards,  $\mathcal{A}$  sends  $M_3 = \langle RM, Y_{\mathcal{A}}, V_{\mathcal{A}}, T_{\mathcal{A}} \rangle$  to  $\mathcal{U}_i$ .
- After receiving  $M_3$ ,  $\mathcal{U}_i$  computes:  $V_{\mathcal{A}} = h(AID_{\mathcal{U}_i} || X_{\mathcal{U}_i} || Y_{\mathcal{A}} || RM || T_{\mathcal{A}})$ . If  $V_{\mathcal{A}}^* = V_{\mathcal{A}}$ ,  $\mathcal{A}$  computes  $K_{UA} = r_{\mathcal{U}_i} \times Y_{\mathcal{A}}$ ,  $sk = h(AID_{\mathcal{U}_i} || K_{UA} || T_{\mathcal{A}})$ .

Therefore, the adversary  $\mathcal{A}$  succeeds in impersonating the sensor node  $SN_j$  and establishing the session key  $sk$  with the user  $\mathcal{U}_i$ .

#### 4.3.6.2 User Impersonation Attack

In Park et al.'s protocol, a legitimate user  $U_k$  can be an adversary  $U_{\mathcal{A}}$  to impersonate the user  $\mathcal{U}_i$  because  $U_k$  can find out the hashed master key  $h(x||y)$  and then it can derive the secret information of user  $\mathcal{U}_i$  as follows:

- First, the adversary  $\mathcal{A}$  extract the information  $\langle V_{U_k}, N_{U_k}, C_{U_k}, h(\cdot), P_{U_k} \rangle$  from the smart card.
- Then,  $\mathcal{A}$  imprints its biometric information  $B'_k$  and computes  $\sigma'_k = \text{Rep}(B'_k, P_{U_k})$  and  $A'_{U_k} = h(\sigma'_k)$ ,  $M_{U_k} = N_{U_k} \oplus A'_{\mathcal{A}}$ .
- Afterwards,  $\mathcal{A}$  generates random number  $r_{\mathcal{A}}$ , selects an identity  $ID_{\mathcal{U}_i}$  and computes:  $X_{\mathcal{A}} = X_{U_k} = r_{U_k} \times P$ ,  $AID_{\mathcal{U}_i} = ID_{\mathcal{U}_i} \oplus h(M_{\mathcal{A}} || T_{\mathcal{A}})$  and  $W_{\mathcal{A}} = W_{U_k} = h(M_{\mathcal{A}} || ID_{\mathcal{U}_i} || X_{\mathcal{A}} || T_{\mathcal{A}})$ . Finally,  $\mathcal{A}$  sends  $M_1 = \langle AID_{\mathcal{U}_i}, X_{\mathcal{A}}, C_{U_k}, T_{U_k}, W_{U_k} \rangle$  to  $GWN$ .
- After receiving  $M_1$ , if  $(T' - T_{U_k}) \leq \Delta T$ ,  $GWN$  computes  $A'_{\mathcal{U}_i} || up_{U_k} = \text{Dec}_x(C_{U_k})$ ,  $M'_{U_k} = h(x||y||A'_{U_k})$ ,  $ID'_{\mathcal{U}_i} = AID_{\mathcal{U}_i} \oplus h(M'_{U_k} || T_{U_k})$ ,  $W'_{\mathcal{U}_i} = h(M_{U_k} || ID'_{\mathcal{U}_i} || X'_{U_k} || T_{U_k})$ .
- If  $(W_{\mathcal{U}_i} = W'_{\mathcal{U}_i})$ , the  $GWN$  finds the current time stamp  $T_{GWN}$  and computes:  $k_{GWN} = h(h(ID_{SN_j} || y) || T_{GWN})$ ,  $C_{GWN} = \text{Enc}_{k_{GWN}}(AID_{U_k} || X_{U_k})$ ,  $W_{GWN} = h(h(ID_{SN_j} || y) || AID_{\mathcal{U}_i} || C_{GWN} || T_{GWN})$ .

- Finally,  $GWN$  sends  $M_2 = \langle AID_{U_k}, W_{GWN}, C_{GWN}, T_{GWN} \rangle$  to  $SN_j$ .
- After receiving  $M_2$ , if  $(T''' - T_{GWN}) \leq \Delta T$  and  $W_{GWN} = h(h(ID_{SN_j} || y) || AID_{U_k} || C_{GWN} || T_{GWN})$ ,  $SN_j$  computes:  $k'_{GWN} = h(h(ID_{SN_j} || y) || T_{GWN})$  and  $(AID'_{U_k} || X'_{U_k}) = Dec'_{GWN}(C_{GWN})$ .
- If  $(AID_{U_k} = AID'_{U_k})$ ,  $SN_j$  generates a random number  $r_{SN_j}$  and computes:  $K_{SU} = r_{SN_j} \times X'_{U_k}$ ,  $Y_{U_i} = r_{SN_j} \times P$ ,  $sk = h(AID_{U_i} || K_{SU} || T_{SN_j})$ ,  $RM = \text{Query Response}$ ,  $V_{SN_j} = h(AID_{U_k} || X_{U_k} || Y_{U_i} || RM || T_{SN_j})$ .
- Then,  $SN_j$  sends  $M_3 = \langle RM, Y_{U_i}, V_{SN_j}, T_{SN_j} \rangle$  to the adversary  $\mathcal{A}$ .
- After receiving  $M_3$ , if  $(T'' - T_{SN_j}) \leq \Delta T$ , the adversary  $\mathcal{A}$  computes:  $V'_{SN_j} = h(AID_{U_k} || X_{U_k} || Y_{U_i} || RM || T_{SN_j})$ . If  $(V_{SN_j} = V'_{SN_j})$ ,  $SN_j$  computes  $K_{US} = r_{U_k} \times Y_{U_i}$  and establishes the session key  $sk = h(AID_{U_k} || K_{US} || T_{SN_j})$  with sensor node  $SN_j$ . Therefore, Park et al.'s protocol is vulnerable to user impersonation attack. Similar attack is possible in Moon et al.'s protocol [64] also, since the value of  $C_{U_i}$  in Moon et al.'s protocol can be evaluated using  $x, y$  and  $N_{U_i}$ .

## 4.4 Proposed Protocol

In our proposed protocol, we consider that the WSNs & IoT consist of several users (with the smart card which can be captured or stolen by the adversary  $\mathcal{A}$ ), hundreds of sensor nodes ( $\mathcal{A}$  can capture these nodes) and trusted gateway node. Regarding these entities, we formulate the protocol which consists of four significant components (i) Set-up before the deployment of WSNs & IoT (ii) Registration of  $\mathcal{U}_i$  by the  $GWN$  (iii)  $\mathcal{U}_i$ 's authentication and session key establishment phase (iv)  $\mathcal{U}_i$ 's credentials update phase.

### 4.4.1 Set-Up before the Deployment of WSNs & IoT

In this phase, we choose a high-performance and trusted computing node as a gateway  $GWN$ . The  $GWN$  designates a unique identity  $ID_{SN_j}$  for each sensor node  $SN_j$  and loads a unique secret key  $K_{GSN_j} = h(ID_{SN_j} || K_{GWN})$  into the memory of  $SN_j$ .

### 4.4.2 Registration of $\mathcal{U}_i$ by the $GWN$ Using Secure Communication Channel

In this phase, a legitimate user  $\mathcal{U}_i$  sends the hashed secret credential to  $GWN$  using a secure communication channel and the  $GWN$  provides a smart card (consisting of some secret parameter which is known only to the  $GWN$ )  $SC_{\mathcal{U}_i}$  to  $\mathcal{U}_i$ . The steps associated with the proposed user registration phase are described in following Steps R1, R2, R3 and summarized in Table 4.6 (using Steps 1–3).

**Table 4.6:** User registration phase of proposed protocol.

Step 1: For User ( $\mathcal{U}_i$ )	Step 2: For Gateway ( $GWN$ )
$\mathcal{U}_i$ inputs $ID_{\mathcal{U}_i}$ , $PW_{\mathcal{U}_i}$ and $BIO_{\mathcal{U}_i}$ Computes: $Gen(BIO_{\mathcal{U}_i}) = (\sigma_{\mathcal{U}_i}, \tau_{\mathcal{U}_i})$ , $PB_{\mathcal{U}_i} = h(PW_{\mathcal{U}_i}    \sigma_{\mathcal{U}_i})$  $\mathcal{U}_i$ transmits $\langle ID_{\mathcal{U}_i}, PB_i \rangle$ to $GWN$ <span style="display: block; text-align: center;"><math>\xrightarrow{\text{ViaSecureChannel}}</math></span>	$GWN$ computes 1024 bit secret key $x$ and Computes: $K_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i}    x) \times P$ , $A_{\mathcal{U}_i} = PB_i \oplus h(ID_{\mathcal{U}_i} \oplus x)$ , $B_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i}    PB_i    h(ID_{\mathcal{U}_i} \oplus x))$ , $W_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i}    PB_i) \oplus K_{\mathcal{U}_i}$ $GWN$ stores the value of $P, A_{\mathcal{U}_i}, B_{\mathcal{U}_i}, W_{\mathcal{U}_i}$ into $SC_{\mathcal{U}_i}$ . <span style="display: block; text-align: center;"><math>\xleftarrow{GWN \text{ transmits } \langle SC_{\mathcal{U}_i} \rangle \text{ to } \mathcal{U}_i}</math></span>
Step 3: For User ( $\mathcal{U}_i$ )	
$\mathcal{U}_i$ stores $\mathcal{T}, h(), Gen(), Rep()$ and the value of $\tau_i$ into $SC_{\mathcal{U}_i}$ .	

**Step R1:** A legitimate user  $\mathcal{U}_i$  selects her identity  $ID_{\mathcal{U}_i}$ , password  $PW_{\mathcal{U}_i}$  and inputs his/her biometric information  $BIO_{\mathcal{U}_i}$  into the generator function  $Gen()$  which generates a secret information  $\sigma_i$  and a public reproduction parameter  $\tau_i$ . Then,  $\mathcal{U}_i$  calculates  $PB_i = h(PW_{\mathcal{U}_i} || \sigma_i)$  using secure hash function  $h()$  and sends  $ID_{\mathcal{U}_i}, PB_i$  to the gateway node  $GWN$ .

**Step R2:**  $GWN$  initiates a secret key  $x$ , chooses a generator or base point  $P$  of  $G$  with order  $q$  and computes:  $K_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i} || x) \times P$  (where “ $\times$ ” is the scalar multiplication operator of elliptic curve),  $A_{\mathcal{U}_i} = PB_i \oplus h(ID_{\mathcal{U}_i} \oplus x)$ ,



$B_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i} || PB_i || h(ID_{\mathcal{U}_i} \oplus x)), W_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i} || PB_i) \oplus K_{\mathcal{U}_i}$  Finally, the gateway node  $GWN$  stores the value of  $P, A_{\mathcal{U}_i}, B_{\mathcal{U}_i}, W_{\mathcal{U}_i}$  into the smart card  $SC_{\mathcal{U}_i}$  and sends  $SC_{\mathcal{U}_i}$  to the user  $\mathcal{U}_i$ .

**Step R3:** After receiving the  $SC_{\mathcal{U}_i}$  from  $GWN$ , the user  $\mathcal{U}_i$  stores function  $h(), Gen(), Rep()$  and the values of  $\mathcal{T}, \tau_i$  into  $SC_{\mathcal{U}_i}$ .

#### 4.4.3 User Authentication and Session Key Establishment Phase

In this section, we use the reproduction procedure  $Rep(.)$  of fuzzy extractor for authentication the user  $\mathcal{U}_i$  with its noisy biometric credential  $BIO'_{\mathcal{U}_i}$  and we use Elliptic curve Diffie-Hellman procedure for sharing the common session key  $SK$  between user  $\mathcal{U}_i$  and sensor node  $SN_j$ . The detailed description of this phase are given in following Steps A1 to A4 and summarized in Table 4.7 (using Steps 1–4).

**Step A1:**  $\mathcal{U}_i$  inputs  $ID_{\mathcal{U}_i}, PW_{\mathcal{U}_i}$ , imprints her noisy biometric information  $BIO'_{\mathcal{U}_i}$  and computes  $\sigma'_i = Rep(BIO'_{\mathcal{U}_i}, \tau_i)$  using reproduction function of fuzzy extractor as described in Definition 1.3.5. Then,  $\mathcal{U}_i$  calculates  $PB'_i = h(PW_{\mathcal{U}_i} || \sigma'_i)$ ,  $h'(ID_{\mathcal{U}_i} \oplus x) = A_{\mathcal{U}_i} \oplus PB'_i$ ,  $B'_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i} || PB'_i || h'(ID_{\mathcal{U}_i} \oplus x))$ .

If the equivalent condition  $B'_{\mathcal{U}_i} = B_{\mathcal{U}_i}$  does not fulfill; abort the protocol. Otherwise,  $\mathcal{U}_i$  evaluates  $K_{\mathcal{U}_i} = W_{\mathcal{U}_i} \oplus h'(ID_{\mathcal{U}_i} || PB'_i)$ , generates a random number  $r_{\mathcal{U}_i} \in \mathbb{Z}_q^*$  and find out her current time stamp  $T_{\mathcal{U}_i}$ . Then, the user  $\mathcal{U}_i$  calculates  $X_{\mathcal{U}_i} = r_{\mathcal{U}_i} \times P$ ,  $X'_{\mathcal{U}_i} = r_{\mathcal{U}_i} \times K_{\mathcal{U}_i}$  (where “ $\times$ ” is the scalar multiplication operator of elliptic curve) and encrypts the message  $(ID_{SN_j} || T_{\mathcal{U}_i})$  considering  $X'_{\mathcal{U}_i}$  as a symmetric key to find:  $\alpha = Enc_{X'_{\mathcal{U}_i}}[ID_{SN_j} || T_{\mathcal{U}_i}]$ . Finally,  $\mathcal{U}_i$  Construct a message  $M_3 = \langle ID_{\mathcal{U}_i}, X_{\mathcal{U}_i}, \alpha \rangle$  and sends  $M_3$  to the gateway node  $GWN$ .

**Step A2:** After receiving the message  $M_3$ , the gateway node  $GWN$  compute  $X'_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i} || x) \times X_{\mathcal{U}_i}$  and decrypts the cipher text  $\alpha$  considering  $X'_{\mathcal{U}_i}$  as a symmetric key to find:  $[ID_{SN_j} || T_{\mathcal{U}_i}] = Dec_{X'_{\mathcal{U}_i}}[\alpha]$ . And if the condition  $T' - T_{\mathcal{U}_i} \leq \Delta T$  does not fulfill; the  $GWN$  aborts the protocol. Otherwise, the gateway node  $GWN$  generates a random number  $r_{SN_j} \in \mathbb{Z}_q^*$  and calculates  $Y_{SN_j} = r_{SN_j} \times P$ , the session key  $sk = r_{SN_j} \times X_{\mathcal{U}_i}$  (where “ $\times$ ” is the scalar multiplication operator of elliptic curve). Then, the gateway node  $GWN$  finds its current time-stamp  $T_{GWN}$  and calculates:  $\beta = Enc_{X'_{\mathcal{U}_i}}[ID_{SN_j} || Y_{SN_j} || T_{GWN}]$ ,

## 4.5 Security Analysis:

$\gamma = Enc_{K_{GSN_j}}[ID_{U_i}||sk||\beta||T_{GWN}]$ . Finally,  $GWN$  constructs the message  $M_4 = \langle \gamma \rangle$  and sends  $M_4$  to the sensor node  $SN_j$ .

**Step A3:** After receiving the message  $M_4$ ,  $SN_j$  decrypts the cipher text  $\beta$  using symmetric key  $K_{GSN_j}$  to find out:  $[ID_{U_i}||sk||\beta||T_{GWN}] = Dec_{K_{GSN_j}}[\gamma]$ . If the condition  $(T'' - T_{GWN}) \leq \Delta T$  fulfills,  $SN_j$  stores the session key  $sk$  and finally transmits the message  $M_5 = \beta$  to  $U_i$ .

**Step A4:** After receiving the message  $M_5$ , the user  $U_i$  decrypts the message  $M_5 = \beta$  considering  $X'_{U_i}$  as a symmetric key and find out:  $[ID_{SN_j}||Y_{SN_j}||T_{GWN}] = D_{X'_{U_i}}[\beta]$ . Once the condition  $(T''' - T_{GWN}) \leq 2\Delta T$  fulfills, the user  $U_i$  establishes the session key  $sk = r_{U_i} \times Y_{SN_j}$  with  $SN_j$ . Where  $r_{U_i} \times Y_{SN_j} = r_{SN_j} \times X_{U_i}$  on the basis of  $ECDH$  problem.

The sequence diagram of the message transmission for the user registration, authentication and key establishment phase is shown in below Figure 4.1.

### 4.4.4 User's Credential Update Phase

If a legitimate user gets authenticated using his/her identity  $ID_{U_i}$ , password  $PW_{U_i}$ , biometric information  $BIO_{U_i}$  and the smart card  $SC_{U_i}$ , it can update own password and biometric information using the mechanism described in Table 4.8.

## 4.5 Security Analysis:

To estimate the security strength of our proposed protocol, we perform the informal and formal analysis of security features.

### 4.5.1 Informal Analysis

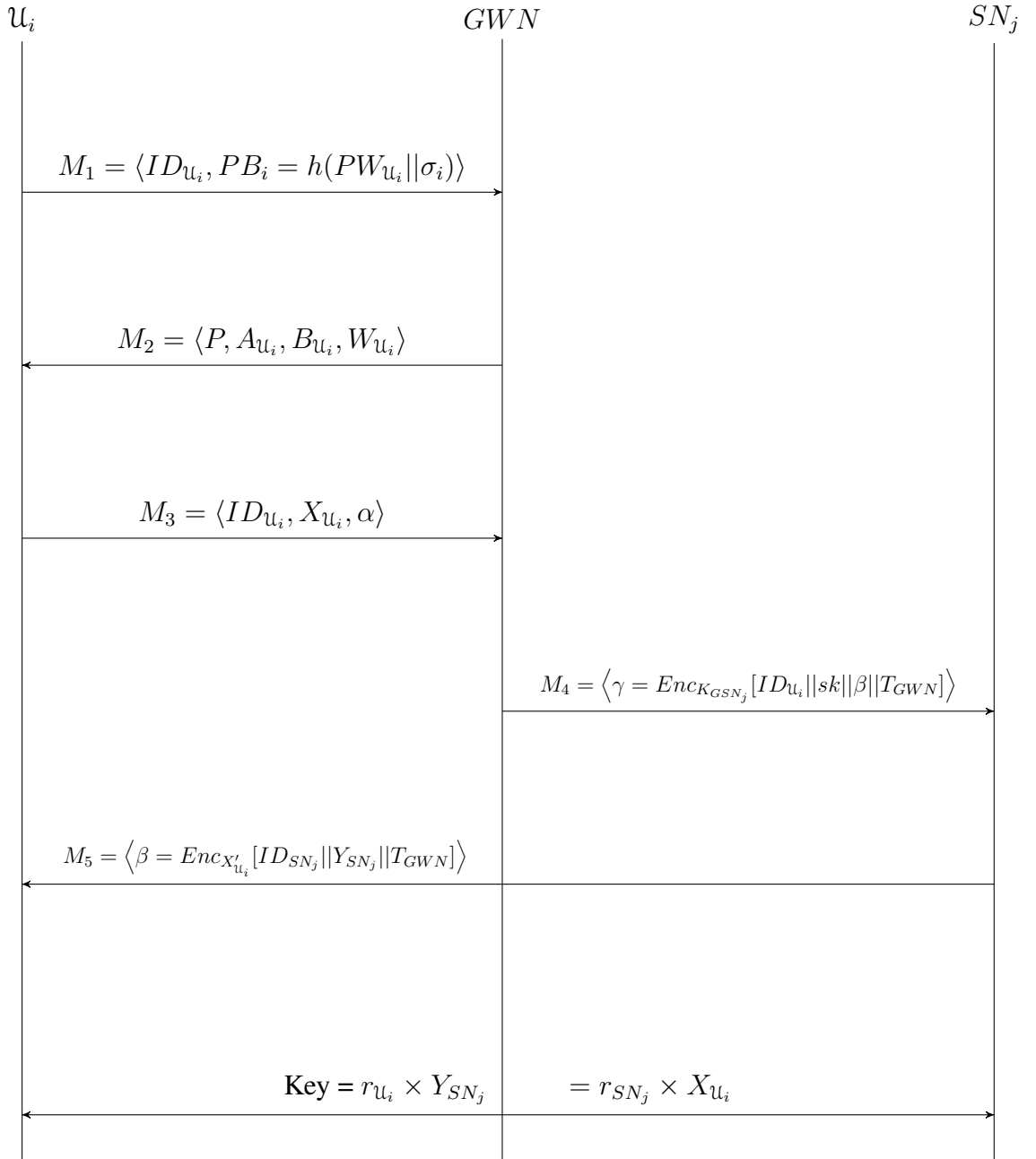
Our proposed protocol can withstand various known security attacks as illustrated in the following propositions.

**Proposition 1.** *The proposed protocol is secure against Stolen Smart Card Attack.*

**Table 4.7:** User authentication and session key establishment phase of the proposed protocol.

Step 1: For User ( $\mathcal{U}_i$ )	Step 2: For Gateway ( $GWN$ )
<p><math>\mathcal{U}_i</math> inputs <math>ID_{\mathcal{U}_i}</math>, <math>PW_{\mathcal{U}_i}</math> and <math>BIO'_{\mathcal{U}_i}</math>.  Computes <math>\sigma'_i = Rep(BIO'_{\mathcal{U}_i}, \tau_i)</math>, <math>PB'_i = h(PW_{\mathcal{U}_i}    \sigma'_i)</math>,  <math>h'(ID_{\mathcal{U}_i} \oplus x) = A_{\mathcal{U}_i} \oplus PB'_i</math>,  <math>B'_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i}    PB'_i    h'(ID_{\mathcal{U}_i} \oplus x))</math>.  <b>if</b> <math>B'_{\mathcal{U}_i} = B_{\mathcal{U}_i}</math> <b>then</b>      Evaluate <math>K_{\mathcal{U}_i} = W_{\mathcal{U}_i} \oplus h'(ID_{\mathcal{U}_i}    PB'_i)</math>.      Generate <math>r_{\mathcal{U}_i} \in \mathbb{Z}_q^*</math>.      Find current time stamp <math>T_{\mathcal{U}_i}</math>,      <math>X_{\mathcal{U}_i} = r_{\mathcal{U}_i} \times P</math>, <math>X'_{\mathcal{U}_i} = r_{\mathcal{U}_i} \times K_{\mathcal{U}_i}</math>,      <math>\alpha = Enc_{X'_{\mathcal{U}_i}}[ID_{SN_j}    T_{\mathcal{U}_i}]</math>.      Construct a message <math>M_3 = \langle ID_{\mathcal{U}_i}, X_{\mathcal{U}_i}, \alpha \rangle</math>      <math>\xrightarrow[\text{ViaPublicChannel}]{\mathcal{U}_i \text{ transmits } M_3 \text{ to } GWN}</math>  <b>end</b>  <b>else</b>      <math>\mathcal{U}_i</math> is unauthenticated, abort this phase.  <b>end</b></p>	<p><math>GWN</math> computes <math>X'_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i}    x) \times X_{\mathcal{U}_i}</math>,  <math>[ID_{SN_j}    T_{\mathcal{U}_i}] = Dec_{X'_{\mathcal{U}_i}}[\alpha]</math>,  <b>if</b> <math>T' - T_{\mathcal{U}_i} \leq \Delta T</math> <b>then</b>      Generates <math>r_{SN_j} \in \mathbb{Z}_q^*</math>,      Calculate <math>Y_{SN_j} = r_{SN_j} \times P</math>,      Session key <math>sk = r_{SN_j} \times X_{\mathcal{U}_i}</math>,      Find Current time-stamp <math>T_{GWN}</math>,      <math>\beta = Enc_{X'_{\mathcal{U}_i}}[ID_{SN_j}    Y_{SN_j}    T_{GWN}]</math>,      <math>\gamma = Enc_{K_{GWN}}[ID_{\mathcal{U}_i}    sk    \beta    T_{GWN}]</math>,      Construct the message <math>M_4 = \langle \gamma \rangle</math>      <math>\xrightarrow[\text{ViaPublicChannel}]{GWN \text{ transmits } M_4 \text{ to } SN_j}</math>  <b>end</b>  <b>else</b>      Replay and energy exhausting attack possible. Abort this phase.  <b>end</b></p>
Step 3: For Sensor Node ( $SN_j$ )	Step 4: For User ( $\mathcal{U}_i$ )
<p><math>SN_j</math> computes  <math>[ID_{\mathcal{U}_i}    sk    \beta    T_{GWN}] = Dec_{K_{GWN}}[\gamma]</math>,  <b>if</b> <math>T'' - T_{GWN} \leq \Delta T</math> <b>then</b>      Store the session key <math>sk</math>      Construct the message <math>M_5 = \langle \beta \rangle</math>      <math>\xrightarrow[\text{ViaPublicChannel}]{SN_j \text{ transmits } M_5 \text{ to } \mathcal{U}_i}</math>  <b>end</b>  <b>else</b>      Replay and energy exhausting attack possible. Abort this phase.  <b>end</b></p>	<p><math>\mathcal{U}_i</math> computes <math>[ID_{SN_j}    Y_{SN_j}    T_{GWN}] = D_{X'_{\mathcal{U}_i}}[\beta]</math>  <b>if</b> <math>T''' - T_{GWN} \leq 2\Delta T</math> <b>then</b>      Establish the session key <math>sk = r_{\mathcal{U}_i} \times Y_{SN_j}</math> with <math>SN_j</math>.      Where <math>r_{\mathcal{U}_i} \times Y_{SN_j} = r_{SN_j} \times X_{\mathcal{U}_i}</math> on the basis of <math>ECDH</math>.  <b>end</b>  <b>else</b>      Replay and energy exhausting attack possible. Abort this phase.  <b>end</b></p>

**Figure 4.1:** Sequence Diagram 2 for Registration, Authentication and Key Establishment



**Table 4.8:** User's credential update phase of proposed protocol.

```

 $\mathcal{U}_i$  inserts  $SC_{\mathcal{U}_i}$  into the card reader and
Inputs  $ID_{\mathcal{U}_i}, PW_{\mathcal{U}_i}, BIO'_{\mathcal{U}_i}$ .
Then,  $\mathcal{U}_i$  computes  $\sigma'_i = Rep(BIO'_{\mathcal{U}_i}, \tau_i)$ , {Using fuzzy extractor }
 $PB'_i = h(PW_{\mathcal{U}_i} || \sigma_i)$ ,
 $h'(ID_{\mathcal{U}_i} \oplus x) = A_{\mathcal{U}_i} \oplus PB'_i$ ,
 $B'_i = h(ID_{\mathcal{U}_i} || PB'_i || h'(ID_{\mathcal{U}_i} \oplus x))$ .
if  $B'_i = B_i$ . then
     $\mathcal{U}_i$  calculates  $h(ID_{\mathcal{U}_i} \oplus x) = A_{\mathcal{U}_i} \oplus PB'_i$ ,
     $K_{\mathcal{U}_i} = W_{\mathcal{U}_i} \oplus h(ID_{\mathcal{U}_i} || PB'_i)$ ,
     $\mathcal{U}_i$  inputs new  $PW_{new_i}, B_{new_i}$ ,
    Then,  $\mathcal{U}_i$  computes  $Gen(B_{new_i}) = (\sigma_{new_i}, \tau_{new_i})$  {Using fuzzy ex-
    tractor },
     $PB_i^{new} = h(PW_{new_i} || \sigma_{new_i})$ ,
     $A_{\mathcal{U}_i}^{new} = PB_i^{new} \oplus h(ID_{\mathcal{U}_i} \oplus x)$ ,
     $B_{\mathcal{U}_i}^{new} = h(ID_{\mathcal{U}_i} || PB_i^{new} || h(ID_{\mathcal{U}_i} \oplus x))$ ,
     $W_{\mathcal{U}_i}^{new} = h(ID_{\mathcal{U}_i} || PB_i^{new}) \oplus K_{\mathcal{U}_i}$ ,
    Finally, replaces the value of  $A_{\mathcal{U}_i}, B_{\mathcal{U}_i}, W_{\mathcal{U}_i}$  with  $A_{\mathcal{U}_i}^{new}, B_{\mathcal{U}_i}^{new}, W_{\mathcal{U}_i}^{new}$ .
    into  $SC_{\mathcal{U}_i}$ 
end
else
    User  $\mathcal{U}_i$  is unauthenticated. Abort protocol to avoid stolen smart card
    attack.
end

```

*Proof.* An adversary  $\mathcal{A}$  who have stolen the smart card  $SC_{\mathcal{U}_i}$  can extract the intimate data such as  $A_{\mathcal{U}_i}, B_{\mathcal{U}_i}, W_{\mathcal{U}_i}, h(\cdot), Rep(\cdot), Gen(\cdot), \tau_i$  from the  $SC_{\mathcal{U}_i}$  using side channel attacks such as differential and simple power analysis [14]. However, in our protocol the most important private information such as  $\sigma_{\mathcal{U}_i}, x$  and  $K_{\mathcal{U}_i}$  are stored in well-protected form. If  $\mathcal{A}$  succeed to find out  $A_{\mathcal{U}_i}$ , it can not find out  $PB_i$  or  $h(ID_{\mathcal{U}_i} \oplus x)$  using frequency analysis attack. The private information  $\sigma_{\mathcal{U}_i}$  also can not be extracted by  $\mathcal{A}$  because it is hashed after concatenated with  $PW_{\mathcal{U}_i}$ .  $\square$

**Proposition 2.** *The proposed protocol is secure against node compromise attack.*

*Proof.* According to our presumption, the sensor node  $SN_j$  is not fixed with tamper resistant hardware, therefore an adversary  $\mathcal{A}$  can capture the sensor node  $SN_j$  and find out the value of the key  $K_{GSN_j}$  and session key  $sk$ . However,  $\mathcal{A}$  can not use the same session key at next session because we made the session key unique using the random number  $r_{\mathcal{U}_i}$  and  $r_{SN_j}$ . If  $\mathcal{A}$  captures the key  $K_{GSN_j}$  from  $SN_j$ , it can establish a session key with any user who wants to access data from  $SN_j$  but it can not establish a session key with any other user associated with non-compromised sensor node because the key  $K_{GSN_j}$  is uniquely given to  $SN_j$ .  $\square$

**Proposition 3.** *The proposed protocol is secure against Man-in-the-middle attack.*

*Proof.* Suppose an adversary  $\mathcal{A}$  eavesdrops the message  $M_3$  during user authentication and session key establishment phase, generates a random number  $r_{\mathcal{A}}$  and the current time-stamp  $T_{\mathcal{A}}$ . However,  $\mathcal{A}$  can not evaluate the value of  $X'_{\mathcal{U}_i}$  without knowing the bio-metric information and smart card credentials of  $\mathcal{U}_i$  in order to decrypt and modify the value of  $\alpha$ . Likewise, it is computationally infeasible for an adversary  $\mathcal{A}$  to modify the value of  $\gamma$  and  $\beta$  without knowing the key  $K_{GSN_j}$  and  $X_{\mathcal{U}_i}$  respectively. Therefore, our protocol is secure against the Man-in-the-middle attack.  $\square$

**Proposition 4.** *The proposed protocol is secure against replay attack.*

*Proof.* Suppose an adversary  $\mathcal{A}$  intercepts the message  $M_3 = \langle ID_{\mathcal{U}_i}, X_{\mathcal{U}_i}, \alpha = Enc_{X'_{\mathcal{U}_i}}[ID_{SN_j} || T_{\mathcal{U}_i}] \rangle$  from the public communication channel established between Step 1 and Step 2 of user authentication and session key establishment phase of our proposed protocol. Sometime later,  $\mathcal{A}$  resends  $M_3$  to the gateway node  $GW_N$ . At the gateway node  $GW_N$ , the message  $M_3$  will be declared as replayed because the time-stamp  $T_{\mathcal{U}_i}$  will not be fresh and the condition  $T' - T_{\mathcal{U}_i} \leq \Delta T$  will

not be satisfied. Similarly, if the adversary  $\mathcal{A}$  intercepts and replays the messages  $M_4$  and  $M_5$  from the public communication channels of user authentication and session key establishment phase, it will be declared (after time-stamp validation) as replayed messages by the sensor node  $SN_j$  and the user  $\mathcal{U}_i$  respectively. Therefore, our protocol is secure against the replay attack.  $\square$

**Proposition 5.** *The proposed protocol is resilient against gateway node capture attack.*

*Proof.* In the registration phase of our proposed protocol, the user  $\mathcal{U}_i$  transmits only the value of  $PB_{\mathcal{U}_i} = h(PW_{\mathcal{U}_i} || \sigma_{\mathcal{U}_i})$ , instead of sending the original biometric information  $BIO_{\mathcal{U}_i}$ , to the gateway node  $GWN$ . Where,  $\sigma_i$  is generated using Fuzzy extractor and the function  $h(\cdot)$  is a secure one-way hash function. Therefore, for an adversary  $\mathcal{A}$ , it is not possible to find out the value of user's password  $PW_{\mathcal{U}_i}$  and biometric information  $BIO_{\mathcal{U}_i}$  from the captured Gateway node  $GWN$ . Then,  $\mathcal{A}$  can not impersonate the user  $\mathcal{U}_i$  on the basis of the authentication phase of our proposed protocol. Hence, our proposed protocol is resilient against gateway node capture attack.  $\square$

### 4.5.2 Formal Security Analysis

In this section, we use random oracle model to perform the formal security analysis of our proposed protocol. Then, we use Scyther tool [19] to verify all the security claims specified in different roles. Afterwards, we automatically validate the safety of our protocol using AVISPA [7] (version v1.1) tool on the basis of Dolev-Yao intruder model with OFMC and CL-AtSe back-ends. We do logical validation using BAN logic to ensure that our protocol works correctly and achieves the specified security features.

#### 4.5.2.1 Formal Security Verification Using Random Oracle Model

The random oracle model (ROM) is a reliable mechanism introduced by Bellare and Rogaway in [23] to make it feasible to produce accurate proofs of security for appropriate fundamental cryptographic protocols. A random oracle is a hypothetical black box that responds to every individual query with a specific random response extracted consistently from its output range. If a query is occurring several times, it responds the same way every time that query is executed. Based on the random oracle model, the following Theorem 1 shows that our protocol can resist various security attacks.

We believe that there exist remarkable random oracles as explained in the following Definitions.

With the help of random oracle model we demonstrate that for an adversary  $\mathcal{A}$  it is not reasonable to obtain the value of legitimate user's identity  $ID_{\mathcal{U}_i}$ , password  $PW_{\mathcal{U}_i}$ , biometric information  $BIO_{\mathcal{U}_i}$ , and the session key  $sk$ . Considering the method of contradiction, we believe that there exist exceptional random oracles as illustrated in following Definitions

**Definition 4.5.1.** *Reveal1: Given a hash value  $y = h(s)$ , this oracle unconditionally outputs the string  $s$ .*

**Definition 4.5.2.** *Reveal2: Given an encrypted value  $Enc_k[s]$ , this oracle unconditionally outputs the string  $s$  without knowing the key  $k$ .*

**Definition 4.5.3.** *Reveal3: Given  $P \in E_p(a, b)$  and the public parameter  $X = r \times P \in E_p(a, b)$ , this oracle outputs the private key  $r$ .*

**Theorem 1.** *If the hash function  $h()$ , encryption mechanism  $Enc$ , and elliptic curve Diffie-Hellman problem  $ECDH$  follows the random oracle Reveal1, Reveal2 and Reveal3 respectively; our protocol resists the adversary  $\mathcal{A}$  for deriving the values of user  $\mathcal{U}_i$ 's secret parameters  $PW_{\mathcal{U}_i}, \sigma_i, K_{\mathcal{U}_i}$  and  $X'_{\mathcal{U}_i}$ .*

*Proof of Theorem.* If we assume that, there exist the oracle Reveal1, Reveal2, Reveal3 which can derive string  $s$  from the hash digest  $d = h(s)$ , string  $s$  from the cipher-text  $Enc_k[s]$  and private key  $r$  from the public parameter  $X = r \times P$  respectively.

Then, the adversary  $\mathcal{A}$  can design a procedure  $EXP_{\mathcal{A}}^{h-Enc-ECDH}$  as shown in Algorithm 4.1.

The adversary designs the Algorithm such that the probability of success of  $EXP_{\mathcal{A}}^{h-Enc-ECDH}$  is  $Success_{\mathcal{A}}^{h-Enc-ECDH} = |P_r[EXP_{\mathcal{A}}^{h-Enc-ECDH} = 1] - 1|$ . The advantage function for  $EXP_{\mathcal{A}}^{h-Enc-ECDH}$  can be represented as:

$$Adv_{\mathcal{A}}^{h-Enc-ECDH}((t_1 + t_2 + t_3), (q_{R_1} + q_{R_2} + q_{R_3})) = [Adv_{\mathcal{A}}^h(t_1) \cdot Adv_{Enc, \mathcal{A}}^{IND-CPA}(t_2) \cdot Adv_{\mathcal{A}}^{ECDH}(t_3)].$$

According to Algorithm 4.1, there exist oracle Reveal1, Reveal2, Reveal3 capable of finding the preimage of  $h()$ , the plain-text  $s$  from the cipher-text  $Enc_k[s]$  and private key  $r$  from the public parameter  $X = r \times P$ .



---

**Algorithm 4.1**  $EX P_{\mathcal{A}}^{h-Enc-ECDH}$ 


---

1. Extract  $\{P, A_{\mathcal{U}_i}, B_{\mathcal{U}_i}, W_{\mathcal{U}_i}, \tau_i, \mathcal{T}, h(), Gen(), Rep()\}$  from  $SC_{\mathcal{U}_i}$  using simple and differential power analysis attacks [14]. Where
  2.  $A_{\mathcal{U}_i} = PB_i \oplus h(ID_{\mathcal{U}_i} \oplus x)$
  3.  $B_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i} || PB_i || h(ID_{\mathcal{U}_i} \oplus x))$ ,
  4.  $W_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i} || PB_i) \oplus K_{\mathcal{U}_i}$ ,
  5. Call **Reveal1** oracle on input  $B_{\mathcal{U}_i}$  to retrieve the information of  $ID_{\mathcal{U}_i}, PB_i, h(ID_{\mathcal{U}_i} || x)$  as  $(ID'_{\mathcal{U}_i} || PB'_i || h(ID_{\mathcal{U}_i} || x)') \leftarrow \text{Reveal1}(B_{\mathcal{U}_i})$
  6. Call **Reveal1** oracle on input  $PB'_i$  to retrieve the information of  $PW_{\mathcal{U}_i}, \sigma_i$  as  $(PW'_{\mathcal{U}_i}, \sigma'_i) \leftarrow \text{Reveal1}(PB'_i)$
  7. Compute  $h(ID'_{\mathcal{U}_i} || PB'_i)$
  8. Compute  $K'_{\mathcal{U}_i} = W_{\mathcal{U}_i} \oplus h(ID'_{\mathcal{U}_i} || PB'_i)$
  9. Intercept the message  $M_3 = \langle ID_{\mathcal{U}_i}, X_{\mathcal{U}_i}, \alpha \rangle$
  10. **if**  $(ID'_{\mathcal{U}_i} = ID_{\mathcal{U}_i})$  **then**

Call **Reveal3** oracle on input  $X_{\mathcal{U}_i}$  to retrieve the private information  $r_{\mathcal{U}_i}$  as  $r_{\mathcal{U}_i} \leftarrow \text{Reveal3}(X_{\mathcal{U}_i})$ ,

Compute the established secret  $X'_{\mathcal{U}_i} = r'_{\mathcal{U}_i} \times K'_{\mathcal{U}_i}$ ,

Call **Reveal2** oracle on input  $\alpha$  to retrieve the information  $ID_{SN_j}, T_{\mathcal{U}_i}$  as  $(ID'_{SN_j} || T'_{\mathcal{U}_i}) \leftarrow \text{Reveal2}$

**if**  $(E_{X'_{\mathcal{U}_i}}[ID'_{SN_j} || T'_{\mathcal{U}_i}] = \alpha)$  **then**

Accept the derived  $ID'_{\mathcal{U}_i}, PW'_i, \sigma'_i$  and  $X'_{\mathcal{U}_i}$  as the correct identity, password, secret biometric data and the established secret information of the user  $\mathcal{U}_i$

Return 1 (Success)

**end**

**else**

Return 0 (Failure)

**end**

**end**

**else**

Return 0 (Failure)

**end**
-

Therefore, the adversary  $\mathcal{A}$  can get the values of  $PW_{\mathcal{U}_i}, \sigma_i, K_{\mathcal{U}_i}, X'_{\mathcal{U}_i}$ . However, according to Definitions 1.3.1–1.3.4 (defined in Chapter 1) we have

$$Adv_{\mathcal{A}}^h(t_1) = Pr[(s, s') \leftarrow_R \mathcal{A} : s \neq s', h(s) = h(s')],$$

$$Adv_{Enc, \mathcal{A}}^{IND-CPA}(t_2) = 2Pr[\mathcal{A} \leftarrow O_k; (b_0, b_1 \leftarrow \mathcal{A}); \tau \leftarrow_R 0, 1; \gamma \leftarrow_R O_k(b_\tau) : \mathcal{A}(\gamma) = \tau] - 1,$$

$$Adv_{\mathcal{A}}^{ECDH}(t_3) = Pr[(r_{\mathcal{U}_i}, P) \leftarrow_R \mathcal{A} : X_{\mathcal{U}_i} = r_{\mathcal{U}_i} \times P]$$

$$\text{Where } Adv_{\mathcal{A}}^{ECDH}(t_1) \leq \tau, Adv_{Enc, \mathcal{A}}^{IND-CPA}(t_2) \leq \tau, Adv_{\mathcal{A}}^{ECDH}(t_3) \leq \tau.$$

$$\text{Therefore, } Adv_{\mathcal{A}}^{h-Enc-ECDH}((t_1 + t_2 + t_3), (q_{R_1} + q_{R_2} + q_{R_3})) \leq \tau.$$

which specifies that  $Adv_{\mathcal{A}}^{h-Enc-ECDH}((t_1 + t_2 + t_3), (q_{R_1} + q_{R_2} + q_{R_3}))$  is negligible for any probabilistic polynomial time adversary  $\mathcal{A}$ . Now, we find that the secure hash function  $h()$ , encryption mechanism  $Enc_k[s]$  and elliptic curve Diffie-Hellman problem  $ECDH$  defined in section 1.3.1 - 1.3.4 (in Chapter 1) contradicts the oracle Reveal1, Reveal2 and Reveal3 respectively considered in Algorithm 4.1. This indicates that our protocol resists the adversary  $\mathcal{A}$  for deriving the values of the secret parameters  $PW_{\mathcal{U}_i}, \sigma_i, K_{\mathcal{U}_i}$ , and  $X'_{\mathcal{U}_i}$ . Hence, the theorem is proved.  $\square$

Where  $q_{R_1}, q_{R_2}, q_{R_3}$  represents the total number of queries made to the Reveal1, Reveal2, Reveal3 oracle respectively.

