UNLINKABLE USER AUTHENTICATED KEY AGREEMENT FOR MULTI-GATEWAY WIRELESS SENSOR NETWORKS

MSc (MATHEMATICAL SCIENCE) THESIS

SOLOMON KUONGA NKHOMA

UNIVERSITY OF MALAWI

FEBRUARY, 2023

UNLINKABLE USER AUTHENTICATED KEY AGREEMENT FOR MULTI-GATEWAY WIRELESS SENSOR NETWORKS

MSc (MATHEMATICAL SCIENCE) THESIS

By

SOLOMON KUONGA NKHOMA
B.Ed (Science)-University of Malawi

Submitted to the Department of Mathematical Sciences, Faculty of Science, in partial fulfilment of the requirements for the degree of Master of Science (Mathematical Science)

University of Malawi

February, 2023

DECLARATION

I the undersigned hereby declare that this thesis/dissertation is my own original work which has not been submitted to any other institution for similar purposes. Where other people's work has been used acknowledgements have been made.

SOLOMON KUONGA NKHOMA

Signature

Aug 31 2023

Date

CERTIFICATE OF APPROVAL

The undersigned certify that this thesis represents the student's own work and effort and has been submitted with our approval.

Signature:_____Date: Aug 30 2023

HYUN SUNG KIM,PhD (Visiting Professor)
Main Supervisor

Signature:_____Date:_____
NELSON DZUPIRE,PhD (Senior Lecturer)
Head of Department

DEDICATION

To my late father Mr. Solomon KuongaNkhoma, who died on 12th October 2011.He was the father everyone would want to have and was a good adviser with so much love. It is sad that he shall not eat the fruits of his labour. When I go to see mother in the country side, I always remember him when sitting at the Khonde telling me about his love and vision for my life. May his soul, rest in eternal peace.

To my lovely wife, Wezzie Nkhoma and my children Rejoice and Calvin Nkhoma, they endured many days and nights of neglect just for the sake of this work. I dedicate it to them with humble heart, love and cherish.

ACKNOWLEDGEMENTS

Firstly, I thank God the Father Almighty who provided light in times of darkness and thick clouds, He gave me hope when I had doubts, He also gave me strength, power and patience to complete this study. I exalt His name and adore Him.

As a continuation, I am indebted to express my sincere thanks giving to my supervisor Professor Hyunsung Kim for the support, guidance and words of encouragement. He always provided to make this research project a successful one and for his constructive criticisms, his willingness and readiness to be consulted. When I started doubting about my success in this research, he supported me and gave me all kinds of encouragements. All kinds of mistakes I made he did not look at them as my downfall but he looked at them and finds the best way he can turn them into positive results.

The Head of Department Dr. Nelson Dzupire for the guidance and words of encouragement he has been providing to me in the process of writing the thesis. When I felt to stop in the way because of fees he came on my way and gave me direction on what to do for my stay.

My Wife, (Wezzie Mkupu) and my children (Rejoice and Calvin Nkhoma) for their support and understanding. This project demanded some weeks and even months away from home. My family never got tired over me and gave me all the support and encouragement I need.

My Mother, Mrs. S.K. Nkhoma, Brothers (Chiliritso, Hanannia Michael and Morgan Nkhoma), Sisters (Lucy, Msiyana, and Lettina Nkhoma), distant relatives, friends and colleagues (and many more difficult to exhaust them here) for their tireless support and encouragement they have given me throughout the process.

My Brother Mr. CK. Nkhoma and his Wife Esnatti Nkhoma (Amayia Rabecca) for understanding when I could not spend time with them doing business because of this dissertation. Paying my school fees from primary to university is a challenging thing but he has been doing that despite all the untruthful part of me. He and his wife have been so wonderful to me. May God bless them.

I am also grateful to the Ministry of Education and Mathematical Department (University of Malawi Chancellor College), for the research grant they provided to me. This support came in, timely when I needed it most for the research to be completed.

I am also grateful to Kari Murphy and Patricia Eddiman for their financial support. When the road seemed to come to an end they came in willingly and timely rescued me. My brother, my wife and I we came to resolution that I first withdraw myself from the university because of school fees since I had a huge amount of fees balance but when they received this massage they contributed to my tuition fee a time I was in need of it, I cherish them. May God bless you all the time of your life.

ABSTRACT

Wireless sensor network (WSN) has wide potential application in various fields such as military, agricultural and healthcare. WSNs need effective security mechanisms because they are deployed in hostile unattended environments. Various user authentication protocols were proposed for WSNs security. However, there are many previous protocols that have various security vulnerabilities including masquerading, password guessing attack and traceability and they do not provide anonymity and unlinkability. Especially, unlinkability is one of important privacy factor in WSNs environment that an attacker cannot adequately distinguish whether the elements are related or not. This paper proposes an unlinkable user authenticated key agreement protocol (UAKA) for multi-gateway WSNs that could achieve desirable security and privacy attributes. The security of UAKA is based on the one-wayness of hash function and secrecy of symmetric key cryptosystem. UAKA supports dynamic node addition and user friendly password change. The security and privacy of UAKA was proven based on BAN logic and informal security analysis. It preserves all the original merits of the related protocols and provides security and privacy. However, UAKA has a bit computational overhead compared to the related protocols due to providing security and privacy. The Analysis results shows that all the related protocols are linkable while UAKA is anonymous and unlinkable and it also provides enough security and privacy to all active and passive attacks

TABLE OF CONTENTS

ABSTRACT vii
LIST OF TABLES x
LIST OF FIGURES xi
NOTATIONS AND SYMBOLS xii
CHAPTER 1 1
INTRODUCTION 1
CHAPTER 2 6
LITERATURE REVIEW 6
2.1 WSN model 6
2.2 Security preliminaries 8
2.2.1 Threat model 8
2.2.2 Security building block 9
2.3 Related works 10
CHAPTER 3 14
UNLINKABLE USER AUTHENTICATED KEY AGREEMENT FOR MULTI-GATEWAY WSNS 14
3.1 System setup phase 15
3.2 Registration phase 15
3.2.1 SN registration phase 15
3.2.2 User registration phase 18
3.3 Login and authenticated key agreement phase 18
3.3.1 Login phase 19
3.3.2 Authenticated key agreement phase 20
3.4 Dynamic node addition phase 23
3.5 Password change phase 23
CHAPTER 4 24
ANALYSIS 24
4.1 Security analysis 24
4.1.1 BAN logic analysis 25
4.1.2 Informal security analysis 31
4.2 Performance analysis 37
CHAPTER 5 41

CONCLUSION 41
BIBILIOGRAPHY 43

LIST OF TABLES

Table 1: Comparison of security features 37
Table 2: Comparison of computational overhead at login and authenticated key agreement 39
Table 3: Comparison of communicational overhead at login and authenticated key agreement 40

LIST OF FIGURES

Figure 1: Multi-GWN based WSN model 7
Figure 2: Sensor node registration phase 16
Figure 3: User registration phase 17
Figure 4: Login phase and authenticated key agreement phase..... 22