#### 4.5.2.2 Verification Using Scyther tool

The Scyther tool algorithm provides some novel features, including:

- Guaranteed termination, after which the result is either unbounded correctness, falsification, or bounded correctness.
- Efficient generation of a finite representation of an infinite set of traces concerning patterns, also known as a complete characterization.
- State-of-the-art performance, which has made new types of protocol analysis feasible, such as multi-protocol analysis.

## 4.5 Security Analysis:

The proposed protocol is specified in Security Protocol Description Language (SPDL). The protocol specification defines sequence of roles of  $\mathcal{U}_i$ ,  $GWN$  and  $SN_j$ . Every role incorporates sequence of events (i.e., send, receive, claim and declarations events). The user authentication protocol specification and the roles of  $\mathcal{U}_i$ ,  $GWN$  and  $SN_j$  are demonstrated in Tables 4.9–4.12 respectively. The verification result produced using Scyther tool is shown in Figure 4.2. The result shows that no attacks were found on each of the claims specified in our protocol.

**Table 4.9:** SPDL specification of the proposed ECC & Fuzzy Extractor based protocol.

```

hashfunction h; /*Secure one way hash function */
const Concat: Function; /*Concatenation Function */
const XOR: Function; /* Bitwise XOR operation */

const Enc: Function; /*Encryption Function*/
const Dec: Function; /*Decryption Function*/
const Gen: Function; /*Generator function of Fuzzy Extractor*/
const Rep: Function; /*Reproduction function of Fuzzy Extractor*/
const EccMul: Function; /*Scalar Point Multiplication Operation of ECC */

/*IDui, PWui, BIOi denotes the identity, password and bio-metric information of the user  $U_i$  respectively.
Kgsnj represents the secret key allocated between sensor node and gateway node. Tui, Tgwn represent the
current time-stamp of user, gateway respectively. Rui and Rsnj denotes the random number generated at user
 $U_i$  and sensor node SNj respectively. */

protocol Protocol( $U_i$ , GWN, SNj)
{ macro SIGi = Gen(BIOi); /*macro defines abbreviations for particular term */
macro PBi = h(Concat(PWui, SIGi));
macro SIGi' = Rep(BIOi', TAUi);
macro PBi' = h(Concat(PWui, SIGi'));
macro Kui = EccMul(h(Concat(IDui, x)), P);
macro Aui = XOR(PBi, h(XOR(IDui, x)));
macro Bui = h(Concat(IDui, PBi, h(XOR(IDui, x))));
macro Wui = XOR(h(Concat(IDui, PBi)), Kui);
macro Xui = EccMul(Rui, P);
macro Xui' = EccMul(Rui, Kui);
macro Ysnj = EccMul(Rsnj, P);
macro sk = EccMul(Rui, Ysnj);
macro Alpha = Enc(Xui', Concat(IDsnj, Tui));
macro Alpha' = Dec(Xui', Enc(Xui', Concat(IDsnj, Tui)));
macro Beta = Enc(Xui', Concat(IDsnj, Ysnj, Tgwn));
macro Beta' = Dec(Enc(Xui', Concat(IDsnj, Ysnj, Tgwn)));
macro Gamma = Enc(Kgsnj, Concat(IDui, sk, Beta, Tgwn));
macro Gamma' = Dec(Enc(Kgsnj, Concat(IDui, sk, Beta, Tgwn)));

```

**Table 4.10:** Specification of the user's role in SPDL.

```

role  $U_i$ 
{
  var Tsnj, Tgwn: Nonce;
  fresh  $Tui$ : Nonce; /*Time-stamp  $Tui$  is freshly generated */
  const  $ID_{ui}$ ,  $PW_{ui}$ ,  $BIO_i$ ,  $BIO_i'$ ,  $PBi$ ,  $ID_{snj}$ ,  $R_{ui}$ ,  $R_{snj}$ ,  $K_{gsnj}$ ,  $X_{ui}$ ,  $X_{ui}'$ ,  $x$ ,  $Tui$ ,  $Tgwn$ ,  $P$ ,  $TAU_i$ : Ticket;
  send_1( $U_i$ , GWN,  $ID_{ui}$ ,  $PBi$ ); /* $U_i$  sends  $ID_{ui}$ ,  $PBi$  to GWN */
  recv_2(GWN,  $U_i$ ,  $P$ ,  $A_{ui}$ ,  $B_{ui}$ ,  $W_{ui}$ ); /* $U_i$  received  $P$ ,  $A_{ui}$ ,  $B_{ui}$ ,  $W_{ui}$  from GWN */
  send_3( $U_i$ , GWN,  $X_{ui}$ , Alpha);
  recv_5(SNj,  $U_i$ , Beta);
  match(Beta', Beta); /*Test the equality of Beta' and Beta */
  claim_Ui1( $U_i$ , Secret,  $BIO_i$ ); /* $BIO_i$  should be secret for  $U_i$  */
  claim_Ui2( $U_i$ , Secret,  $PW_{ui}$ );
  claim_Ui3( $U_i$ , Secret,  $x$ );
  claim_Ui4( $U_i$ , Secret,  $X_{ui}'$ );
  claim_Ui5( $U_i$ , Secret,  $Tui$ );
  claim_Ui6( $U_i$ , SKR, sk); /*Session key sk should be secret */
  claim_Ui7( $U_i$ , Niagree); /*Non-injective agreement */
  claim_Ui78( $U_i$ , Nisynch); /*Non-injective synchronization */
}

```

**Table 4.11:** Specification of the gateway node's role in SPDL.

```

role GWN
{
  fresh Tgwn: Nonce;
  var  $Tui$ : Nonce;
  const  $ID_{ui}$ ,  $PW_{ui}$ ,  $ID_{snj}$ ,  $BIO_i$ ,  $P$ ,  $x$ ,  $R_{ui}$ ,  $Tui$ ,  $BIO_i$ ,  $PW_{ui}$ : Ticket;
  recv_1( $U_i$ , GWN,  $ID_{ui}$ ,  $IPBi$ );
  send_2(GWN,  $U_i$ ,  $P$ ,  $A_{ui}$ ,  $B_{ui}$ ,  $W_{ui}$ );
  recv_3( $U_i$ , GWN,  $ID_{ui}$ ,  $ID_{snj}$ ,  $X_{ui}$ ,  $TSui$ , Alpha);
  match (Alpha, Alpha');
  send_4(GWN, SNj, Beta, Gamma,  $X_{ui}$ ,  $TG_{gwn}$ ,  $TU_{gwn}$ );
  claim_GWN1(GWN, Secret, Tgwn);
  claim_GWN2(GWN, Secret, x);
  claim_GWN3(GWN, Secret, k(GWN, SNj));
  claim_GWN4(GWN, Secret,  $K_{ui}$ );
  claim_GWN5(GWN, Secret,  $X_{ui}'$ );
}

```

## 4.5 Security Analysis:

**Table 4.12:** Specification of the sensor's role in SPDL.

```

role SNj
{
  var Tgwn: Nonce;
  fresh Tsnj: Nonce;
  const IDui, IDsnj, x, Rui, Tui, P, BIOi, PWui, Rsnj: Ticket;
  recv_4(GWN, SNj, Beta, Gamma, Xui, TGwn, TUGwn);
  match(Beta, Beta');
  send_5(SNj, Ui, Delta, Gamma, Ysnj, Tsnj, TUGwn);
  claim_SNj1(SNj, Secret, Tgwn);
  claim_SNj2(SNj, Secret, Rsnj);
  claim_SNj3(SNj, Secret, Tsnj);
  claim_SNj4(SNj, Secret, k(GWN, SNj));
  claim_SNj5(SNj, SKR, h(EccMul(Rsnj, Xui)));
}

```

Scyther results : verify

Claim				Status	Comments
Protocol	Ui	Protocol,Ui1	Secret Bi	Ok	No attacks within bounds.
		Protocol,Ui2	Secret PWui	Ok	No attacks within bounds.
		Protocol,Ui3	Secret x	Ok	No attacks within bounds.
		Protocol,Ui4	Secret $\text{EccMul}(\text{Rui}, \text{EccMul}(\text{h}(\text{Concat}(\text{IDui}, \text{x})), \text{P}))$	Ok	No attacks within bounds.
		Protocol,Ui5	Secret Tui	Ok	No attacks within bounds.
		Protocol,Ui6	SKR $\text{EccMul}(\text{Rui}, \text{EccMul}(\text{Rsnj}, \text{P}))$	Ok	No attacks within bounds.
		Protocol,Ui7	Niagree	Ok	No attacks within bounds.
		Protocol,Ui78	Nisynch	Ok	No attacks within bounds.
GWN		Protocol,GWN1	Secret x	Ok	No attacks within bounds.
		Protocol,GWN2	Secret Rsnj	Ok	No attacks within bounds.
		Protocol,GWN3	Secret Tgwn	Ok	No attacks within bounds.
		Protocol,GWN4	Secret Kgsnj	Ok	No attacks within bounds.
SNj		Protocol,SNj1	Secret Rsnj	Ok	No attacks within bounds.
		Protocol,SNj2	SKR $\text{EccMul}(\text{Rui}, \text{EccMul}(\text{Rsnj}, \text{P}))$	Ok	No attacks within bounds.
		Protocol,SNj3	Secret Kgsnj	Ok	No attacks within bounds.

Done.

**Figure 4.2:** Security validation result obtained using Scyther tool.

### 4.5.3 Verification Using AVISPA Tool

In this section, we first describe the setup methodology and some basic characteristics of AVISPA tool which we apply for the formal security analysis of our proposed protocol. Afterwards, we describe the implementation of our protocol using High-Level Protocol Specification Language (HLPSL). Finally, we discuss about the results obtained.

#### 4.5.3.1 Experimental Setup and the Size of the Entities Involved in WSNs & IoT for the Simulation of Proposed Protocol Using AVISPA Tool

To simulate the proposed protocol on AVISPA v1.1, we utilize a Security Protocol ANimator (SPAN) Version 1.6 on a PC framework having ubuntu 16.04 LTS working framework (64 bit), Intel (R) core (TM) i7-6500U CPU @ 2.50 GHz x4 processor, and 8 GB RAM. We extract the archive avispac-package-1.1.Linux-i686.tgz, set up the environment variable AVISPA\_PACKAGE and keep the script of the avispac protocol in the execution path. We implemented our protocol considering minimal number of entities involved in WSNs & IoT (i.e, one user  $\mathcal{U}_i$ , one sensor node  $SN_j$  and one gateway node  $GWN$ ) using Dolev-Yao model [110] with a bounded number of sessions, specified goal, On-the-Fly Model-Checker (OFMC) and Constraint-Logic based Attack Searcher (CL-AtSe) backend.

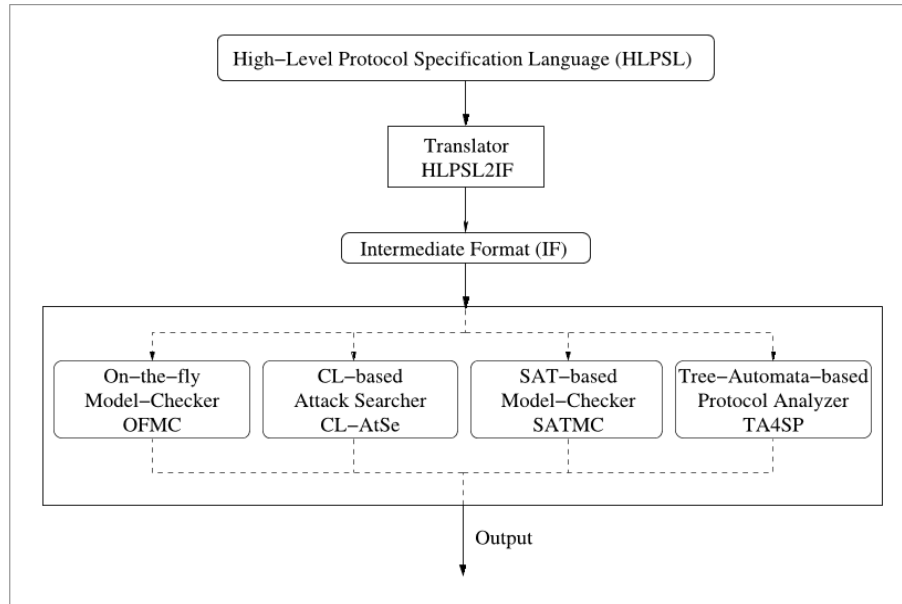
#### 4.5.3.2 Basic Features of AVISPA Tool

AVISPA is an extensively acknowledged and strong programming tool for consequently approving (utilizing push-catch system) the security highlights of the conventions utilized in Internet of Things. The engineering of AVISPA tool is appeared in following Figure4.3.

AVISPA requires HLPSL to define the security protocol in a file with.hlpal extension. It carry out a static security analysis to check the executability of the protocol. A HLPSL2IF translator is used to translate the HLPSL specification into an Intermediate Formate (IF) specification, which is tool-independent language and compatible for automated deduction. The IF specifications are provided as an input to one of the four back-ends. The back-ends are as follows:

1. Constraint-logic based attack searcher (CL-AtSe)
2. On-the-fly model-checker (OFMC)
3. SAT-based model-checker (SATMC)
4. Tree automata based on automatic approximation for the analysis of security protocols (TA4SP).

**Figure 4.3:** AVISPA Architecture [7].



### 4.5.4 Implementation of the Proposed Protocol Using HLP SL

The HLP SL specification of the protocol consists of the following module:

1. **Basic Role:** Basic role illustrates the activity of the entities (e.g., User  $U_i$ , Gateway  $GW_N$  and Sensor node  $SN_j$ ) involve in the protocol.
  - Each role may have some parameters like  $U_i$ ,  $GW_N$ ,  $SN_j$  of type agent and  $K_{ui1}$ ,  $K_{gsnj}$  of type symmetric\_key.
  - The parameter RCV and SND denotes the agent's communication channels for receiving and sending the information.

- The parameter (dy) represents the Dolev-Yao intruder model for the channel.
- The function H, Gen, Rep, EccMul, Enc, Dec and XOR are corresponding to the hash function, fuzzy extractor's generator, fuzzy extractor's reproduction, elliptic curve scalar multiplication, encryption, decryption and logical XOR operations respectively.
- We have used the term hash\_func which constitutes all the cryptography functions which are not easily invertible, because the random non-invertible arithmetic operators are not supportable in HLPSL.
- The term "played\_by  $Ui$ " denotes that the role "User" is played by  $Ui$ .

The HLPSL specification of roles of  $U_i$ ,  $GW_N$  and  $SN_j$  are shown in Tables 4.13–4.15 respectively.

2. **Transitions:** The transitions are expressed in States. It involves trigger that fires at the time of event occurrence. For every States during a transition if a message accepted on channel RCV, then transition fires and assigns a new value to the State.
3. **Composed Roles:** It models one or more basic roles to execute together and represent the sessions involve in the protocol. The operator  $\wedge$  represents the parallel execution of the roles.

The HLPSL specification of proposed protocol's session is shown in Table 4.16.

4. **Environment:** It consists of global constant and session composition, where the adversary may execute some role as a authorized user.

The HLPSL specification of proposed protocol's environment is shown in Table 4.17.

5. **Security Goal:** This module specifies the security protection Goal of the protocol. Some necessary predicates utilized in this module are as follows:
  - $\text{secret}(\{PW_{ui}, BIO_i, SIG_i'\}, \text{sub1}, Ui)$ : It indicates that the information  $\{PW_{ui}, BIO_i, SIG_i'\}$  is secretly shared to  $Ui$  and it can be recognize with a constant identity  $\text{sub1}$  in goal module.



- $witness(U_i, GWN, gateway\_user\_gu, T_{ui}, \text{Alpha}')$ : It represents the weak authenticity of  $U_i$  by  $GWN$  and  $U_i$  is the witness for the data  $\{T_{ui}', \text{Alpha}'\}$ . The identity of this goal is represented as  $gateway\_user\_gu$  in goal module.
- $request(U_i, SN_j, user\_sensor\_us, Skey')$ : It represents the strong authenticity of  $U_i$  by  $SN_j$  on  $Skey$  with an identity  $user\_sensor\_us$ .
- Symbols: Concatenation ( $.$ ) is used for message composition (e.g.,  $SND(ID_i.PBi')$ ) and Commas ( $,$ ) is used in case of multiple arguments of events or functions (e.g.,  $secret(PW_{ui}, BIO_i, SIG_i', sub1, U_i)$ ).

### 4.5.5 Description of the Output Format Generated by AVISPA Tool

The output generated by AVISPA tool describes the final result obtained under various conditions after the security analysis of the protocol. The output produced by the AVISPA tool consist of following modules:

- **Summary:** This module specifies the security reliability of the protocol regarding safe, unsafe or inconclusive.
- **Details:** In this portion, the output specifies the environment and the context under which the protocol is claimed to be safe, unsafe or inconclusive.
- **Protocol:** It indicates the name of the protocol given as an input for security verification.
- **Goal:** This module represents the specified security goal of the protocol.
- **Backend:** This module represents one of the four back-ends used for the analysis of the protocol.

The verification result of AVISPA [7] tool is shown in Table 4.19 which represents that the proposed protocol is safe from various attacks (like man-in-the-middle attack, replay attack etc.) using Dolev-Yao model [110] with bounded number of sessions, specified goal, On-the-Fly Model-Checker (OFMC) and Constraint-Logic based Attack Searcher (CL-AtSe) backend.

**Table 4.13:** Specification of  $U_i$ 's role in HLPSL.

```

role user( $U_i$ , GWN, SNj: agent,
 $X_{ui}1$ , Kgsnj: symmetric_key,
H, Gen, Rep, EccMul, Enc, Dec, XOR: hash_func,
SND, RCV: channel(dy))
played_by  $U_i$  def=
local
State: nat,
ID $_{ui}$ , IDsnj, PW $_{ui}$ , BIOi, BIOi1, SIGi, SIGi1, TAU $_i$ , PBi, PBi1, P, K $_{ui}1$ , R $_{ui}$ , A $_{ui}$ ,
B $_{ui}$ , W $_{ui}$ , Alpha, Beta, Gamma, Ysnj, Ysnj1, T $_{ui}$ , Tgwn, X $_{ui}$ , X, Beta1, K $_{ui}$ , Rsnj,
Gamma1, Skey, Skey1: text
const sub1, sub2, sub3, sub4, sub5, sub6, sub7, sub8,
gateway_sensor_gs, gateway_user_gu, user_sensor_us: protocol_id
init
State: = 0
transition
0. State = 0  $\wedge$  RCV (start) =  $\triangleright$ 
State': = 2  $\wedge$  SIGi': = Gen(BIOi)  $\wedge$  PBi': = H(PW $_{ui}$ .SIGi')
 $\wedge$  secret(PW $_{ui}$ , BIOi, SIGi', sub1,  $U_i$ )
 $\wedge$  SND (ID $_{ui}$ .PBi')
2. State = 2  $\wedge$  RCV (P.A $_{ui}'$ .B $_{ui}'$ .W $_{ui}'$ ) =  $\triangleright$ 
State': = 5  $\wedge$  R $_{ui}'$ : = new()
 $\wedge$  T $_{ui}'$ : = new()
 $\wedge$  secret(R $_{ui}'$ , sub2,  $U_i$ )
 $\wedge$  SIGi1': = Rep(BIOi1.TAU $_i$ )  $\wedge$  PBi1': = H(PW $_{ui}$ .SIGi1')
 $\wedge$  K $_{ui}1'$ : = XOR(W $_{ui}$ , H(ID $_{ui}$ .PBi1'))
 $\wedge$  X $_{ui}'$ : = EccMul(R $_{ui}'$ .P)
 $\wedge$  X $_{ui}1'$ : = EccMul(R $_{ui}'$ .K $_{ui}1'$ )
 $\wedge$  secret(X $_{ui}1'$ , sub3,  $U_i$ , GWN)
 $\wedge$  Alpha': = Enc(IDsnj.T $_{ui}$ )
 $\wedge$  SND(ID $_{ui}$ .X $_{ui}'$ .Alpha')
 $\wedge$  witness( $U_i$ , GWN, gateway_user_gu, T $_{ui}$ , Alpha')
6. State = 5  $\wedge$  RCV(Beta1') =  $\triangleright$ 
State': = 6  $\wedge$  Ysnj1': = Dec(Beta1')  $\wedge$  Skey': = EccMul(R $_{ui}'$ .Ysnj1')
 $\wedge$  request( $U_i$ , SNj, user_sensor_us, Skey')
end role

```

**Table 4.14:** Specification of *GW<sub>N</sub>*'s role in HLP<sub>SL</sub>.

```

role gateway(Ui, GWN, SNj: agent,
Xui1, Kgsnj: symmetric_key,
H, Gen, Rep, EccMul, Enc, Dec, XOR: hash_func,
SND, RCV: channel(dy))
played_by GWN def=
local
State: nat,
IDui, IDsnj, PWui, BIOi, BIOi1, SIGi, SIGi1, TAUi, PBi, PBi1, P, Kui1, Rui, Aui,
Bui, Wui, Alpha, Beta, Gamma, Ysnj, Tui, Tgwn, Xui, X, Beta1, Kui, Rsnj, Gamma1,
Skey, Skey1: text
const sub1, sub2, sub3, sub4, sub5, sub6, sub7, sub8,
gateway_sensor_gs, gateway_user_gu, user_sensor_us: protocol_id
init
State: = 1
transition
1. State = 1 ∧ RCV (IDui.PBi') = ▷
State': = 2 ∧ X': = new()
    ∧ Kui': = EccMul(H(IDui.X').P)
    ∧ Aui': = XOR(PBi'.H(XOR(IDui.X')))
    ∧ Bui': = H(IDui.PBi'.XOR(IDui.X'))
    ∧ secret(X', sub4, GWN)
    ∧ Wui': = XOR(H(IDui.PBi).Kui')
    ∧ secret(Kui', sub5, GWN, Ui)
    ∧ SND(P.Aui'.Bui'.Wui')
3. State = 3 ∧ RCV(IDui.Xui'.Alpha') = ▷
State': = 4 ∧ Tgwn': = new()
    ∧ request(GWN, Ui, gateway_user_gu, Alpha')
    ∧ IDsnj': = Dec(Alpha') ∧ Rsnj': = new()
    ∧ Ysnj': = EccMul(Rsnj'.P)
    ∧ Beta': = Enc(IDsnj'.Ysnj'.Tgwn)
    ∧ secret(Kgsnj, sub6, GWN, SNj)
    ∧ Gamma': = Enc(IDui.Skey'.Beta'.Tgwn')
    ∧ SND(Gamma')
    ∧ witness(GWN, Ui, gateway_user_gu, Tgwn')
end role

```

## 4.5 Security Analysis:

**Table 4.15:** Specification of  $SN_j$ 's role in HLPSSL.

```

role sensor( $U_i$ , GWN,  $SN_j$ : agent,
 $X_{ui1}$ ,  $K_{gsnj}$ : symmetric_key,
H,Gen, Rep, EccMul, Enc, Dec, XOR: hash_func,
SND, RCV: channel(dy))
played_by  $SN_j$  def=
local
State: nat,
ID $_{ui}$ , ID $_{snj}$ , PW $_{ui}$ , BIO $_{i1}$ , BIO $_{i1}$ , SIG $_{i1}$ , SIG $_{i1}$ , TAU $_{i1}$ , PBi, PBi1, P,  $K_{ui1}$ ,  $R_{ui}$ ,  $A_{ui}$ ,  $B_{ui}$ ,  $W_{ui}$ , Alpha, Beta, Gamma,  $Y_{snj}$ ,  $T_{ui}$ ,
Tgwn,  $X_{ui}$ , X, Beta1,  $K_{ui}$ ,  $R_{snj}$ , Gamma1, Skey, Skey1: text
const sub1, sub2, sub3, sub4, sub5, sub6, sub7, sub8,
gateway_sensor_gs, gateway_user_gu, user_sensor_us: protocol_id
init
State: = 4
transition
4. State = 4  $\wedge$  RCV (Gamma') =  $\triangleright$ 
State': = 5  $\wedge$  Skey1': = Dec(Gamma'. $K_{gsnj}$ )
     $\wedge$  secret(Skey1', sub7,  $SN_j$ )
     $\wedge$  Beta1': = Dec(Gamma')
     $\wedge$  secret(Skey1', sub8,  $SN_j$ )
     $\wedge$  SND(Beta1')
end role

```

**Table 4.16:** Specification of session of the proposed protocol in HLPSSL.

```

role session( $U_i$ , GWN,  $SN_j$ : agent,
 $X_{ui1}$ ,  $K_{gsnj}$ : symmetric_key,
H,Gen, Rep, EccMul, Enc, Dec, XOR: hash_func)
def=
local GWN $U_i$ ,  $R_{U_i}$ , GWN $SN_j$ ,  $R_{SN_j}$ , GWN $GWN$ ,  $R_{GWN}$ : channel(dy)
composition
    user( $U_i$ , GWN,  $SN_j$ ,  $X_{ui1}$ ,  $K_{gsnj}$ , H,Gen, Rep, EccMul, Enc, Dec, XOR, GWN $U_i$ ,  $R_{U_i}$ )
     $\wedge$  sensor( $U_i$ , GWN,  $SN_j$ ,  $X_{ui1}$ ,  $K_{gsnj}$ , H,Gen, Rep, EccMul, Enc, Dec, XOR, GWN $SN_j$ ,  $R_{SN_j}$ )
     $\wedge$  gateway( $U_i$ , GWN,  $SN_j$ ,  $X_{ui1}$ ,  $K_{gsnj}$ , H,Gen, Rep, EccMul, Enc, Dec, XOR, GWN $GWN$ ,  $R_{GWN}$ )
end role

```

**Table 4.17:** Specification of environment of the proposed protocol in HLPSSL.

<pre> role environment() def= const ui, gwn, snj: agent, xui1,kgsnj,kig: symmetric_key, h,gen, rep, eccMul, enc, dec, xOR: hash_func, sub1, sub2, sub3, sub4, sub5, sub6, sub7, sub8, gateway_sensor_gs, gateway_user_gu, user_sensor_us: protocol_id intruder_knowledge = ui,gwn,snj,kig composition session(ui,snj,gwn,xui1,kig,h,gen, rep, eccMul, enc, dec, xOR)     ∧ session(ui,snj,gwn,kgsnj,kig,h,gen, rep, eccMul, enc, dec, xOR)     ∧ session(ui,snj,gwn,kig,kgsnj,h,gen, rep, eccMul, enc, dec, xOR) end role </pre>
--

**Table 4.18:** Specification of goal of the proposed protocol in HLPSSL.

<pre> goal secrecy_of sub1, sub2, sub3, sub4, sub5, sub6, sub7, sub8 authentication_on gateway_sensor_gs, gateway_user_gu, user_sensor_us end goal environment() </pre>
---

### 4.5.5.1 Logical Verification Using BAN Logic

In this module, we use BAN logic [20] to verify the freshness of time-stamp to avoid replay attack and we validate the message origin to achieve authenticity.

The notation we use for logical verification is shown in Table 1.

**Rule 1** Message meaning rule:  $\frac{P_{BAN}|\equiv P_{BAN} \xrightarrow{K} Q_{BAN}, P_{BAN} \triangleleft \{S\}_K}{P_{BAN}|\equiv Q_{BAN} \sim S}$ . It denotes that, if  $P_{BAN}$  believes that she shared the key  $K$  with  $Q_{BAN}$ , and  $P_{BAN}$  sees the message  $\{S\}$  encrypted with key  $K$ ,  $P_{BAN}$  believes that  $Q_{BAN}$  once said  $S$ .

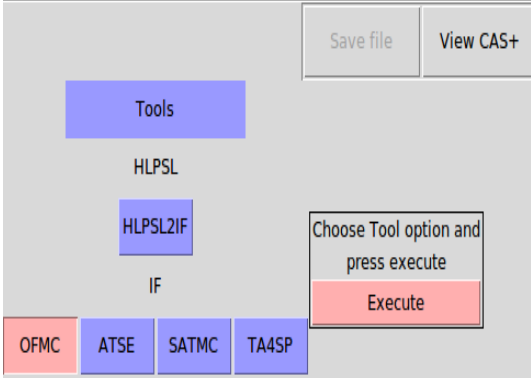
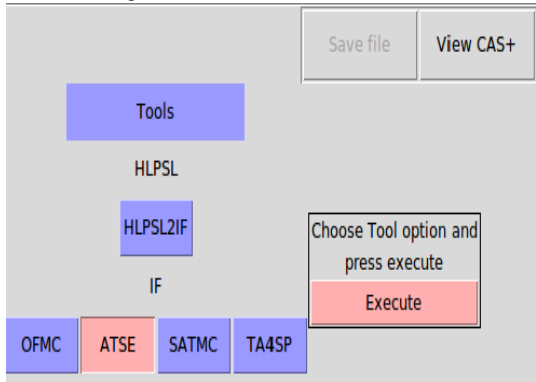
**Rule 2** Nonce verification rule:  $\frac{P_{BAN}|\equiv \#(S), P_{BAN}|\equiv Q_{BAN} \sim S}{P_{BAN}|\equiv Q_{BAN} \equiv S}$ . It denotes that, if  $P_{BAN}$  believes  $S$  is fresh and  $Q_{BAN}$  once said  $S$ ,  $P_{BAN}$  believes  $Q_{BAN}$  believes  $S$ .

**Rule 3** Jurisdiction rule:  $\frac{P_{BAN}|\equiv Q_{BAN} \Rightarrow S, P_{BAN}|\equiv Q_{BAN} \equiv S}{P_{BAN}|\equiv S}$ . It denotes that, if  $P_{BAN}$  believes that  $Q_{BAN}$  had jurisdiction right to  $S$  and believes  $Q_{BAN}$  believes  $S$ ,  $P_{BAN}$  believes  $S$ .

In order to achieve better security features, the proposed protocol should achieve the security Goals as defined in Table 4.20.

## 4.5 Security Analysis:

**Table 4.19:** Security verification result obtained using AVISPA tool.

Using OFMC BACKEND	Using CL-AtSe BACKEND
<p>SUMMARY SAFE</p> <p>DETAILS BOUNDED_NUMBER_OF_SESSIONS</p> <p>PROTOCOL /home/cmb-lab-22/Desktop/Proto.if</p> <p>GOAL as_specified</p> <p>BACKEND OFMC</p> <p>STATISTICS Time: 984 ms parseTime: 0 ms visitedNodes: 456 nodes depth: 9 piles</p> 	<p>SUMMARY SAFE</p> <p>DETAILS BOUNDED_NUMBER_OF_SESSIONS TYPED_MODEL</p> <p>PROTOCOL /home/cmb-lab-22/Desktop/Proto.if</p> <p>GOAL as_specified</p> <p>BACKEND CL-AtSe</p> <p>STATISTICS Analysed: 1956 states Reachable: 1956 states Translation: 0.06 s Computation: 0.01 s</p> 

**Table 4.20:** Goals: The goals made to analyze the proposed protocol.

<p><b>Goal 1</b> <math>\mathcal{U}_i   \equiv K_{\mathcal{U}_i}</math></p> <p><b>Goal 2</b> <math>SN_j   \equiv K_{GSN_j}</math></p> <p><b>Goal 3</b> <math>GWN   \equiv \mathcal{U}_i   \equiv T_{\mathcal{U}_i}</math></p> <p><b>Goal 4</b> <math>SN_j   \equiv GWN   \equiv T_{GWN}</math></p>	<p><b>Goal 5</b> <math>\mathcal{U}_i   \equiv GWN   \equiv T_{GWN}</math></p> <p><b>Goal 6</b> <math>GWN   \equiv \mathcal{U}_i   \sim ID_{SN_j}</math></p> <p><b>Goal 7</b> <math>SN_j   \equiv GWN   \sim ID_{\mathcal{U}_i}</math></p> <p><b>Goal 8</b> <math>\mathcal{U}_i   \equiv GWN   \sim Y_{SN_j}</math></p>
---	--

## 4.5 Security Analysis:

**Message 1**  $\mathcal{U}_i \rightarrow GWN : ID_{\mathcal{U}_i}, X_{\mathcal{U}_i}, \langle ID_{SN_j} || T_{\mathcal{U}_i} \rangle_{X'_{\mathcal{U}_i}}$

**Message 2**  $GWN \rightarrow SN_j : \left\langle ID_{\mathcal{U}_i} || sk || \left\langle ID_{SN_j} || Y_{SN_j} || T_{GWN} \right\rangle_{X'_{\mathcal{U}_i}} || T_{GWN} \right\rangle_{K_{GSN_j}}$ ,

**Message 3**  $SN_j \rightarrow \mathcal{U}_i : \langle ID_{SN_j} || Y_{SN_j} || T_{GWN} \rangle_{X'_{\mathcal{U}_i}}$

Hypotheses: Some significant assumptions about the initial state are made (as shown in Table 4.21) to analyze security features of the proposed protocol.

Now, on the basis of the hypothesis as described in Table 4.21 and the rules of the BAN logic, we validate that the proposed protocol can accomplish the intended goals and the explanations are as follows:

**Table 4.21:** Hypotheses: The assumptions made to analyze the proposed protocol.

<p><b>H 1:</b> <math>\mathcal{U}_i   \equiv \#T_{\mathcal{U}_i}</math></p> <p><b>H 2:</b> <math>GWN   \equiv \#T_{GWN}</math></p> <p><b>H 3:</b> <math>SN_j   \equiv \#T_{SN_j}</math></p> <p><b>H 4:</b> <math>\mathcal{U}_i   \equiv GWN \Rightarrow K_{\mathcal{U}_i}</math></p> <p><b>H 5:</b> <math>\mathcal{U}_i   \equiv GWN   \equiv K_{\mathcal{U}_i}</math></p> <p><b>H 6:</b> <math>SN_j   \equiv GWN \Rightarrow K_{GSN_i}</math>,</p> <p><b>H 7:</b> <math>SN_j   \equiv GWN   \equiv K_{GSN_j}</math></p> <p><b>H 8:</b> <math>GWN   \equiv \mathcal{U}_i \stackrel{X'_{\mathcal{U}_i}}{\rightleftharpoons} GWN</math>,</p> <p><b>H 9:</b> <math>GWN \triangleleft \langle T_{\mathcal{U}_i} \rangle_{X'_{\mathcal{U}_i}}</math></p> <p><b>H 10:</b> <math>GWN   \equiv \#(T_{\mathcal{U}_i})</math></p> <p><b>H 11:</b> <math>SN_j   \equiv GWN \stackrel{K_{GSN_j}}{\rightleftharpoons} SN_j</math>,</p> <p><b>H 12:</b> <math>SN_j \triangleleft \langle T_{GWN} \rangle_{K_{GSN_j}}</math></p>	<p><b>H 13:</b> <math>SN_j   \equiv \#(T_{GWN})</math></p> <p><b>H 14:</b> <math>\mathcal{U}_i   \equiv GWN \stackrel{X'_{\mathcal{U}_i}}{\rightleftharpoons} GWN</math>,</p> <p><b>H 15:</b> <math>\mathcal{U}_i \triangleleft \langle T_{GWN} \rangle_{X'_{\mathcal{U}_i}}</math></p> <p><b>H 16:</b> <math>\mathcal{U}_i   \equiv \#(T_{GWN})</math></p> <p><b>H 17:</b> <math>GWN   \equiv \mathcal{U}_i \stackrel{X'_{\mathcal{U}_i}}{\rightleftharpoons} GWN</math>,</p> <p><b>H 18:</b> <math>GWN \triangleleft \langle ID_{SN_j} \rangle_{X'_{\mathcal{U}_i}}</math></p> <p><b>H 19:</b> <math>SN_j   \equiv GWN \stackrel{K_{GSN_j}}{\rightleftharpoons} SN_j</math>,</p> <p><b>H 20:</b> <math>SN_j \triangleleft \langle ID_{\mathcal{U}_i} \rangle_{K_{GSN_j}}</math></p> <p><b>H 21:</b> <math>\mathcal{U}_i   \equiv GWN \stackrel{X'_{\mathcal{U}_i}}{\rightleftharpoons} \mathcal{U}_i</math></p> <p><b>H 22:</b> <math>\mathcal{U}_i \triangleleft \langle Y_{SN_j} \rangle_{X'_{\mathcal{U}_i}}</math></p>
--	--

1. Derivation of user  $\mathcal{U}_i$ 's trusts on the truth of secret information  $K_{\mathcal{U}_i}$ .

$$\bullet \quad \frac{U_i | \equiv GWN \Rightarrow K_{U_i}, U_i | \equiv GWN | \equiv K_{U_i}}{U_i | \equiv K_{U_i}}$$

It denotes that, if  $U_i$  believes that  $GWN$  has jurisdiction over  $K_{U_i}$  then  $U_i$  trusts  $GWN$  on the truth of  $K_{U_i}$ . Therefore, we achieve Goal 1.

2. Derivation of sensor node  $SN_j$ 's trusts on the truth of secret information  $K_{GSN_j}$ .

$$\bullet \quad \frac{SN_j | \equiv GWN \Rightarrow K_{GSN_j}, SN_j | \equiv GWN | \equiv K_{GSN_j}}{SN_j | \equiv K_{GSN_j}}$$

It denotes that, if sensor node  $SN_j$  believes that the gateway node  $GWN$  has jurisdiction over  $K_{GSN_j}$  then  $SN_j$  trusts  $GWN$  on the truth of  $K_{GSN_j}$ . Therefore, we achieve Goal 2.

3. Validation of freshness of user's time-stamp  $T_{U_i}$  on the gateway node  $GWN$  (using message-meaning and nonce verification rule):

$$\bullet \quad \frac{GWN | \equiv U_i \stackrel{X'_{U_i}}{\rightleftharpoons} GWN, GWN \triangleleft \langle T_{U_i} \rangle_{X'_{U_i}}}{GWN | \equiv U_i | \sim T_{U_i}} \quad (\text{on the basis of message-meaning rule})$$

It denotes that, if  $GWN$  believes the secret  $X'_{U_i}$  is shared with  $U_i$  and sees  $< T_{U_i} >_{X'_{U_i}}$ , then  $GWN$  believes  $U_i$  once said  $T_{U_i}$ .

$$\bullet \quad \frac{GWN | \equiv \#(T_{U_i}), GWN | \equiv U_i | \sim T_{U_i}}{GWN | \equiv U_i | \equiv T_{U_i}} \quad (\text{on the basis of nonce verification rule})$$

It denotes that, if  $GWN$  believes that the time-stamp  $T_{U_i}$  is fresh and  $U_i$  once said  $T_{U_i}$ , then  $GWN$  believes  $U_i$  believes  $T_{U_i}$ . Therefore, we achieve Goal 3.

4. Validation of freshness of gateway node's time-stamp  $T_{GWN}$  on the sensor node  $SN_j$  (using message-meaning and nonce verification rule):



## 4.5 Security Analysis:

- $$\frac{SN_j | \equiv GWN \xrightarrow{K_{GSN_j}} SN_j, SN_j \triangleleft \langle T_{GWN} \rangle_{K_{GSN_j}}}{SN_j | \equiv GWN | \sim T_{GWN}} \quad (\text{Based on message-meaning rule})$$

It denotes that, if  $SN_j$  believes the secret  $K_{GSN_j}$  is shared with  $GWN$  and sees  $\triangleleft T_{GWN} \triangleright_{K_{GSN_j}}$ , then  $SN_j$  believes  $GWN$  once said  $T_{GWN}$ .

- $$\frac{SN_j | \equiv \#(T_{GWN}), SN_j | \equiv GWN | \sim T_{GWN}}{SN_j | \equiv GWN | \equiv T_{GWN}} \quad (\text{on the basis of nonce-verification rule})$$

It denotes that, if  $SN_j$  believes that the time-stamp  $T_{GWN}$  is fresh and  $GWN$  once said  $T_{GWN}$ , then  $SN_j$  believes  $GWN$  believes  $T_{GWN}$ . Therefore, we achieve Goal 4.

5. Validation of freshness of gateway node's time-stamp  $T_{GWN}$  on user  $\mathcal{U}_i$  (using message-meaning and nonce verification rule):

- $$\frac{\mathcal{U}_i | \equiv GWN \xrightarrow{X'_{\mathcal{U}_i}} GWN, \mathcal{U}_i \triangleleft \langle T_{GWN} \rangle_{X'_{\mathcal{U}_i}}}{\mathcal{U}_i | \equiv GWN | \sim T_{GWN}} \quad (\text{on the basis of message-meaning rule})$$

It denotes that, if  $\mathcal{U}_i$  believes the secret  $X'_{\mathcal{U}_i}$  is shared with  $GWN$  and sees  $\triangleleft T_{GWN} \triangleright_{X'_{\mathcal{U}_i}}$ , then  $\mathcal{U}_i$  believes  $GWN$  once said  $T_{GWN}$ .

- $$\frac{\mathcal{U}_i | \equiv \#(T_{GWN}), \mathcal{U}_i | \equiv GWN | \sim T_{GWN}}{\mathcal{U}_i | \equiv GWN | \equiv T_{GWN}} \quad (\text{on the basis of nonce-verification rule})$$

It denotes that, if  $\mathcal{U}_i$  believes that the time-stamp  $T_{GWN}$  is fresh and  $GWN$  once said  $T_{GWN}$ , then  $\mathcal{U}_i$  believes  $GWN$  believes  $T_{GWN}$ . Therefore, we achieve Goal 5.

6. Validation of sensor node's identity  $ID_{SN_j}$  on the gateway node  $GW_N$ :

$$\bullet \frac{GW_N | \equiv \mathcal{U}_i \stackrel{X'_{\mathcal{U}_i}}{\rightleftharpoons} GW_N, GW_N \triangleleft \langle ID_{SN_j} \rangle_{X'_{\mathcal{U}_i}}}{GW_N | \equiv \mathcal{U}_i | \sim ID_{SN_j}} \text{ (on the basis of message-meaning rule)}$$

It denotes that, if  $GW_N$  believes the secret  $X'_{\mathcal{U}_i}$  is shared with  $\mathcal{U}_i$  and sees  $\langle ID_{SN_j} \rangle_{X'_{\mathcal{U}_i}}$ , then  $GW_N$  believes  $\mathcal{U}_i$  once said  $ID_{SN_j}$ . Therefore, we achieve Goal 6.

7. Validation of user's identity  $ID_{\mathcal{U}_i}$  on the sensor node  $SN_j$ :

$$\bullet \frac{SN_j | \equiv GW_N \stackrel{K_{GSN_j}}{\rightleftharpoons} SN_j, SN_j \triangleleft \langle ID_{\mathcal{U}_i} \rangle_{K_{GSN_j}}}{SN_j | \equiv GW_N | \sim ID_{\mathcal{U}_i}} \text{ (on the basis of message-meaning rule)}$$

It denotes that, if  $SN_j$  believes the secret  $K_{GSN_j}$  is shared with  $GW_N$  and sees  $\langle ID_{\mathcal{U}_i} \rangle_{K_{GSN_j}}$ , then  $SN_j$  believes  $GW_N$  once said  $ID_{\mathcal{U}_i}$ . Therefore, we achieve Goal 7.

8. Validation of the public key  $Y_{SN_j}$  by user  $\mathcal{U}_i$ :

$$\bullet \frac{\mathcal{U}_i | \equiv GW_N \stackrel{X'_{\mathcal{U}_i}}{\rightleftharpoons} \mathcal{U}_i, \mathcal{U}_i \triangleleft \langle Y_{SN_j} \rangle_{X'_{\mathcal{U}_i}}}{\mathcal{U}_i | \equiv GW_N | \sim Y_{SN_j}} \text{ (on the basis of message-meaning rule)}$$

It denotes that, if  $\mathcal{U}_i$  believes the secret  $X'_{\mathcal{U}_i}$  is shared with  $GW_N$  and sees  $\langle Y_{SN_j} \rangle_{X'_{\mathcal{U}_i}}$ , then  $\mathcal{U}_i$  believes  $GW_N$  once said  $Y_{SN_j}$ . Therefore, we achieve Goal 8.

## 4.6 Comparative Analysis based on Security Features and Computational Overhead

### 4.6.1 Relative Security Analysis

Our comparative analysis of security features is based the popular features which need to be considered and the resistant against well-known attacks. Table 4.22 shows that our protocol overcomes the major attacks and provides more security.

**Table 4.22:** Comparison of protocols on the basis of security features.

Security Feature	A.K.Das [81]	Choi et al. [61]	Park et al. [63]	Moon et al. [64]	Proposed Protocol
Resist stolen smart card attack	No	No	No	No	Yes
Resists Replay attack	Yes	Yes	No	Yes	Yes
Resists Man-in-the-middle attack	No	No	No	Yes	Yes
Resists user impersonation attack	No	No	No	Yes	Yes
Resists sensor impersonation attack	No	No	No	Yes	Yes
Resists insider attack	Yes	Yes	Yes	Yes	Yes
Offers mutual authentication	Yes	Yes	Yes	Yes	Yes
Offers biometric data updating	Yes	Yes	Yes	Yes	Yes
Offers secure password updating	No	No	No	Yes	Yes
Offers formal security analysis	Yes	Yes	Yes	Yes	Yes

### 4.6.2 Relative Performance based on Computational Cost

The execution time as considered in [113, 114], for the different cryptographic operation (performed by user  $\mathcal{U}_i$  and the gateway node  $GW_N$  with a computer system having windows 7 operating system, Intel (R) core (TM) 2 Quad CPU Q8300, @2.50 Hz processor, and 2 GB RAM) are listed in following Table 4.23. We assume that time for executing a fuzzy extractor is the same as that for executing a hash function because the fuzzy extractor [18] can be constructed from universal hash functions or error-correcting codes requiring only lightweight operations.

The computational time and energy consumed by the various cryptographic operations (performed by MicaZ sensor node  $SN_j$  with 8-bit ATmega128L Atmel processor, 4 K bytes ROM, 128 K bytes ROM, 512 K bytes EEPROM, 2 AA battery with TinyOS [111] and nesC [112] programming language) are listed in following Table 4.24.

## 4.6 Comparative Analysis based on Security Features and Computational Overhead

**Table 4.23:** Execution time on computer system for cryptographic operation.

Notation	Operation	Time Taken (in <i>Millisecond</i> )
$T_h$	One-way cryptographic hash function	0.5
$T_e$	Elliptic curve point multiplication	50.3
$T_f$	Fuzzy extractor used in biometric verification	0.5
$T_E$	Symmetric key encryption/decryption	8.7

The comparison of user authentication protocols on the basis of computational cost is shown in Table 4.25. In the proposed protocol, the registration phase has computation costs  $T_h \approx 0.50$  millisecond and  $4T_h + T_e \approx ((4 \times 0.50 + 50.3) = 52.30)$  millisecond associated with  $\mathcal{U}_i$  and  $GWN$  respectively; the authenticated session key establishment phase has computational costs  $3T_h \approx 1.50$  millisecond,  $3T_h + T_e \approx ((1.50 + 50.3) = 51.80)$  millisecond and  $TS_E \approx 5.05$  millisecond associated with  $\mathcal{U}_i$ ,  $GWN$  and  $SN_j$  respectively. Similarly the computational cost for Das et al. [81], Choi et al. [61], Park et al. [63] and Moon et al.'s [64] protocols are evaluated, represented and compared in Table 4.25.

**Table 4.24:** Execution time and energy consumption on MicaZ sensor node for cryptographic operations.

Function	Time (in <i>millisecond</i> )	Energy (in $\mu$ Joule)
Symmetric Encryption and Decryption (AES-128) [115]	$TS_E \approx 5.05$	121.2
Hashing (SHA-1) [103]	$TS_h \approx 3.63$	87.12
Elliptic curve Fixed Point Multiplication (MoTE ECC-160) [116]	$TS_e \approx 370$	8880

The comparative analysis illustrates that the performance time for the sensor node  $SN_j$  is very low (because we have moved the computational burden of elliptic curve point multiplication from sensor node  $SN_j$  to the gateway node  $GWN$  with satisfying security properties) for the proposed protocol.

The energy consumption of the cryptography operations on the sensor node  $SN_j$  is estimated depending on the following relation:

$$Energy = Voltage \times Current \times Times$$

where Voltage = 3.0 Volts and current = 8 Milliampere for the micaZ sensor node  $SN_j$  including AA batteries. Accordingly, the energy depletion for Das et al. [81],

## 4.6 Comparative Analysis based on Security Features and Computational Overhead

**Table 4.25:** Comparison of protocols on the basis of computational cost.

protocol	Registration Phase Time (in millisecond)		Authentication and Session Key Establishment Phase Time (in millisecond)		
	$U_i$	$GW N$	$U_i$	$GW N$	$SN_j$
A.K.Das [81]	$4T_h + T_f$ $\approx 2.50$	$2T_h$ $\approx 1.00$	$6T_h + T_f$ $+T_E \approx 12.20$	$3T_h + 2T_E$ $\approx 18.90$	$2TS_h + TS_E$ $\approx 12.31$
Choi et al. [61]	$T_h + T_f$ $\approx 1.00$	$3T_h$ $\approx 1.50$	$10T_h + T_f$ $+T_E + 2T_e$ $\approx 114.80$	$10T_h + 2T_E$ $\approx 22.40$	$6TS_h + TS_E$ $+2TS_e$ $\approx 766.83$
Park et al. [63]	$T_h + T_f$ $\approx 1.00$	$5T_h$ $\approx 2.50$	$10T_h + T_f$ $+2T_e \approx 106.10$	$11T_h$ $\approx 6.50$	$4TS_h + 2TS_e$ $\approx 754.52$
Moon et al. [64]	$T_h + T_f$ $\approx 1.00$	$3T_h + T_e$ $\approx 51.80$	$6T_h + T_f$ $+3T_e$ $\approx 53.80$	$6T_h + T_E + T_e$ $\approx 62$	$4TS_h + TS_E$ $+2TS_e$ $\approx 759.57$
Proposed Protocol	$T_h$ $\approx 0.50$	$4T_h + T_e$ $\approx 52.30$	$3T_h$ $\approx 1.50$	$3T_h + T_e$ $\approx 51.80$	$TS_E$ $\approx \mathbf{5.05}$

Choi et al. [61], Park et al. [63] and Moon et al.'s [64] protocols are  $((8 \times 3.0 \times (2TS_h + TS_E)) = 295.44)$ ,  $((8 \times 3.0 \times (6TS_h + TS_E + 2TS_e)) = 18,403.92)$ ,  $((8 \times 3.0 \times (4TS_h + 2TS_e)) = 18,108.48)$  and  $((8 \times 3.0 \times (4TS_h + TS_E)) = 18,229.68)$  respectively. The energy consumption for the proposed protocol is  $((8 \times 3.0 \times TS_E) = 5.05)$ . The comparative analysis on the basis of energy exploitation is shown in Table 4.26 which represents that the proposed protocol consumes less energy compared to other existing protocols.

We assume that  $ID_{U_i}$ , message request  $req$ , message response  $R/RM$ , encrypted message  $Enc_k[s]$ , time-stamp  $T_{U_i}/T_{GWN}/T_{SN_j}$ , hash function  $h(.)$  and the point on elliptic curve take 160, 32, 32, 128, 32, 160 and 160 bits respectively, for the comparative analysis of message communications overhead. In our proposed protocol, over the authentication and session key establishment phase, the message  $ID_{U_i}, X_{U_i}, \alpha$  requires  $(160 + 160 + 128 = 448)$  bits, whereas the messages  $(\beta)$  and  $\alpha$  require  $(128 + 128 = 265)$  bits. As a result, the total communication overhead of our proposed protocol becomes 713 bits on the basis of 3 communicated messages. For A.K.Das's [81] protocol, in the login phase, the message  $(ID_{U_i}, req)$  requires  $(160 + 32) = 192$  bits, whereas in the authentication and key agreement phase, the messages  $R, Enc_{ek_i}(R, T_1, ID_{SN_j}), (ID_{U_i}, Y_j)$  and  $(h(SK_{ij}), T_3)$  require 32, 128, 288, and 352 bits, respectively. As a result, the total communication overhead of A.K.Das's protocol becomes 832 bits. Similarly the communications overhead for Choi et al. [61], Park

## 4.7 Overall Analysis and Lessons Learned

et al. [63] and Moon et al.'s [64] protocols are evaluated, represented and compared in Table 4.27. The comparative analysis of Table 4.27 illustrates that the proposed protocol has less communication overhead (which saves communication energy and bandwidth) compared to other existing protocols.

**Table 4.26:** Comparison of protocols on the basis of energy consumption on sensor node  $SN_j$ .

	<b>A.K.Das [81]</b>	<b>Choi [61]</b>	<b>Park [63]</b>	<b>Moon [64]</b>	<b>Proposed Protocol</b>
<b>Energy (in <math>\mu</math> Joule)</b>	295.44	18,403.92	18,108.48	18,229.68	121.2

**Table 4.27:** Comparison of protocols on the basis of communication overhead.

	<b>A.K.Das [81]</b>	<b>Choi [61]</b>	<b>Park [63]</b>	<b>Moon [64]</b>	<b>Proposed Protocol</b>
<b>Communication Overhead (in bits)</b>	832	1504	1696	1920	713
<b>Number of Messages Communicated</b>	5	3	3	3	3

## 4.7 Overall Analysis and Lessons Learned

The reliability and security evaluation of prevalent user authentication protocols of WSNs & IoT of the literature shows that the conventions are vulnerable to various attacks like sensor node impersonation attack, user impersonation attack and the attacks on the basis of legitimate users.

The performance analysis represents that the existing protocols are inefficient considering the computational cost. Whereas, the comparative security and performance analysis represents that our proposed protocol is secure against stolen smart card attack, user impersonation attack, sensor node impersonation attack, sensor node capture attack, replay attack, man-in-the-middle attack.

The proposed authentication protocol contains various security features such as integrity, confidentiality, mutual authentication, data freshness, three-factor authentication, password and bio-metric data update. The proposed protocol is effective concerning the computational overhead of the resource-constrained sensor nodes, and it conserves the communication bandwidth, energy.

As a consequence, the protocol is relevant for purposes of resource-constrained ubiquitous computing devices. Hence, the proposed protocol can be applied in several real-world applications consisting of resource confinement sensor devices of WSNs & IoT where bio-metric based secure user authentication and dynamic session key establishment are required. The proposed protocol can be utilized for the implementation of bio-metric based secure trustworthy banking and financial transactions accepting the smart card, point-of-sale (POS) machines, automated teller machines (ATM).

## 4.8 Summary

In this chapter, we have addressed the security issues concerned among the user and sensor nodes of WSNs & IoT and did the security analysis of several existing protocols of user authentication for WSNs & IoT. We have proposed a valid and secure user authentication, session key establishment protocol for WSNs and IoT based on the smart card, fuzzy extractor and ECDH mechanism.

We have presented security proof using the random oracle model and BAN logic to assure the accuracy of various security specialities involved in the proposed protocol. Then, we have performed the security analysis and verification using universally accepted and robust security tools such as AVISPA and Scyther.

By the well-defined security analysis applying mathematical functions and simulation tools, we have illustrated that the proposed protocol achieves the desirable security requirements and resists the security attacks found in other related existing protocols of user authentication for WSNs & IoT. Subsequently, we have performed the comparative analysis of our protocol with other existing protocols on the grounds of security features and computational cost which justifies that our proposed protocol is reliable, secure, efficient and fit for WSNs & IoT.

## **Chapter 5**

# **LU Decomposition based User Authentication and Key Establishment protocol for WSNs & IoT**

“Nothing is particularly hard if you divide it into small jobs.” Henry Ford

This chapter presents an efficient user authentication and light-weight session key establishment protocol for WSNs & IoT based on LU Decomposition. It concisely illustrates the problem, security challenges and contributions made thereof. Subsequently, it elaborates the proposed protocol, and then, the relative security and computational overhead are analyzed.

### **5.1 Introduction and Problem Definition**

Addressing security-sensitive wireless networks of sensor devices, the authenticity of the genuine user is the prominent requirement. Because of constrained-resources in these sensor devices executing conventional cryptographic mechanism is not a simple task. Therefore, in this chapter, we propose a lightweight mechanism for authenticating users of a sensor network using fuzzy extractor along with a novel matrix based session key establishment protocol. Following that, we perform the security analysis of our proposed protocol applying universally trusted automated verification tools such as AVISPA and Scyther. Later, we present logical verification



using BAN Logic. Finally, we do the computational analysis and we demonstrate by comparative analysis the superiority of our protocol with respect to computational overhead and security features. The necessary notations applied in the design and analysis of the proposed protocol are represented in Table 1.

**Problem Definition:** The problem definition of this chapter is as follows:

Design and analysis of secure and efficient user authentication and light-weight session key establishment protocol for resource constrained WSNs & IoT which can provide the significant security features (such as mutual authentication, secure session key establishment, confidentiality, integrity, freshness etc.) and prevent the considerable security attacks (such as stolen smart card, user impersonation, sensor node impersonation, etc.) with low computational overhead.

## 5.2 Our Contributions

In this chapter, we propose a novel user authentication protocol based on LU Decomposition to give the user access to the real-time information by authorizing directly at the sensor node level. It makes possible for users of WSNs & IoT to communicate securely with the sensor nodes to have responses to their queries. Our proposed protocol has the following exciting features:

- It constructs a secure session key between the legitimate user  $\mathcal{U}_i$  and a sensor node  $SN_j$  (based on a proposed light weight LU decomposition technique) for future secure communication (utilizing the established session key.) of the real-time information inside WSNs & IoT.
- It provides higher security as compared with the other related protocols, since it supports mutual authentication between the user and the sensor node, withstands denial-of-service (DOS) attack, smart card breach attack, privileged-insider attack, and sensor node capture attack.
- It supports dynamic sensor node inclusion after preliminary deployment of sensor nodes in the WSNs & IoT. The proposed protocol does not need to update information for new sensor nodes inclusion in the user's smart card.

- It supports user's credential update locally without the assistance of the gateway node.
- It provides absolute security against sensor node capture attacks. Therefore, compromise of a sensor does not disclose any secret information of other sensor nodes and it does not help to compromise any other secure data communication between the legitimate user and the non-compromised sensor nodes in the WSNs & IoT.
- Subsequently, the formal security verification using Scyther and AVISPA tool represents that the proposed protocol is robust, safe and it satisfies all the significant security claims defined for the user authentication mechanism in WSNs & IoT.
- The logical security verification using BAN logic ensures that the exchanged messages of the proposed protocol are trustworthy and secure against eavesdropping.
- Also, we have compared the security and computational features provided by our protocol with other existing related protocols. Overall, our proposed protocol has adequate security and computational performance than other existing protocols.

## 5.3 Discussions and Proposal

To design a secure and efficient user validation protocol of WSNs & IoT, we use the concept of fuzzy extractor [18] for authenticating the user and LU decomposition for establishing the session key between user and sensor node.

In this section, we first describe the concept of fuzzy extractor and efficient way of using LU decomposition for establishing the session key. Afterwards, we propose the pre-deployment protocol for user, sensor, gateway and the procedure of registering the user  $\mathcal{U}_i$  and the mechanism of login, authentication and session key establishment between  $\mathcal{U}_i$  and  $SN_j$ . Finally, we describe the user's credential update mechanism.

### 5.3.1 LU Decomposition of $Mat$ and secret sharing:

LU decomposition of a matrix  $Mat$  is a process of decomposing  $Mat$  into a lower triangular matrix  $LO$  and an upper triangular matrix  $UP$  such that  $Mat = LO \times UP$  and

$$LO_{ij} = \begin{cases} LO_{ij}, & \text{if } i \geq j \\ 0, & \text{otherwise} \end{cases} \quad \text{and} \quad UP_{ij} = \begin{cases} UP_{ij} & \text{if } i \leq j \\ 0, & \text{otherwise} \end{cases}$$

As demonstrated by [117], we assume any two entities  $E_x$  and  $E_y$  have  $\{LO_r(E_x)$  ( $x^{th}$  row of  $LO$ ) ,  $UP_c(E_x)$  ( $x^{th}$  column of  $UP$ )  $\}$  and  $\{LO_r(E_y)$  ( $y^{th}$  row of  $LO$ ),  $UP_c(E_y)$  ( $y^{th}$  column of  $UP$ )  $\}$  respectively. If  $E_x$  shares  $UP_c(E_x)$  with  $E_y$  and  $E_y$  shares  $UP_c(E_y)$  with  $E_x$ ,  $E_x$  and  $E_y$  can calculate a common shared key as follows:

$E_x$  calculates :  $LO_r(E_x) \times UP_c(E_y) = Mat_{xy}$

$E_y$  calculates :  $LO_r(E_y) \times UP_c(E_x) = Mat_{yx}$

Since  $Mat$  is a symmetric matrix i.e.  $Mat_{xy} = Mat_{yx}$ , therefore  $E_x$  and  $E_y$  discovers the same key.

The values of  $LO_{ij}$  and  $UP_{ij}$  are 0 for  $i < j$  and  $i > j$  respectively and therefore it has no effect on the final result obtained after multiplication of  $i^{th}$  row of  $LO$  and  $j^{th}$  column of  $UP$ . As the sensor nodes and smart cards of the users have limited memory and processing power, hence we propose to store the value of  $LO_{ij}$  and  $UP_{ij}$  for  $i \geq j$  and  $i \leq j$  respectively. We can assign these value as follow:

$$LO_r(U_i) = [LO_{i1} \dots LO_{ii}] \quad \text{and} \quad LO_r(SN_j) = [LO_{i1} \dots LO_{jj}],$$

$$UP_c(U_i) = \begin{bmatrix} UP_{1i} \\ \vdots \\ \vdots \\ UP_{ii} \end{bmatrix} \quad \text{and} \quad UP_c(SN_j) = \begin{bmatrix} UP_{1j} \\ \vdots \\ \vdots \\ UP_{jj} \end{bmatrix}$$

For efficient multiplication of these row and column matrix we use the following approach:

$$LO_r(U_i) \times UP_c(SN_j) = \begin{cases} \sum_{k=0}^j LO_r(U_i)_k \times UP_c(SN_j)_k, & \text{if } i \geq j \\ \sum_{k=0}^i LO_r(U_i)_k \times UP_c(SN_j)_k, & \text{otherwise} \end{cases}$$

Here, for  $LO_r(U_i)$  and  $UP_c(SN_j)$  the value of  $i$  and  $j$  represents the  $i^{th}$  user and  $j^{th}$  sensor node respectively,  $i$  and  $j$  are also equal to the number of elements of row matrix  $LO_r(U_i)$  and column matrix  $UP_c(SN_j)$  respectively.  $LO_r(U_i)_k$  represents the  $k^{th}$  element of  $LO_r(U_i)$ .

$$LO_r(SN_j) \times UP_c(U_j) = \begin{cases} \sum_{k=0}^j LO_r(SN_i)_k \times UP_c(U_j)_k, & \text{if } j \geq i \\ \sum_{k=0}^i LO_r(SN_i)_k \times UP_c(U_j)_k, & \text{otherwise} \end{cases}$$

### 5.3.1.1 Storage Analysis

If  $len$  be the number of bits or length of each keying elements of  $LO$  or  $UP$ ,  $z$  be the number of bits to represent  $n - 1$  zero elements. Then, the total memory required to store keys as per Choi et. al's protocol [117] is,

$$\Gamma_{[117]} = 2 \times n^2 \times len$$

Total memory required to store keys as per Pathan et.al's [118] is,

$$\Gamma_{[118]} = len \times \sum_{i=1}^n i + n \times (2 \times z) = len \times \frac{n \times (n+1)}{2} + n \times (2 \times z)$$

Total memory required to store keys in our protocol is,

$$\Gamma_{our} = len \times \sum_{i=1}^n i = len \times \frac{n \times (n+1)}{2}$$

Therefore, we can say that  $\Gamma_{our} < \Gamma_{[118]} < \Gamma_{[117]}$ .

### 5.3.2 Pre-deployment Phase:

Under this section, we believe that the WSNs & IoT consist of users (with the smart card which can be captured or stolen by the adversary  $\mathcal{A}$ ), sensor nodes ( $\mathcal{A}$  can capture it) and gateway (it is trusted, and  $\mathcal{A}$  can not compromise it). The  $GWN$  first produces a set  $p$  of a large pool of keys and creates a symmetric matrix  $Mat$  of size  $n \times n$  utilising the set  $p$ . The  $GWN$  performs the  $LU$  decomposition operation on  $Mat$  to get  $LO$ ,  $UP$ . Eventually, the  $GWN$  securely produces the row matrix  $LO_r(SN_j)$  and the column matrix  $UP_c(SN_j)$  to the sensor node  $SN_j$ .

### 5.3.3 User Registration Phase:

A legitimate user  $\mathcal{U}_i$  who wants to access the confidential report of WSNs, follows the procedures as shown in following Table 5.1.

**Table 5.1:** User Registration Phase

Step 1: for user $\mathcal{U}_i$	Step 2: for gateway node $GW N$
$\mathcal{U}_i$ selects $ID_{\mathcal{U}_i}, PW_{\mathcal{U}_i}$ and provides $BIO_{\mathcal{U}_i}$ . Then, generates $(\sigma_i, \tau_i) = Gen(BIO_{\mathcal{U}_i})$ , and assign $IPB_i = h(ID_{\mathcal{U}_i}    PW_{\mathcal{U}_i}    h(\sigma_i))$ ,  $\mathcal{U}_i$ transmits $\langle ID_{\mathcal{U}_i}, IPB_i \rangle$ to $GW N$ , $\xrightarrow{\text{SecureChannel}}$	$GW N$ extracts $LO_r(U_i)$ and $UP_c(U_i)$ from $Mat$ and derives $A_{\mathcal{U}_i} = IPB_i \oplus LO_r(U_i)$ , $B_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i}    IPB_i    LO_r(U_i))$ , $W_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i}    IPB_i) \oplus UP_c(U_i)$ Then, $GW N$ stores $A_{\mathcal{U}_i}, B_{\mathcal{U}_i}, W_{\mathcal{U}_i}$ into $SC_{\mathcal{U}_i}$ $\xleftarrow{\text{SecureChannel}}$ $GW N$ transfers $SC_{\mathcal{U}_i}$ to $\mathcal{U}_i$
<b>Step3.</b> $\mathcal{U}_i$ stores $Gen(.), Rep(.)$ and $h(.), \tau_i, \mathcal{T}$ into $SC_{\mathcal{U}_i}$ .	

### 5.3.4 User Authentication and Session Key Establishment Phase:

In order to retrieve data from  $SN_j$ ,  $\mathcal{U}_i$  gets authenticated using  $SC_{\mathcal{U}_i}, ID_{\mathcal{U}_i}, PW_{\mathcal{U}_i}$ , noisy bio-metric information  $BIO'_{\mathcal{U}_i}$  and fuzzy extractor function  $Rep(.)$ . Afterwards, the  $GW N$  verifies the credentials of  $\mathcal{U}_i$  and sends a secure message to  $SN_j$  for establishing a secure session with  $\mathcal{U}_i$ .  $SN_j$  verifies the message and establishes the key with  $\mathcal{U}_i$ . Table 5.2 describes the authentication and key sharing mechanism in detail.

The sequence diagram of the message transmission for the user registration, authentication and key establishment phase is shown in the following Figure 5.1.

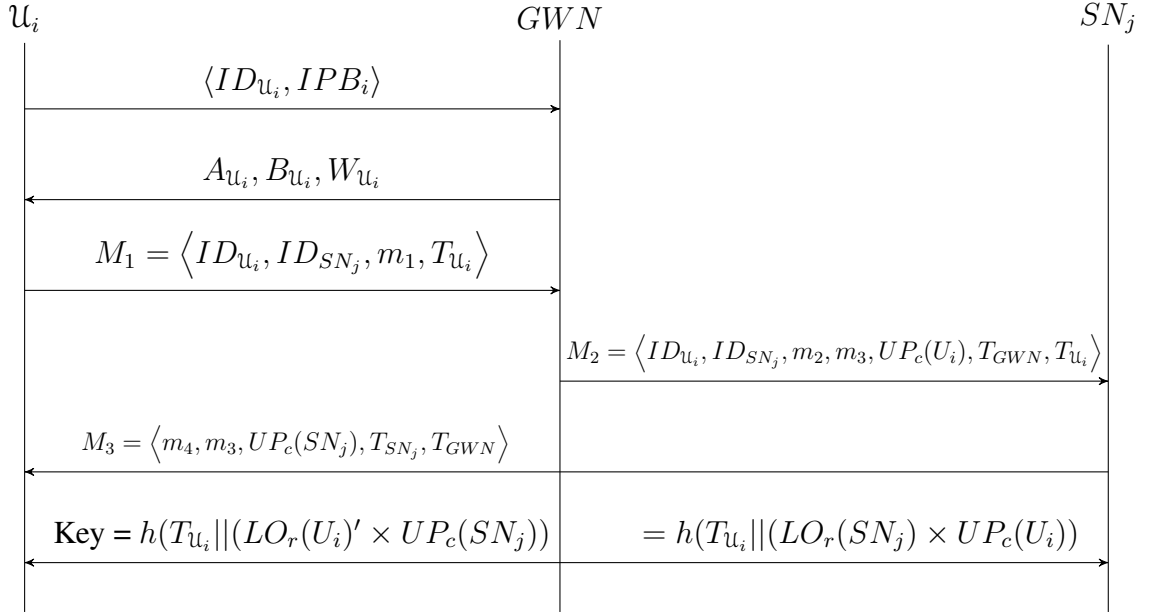
### 5.3.5 User's Credential Update Phase:

We provide a mechanism for the user  $\mathcal{U}_i$  to change his/her password and bio-metric information before an adversary (who can steal user's credential without his/her knowl-

## 5.4 Security Analysis:

edge ) get an opportunity to use it. The procedure for updating the credential is shown in Table 5.3.

**Figure 5.1:** Sequence Diagram 3 for Registration, Authentication and Key Establishment



## 5.4 Security Analysis:

To validate the security feature of our protocol, we first perform the informal analysis considering major and minor attacks in WSNs. Afterwards, we implement our protocol using Security Protocol Description Language and evaluate our security claims using Sythet tool [19]. For automated validation of the protocol using AVISPA tool [7], we use High- Level Protocols Specification Language. Finally, we do the logical verification of the protocol using BAN logic [20].

### 5.4.1 Informal Security Analysis:

The informal security analysis indicates that our protocol is designed to withstand the popular security attacks as follows:

## 5.4 Security Analysis:

**Table 5.2:** Authenticated Key Exchange Phase

Step 1: for $\mathcal{U}_i$	Step 2: for $GW_N$
<p><math>\mathcal{U}_i</math> puts <math>SC_{\mathcal{U}_i}</math> into card reader and provides <math>ID_{\mathcal{U}_i}, PW_{\mathcal{U}_i}, BIO'_{\mathcal{U}_i}</math>. Then find out <math>\sigma'_i = Rep(BIO'_{\mathcal{U}_i}, \tau_i)</math>, <math>IPB'_i = h(ID_{\mathcal{U}_i}    PW_{\mathcal{U}_i}    h(\sigma'_i))</math>, <math>LO_r(U_i)' = A_{\mathcal{U}_i} \oplus IPB'_i</math>, <math>B'_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i}    IPB'_i    LO_r(U_i)')</math>,</p> <p><b>if</b> <math>B_{\mathcal{U}_i} = B'_{\mathcal{U}_i}</math> <b>then</b></p> <p style="padding-left: 20px;"><math>\mathcal{U}_i</math> computes <math>UP_c(U_i)' = W_{\mathcal{U}_i} \oplus h(ID_{\mathcal{U}_i}    IPB'_i)</math> and finds the current time-stamp <math>T_{\mathcal{U}_i}</math>. Then <math>\mathcal{U}_i</math> evaluates <math>m_1 = h(ID_{\mathcal{U}_i}    ID_{SN_j}    LO_r(U_i)'    T_{\mathcal{U}_i})</math>, <math>M_1 = \langle ID_{\mathcal{U}_i}, ID_{SN_j}, m_1, T_{\mathcal{U}_i} \rangle</math></p> <p style="text-align: center;"><math>\xrightarrow{\text{ViaPublicChannel}}</math></p> <p style="padding-left: 20px;"><math>\mathcal{U}_i</math> transmits <math>\langle M_1 \rangle</math> to <math>GW_N</math></p> <p><b>end</b></p> <p><b>else</b></p> <p style="padding-left: 20px;">Reject <math>\mathcal{U}_i</math></p> <p><b>end</b></p>	<p><b>if</b> <math>T' - T_{\mathcal{U}_i} \leq \Delta T</math> <b>then</b></p> <p style="padding-left: 20px;"><math>GW_N</math> Extract <math>LO_r(U_i), UP_c(U_i), LO_r(SN_j), UP_c(SN_j)</math> from <math>Mat</math> and computes <math>m'_1 = h(ID_{\mathcal{U}_i}    ID_{SN_j}    LO_r(U_i)    UP_c(U_i)    T_{\mathcal{U}_i})</math>,</p> <p style="padding-left: 20px;"><b>if</b> <math>m_1 = m'_1</math> <b>then</b></p> <p style="padding-left: 40px;">Find current time-stamp <math>T_{GW_N}</math>, and computes <math>m_2 = h(LO_r(SN_j)    ID_{\mathcal{U}_i}    UP_c^{\mathcal{U}_i}    T_{GW_N})</math>, <math>m_3 = h(LO_r(U_i)    UP_c(SN_j)    T_{\mathcal{U}_i}    T_{GW_N})</math>, <math>M_2 = \langle ID_{\mathcal{U}_i}, ID_{SN_j}, m_2, m_3, UP_c(U_i), T_{GW_N}, T_{\mathcal{U}_i} \rangle</math></p> <p style="padding-left: 20px;"><b>end</b></p> <p style="padding-left: 20px;"><b>else</b></p> <p style="padding-left: 40px;">Rejects <math>\mathcal{U}_i</math></p> <p style="padding-left: 20px;"><b>end</b></p> <p style="text-align: center;"><math>\xrightarrow{\text{ViaPublicChannel}}</math></p> <p style="padding-left: 20px;"><math>GW_N</math> sends <math>M_2</math> to <math>SN_j</math></p> <p><b>end</b></p> <p><b>else</b></p> <p style="padding-left: 20px;">Reject <math>\mathcal{U}_i</math></p> <p><b>end</b></p>
Step 3: for $SN_j$	Step 4: for $U_i$
<p><b>if</b> <math>(T'' - T_{\mathcal{U}_i} \leq 2\Delta T, T'' - T_{GW_N} \leq \Delta T)</math> <b>then</b></p> <p style="padding-left: 20px;"><math>SN_j</math> extracts <math>LO_r(SN_j)</math> from its memory and computes <math>m'_2 = h(LO_r(SN_j)    ID_{\mathcal{U}_i}    UP_c(U_i)    T_{GW_N})</math>,</p> <p style="padding-left: 20px;"><b>if</b> <math>m_2 = m'_2</math> <b>then</b></p> <p style="padding-left: 40px;">Find current time-stamp <math>T_{SN_j}</math> and compute the session key <math>SK = h(T_{\mathcal{U}_i}    (LO_r(SN_j) \times UP_c(U_i)))</math>. Then, calculate <math>m_4 = h(ID_{\mathcal{U}_i}    ID_{SN_j}    SK    m_3    T_{SN_j})</math>, <math>M_3 = \langle m_4, m_3, UP_c(SN_j), T_{SN_j}, T_{GW_N} \rangle</math></p> <p style="text-align: center;"><math>\xrightarrow{\text{ViaPublicChannel}}</math></p> <p style="padding-left: 20px;"><math>SN_j</math> sends <math>M_3</math> to <math>\mathcal{U}_i</math></p> <p><b>end</b></p> <p><b>else</b></p> <p style="padding-left: 20px;">Reject <math>\mathcal{U}_i</math></p> <p><b>end</b></p> <p><b>end</b></p> <p><b>else</b></p> <p style="padding-left: 20px;">Reject <math>\mathcal{U}_i</math></p> <p><b>end</b></p>	<p><b>if</b> <math>(T''' - T_{GW_N} \leq 2\Delta T, T''' - T_{SN_j} \leq \Delta T)</math> <b>then</b></p> <p style="padding-left: 20px;"><math>\mathcal{U}_i</math> computes <math>m'_3 = h(LO_r(U_i)'    UP_c(SN_j)    T_{\mathcal{U}_i}    T_{GW_N})</math>,</p> <p style="padding-left: 20px;"><b>if</b> <math>m_3 = m'_3</math> <b>then</b></p> <p style="padding-left: 40px;">Compute the session key <math>SK' = h(T_{\mathcal{U}_i}    (LO_r(U_i)' \times UP_c(SN_j)))</math>. Then, calculates <math>m'_4 = h(ID_{\mathcal{U}_i}    ID_{SN_j}    SK'    m'_3    T_{SN_j})</math>, <b>if</b> <math>m_4 = m'_4</math> <b>then</b></p> <p style="padding-left: 60px;">Establish the session key <math>SK' = SK</math> with sensor node <math>SN_j</math></p> <p style="padding-left: 40px;"><b>end</b></p> <p style="padding-left: 20px;"><b>else</b></p> <p style="padding-left: 40px;">Reject <math>\mathcal{U}_i</math></p> <p style="padding-left: 20px;"><b>end</b></p> <p><b>end</b></p> <p><b>else</b></p> <p style="padding-left: 20px;">Reject <math>\mathcal{U}_i</math></p> <p><b>end</b></p> <p><b>end</b></p> <p><b>else</b></p> <p style="padding-left: 20px;">Reject <math>\mathcal{U}_i</math></p> <p><b>end</b></p>

**Table 5.3:** User's Credential Update Phase

For User ( $\mathcal{U}_i$ )
$\mathcal{U}_i$ puts $SC_{\mathcal{U}_i}$ into card reader and provides $ID_{\mathcal{U}_i}, PW_{\mathcal{U}_i}, BIO_{\mathcal{U}_i}$ , It produces $\sigma'_i = Rep(BIO'_i, \tau_i)$ , $IPB'_i = h(ID_{\mathcal{U}_i}    PW_{\mathcal{U}_i}    h(\sigma'_i))$ , $LO_r(U_i)' = A_{\mathcal{U}_i} \oplus IPB'_i$ , $B'_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i}    IPB'_i    LO_r(U_i)')$ , <b>if</b> $B_{\mathcal{U}_i} = B'_{\mathcal{U}_i}$ <b>then</b> Computes $UP_c(U_i)' = W_{\mathcal{U}_i} \oplus h(ID_{\mathcal{U}_i}    IPB'_i)$ , $\mathcal{U}_i$ provides new password $PW_{\mathcal{U}_i}^{new}$ and bio-metric information $BIO_i^{new}$ and computes $(\sigma_i^{new}, \tau'_i) = Gen(BIO_i^{new})$ $IPB_i^{new} = h(ID_{\mathcal{U}_i}    PW_{\mathcal{U}_i}^{new}    h(\sigma_i^{new}))$ $A_{\mathcal{U}_i}^{new} = IPB_i^{new} \oplus LO_r(U_i)$ $B_{\mathcal{U}_i}^{new} = h(ID_{\mathcal{U}_i}    IPB_i^{new}    LO_r(U_i))$ $W_{\mathcal{U}_i}^{new} = h(ID_{\mathcal{U}_i}    IPB_i^{new}) \oplus UP_c(U_i)$ Finally, replace $A_{\mathcal{U}_i}, B_{\mathcal{U}_i}, W_{\mathcal{U}_i}$ of $SC_{\mathcal{U}_i}$ with $A_{\mathcal{U}_i}^{new}, B_{\mathcal{U}_i}^{new}, W_{\mathcal{U}_i}^{new}$ respectively. <b>end</b> <b>else</b> Reject $\mathcal{U}_i$ <b>end</b>

#### 5.4.1.1 Attack based on stolen smart card:

Our protocol is safe from stolen smart card of legitimate user  $\mathcal{U}_i$  because an adversary  $\mathcal{A}$  can not extract the secret credential  $PW_{\mathcal{U}_i}, \sigma_i, LO_r(U_i)$  etc. without having the authentic bio-metric credential  $BIO_{\mathcal{U}_i}$  of  $\mathcal{U}_i$ .