NOTATIONS AND SYMBOLS

Notations	Description
$\|\|$	Concatenation of data
$\oplus$	Exclusive-OR of data
GWN	Gateway node (base station) in WSNs
$h(\cdot)$	Secure collision-free cryptographic one-way hash function
$E(\cdot)$	Encryption based on AES
$D(\cdot)$	Decryption based on AES
ID_i	Identity of user U_i
ID_{HGWN}	Identity of home gateway node *HGWN*
ID_{FGWN}	Identity of foreign gateway node *FGWN*
ID_{SNj}	Identity of sensor node SN_j
PW_i	Password of user U_i
r_i, r_j, r_h, r_f	Random nonces used by U_i , SN_j, *HGWN* and *FGWN*, respectively
SC	Smartcard of user U_i
SK	Session key shared
SN_j	j-th sensor node
T_c	Time when a message received by an entity
ΔT	Time interval for the allowed transmission delay
U_i	i-th remote user
X_{HGWN}	Master secret key of *HGWN*

ABBREVIATIONS

AES	Advanced Encryption Standard
DES	Data Encryption Standard
FGWN	Foreign Gateway Node
GWN	Gateway Node
HGWN	Home Gateway Node
ID	Login Identity
SC	Smart Card
SK	Session Key
SN	Sensor Node
U_i	User *i*
UAKA	User Authenticated Key Agreement

CHAPTER 1

INTRODUCTION

Wireless sensor network (WSN) is an emerging technology which consists of hundreds or thousands of small devices. Recently, WSN has been widely researched. WSN does not need much infrastructure to operate in and it is beneficial in environments and infrastructures where wires are not suitable. They also provide cheap solutions to real world problems (Huang et al., 2019; Al-Mousawi & Al-Hassani, 2018; Vatsala et al., 2017). Each device has ability of sensing, processing and communication capabilities over a wireless channel to monitor the real-world environment. Sensor nodes (*SN*s) are wirelessly interconnected with each other and with the gateway node (*GWN*) (Lakshmanan, 2009). Due to their numerous advantages, WSN is applied in various fields such as military, environmental applications detection of forest fires, industrial control, environmental monitoring, health care monitoring, smart building, facility management, intelligent agriculture, earthquake and weather forecast, target tracking and military security (Liu et al., 2012). *SN*s in the generalized WSN capture data of interest and report it to a single *GWN*. We observed that in a single *GWN*, WSN data traffic is concentrated to the *GWN*. Thus, single *GWN* can result into a cause of congestion and this can decrease reliability and increase latency. On the other hand, multi-*GWN* WSN not only helps to reduce hot spots but also provides significant capacity benefits.

Thereby, multi-*GWN* can help to increase reliability and reduce latency of WSNs (Omer et al., 2017). So, this thesis will only consider the multi-*GWN*WSN environment.

Despite that WSN has various benefits, it also generates new security threats (Gandotra and Jha, 2017; Pietro et al., 2014).WSN has various security issues due to its design, storage and energy limitations and *SN*s and *GWN*s are sometimes deployed in unattended environments (Vatsala, 2017). The attackers would use the security flaws making the network vulnerable to various types of attacks. So, security is one of the fundamental requirements for any network. The major security goals will always remain the same as with traditional networks, which include confidentiality, data integrity, authentication, key agreement and availability (Asimi et al., 2018; Yousefpoor & Barati, 2019). Confidentiality can be ensured through encryption. The message communicated through the network must remain confidential. Any *SN* must not disclose its data to the neighboring *SN*s. It is very important because *SN*s may carry same sensitive data. Integrity should ensure the reliability of data and should confirm that the data has not been tampered with. There can be a loss of integrity if there is a loss or damage of data in the WSN. The WSN must be available to communicate messages and should be able to use the resources (Jadhav & Vatsala, 2017).

In addition to security concerns, privacy in the context of WSNs involves both privacy of monitored subjects and privacy of nodes and *GWN*. Privacy of these

parties is usually bound together to some extent (Singh et al., 2016; Debnath et al., 2014). Since breach of node privacy can lead to violation of the monitored subject privacy and vice versa. Privacy in WSNs can be considered for anonymity and unlinkability in this thesis. Anonymity typically refers to the state in which an individual's personal identity or personally identifiable information is not known publicly. The unlinkability of two or more items of interest, from an attacker's perspective, means that within the system, the attacker cannot identify whether these items are related. A number of researchers are presenting their researches on ensuring the security and privacy goals (Gao et al., 2018; Singh et al., 2016; Al-Janabi et al., 2017; Lee & Kim, 2014; Finogeev & Finogeev, 2017).The major challenge in WSN is that the sensed data should be transmitted via public networks. The adversary has capability to intercept the communication over a public network. This makes the WSNs environment vulnerable to attacks. Therefore, a communication protocol should achieve the mutual authentication and support key agreement while providing unlinkability between or among parties. Some researchers have presented their researches on security of WSNs (Khan & Alghathbar, 2010; Vaidya et al., 2010; Deebak, 2016; Das et al., 2012;Turkanovic et al., 2014; Farash et al., 2016; Amin & Biswas, 2016;Srinivas et al., 2017). However, it was observed that all these protocols are weak against known active and passive attacks and they also fail to provide anonymity and unlinkability. Thus the design of UAKA is to solve these security concerns.

The authentication and key agreement protocols usually try to achieve the goal of computational efficiency and security attributes. Lightweight authentication protocols for example involve non-public key parameter, while the generalized network environment design uses public key parameter. Researchers have shown that non-public key parameters based protocols presents computationally efficient solution for low-capacity device, this results also making these designs low-cost. However these designs have many limitations, for example they are inefficient to provide unlinkabiliy.

The purpose of this thesis is to propose an unlinkable user authenticated key agreement (UAKA) for multi-gateway WSNs that could achieve desirable security and privacy attributes. The security of UAKA is based on the one-wayness of hash function and secrecy of symmetric key cryptosystem. UAKA supports dynamic node addition and user friendly password change. The security and privacy of UAKA was proven based on BAN logic and informal security analysis. It preserves all the original merits of the related protocols and provides security and privacy, which are unlinkability and anonymity, *GWN* and *SN* masquerading attacks, replay attack, trace attack, insider attack, password guessing attack and denial of service attack. However, UAKA has a bit computational overhead compared to the related protocols due to providing security and privacy.

The road-map of this thesis is organized as follows: In chapter 2, we present literature review, which are required for the better understanding on this thesis

context. We also present the network model, security building blocks. Chapter 3 proposed an unlikable user authenticated key agreement over multi-*GWN*WSN. Security analysis and performance analysis are given at chapter 4 with the proper comparisons with the related protocols. Finally, chapter 5 concludes this thesis.

CHAPTER 2

LITERATURE REVIEW

This section provides preliminaries for the targeting WSN environment, security primitives and related work reviews. They could provide basic information to understand UAKA.

2.1 WSN model

Fig. 1shows our target WSN model. The model consists of three types of entities, SNs, GWNs and users. Their roles are defined as follows

- SNs: They are responsible for sensing the real-time data and forward them to the nearest GWN node directly.
- GWNs: They are responsible for receiving and forwarding the relevant data to the user and sensor node. Furthermore, they keep a database of sensor nodes to be related among GWNs.
- Users: They can access the sensed data of the sensor node through GWN after performing mutual authentication and key agreement.