#### 5.4.1.2 Replay Attack:

The time-stamps  $T_{\mathcal{U}_i}, T_{GWN}, T_{SN_j}$  are stored in variable  $m_1, m_3, m_4$  after secure hashing. Therefore an adversary  $\mathcal{A}$  can not perform the replay attack using message  $M_1, M_2, M_3$ .

#### 5.4.1.3 User impersonation attack:

We have prevented the user's impersonation attack applying fuzzy extractor on the bio-metric information  $BIO_{\mathcal{U}_i}$  of  $\mathcal{U}_i$ . The adversary  $\mathcal{A}$  can not impersonate the user  $\mathcal{U}_i$  without the bio-metric credential  $BIO_{\mathcal{U}_i}$  of  $\mathcal{U}_i$ .



### 5.4.1.4 Sensor node impersonation attack:

We uniquely and securely distribute a key  $LO_r(SN_j)$  to each sensor node  $SN_j$  and we test the message  $m_3$  and  $m_4$  at user  $\mathcal{U}_i$ . Accordingly, an adversary  $\mathcal{A}$  cannot do the sensor impersonation attack.

### 5.4.1.5 Man-in-the-middle attack(MITM) :

The verification of the message  $m_1$  on  $GW N$ ,  $m_2$  on  $SN_j$ ,  $m_3$  and  $m_4$  on  $\mathcal{U}_i$  stops an adversary  $\mathcal{A}$  to perform MITM attack.

## 5.4.2 Security verification using Scyther and AVISPA tool :

We stipulate our protocol applying Security Protocol Description Language (SPDL) based on the operational interpretation of Scyther tool. Table 5.4 outlines the SPDL designation of our proposed protocol:

The result of security verification using Scyther tool is shown Figure 5.2 . The result indicates that no attacks found on all the claims which we specified for the three roles  $U_i, GW N, SN_j$ . The result obtained (Figure 5.3 ) using OFMC back-ends of AVISPA tool indicates that our protocol is safe from Dolev-Yao [110] intruder model.

### 5.4.2.1 Logical verification using BAN logic

We use BAN logic [20] to verify the freshness of time-stamp to avoid replay attack and we validate the message origin to achieve authenticity. The notations we use for logical verification is shown in Table 1

1. Verification of freshness of  $T_{\mathcal{U}_i}$  by  $GW N$  (using message - meaning and nonce verification rule):

$$\bullet \quad \frac{GW N | \equiv U_i \xrightarrow{LO_r(U_i)} GW N, GW N \triangleleft \langle T_{\mathcal{U}_i} \rangle_{LO_r(U_i)}}{GW N | \equiv U_i | \sim T_{\mathcal{U}_i}}$$

That is, if  $GW N$  believe the secret  $LO_r(U_i)$  is shared with  $\mathcal{U}_i$  and sees  $\langle T_{\mathcal{U}_i} \rangle_{X_{\mathcal{U}_i}}$ , then  $GW N$  believe  $(| \equiv) U_i$  once said  $T_{\mathcal{U}_i}$

**Table 5.4:** The spdl specification of the proposed protocol

<pre> hashfunction h; const XOR : Function; const MatMul : Function ; const Gen : Function ; const Rep : Function; protocol UserValidation(Ui, GWN, SNj) { macro SIGi = Gen(Bi); macro IPBi = h(IDui, PWui, h(SIGi)); macro SIGi' = Rep(Bi', TAUi'); macro IPBi' = h(IDui, PWui, h(SIGi')); macro Aui = XOR(IPBi, LOrUi); macro Bui = h(IDui, IPBi, LOrUi); macro Wui = XOR(h(IDui, IPBi), UPcUi); macro LOrUi' = XOR(Aui, IPBi'); macro Bui' = h(IDui, IPBi', LOrUi'); macro UPcUi' = XOR(Wui, h(IDui, IPBi')); macro SK = h(Tui, MatMul(LOrSNj, UPcUi)); macro m1 = h(IDui, IDsnj, LOrUi', UPcUi', Tui); macro m2 = h(LOrSNj, IDui, UPcUi, Tgwn); macro m3 = h(LOrUi, Tui, Tgwn); macro m4 = h(IDui, IDsnj, SK, m3, Tsnj); macro m1' = h(IDui, IDsnj, LOrUi, UPcUi, Tui); macro m2' = h(LOrSNj', IDui, UPcUi, Tgwn); macro m3' = h(LOrUi', Tui, Tgwn); macro m4' = h(IDui, IDsnj, SK, m3', Tsnj); role Ui { var Tgwn, Tsnj : Nonce; fresh Tui : Nonce; const IDui, PWui, Bi, Bi', IDsnj, TAUi', LOrUi, UPcUi, UPcSNj, LOrSNj : Ticket; send_1(Ui, GWN, IDui, IPBi); recv_2(GWN, Ui, Aui, Bui, Wui); match(Bui, Bui') ; send_3(Ui, GWN, IDui, IDsnj, m1, Tui); recv_5(SNj, Ui, m4, m3, UPcSNj, Tsnj, Tgwn); match(m3, m3'); match(m4, m4'); claim_Ui1(Ui, Secret, Tui); claim_Ui2(Ui, Secret, LOrUi); claim_Ui3(Ui, Secret, UPcUi); claim_Ui4(Ui, Secret, UPcSNj); claim_Ui6(Ui, SKR, SK)); claim_Ui7 (Ui, Niagree); claim_Ui8 (Ui, Nisynch); } </pre>	<pre> role GWN { fresh Tgwn: Nonce; var Tui : Nonce; const IDui, PWui, IDsnj, Bi, TAUi', Tui, Bi', PWui, LOrUi, LOrSNj, UPcUi, UPcSNj: Ticket; recv_1(Ui, GWN, IDui, IPBi); send_2(GWN, Ui, Aui, Bui, Wui); recv_3(Ui, GWN, IDui, IDsnj, m1, Tui); match (m1, m1'); send_4(GWN, SNj, m2, m3, UPcUi, Tgwn, Tui); claim_GWN1(GWN, Secret, Tgwn); claim_GWN2(GWN, Secret, LOrUi); claim_GWN3(GWN, Secret, LOrSNj); claim_GWN4(GWN, Secret, UPcUi); claim_GWN5(GWN, Secret, UPcSNj); } role SNj { var Tgwn, Tui: Nonce; fresh Tsnj : Nonce; const IDui, IDsnj, Tui, Bi, Bi', TAUi', PWui, LOrSNj, LOrSNj', UPcUi, LOrUi, UPcSNj: Ticket; recv_4(GWN, SNj, m2, m3, UPcUi, Tgwn, Tui); match(m2, m2'); send_5(SNj, Ui, m4, m3, UPcSNj, Tsnj, Tgwn); claim_SNj1(SNj, Secret, Tgwn); claim_SNj2(SNj, Secret, LOrSNj'); claim_SNj3(SNj, Secret, Tsnj); claim_SNj4(SNj, SKR, SK)); } } </pre>
---	---

## 5.4 Security Analysis:

Scyther results : verify

Claim				Status	Comments
UserValidation	Ui	UserValidation,Ui1	Secret Tui	Ok	No attacks within bounds.
		UserValidation,Ui2	Secret LOrUi	Ok	No attacks within bounds.
		UserValidation,Ui3	Secret UPcUi	Ok	No attacks within bounds.
		UserValidation,Ui4	Secret UPcSNj	Ok	No attacks within bounds.
		UserValidation,Ui6	SKR h(Tui,MatMul(LOrUi,UPcSNj))	Ok	No attacks within bounds.
		UserValidation,Ui7	Niagree	Ok	No attacks within bounds.
		UserValidation,Ui8	Nisynch	Ok	No attacks within bounds.
		GWN	UserValidation,GWN1	Secret Tgwn	Ok
	UserValidation,GWN2		Secret LOrUi	Ok	No attacks within bounds.
	UserValidation,GWN3		Secret LOrSNj	Ok	No attacks within bounds.
	UserValidation,GWN4		Secret UPcUi	Ok	No attacks within bounds.
	UserValidation,GWN5		Secret UPcSNj	Ok	No attacks within bounds.
	SNj	UserValidation,SNj1	Secret Tgwn	Ok	No attacks within bounds.
		UserValidation,SNj2	Secret LOrSNj'	Ok	No attacks within bounds.
		UserValidation,SNj3	Secret Tsnj	Ok	No attacks within bounds.
		UserValidation,SNj4	SKR h(Tui,MatMul(LOrSNj,UPcUi))	Ok	No attacks within bounds.

Done.

**Figure 5.2:** Result obtained using Scyther tool.

- $$\frac{GWN| \equiv \#(T_{\mathcal{U}_i}), GWN| \equiv U_i| \sim T_{\mathcal{U}_i}}{GWN| \equiv U_i| \equiv T_{\mathcal{U}_i}}$$

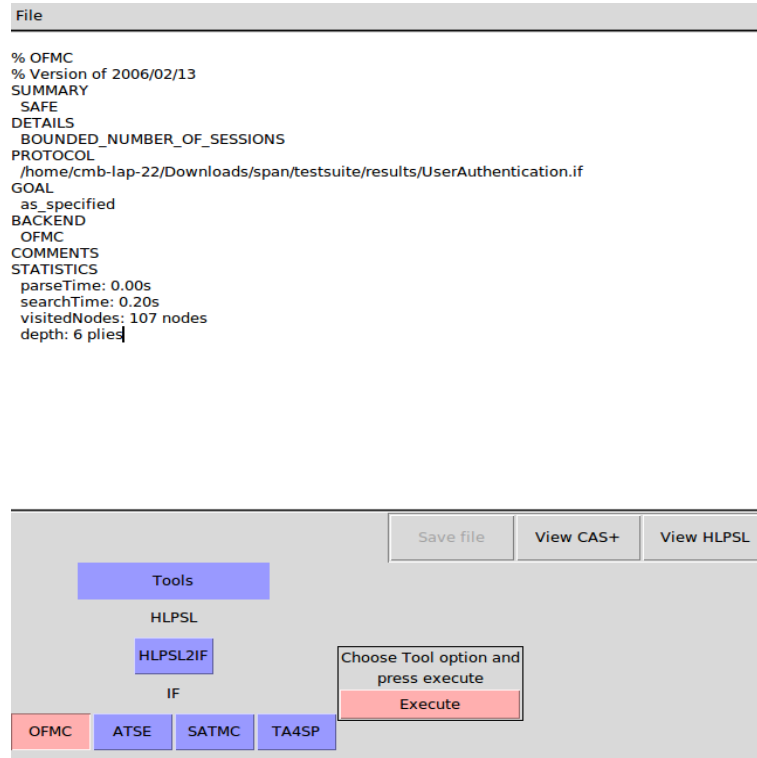
That is, if  $GWN$  believes  $T_{\mathcal{U}_i}$  is fresh and  $GWN$  believes  $\mathcal{U}_i$  once said  $T_{\mathcal{U}_i}$ , then  $GWN$  believe  $U_i$  believes on  $T_{\mathcal{U}_i}$

- Verification of freshness of  $T_{GWN}$  by  $SN_j$  (using message - meaning and nonce verification rule):

- $$\frac{SN_j| \equiv GWN \stackrel{LO_r(SN_j)}{\rightleftharpoons} SN_j, SN_j \triangleleft \langle T_{GWN} \rangle_{K_{GSN_j}}}{SN_j| \equiv GWN| \sim T_{GWN}}$$

That is, if  $SN_j$  believe the secret  $LO_r(SN_j)$  is shared with  $GWN$  and sees

## 5.4 Security Analysis:



**Figure 5.3:** Result obtained using AVISPA tool.

$\langle T_{GWN} \rangle_{K_{GSN_j}}$ , then  $SN_j$  believe  $GWN$  once said  $T_{GWN}$

$$\bullet \frac{SN_j | \equiv \#(T_{GWN}), SN_j | \equiv GWN | \sim T_{GWN}}{SN_j | \equiv GWN | \equiv T_{GWN}}$$

That is, if  $SN_j$  believes  $T_{GWN}$  is fresh and  $SN_j$  believes  $GWN$  once said  $T_{GWN}$ , then  $SN_j$  believes  $GWN$  believes on  $T_{GWN}$

3. Verification of freshness of  $T_{SN_j}$  by  $U_i$  (using message - meaning and nonce verification rule):

$$\bullet \frac{U_i | \equiv \#(T_{SN_j}), U_i | \equiv SN_j | \sim T_{SN_j}}{U_i | \equiv SN_j | \equiv T_{SN_j}}$$

## 5.4 Security Analysis:

That is, if  $\mathcal{U}_i$  believes  $T_{SN_j}$  is fresh and  $\mathcal{U}_i$  believes  $SN_j$  once said  $T_{SN_j}$ , then  $\mathcal{U}_i$  believe  $SN_j$  believes on  $T_{\mathcal{U}_i}$

4. Verification of the authenticity of the message  $m_1$  by  $GW N$  (using message - meaning rule)

$$\bullet \frac{GW N \equiv U_i \xrightarrow{LO_r(U_i)} GW N, GW N \triangleleft \langle m_1 \rangle_{X'_{\mathcal{U}_i}}}{GW N \equiv U_i \mid \sim m_1}$$

That is, if  $GW N$  believes the secret  $LO_r \mathcal{U}_i$  is shared with  $\mathcal{U}_i$  and sees  $\langle m_1 \rangle_{X'_{\mathcal{U}_i}}$ , then  $GW N$  believe  $U_i$  once said  $m_1$

5. Verification of the authenticity of the message  $m_2$  by  $SN_j$  (using message - meaning rule)

$$\bullet \frac{SN_j \equiv GW N \xrightarrow{LO_r(SN_j)} SN_j, SN_j \triangleleft \langle m_2 \rangle_{K_{GSN_j}}}{SN_j \equiv GW N \mid \sim m_2}$$

That is, if  $SN_j$  believes the secret  $LO_r SN_j$  is shared with  $GW N$  and sees  $\langle m_2 \rangle_{K_{GSN_j}}$ , then  $SN_j$  believe  $GW N$  once said  $m_2$

6. Verification of the authenticity of the message  $m_3$  by  $U_i$  (using message - meaning rule)

$$\bullet \frac{U_i \equiv GW N \xrightarrow{LO_r(U_i)} U_i, U_i \triangleleft \langle m_3 \rangle_{LO_r(U_i)}}{U_i \equiv GW N \mid \sim m_3}$$

That is, if  $\mathcal{U}_i$  believes the secret  $LO_r \mathcal{U}_i$  is shared with  $GW N$  and sees  $\langle m_3 \rangle_{LO_r \mathcal{U}_i}$ , then  $\mathcal{U}_i$  believe  $GW N$  once said  $m_3$

## 5.5 Performance Comparison

Table 5.5 shows the comparison of our proposed protocol based on security features and it indicates that our protocol is relatively more secure compared to the existing protocols. . Table 5.6 represents the computational cost comparison, it shows that our protocol provides better computational cost on all the three entities i.e.,  $U_i$ ,  $GW N$  and  $SN_j$ .

**Table 5.5:** Comparisons of Protocols based on Security Features

Security Feature	Yoo et al. [78]	Sun et al. [108]	Xue et al. [79]	Jiang et al. [80]	Althobaiti et al. [59]	Ours
$SF_1$	No	Yes	No	Yes	Yes	Yes
$SF_2$	Yes	No	No	No	No	Yes
$SF_3$	No	No	No	No	No	Yes
$SF_4$	No	No	No	No	Yes	Yes
$SF_5$	No	No	No	No	No	Yes
$SF_6$	Yes	Yes	No	No	Yes	Yes

Note:  $SF_1, SF_2, SF_3, SF_4, SF_5$  are the security features.  $SF_1$  resist the attack based on stolen smart card,  $SF_2$  indicates the secure password updating,  $SF_3$  represents secure bio-metric information updating,  $SF_4$  indicates non-repudiation,  $SF_5$  offers formal security analysis,  $SF_6$  represents no privileged-insider attack

**Table 5.6:** Comparison of Protocols based on Computational Performance

Protocol	Computational Overhead on $U_i, SN_j, GW N$		
	$U_i$	$SN_j$	$GW N$
Yoo et al.'s [78]	$7 T_H$	$2 T S_H$	$11 T_H$
Sun et al.'s [108]	$2 T_H$	$2 T S_H$	$7 T_H$
Xue et al.'s [79]	$12 T_H$	$6 T S_H$	$17 T_H$
Jiang et al.'s [80]	$8 T_H$	$5 T S_H$	$11 T_H$
Althobaiti et al.'s [59]	$2 T_{BFE} + 2 T_{Enc}/T_{Dec} + 6 T_H$	$T S_{Dec} + T S_{MAC} + T S_H$	$T_{Enc} + T_{MAC} + 4 T_H$
A.K.Das's [81]	$2 T_{FE} + T_{Enc} + 10 T_H$	$T S_{Dec} + 2 T S_H$	$2 T_{Enc}/T_{Dec} + 5 T_H$
Our Proposed	$2 T_{FE} + 9 T_H + T_{M1}$	$2 T S_H + T S_{M1}$	$5 T_H$

Note:  $T_H, T_{FE}, T_{Enc}, T_{Dec}, T_{BFE}, T_{MAC}, T_{M1}$  indicates the time required to perform secure hashing, Gen(.) Rep(.), encryption, decryption, bio-metric feature extraction and message authentication code operation, matrix row-column multiplication respectively for User and Gateway Node.  $T S_H, T S_{Dec}, T S_{MAC}, T S_{M1}$  indicates the time required to perform secure hashing, decryption, message authentication code operation and matrix row-column multiplication respectively for Sensor Node.

## **5.6 Summary**

In this chapter, we have first discussed the security issues involve in sensor nodes of WSNs & IoT and proposed a user validation, session key sharing protocol using smart card, fuzzy extractor, matrix decomposition operation. Afterwards, we have performed the security analysis and verification using a widely accepted and robust tools AVISPA and Scyther. To ensure the correctness of the security features involved in the protocol, we have performed the logical verification using BAN logic. Finally, we did the comparative analysis of our protocol with other existing protocols based on security features and computational overhead which indicates that our protocol is secure and efficient.

## **Chapter 6**

# **Chinese Remainder Theorem based User Authentication and Key Establishment Protocol for WSNs & IoT**

“Confidentiality is a virtue of the loyal, as loyalty is the virtue of faithfulness” Edwin  
Louis Cole

Because of resource and computing constraints in the WSNs & IoT, a secure, rapid and cost-effective user authentication protocol needs to be designed. Therefore, this chapter presents an efficient user authentication and key establishment protocol for WSNs & IoT based on Chinese Remainder Theorem (CRT). It briefly describes the problem, security challenges and contributions made thereof. Later, it elaborates the proposed protocol, and then, the security and computational overhead are analyzed.

### **6.1 Introduction and Problem Definition**

Authenticated querying is one of the prominent requirements of Internet of Things (IoT) or wireless networks of sensor devices to resist unauthorized users from accessing real time and confidential data. The Chinese Remainder Theorem (CRT) declares that if one associates the remainders of the Euclidean division of an integer



$n$  by several integers, then one can determine the remainder of the division of  $n$  by the product of these integers uniquely, under the provision that the divisors are pairwise coprime. In this chapter, we propose an efficient authenticated key exchange mechanism using the concepts of Fuzzy Extractor (explained in Section 1.3.5) and Chinese Remainder Theorem (described in Section 1.3.6). Afterwards, we perform the security analysis of our protocol using widely accepted automated verification tools such as AVISPA and Scyther. Then, we perform logical verification using BAN Logic. Finally, we do the computational analysis, and we demonstrate the comparative analysis with respect to computational overhead and security features. The necessary notations used in the subsequent sections of this Chapter are described in Table 1.

**Problem Definition:** The problem definition of this chapter is as follows:

Design and analysis of secure and efficient multi-factor user authentication and session key establishment protocol for resource constrained WSNs & IoT which does not need any obligation of the gateway node during the authentication and key establishment phase, provides the major security features (such as mutual authentication, secure session key establishment, confidentiality, integrity, freshness etc.) and prevents the major security attacks (such as stolen smart card, user impersonation, sensor node impersonation, etc.) with minor computational overhead.

## 6.2 Our Contributions

In this chapter, we introduce a novel user authentication protocol based on Chinese Remainder Theorem (CRT) to implement user access to the real-time data by immediately validating the user at the node level and also getting it feasible for users to communicate with the nodes to have replied to their queries. Our proposed protocol has the following attractive features:

- The proposed protocol has the benefit that no time synchronization is needed and then the sensor can authenticate messages immediately without buffering data packets and it does not need any obligation of the gateway node during the authentication and session key establishment phase.

- We have analysed the security features of our protocol using the widely-accepted AVISPA and Scyther tool which assures that the protocol is secure, robust and feasible for the applications of WSNs & IoT.

## 6.3 Proposed protocol

Our proposed protocol involves multiple phases. The following subsection explains the pre-deployment phase and Table 6.1, 6.2, 6.3 describe the registration, authenticated key exchange, user's credentials update phases respectively.

### 6.3.1 Pre-Deployment Phase

*GW**N* generates a key  $r_{SN_j}$  for each sensor node  $SN_j$  and a key  $r_{U_i}$  for each user  $U_i$ , where  $r_{SN_j}$  and  $r_{U_i}$  are relatively prime integers. *GW**N* generates a system of simultaneous congruence (considering Chinese Remainder Theorem as described in section 1.3.6) such as :

$$\begin{aligned} X_{old} &\equiv x_i^{old} \pmod{r_{U_i}}, X_{old} \equiv x_i^{old} \pmod{r_{SN_j}}, \\ X_{new} &\equiv x_i^{new} \pmod{r_{U_i}}, X_{new} \equiv x_i^{new} \pmod{r_{SN_j}} \end{aligned}$$

### 6.3.2 Registration Phase

To get registered by the *GW**N*, an authentic user  $U_i$  chooses his/her identity  $ID_{U_i}$ , password  $PW_{U_i}$  and biometric information  $BIO_i$  as a input for the  $Gen()$  function of fuzzy extractor. Then,  $U_i$  and *GW**N* follows the steps 1,2,3 consecutively as proposed in Table 6.1.

### 6.3.3 Authenticated Key Establishment Phase

For authenticated key establishment,  $U_i$  provides  $ID_{U_i}$ ,  $PW_{U_i}$  and the noisy biometric information  $BIO'_i$  as a input to the  $Rep()$  function of the fuzzy extractor. Then,  $U_i$ , *GW**N* and  $SN_j$  follows the steps 4,5,6,7,8,9 consecutively as proposed in Table 6.2.

Table 6.1: User Registration Phase

Step 1: for user $\mathcal{U}_i$	Step 2: for gateway node $GWN$
$\mathcal{U}_i$ Selects $ID_{\mathcal{U}_i}, PW_{\mathcal{U}_i}$ and imprints the biometric information $BIO_i$ . Computes $(\sigma_i, \tau_i) = Gen(BIO_i)$ and evaluates $IPB_i = h(ID_{\mathcal{U}_i}    PW_{\mathcal{U}_i}    h(\sigma_i))$ . Transmits $\langle ID_{\mathcal{U}_i}, IPB_i \rangle$ to $GWN$ $\xrightarrow{\text{SecureChannel}}$	$GWN$ computes $\alpha = IPB_i \oplus r_{\mathcal{U}_i}$ , $\beta = h(IPB_i    r_{\mathcal{U}_i})$ , $\gamma = h(ID_{\mathcal{U}_i} \oplus IPB_i) \oplus X_{old}$ . $\xleftarrow[\text{SecureChannel}]{GWN \text{ sends } \langle \alpha, \beta, \gamma \rangle \text{ to } \mathcal{U}_i}$
Step 3: for user $\mathcal{U}_i$	
$\mathcal{U}_i$ stores $h(), Gen(), Rep(), \alpha, \beta, \gamma, \tau_i, \mathcal{T}$ into $SC_{\mathcal{U}_i}$	

**Step A1:**  $\mathcal{U}_i$  inserts  $SC_{\mathcal{U}_i}$  into the card reader and inputs  $ID_{\mathcal{U}_i}, PW_{\mathcal{U}_i}, BIO_i$ . Then, evaluates  $\sigma'_i = Rep(BIO'_i, \tau_i)$ ,  $IPB'_i = h(ID_{\mathcal{U}_i} || PW_{\mathcal{U}_i} || h(\sigma'_i))$ ,  $\beta' = h(IPB'_i || r'_{\mathcal{U}_i})$ . If  $(\beta' = \beta)$ , evaluates  $X_{old} = \gamma \oplus h(ID_{\mathcal{U}_i} \oplus IPB_i)$ ,  $x_i^{old} = X_{old} \bmod r'_{\mathcal{U}_i}$ ,  $m_1 = h(x_i^{old} || T_{\mathcal{U}_i})$ ,  $m_2 = h(ID_{\mathcal{U}_i} || ID_{SN_j} || r'_{\mathcal{U}_i} || T_{\mathcal{U}_i})$ . Constructs the message  $M_1 = \langle ID_{\mathcal{U}_i}, ID_{SN_j}, X_{old}, m_1, m_2, T_{\mathcal{U}_i} \rangle$ . Finally,  $\mathcal{U}_i$  transmits  $\langle M_1 \rangle$  to  $GWN$ .

**Step A2:** If  $(T' - T_{\mathcal{U}_i} \leq \Delta T)$ ,  $SN_j$  computes  $x_i^{old'} \equiv X_{old} \bmod r_{SN_j}$ ,  $m'_1 = h(x_i^{old'} || T_{\mathcal{U}_i})$ . If  $m'_1 = m_1$ ,  $SN_j$  computes  $m_3 = h(ID_{SN_j} || r_{SN_j} || m_2 || T_{SN_j})$ ,  $M_2 = \langle ID_{\mathcal{U}_i}, ID_{SN_j}, m_3, T_{\mathcal{U}_i}, T_{SN_j} \rangle$ . Finally,  $SN_j$  sends  $M_2$  to  $GWN$ .

**Step A3:** If  $(T'' - T_{SN_j} \leq \Delta T)$ ,  $GWN$  computes  $m'_2 = h(ID_{\mathcal{U}_i} || ID_{SN_j} || r_{\mathcal{U}_i} || T_{\mathcal{U}_i})$ ,  $m'_3 = h(ID_{SN_j} || r_{SN_j} || m'_2 || T_{SN_j})$ . If  $m'_2 = m_2$  and  $m'_3 = m_3$ ,  $GWN$  computes  $m_4 = h(ID_{\mathcal{U}_i} || ID_{SN_j} || X_{new} || x_i^{new} || T_{\mathcal{U}_i} || T_{SN_j} || T_{GWN})$ ,  $M_3 = \langle X_{new}, m_4, T_{GWN} \rangle$ . Finally,  $GWN$  sends  $M_3$  to  $SN_j$ .

**Step A4:** After receiving message  $M_3$ ,  $SN_j$  computes  $x_i^{new} \equiv X_{new} \bmod r_{SN_j}$ ,  $m'_4 = h(ID_{\mathcal{U}_i} || ID_{SN_j} || X_{new} || x_i^{new} || T_{\mathcal{U}_i} || T_{SN_j} || T_{GWN})$ . If  $m'_4 = m_4$ ,  $SN_j$  evaluates  $K = h(T_{\mathcal{U}_i} || T_{SN_j} || x_i^{new})$ ,  $m_5 = h(K)$ , constructs a message  $M_4 = \langle X_{new}, m_4, m_5, T_{SN_j}, T_{GWN} \rangle$ . Finally,  $SN_j$  sends  $M_4$  to  $\mathcal{U}_i$ .

**Step A5:** If  $(T''' - T_{GWN} \leq \Delta T)$ ,  $\mathcal{U}_i$  evaluates  $x_i^{new} \equiv X_{new} \bmod r_{SN_j}$ ,  $m_4'' =$

$h(ID_{\mathcal{U}_i} || ID_{SN_j} || X_{new} || x_i^{new} || T_{\mathcal{U}_i} || T_{SN_j} || T_{GWN})$ . If  $(m_4'' = m_4)$ ,  $\mathcal{U}_i$  computes  $K' = h(T_{\mathcal{U}_i} || T_{SN_j} || x_i^{new})$ ,  $m_5 = h(K')$ . If  $(m_5' = m_5)$ ,  $\mathcal{U}_i$  establishes the session key  $K'$  with  $SN_j$ . Otherwise, reject  $\mathcal{U}_i$ .

**Step A6:** The user  $U_i$  makes  $X_{old} = X_{new}$  and stores  $\gamma = h(ID_{\mathcal{U}_i} \oplus IPB_i) \oplus (X_{old} = X_{new})$  into  $SC_{\mathcal{U}_i}$

The sequence diagram of the message transmission for the user registration, authentication and key establishment phase is shown in the following Figure 6.1.

### 6.3.4 User's Credential Update Phase

For user  $\mathcal{U}_i$ , credential update is required to ensure that an adversary  $\mathcal{A}$  can not acquire or snoop the user's secret credentials like password  $PW_{\mathcal{U}_i}$  and biometric information  $BIO_i$ . To update the credential,  $\mathcal{U}_i$  follows the step as proposed in Table 6.3.

## 6.4 Security Analysis

To verify the security features present in our protocol, we first perform the informal analysis considering major and minor attacks in WSNs. Afterward, we implement our protocol using Security Protocol Description Language and evaluate our security claims using Synter tool [19]. For automated validation of the protocol using AVISPA tool [7], we use High-Level Protocols Specification Language. Finally, we do the logical verification of the protocol using BAN logic [20].

### 6.4.1 Informal Security Analysis

The informal security analysis indicates that our protocol is designed to withstand the popular security attacks as follows:

- **Exhausting Constrained Resources** To avoid false message flooding (which exhausts the resources of WSNs), we eliminate the illegitimate users at the initial level (i.e. at sensor node itself) of message transmission. For a sensor node  $SN_j$ , the energy required for computation is less compared to data transmission

**Table 6.2:** Authenticated Key Exchange Phase

<p><b>Step 4:</b> for <math>\mathcal{U}_i</math></p> <p><math>\mathcal{U}_i</math> inserts <math>SC_{\mathcal{U}_i}</math> into the card reader and inputs <math>ID_{\mathcal{U}_i}, PW_{\mathcal{U}_i}, BIO_i</math>. Then evaluates <math>\sigma'_i = Rep(BIO'_i, \tau_i)</math>, <math>IPB'_i = h(ID_{\mathcal{U}_i}    PW_{\mathcal{U}_i}    h(\sigma'_i))</math>, <math>\beta' = h(IPB'_i    r'_{\mathcal{U}_i})</math></p> <p><b>if</b> <math>\beta' = \beta</math> <b>then</b></p> <p>    Evaluates <math>X_{old} = \gamma \oplus h(ID_{\mathcal{U}_i} \oplus IPB_i)</math>,  <math>x_i^{old} = X_{old} \bmod r'_{\mathcal{U}_i}</math>, <math>m_1 = h(x_i^{old}    T_{\mathcal{U}_i})</math>,  <math>m_2 = h(ID_{\mathcal{U}_i}    ID_{SN_j}    r'_{\mathcal{U}_i}    T_{\mathcal{U}_i})</math>. Construct the mes-  sage <math>M_1 = \langle ID_{\mathcal{U}_i}, ID_{SN_j}, X_{old}, m_1, m_2, T_{\mathcal{U}_i} \rangle</math></p> <p>    <math>\mathcal{U}_i</math> transmits <math>\langle M_1 \rangle</math> to <math>GW_N</math></p> <p style="text-align: center;">ViaPublicChannel</p> <p><b>end</b></p> <p><b>else</b></p> <p>    Reject <math>\mathcal{U}_i</math></p> <p><b>end</b></p>	<p><b>Step 5:</b> for <math>SN_j</math></p> <p><b>if</b> <math>T' - T_{\mathcal{U}_i} \leq \Delta T</math> <b>then</b></p> <p>    Find <math>x_i^{old'} \equiv X_{old} \bmod r_{SN_j}</math>, <math>m'_1 = h(x_i^{old'}    T_{\mathcal{U}_i})</math></p> <p>    <b>if</b> <math>m'_1 = m_1</math> <b>then</b></p> <p>        Computes <math>m_3 = h(ID_{SN_j}    r_{SN_j}    m_2    T_{SN_j})</math>,  <math>M_2 = \langle ID_{\mathcal{U}_i}, ID_{SN_j}, m_3, T_{\mathcal{U}_i}, T_{SN_j} \rangle</math></p> <p>    <b>end</b></p> <p>    <b>else</b></p> <p>        Reject <math>\mathcal{U}_i</math></p> <p>    <b>end</b></p> <p>    <math>SN_j</math> sends <math>M_2</math> to <math>GW_N</math></p> <p style="text-align: center;">ViaPublicChannel</p> <p><b>end</b></p> <p><b>else</b></p> <p>    Reject <math>\mathcal{U}_i</math></p> <p><b>end</b></p>
<p><b>Step 6:</b> for <math>GW_N</math></p> <p><b>if</b> <math>T'' - T_{SN_j} \leq \Delta T</math> <b>then</b></p> <p>    Computes <math>m'_2 = h(ID_{\mathcal{U}_i}    ID_{SN_j}    r_{\mathcal{U}_i}    T_{\mathcal{U}_i})</math>,  <math>m'_3 = h(ID_{SN_j}    r_{SN_j}    m'_2    T_{SN_j})</math></p> <p>    <b>if</b> <math>m'_2 = m_2</math> and <math>m'_3 = m_3</math> <b>then</b></p> <p>        Compute <math>m_4 = h(ID_{\mathcal{U}_i}    ID_{SN_j}    X_{new}   </math>  <math>x_i^{new}    T_{\mathcal{U}_i}    T_{SN_j}    T_{GW_N})</math>,  <math>M_3 = \langle X_{new}, m_4, T_{GW_N} \rangle</math></p> <p>        <math>GW_N</math> sends <math>M_3</math> to <math>SN_j</math></p> <p style="text-align: center;">ViaPublicChannel</p> <p>    <b>end</b></p> <p>    <b>else</b></p> <p>        Reject <math>\mathcal{U}_i</math></p> <p>    <b>end</b></p> <p><b>end</b></p> <p><b>else</b></p> <p>    Reject <math>\mathcal{U}_i</math></p> <p><b>end</b></p>	<p><b>Step 7:</b> for <math>SN_j</math></p> <p>Find <math>x_i^{new} \equiv X_{new} \bmod r_{SN_j}</math>, <math>m'_4 =</math>  <math>h(ID_{\mathcal{U}_i}    ID_{SN_j}    X_{new}    x_i^{new}    T_{\mathcal{U}_i}    T_{SN_j}    T_{GW_N})</math></p> <p><b>if</b> <math>m'_4 = m_4</math> <b>then</b></p> <p>    Evaluates <math>K = h(T_{\mathcal{U}_i}    T_{SN_j}    x_i^{new})</math>, <math>m_5 = h(K)</math>,  Construct a message  <math>M_4 = \langle X_{new}, m_4, m_5, T_{SN_j}, T_{GW_N} \rangle</math></p> <p>    <math>SN_j</math> sends <math>M_4</math> to <math>\mathcal{U}_i</math></p> <p style="text-align: center;">ViaPublicChannel</p> <p><b>end</b></p> <p><b>else</b></p> <p>    Reject <math>\mathcal{U}_i</math></p> <p><b>end</b></p>
<p><b>Step 8:</b> for <math>\mathcal{U}_i</math></p> <p><b>if</b> <math>T'' - T_{GW_N} \leq \Delta T</math> <b>then</b></p> <p>    Evaluates <math>x_i^{new} \equiv X_{new} \bmod r_{SN_j}</math>, <math>m_4'' =</math>  <math>h(ID_{\mathcal{U}_i}    ID_{SN_j}    X_{new}    x_i^{new}    T_{\mathcal{U}_i}    T_{SN_j}    T_{GW_N})</math></p> <p>    <b>if</b> <math>m_4'' = m_4</math> <b>then</b></p> <p>        Computes <math>K' = h(T_{\mathcal{U}_i}    T_{SN_j}    x_i^{new})</math>, <math>m_5 =</math>  <math>h(K')</math></p> <p>        <b>if</b> <math>m'_5 = m_5</math> <b>then</b></p> <p>            Establish the session key <math>K'</math> with <math>SN_j</math></p> <p>        <b>end</b></p> <p>        <b>else</b></p> <p>            Reject <math>\mathcal{U}_i</math></p> <p>        <b>end</b></p> <p>    <b>end</b></p> <p><b>end</b></p>	<p><b>Step 9:</b> for <math>\mathcal{U}_i</math></p> <p>Make <math>X_{old} = X_{new}</math> and  Store <math>\gamma = h(ID_{\mathcal{U}_i} \oplus IPB_i) \oplus (X_{old} = X_{new})</math> into <math>SC_{\mathcal{U}_i}</math></p>

**Figure 6.1:** Sequence Diagram 4 for Registration, Authentication and Key Establishment

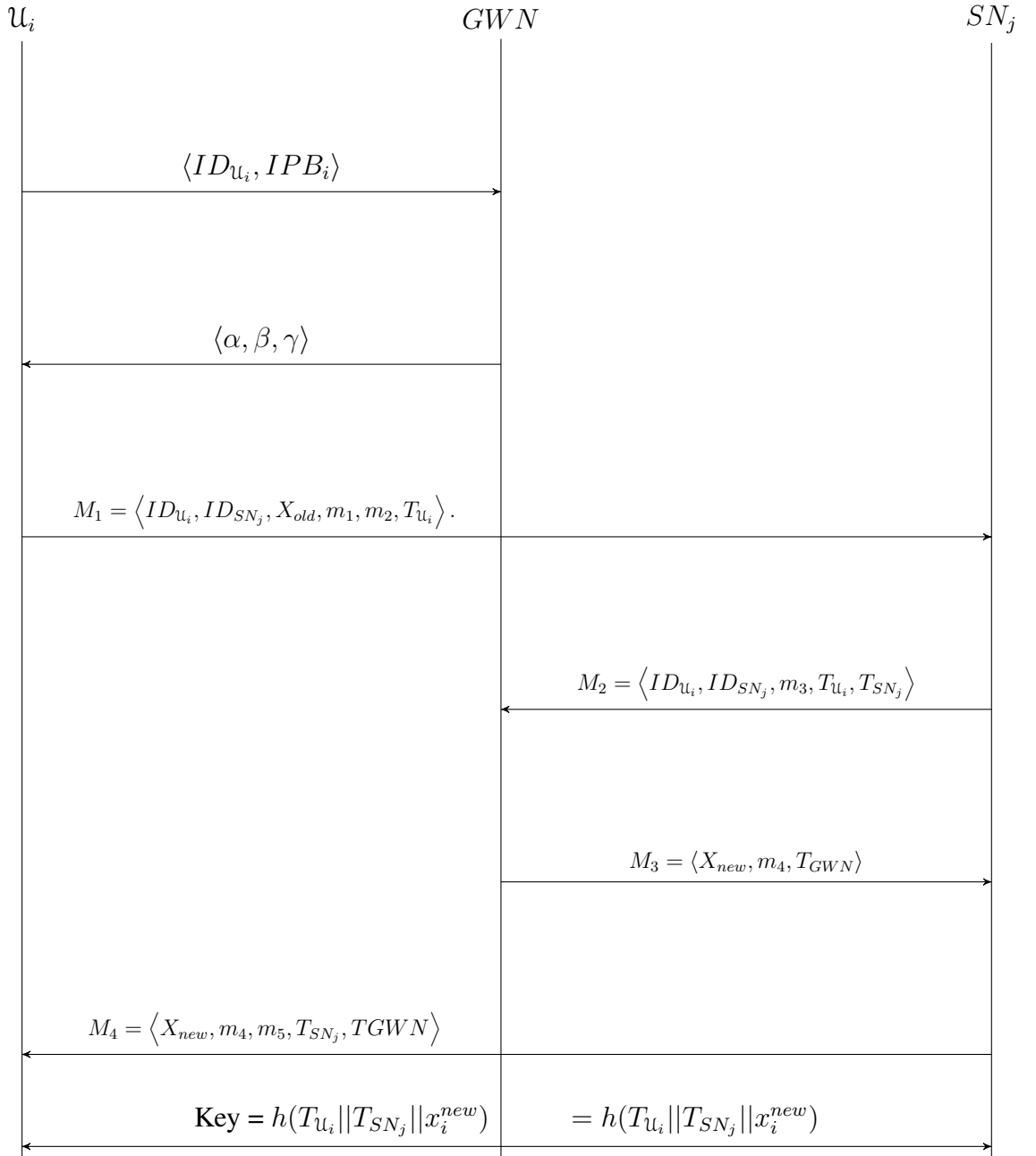


Table 6.3: User's Credential Update Phase

User ( $\mathcal{U}_i$ )
$\mathcal{U}_i$ puts $SC_{\mathcal{U}_i}$ into card reader and provides $ID_{\mathcal{U}_i}, PW_{\mathcal{U}_i}, BIO_i$ , It produces $\sigma'_i = Rep(BIO'_i, \tau_i)$ , $IPB'_i = h(ID_{\mathcal{U}_i}    PW_{\mathcal{U}_i}    h(\sigma'_i))$ , $r'_{\mathcal{U}_i} = \alpha \oplus IPB'_i$ , $\beta' = h(IPB'_i    r'_{\mathcal{U}_i})$ , <b>if</b> $\beta = \beta'$ <b>then</b> Computes $X'_{old} = \gamma \oplus h(ID_{\mathcal{U}_i}    IPB'_i)$ , $\mathcal{U}_i$ provides new password $PW_{\mathcal{U}_i}^{new}$ and bio-metric information $BIO_i^{new}$ . $(\sigma_i^{new}, \tau'_i) = Gen(BIO_i^{new})$ , $IPB_i^{new} = h(ID_{\mathcal{U}_i}    PW_{\mathcal{U}_i}^{new}    h(\sigma_i^{new}))$ $\alpha^{new} = IPB_i^{new} \oplus r_{\mathcal{U}_i}$ $\beta^{new} = h(IPB_i^{new}    r_{\mathcal{U}_i})$ $\gamma^{new} = h(ID_{\mathcal{U}_i}    IPB_i^{new}) \oplus X'_{old}$ Replace $\alpha, \beta, \gamma$ of $SC_{\mathcal{U}_i}$ with $\alpha^{new}, \beta^{new}, \gamma^{new}$ respectively. <b>end</b> <b>else</b> Reject $\mathcal{U}_i$ <b>end</b>

(the energy required in 2090 clock cycles of computation is equivalent to the energy required for transmitting 1-bit data [119] ). We verify the correctness of  $m_1 = h(x_i^{old} || T_{\mathcal{U}_i})$  at sensor node, where  $x_i^{old} \equiv \text{mod } r_{SN_j}$ , the correct value of  $m_1$  ensures the user belongs to the authorized group. Energy required in transceiving and receiving 1-bit data (at the data rate of 12.4 Kb/s) are  $59.2 \mu \text{ Joule}$  ,  $28.6 \mu \text{ Joule}$  respectively [119] . Furthermore, we assume  $N, n$  are the size and density (total number of nodes within the circular area with radius equal to the communication range of sensor node) of WSNs. If an illegitimate user  $\mathcal{A}$  is not eliminated or filtered at initial level,  $\mathcal{A}$  can consume total energy equal to  $E$  by sending a message  $M'_1 = \langle ID_{\mathcal{U}_i}, ID_{SN_j}, hash\_value/signature, T_{\mathcal{U}_i} \rangle$  of size  $S$  bytes. Where  $E$  can be evaluated as follows :

$$E = N \left( S \times (59.2 + 28.6n) \right) \mu \text{ Joule}.$$

But in our protocol we eliminate  $\mathcal{A}$  at the initial level which saves the total energy

of  $(E - E_h)\mu$  Joule . Where  $E_h$  is the energy required in computing and verifying the hash value  $m_1 = h(x_i^{old} || T_{\mathcal{U}_i})$ . The energy required by SHA-1 hash function is  $5.9\mu$  Joule/byte [119] . Hence, our protocol withstand the energy exhausting attacks.