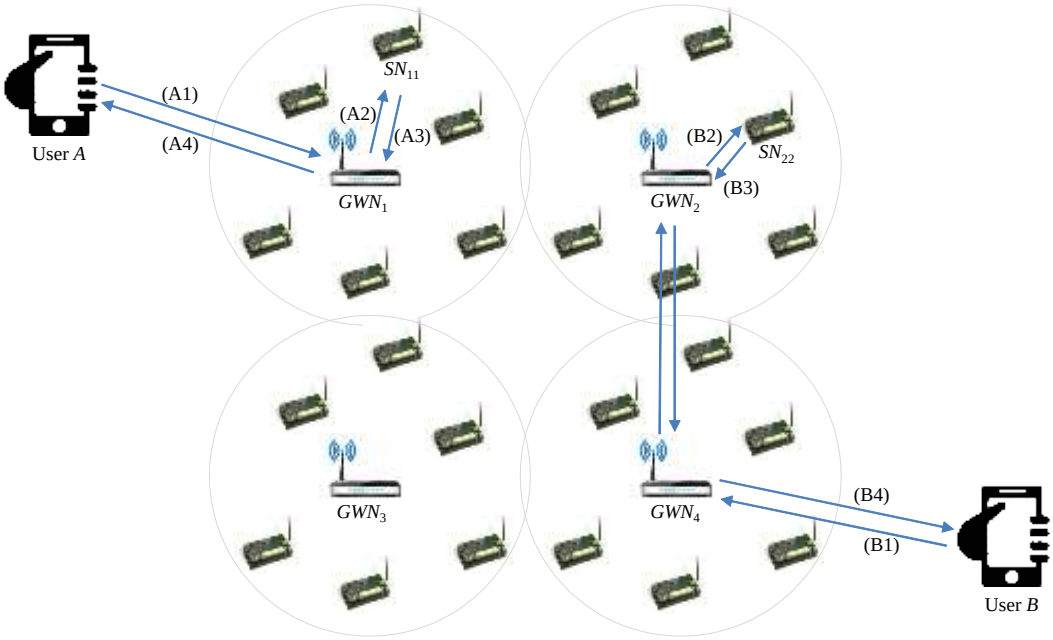

Figure 1: Multi-GWN based WSN model

GWN and sensor nodes are stationary after deployment, which is shown in Fig. 1. As mentioned in (Gao et al., 2018; Gandotra and Jha, 2017; He et al., 2010), the receiver end can measure the distance based on the received signal strength. Therefore, it is our valid assumption that all the deployed sensor nodes execute registration phase to the nearest *GWN*. In order to access the desired sensor node, the user can execute registration phase to any one of *GWNs* of our WSN model. While a user completes the registration procedure to any one of *GWNs*, called as home *GWN* (*HGWN*) and rest of the others are foreign *GWNs* (*FGWN*) with respect to that user. It is our effortless contribution that the user can access all GWNs of WSNs, although he (or she) has performed registration to only one *HGWN*. There are two scenarios in Fig. 1, which are for users *A* and *B*. The first case is for the situation when he (or she) wants to access sensor node in *HGWN*. *A* can communicate with

GWN_1 as his (or her) *HGWN* to access data from SN_{11}. However, if the *GWN* could not find the target sensor node in its own database, it checks the sensor node and *GWN* database and forwards the request to the target *GWN* as in case *B*. It is recommendable that the user cannot directly access the desired sensor nodes but only via *GWNs.*

2.2 Security preliminaries

2.2.1 Threat model

In this threat model, we discuss some widely accepted valid assumptions regarding user authenticated key agreement protocol. We use the threat model of Dolev and Yao, which includes the capabilities of attackers (Dolev & Yao, 1983).

- An attacker can extract the information from the smart card by examining the power consumption or leaked information (Kocher et al., 1999; Messerges et al., 2002).
- An attacker has ability to eavesdrop all the communications between the parties in WSN over a public channel.
- An attacker has the potential to modify, delete, redirect and resent the eavesdropped transmitted messages.
- An attacker can be a legal user or an outsider in any system.
- An attacker can guess low entropy password and identity individually easily but guessing two secret parameters at the same time are computationally infeasible in polynomial time.
- Practically, it is assumed that the protocol used in the authentication system is known to the attacker.
- Kerckhoffs's principle: A cryptosystem should be secure even if everything about the system, except the key, is public knowledge (Kerckoffs, 1883).

2.2.2 Security building block

This subsection describes a hash function and a symmetric key cryptography which are the security basis of the proposing protocol.

[Hash function] One-way hash function maps an arbitrary length input to a fixed size output. A secure one-way hash function can be symbolized as $h(X)\rightarrow Y$, where $X=\{0, 1\}^*$ and $Y=\{0,1\}^n$. X is a binary string of arbitrary length and Y is a binary string of fixed length n (Schneier, 1996). It is used in many cryptographic applications such as digital signature, random sequence generator in key agreement, authentication and authenticated key agreement. One-way hash function should satisfy the following properties:

- Easiness: Given $s \in X$, it can be easily compute y where $y = h(s)$.
- Pre image resistant*:* It is hard to find s from given y, where $y = h(s)$.
- Second pre image resistant: It is hard to find input $s' \in X$ such that $h(s)=h(s')$ for given input $s \in X$ and $s' \neq s$.
- Collision resistant: It is hard to find a pair $(s, s') \in X \times X$ such that $h(s)=h(s')$, where $s \neq s'$.
- Mixing transformation*:* On any input $s \in X$, the hashed value $y=h(s)$ is computationally indistinguishable from a uniform binary string in the interval $\{0,2^n\}$,where n is the output length of hash $h(\cdot)$.

There are various hash functions. SHA-1is one of famous hash function, which is used for the varieties of systems security and privacy. We useSHA-1 with 160-bit hash value because it is most often used to verify a file has been unaltered.

[Symmetric key cryptography] A symmetric key cryptography is the use of only a key, for both in the encryption and decryption of data (Schneier, 1996). It is valuable because of the following three reasons (IBM, 2019).

- It is relatively inexpensive to produce a strong key for the cryptosystem.
- The key tends to be much smaller for the level of protection it affords.
- The algorithm is relatively inexpensive to process.

There are some symmetric key cryptography including Blowfish, Two fish, data encryption standard (DES), and advanced encryption standard (AES). AES is the current standard and has variable key length of 128, 192, or 256 bits. We will consider using AES with a 128-bit key for our symmetric key cryptography. AES with 128-bit key is the most widely used cryptography in applications. Hence, we have adopted it. It uses 10 transformation rounds to convert plaintext into cipher text and is approved by the National Security Agency unlike other AESs and other symmetric key cryptography.

2.3 Related works

This section provides review of related works focused on authenticated key agreement over WSNs. Wong et al. proposed a user authentication protocol for WSNs in 2006(Wong et al., 2006). Wong et al.'s protocol is a lightweight architecture, which requires only the

computation of hash functions. However, it was later proved to be vulnerable to stolen verifier attack and many logged in users with the same login identifier (ID) attack. Das improved the security of Wong et al.'s protocol (Das, 2009). Das proposed an efficient password based user authentication, which uses the temporal credentials for verification. Das's protocol is also shown to be vulnerable to denial-of-service (DoS) attack and node capture attack. Later on, Nyang and Lee and He et al. proposed some improvements of Das's protocol (Nyangand Lee, 2009; He et al., 2010).But the presented protocols failed to overcome the security flaws found in Das's protocol. In 2010, Khan and Alghathbar presented an improvement in Das's protocol (Khan & Alghathbar, 2010).They solved the problem of mutual authentication and unsecured password by introducing pre-shared keys and masked passwords. Vaidya et al. identified the security pitfalls in Khan and Alghathbar's protocol (Vaidya et al., 2010).To overcome these security pitfalls, Vaidya et al. proposed an improved version of Khan and Alghathbar's protocol. In 2010, Chen and Shih also proposed an improvement of Das's protocol (Chen & Shih, 2010).Their protocol ensures the mutual authentication among all the involved parties. However, their protocol does not resist replay attack and forgery attack. Das et al. and Xue et al. proposed authentication and key agreement protocols for WSNs using *SC*, independently (Das et al., 2012; Xue et al., 2012). They outlined the protocols to support user to viably and securely connect to the nodes of a WSNs. Both the protocols assure several security components like password protection, key agreement, mutual authentication, and resilience against several attacks. In addition, their protocols have a dynamic node addition phase. Both protocols use the hash and XOR reckonings, and are in this way lightweight and exceptionally suitable for WSNs. In2012, Vaidya et al. presented that the protocols,

showing that Das's, Khan et al. and Chen and Shih are not secured for the attacks, like stolen smartcard, node capture and *SN* impersonation (Vaidyaet al., 2012).Thus, Vaidya et al. presented a two factor user authentication protocol to prevent most of the potential attacks and to provide mutual authentication and session key establishment to the user. Deebak identified that Vaidya et al.'s protocol is vulnerable to stolen smartcard, *GWN* bypassing and *SN* key impersonation (Deebak, 2016). Xu and Wang and Turkanovic et al., showed that Das et al.'s design has flaws and is infeasible for executions, independently (Xu & Wang, 2013; Turkanovic et al., 2014).They have proposed enhanced version of Das et al.'s protocol. Similar to Das et al.'s protocol, it was demonstrated that Xu et al.'s protocol has also security pitfalls, which were presented and corrected by Li et al. and Turkanovic and Holbl Yousef poor and Barati. In spite of the fact that Das et al.'s protocol was produced for hierarchical WSNs and the key agreement executes among user, cluster head and base station (*BS*), and Xue et al.'s key agreement executes between user, *GWN* and *SN*, both utilize the same authentication model. Xue et al. argue that such a model is efficient because it runs the last two communications, acknowledgment for *BS* or *GWN* and user, simultaneously. However, since both communications have to be run, it is insignificant regarding efficiency. In 2014, Turkanovic et al. proposed a user authentication and key agreement model to overcome the security flaws of the earlier designed protocols (Turkanovic et al., 2014). Farash et al. shown that Turkanovic et al.'s protocol is insecure and inefficient for various security drawbacks such as session key agreement, mutual authentication between all parties, traceability, preservation of user anonymity, privileged insider attack and password guessing attack (Farash et al., 2016). Farash et al.'s protocol still has the security pitfalls such as off-line password guessing attack, off-line identity

guessing attack. Simultaneously, Amin and Biswas shown that Turkanovic et al.'s protocol is insecure and inefficient for various security drawbacks such as off-line password guessing attack, off-line identity guessing attack, smart card theft attack, user impersonation attack, *SN* impersonation attack and also shown the protocol is vulnerable to inefficient authentication phase(Amin & Biswas, 2016).To overcome these shortcomings, Amin and Biswas presented a secure lightweight protocol for user authentication and key agreement in multi-*GWN* based WSNs. However, Sirinivas et al. observed that Amin and Biswas protocol is also vulnerable to a series of attacks, such as man in the middle attack, impersonation attack and password guessing attack (Srinivas et al., 2017). They also observed that Amin and Biswas's protocol has leakages of sensors secret keys and the system key, and is weak against server spoofing attack, user impersonation attack, stolen smart card attack, off-line password guessing attack and ID guessing attack.To overcome all the mentioned shortcomings Srinivas et al. presented a secure and efficient user authentication protocol for multi-*GWN*WSNs. They argued that their protocol is secure enough to withstand various kinds of attacks. However, Kuonga et al. showed weaknesses of Srinivas et al.'s protocol, which are weak against *GWN* masquerading attack, *SN* masquerading attack and it does not provide unlinkability and anonymity (Kuonga et al., 2019). Therefore, Kuonga et al. proposed unlinkable user authenticated key agreement for multi-gate way WSN to overcome all the shortcomings in Srinivas's et al.'s protocol as part of this Thesis project.

CHAPTER 3

UNLINKABLE USER AUTHENTICATED KEY AGREEMENT FOR MULTI-GATEWAY WSNS

This chapter proposes a new unlinkable user authenticated key agreement for multi-*GWN*WSNs, which is denoted as UAKA. The security of UAKA is based on the one-wayness of the hash function and the secrecy of the symmetric key cryptography. There by, UAKA is very lightweight but preserves privacy and provides security. UAKA uses message transfer between different *GWN*s if the target *SN* is not in the reign of the communicating *HGWN*. So, we need only one scenario for the login and authenticated key agreement compared to the related multi-*GWN* protocols, which requires two. UAKA has five phases including system setup phase, registration phase, login authenticated key agreement phase, dynamic node addition phase and password change phase.

To start up a communication, system administrator (*SA*) performs system setup for a specific WSN. *SA* generates identity and security parameters for every *SN*. If this is successful, both *SN* and a user U_i needs to get registered to *GWN* so that U_i can connect with the opted *SN*. After the successful registration, login and authenticated key agreement is executed if U_i wants to access any data from *SN* via *HGWN* or *FGWN* where the shared session key is also established.

Sometimes, it is of great important that *SN* is dynamically added in the target field in order to increase scalability and strength of *SN* based on dynamic node addition. UAKA gives an opportunity to U_i to change his (or her) password any time he (or she) feels like doing so.

3.1 System setup phase

It is an off-line mode, where *SA* generates identity and security parameters for each *SN*. First, *SA* generates the identities { ID_{SN1}, ID_{SN2}, ⋯, ID_{SNm}} for each *SN*{ SN_1, SN_2, …,SN_m} such that no two distinct SNs will get same identity. Then, *SA* computes $P_j=h(ID_{SNj} \oplus S_{ran})$ for $1 \leq j \leq m$, where S_{ran} is a random secret known to all *GWNs*. *SA* stores $<ID_{SNj}, P_j>$ for $1 \leq j \leq m$ into the memory of *SN*s before their deployment.