- **Stolen Smart Card Attack** To defend the attacks based on stolen  $SC_{\mathcal{U}_i}$ , we keep the secret credentials of  $\mathcal{U}_i$  in  $SC_{\mathcal{U}_i}$  protected with fuzzy extractor mechanism. An adversary  $\mathcal{A}$  can extract the value of  $\alpha, \beta, \gamma$  from stolen  $SC_{\mathcal{U}_i}$  using power analysis attacks. But it is hard find out the value of secret credentials such as :  $r_{\mathcal{U}_i}, \sigma_i, PW_{\mathcal{U}_i}$  for an adversary  $\mathcal{A}$  without knowing the bio-metric information and password of the user  $\mathcal{U}_i$ . Therefore, our protocol resist the stolen  $SC_{\mathcal{U}_i}$  attacks.
- **Man-in-the-middle attack** To avoid the Man-in-the-middle attack, we ensure mutual authentication among the  $\mathcal{U}_i, SN_j, GWN$  by verifying the secret parameters such as  $m_1, m_2, m_3, m_4$ . The parameters  $m_1, m_2, m_3, m_4$  also ensures the message integrity.
- **Replay Attack**  
Verification of timestamps  $T_{\mathcal{U}_i}, T_{SN_j}, T_{GWN}$  along with their hashed values protects the replay attacks.
- **Impersonation Attack**  
The verification of legitimate bio-metric information  $BIO'_i$  (using fuzzy extractor) and password  $PW_{\mathcal{U}_i}$  at the time of user authentication ensures that an adversary  $\mathcal{A}$  can not impersonate the user  $\mathcal{U}_i$ .

### 6.4.2 Security verification using Scyther and AVISPA tool :

We specify our protocol using Security Protocol Description Language (spdl) based on the operational semantics of Scyther tool. Table 6.4 represents the spdl specification of our protocol. The result of security verification using Scyther tool is shown Figure 6.2 . The result indicates that no attacks found on all the claims which we specified for the three roles  $\mathcal{U}_i, GWN, SN_j$ .

The result obtained (Figure 6.3 ) using OFMC back-ends of AVISPA tool indicates that our protocol is safe from Dolev-Yao [110] intruder model.



**Table 6.4:** The spdl specification of the proposed protocol

<pre> hashfunction h; const XOR : Function; const Modulo: Function ; const Gen : Function ; const Rep : Function; protocol Protocol(Ui, GWN, SNj) { macro SIGi = Gen(Bi); macro IPBi = h(IDui, PWui, h(SIGi)); macro SIGi' = Rep(Bi', TAUi); macro IPBi' = h(IDui, PWui, h(SIGi')); macro Alpha = XOR( IPBi,Rui ); macro Beta = h(IPBi, Rui ); macro Gamma = XOR(h(IDui,IPBi), Xold ); macro xiold = Modulo(Xold,Rui'); macro xiold' = Modulo(Xold,Rsnj); macro xnew = Modulo(Xnew,Rsnj); macro xnew' = Modulo(Xnew,Rui'); macro m1 = h(xiold, Tui); macro m2 = h(IDui, IDsnj,Rui',Tui); macro m3 = h(IDsnj, Rsnj, m2,Tui, Tsnj); macro m4 = h(IDui,IDsnj,Xnew, xnew,Tui,Tsnj,Tgwn); macro m5 = h(h(Tui,Tsnj,xnew)); macro m1' = h(xiold', Tui); macro m2' = h(IDui, IDsnj,Rui,Tui); macro m3' = h(IDsnj, Rsnj, m2',Tui, Tsnj); macro m4' = h(IDui,IDsnj,Xnew, xnew',Tui,Tsnj,Tgwn); macro m5' = h(h(Tui,Tsnj,xnew)); role Ui { var Tsnj ,Tgwn : Nonce; fresh Tui: Nonce; const IDui, PWui, Bi, Bi', IDsnj, Xold, xiold,xiold', Xnew,xnew',xnew , Rui, Rui',Rsnj, TAUi: Ticket; send_1(Ui, GWN, IDui, IPBi); recv_2(GWN, Ui,Alpha,Beta,Gamma); send_3(Ui, SNj, IDui, IDsnj, Xold,m1,m2,Tui); recv_6(SNj, Ui, Xnew,m4,m5,Tsnj,Tgwn); match(m4, m4'); match(m5, m5'); </pre>	<pre> claim_Ui1(Ui,Secret,Bi); claim_Ui2(Ui,Secret,PWui); claim_Ui3(Ui,Secret,Rui'); claim_Ui4(Ui,Secret,xiold); claim_Ui5(Ui,Secret,xnew'); claim_Ui6(Ui,SKR,h(Tui,Tsnj,xnew)); claim_Ui7(Ui,Niagree); claim_Ui78(Ui,Nisynch); } role GWN { fresh Tgwn: Nonce; var Tui,Tsnj : Nonce; const IDui, PWui, Bi, Bi', IDsnj, Xold, Xnew,Rui, Rui',Rsnj, TAUi: Ticket; recv_1(Ui, GWN, IDui, IPBi); send_2(GWN, Ui, Alpha, Beta, Gamma); recv_4(SNj, GWN, IDui,IDsnj,m3,Tui,Tsnj); match (m2', m2); match (m3', m3); send_5(GWN, SNj, Xnew, m4, Tgwn); claim_GWN1(GWN,Secret,xnew); claim_GWN2(GWN,Secret,Rsnj); claim_GWN3(GWN,Secret,Rui'); } role SNj { } } </pre>
--	---

## 6.4 Security Analysis

Scyther results : verify					
Claim				Status	Comments
Protocol	Ui	Protocol,Ui1	Secret Bi	Ok	No attacks within bounds.
		Protocol,Ui2	Secret PWui	Ok	No attacks within bounds.
		Protocol,Ui3	Secret Rui'	Ok	No attacks within bounds.
		Protocol,Ui4	Secret Modulo(Xold,Rui')	Ok	No attacks within bounds.
		Protocol,Ui5	Secret Modulo(Xnew,Rui')	Ok	No attacks within bounds.
		Protocol,Ui6	SKR h(Tui,Tsnj,Modulo(Xnew,Rsnj))	Ok	No attacks within bounds.
		Protocol,Ui7	Niagree	Ok	No attacks within bounds.
		Protocol,Ui78	Nisynch	Ok	No attacks within bounds.
GWN		Protocol,GWN1	Secret Modulo(Xnew,Rsnj)	Ok	No attacks within bounds.
		Protocol,GWN2	Secret Rsnj	Ok	No attacks within bounds.
		Protocol,GWN3	Secret Rui'	Ok	No attacks within bounds.
SNj		Protocol,SNj1	Secret Rsnj	Ok	No attacks within bounds.
		Protocol,SNj2	Secret Tsnj	Ok	No attacks within bounds.
		Protocol,SNj3	Secret Modulo(Xnew,Rsnj)	Ok	No attacks within bounds.
		Protocol,SNj4	SKR h(Tui,Tsnj,Modulo(Xnew,Rsnj))	Ok	No attacks within bounds.

Done.

Figure 6.2: Result obtained using Scyther tool.

```
% OFMC
% Version of 2006/02/13
SUMMARY
SAFE
DETAILS
BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
/home/cmb-lap-22/Downloads/span/testsuite/results/proto5.if
GOAL
as_specified
BACKEND
OFMC
COMMENTS
STATISTICS
parseTime: 0.00s
searchTime: 6.07s
visitedNodes: 2454 nodes
depth: 9 plies
```

Figure 6.3: Result obtained using AVISPA tool.

### 6.4.3 Logical verification using BAN logic

In this subsection, we use BAN logic [20] to verify the freshness of time-stamp to avoid replay attack, and we validate the message origin to achieve authenticity. The notations we use for logical verification is shown in Table 1

1. Verification of freshness of  $T_{\mathcal{U}_i}, T_{SN_j}, T_{GWN}$  (using message - meaning and nonce verification rule of BAN logic):

- $$\frac{GWN| \equiv \mathcal{U}_i \stackrel{r_{\mathcal{U}_i}}{\rightleftharpoons} GWN, GWN \triangleleft < T_{\mathcal{U}_i} >_{r_{\mathcal{U}_i}}}{GWN| \equiv \mathcal{U}_i | \sim T_{\mathcal{U}_i}}$$

That is, if  $GWN$  believes the secret  $r_{\mathcal{U}_i}$  is shared with  $\mathcal{U}_i$  and sees  $< T_{\mathcal{U}_i} >_{r_{\mathcal{U}_i}}$ , then  $GWN$  believe  $\mathcal{U}_i$  once said  $T_{\mathcal{U}_i}$

- $$\frac{GWN| \equiv \#(T_{\mathcal{U}_i}), GWN| \equiv \mathcal{U}_i | \sim T_{\mathcal{U}_i}}{GWN| \equiv \mathcal{U}_i | \equiv T_{\mathcal{U}_i}}$$

That is, if  $GWN$  believes  $T_{\mathcal{U}_i}$  is fresh and  $GWN$  believes  $\mathcal{U}_i$  once said  $T_{\mathcal{U}_i}$ , then  $GWN$  believe  $\mathcal{U}_i$  believes on  $T_{\mathcal{U}_i}$

- $$\frac{GWN| \equiv SN_j \stackrel{r_{SN_j}}{\rightleftharpoons} GWN, GWN \triangleleft < T_{SN_j} >_{r_{SN_j}}}{GWN| \equiv SN_j | \sim T_{SN_j}}$$

That is, if  $GWN$  believes the secret  $r_{SN_j}$  is shared with  $SN_j$  and sees  $< T_{SN_j} >_{r_{SN_j}}$ , then  $GWN$  believe  $SN_j$  once said  $T_{\mathcal{U}_i}$

- $$\frac{GWN| \equiv \#(T_{SN_j}), GWN| \equiv SN_j | \sim T_{SN_j}}{GWN| \equiv SN_j | \equiv T_{SN_j}}$$

That is, if  $GWN$  believes  $T_{SN_j}$  is fresh and  $GWN$  believes  $SN_j$  once said  $T_{SN_j}$ , then  $GWN$  believe  $SN_j$  believes on  $T_{SN_j}$

- $$\frac{\mathcal{U}_i | \equiv \#(T_{SN_j}), \mathcal{U}_i | \equiv SN_j | \sim T_{SN_j}}{\mathcal{U}_i | \equiv SN_j | \equiv T_{SN_j}}$$

That is, if  $\mathcal{U}_i$  believes  $T_{SN_j}$  is fresh and  $\mathcal{U}_i$  believes  $SN_j$  once said  $T_{SN_j}$ , then  $\mathcal{U}_i$  believe  $SN_j$  believes on  $T_{SN_j}$

2. Verification of the authenticity of the message  $m_2$  by  $GW N$  (using message - meaning rule)

$$\bullet \frac{GW N | \equiv \mathcal{U}_i \stackrel{r_{\mathcal{U}_i}}{\rightleftharpoons} GW N, \mathcal{U}_i \triangleleft \langle m_2 \rangle_{r_{\mathcal{U}_i}}}{GW N | \equiv \mathcal{U}_i | \sim m_2}$$

That is, if  $GW N$  believes the secret  $r_{\mathcal{U}_i}$  is shared with  $\mathcal{U}_i$  and sees  $\langle m_2 \rangle_{r_{\mathcal{U}_i}}$ , then  $GW N$  believe  $\mathcal{U}_i$  once said  $m_2$

## 6.5 Performance Comparison

Table 6.5 shows the comparison based on security features which indicates that our protocol is relatively secure compared to the existing protocol. Table 6.6 represent the computational cost comparison, it shows that our protocol is suitable for secure WSNs and IoT.

**Table 6.5:** Comparisons of Security Features

<b>Security Feature</b>	Sun et al. [108]	Xue et al. [79]	Jiang et al. [80]	Althobaiti et al. [59]	Our protocol
$SF_1$	Yes	No	Yes	Yes	Yes
$SF_2$	No	No	No	No	Yes
$SF_3$	No	No	No	No	Yes
$SF_4$	No	No	No	Yes	Yes
$SF_5$	No	No	No	No	Yes
$SF_6$	Yes	No	No	Yes	Yes

Note:  $SF_1, SF_2, SF_3, SF_4, SF_5$  are the security features.  $SF_1$  resist the attack based on stolen smart card,  $SF_2$  indicates the secure password updating,  $SF_3$  represents secure bio-metric information updating,  $SF_4$  indicates non-repudiation,  $SF_5$  offers formal security analysis,  $SF_6$  represents no privileged-insider attack

---

**Table 6.6:** Computational Cost Comparison

Protocols	Computational Overhead on $\mathcal{U}_i, SN_j, GWN$		
	$\mathcal{U}_i$	$SN_j$	$GWN$
Yoo et al.'s [78]	$7 T_H$	$2 T_{SH}$	$11 T_H$
Sun et al.'s [108]	$2 T_H$	$2 T_{SH}$	$7 T_H$
Xue et al.'s [79]	$12 T_H$	$6 T_{SH}$	$17 T_H$
Jiang et al.'s [80]	$8 T_H$	$5 T_{SH}$	$11 T_H$
Shi et al.'s [109]	$3 T_M + 5 T_H$	$2 T_{SM} + 3 T_{SH}$	$T_M + 4 T_H$
Choi et al.'s [1]	$3 T_M + 7 T_H$	$2 T_{SM} + 4 T_{SH}$	$T_M + 4 T_H$
A.K.Das's [81]	$2 T_{FE} + T_{ENC} + 10 T_H$	$T_{SDEC} + 2 T_{SH}$	$2 T_{ENC} / T_{DEC} + 5 T_H$
Proposed	$T_{FE} + 8 T_H + T_{MO}$	$5 T_{SH} + 2 T_{SMO}$	$3 T_H$

Note:  $T_H, T_{FE}, T_{ENC}, T_{DEC}, T_{BFE}, T_{MAC}, T_M, T_{MO}$  indicates the time required to perform secure hashing, Gen(.) / Rep(.), encryption, decryption, bio-metric feature extraction and message authentication code, scalar-point multiplication, mod operation respectively for User and Gateway Node.  $T_{SH}, T_{SDEC}, T_{SMAC}, T_{SM}, T_{SMO}$  indicates the time required to perform secure hashing, decryption, message authentication code, scalar-point multiplication, mod operation respectively for Sensor Node.

## 6.6 Summary

In this chapter, we have first discussed the security issues involved in sensor nodes of WSNs & IoT and identified vulnerabilities involve in existing authentication protocols. Based on the security requirement of WSNs & IoT, we have proposed an efficient authenticated key exchange mechanism using the concepts of the Fuzzy Extractor and Chinese Remainder Theorem. After that, we performed the security analysis of our protocol using widely accepted automated verification tools such as AVISPA and Scyther. Then, we performed logical verification using BAN Logic. Finally, we did the computational analysis, and we demonstrated the comparative analysis in respect of computational overhead and security features which indicate that our protocol is secure and effective.

## Chapter 7

# Bloom Filter based User Authentication and Key Establishment Protocol for WSNs & IoT

“I am not convinced that lack of encryption is the primary problem. The problem with the Internet is that it is meant for communications among non-friends.” Whitfield Diffie

This chapter presents an efficient user authentication and key establishment protocol for WSNs & IoT based on Bloom Filter. It concisely demonstrates the problem, the motivation behind the research work, security challenges and contributions made thereof. Afterwards, it elaborates the proposed protocol, and then, the relative security and computational overhead are analysed.

### 7.1 Introduction and Problem Definition

As the number of sensor nodes in some WSNs & IoT is large, multihop communication is preferred. Moreover, we need to eliminate unauthorized querying message transmission into the network so that it can withstand various security attacks in WSNs & IoT.

As the sensor nodes  $SN_j$  buffer all the messages obtained in one time interval, an adversary  $\mathcal{A}$  can flood the entire WSNs & IoT randomly. All it has to do is to claim

that the flooding messages belong to the current time interval which should be buffered for authentication until the next time interval. Since wireless message transmission is very expensive in WSNs & IoT, and WSNs & IoT are extremely energy constrained, the ability to flood the WSNs & IoT randomly could seed destructive Denial of Service (DoS) attack. Furthermore, this category of energy-exhaustion DoS attacks become more destructive in multiuser framework as the adversary  $\mathcal{A}$  now can have additional target points and hence new possibility to produce bogus messages without being recognized. Obviously, all these attacks are because of delayed authentication of the user's messages. Hence, in this chapter we aim to design a protocol which does not require any involvement of the gateway node during the authentication, key establishment phase and eliminate unauthorized querying message transmission at the initial level (i.e., at the sensor node itself). The basic notations used in the following sections of this Chapter are described in Table 1.

**Problem Definition:** The problem definition of this chapter is as follows:

Design and analysis of secure and efficient multi-factor user authentication and session key establishment protocol for resource constrained WSNs & IoT eliminates unauthorized querying message transmission at the initial level (i.e., at the sensor node itself) to avoid bogus message flooding from the sensor nodes to the gateway node (which exhausts the resources of WSNs & IoT), provides the major security features (such as mutual authentication, secure session key establishment, confidentiality, integrity, freshness etc.) and prevents the major security attacks (such as stolen smart card, user impersonation, sensor node impersonation, etc.) with minor computational overhead.

## 7.2 Our Contributions

In this chapter, we propose a novel user authentication protocol based on bloom filter to provide user access to the important real-time data by authorizing him/her directly at sensor node level and also making it possible for users to communicate with the sensor nodes in order to have quick responses to their queries. Our protocol has the following attractive properties:

- The novelty of our proposed protocol is, it eliminates unauthorized querying message transmission at the initial level (i.e., at the sensor node itself) to avoid bogus message flooding from the sensor nodes to the gateway node (which exhausts the resources of WSNs).
- It resists various security attacks. The resilience level against node compromise attack of our protocol is much higher than other existing protocols.
- It requires effective communication, computation, and storage overheads as compared to other existing protocols.
- It does not require any involvement of the gateway node during the authentication and key establishment phase.
- Higher security along with lower communication, computation and storage overheads make our protocol much suitable for practical applications of WSNs & IoT.
- It provides better security as compared with other related protocols, since it supports mutual authentication between the user  $\mathcal{U}_i$  and the sensor node  $SN_j$ , resists denial-of-service attack, privileged-insider attack, smart card breach attack and sensor node capture attack.
- It supports dynamic node addition after initial deployment of nodes in the network. The proposed protocol does not require to update information for new nodes addition in the user's smart card.
- It supports changing the user's password locally without the help of the  $GW N$ .
- It provides unconditional security against node capture attacks. That is, compromise of a sensor node  $SN_j$  does not reveal any secret information of other sensor nodes  $SN_k$  and it does not lead to compromise any other secure communication between the user  $\mathcal{U}_i$  and the non-compromised nodes  $SN_k$  in the network.
- It establishes a secret session key between the user and sensor node for future secret communication of the real-time data inside WSNs & IoT between them based on the established session key.
- We have analyzed the security of our proposed protocol using the formal and informal methods based on the random oracle model and AVISPA tool. The formal



security under the random oracle models reveals that our protocol is secure. Furthermore, we have simulated our protocol for the formal security validation using the widely-accepted AVISPA tool. We have implemented other protocol in the HLPSL language and then compared our protocol for formal security verification using AVISPA OFMC model checker with other existing protocols.

- We have performed the computational analysis of the proposed protocol which shows that our protocol is suitable for resource constrained sensor nodes like TelosB and MicaZ. The relative security and performance analysis results indicate that our protocol is more secure, efficient and robust in comparison to other existing protocol.
- In addition, we have compared the functionality provided by our protocol with other protocols. Overall, our proposed protocol has better performance than other existing protocols.

## 7.3 Proposed Protocol

Our proposed protocol is suitable for the WSNs and IoT in which the hop counts between the source sensor node and the gateway node is large and therefore requires high communication cost to transmit a data packet from the sensor of origin node to the gateway node. For the sensor nodes of WSNs and WSNs based IoT; the communication cost is always higher than the computational cost. Therefore, we have proposed a protocol which validates the users with efficient computation at sensor node itself and eliminates unauthorized querying data flooding from the sensor nodes to the gateway node to resist sensor node's resource exhausting attack. Our proposed protocol consists of following three essential phases:

### 7.3.1 Pre-Deployment Phase:

In this phase, the gateway node  $GWN$  generates and distributes some secret, public parameters to the user  $\mathcal{U}_i$  and the sensor node  $SN_j$  (before deployment of  $SN_j$ ) through a secure communication channel. The following steps P1, P2 and P3 (based on Bloom Filter as define in section 1.3.6) are used to implement this phase.

**Step P1:** In this phase, the *GW**N* chooses a generator point  $P$  of order  $q$  over  $E(F_p)$  and generates a set of private, public key pairs as  $(r_{U_1}, r_{U_1} \times P), (r_{U_2}, r_{U_2} \times P), \dots (r_{U_\alpha}, r_{U_\alpha} \times P)$ . Where  $r_{U_i} \in \mathbb{Z}_q^*, \forall i \in [1, \alpha]$ .

**Step P2:** The *GW**N* constructs a vector  $V = v_0, v_1, \dots, v_{\beta-1}$  using a set of hash functions  $H = \{h_1, h_2 \dots h_\gamma\}$ . Where the size of vector  $V$  is equal to  $\beta$ , the cardinality of the set  $H$  is equal to  $\gamma$  and

$$v_i = \begin{cases} 1, & \forall l \in [1, \gamma], t \in [1, \alpha], h_l(r_{U_t} \times P) = i \\ 0, & \text{otherwise} \end{cases}$$

**Step P3:** Afterwards, The *GW**N* loads the vector  $V$ , the set  $H$  and the key pair  $(r_{SN_j}, K_{SN_j} = r_{SN_j} \times P)$  into the memory of  $SN_j$ .

#### 7.3.2 User Registration Phase:

In this phase,  $\mathcal{U}_i$  selects its identity  $ID_{\mathcal{U}_i}$ , password  $PW_{\mathcal{U}_i}$ , bio-metric information  $BIO_{\mathcal{U}_i}$  and follows the procedure as described in step R1, R2, R3 and summarized in Table 7.1.

**Step R1:** In this step, the user  $\mathcal{U}_i$  chooses a unique identity  $ID_{\mathcal{U}_i}$ , selects a secret password  $PW_{\mathcal{U}_i}$  and imprints his/her bio-metric information  $BIO_{\mathcal{U}_i}$  into a smart card reader. Then,  $\mathcal{U}_i$  takes a generator function  $Gen()$  of fuzzy extractor and computes a pair of the secret and public parameter:  $(\sigma_i, \tau_i) = Gen(BIO_{\mathcal{U}_i})$ , Finally,  $\mathcal{U}_i$  evaluates  $IPB_i = h(ID_{\mathcal{U}_i} || PW_{\mathcal{U}_i} || \sigma_i)$  based on a secure one-way hash function  $h()$  and sends  $\langle ID_{\mathcal{U}_i}, IPB_i \rangle$  to the trusted gateway node *GW**N* using a secure communication channel.

**Step R2:** After receiving  $\langle ID_{\mathcal{U}_i}, IPB_i \rangle$ , the gateway node *GW**N* selects the private and public key pair  $r_{\mathcal{U}_i}, r_{\mathcal{U}_i} \times P$  corresponding to the user  $\mathcal{U}_i$ . Then, *GW**N* computes  $K_{\mathcal{U}_i} = r_{\mathcal{U}_i} \times P$  and evaluates the summation  $HK_{\mathcal{U}_i} = h_1(K_{\mathcal{U}_i}) + h_2(K_{\mathcal{U}_i}) + \dots + h_\gamma(K_{\mathcal{U}_i})$  using set  $H$  of hash functions. Afterwards, *GW**N* computes  $A_{\mathcal{U}_i} = IPB_i \oplus K_{\mathcal{U}_i}$ ,  $B_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i} || IPB_i || K_{\mathcal{U}_i})$ ,  $W_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i} || IPB_i) \oplus h(ID_{\mathcal{U}_i} || HK_{\mathcal{U}_i})$ , and transmits the message  $\langle h(), A_{\mathcal{U}_i}, B_{\mathcal{U}_i}, W_{\mathcal{U}_i} \rangle$  to the user  $\mathcal{U}_i$  using a secure communication channel.

**Step R3:** After receiving the message  $\langle h(), A_{\mathcal{U}_i}, B_{\mathcal{U}_i}, W_{\mathcal{U}_i} \rangle$  from  $GW N$ , the user  $\mathcal{U}_i$  stores  $h(), Gen(), Rep(), \mathcal{T}, P, A_{\mathcal{U}_i}, B_{\mathcal{U}_i}, W_{\mathcal{U}_i}, \tau_i$  into the smart card  $SC_{\mathcal{U}_i}$

**Table 7.1:** Summary of User Registration Phase

Step 1:For User ( $\mathcal{U}_i$ )	Step 2:For the gateway node $GW N$
$\mathcal{U}_i$ inputs $ID_{\mathcal{U}_i}, PW_{\mathcal{U}_i}$ and $BIO_{\mathcal{U}_i}$ and computes: $(\sigma_i, \tau_i) = Gen(BIO_{\mathcal{U}_i})$ , {Using fuzzy extractor} then $\mathcal{U}_i$ computes $IPB_i = h(ID_{\mathcal{U}_i}    PW_{\mathcal{U}_i}    \sigma_i)$  $\mathcal{U}_i$ transmits $\langle ID_{\mathcal{U}_i}, IPB_i \rangle$ to $GW N$ <div style="text-align: center;"><math>\xrightarrow{ViaSecureChannel}</math></div>	$GW N$ computes $K_{\mathcal{U}_i} = r_{\mathcal{U}_i} \times P$ , $HK_{\mathcal{U}_i} = h_1(K_{\mathcal{U}_i}) + h_2(K_{\mathcal{U}_i}) \dots h_\gamma(K_{\mathcal{U}_i})$ , $A_{\mathcal{U}_i} = IPB_i \oplus K_{\mathcal{U}_i}$ , $B_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i}    IPB_i    K_{\mathcal{U}_i})$ , $W_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i}    IPB_i) \oplus h(ID_{\mathcal{U}_i}    HK_{\mathcal{U}_i})$ , $GW N$ transmits $\langle h(), A_{\mathcal{U}_i}, B_{\mathcal{U}_i}, W_{\mathcal{U}_i} \rangle$ to $\mathcal{U}_i$ <div style="text-align: center;"><math>\xleftarrow{ViaSecureChannel}</math></div>
Step 3:For User ( $\mathcal{U}_i$ )	
$\mathcal{U}_i$ stores $h(), Gen(), Rep(), \mathcal{T}, P, A_{\mathcal{U}_i}, B_{\mathcal{U}_i}, W_{\mathcal{U}_i}, \tau_i$ into $SC_{\mathcal{U}_i}$	

The Pre-Deployment and Registration Phase are described in following Figure 7.1.

### 7.3.3 Authenticated Key Exchange Phase:

In this phase, we use the reproduction procedure  $Rep()$  of fuzzy extractor for authentication the user  $\mathcal{U}_i$  with its noisy bio-metric data  $BIO'_{\mathcal{U}_i}$  and the vector  $V$  of bloom filter [8] for validating the user  $\mathcal{U}_i$ . We use Elliptic curve Diffie Hellman procedure for exchanging the session key. The procedures of this phase are described in step A1, A2, A3 and summarized in Table 7.2.

**Step A1:** In this step, the user  $\mathcal{U}_i$  inserts the smart card  $SC_{\mathcal{U}_i}$  into the card reader. Subsequently,  $\mathcal{U}_i$  enters his/her unique identity  $ID_{\mathcal{U}_i}$ , the secret password  $PW_{\mathcal{U}_i}$ ,

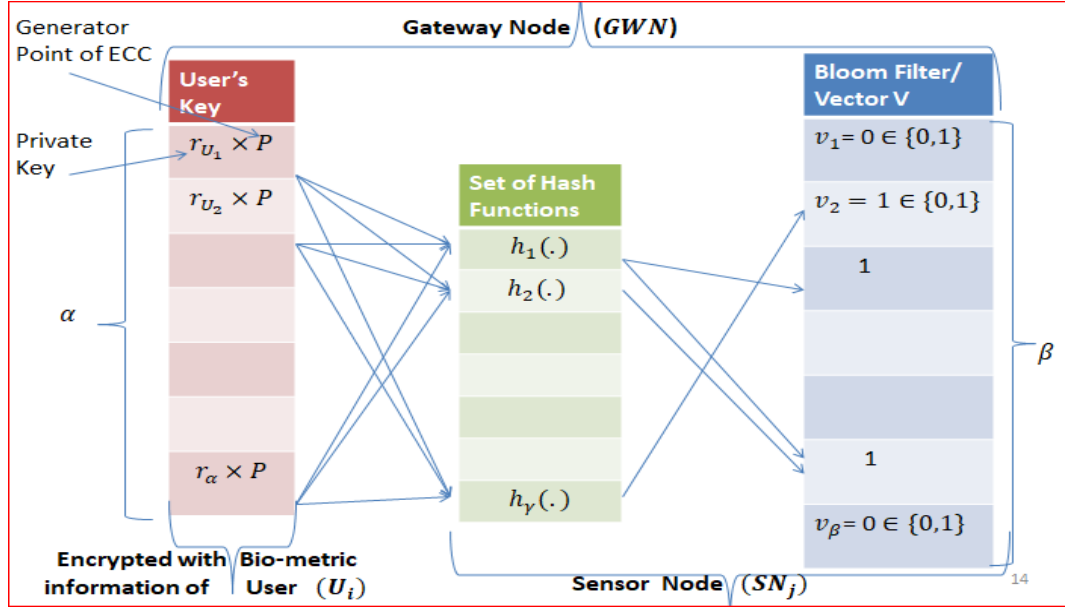


Figure 7.1: Pre-Deployment and Registration Phase

and the noisy bio-metric information  $BIO'_{u_i}$  with an error tolerance limit  $\mathcal{T}$  such that  $dis(BIO_{u_i}, BIO'_{u_i}) \leq \mathcal{T}$ . Then, the  $\mathcal{U}_i$  computes the secret parameter  $\sigma_i = Rep(BIO'_{u_i}, \tau_i)$  using the reproduction function  $Rep()$  of the fuzzy extractor. Afterwards,  $\mathcal{U}_i$  evaluates  $IPB_i = h(ID_{u_i} || PW_{u_i} || \sigma_i)$ ,  $K_{u_i} = IPB_i \oplus A_{u_i}$ , and  $B'_{u_i} = h(ID_{u_i} || IPB_i || K_{u_i})$  using the secure hash function  $h()$ . To ensure the correctness of the secret credential (identity, password, bio-metric information and the secret parameter provided by the  $GWN$ ) of the user  $\mathcal{U}_i$ ,  $SC_{u_i}$  verifies the condition  $B'_{u_i} = B_{u_i}$ . If the condition is not satisfied, it terminates the protocol. Otherwise,  $SC_{u_i}$  Computes  $h(ID_{u_i} || HK_{u_i}) = W_{u_i} \oplus h(ID_{u_i} || IPB_i)$  and finds the current time-stamp  $T_{u_i}$ . Then, it computes  $m_1 = h(ID_{u_i} || ID_{SN_j} || K_{u_i} || h(ID_{u_i} || HK_{u_i}) || T_{u_i})$ . Subsequently,  $\mathcal{U}_i$  constructs the message  $M_1 = \langle ID_{u_i}, ID_{SN_j}, K_{u_i}, m_1, T_{u_i} \rangle$  and transmits  $\langle M_1 \rangle$  to the sensor node  $SN_j$ .

**Step A2:** In this step, after receiving  $M_1$ , the sensor node  $SN_j$  first examines the satisfactory condition of the time-stamp  $T' - T_{u_i} < \Delta T$  in order to withstand the security attacks such as replay, energy exhausting attacks. If this condition is not satisfied, it terminates the protocol and declares the possibilities of security

attacks. Otherwise,  $SN_j$  verifies the condition  $v_i = 1, \forall l \in [1, \gamma]$  such that  $h_l(K_{\mathcal{U}_i}) = i$  to ensure the validity of the user using the concepts of bloom filter. If this condition is not satisfied,  $SN_j$  terminates the protocol invalidating the user  $\mathcal{U}_i$ . Otherwise,  $SN_j$  computes  $h(ID_{\mathcal{U}_i} || HK_{\mathcal{U}_i})' = h(ID_{\mathcal{U}_i} || \sum_{l=0}^{\gamma} h_l(K_{\mathcal{U}_i}))$  and  $m'_1 = h(ID_{\mathcal{U}_i} || ID_{SN_j} || K_{\mathcal{U}_i} || h(ID_{\mathcal{U}_i} || HK_{\mathcal{U}_i})' || T_{\mathcal{U}_i})$ . Then, in order to reduce the false acceptance rate of user validation, the sensor node verifies the condition  $m'_1 = m_1$ . If this condition is not satisfied, the sensor node  $SN_j$  terminates the protocol invalidating the user  $\mathcal{U}_i$ . Otherwise,  $SN_j$  finds the current time-stamp  $T_{SN_i}$  and computes the session  $Key = h(T_{\mathcal{U}_i} || T_{SN_j} || (r_{SN_j} \times K_{\mathcal{U}_i}))$  (Based on *ECDH* procedure). Then  $SN_j$  computes  $m_2 = h(ID_{\mathcal{U}_i} || ID_{SN_j} || K_{SN_j} || h(ID_{\mathcal{U}_i} || HK_{\mathcal{U}_i})' || T_{SN_j})$  and subsequently constructs the message  $M_2 = \langle ID_{\mathcal{U}_i}, ID_{SN_j}, K_{SN_i}, m_2, T_{SN_j} \rangle$ . Finally,  $SN_j$  transmits  $M_2$  to  $\mathcal{U}_i$ .

**Step A3:** In this step, after receiving  $M_2$ , the user  $\mathcal{U}_i$  first examines the satisfactory condition of the time-stamp  $T'' - T_{SN_j} < \Delta T$  in order to resist the security attacks such as replay and energy exhausting attacks. If the condition is not satisfied, the protocol is terminated with the declaration of possible security attacks. Otherwise,  $\mathcal{U}_i$  Computes  $m'_2 = h(ID_{\mathcal{U}_i} || ID_{SN_j} || K_{SN_j} || h(ID_{\mathcal{U}_i} || HK_{\mathcal{U}_i})' || T_{SN_j})$ . Then, to ensure the validity of the sensor node  $SN_j$ ,  $\mathcal{U}_i$  verifies the condition  $m'_2 = m_2$ . If this condition is not satisfied, the protocol is terminated. Otherwise,  $\mathcal{U}_i$  establishes the session  $Key = h(T_{\mathcal{U}_i} || T_{SN_j} || (r_{\mathcal{U}_i} \times K_{SN_j}))$  with  $SN_j$  (Based on *ECDH* procedure).

The sequence diagram of the message transmission for the user registration, authentication and key establishment phase is shown in following Figure 7.2.