3.2 Registration phase

It is divided into two sub-phases, *SN* registration and user registration. *SN* should be registered to its' *HGWN* only once right after the deployment for the security reason. Also for user to get the services from any *SN*, any user needs to be registered. Once the user gets registered, he (or she) will be able to connect with the opted *SN*. Both users and *SN*s undergo this registration process, respectively.

3.2.1 SN registration phase

Soon after the deployment, *SN* has to be one of *GWN*s realm, which could be its' *HGWN*, by applying this phase. This is done through an open channel.*SN* registration phase is outlined in Fig. 2 and the detailed phase is as follows:

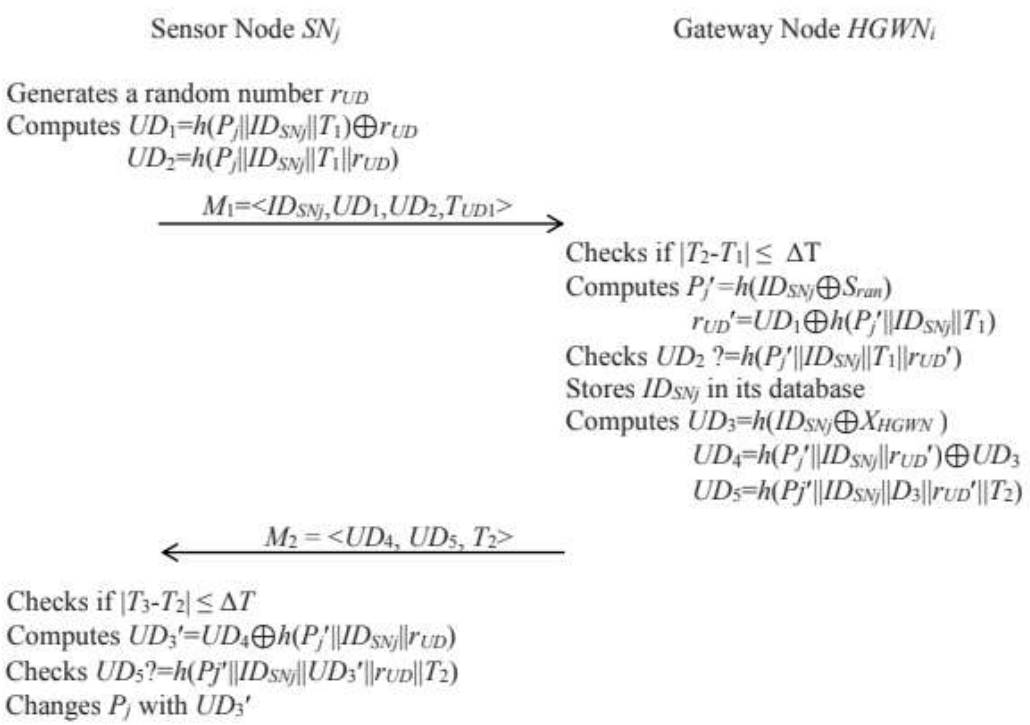

Figure 2: Sensor node registration phase

SR1: SN_j generates a random number r_{UD} and computes $UD_1=h(P_j||ID_{SNj}||T_1)$ and $UD_2=h(P_j||ID_{SNj}||T_1||r_{UD})$, where$T_1$ is the timestamp of SN_j. SN_j sends a registration request message $M_1 = <ID_{SNj}, UD_1, UD_2, T_1>$ to the nearest *HGWN* through an open channel.

SR2:*HGWN*checks$|T_2 - T_1| \leq \Delta T$, where ΔT is the predefined permitted transmission delay. Only if it holds, *HGWN* computes $P_j'=h(ID_{SNj}\oplus S_{ran})$ and $r_{UD}'=UD_1\oplus h(P_j'||ID_{SNj}||T_1)$. After that *HGWN* verifies UD_2 ?$=h(P_j'||ID_{SNj}||T_1||r_{UD}')$. *HGWN* terminates the session if the verification fails. Otherwise, *HGWN* stores ID_{SNj} in its database and computes $UD_3=h(ID_{SNj}\oplus X_{HGWN})$,$UD_4=h(P_j'||ID_{SNj}||r_{UD}')\oplus UD_3$and

$UD_5=h(P_j'||ID_{SNj}||UD_3||r_{UD}'||T_2)$, which X_{HGWN} is a 1,024 bits secret key of *HGWN*. Then *HGWN* sends $M_2=<UD_4, UD_5, T_2>$ to SN_j.

SR3 : Upon receiving the message, SN_j checks if $|T_3-T_2|\leq\Delta T$. Only if it satisfies, SN_j computes $UD_3'=UD_4\oplus h(P_j'||ID_{SNj}||r_{UD})$ and verifies $UD_5?=h(P_j'||ID_{SNj}||UD_3'||r_{UD}||T_2)$. Only if the verification holds, SN_j changes P_j with UD_3'.

This phase has two major functions, to update the secret parameter S_{ran} in P_j so that every *HGWN* has its own secrete parameter and also to make sure that *HGWN* knows which SN_j is in its region.

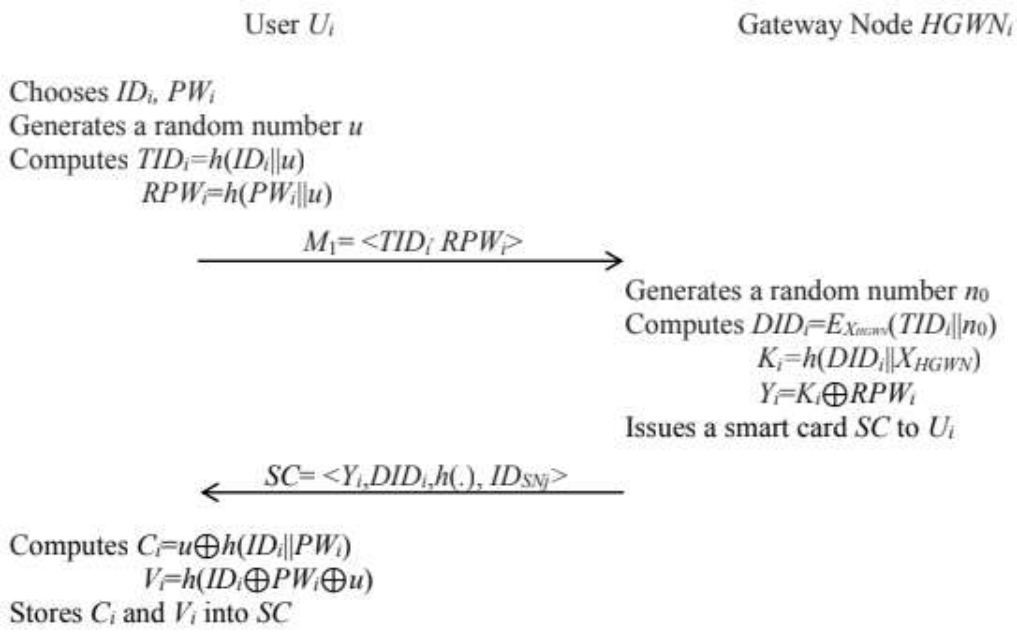

Figure 3: User registration phase

3.2.2 User registration phase

User U_i needs to complete his (or her) registration at *HGWN*, and achieve personalized security parameters to access *SN*. The user registration phase is shown in Fig. 3 and the detailed description is as follows:

UR1 : U_i selects his (or her) identity ID_i, password PW_i and a random number u. U_i computes $TID_i=h(ID_i||u)$and $RPW_i=h(PW_i||u)$, and then submits a registration message $M_1=<TID_i, RPW_i>$ to *HGWN* via a secure channel.