## 7.4 Performance and Security Analysis:

For a binary vector  $V$  of size  $\beta$ , the probability that a certain bit is not set to one by any of the  $\gamma$  number of hash functions is:

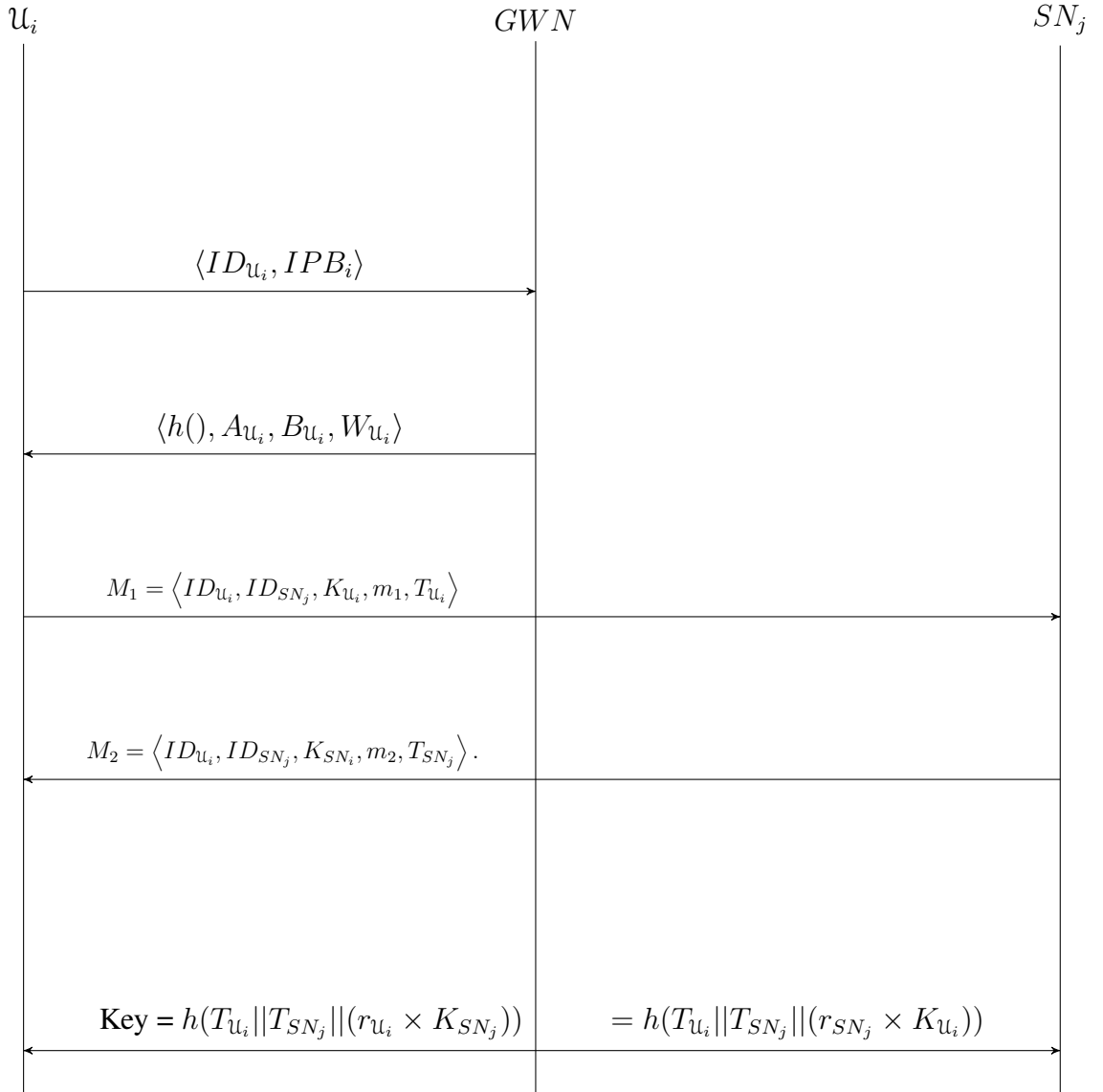
$$\left(1 - \frac{1}{\beta}\right)^{\gamma}$$

## 7.4 Performance and Security Analysis:

**Table 7.2:** Authenticated Key Exchange Phase

Step 4: For User ( $\mathcal{U}_i$ )	Step 5: For the sensor node ( $SN_j$ )
<p><math>\mathcal{U}_i</math> provides <math>ID_{\mathcal{U}_i}, PW_{\mathcal{U}_i}, BIO'_{\mathcal{U}_i}</math> and Computes  <math>\sigma_i = Rep(BIO'_{\mathcal{U}_i}, \tau_i)</math> {Using fuzzy extractor },  <math>IPB_i = h(ID_{\mathcal{U}_i}    PW_{\mathcal{U}_i}    \sigma_i)</math>,  <math>K_{\mathcal{U}_i} = IPB_i \oplus A_{\mathcal{U}_i}, B'_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i}    IPB_i    K_{\mathcal{U}_i})</math>  <b>if</b> <math>B'_{\mathcal{U}_i} = B_{\mathcal{U}_i}</math> <b>then</b>              <math>\mathcal{U}_i</math> Computes              <math>h(ID_{\mathcal{U}_i}    HK_{\mathcal{U}_i}) = W_{\mathcal{U}_i} \oplus h(ID_{\mathcal{U}_i}    IPB_i)</math> and              finds the current time-stamp <math>T_{\mathcal{U}_i}</math>. Then computes <math>m_1 =</math>              <math>h(ID_{\mathcal{U}_i}    ID_{SN_j}    K_{\mathcal{U}_i}    h(ID_{\mathcal{U}_i}    HK_{\mathcal{U}_i})    T_{\mathcal{U}_i})</math>.              Subsequently, <math>\mathcal{U}_i</math> construct the message              <math>M_1 = \langle ID_{\mathcal{U}_i}, ID_{SN_j}, K_{\mathcal{U}_i}, m_1, T_{\mathcal{U}_i} \rangle</math>              <math>\mathcal{U}_i</math> transmits <math>\langle M_1 \rangle</math> to <math>SN_j</math>              ViaPublicChannel  <b>end</b>  <b>else</b>              <math>\mathcal{U}_i</math> is inauthentic. Stolen smart card attack possible.              Abort the protocol.  <b>end</b></p>	<p><b>if</b> <math>T' - T_{\mathcal{U}_i} &lt; \Delta T</math>, <b>then</b>              <b>if</b> <math>v_i = 1, \forall l \in [1, \gamma]</math> such that <math>h_l(K_{\mathcal{U}_i}) = i</math> <b>then</b>                  Computes <math>h(ID_{\mathcal{U}_i}    HK_{\mathcal{U}_i})' =</math>                  <math>h(ID_{\mathcal{U}_i}    \sum_{l=0}^{\gamma} h_l(K_{\mathcal{U}_i}))</math>,                  <math>m'_1 =</math>                  <math>h(ID_{\mathcal{U}_i}    ID_{SN_j}    K_{\mathcal{U}_i}    h(ID_{\mathcal{U}_i}    HK_{\mathcal{U}_i})'</math>                  <math>   T_{\mathcal{U}_i})</math>              <b>end</b>              <b>else</b>                  <math>\mathcal{U}_i</math> is not validated at <math>SN_j</math>. Rejects <math>\mathcal{U}_i</math>.              <b>end</b>              <b>if</b> <math>m'_1 = m_1</math> <b>then</b>                  Finds the current time-stamp <math>T_{SN_j}</math>,                  <math>SN_j</math> computes the session                  <math>Key = h(T_{\mathcal{U}_i}    T_{SN_j}    (r_{SN_j} \times K_{\mathcal{U}_i}))</math> (Based on                  ECDH problem).                  Then <math>SN_j</math> computes <math>m_2 =</math>                  <math>h(ID_{\mathcal{U}_i}    ID_{SN_j}    K_{SN_j}    h(ID_{\mathcal{U}_i}    HK_{\mathcal{U}_i})'</math>                  <math>   T_{SN_j})</math>.                  Subsequently <math>SN_j</math> constructs the message                  <math>M_2 = \langle ID_{\mathcal{U}_i}, ID_{SN_j}, K_{SN_j}, m_2, T_{SN_j} \rangle</math>              <b>end</b>              <b>else</b>                  <math>\mathcal{U}_i</math> is not validated at <math>SN_j</math>. Rejects <math>\mathcal{U}_i</math>.              <b>end</b>              <math>SN_j</math> transmits <math>M_2</math> to <math>\mathcal{U}_i</math>              ViaPublicChannel  <b>end</b>  <b>else</b>              Replay, bogus message flooding and energy exhausting              attack possible. Rejects <math>\mathcal{U}_i</math> and aborts the protocol.  <b>end</b></p>
Step 6: For User ( $\mathcal{U}_i$ )	
<p><b>if</b> <math>T'' - T_{SN_j} &lt; \Delta T</math> <b>then</b>              <math>\mathcal{U}_i</math> Computes              <math>m'_2 = h(ID_{\mathcal{U}_i}    ID_{SN_j}    K_{SN_j}    h(ID_{\mathcal{U}_i}    HK_{\mathcal{U}_i})    T_{SN_j})</math>              <b>if</b> <math>m'_2 = m_2</math> <b>then</b>                  Establish the session                  <math>Key = h(T_{\mathcal{U}_i}    T_{SN_j} (r_{\mathcal{U}_i} \times K_{SN_j}))</math> with <math>SN_j</math>                  (Based on ECDH problem)              <b>end</b>              <b>else</b>                  <math>SN_j</math> is not validated by <math>\mathcal{U}_i</math>. Aborts the protocol.              <b>end</b>  <b>end</b>  <b>else</b>              Replay and energy exhausting attack possible. Reject <math>\mathcal{U}_i</math>  <b>end</b></p>	

**Figure 7.2:** Sequence Diagram 5 for Registration, Authentication and Key Establishment



## 7.4 Performance and Security Analysis:

And after representing  $\alpha$  numbers of users, the probability of the given bit is not set to zero is:

$$\left(1 - \frac{1}{\beta}\right)^{\gamma\alpha}$$

Therefore, the probability of the bit set to one is:

$$1 - \left(1 - \frac{1}{\beta}\right)^{\gamma\alpha}$$

If all the  $\gamma$  positions of the vector  $V$  computed by each hash function of the set  $H$  sets to one, the user belongs to the set  $V$ . Hence, the probability of a user not belonging to the set  $V$  is given by:

$$f_{positive} = \left(1 - \left(1 - \frac{1}{\beta}\right)^{\gamma\alpha}\right)^{\alpha} \equiv \left(1 - e^{-\frac{\gamma\alpha}{\beta}}\right)^{\gamma} \quad (1)$$

Therefore, we have

$$f_{positive} = e^{\gamma \ln\left(1 - e^{-\frac{\gamma\alpha}{\beta}}\right)} \quad (2)$$

Equation (2) indicates that the probability of false positive  $f_{positive}$  decreases as the size of vector  $V$  increases and increases as the number of users  $\alpha$  increases.

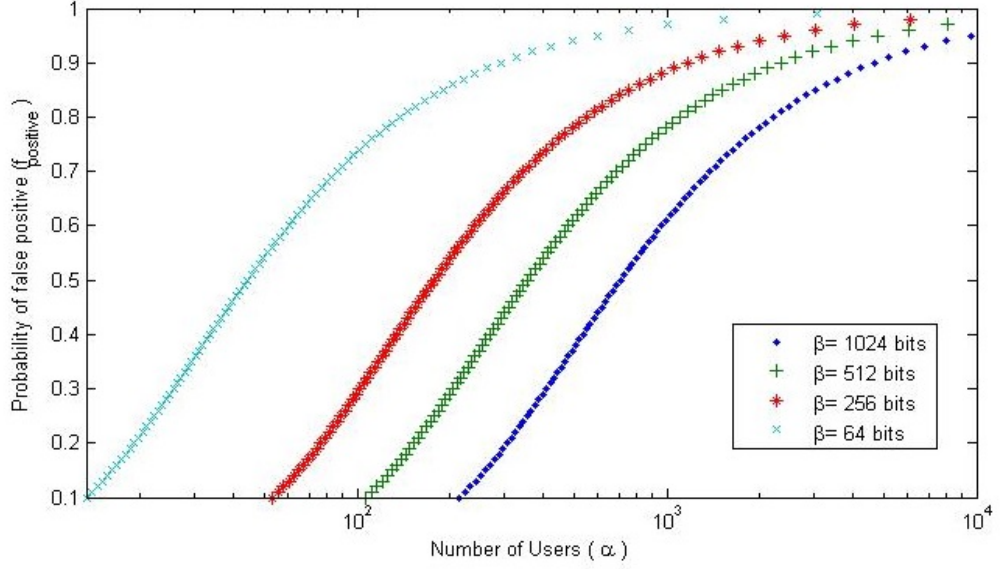
To minimize the value of  $f_{positive}$ , take the derivatives of  $\gamma \ln\left(1 - e^{-\frac{\gamma\alpha}{\beta}}\right)$  with respect to  $\gamma$  and find out:

$$\frac{d\left(\gamma \ln\left(1 - e^{-\frac{\gamma\alpha}{\beta}}\right)\right)}{d\gamma} = \ln\left(1 - e^{-\frac{\gamma\alpha}{\beta}}\right) + \frac{\gamma\alpha}{\beta} \frac{e^{-\frac{\gamma\alpha}{\beta}}}{1 - e^{-\frac{\gamma\alpha}{\beta}}} \quad (3)$$

To get the optimal value of  $\gamma$ , make the derivatives (Equation (3)) equal to zero and finds out  $\gamma_{optimal} = \frac{\beta}{\alpha} \ln 2$ . [8]

The following graph 7.3 represents the probability of false positive  $f_{positive}$  with respect to the number of users  $\alpha$  and the size of the bloom filter  $\beta$ .





**Figure 7.3:**  $f_{positive}$  with respect to  $\alpha$  and  $\beta$  [8]

### 7.4.1 Security Analysis

The formal and informal security analysis of our proposed protocol specifies the features involved to deal with possible attacks.

#### 7.4.1.1 Formal security analysis using Random Oracle Model

The random oracle model (ROM) is a robust tool proposed by Bellare and Rogaway [21] in 1993 to make it possible to execute meticulous proofs of security for particular fundamental cryptographic protocols.

A random oracle is a theoretical black box that responds to every individual query with an accurate random response chosen uniformly from its output domain. If a query is occurring several times, it responds the same way every time that query is performed.

Based on random oracle model, the following Theorem 2 shows that our protocol can resist various security attacks.

**Theorem 2.** *If  $h()$  follows the random oracle Reveal1, our protocol resists the adversary  $\mathcal{A}$  for deriving the value of secret parameters  $PW_{u_i}, \sigma_i, K_{u_i}, h(ID_{u_i} || HK_{u_i})$*

## 7.4 Performance and Security Analysis:

*Proof.* If we assume that, there exists a Reveal oracle which can derive string  $s$  from the hash digest  $d = h(s)$ . Then, the adversary  $\mathcal{A}$  can design a procedure  $EXP_{B,\mathcal{A}}^H$  as shown in Algorithm 7.1 such that probability of success of  $EXP_{B,\mathcal{A}}^H$  is  $Succ_{B,\mathcal{A}}^H = |Pr[EXP_{B,\mathcal{A}}^H = 1] - 1|$ .

---

### Algorithm 7.1 $EXP_{B,\mathcal{A}}^H$

---

1. Extract  $\{A_{\mathcal{U}_i}, B_{\mathcal{U}_i}, W_{\mathcal{U}_i}, \tau_i, \mathcal{T}, h(), Gen(), Rep()\}$  from  $SC_{\mathcal{U}_i}$  using power analysis attacks [14]. Where
  2.  $A_{\mathcal{U}_i} = IPB_i \oplus K_{\mathcal{U}_i}$ ,
  3.  $B_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i} || IPB_i || K_{\mathcal{U}_i})$ ,
  4.  $W_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i} || IPB_i) \oplus h(ID_{\mathcal{U}_i} || HK_{\mathcal{U}_i})$ ,
  5. Use Reveal oracle on  $B_{\mathcal{U}_i}$  to get  $ID'_{\mathcal{U}_i}, IPB'_i, K'_{\mathcal{U}_i}$
  6. Use Reveal oracle on  $IPB'_i$  to get  $PW'_i, \sigma'_i$
  7. Eavesdrop the message  $M_1$
  8. **if** ( $ID'_{\mathcal{U}_i} = ID_{\mathcal{U}_i}$ ) **then**
    - Use  $\sigma'_i$  to find  $K'_{\mathcal{U}_i} = A_{\mathcal{U}_i} \oplus h(ID'_{\mathcal{U}_i} || PW'_{\mathcal{U}_i} || \sigma'_i)$  and compute  $h(HK_{\mathcal{U}_i})' = W_{\mathcal{U}_i} \oplus h(ID'_{\mathcal{U}_i} || h(ID'_{\mathcal{U}_i} || PW'_{\mathcal{U}_i} || \sigma'_i))$
    - if** ( $K_{\mathcal{U}_i} = IPB'_i \oplus A_{\mathcal{U}_i}$ ) **and**  $h(HK_{\mathcal{U}_i}) = W_{\mathcal{U}_i} \oplus h(ID'_{\mathcal{U}_i} || IPB'_i)$  **then**
      - Accept  $K'_{\mathcal{U}_i}$  and  $h(ID_{\mathcal{U}_i} || HK_{\mathcal{U}_i})'$  as the correct secret parameters of the user  $\mathcal{U}_i$
      - Return 1
  - end**
  - else**
    - Return 0
  - end**
  - end**
  - else**
    - Return 1
  - end**
- 

According to Algorithm 7.1, there exists a Reveal oracle capable of finding the preimage of  $h()$ . Therefore, the adversary  $\mathcal{A}$  can get the values of  $PW_{\mathcal{U}_i}, \sigma_i, K_{\mathcal{U}_i}, h(ID_{\mathcal{U}_i} || HK_{\mathcal{U}_i})$ . However, according to Definition 1.3.1 we get

$$Adv_{\mathcal{A}}^h(t_1) = Pr[(s, s') \leftarrow_R \mathcal{A} : s \neq s', h(s) = h(s')]$$

and  $Adv_{\mathcal{A}}^h(t_1) \leq \tau$ , for any sufficiently small  $\tau > 0$ . If we consider  $Adv_{\mathcal{A}}^h(t_1)$  as the

advantage function of the procedure  $EXP_{B,\mathcal{A}}^h$ , our protocol is secure for  $Adv_{\mathcal{A}}^h(t_1) \leq \tau$ . Now, we find that  $h()$  contradicts the Reveal oracle considered in Algorithm 7.1. This indicates that our protocol resists the adversary  $\mathcal{A}$  for deriving the values of secret parameters  $PW_{\mathcal{U}_i}, \sigma_i, K_{\mathcal{U}_i}, h(ID_{\mathcal{U}_i}||HK_{\mathcal{U}_i})$ . Hence, the theorem is proved.  $\square$

#### 7.4.1.2 Informal security analysis

In this section, we show that our proposed protocol resists various security attacks based on following proposition:

**Proposition 1.** The proposed protocol is secure against stolen smart card attack.

*Proof.* An adversary  $\mathcal{A}$  can extract the value of  $A_{\mathcal{U}_i} = IPB_i \oplus K_{\mathcal{U}_i}$ ,  $B_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i}||IPB_i||K_{\mathcal{U}_i})$ ,  $W_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i}||IPB_i) \oplus h(ID_{\mathcal{U}_i}||HK_{\mathcal{U}_i})$  from stolen  $SC_{\mathcal{U}_i}$  [14]. But  $\mathcal{A}$  can not find out the value of secret parameters  $K_{\mathcal{U}_i}, h(ID_{\mathcal{U}_i}||HK_{\mathcal{U}_i})$  without having  $BIO_{\mathcal{U}_i}, PW_{\mathcal{U}_i}$ . Because the secret credentials of the user  $\mathcal{U}_i$  are protected with user's inherence ( $BIO_{\mathcal{U}_i}$ ) possession ( $SC_{\mathcal{U}_i}$ ), knowledge ( $PW_{\mathcal{U}_i}$ ).  $\square$

**Proposition 2.** The proposed protocol is secure against replay attack.

*Proof.* The verification of time-stamps  $T_{\mathcal{U}_i}, T_{SN_j}$ , the hashed value of secret credential and the messages i.e.,  $m_1, m_2$  ensures that our protocol resists replay attack.  $\square$

**Proposition 3.** The proposed protocol is secure against man-in-the middle attack.

*Proof.* The integrity feature provided by the message  $m_1, m_2$  ensures non-feasibility of man-in-the middle attack. An adversary  $\mathcal{A}$  can not alter the message  $m_1, m_2$ , because of preimage resistant feature of the hash function used.  $\square$

**Proposition 4.** The proposed protocol is secure against energy exhausting attack.

*Proof.* To avoid bogus message flooding (which exhausts the resources of WSNs), we eliminate the illegitimate users at the initial level (i.e., at sensor node itself) of message transmission. For a sensor node  $SN_j$ , the energy needed for computation is less compared to data transmission (the energy required in 2090 clock cycles of computation is equivalent to the energy required for transmitting 1-bit data). We verify

the correctness of  $v_i = 1, \forall l \in [1, \gamma]$  such that  $h_l(K_{u_i}) = i$  at sensor node  $SN_j$  and the correct value of  $m_1$  ensures that the user belongs to the authorized group. Energy required in transmitting and receiving 1-bit data (at the data rate of 12.4 Kb/s) are  $59.2 \mu \text{ Joule}$ ,  $28.6 \mu \text{ Joule}$  respectively [119]. Furthermore, we assume that  $N, n$  are the size and density (total number of nodes within the circular area with radius equal to the communication range of sensor node) of WSNs. If an illegitimate user  $\mathcal{A}$  is not eliminated or filtered at initial level,  $\mathcal{A}$  can consume total energy equal to  $E$  by sending a message  $M'_1 = \langle ID_{u_i}, ID_{SN_j}, K_{u_i}, m_1, T_{u_i} \rangle$  of size  $S$  bytes. Where  $E$  can be evaluated as follows :

$$E = N \left( S \times (59.2 + 28.6n) \right) \mu \text{ Joule}.$$

But in our protocol, we eliminate  $\mathcal{A}$  at the initial level which saves the total energy of  $(E - E_h) \mu \text{ Joule}$ . Where  $E_h$  is the energy required in computing and verifying the value of  $v_i$  and  $m_1$ . The hash function SHA-1 requires the energy of  $5.9 \mu \text{ Joule/byte}$  to verify the value of  $v_i$ . Hence, our proposed protocol withstands the energy exhausting attacks.  $\square$

**Proposition 5.** The proposed protocol avoids impersonation attack and forgery attack.

*Proof.* Verifying  $h(ID_{u_i} || HK_{u_i})' = h(ID_{u_i} || \sum_{l=0}^{\gamma} h_l(K_{u_i}))$ ,  $\forall l \in [1, \gamma]$  ensures the integrity and authenticity which in turn avoids impersonation attacks, forgery attack, etc.  $\square$

## 7.5 Performance Comparison

Table 7.3 shows the comparison based on security features, and it indicates that our protocol is relatively secure in comparison to other existing protocols. Table 7.4 represents the computational cost comparison, it shows that our protocol provides efficient computational cost for all the three entities i.e.,  $\mathcal{U}_i, GWN$  and  $SN_j$ .

To evaluate the computational performance, we use SHA-1 hash function and ECDH key exchange mechanism on MicaZ sensor node (processor 8-bit ATmega128L Atmel, ROM 4K bytes, ROM 128K bytes, 512K bytes EEPROM and 2AA battery) with TinyOS [111] operating system and nesC [112] programming language. We verify the security features of our proposed protocol using widely accepted AVISPA

## 7.5 Performance Comparison

**Table 7.3:** Comparisons of Security Features

Security Feature	Xue et al. [79]	Jiang et al. [80]	Althobaiti et al. [59]	A.K.Das [81]	Choi et al. [61]	Proposed Protocol
$SF_1$	No	Yes	Yes	No	No	Yes
$SF_2$	No	No	No	Yes	Yes	Yes
$SF_3$	No	No	No	Yes	Yes	Yes
$SF_4$	No	No	Yes	Yes	Yes	Yes
$SF_5$	No	No	No	Yes	No	Yes
$SF_6$	No	No	Yes	Yes	Yes	Yes
$SF_7$	Yes	Yes	Yes	Yes	Yes	Yes
$SF_8$	Yes	Yes	No	Yes	Yes	Yes

Note:  $SF_1, SF_2, SF_3, SF_4, SF_5$  are the security features.  $SF_1$  resist the attack based on stolen smart card,  $SF_2$  indicates the secure password updating,  $SF_3$  represents secure bio-metric information updating,  $SF_4$  indicates non-repudiation,  $SF_5$  offers formal security analysis,  $SF_6$  represents no privileged-insider attack,  $SF_7$  resist password guessing attack,  $SF_8$  provides key agreement between  $U_i$  and  $SN_j$ .

**Table 7.4:** Computational Cost Comparison

Protocol	Computational Overhead on $U_i, SN_j, GWN$		
	$U_i$	$SN_j$	$GWN$
Xue et al.'s [79]	$12 T_H$	$6 TS_H$	$17 T_H$
Jiang et al.'s [80]	$8 T_H$	$5 TS_H$	$11 T_H$
Althobaiti et al.'s [59]	$2 T_{BFE} + 2 T_{ENC} / T_{DEC} + 6 T_H$	$TS_{DEC} + TS_{MAC} + TS_H$	$T_{ENC} + T_{MAC} + 4 T_H$
A.K.Das's [81]	$2 T_{FE} + T_{ENC} + 10 T_H$	$TS_{DEC} + 2 TS_H$	$2 T_{ENC} / T_{DEC} + 5 T_H$
Choi et al.'s [61]	$3 T_{FE} + 2 T_M + T_{ENC} + 12 T_H$	$2 TS_{DEC} + TS_M + 6 TS_H$	$2 T_{ENC} / T_{DEC} + 16 T_H$
Proposed Protocol	$8 T_H + T_M$	$(\gamma + 3) TS_H + TS_M$	$T_M$

Note:  $T_H, T_{FE}, T_{ENC}, T_{DEC}, T_{BFE}, T_{MAC}, T_M$  indicates the time required to perform secure hashing, Gen./ Rep(.), encryption, decryption, bio-metric feature extraction, message authentication code, point multiplication on elliptic curve operations, respectively for User and Gateway Node.  $TS_H, TS_{DEC}, TS_{MAC}, TS_M$  indicates the time required to perform secure hashing, decryption, message authentication code, point multiplication on elliptic curve operations, respectively for Sensor Node.

Table 7.5: Security Verification Result of AVISPA tool

SUMMARY
SAFE
DETAILS
BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
/home/cmb-lab-22/Desktop/UserAuthentication.if
GOAL
as_specified
BACKEND
OFMC
COMMENTS
STATISTICS
parseTime: 0.00s
searchTime: 0.27s
visitedNodes: 106 nodes
depth: 8 piles
<div><div>Tools</div><div>HLPSTL</div><div>HLPSTL2IF</div><div>IF</div><div>OFMC</div><div>ATSE</div><div>SATMC</div><div>TA4SP</div><div><div>Save file</div><div>View CAS+</div></div><div><div>Choose Tool option and press execute</div><div>Execute</div></div></div>

(Automated Validation of Internet Security Protocols and Applications) tool. The verification result of AVISPA [7] tool is shown in Fig. 7.5 which represents that our protocol is safe from various attacks (like man-in-the middle attack, replay attack etc.) using Dolev-Yao model [110] with the bounded number of sessions, specified goal and On-the-Fly Model-Checker(OFMC) backend.

## 7.6 Summary

In this chapter, we have presented the security issues involved in sensor nodes of WSNs & IoT and some basic concepts of a user authentication, session key establishment protocol, fuzzy extractor and bloom filter. We have proposed a Bloom filter based secure and efficient authenticated key exchange protocol. We have performed the security analysis and verification using a widely accepted and robust tool AVISPA. To ensure the correctness of the security features involved in the protocol, we have performed the formal security verification using random oracle model. Finally, we have presented the comparative analysis of our proposed protocol with other existing protocols based on security features and computational overhead which indicates that our protocol is secure and efficient.

## **Chapter 8**

# **Symmetric Hash Function based User Authentication and Key Establishment Protocol for WSNs & IoT**

“Cryptography is the essential building block of independence for organizations on the Internet, just like armies are the essential building blocks of states, because otherwise one state just takes over another.” Julian Assange

This chapter presents an efficient user authentication and key establishment protocol for WSNs & IoT based on Symmetric Hash Function. It concisely explains the problem, security challenges and contributions made thereof. Consequently, it elaborates the proposed protocol, and then, the relative security and computational overhead are analysed.

### **8.1 Introduction and Problem Definition**

WSNs & IoT have gained popularity in recent years in real-time applications ranging from critical applications like battlefield surveillance and border monitoring, health care applications like remote medical diagnosis, smart homes to add intelligence to the home equipment for better comfort and security, etc. As the sensor nodes operate in unattended environment and communicate using wireless medium, it is essential to



protect the sensed information and ensure secure communication to transmit the information to the gateway nodes for further processing. An adversary in a wireless medium not only has the capability to eavesdrop but also can intercept and modify the legitimate data traffic. Hence, security becomes primary concern in WSNs & IoT and many of the security protocols do not simply work given the resource constrained nature of sensor nodes in terms of computing capabilities and storage resources. Therefore, in this chapter, we propose a protocol for authenticating the users without pre-alignment between the test and the registered minutia points of the fingerprint which is suitable for multi-hop WSNs & IoT resource constrained nature of sensor nodes in terms of computing capabilities and storage resources.

In this chapter, we propose symmetric hash function and bloom filter based secure authenticated key exchange (AKE) protocol for WSNs. The proposed protocol is appropriate for authentication of the users without pre-alignment between the test and the registered minutia points of the fingerprint for multi-hop WSNs & IoT. The proposed protocol eliminates unauthorized query information circulation at the initial level (i.e., at the sensor node itself) to prevent bogus information flooding from the sensor nodes to the gateway node. We show that the proposed protocol resists the resource exhaustion attacks associated with WSNs & IoT of large hop count (i.e., a large number of intermediary sensor nodes through which information must pass between source sensor node and the trusted gateway node). We present both the formal and informal security analysis of the proposed protocol using AVISPA tool and basic cryptography concepts. The analysis of computational overhead demonstrates that our proposed protocol is preferable for resource-constrained sensor motes like MicaZ. The security analysis and performance evaluation reveal that the proposed protocol is more secure, effective and resilient in comparison to other existing protocols.

**Problem Definition:** The problem definition of this chapter is as follows: Design and analysis of secure and efficient multi-factor user authentication and session key establishment protocol for resource constrained WSNs & IoT is appropriate for authenticating the users without pre-alignment between the test and the registered minutia points of the fingerprint for multi-hop WSNs & IoT, provides the major security features (such as mutual authentication, secure session key establishment, confidentiality, integrity, freshness etc.) and prevents the major security attacks (such as stolen smart

card, user impersonation, sensor node impersonation, etc.) with minor computational overhead.

## 8.2 Our Contributions

In this chapter, we introduce a novel user authentication protocol based on symmetric hash function to implement user access to the real-time data by authorizing the user immediately at the sensor node level and also making it feasible for users to communicate with the nodes to have replied to their queries. Our protocol has the following attractive features:

- The proposed protocol is appropriate for authenticating the users without pre-alignment between the test and the registered minutia points of the fingerprint for multi-hop WSNs & IoT.
- It is secure against different attacks. The resilience of our proposed protocol against node compromise attack is much higher than other existing protocols.
- It requires less communication, computation, and storage overheads as compared to other existing protocols.
- It does not require any involvement of the gateway node during the authentication and key establishment phase as well as dynamic sensor node addition phase.
- Higher security along with lower communication, computation and storage overheads make our protocol much suitable for practical applications of WSNs & IoT.
- It eliminates unauthorized query information circulation at the initial level (i.e., at the sensor node itself) to prevent bogus information flooding from the sensor nodes to the gateway node.
- It resists the resource exhaustion attacks associated with WSNs & IoT of large hop count (i.e., a large number of intermediary sensor nodes through which information must pass between source sensor node and the trusted gateway node).
- It provides better security as compared with other related protocols, since it supports mutual authentication between the user and the sensor nodes, resists denial-

of-service attack, privileged-insider attack, stolen smart card attack and node capture attack.

- It supports dynamic node addition after initial deployment of sensor nodes in the WSNs & IoT. The proposed protocol does not require to update information for new nodes addition in the user's  $\mathcal{U}_i$  smart card  $SC_{\mathcal{U}_i}$ .
- It supports updating the user's credentials locally without the help of the gateway node.
- It provides unconditional security against node capture attacks. That is, compromise of a sensor node does not reveal any secret information of other sensor nodes and it does not lead to compromise of any other secure communication between the user and the non-compromised nodes in the network.
- It establishes a secret session key between the user  $\mathcal{U}_i$  and a sensor node  $SN_j$  for future encrypted communication of the real-time information inside WSNs & IoT between them using the established session key.
- The formal and informal security analysis of the proposed protocol using AVISPA tool and basic cryptography concepts ensures that our protocol is safe from security attacks and it has major security features for user authentication for the applications of WSNs & IoT.
- The analysis of computational overhead demonstrates that our proposed protocol is preferable for resource-constrained sensor motes like MicaZ.
- The security analysis and performance evaluation reveal that the proposed protocol is more secure, effective and resilient in comparison to other existing protocols.

## 8.3 Proposed Protocol

The proposed protocol of this chapter is convenient for the multi-hop WSNs & IoT possessing large hop count which requests high communication overhead to transmit a message from the source sensor node  $SN_j$  to the gateway node  $GW_N$ . For WSNs & IoT the communication cost is constantly higher than the computational cost. Therefore, we have proposed a protocol which approves or authenticate the users without

pre-alignment between the test and the enrolled minutia points of the fingerprint at sensor node itself. The proposed protocol disposes of (utilizing bloom filter) unapproved information flooding from the sensor nodes to the gateway node. Our proposed protocol comprises of following three vital phases:

### 8.3.1 Pre-deployment Phase:

- In this phase, the gateway node  $GWN$  takes a generator or base point  $P$  of order  $q$  over  $E(F_p)$  and produces a set of private, public key elements as  $\langle (R_{U_1}, R_{U_1} \times P), (R_{U_2}, R_{U_2} \times P), \dots (R_{U_\alpha}, R_{U_\alpha} \times P) \rangle$ . Where  $R_{U_i} \in \mathbb{Z}_q^*, \forall i \in [1, \alpha]$ .
- The  $GWN$  produces a vector  $B = b_0 b_1 \dots b_{V-1}$  using a set of hash function  $H = \{h_1, h_2 \dots h_\beta\}$ . Where the size of vector  $B$  is equal to  $V$ , the cardinality of the set  $H$  is equal to  $\beta$  and
$$b_i = \begin{cases} 1, & \forall l \in [1, \beta], t \in [1, \alpha], h_l(R_{U_t} \times P) = i \\ 0, & \text{otherwise} \end{cases}$$
- Then, the gateway node  $GWN$  stores the vector  $B$ , the set  $H$  and the key pair  $(R_{SN_j}, K_{SN_j} = R_{SN_j} \times P)$  into the memory of sensor node  $SN_j$ .

### 8.3.2 User registration based on Symmetric Hash Function:

In this phase, the user  $\mathcal{U}_i$  extracts the  $n$  minutia points  $\{c_{1j}, c_{2j}, \dots, c_{nj}\}$  from his/her  $j^{th}$  localized set of minutia points of fingerprint  $BIO_{\mathcal{U}_i}$  (bio-metric information) and evaluates the  $m$  symmetric hash function [16] as follows:

$$\begin{aligned} h_{1j}(c_{1j}, c_{2j}, \dots, c_{nj}) &= c_{1j} + c_{2j} + \dots + c_{nj} \\ h_{2j}(c_{1j}, c_{2j}, \dots, c_{nj}) &= c_{1j}^2 + c_{2j}^2 + \dots + c_{nj}^2 \\ &\dots\dots\dots \\ h_{mj}(c_{1j}, c_{2j}, \dots, c_{nj}) &= c_{1j}^m + c_{2j}^m + \dots + c_{nj}^m \end{aligned}$$

If  $m < n$ , it is difficult for an adversary  $\mathcal{A}$  to find out the minutia point  $c_{ij}$  from given hash values. Therefore, the user  $\mathcal{U}_i$  chooses  $m < n$  and finds out the aggregate hash function as follows:

$$\begin{aligned} h_{SUMj}(c_{1j}, c_{2j}, \dots, c_{nj}) &= \\ &= h_{1j}(c_{1j}, c_{2j}, \dots, c_{nj}) + h_{2j}(c_{1j}, c_{2j}, \dots, c_{nj}) + \dots + h_{mj}(c_{1j}, c_{2j}, \dots, c_{nj}) \\ &= (c_{1j} + c_{1j}^2 + \dots + c_{1j}^m) + (c_{2j} + c_{2j}^2 + \dots + c_{2j}^m) + \dots + (c_{nj} + c_{nj}^2 + \dots + c_{nj}^m) \\ &= \frac{c_{1j}(1 - c_{1j}^m)}{1 - c_{1j}} + \frac{c_{2j}(1 - c_{2j}^m)}{1 - c_{2j}} + \dots + \frac{c_{nj}(1 - c_{nj}^m)}{1 - c_{nj}} \end{aligned}$$

Therefore,

$$h_{SUMj}(c_{1j}, c_{2j}, \dots, c_{nj}) = \sum_{i=1}^n \frac{c_{ij}(1 - c_{ij}^m)}{1 - c_{ij}}$$

If the total number of localized minutia sets is  $k$ , the user  $\mathcal{U}_i$  finds out the set of aggregate hash function as follows:

$$H_{SUM} = \{h_{SUM1}(c_{11}, c_{21}, \dots, c_{n1}), h_{SUM2}(c_{12}, c_{22}, \dots, c_{n2}), \dots, h_{SUMk}(c_{1k}, c_{2k}, \dots, c_{nk})\}$$

The user  $\mathcal{U}_i$  selects its identity  $ID_{\mathcal{U}_i}$ , password  $PW_{\mathcal{U}_i}$ , bio-metric information  $BIO_{\mathcal{U}_i}$  and follows the procedure of step 1,2 and 3 as summarized in Table 8.1.

### 8.3.3 Authenticated Key Exchange Phase:

In this phase, we use the symmetric hash function for authenticating the user  $\mathcal{U}_i$  with its noisy bio-metric data  $BIO'_{\mathcal{U}_i}$ . We use the vector  $B$  of bloom filter [8] for validating the user  $\mathcal{U}_i$ . and Elliptic curve Diffie Hellman procedure for exchanging the session key. The following steps A1, A2 and A3 describes the Authenticated Key Exchange Phase. Table 8.2 (Step 4,5 and 6) summaries our proposed authenticated key exchange phase in details.