UR2 : On receiving the registration request, *HGWN* generates a random number n_0, encrypts $DID_i=E_{XHGWN}(TID_i||n_0)$, and computes $K_i=h(DID_i||X_{HGWN})$and $Y_i=K_i \oplus RPW_i$. *HGWN* issues a smart card (*SC*) for U_i, such that $SC = <Y_i, DID_i, h(\cdot), ID_{SNj}>$and sends it to U_i.

UR3 : Upon receiving *SC*, U_i computes $C_i=u \oplus h(ID_i||PW_i)$and $V_i=h(ID_i \oplus PW_i \oplus u)$. U_i stores C_i and V_i into *SC*.

3.3 Login and authenticated key agreement phase

Upon the successful registration of *SN* and U_i, both *SN* and U_i have to perform authenticated key agreement with the *GWN*. Only authentic *SN* and U_i can communicate but if the authenticity of one of these entities does not hold then the process is terminated.

HGWN maintains the public directory, which comprises *SN*s identities in the WSN. This enables any user U_i to select a *SN* as per his (or her) requirement. To get services from SN_j, U_i extracts ID_{SNj} from the public directory of *HGWN*. The registered U_i inserts his (or her)

SC into a card reader to initiate the login and authenticated key agreement phase. *HGWN* maintains the public directory of all the *SN*s. So, whenever a registered user, U_i wants to get services from a *SN*, SN_j, U_i can pick the appropriate SN_j's identity in the service environment of the WSN. In order to access the services, U_i first initiates the login session using his (or her) *SC*. The authenticity of U_i is verified in the *SC* authentication. Once the legitimacy of U_i is verified, a login message is forwarded to the *HGWN* which includes SN's identity, ID_{SNj}, where the existence of ID_{SNj} is checked in its database. If ID_{SNj} exists in *HGWN*'s database. This phase can be seen as two sub-phases. Once the legitimacy of U_i is verified, a login message is forwarded to the proper *HGWN* through the public channel in order to login to the desired SN_j. The procedure of the login and authenticated key agreement phase is described in the following sub-sections.

3.3.1 Login phase

First, *SC* needs to verify the legitimacy of U_i. For the valid U_i, *SC* processes the login request. U_i executes the login request as follows:

LG1: To start the login process, U_i inserts *SC* into a terminal and inputs his (or her) ID_i' and PW_i'. Then *SC* computes $u'=C_i \oplus h(ID_i' \| PW_i')$ and checks V_i ?= $h(ID_i' \oplus PW_i' \oplus u')$. *SC* inquires for sensor identity as per requirement to *HGWN* upon the successful verification, by observing the user requirement and sensors availability, *HGWN* sends the available sensor's identity ID_{SNj} to U_i. Only if the verification holds, *SC* generates a random number r_i, and computes $K_i'=Y_i \oplus h(PW_i' \| u')$, $D_1=h(K_i' \| DID_i \| ID_{SNj}) \oplus r_i$ and

$D_2=h(K_i'||r_i||T_1||DID_i||ID_{SNj}||TID_i)$. After that, U_i sends a login message $M_1=<DID_i,ID_{SNj}, D_1, D_2, T_1>$ to *HGWN*.

3.3.2 Authenticated key agreement phase

After receiving the login message from U_i, *HGWN* checks whether the requested SN_j is in the registered *SN* list or not by checking its database. Only if SN_j is in its database, *HGWN* executes the authenticated key agreement. Otherwise, it forwards the message to the appropriate other *FGWN*. The message exchange of login and authenticated key agreement is discussed in Fig. 4 and the details of this phase are as follows:

AK1: On receiving the login message $<DID_i, ID_{SNj}, D_1, D_2, T_1>$ at T_2, *HGWN* checks the freshness of the message as $|T_2–T_1| \leq \Delta T$. Only if the verification passes, *HGWN* decrypts $TID_i||n_0=D_{XHGWN}(DID_i)$, computes $K_i''=h(TID_i||X_{HGWN})$ and retrieves $r_i'=D_1 \oplus h(K_i''||DID_i||ID_{SNj})$. Then, *HGWN* verifies $D_2?=h(K_i''||r_i'||T_1||DID_i||ID_{SNj}||TID_i)$. Only if the verification holds, *HGWN* authenticates U_i. Otherwise, the connection is terminated. *HGWN* generates a random number r_h and computes $P_j=h(ID_{SNj} \oplus X_{HGWN})$, $D_3=h(P_j||ID_{SNj}||T_2) \oplus r_h$, $D_4=h(P_j||r_h||T_2) \oplus r_i'$ and $D_5=h(DID_i||P_j||r_i'||r_h||T_2)$. *HGWN* forms message $M_2 = <DID_i, D_3, D_4, D_5, T_2>$ and sends it to SN_j.

AK2: On receiving the message at T_3, SN_j checks the freshness of the message as $|T_3 - T_2| \leq \Delta T$. Only if the verification is valid, SN_j extracts $r_h'=D_3 \oplus h(P_j||ID_{SNj}||T_2)$, $r_i''=D_4 \oplus h(P_j||r_h'||T_2)$, and then verifies $D_5 ?=h(DID_i||P_j||r_i''||r_h'||T_2)$. If the verification does not hold, the connection is aborted. Otherwise, SN_j generates a

random number r_j and computes $D_6=h(P_j||r_h'||T_3)\oplus r_j$ and $D_7=h(P_j||r_i''||r_h'||r_j||T_3)$. Then, SN_j sends the message $M_3= <D_6, D_7, T_3>$ to *HGWN*.

AK3: On receiving the message at T_4, *HGWN* checks $|T_4 - T_3| \leq \Delta T$. If the verification is valid, *HGWN* computes $r_j'=D_6\oplus h(P_j||r_h||T_3)$ and verifies $D_7 ?=h(P_j||r_i'||r_h||r_j'||T_3)$. If the verification does not hold, the connection is aborted. Otherwise, *HGWN* computes $D_8=h(K_i''||DID_i||r_i')\oplus r_h$, $D_9=h(K_i''||DID_i||r_i'||r_h)\oplus r_j'$, $D_{10}=E_{XHGWN}(TID_i||r_h)$, $D_{11}=h(K_i''||DID_i||r_i'||r_h)\oplus D_{10}$ and $D_{12}=h(K_i''||DID_i||D_{10}||r_i'||r_h||r_j'||T_4)$ and sends a message $M_4 = <D_8, D_9, D_{10}, D_{11}, D_{12}, T_4>$ to U_i.

AK4: On receiving the message at T_5, U_i checks $|T_5 - T_4| \leq \Delta T$. If the verification is valid, U_i computes $r_h''=D_8\oplus h(K_i||DID_i||r_i)$, $r_j''=D_9\oplus h(K_i||DID_i||r_i||r_h'')$ and $D_{10}'=D_{11}\oplus h(K_i||DID_i||r_i||r_h'')$ and then verifies $D_{12}?=h(K_i||DID_i||D_{10}'||r_i||r_h''||r_j''||T_4)$. If the verification does holds, U_i changes DID_i with D_{10}'. Hence, it is confirmed that SN_j is authentic. But if not, the connection is aborted. On the success of mutual authentication, a session key $SK = h(DID_i||r_i||r_j||r_h||ID_{SNj})$ is constructed by involved entities in the system.

User U_i | Gateway node *HGWN* | Sensor node SN_j

Inserts *SC* into a terminal
Inputs ID_i' and PW_i'
Computes $u'=c_i \oplus h(ID_i'||PW_i')$
Verifies $V_i^? = h(ID_i'||PW_i'||u')$
Generates a random number r_i
Computes $K_i'=Y_i \oplus h(PW_i'||u')$
$D_1=h(K_i'||DID_i||ID_{SNj}) \oplus r_i$
$D_2=h(K_i'||r_i||T_1||DID_i||ID_{SNj}||TID_i)$

$M_1=<DID_i, ID_{SNj}, D_1, D_2, T_1>$ →

Checks if $|T_2 - T_1| \leq \Delta T$
Computes $K_i''=h(DID_i||X_{HGWN})$
$TID_i||n_0=D_{XHGWN}(DID_i)$
$r_i'=D_1 \oplus h(K_i''||DID_i||ID_{SNj})$
Verifies $D_2 ?= h(K_i''||r_i'||T_1||DID_i||ID_{SNj}||TID_i)$
Generates a random number r_h
Computes $P_j=h(ID_{SNj} \oplus X_{HGWN})$
$D_3=h(P_j||ID_{SNj}||T_2) \oplus r_h$
$D_4=h(P_j||r_h||T_2) \oplus r_i'$
$D_5=h(DID_i||P_j||r_i||r_h||T_2)$

$M_2=<DID_i, D_3, D_4, D_5, T_2>$ →

Checks if $|T_3 - T_2| \leq \Delta T$
Computes
$r_h'=D_3 \oplus h(P_j||ID_{SNj}||T_2)$
$r_i''=D_4 \oplus h(P_j||r_h'||T_2)$
Verifies $D_5 ?= h(DID_i||P_j||r_i''||r_h'||T_2)$
Generates a random number r_j
Computes $D_6=h(P_j||r_h'||T_3) \oplus r_j$
$D_7=h(P_j||r_i''||r_h'||r_j||T_3)$

← $M_3=<D_6, D_7, T_3>$

Checks if $|T_4 - T_3| \leq \Delta T$
Computes $r_j'=D_6 \oplus h(P_j||r_h||T_3)$
Verifies $D_7 ?= h(P_j||r_i'||r_h||r_j'||T_3)$
Computes $D_8=h(K_i''||DID_i||r_i') \oplus r_h$
$D_9=h(K_i''||DID_i||r_i'||r_h) \oplus r_j'$
$D_{10}=E_{XHGWN}(TID_i||r_h)$
$D_{11}=h(K_i''||DID_i||r_i'||r_h) \oplus D_{10}$
$D_{12}=h(D_{10}||X_{HGWN})$
$D_{13}=h(k_i''||DID_i||r_i'||r_h) \oplus D_{12}$
$D_{14}=h(K_i''||DID_i||D_{10}||D_{12}||r_i'||r_h||r_j'||T_4)$

← $M_4=<D_8, D_9, D_{11}, D_{13}, D_{14}, T_4>$

Checks if $|T_5 - T_4| \leq \Delta T$
Computes $r_h''=D_8 \oplus h(K_i'||DID_i||r_i)$
$r_j''=D_{19} \oplus h(K_i'||DID_i||r_i||r_h'')$
$D_{10}'=D_{11} \oplus h(K_i'||DID_i||r_i||r_h'')$
$D_{12}'=D_{13} \oplus h(K_i'||DID_i||r_i||r_h'')$
Verifies $D_{14} ?= h(K_i'||DID_i||D_{10}'||D_{12}'||r_i||r_h''||r_j''||T_4)$
Changes DID_i with D_{10}' and K_i'' with D_{12}'
Shared session key $SK= h(DID_i||r_i||r_j||r_h||ID_{SNj})$

Figure 4: Login phase and authenticated key agreement phase

3.4 Dynamic node addition phase

It may happen that a new *SN* needs to be added over the target WSN field as and when required, after the establishment of the WSN. So, *SA* deploys the new *SN* over the target WSN region by performing the system setup phase in off-line mode. Then after, the newly added *SN* under-goes the *SN*registration phase and introduce the new *SN* into the WSN.

3.5 Password change phase

In *SC* based authenticated key agreement, protocols should be able to address password related attacks so that user with valid *SC* and personal credentials can initiate the password change phase. Additionally, user should be able to choose and change the password without interaction with *SA* or *HGWN*, which is to provide user-friendly password selection and change. The proposed password change phase requires user to change the password without interaction with the other network entity.

Any user U_i with valid credentials and *SC* can initiate this phase by inputting ID_i' and PW_i'. *SC* computes $u'=C_i \oplus h(ID_i'||PW_i')$. To resist against password related attacks, *SC* verifies $V_i ?= h(ID_i'||PW_i'||u')$. Using this condition, *SC* identifies the correctness of user credentials. If verification holds, *SC* asks for a new password PW_{new} to U_i. On receiving PW_{new}, *SC* computes $RPW_{new}=h(PW_{new}||u')$ and updates $Y_i=Y_i \oplus RPW_i \oplus RPW_{new}$, $C_i=C_i \oplus h(ID_i||PW_i) \oplus h(ID_i'||PW_{new})$ and $V_i=h(ID_i' \oplus PW_{new} \oplus u')$ on *SC*.

CHAPTER 4

ANALYSIS

This chapter provides security analysis and performance analysis. First of all, we provide BAN logic analysis and informal security analysis to show the security and privacy of UAKA. BAN logic is used to verify the correctness of the authentication protocol with key agreement or the authenticated key agreement protocol but it does not provide an explanation on how it will deal with the detailed attack scenarios. Hence the use of informal security analysis helps to know how UAKA cope from various forms of attacks. Performance analysis is focused on computational and communicational overheads with the comparisons of UAKA with the related protocols (Farash et al., 2016; Amin and Biswas, 2016; Srinivas et al., 2017).

4.1 Security analysis

In this section, we first provide a proof of the mutual authentication and session key agreement using the BAN logic. Secondly, we provide an informal security proof to check the strength of UAKA against various attacks namely masquerading attack, password guessing attack, DoS attack, privacy attack and many more attacks.