**Step A1:**  $\mathcal{U}_i$  insets the smart card  $SC_{\mathcal{U}_i}$  into card reader and provides  $ID_{\mathcal{U}_i}, PW_{\mathcal{U}_i}, BIO'_{\mathcal{U}_i}$ . Then, computes  $IP_i = h(ID_{\mathcal{U}_i} || PW_{\mathcal{U}_i} || RN_{\mathcal{U}_i})$   
 $K_{\mathcal{U}_i} = IP_i \oplus A_{\mathcal{U}_i}$ ,  $D_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i} || IP_i || K_{\mathcal{U}_i})$ ,  $H_{SUM} = GH_{SUM} \oplus K_{\mathcal{U}_i} =$   
 $h_{SUM1}(c_{11}, c_{21}, \dots, c_{n1}) \oplus K_{\mathcal{U}_i} \oplus K_{\mathcal{U}_i}$ ,  $h_{SUM2}(c_{12}, c_{22}, \dots, c_{n2}) \oplus K_{\mathcal{U}_i} \oplus K_{\mathcal{U}_i}, \dots,$   
 $h_{SUMk}(c_{1k}, c_{2k}, \dots, c_{nk}) \oplus K_{\mathcal{U}_i} \oplus K_{\mathcal{U}_i}$ ,  $W_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i} || IP_i) \oplus h(ID_{\mathcal{U}_i} || H_{\mathcal{U}_i})$ . Af-  
 terwards,  $\mathcal{U}_i$  Extract the minutia points  $c'_{ij}$  from the fingerprint  $BIO'_{\mathcal{U}_i}$  such that  
 $c'_{ij} = rc_{ij} + t$ , where the parameter  $r$  and  $t$  represents the accidental rotational

## 8.3 Proposed Protocol

**Table 8.1:** User Registration Phase

Step 1:for User ( $\mathcal{U}_i$ )	Step 2:for the gateway node $GW_N$
<p><math>\mathcal{U}_i</math> inputs <math>ID_{\mathcal{U}_i}, PW_{\mathcal{U}_i}, BIO_{\mathcal{U}_i}</math> and generates a 1024 bit random number <math>RN_{\mathcal{U}_i}</math> and computes <math>IP_i = h(ID_{\mathcal{U}_i}    PW_{\mathcal{U}_i}    RN_{\mathcal{U}_i}), H_{SUM}</math>.</p> <p><math>\mathcal{U}_i</math> transmits <math>\langle ID_{\mathcal{U}_i}, IP_i, H_{SUM} \rangle</math> to <math>GW_N</math></p> <p style="text-align: center;"><math>\xrightarrow{\text{Via Secure Communication Channel}}</math></p>	<ul style="list-style-type: none"> <li><math>GW_N</math> computes <math>K_{\mathcal{U}_i} = R_{\mathcal{U}_i} \times P</math>, <ol style="list-style-type: none"> <li><math>K_{\mathcal{U}_i} = R_{\mathcal{U}_i} \times P</math></li> <li><math>H_{\mathcal{U}_i} = h_1(K_{\mathcal{U}_i}) + h_2(K_{\mathcal{U}_i}) \dots h_\beta(K_{\mathcal{U}_i})</math></li> <li><math>A_{\mathcal{U}_i} = IP_i \oplus K_{\mathcal{U}_i}</math></li> <li><math>GH_{SUM} = H_{SUM} \oplus K_{\mathcal{U}_i}</math>  <math>= h_{SUM1}(c_{11}, c_{21}, \dots, c_{n1})</math>  <math>\oplus K_{\mathcal{U}_i}, h_{SUM2}(c_{12}, c_{22}, \dots, c_{n2}) \oplus</math>  <math>K_{\mathcal{U}_i}, \dots, h_{SUMk}(c_{1k}, c_{2k}, \dots, c_{nk}) \oplus K_{\mathcal{U}_i}</math></li> <li><math>D_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i}    IP_i    K_{\mathcal{U}_i})</math>,</li> <li><math>W_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i}    IP_i) \oplus h(ID_{\mathcal{U}_i}    H_{\mathcal{U}_i})</math>,</li> <li><math>D_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i}    IP_i    K_{\mathcal{U}_i})</math>,</li> <li><math>W_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i}    IP_i) \oplus h(ID_{\mathcal{U}_i}    H_{\mathcal{U}_i})</math>,</li> </ol> </li> <li><math>GW_N</math> selects the parameter <math>r</math> and <math>t</math> which represents the accidental rotational and translation shift of minutia points under registration and authentication phase.</li> <li>Then <math>GW_N</math> defines a minimum matching score <math>\mathcal{T}</math> (threshold) between two sets of hash values .</li> </ul> <p style="text-align: center;"><math>\xleftarrow{\text{Via Secure Communication Channel}} \langle h(), A_{\mathcal{U}_i}, D_{\mathcal{U}_i}, W_{\mathcal{U}_i}, GH_{SUM}, r, t, \mathcal{T} \rangle</math> to <math>\mathcal{U}_i</math></p>
<b>Step 3:</b> for $\mathcal{U}_i$	
<p><math>\mathcal{U}_i</math> stores <math>h(), P, A_{\mathcal{U}_i}, D_{\mathcal{U}_i}, W_{\mathcal{U}_i}, GH_{SUM}, r, t, \mathcal{T}</math> into <math>SC_{\mathcal{U}_i}</math></p>	

and translation shift of minutia points under registration and authentication phase.

Then,  $\mathcal{U}_i$  computes  $H'_{SUM} = \{h'_{SUM1}(c'_{11}, c'_{21}, \dots, c'_{n1}), h'_{SUM2}(c'_{12}, c'_{22}, \dots, c'_{n2}), \dots, h'_{SUMk}(c'_{1k}, c'_{2k}, \dots, c'_{nk})\}$ . Where  $h'_{SUMj}(c'_{1j}, c'_{2j}, \dots, c'_{nj}) = \sum_{i=1}^n \frac{(rc_{ij}+t)(1-(rc_{ij}+t)^m)}{1-(rc_{ij}+t)}$ . Finds the matching score  $S = H_{SUM} \cap \{H'_{SUM} \pm \epsilon\}$ . Where  $\{H'_{SUM} \pm \epsilon\} = \{h'_{SUM1}(c'_{11}, c'_{21}, \dots, c'_{n1}) \pm \epsilon, h'_{SUM2}(c'_{12}, c'_{22}, \dots, c'_{n2}) \pm \epsilon, \dots, h'_{SUMk}(c'_{1k}, c'_{2k}, \dots, c'_{nk}) \pm \epsilon\}$  and  $\epsilon$  represents the minimum error. If  $(D'_{\mathcal{U}_i} = D_{\mathcal{U}_i})$  and  $(S > \mathcal{T})$ , then  $\mathcal{U}_i$  Computes  $h(ID_{\mathcal{U}_i} || H_{\mathcal{U}_i}) = W_{\mathcal{U}_i} \oplus h(ID_{\mathcal{U}_i} || IP_i)$  and finds the current time-stamp  $T_{\mathcal{U}_i}$ . Then, computes  $m_1 = h(ID_{\mathcal{U}_i} || ID_{SN_j} || h(ID_{\mathcal{U}_i} || H_{\mathcal{U}_i}) || K_{\mathcal{U}_i} || T_{\mathcal{U}_i})$ . Subsequently,  $\mathcal{U}_i$  construct the message  $M_1 = \langle ID_{\mathcal{U}_i}, ID_{SN_j}, K_{\mathcal{U}_i}, m_1, T_{\mathcal{U}_i} \rangle$ . Finally,  $\mathcal{U}_i$  transmits  $\langle M_1 \rangle$  to  $SN_j$

**Step A2:** After receiving the message  $M_1$ ,  $SN_j$  confirms if  $(T' - T_{\mathcal{U}_i} < \Delta T)$ , then, if  $(b_i = 1, \forall i \in [1, \beta]$  such that  $h_l(K_{\mathcal{U}_i}) = i)$ , then  $SN_j$  computes  $h(ID_{\mathcal{U}_i} || H_{\mathcal{U}_i})' = h(ID_{\mathcal{U}_i} || \sum_{l=0}^{\beta} h_l(K_{\mathcal{U}_i}))$ ,  $m'_1 = h(ID_{\mathcal{U}_i} || ID_{SN_j} || h(ID_{\mathcal{U}_i} || H_{\mathcal{U}_i})' || K_{\mathcal{U}_i} || T_{\mathcal{U}_i})$ .

If  $(m'_1 = m_1)$ ,  $SN_j$  finds its current time-stamp  $T_{SN_j}$  and computes the session  $Key = h((R_{SN_j} \times K_{\mathcal{U}_i}) || T_{\mathcal{U}_i} || T_{SN_j})$  (Based on *ECDH* key exchange).

Then  $SN_j$  computes  $m_2 = h(ID_{\mathcal{U}_i} || ID_{SN_j} || h(ID_{\mathcal{U}_i} || H_{\mathcal{U}_i})' || K_{SN_j} || T_{SN_j})$ .

Subsequently  $SN_j$  evaluates the message  $M_2 = \langle ID_{\mathcal{U}_i}, ID_{SN_j}, K_{SN_j}, m_2, T_{SN_j} \rangle$ . Finally,  $SN_j$  transmits  $M_2$  to  $\mathcal{U}_i$ .

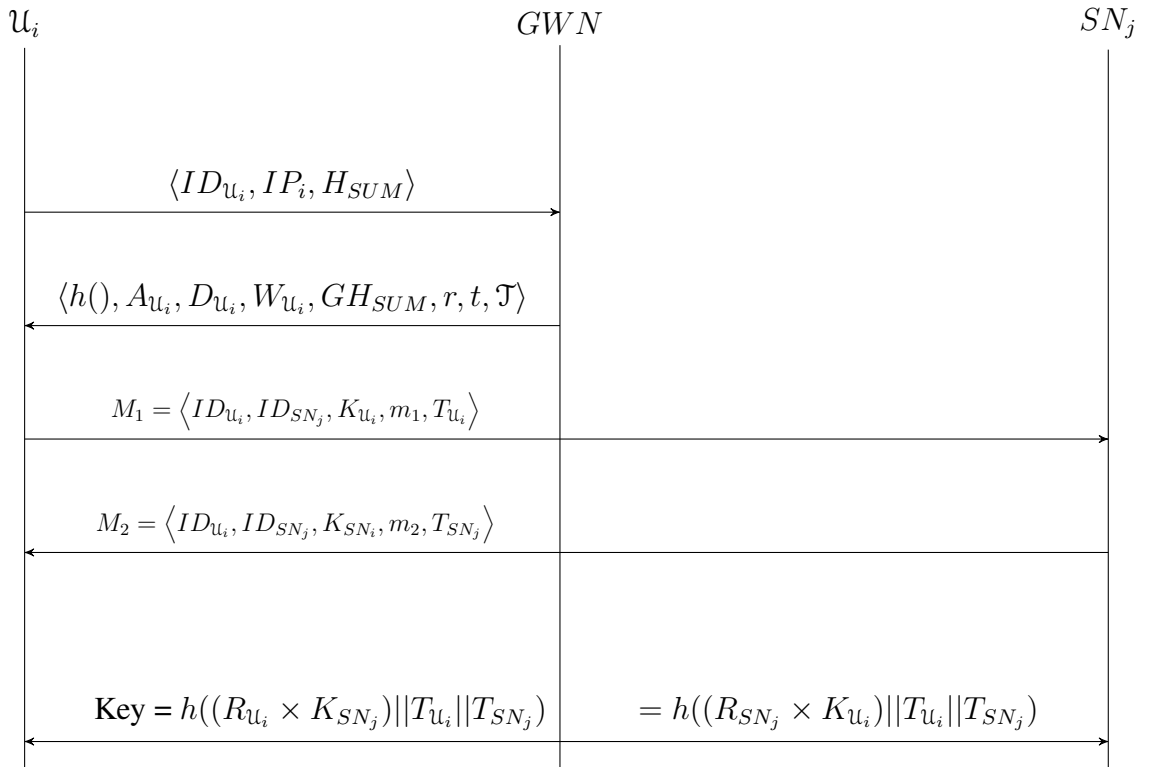
**Step A3:** If  $(T'' - T_{SN_j} < \Delta T)$ ,  $\mathcal{U}_i$  computes  $m'_2 = h(ID_{\mathcal{U}_i} || ID_{SN_j} || h(ID_{\mathcal{U}_i} || H_{\mathcal{U}_i}) || K_{SN_j} || T_{SN_j})$ . If  $(m'_2 = m_2)$ , then,  $\mathcal{U}_i$  establish the session  $Key = h((R_{\mathcal{U}_i} \times K_{SN_j}) || T_{\mathcal{U}_i} || T_{SN_j})$  with  $SN_j$  (Based on *ECDH* key exchange). Otherwise, the protocol rejects  $\mathcal{U}_i$ .

The sequence diagram of the message transmission for the user registration, authentication and key establishment phase is shown in following Figure 8.1.

## 8.4 Performance and Security Analysis:

For the binary vector  $B$  of size  $V$ , the probability that a definite bit is not set to 1 by any of the  $\beta$  number of hash functions is  $\left(1 - \frac{1}{V}\right)^\beta$ . After representing  $\alpha$  numbers of

**Figure 8.1:** Sequence Diagram 6 for Registration, Authentication and Key Establishment





## 8.4 Performance and Security Analysis:

**Table 8.2:**  $\mathcal{U}_i$ 's authentication and Key Exchange Phase

Step 4: for User ( $\mathcal{U}_i$ )	Step 5: for the sensor node ( $SN_j$ )
<ul style="list-style-type: none"> <li><math>\mathcal{U}_i</math> inserts the smart card <math>SC_{\mathcal{U}_i}</math> into card reader and provides <math>ID_{\mathcal{U}_i}, PW_{\mathcal{U}_i}, BIO'_{\mathcal{U}_i}</math>. Then, computes  1. <math>IP_i = h(ID_{\mathcal{U}_i}    PW_{\mathcal{U}_i}    RN_{\mathcal{U}_i})</math>  2. <math>K_{\mathcal{U}_i} = IP_i \oplus A_{\mathcal{U}_i}, D_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i}    IP_i    K_{\mathcal{U}_i}),</math>  3. <math>H_{SUM} = GH_{SUM} \oplus K_{\mathcal{U}_i} = h_{SUM1}(c_{11}, c_{21}, \dots, c_{n1}) \oplus K_{\mathcal{U}_i} \oplus h_{SUM2}(c_{12}, c_{22}, \dots, c_{n2}) \oplus K_{\mathcal{U}_i} \oplus \dots, h_{SUMk}(c_{1k}, c_{2k}, \dots, c_{nk}) \oplus K_{\mathcal{U}_i} \oplus K_{\mathcal{U}_i},</math>  4. <math>W_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i}    IP_i) \oplus h(ID_{\mathcal{U}_i}    H_{\mathcal{U}_i}),</math></li> <li><math>\mathcal{U}_i</math> Extract the minutia points <math>c'_{ij}</math> from the fingerprint <math>BIO'_{\mathcal{U}_i}</math> such that <math>c'_{ij} = rc_{ij} + t</math>, where the parameter <math>r</math> and <math>t</math> represents the accidental rotational and translation shift of minutia points under registration and authentication phase.</li> <li>Then, <math>\mathcal{U}_i</math> computes <math>H'_{SUM} = \{h'_{SUM1}(c'_{11}, c'_{21}, \dots, c'_{n1}), h'_{SUM2}(c'_{12}, c'_{22}, \dots, c'_{n2}), \dots, h'_{SUMk}(c'_{1k}, c'_{2k}, \dots, c'_{nk})\}</math>  Where <math display="block">h'_{SUMj}(c'_{1j}, c'_{2j}, \dots, c'_{nj}) = \sum_{i=1}^n \frac{(rc_{ij}+t)(1-(rc_{ij}+t)^m)}{1-(rc_{ij}+t)}</math></li> <li>Find the matching score <math>S = H_{SUM} \cap \{H'_{SUM} \pm \epsilon\}</math>.  Where <math>\{H'_{SUM} \pm \epsilon\} = \{h'_{SUM1}(c'_{11}, c'_{21}, \dots, c'_{n1}) \pm \epsilon, h'_{SUM2}(c'_{12}, c'_{22}, \dots, c'_{n2}) \pm \epsilon, \dots, h'_{SUMk}(c'_{1k}, c'_{2k}, \dots, c'_{nk}) \pm \epsilon\}</math> and <math>\epsilon</math> represents the minimum error.</li> </ul> <p><b>if</b> (<math>D'_{\mathcal{U}_i} = D_{\mathcal{U}_i}</math>) <b>and</b> (<math>S &gt; \mathcal{T}</math>) <b>then</b>  <math>\mathcal{U}_i</math> Computes <math>h(ID_{\mathcal{U}_i}    H_{\mathcal{U}_i}) = W_{\mathcal{U}_i} \oplus h(ID_{\mathcal{U}_i}    IP_i)</math> and finds the current time-stamp <math>T_{\mathcal{U}_i}</math>. Then computes <math>m_1 = h(ID_{\mathcal{U}_i}    ID_{SN_j}    h(ID_{\mathcal{U}_i}    H_{\mathcal{U}_i})    K_{\mathcal{U}_i}    T_{\mathcal{U}_i})</math>. Subsequently, <math>\mathcal{U}_i</math> construct the message <math>M_1 = \langle ID_{\mathcal{U}_i}, ID_{SN_j}, K_{\mathcal{U}_i}, m_1, T_{\mathcal{U}_i} \rangle</math>  <math>\mathcal{U}_i</math> transmits <math>\langle M_1 \rangle</math> to <math>SN_j</math>  ViaPublicCommunicationChannel  <b>end</b>  <b>else</b>  <math>\mathcal{U}_i</math> is inauthentic. Stolen smart card attack possible. Abort the protocol.  <b>end</b></p>	<p>After receiving the message <math>M_1</math>, <math>SN_j</math> confirms</p> <p><b>if</b> <math>T' - T_{\mathcal{U}_i} &lt; \Delta T</math>, <b>then</b>  <b>if</b> <math>b_i = 1, \forall l \in [1, \beta]</math> such that <math>h_l(K_{\mathcal{U}_i}) = i</math> <b>then</b>  <math>SN_j</math> computes <math>h(ID_{\mathcal{U}_i}    H_{\mathcal{U}_i})' = h(ID_{\mathcal{U}_i}    \sum_{l=0}^{\beta} h_l(K_{\mathcal{U}_i}))</math>,  <math>m'_1 = h(ID_{\mathcal{U}_i}    ID_{SN_j}    h(ID_{\mathcal{U}_i}    H_{\mathcal{U}_i})'    K_{\mathcal{U}_i}    T_{\mathcal{U}_i})</math>  <b>end</b>  <b>else</b>  Reject <math>\mathcal{U}_i</math> and abort this phase.  <b>end</b>  <b>if</b> (<math>m'_1 = m_1</math>) <b>then</b>  <math>SN_j</math> finds its current time-stamp <math>T_{SN_j}</math>,  <math>SN_j</math> computes the session <math>Key = h((R_{SN_j} \times K_{\mathcal{U}_i})    T_{\mathcal{U}_i}    T_{SN_j})</math> (Based on <math>ECDH</math> key exchange).  Then <math>SN_j</math> computes <math>m_2 = h(ID_{\mathcal{U}_i}    ID_{SN_j}    h(ID_{\mathcal{U}_i}    H_{\mathcal{U}_i})'    K_{SN_j}    T_{SN_j})</math>.  Subsequently <math>SN_j</math> evaluates the message <math>M_2 = \langle ID_{\mathcal{U}_i}, ID_{SN_j}, K_{SN_j}, m_2, T_{SN_j} \rangle</math>  <b>end</b>  <b>else</b>  <math>\mathcal{U}_i</math> is not authorized at <math>SN_j</math>. <math>SN_j</math> rejects <math>\mathcal{U}_i</math> and abort this phase.  <b>end</b>  <math>SN_j</math> transmits <math>M_2</math> to <math>\mathcal{U}_i</math>  ViaPublicCommunicationChannel  <b>end</b>  <b>else</b>  Replay, denial of service, bogus message flooding and resource exhaustion attacks possible. <math>SN_j</math> rejects <math>\mathcal{U}_i</math> and aborts this phase.  <b>end</b></p>
<p><b>Step 6: for <math>\mathcal{U}_i</math></b>  <b>if</b> <math>T'' - T_{SN_j} &lt; \Delta T</math> <b>then</b>  <math>\mathcal{U}_i</math> Computes <math>m'_2 = h(ID_{\mathcal{U}_i}    ID_{SN_j}    h(ID_{\mathcal{U}_i}    H_{\mathcal{U}_i})    K_{SN_j}    T_{SN_j})</math> <b>if</b> <math>m'_2 = m_2</math> <b>then</b>  Establish the session <math>Key = h((R_{\mathcal{U}_i} \times K_{SN_j})    T_{\mathcal{U}_i}    T_{SN_j})</math> with <math>SN_j</math> (Based on <math>ECDH</math> key exchange)  <b>end</b>  <b>else</b>  <math>SN_j</math> is not validated by <math>\mathcal{U}_i</math>. Aborts the protocol.  <b>end</b>  <b>end</b>  <b>else</b>  Replay and energy exhausting attack possible. Abort this key exchange phase.  <b>end</b></p>	

## 8.4 Performance and Security Analysis:

users, the probability of the given bit is not set to zero is  $\left(1 - \frac{1}{V}\right)^{\beta\alpha}$ . Therefore, the probability of the bit set to 1 is  $1 - \left(1 - \frac{1}{V}\right)^{\beta\alpha}$ . If all the  $\beta$  positions of the vector  $B$  computed by each hash function of the set  $H$  sets to 1, the user belongs to the set  $B$ . Therefore, the probability of a user not belonging to the set  $B$  is stated as follows:

$$f_+ = \left(1 - \left(1 - \frac{1}{V}\right)^{\beta\alpha}\right)^\alpha \equiv \left(1 - e^{-\frac{\beta\alpha}{V}}\right)^\beta.$$

Hence, we have

$$f_+ = e^{\beta \ln \left(1 - e^{-\frac{\beta\alpha}{V}}\right)}$$

This represents that the probability of false positive  $f_+$  decreased as the size of vector  $V$  increases and  $f_+$  increases as the number of users  $\alpha$  increases.

To minimize the value of  $f_+$ , take the derivatives of  $\beta \ln \left(1 - e^{-\frac{\beta\alpha}{V}}\right)$  with respect to  $\beta$  and find out:

$$\frac{d\left(\beta \ln \left(1 - e^{-\frac{\beta\alpha}{V}}\right)\right)}{d\beta} = \ln \left(1 - e^{-\frac{\beta\alpha}{V}}\right) + \frac{\beta\alpha}{V} \frac{e^{-\frac{\beta\alpha}{V}}}{1 - e^{-\frac{\beta\alpha}{V}}}$$

To get the optimal value of  $\beta$ , make the derivatives equal to zero and find out  $\beta_{optimal} = \frac{V}{\alpha} \ln 2$ .

Therefore, the minimum probability of false positive is :

$$f_+ = e^{\left(\frac{V}{\alpha} \ln 2\right) \ln \left(1 - e^{-\frac{(\frac{V}{\alpha} \ln 2)\alpha}{V}}\right)} = (0.6185)^{\frac{V}{\alpha}}$$

### 8.4.1 Security Analysis

The formal and informal validation, security analysis of our proposed AKE mechanism based on basic cryptographic techniques and AVISPA tool indicates that the proposed mechanism provides important security features and resists various well-known security attacks.

#### 8.4.1.1 Informal security analysis

**Proposition 1.** The proposed protocol is resilient to stolen smart card attack.

*Proof.* If An adversary  $\mathcal{A}$  extracts the value of  $A_{\mathcal{U}_i} = IP_i \oplus K_{\mathcal{U}_i}$ ,  $D_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i} || IP_i || K_{\mathcal{U}_i})$ ,  $W_{\mathcal{U}_i} = h(ID_{\mathcal{U}_i} || IP_i) \oplus h(ID_{\mathcal{U}_i} || H_{\mathcal{U}_i})$  from stolen  $SC_{\mathcal{U}_i}$  using simple or differential power analysis techniques [14],  $\mathcal{A}$  can not ascertain the value of secret parameters such as  $K_{\mathcal{U}_i}$ ,  $h(ID_{\mathcal{U}_i} || H_{\mathcal{U}_i})$ . Because the secret credentials  $A_{\mathcal{U}_i}$ ,  $D_{\mathcal{U}_i}$ ,  $W_{\mathcal{U}_i}$  of the user  $\mathcal{U}_i$  is protected with user's inherence ( $BIO_{\mathcal{U}_i}$ ) and knowledge ( $PW_{\mathcal{U}_i}$ ).  $\square$

**Proposition 2.** The proposed protocol is resilient to the man-in-the middle attack.

*Proof.* Assume that  $\mathcal{A}$  intercepts the message  $M_1, M_2$  during the user authentication and key exchange phase.

Suppose  $\mathcal{A}$  generates its own current time-stamp  $T_{\mathcal{A}}$ . But,  $\mathcal{A}$  does not know the value of  $h(ID_{\mathcal{U}_i} || H_{\mathcal{U}_i})$  and also the set of hash function  $H$ . Without knowing  $h(ID_{\mathcal{U}_i} || H_{\mathcal{U}_i})$ , it is computationally unfeasible for  $\mathcal{A}$  to modify the message  $M_1, M_2$ . Therefore, it is clear that our protocol resists the man-in-the-middle attack.  $\square$

**Proposition 3.** The proposed protocol is resilient to replay attack.

*Proof.* Assume  $\mathcal{A}$  taps the message  $M_1, M_2$  through the authentication and key exchange phase and replays this information again after some interval to the gateway node  $GW_N$ . But, this replayed information will be recognized as a replay one due to validation of the time-stamp  $T_{\mathcal{U}_i}, T_{SN_j}$  by  $SN_j, \mathcal{U}_i$  respectively.  $\square$

**Proposition 4.** The proposed protocol is resilient to user impersonation attacks and forgery attack.

*Proof.* To imitate a user a user  $\mathcal{U}_i$ , the adversary  $\mathcal{A}$  should know the value  $H_{\mathcal{U}_i}$  for generating the message  $M_1 = \langle ID_{\mathcal{U}_i}, ID_{SN_j}, K_{\mathcal{U}_i}, m_1, T_{\mathcal{U}_i} \rangle$ . Although, the value of  $H_{\mathcal{U}_i}$  can not be decided (based on the intercepted message  $M_1$ ) by the adversary  $\mathcal{A}$  without the password  $PW_{\mathcal{U}_i}$ , bio-metric information  $BIO_{\mathcal{U}_i}$ , smart card  $SC_{\mathcal{U}_i}$  and the set of hash function  $H$ . The verification of  $h(ID_{\mathcal{U}_i} || H_{\mathcal{U}_i})' = h(ID_{\mathcal{U}_i} || \sum_{l=0}^{\beta} h_l(K_{\mathcal{U}_i})), \forall l \in [1, \beta]$  at the sensor node  $SN_j$  achieves the integrity and authenticity which supports to counter the user impersonation attacks and reproduction or forgery attack.  $\square$

**Proposition 5.** The proposed protocol is resilient to energy exhausting attack.

*Proof.* To withstand the exhausting energy attack or false message flooding (which can drain the resources of WSNs & IoT), we reject the unacceptable users (the adversary  $\mathcal{A}$ ) at the fundamental level (i.e. at  $SN_j$  sensor node itself) of message communication. For a sensor node  $SN_j$ , the energy needed for computation is less related to message transmission. Consequently, we verify the correctness of  $b_i = 1, \forall i \in [1, \beta]$  such that  $h_l(K_{u_i}) = i$  such that  $h_l(K_{u_i}) = i$  at sensor node  $SN_j$ .

The correct use of  $m_1$  assures the user applies to the authorized group. Energy expected in receiving and transmitting 1-bit data (at the data rate of 12.4 Kb/s) are  $28.6 \mu \text{ Joule}$ ,  $28.6 \mu \text{ Joule}$  respectively [119]. Moreover, we consider  $S, D$  are the size and density (total number of nodes in the circular area with a radius equal to the communication range of the sensor node) of WSNs & IoT. If an wrong user  $\mathcal{A}$  is not dropped at primary level,  $\mathcal{A}$  can incorporate total energy equal to  $E_{total}$  by transmitting a message  $M'_1 = \langle ID_{u_i}, ID_{SN_j}, K_{u_i}, m_1, T_{u_i} \rangle$  of size  $L$  bytes. Where  $E_{total}$  can be estimated as follows :

$$E_{total} = S \left( L \times (59.2 + 28.6D) \right) \mu \text{ Joule}.$$

However, in the proposed protocol we dismiss the adversary  $\mathcal{A}$  at the elementary level which conserves the total energy of  $(E_{total} - E_h) \mu \text{ Joule}$ . Where  $E_h$  is the energy needed in computing and checking the condition of  $b_i$  and  $m_1$ . Therefore, our protocol resists the exhausting energy attacks.  $\square$

#### 8.4.1.2 Formal security analysis

The verifying result of security traits of our proposed protocol applying AVISPA (Automated Validation of Internet Security Protocols and Applications) [7] tool is demonstrated in Fig. 8.3. This result describes that our proposed protocol counters several attacks (like impersonation attack, replay attack, man-in-the-middle attack etc.) utilising Dolev-Yao model [110] including a bounded number of sessions, On-the-Fly Model-Checker (OFMC) backend and stipulated goals.

Table 8.3: Security Verification Output by AVISPA tool

SUMMARY
SAFE
DETAILS
BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
/home/cmb-lab-22/Desktop/SecureAuthentication.if
GOAL
as_specified
BACKEND
OFMC
COMMENTS
STATISTICS
parseTime: 0.00s
searchTime: 0.37s
visitedNodes: 108 nodes
depth: 9 piles
<div><div>Tools</div><div>HPSL</div><div>HPSL2IF</div><div>IF</div><div>OFMC</div><div>ATSE</div><div>SATMC</div><div>TA4SP</div><div><div>Save file</div><div>View CAS+</div></div><div><div>Choose Tool option and press execute</div><div>Execute</div></div></div>

## 8.5 Comparison of Security Features and Computational Overhead

Based on the important security points of WSNs & IoT, Table 8.4 describes that our proposed protocol is pretty secure related to other existing AKE protocols. We execute ECDH key exchange and SHA-1 secure hash function on MicaZ sensor node (processor 8-bit ATmega128L Atmel, RAM 4 KB, ROM 128 KB, 512 KB EEPROM and 2 AA battery) with TinyOS [111] component-based working framework and nesC [112] event-driven programming language. The time (in milliseconds) and energy (in Joule) estimated by these cryptographic operations is shown in Table 8.5. Table 8.6 depicts the computational overhead comparison and confirms that our proposed protocol is suited for the sensor node  $SN_j$ .

**Table 8.4:** Comparisons of Security Features

Security Feature	Althobaiti et al. [59]	A.K.Das [81]	Choi et al. [61]	Proposed Proposed
Resist stolen smart card attack	Yes	No	No	Yes
secure password update	No	Yes	Yes	Yes
Secure bio-metric update	No	Yes	Yes	Yes
Non-repudiation	Yes	Yes	Yes	Yes
Formal security analysis	No	Yes	No	Yes
No privileged-insider attack	Yes	Yes	Yes	Yes

**Table 8.5:** Time and Energy consumed by MicaZ sensor node

Procedure	Time (in millisecond)	Energy (in $\mu$ Joule)
AES-128 Encryption and Decryption	$TS_{Enc/Dec} \approx 5.05$	121.2
SHA-1	$TS_H \approx 3.63$	87.12
Message authentication code	$TS_{Mac} \approx 4.82$	115.68
Elliptic Curve Point Multiplication	$TS_M \approx 370$	8880

**Table 8.6:** Comparison of Protocols based on Computational Performance

protocol	Computational Overhead on $SN_j$	Time (in millisecond)	Energy (in $\mu$ Joule)
Althobaiti et al.'s [59]	$TS_{Dec} + TS_{Mac} + TS_H$	13.5	324
A.K.Das's [81]	$TS_{Dec} + 2 TS_H$	12.31	295.44
Choi et.al.'s [61]	$2TS_{Dec} + TS_M + 6TS_H$	401.88	9645.12
Proposed Protocol	$(\beta + 3) TS_H + TS_M$	$3.63 \beta + 380.89$	$87.12 \beta + 9141.36$

Note:  $TS_H$ ,  $TS_{Dec}$ ,  $TS_{Mac}$  and  $TS_M$  represent the time required to execute secure hashing, message encryption/decryption, message authentication code and point multiplication operation on elliptic curve respectively on sensor node  $SN_j$ .

## 8.6 Summary

In this chapter, we have illustrated the importance of authenticated key exchange mechanism for the real-world applicability of WSNs & IoT. Next, we have addressed security issues associated with the sensor nodes of WSNs & IoT and some fundamental ideas of user authentication, session key establishment procedure, symmetric hash function and bloom filter. Then, we have proposed a symmetric hash function and a bloom filter based reliable and cost-effective authenticated key exchange method. We have applied the symmetric hash function for defending user's fingerprint minutia from being compromised by the adversary and authenticating the user without a pre-alignment among the test and the registered fingerprint templates. We have applied the bloom filter to stop unauthorized querying information transmission at the beginning level (i.e., at the sensor node itself) to avoid false data flooding from the sensor nodes to the gateway node. We have done the security analysis and verification utilising a broadly trusted and robust tool AVISPA. Finally, we have shown the comparative study of our proposed protocol with other existing protocols based on security traits and computational costs which imply that the proposed protocol is reliable, secure and efficient.

## Chapter 9

# Conclusions and Future Work

“Problems worthy of attack / Prove their worth by hitting back.” Piet Hein

This chapter concludes the significant contributions of the thesis, and it also highlights some future research trends in the area of user authentication and key establishment for WSNs & IoT. In this thesis, we have discussed the importance, security issues, challenges, motivation, objectives, literature survey, performance analysis and proposed protocols of user authentication and key establishment for real-time applications of WSNs & IoT.

A user who wants to avail the services of a WSNs & IoT must get authenticated by the WSNs & IoT. The gateway nodes, present in the WSN & IoT, collect the real-time information from sensor nodes and store it in their memory which the legitimate users can query. However, since the *GWN* collect the data from sensor nodes in regular intervals, the data present at the *GWN* may not be the real-time data and such data can mainly be use for statistical purpose or analytics. Hence, it is also required for the users to communicate with the sensor nodes directly to collect the real-time information. To communicate with the sensor nodes directly, the user needs to authenticate against the gateway nodes as well as the sensor nodes to ensure secure communication. Given the resource-constrained nature of sensor nodes, it is essential to design efficient, reliable and light-weight user authentication and key establishment protocols with no compromise on security.

Well-defined security analysis (described in Chapter 2,3 and 4) of existing user authentication protocols of the literature proves that the protocols are exposed to several



---

attacks like sensor node impersonation, user impersonation, stolen smart card, replay, man-in-the-middle, sensor energy and resources exhausting, gateway node bypassing and the attacks based on legitimate users. The performance analysis demonstrates that the existing protocols are inadequate considering the computational overhead.

Therefore, considering the major security requirements, various security attacks, topology, resource constraint, challenges and applications of WSNs & IoT, we have proposed total six secure and efficient user authentication and key establishment protocols.

1. The first proposed protocol based on smart card and ECC is efficient for two-factor user authentication. In this protocol, we have design secure time-stamp to avoid replay and energy exhausting attack. Choi et al. 's protocol (proposed in 2014) stores a confidential parameter which can help in finding the users password. Since we are not saving any confidential parameter into the sensor node which can assist in finding the users password, therefore, our protocol is resilient against node capture attacks. Choi et al. 's protocol stores a unique confidential parameter in the different smart cards which can help in frequency analysis attack to find the user's password. Since we are not saving any unique confidential parameter into the different smart cards, therefore, our proposed protocol is resilient against stolen smart card attack. Our proposed protocol withstood the security pitfalls of Choi et al.'s protocol and improved the computational performance also. Since we are using a unique long-term secret key for each sensor node, therefore, sensor impersonation attack is not possible. Our proposed protocol yields mutual authentication and session key establishment which can help in resisting man-in-the-middle and user impersonation attacks.
2. The second protocol based on fuzzy extractor and ECDH is suitable for efficient multi-factor user authentication. In this protocol, the execution time for the sensor node is very less compared to other existing protocol because we shifted the overload of the performance of elliptic curve point multiplication from the sensor node to the gateway node with improved security features.
3. The third protocol based on LU Decomposition is suitable for light-weight session key establishment between the user and the sensor node. In this protocol, we have

---

improved the security and computational performance for the key establishment using LU Decomposition with lesser memory storage requirement.

4. The fourth protocol based on Chinese Remainder Theorem is ideal for quickly authenticating the user without the help of the gateway node.
5. The fifth protocol based on Bloom Filter is used for the WSNs & IoT of large hop count (i.e., the large number of intermediate sensor nodes through which data must pass between source sensor node and gateway node). The novelty of this protocol is, it eliminates unauthorized querying message transmission at the initial level (i.e., at the sensor node itself) to avoid bogus message flooding from the sensor nodes to the gateway node (which exhausts the resources of WSNs).
6. The sixth protocol based on Symmetric Hash Functions is significant for authenticating the users without pre-alignment between the test and the registered minutia points of the fingerprint for multi-hop WSNs & IoT.

Our proposed protocols are secure against stolen smart card attack, user impersonation attack, sensor node impersonation attack, sensor node capture attack, replay attack, man-in-the-middle attack. The proposed authentication protocols provide various security features such as a robust session key establishment, mutual authentication, multi-factor authentication, efficient password and biometric data update, credentials' confidentiality, information freshness, message integrity. The proposed protocols are effective concerning the computational overhead of the resource-constrained sensor nodes, and they conserve communication bandwidth, energy. Consequently, the protocols are suitable for management of resource-constrained ubiquitous computing devices. Accordingly, the proposed protocols can be applied in several real-world applications consisting of resource constraint sensor devices of WSNs & IoT wherever bio-metric based secure user authentication and efficient session key establishment is needed. The proposed protocols can be applied for the implementation of bio-metric based secure authentic banking and financial transactions using the smart card at automated teller machines (ATM) and mobile point-of-sale (POS) devices.

We have given security proof using the random oracle model and BAN logic to assure the correctness of various security traits suggested in the proposed protocols. Then, we have done the security analysis and verification applying popular and robust

---

tools such as AVISPA and Scyther. Through the precise security analysis utilizing mathematical functions and simulation tools, we have confirmed that the proposed protocols fulfill the acceptable security specifications and resist the security attacks observed in existing protocols of user authentication and key establishment for WSNs & IoT. Finally, we have performed the comparative analysis of our protocols with other existing protocols based upon security properties and computational cost which prove that our proposed protocols are efficient, secure and significant for WSNs & IoT.

In future, we would like to recommend Artificial Neural Network, Hyper-Elliptic Curve Cryptography, Optimized Bloom Filter, Blockchain based authenticated key exchange protocols which would be ideal for WSNs, IoT and IoT based Cloud Services.

# References

- [1] YOUNSUNG CHOI, DONGHOON LEE, JIYE KIM, JAEWOOK JUNG, JUNGHYUN NAM, AND DONGHO WON. **Security enhanced user authentication protocol for wireless sensor networks using elliptic curves cryptography.** *Sensors*, **14**(6):10081–10106, 2014.
- [2] IAN F AKYILDIZ, WEILIAN SU, YOGESH SANKARASUBRAMANIAM, AND ERDAL CAYIRCI. **Wireless sensor networks: a survey.** *Computer networks*, **38**(4):393–422, 2002.
- [3] PAULA FRAGA-LAMAS, TIAGO M FERNÁNDEZ-CARAMÉS, MANUEL SUÁREZ-ALBELA, LUIS CASTEDO, AND MIGUEL GONZÁLEZ-LÓPEZ. **A review on internet of things for defense and public safety.** *Sensors*, **16**(10):1644, 2016.
- [4] I-PIN CHANG, TIAN-FU LEE, TSUNG-HUNG LIN, AND CHUAN-MING LIU. **Enhanced two-factor authentication and key agreement using dynamic identities in wireless sensor networks.** *Sensors*, **15**(12):29841–29854, 2015.
- [5] **A Multi Factor Authentication. Available online.** <https://dev98.de/2016/11/19/a-multi-factor-authentication-quickstart/>, accessed on 23 August 2018.
- [6] **Power analysis.** [https://en.wikipedia.org/wiki/Power\\_analysis](https://en.wikipedia.org/wiki/Power_analysis), accessed on 23 August 2018.
- [7] **AVISPA Tool. Available online:.** <http://www.avispa-project.org/>, accessed on 29 July 2018.

- [8] MICHAEL MITZENMACHER. **Compressed bloom filters.** *IEEE/ACM Transactions on Networking (TON)*, **10**(5):604–612, 2002.
- [9] KEVIN ASHTON ET AL. **That internet of things thing.** *RFID journal*, **22**(7):97–114, 2009.
- [10] MUHAMMAD UMAIR ASLAM, ABDELOUAHID DERHAB, KASHIF SALEEM, HAIDER ABBAS, MEHMET ORGUN, WASEEM IQBAL, AND BABER ASLAM. **A survey of authentication schemes in telecare medicine information systems.** *Journal of medical systems*, **41**(1):14, 2017.
- [11] ZINAIDA BENENSON, FELIX C GÄRTNER, AND DOGAN KESDOGAN. **User Authentication in Sensor Networks.** In *GI Jahrestagung (2)*, pages 385–389. Citeseer, 2004.
- [12] MOHAMED AMINE FERRAG, LEANDROS A MAGLARAS, HELGE JANICKE, JIANMIN JIANG, AND LEI SHU. **Authentication protocols for Internet of Things: a comprehensive survey.** *Security and Communication Networks*, vol. **2017**, 2017.
- [13] ANDREW TEOH BENG JIN, DAVID NGO CHEK LING, AND ALWYN GOH. **Bio-hashing: two factor authentication featuring fingerprint data and tokenised random number.** *Pattern recognition*, **37**(11):2245–2255, 2004.
- [14] PAUL KOCHER, JOSHUA JAFFE, AND BENJAMIN JUN. **Differential power analysis.** In *Annual International Cryptology Conference*, pages 388–397. Springer, 1999.
- [15] DOUGLAS R STINSON. **Some observations on the theory of cryptographic hash functions.** *Designs, Codes and Cryptography*, **38**(2):259–277, 2006.
- [16] SERGEY TULYAKOV, FAISAL FAROOQ, PRAVEER MANSUKHANI, AND VENU GOVINDARAJU. **Symmetric hash functions for secure fingerprint biometric systems.** *Pattern Recognition Letters*, **28**(16):2427–2436, 2007.

- [17] VICTOR S MILLER. **Use of elliptic curves in cryptography.** In *Conference on the theory and application of cryptographic techniques*, pages 417–426. Springer, 1985.
- [18] YEVGENIY DODIS, LEONID REYZIN, AND ADAM SMITH. **Fuzzy extractors: How to generate strong keys from biometrics and other noisy data.** In *International conference on the theory and applications of cryptographic techniques*, pages 523–540. Springer, 2004.
- [19] CASIMIER JOSEPH FRANCISCUS CREMERS. *Scyther: Semantics and verification of security protocols.* Eindhoven University of Technology Eindhoven, Netherlands, 2006.
- [20] MICHAEL BURROWS, MARTIN ABADI, AND ROGER MICHAEL NEEDHAM. **A logic of authentication.** *Proc. R. Soc. Lond. A*, **426**(1871):233–271, 1989.
- [21] MIHIR BELLARE AND PHILLIP ROGAWAY. **Random oracles are practical: A paradigm for designing efficient protocols.** In *Proceedings of the 1st ACM conference on Computer and communications security*, pages 62–73. ACM, 1993.
- [22] LAURENT ESCHENAUER AND VIRGIL D GLIGOR. **A key-management scheme for distributed sensor networks.** In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pages 41–47. ACM, 2002.
- [23] ROBERTO DI PIETRO, LUIGI V MANCINI, AND ALESSANDRO MEI. **Random key-assignment for secure wireless sensor networks.** In *Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks*, pages 62–71. ACM, 2003.
- [24] HAOWEN CHAN, ADRIAN PERRIG, AND DAWN SONG. **Random key predistribution schemes for sensor networks.** In *Security and Privacy, 2003. Proceedings. 2003 Symposium on*, pages 197–213. IEEE, 2003.
- [25] WENLIANG DU, JING DENG, YUNGHSIANG S HAN, SHIGANG CHEN, AND PRAMOD K VARSHNEY. **A key management scheme for wireless sensor networks using deployment knowledge.** In *INFOCOM 2004. Twenty-third Annu-*

- alJoint conference of the IEEE computer and communications societies*, **1**. IEEE, 2004.
- [26] MOHAMED ELTOWEISSY, MOHAMMED MOHARRUM, AND RAVI MUKKAMALA. **Dynamic key management in sensor networks**. *IEEE Communications magazine*, **44**(4):122–130, 2006.
- [27] BOCHENG LAI, SUNGHA KIM, AND INGRID VERBAUWHEDE. **Scalable session key construction protocol for wireless sensor networks**. In *IEEE Workshop on Large Scale RealTime and Embedded Systems (LARTES)*, page 7. Citeseer, 2002.
- [28] YONG HO KIM, HWASEONG LEE, DONG HOON LEE, AND JONGIN LIM. **A key management scheme for large scale distributed sensor networks**. In *IFIP International Conference on Personal Wireless Communications*, pages 437–446. Springer, 2006.
- [29] SENCUN ZHU, SANJEEV SETIA, AND SUSHIL JAJODIA. **LEAP+: Efficient security mechanisms for large-scale distributed sensor networks**. *ACM Transactions on Sensor Networks (TOSN)*, **2**(4):500–528, 2006.
- [30] ASHOK KUMAR DAS AND INDRANIL SENGUPTA. **An effective group-based key establishment scheme for large-scale wireless sensor networks using bi-variate polynomials**. In *Communication Systems Software and Middleware and Workshops, 2008. COMSWARE 2008. 3rd International Conference on*, pages 9–16. IEEE, 2008.
- [31] XINXIN FAN AND GUANG GONG. **Lpkm: A lightweight polynomial-based key management protocol for distributed wireless sensor networks**. In *International Conference on Ad Hoc Networks*, pages 180–195. Springer, 2012.
- [32] ANUP KUMAR MAURYA, VN SASTRY, AND SIBA KUMAR UDGATA. **Crypt-analysis and Improvement of ECC-Based Security Enhanced User Authentication Protocol for Wireless Sensor Networks**. In *International Symposium on Security in Computing and Communication*, pages 134–145. Springer, 2015.

- [33] ANUP KUMAR MAURYA AND VINJAMURI NARSIMHA SASTRY. **Fuzzy Extractor and Elliptic Curve Based Efficient User Authentication Protocol for Wireless Sensor Networks and Internet of Things.** *Information*, **8**(4):136, 2017.
- [34] ANUP KUMAR MAURYA AND VN SASTRY. **User Authentication Scheme for Wireless Sensor Networks and Internet of Things Using Chinese Remainder Theorem.** In *International Symposium on Security in Computing and Communication*, pages 79–94. Springer, 2017.
- [35] ANUP KUMAR MAURYA AND VN SASTRY. **User Authentication Scheme for Wireless Sensor Networks and Internet of Things Using LU Decomposition.** In *International Symposium on Security in Computing and Communication*, pages 39–53. Springer, 2017.
- [36] ANUP KUMAR MAURYA AND VN SASTRY. **Symmetric hash function based secure and efficient authenticated key exchange mechanism for wireless sensor networks.** In *2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pages 1–6. IEEE, 2017.
- [37] ANUP KUMAR MAURYA AND VN SASTRY. **Secure and Efficient Authenticated Key Exchange Mechanism for Wireless Sensor Networks and Internet of Things Using Bloom Filter.** In *Collaboration and Internet Computing (CIC), 2017 IEEE 3rd International Conference on*, pages 173–180. IEEE, 2017.
- [38] ZAHID MAHMOOD, HUANSHENG NING, AND ATAULLAH GHAFOR. **A polynomial subset-based efficient multi-party key management system for lightweight device networks.** *Sensors*, **17**(4):670, 2017.
- [39] RAMON SANCHEZ-IBORRA, JESÚS SÁNCHEZ-GÓMEZ, SALVADOR PÉREZ, PEDRO J FERNÁNDEZ, JOSÉ SANTA, JOSÉ L HERNÁNDEZ-RAMOS, AND ANTONIO F SKARMETA. **Enhancing LoRaWAN Security through a Lightweight and Authenticated Key Management Approach.** *Sensors (Basel, Switzerland)*, **18**(6), 2018.
- [40] LESLIE LAMPORT. **Password authentication with insecure communication.** *Communications of the ACM*, **24**(11):770–772, 1981.



## REFERENCES

---

- [41] RONALD WATRO, DERRICK KONG, SUE-FEN CUTI, CHARLES GARDINER, CHARLES LYNN, AND PETER KRUUS. **TinyPK: securing sensor networks with public key technology.** In *Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, pages 59–64. ACM, 2004.
- [42] HUEI-RU TSENG, RONG-HONG JAN, AND WUU YANG. **An improved dynamic user authentication scheme for wireless sensor networks.** In *Global Telecommunications Conference, 2007. GLOBECOM'07. IEEE*, pages 986–990. IEEE, 2007.
- [43] TSERN-HUEI LEE. **Simple dynamic user authentication protocols for wireless sensor networks.** In *Sensor Technologies and Applications, 2008. SENSORCOMM'08. Second International Conference on*, pages 657–660. IEEE, 2008.
- [44] LEE-CHUN KO. **A novel dynamic user authentication scheme for wireless sensor networks.** In *Wireless Communication Systems. 2008. ISWCS'08. IEEE International Symposium on*, pages 608–612. IEEE, 2008.
- [45] XIAO-YU LIU, TING-LEI HUANG, XIN WANG, AND XIANG-JIAO TANG. **A user authentication scheme based on dynamic password for wireless sensor networks.** In *Intelligent Computing and Integrated Systems (ICISS), 2010 International Conference on*, pages 145–148. IEEE, 2010.
- [46] SK HAFIZUL ISLAM AND GP BISWAS. **Design of improved password authentication and update scheme based on elliptic curve cryptography.** *Mathematical and Computer Modelling*, **57**(11-12):2703–2717, 2013.
- [47] MARTIN MIHAJLOV AND BORKA JERMAN-BLAŽIČ. **On designing usable and secure recognition-based graphical authentication mechanisms.** *Interacting with Computers*, **23**(6):582–593, 2011.
- [48] SHI-QI WANG, JING-YA WANG, AND YONG-ZHEN LI. **The web security password authentication based the single-block hash function.** *IERI Procedia*, **4**:2–7, 2013.

- [49] C-C CHANG AND T-C WU. **Remote password authentication with smart cards.** *IEE Proceedings E (Computers and Digital Techniques)*, **138**(3):165–168, 1991.
- [50] MANIK LAL DAS. **Two-factor user authentication in wireless sensor networks.** *IEEE transactions on wireless communications*, **8**(3):1086–1090, 2009.
- [51] BINOD VAIDYA, JORGE SÁ SILVA, AND JOEL JPC RODRIGUES. **Robust dynamic user authentication scheme for wireless sensor networks.** In *Proceedings of the 5th ACM symposium on QoS and security for wireless and mobile networks*, pages 88–91. ACM, 2009.
- [52] MUHAMMAD KHURRAM KHAN AND KHALED ALGHATHBAR. **Cryptanalysis and security improvements of two-factor user authentication in wireless sensor networks.** *Sensors*, **10**(3):2450–2459, 2010.
- [53] DAOJING HE, YI GAO, SAMMY CHAN, CHUN CHEN, AND JIAJUN BU. **An Enhanced Two-factor User Authentication Scheme in Wireless Sensor Networks.** *Ad hoc & sensor wireless networks*, **10**(4):361–371, 2010.
- [54] RONGGONG SONG. **Advanced smart card based password authentication protocol.** *Computer Standards & Interfaces*, **32**(5-6):321–325, 2010.
- [55] TONG LI, YUHUI ZHENG, AND TI ZHOU. **Efficient Anonymous Authenticated Key Agreement Scheme for Wireless Body Area Networks.** *Security and Communication Networks*, **Vol. 2017**, 2017.
- [56] RUHUL AMIN, SK ISLAM, MUHAMMAD KHURRAM KHAN, ARIJIT KARATI, DEBASIS GIRI, AND SARU KUMARI. **A two-factor RSA-based robust authentication system for multiserver environments.** *Security and Communication Networks*, **vol. 2017**, 2017.
- [57] CHENYU WANG, GUOAI XU, AND WENTING LI. **A Secure and Anonymous Two-Factor Authentication Protocol in Multiserver Environment.** *Security and Communication Networks*, **vol. 2018**, 2018.

- [58] KE ZHANG, KAI XU, AND FUSHAN WEI. **A Provably Secure Anonymous Authenticated Key Exchange Protocol Based on ECC for Wireless Sensor Networks.** *Wireless Communications and Mobile Computing*, vol. 2018, 2018.
- [59] OHODD ALTHOBAITI, MZNAH AL-RODHAAN, AND ABDULLAH AL-DHELAAN. **An efficient biometric authentication protocol for wireless sensor networks.** *International Journal of Distributed Sensor Networks*, 9(5):407971, 2013.
- [60] EUN-JUN YOON AND CHEONSHIK KIM. **Advanced biometric-based user authentication scheme for wireless sensor networks.** *Sensor Letters*, 11(9):1836–1843, 2013.
- [61] YOUNSUNG CHOI, YOUNGSOOK LEE, AND DONGHO WON. **Security improvement on biometric based authentication scheme for wireless sensor networks using fuzzy extraction.** *International Journal of Distributed Sensor Networks*, 12(1):8572410, 2016.
- [62] YAMAN SHARAF-DABBAGH AND WALID SAAD. **On the authentication of devices in the Internet of Things.** In *2016 IEEE 17th International Symposium on*, pages 1–3. IEEE, 2016.
- [63] YOHAN PARK AND YOUNGHO PARK. **Three-factor user authentication and key agreement using elliptic curve cryptosystem in wireless sensor networks.** *Sensors*, 16(12):2123, 2016.
- [64] JONGHO MOON, DONGHOON LEE, YOUNGSOOK LEE, AND DONGHO WON. **Improving biometric-based authentication schemes with smart card revocation/reissue for wireless sensor networks.** *Sensors*, 17(5):940, 2017.
- [65] DONGWOO KANG, JAEWOOK JUNG, HYOUNGSHICK KIM, YOUNGSOOK LEE, AND DONGHO WON. **Efficient and Secure Biometric-Based User Authenticated Key Agreement Scheme with Anonymity.** *Security and Communication Networks*, vol. 2018, 2018.

## REFERENCES

---

- [66] SEYIT A CAMTEPE AND BÜLENT YENER. **Combinatorial design of key distribution mechanisms for wireless sensor networks.** In *European Symposium on Research in Computer Security*, pages 293–308. Springer, 2004.
- [67] DONGGANG LIU AND PENG NING. **Location-based pairwise key establishments for static sensor networks.** In *Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks*, pages 72–82. ACM, 2003.
- [68] DIBYENDU CHAKRABARTI, SUBHAMOY MAITRA, AND BIMAL ROY. **A key pre-distribution scheme for wireless sensor networks: merging blocks in combinatorial design.** *International Journal of Information Security*, **5**(2):105–114, 2006.
- [69] XIAOJIANG DU, YANG XIAO, MOHSEN GUIZANI, AND HSIAO-HWA CHEN. **An effective key management scheme for heterogeneous sensor networks.** *Ad Hoc Networks*, **5**(1):24–34, 2007.
- [70] SEYED HOSSEIN ERFANI, HAMID HS JAVADI, AND AMIR MASOUD RAHMANI. **A dynamic key management scheme for dynamic wireless sensor networks.** *Security and Communication Networks*, **8**(6):1040–1049, 2015.
- [71] KIRK HM WONG, YUAN ZHENG, JIANNONG CAO, AND SHENGWEI WANG. **A dynamic user authentication scheme for wireless sensor networks.** In *Sensor Networks, Ubiquitous, and Trustworthy Computing, 2006. IEEE International Conference on*, **1**, pages 8–pp. IEEE, 2006.
- [72] ANIL KUMAR SUTRALA, ASHOK KUMAR DAS, NEERAJ KUMAR, ALAVALAPATI GOUTHAM REDDY, ATHANASIOS V VASILAKOS, AND JOEL JPC RODRIGUES. **On the design of secure user authenticated key management scheme for multigateway-based wireless sensor networks using ECC.** *International Journal of Communication Systems*, **31**(8):e3514, 2018.
- [73] RONALD L RIVEST, ADI SHAMIR, AND LEONARD ADLEMAN. **A method for obtaining digital signatures and public-key cryptosystems.** *Communications of the ACM*, **21**(2):120–126, 1978.

- [74] WHITFIELD DIFFIE AND MARTIN HELLMAN. **New directions in cryptography.** *IEEE transactions on Information Theory*, **22**(6):644–654, 1976.
- [75] JIANJUN YUAN, CHANGJUN JIANG, AND ZUOWEN JIANG. **A biometric-based user authentication for wireless sensor networks.** *Wuhan University Journal of Natural Sciences*, **15**(3):272–276, 2010.
- [76] ZINAIDA BENENSON, NILS GEDICKE, AND OSSI RAIVIO. **Realizing robust user authentication in sensor networks.** *Real-World Wireless Sensor Networks (REALWSN)*, **14**:52, 2005.
- [77] KUI REN, SHUCHENG YU, WENJING LOU, AND YANCHAO ZHANG. **Multi-user broadcast authentication in wireless sensor networks.** *IEEE Transactions on Vehicular Technology*, **58**(8):4554–4564, 2009.
- [78] SANG GUUN YOO, KEUN YOUNG PARK, AND JUHO KIM. **A security-performance-balanced user authentication scheme for wireless sensor networks.** *International journal of distributed sensor networks*, **8**(3):382810, 2012.
- [79] KAIPING XUE, CHANGSHA MA, PEILIN HONG, AND RONG DING. **A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks.** *Journal of Network and Computer Applications*, **36**(1):316–323, 2013.
- [80] QI JIANG, JIANFENG MA, XIANG LU, AND YOU LIANG TIAN. **An efficient two-factor user authentication scheme with unlinkability for wireless sensor networks.** *Peer-to-peer Networking and Applications*, **8**(6):1070–1081, 2015.
- [81] ASHOK KUMAR DAS. **A secure and effective biometric-based user authentication scheme for wireless sensor networks using smart card and fuzzy extractor.** *International Journal of Communication Systems*, **30**(1):e2933, 2017.
- [82] MOJTABA ALIZADEH, SAEID ABOLFAZLI, MAZDAK ZAMANI, SABARIAH BAHARUN, AND KOUICHI SAKURAI. **Authentication in mobile cloud computing: A survey.** *Journal of Network and Computer Applications*, **61**:59–80, 2016.

- [83] CHIN-CHEN CHANG AND HAI-DUONG LE. **A provably secure, efficient, and flexible authentication scheme for ad hoc wireless sensor networks.** *IEEE Transactions on Wireless Communications*, **15**(1):357–366, 2016.
- [84] ASHOK KUMAR DAS, SARU KUMARI, VANGA ODELU, XIONG LI, FAN WU, AND XINYI HUANG. **Provably secure user authentication and key agreement scheme for wireless sensor networks.** *Security and Communication Networks*, **9**(16):3670–3687, 2016.
- [85] RUHUL AMIN AND GP BISWAS. **A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks.** *Ad Hoc Networks*, **36**:58–80, 2016.
- [86] FAN WU, LILI XU, SARU KUMARI, XIONG LI, JIAN SHEN, KIM-KWANG RAYMOND CHOO, MOHAMMAD WAZID, AND ASHOK KUMAR DAS. **An efficient authentication and key agreement scheme for multi-gateway wireless sensor networks in IoT deployment.** *Journal of Network and Computer Applications*, **89**:72–85, 2017.
- [87] JANGIRALA SRINIVAS, SOURAV MUKHOPADHYAY, AND DHEERENDRA MISHRA. **Secure and efficient user authentication scheme for multi-gateway wireless sensor networks.** *Ad Hoc Networks*, **54**:147–169, 2017.
- [88] DHEERENDRA MISHRA, ASHOK KUMAR DAS, AND SOURAV MUKHOPADHYAY. **A secure user anonymity-preserving biometric-based multi-server authenticated key agreement scheme using smart cards.** *Expert Systems with Applications*, **41**(18):8129–8143, 2014.
- [89] CHIN-CHEN CHANG AND NGOC-TU NGUYEN. **An Untraceable Biometric-Based Multi-server Authenticated Key Agreement Protocol with Revocation.** *Wireless Personal Communications*, **90**(4):1695–1715, 2016.
- [90] TIEN-HO CHEN AND WEI-KUAN SHIH. **A robust mutual authentication protocol for wireless sensor networks.** *ETRI journal*, **32**(5):704–712, 2010.

- 
- [91] BINOD VAIDYA, DIMITRIOS MAKRAKIS, AND HUSSEIN MOUFTAH. **Two-factor mutual authentication with key agreement in wireless sensor networks.** *Security and Communication Networks*, **9**(2):171–183, 2016.
- [92] FAN WU, XIONG LI, ARUN KUMAR SANGAIAH, LILI XU, SARU KUMARI, LIUXI WU, AND JIAN SHEN. **A lightweight and robust two-factor authentication scheme for personalized healthcare systems using wireless medical sensor networks.** *Future Generation Computer Systems*, **82**:727–737, 2018.
- [93] QI JIANG, JIANFENG MA, FUSHAN WEI, YOU LIANG TIAN, JIAN SHEN, AND YUANYUAN YANG. **An untraceable temporal-credential-based two-factor authentication scheme using ECC for wireless sensor networks.** *Journal of Network and Computer Applications*, **76**:37–48, 2016.
- [94] JUNGHYUN NAM, JURYON PAIK, H-K KANG, UNG MO KIM, AND DONGHO WON. **An off-line dictionary attack on a simple three-party key exchange protocol.** *IEEE Communications Letters*, **13**(3):205–207, 2009.
- [95] MUHAMMAD BILAL AND SHIN-GAK KANG. **An authentication protocol for future sensor networks.** *Sensors*, **17**(5):979, 2017.
- [96] OMAR CHEIKHROUHO, ANIS KOUBAA, MANEL BOUJELBEN, AND MOHAMED ABID. **A lightweight user authentication scheme for wireless sensor networks.** In *Computer Systems and Applications (AICCSA), 2010 IEEE/ACS International Conference on*, pages 1–7. IEEE, 2010.
- [97] JUNGHYUN NAM, KIM-KWANG RAYMOND CHOO, SANGCHUL HAN, MOONSEONG KIM, JURYON PAIK, AND DONGHO WON. **Efficient and anonymous two-factor user authentication in wireless sensor networks: achieving user anonymity with lightweight sensor computation.** *Plos one*, **10**(4):e0116709, 2015.
- [98] XIONG LI, JIANWEI NIU, SARU KUMARI, FAN WU, ARUN KUMAR SANGAIAH, AND KIM-KWANG RAYMOND CHOO. **A three-factor anonymous authentication scheme for wireless sensor networks in internet of things environments.** *Journal of Network and Computer Applications*, **103**:194–204, 2018.

- [99] CARLO BLUNDO, ALFREDO DE SANTIS, AMIR HERZBERG, SHAY KUTTEN, UGO VACCARO, AND MOTI YUNG. **Perfectly secure key distribution for dynamic conferences.** *Information and Computation*, **146**(1):1–23, 1998.
- [100] FRANCIS BEGNAUD HILDEBRAND. *Introduction to numerical analysis*. Courier Corporation, 1987.
- [101] DONGGANG LIU, PENG NING, AND RONGFANG LI. **Establishing pairwise keys in distributed sensor networks.** *ACM Transactions on Information and System Security (TISSEC)*, **8**(1):41–77, 2005.
- [102] ALESSANDRA LUMINI AND LORIS NANNI. **An improved biohashing for human authentication.** *Pattern recognition*, **40**(3):1057–1065, 2007.
- [103] D EASTLAKE 3RD AND PAUL JONES. **US secure hash algorithm 1 (SHA1).** Technical report, 2001.
- [104] FILIPPO GANDINO, RENATO FERRERO, AND MAURIZIO REBAUDENGO. **A Key Distribution Scheme for Mobile Wireless Sensor Networks: q-s - Composite.** *Trans. Info. For. Sec.*, **12**(1):34–47, January 2017.
- [105] ASHOK KUMAR DAS. **A random key establishment scheme for multi-phase deployment in large-scale distributed sensor networks.** *International journal of information security*, **11**(3):189–211, 2012.
- [106] SUNG JIN CHOI, KYUNG TAE KIM, AND HEE YONG YOUN. **An energy-efficient key predistribution scheme for secure wireless sensor networks using eigenvector.** *International Journal of Distributed Sensor Networks*, **9**(6):216754, 2013.
- [107] NAN-RUN ZHOU, QIONG-XI JIANG, AND LI-HUA GONG. **Key predistribution scheme based on CRT and LU matrix for wireless sensor networks.** *Journal of Shanghai jiaotong university*, **46**(11), 2012.
- [108] DA-ZHI SUN, JIAN-XIN LI, ZHI-YONG FENG, ZHEN-FU CAO, AND GUANG-QUAN XU. **On the security and improvement of a two-factor user authentication scheme in wireless sensor networks.** *Personal and ubiquitous computing*, **17**(5):895–905, 2013.



- 
- [109] WENBO SHI AND PENG GONG. **A new user authentication protocol for wireless sensor networks using elliptic curves cryptography.** *International Journal of Distributed Sensor Networks*, **9**(4):730831, 2013.
- [110] DANNY DOLEV AND ANDREW YAO. **On the security of public key protocols.** *IEEE Transactions on information theory*, **29**(2):198–208, 1983.
- [111] **TinyOS: An Open-Source OS for the Networked Sensor Regime.** Available on. <https://github.com/tinyos/tinyos-main>, accessed on 29 July 2018.
- [112] DAVID GAY, PHILIP LEVIS, ROBERT VON BEHREN, MATT WELSH, ERIC BREWER, AND DAVID CULLER. **The nesC language: A holistic approach to networked embedded systems.** *Acm Sigplan Notices*, **49**(4):41–51, 2014.
- [113] PRERNA MOHIT, RUHUL AMIN, ARIJIT KARATI, GP BISWAS, AND MUHAMMAD KHURRAM KHAN. **A standard mutual authentication protocol for cloud computing based health care system.** *Journal of medical systems*, **41**(4):50, 2017.
- [114] SHIN-YAN CHIOU, ZHAOQIN YING, AND JUNQIANG LIU. **Improvement of a privacy authentication scheme based on cloud for medical environment.** *Journal of medical systems*, **40**(4):101, 2016.
- [115] JONGDEOG LEE, KRASIMIRA KAPITANOVA, AND SANG H SON. **The price of security in wireless sensor networks.** *Computer Networks*, **54**(17):2967–2978, 2010.
- [116] ZHE LIU, ERICH WENGER, AND JOHANN GROSSSCHÄDL. **MoTE-ECC: Energy-scalable elliptic curve cryptography for wireless sensor networks.** In *International Conference on Applied Cryptography and Network Security*, pages 361–379. Springer, 2014.
- [117] SUNG JIN CHOI AND HEE YONG YOUN. **An efficient key pre-distribution scheme for secure distributed sensor networks.** In *International Conference on Embedded and Ubiquitous Computing*, pages 1088–1097. Springer, 2005.

## REFERENCES

---

- [118] AL-SAKIB KHAN PATHAN, TRAN THANH DAI, AND CHOONG SEON HONG. **An efficient LU decomposition-based key pre-distribution scheme for ensuring security in wireless sensor networks.** In *null*, page 227. IEEE, 2006.
- [119] ARVINDERPAL S WANDER, NILS GURA, HANS EBERLE, VIPUL GUPTA, AND SHEUELING CHANG SHANTZ. **Energy analysis of public-key cryptography for wireless sensor networks.** In *Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on*, pages 324–328. IEEE, 2005.



# List of Publications

- [1] Maurya, A.K.; Sastry, V.N.. **Fuzzy Extractor and Elliptic Curve Based Efficient User Authentication Protocol for Wireless Sensor Networks and Internet of Things.** *Information (MDPI)*, 2017, Vol. 8, 136. pp. 1-38. (Indexed by: E-SCI, DBLP, Scopus, Compendex (Elsevier), Inspec (IET), Web of Science.)
- [2] Maurya, A.K.; Sastry, V.N.; Udgata, S.K.. **Cryptanalysis and Improvement of ECC -based Security Enhanced User Authentication Protocol for Wireless Sensor Networks.** *Third International Symposium on Security in Computing and Communications*, August 10-13, 2015, Kochi, India. pp. 134-145. (Indexed by: DBLP, SCOPUS)
- [3] Maurya, A.K.; Sastry, V.N.. **Secure and Efficient Authenticated Key Exchange Mechanism for Wireless Sensor Networks and Internet of Things using Bloom Filter.** *The 3rd IEEE International Conference on Collaboration and Internet Computing*, Oct 15 - 17, 2017, San Jose, California, USA. pp.173-180. (Indexed by: IEEE Xplore Digital Library).
- [4] Maurya, A.K.; Sastry, V.N.. **User Authentication Scheme for Wireless Sensor Networks and Internet of Things using Chinese Remainder Theorem.** *Fifth International Symposium on Security in Computing and Communications*, September 13-16, 2017, Manipal, India. pp. 79-94. (Indexed by: DBLP, SCOPUS).  
**Best Paper Award**
- [5] Maurya, A.K.; Sastry, V.N.. **User Authentication Scheme for Wireless Sensor Networks and Internet of Things using LU Decomposition.** *Fifth International Symposium on Security in Computing and Communications*, September 13-16, 2017, Manipal, India. pp. 39-53. (Indexed by: DBLP, SCOPUS).

## LIST OF PUBLICATIONS

---

- [6] Maurya, A.K.; Sastry, V.N.. **Symmetric Hash Function based Secure and Efficient Authenticated Key Exchange Mechanism for Wireless Sensor Networks.** *11th IEEE International Conference on Advanced Networks and Telecommunications Systems*. December 17-20, 2017, Bhubaneswar, Odisha, India. pp. 1-6, (Indexed by: IEEE Xplore Digital Library).
  
- [7] Maurya, A.K.; Sastry, V.N.. **A Survey on User Authentication and Key Establishment Protocols for Wireless Sensor Networks and Internet of Things with Cryptanalysis.** *Future Generation Computer Systems*, 2018, (Communicated).

# Appendix A

## Annexure (Publications Online)

**Table A.1:** Fact sheet of SSCC-2015 Publication [2]

Title	Cryptanalysis and Improvement of ECC - Based Security Enhanced User Authentication Protocol for Wireless Sensor Networks
Authors	Maurya, A.K.; Sastry, V.N. and Udgata, S.K.
Publication	Third International Symposium on Security in Computing and Communications, August 10-13, 2015, Kochi, Kerala, India. pp. 134-145.
ISBN/ISSN	978-3-319-22914-0
DOI	10.1007/978-3-319-22915-7_13
Status	Published
Publisher	Springer, Singapore
Publication Type	Conference Proceedings

The screenshot shows the SpringerLink interface for a conference paper. The browser address bar displays the URL: [https://link.springer.com/chapter/10.1007/978-3-319-22915-7\\_13](https://link.springer.com/chapter/10.1007/978-3-319-22915-7_13). The page title is "Cryptanalysis and Improvement of ECC - Based Security Enhanced User Authentication Protocol for Wireless Sensor Networks". The authors listed are Anup Kumar Maurya, V. N. Sastry, and Siba Kumar Udgata. It is identified as a "Conference paper" first online on "08 August 2015". The page has 827 downloads. The abstract states: "User authentication and secret session key exchange between a user and a sensor node are important security requirements of wireless sensor networks for retrieving the important,". On the right side, there are options to "Buy eBook" for EUR 67.82 and "Buy paper (PDF)" for EUR 24.95. A sidebar on the right also lists benefits like "Instant download", "Readable on all devices", and "Own it forever".

**Figure A.1:** Screenshot of SSCC-2015 Publication [2]

**Table A.2:** Fact sheet of Information-2017, MDPI Publication [1]

Title	Fuzzy Extractor and Elliptic Curve Based Efficient User Authentication Protocol for Wireless Sensor Networks and Internet of Things
Authors	Maurya, A.K.; Sastry, V.N.
Publication	Information , 2017, 8(4).
ISBN/ISSN	2048-8386
DOI	<a href="https://doi.org/10.3390/info8040136">https://doi.org/10.3390/info8040136</a>
Status	Published
Publisher	MDPI
Publication Type	Journal

The screenshot displays the MDPI Information journal website. The top navigation bar includes links for MDPI, Journals A-Z, Information & Guidelines, Initiatives, and About. The main header features the journal's logo and a search bar with fields for Title/Keyword, Author/Affiliation, Journal, and Article Type. The article being viewed is from Volume 8, Issue 4, published in 2017, 8(4), page 136. The article title is 'Fuzzy Extractor and Elliptic Curve Based Efficient User Authentication Protocol for Wireless Sensor Networks and Internet of Things' by Anup Kumar Maurya and V. N. Sastry. The article has 1617 views, 714 downloads, and 1 citation. The abstract states: 'To improve the quality of service and reduce the possibility of security attacks, a secure and efficient user authentication mechanism is required for Wireless Sensor Networks (WSNs) and the Internet of Things (IoT). Session key establishment between the sensor node and the user is also required for secure communication.' The article is available in full-text, PDF, and figures formats.

**Figure A.2:** Screenshot of Information-2017, MDPI Publication [1]

**Table A.3:** Fact sheet of CIC-2017 Publication [3]

Title	Secure and Efficient Authenticated Key Exchange Mechanism for Wireless Sensor Networks and Internet of Things using Bloom Filter
Authors	Maurya, A.K.; Sastry, V.N.
Publication	The 3rd IEEE International Conference on Collaboration and Internet Computing, Oct 15 - 17, 2017, San Jose, California, USA. pp.173-180.
ISBN/ISSN	978-1-5386-2565-1
DOI	10.1109/CIC.2017.00032
Status	Published
Publisher	IEEE
Publication Type	Conference Proceeding

The screenshot shows the IEEE Xplore digital library interface. At the top, the browser address bar displays the URL <https://ieeexplore.ieee.org/document/8181493/>. The page title is "Secure and Efficient Authenticated Key Exchange Mechanism for Wireless Sensor Networks and Internet of Things Using Bloom Filter". Below the title, there is a "Sign In or Purchase to View Full Text" button and a "72 Full Text Views" indicator. The authors are listed as "2 Author(s): Anup Kumar Maurya; V.N. Sastry". A navigation bar includes tabs for "Abstract", "Authors", "Figures", "References", "Citations", "Keywords", "Metrics", and "Media". The "Abstract" tab is selected, showing the following text: "For security sensitive Internet of Things (IoT) and Wireless Sensor Networks(WSNs), an efficient authenticated key exchange (AKE) mechanism improves the quality of services and reduces the possibility of security attacks. In this paper, we propose Bloom filter based authentication scheme suitable for the WSNs and IoT of large hop count (i.e., the large number of intermediate sensor nodes through which data must pass between source sensor node and gateway node). The novelty of our proposed protocol is, it eliminates unauthorized querying message transmission at the initial level (i.e., at the sensor node itself) to avoid bogus message flooding from the sensor nodes to the gateway node (which exhausts the resources of WSNs). We perform the formal and informal security analysis of the proposed schemes using widely accepted AVISPA tool and random oracle model. The computational analysis shows that our system is suitable for resource constrained sensor nodes like TelosB and MicaZ. The relative security and performance analysis results indicate that our scheme is more secure, efficient and robust in comparison to other existing systems." Below the abstract, it states "Published in: 2017 IEEE 3rd International Conference on Collaboration and Internet Computing (CIC)". At the bottom, there is a table with publication details: "Date of Conference: 15-17 Oct. 2017", "INSPEC Accession Number: 17431165", "Date Added to IEEE Xplore: 14 December 2017", "DOI: 10.1109/CIC.2017.00032", "Publisher: IEEE", and "Conference Location: San Jose, CA, USA".

**Figure A.3:** Screenshot of CIC-2017 Publication [3]



**Table A.4:** Fact sheet of SSCC-2017 Publication [4]

Title	User Authentication Scheme for Wireless Sensor Networks and Internet of Things using Chinese Remainder Theorem.
Authors	Maurya, A.K.; Sastry, V.N.
Publication	Fifth International Symposium on Security in Computing and Communications, September 13-16, 2017, Manipal, India. pp. 79-94.
ISBN/ISSN	978-981-10-6898-0
DOI	10.1007/978-981-10-6898-0_7
Status	Published
Publisher	Springer, Singapore
Publication Type	Conference Proceeding

The screenshot displays the SpringerLink interface for a conference paper. At the top, the browser address bar shows the URL: [https://link.springer.com/chapter/10.1007/978-981-10-6898-0\\_7](https://link.springer.com/chapter/10.1007/978-981-10-6898-0_7). The page header includes the SpringerLink logo and a search bar. The main content area features a book cover thumbnail on the left and the title 'User Authentication Scheme for Wireless Sensor Networks and Internet of Things Using Chinese Remainder Theorem' on the right. Below the title, the authors 'Anup Kumar Maurya' and 'V. N. Sastry' are listed. The page also indicates it is a 'Conference paper' and 'First Online: 10 November 2017'. A statistics section shows '1' reader and '420' downloads. The abstract section begins with the text: 'Authenticated querying is one of the prominent requirement of Internet of Things (IoT) or wireless networks of sensor devices to resist unauthorized users from accessing real time and confidential data. In this paper, we perform security analysis and find drawbacks of Das's user authentication scheme (proposed in 2015). We propose an efficient authenticated key exchange'.

**Figure A.4:** Screenshot of SSCC-2017 Publication [4]

**Table A.5:** Fact sheet of SSCC-2017 Publication [5]

Title	User Authentication Scheme for Wireless Sensor Networks and Internet of Things using LU Decomposition
Authors	Maurya, A.K.; Sastry, V.N.
Publication	Fifth International Symposium on Security in Computing and Communications, September 13-16, 2017, Manipal, India. pp. 39-53
ISBN/ISSN	ISBN 978-981-10-6898-0
DOI	10.1007/978-981-10-6898-0_4
Status	Published
Publisher	Springer, Singapore
Publication Type	Conference Proceeding

The screenshot shows a web browser window with the URL [https://link.springer.com/chapter/10.1007/978-981-10-6898-0\\_4](https://link.springer.com/chapter/10.1007/978-981-10-6898-0_4). The page is titled "User Authentication Scheme for Wireless Sensor Networks and Internet of Things Using LU Decomposition" and is part of the "International Symposium on Security in Computing and Communication" (SSCC 2017). The authors listed are Anup Kumar Maurya and V. N. Sastry. The page indicates it is a conference paper, first online on 10 November 2017, and has 2 readers and 409 downloads. The abstract is visible, starting with "In security-sensitive wireless networks of sensor devices, the authenticity of the legitimate user is the prominent requirement. Because of constraints-resources of these sensor devices implementing traditional cryptographic mechanism is not an easy task. In this paper, we propose a lightweight mechanism for authenticating users of a sensor network using fuzzy extractor along".

https://link.springer.com/chapter/10.1007/978-981-10-6898-0\_4

User Authentication Scheme...

Springer Link

International Symposium on Security in Computing and Communication  
SSCC 2017: Security in Computing and Communications pp 39-53 | Cite as

**User Authentication Scheme for Wireless Sensor Networks and Internet of Things Using LU Decomposition**

Authors [Authors and affiliations](#)

Anup Kumar Maurya, V. N. Sastry

Conference paper  
**First Online:** 10 November 2017

2 Readers 409 Downloads

Part of the [Communications in Computer and Information Science](#) book series (CCIS, volume 746)

**Abstract**

In security-sensitive wireless networks of sensor devices, the authenticity of the legitimate user is the prominent requirement. Because of constraints-resources of these sensor devices implementing traditional cryptographic mechanism is not an easy task. In this paper, we propose a lightweight mechanism for authenticating users of a sensor network using fuzzy extractor along

**Figure A.5:** Screenshot of SSCC-2017 Publication [5]

**Table A.6:** Fact sheet of ANTS-2017 Publication [6]

Title	Symmetric Hash Function based Secure and Efficient Authenticated Key Exchange Mechanism for Wireless Sensor Networks
Authors	Maurya, A.K.; Sastry, V.N.
Publication	Proceedings of 11th IEEE International Conference on Advanced Networks and Telecommunication Systems. December 17-20, 2017, Bhubaneswar, India.
ISBN/ISSN	ISBN 978-1-5386-2347-3
DOI	10.1109/ANTS.2017.8384158
Status	Published
Publisher	IEEE
Publication Type	Conference Proceeding

The screenshot shows the IEEE Xplore abstract page for the paper "Symmetric hash function based secure and efficient authenticated key exchange mechanism for wireless sensor networks" by Anup Kumar Maurya and V.N. Sastry. The page includes the title, authors, publication details, and an abstract. The abstract describes a proposed symmetric hash function and bloom filter based secure authenticated key exchange (AKE) protocol for WSNs. It mentions that the protocol is appropriate for authentication of users without pre-alignment between the test and the registered minutia points of the fingerprint for multi-hop WSNs. The proposed protocol eliminates unauthorized query information circulation at the initial level (i.e., at the sensor node itself) to prevent bogus information flooding from the sensor nodes to the gateway node. It also states that the proposed protocol resists the resource exhaustion attacks associated with WSNs of large hop count (i.e., a large number of intermediary sensor nodes through which information must pass between source sensor node and the trusted gateway node). The analysis of computational overhead demonstrates that the proposed protocol is preferable for resource-constrained sensor nodes like MicaZ. The security analysis and performance evaluation reveal that the proposed protocol is more secure, effective and resilient in comparison to other existing protocols.

**Published in:** 2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)

**Date of Conference:** 17-20 Dec. 2017  
**Date Added to IEEE Xplore:** 14 June 2018

**INSPEC Accession Number:** 17856669  
**DOI:** 10.1109/ANTS.2017.8384158  
**Publisher:** IEEE  
**Conference Location:** Bhubaneswar, India

**Figure A.6:** Screenshot of ANTS-2017 Publication [6]