4.1.1 BAN logic analysis

In this section, we provide a formal protocol analysis of UAKA using the BAN logic (Burrows et al., 1990). The BAN logic is used to verify the correctness of the authentication protocol with key agreement or the authenticated key agreement protocol. The formal analysis of UAKA using BAN logic involves following steps:

(1) Converting original protocol statements to their idealized form.

(2) Determining the assumptions about the initial state of the system.

(3) Representation of the state of the system after executing each statement as logical assertions by attaching logical formulas to each statement.

(4) Application of logical postulates to assumptions and assertions.

The following notations are used in formal security analysis using the BAN logic:

- $Q \mid\equiv X$: Principal Q believes the statement X.
- $\#(X)$: Formula X is fresh.
- $Q \mid\Longrightarrow X$: Principal Q has jurisdiction over the statement X.
- $\overset{K}{\mid\rightarrow} Q$: Principal Q has a public key K.
- $Q \lhd X$: Principal Q sees the statement X.
- $Q \mid\sim X$: Principal Q once said the statement X.
- (X, Y): Formula X or Y is one part of the formula (X, Y).
- $\langle P \rangle_Q$: Formula P combined with the formula Q.

- $Q \overset{SK}{\leftrightarrow} R$: Principal *Q* and *R* may use the shared session key, *SK* to communicate among each other. The session key *SK* is good, in that it will never be discovered by any principal except *Q* and *R*.

In addition, the following four BAN logic rules are used to prove that UAKA provides a secure mutual authentication among U_i, *HGWN* and SN_j :

Rule 1. **Message-meaning rule**: $\frac{R \mid\equiv R \overset{Y}{\leftrightarrow} S,\ \ R \lhd <X>_Y}{R \mid\equiv S \mid\sim X}$

Rule 2. **Nonce-verification rule**: $\frac{R \mid\equiv \#(X),\ \ R \mid\equiv S \mid\sim X}{R \mid\equiv S \mid\equiv X}$

Rule 3. **Jurisdiction rule**: $\frac{R \mid\equiv S \mid\Rightarrow X,\ \ R \mid\equiv S \mid\equiv X}{R \mid\equiv X}$

Rule 4. **Freshness-concatenation rule**: $\frac{R \mid\equiv \#(X)}{R \mid\equiv \#(X,Y)}$

In order to show that UAKA provides secure mutual authentication between among U_i, *HGWN* and SN_j, we need to achieve the following goals:

Goal 1: $U_i \mid\equiv (U_i \overset{SK}{\leftrightarrow} SN_j)$

Goal 2: $SN_j \mid\equiv (SN_j \overset{SK}{\leftrightarrow} U_i)$

Goal 3: $U_i \mid\equiv SN_j \mid\equiv (SN_j \overset{SK}{\leftrightarrow} U_i)$

Goal 4: $SN_j \mid\equiv U_i \mid\equiv (U_i \overset{SK}{\leftrightarrow} SN_j)$

Idealized form: The arrangement of the transmitted messages among U_i, *HGWN* and SN_j in UAKA to the idealized forms is as follows:

Message 1. $U_i \rightarrow HGWN$: $<DID_i>_{XHGWN}$, $<ID_{SNj}>$, $<D_1>_{Ki}$, $<D_2>_{Ki}$, $<T_1>$

Message 2. $HGWN \rightarrow SN_j$:$<DID_i>_{XHGWN}$, $<D_3>_{Pj}$, $<D_4>_{Pj}$, $<D_5>_{Pj}$, $<T_2>$

Message 3. $SN_j \rightarrow HGWN$:$<D_6>_{Pj}$, $<D_7>_{Pj}$, $<T_3>$

Message 4. $HGWN \rightarrow U_i$:$<D_8>_{Ki}$, $<D_9>_{Ki}$, $<D_{11}>_{Ki}$, $<D_{13}>_{Ki}$, $<D_{14}>_{Ki}$, $<T_2>$

Assumptions: The following are the initial assumptions of UAKA:

A1: $U_i| \equiv \#(r_i, T_1)$

A2: $HGWN| \equiv \#(r_h, T_2, T_4)$

A3: $SN_j| \equiv \#(r_j, T_3)$

A4: $U_i| \equiv (U_i \overset{(K_i)}{\longleftrightarrow} HGWN)$

A5: $HGWN| \equiv (HGWN \overset{(K_i)}{\longleftrightarrow} U_i)$

A6: $HGWN| \equiv (HGWN \overset{P_j}{\leftrightarrow} SN_j)$

A7: $SN_j| \equiv (SN_j \overset{P_j}{\leftrightarrow} HGWN)$

A8: $U_i| \equiv SN_j| \Longrightarrow U_i \overset{SK}{\leftrightarrow} SN_j$

A9: $SN_j| \equiv U_i| \Longrightarrow SN_j \overset{SK}{\leftrightarrow} U_i$

Proof:

In the following, we prove the test goals in order to show the secure authentication using the BAN logic rules and the assumptions.

Based on Message 1, we could derive:

Step 1. $HGWN \lhd <DID_i>_{XHGWN}$, $<ID_{SNj}>$, $<D_1>_{Ki}$, $<D_2>_{Ki}$, $<T_1>$

According to assumption A4 and the message meaning rule, we get:

Step 2. $HGWN|\equiv U_i|\sim(<DID_i>_{XHGWN}, <ID_{SNj}>, <D_1>_{Ki}, <D_2>_{Ki}, <T_1>)$

According to assumption A1 and the freshness concatenation rule, we get:

Step 3.$HGWN|\equiv \#(<DID_i>_{XHGWN}, <ID_{SNj}>, <D_1>_{Ki}, <D_2>_{Ki}, <T_1>)$

According to Step 2, Step 3 and the nonce verification rule, we get:

Step 4. $HGWN|\equiv U_i|\equiv (<DID_i>_{XHGWN}, <ID_{SNj}>, <D_1>_{Ki}, <D_2>_{Ki}, <T_1>)$

According to Step 4, assumption A4 and the believe rule, we get:

Step 5. $HGWN|\equiv U_i|\equiv (U_i \overset{(K_i)}{\longleftrightarrow} HGWN)$

According to the jurisdiction rule, we get:

Step 6. $HGWN|\equiv (HGWN \overset{(K_i)}{\longleftrightarrow} U_i)$

Based on Message 2, we derive

Step 7. $SN_j \lhd <DID_i>_{XHGWN}, <D_3>_{Pj}, <D_4>_{Pj}, <D_5>_{Pj}, <T_2>$

According to assumption A7 and the message meaning rule, we get:

Step 8. $SN_j|\equiv HGWN|\sim(<DID_i>_{XHGWN}, <D_3>_{Pj}, <D_4>_{Pj}, <D_5>_{Pj}, <T_2>)$

According to assumption A2 and the freshness concatenation rule, we get:

Step 9.$SN_j|\equiv \#(<DID_i>_{XHGWN}, <D_3>_{Pj}, <D_4>_{Pj}, <D_5>_{Pj}, <T_2>)$

According to Step 8, Step 9 and the nonce verification rule, we get:

Step 10. $SN_j|\equiv HGWN\ |\equiv (<DID_i>_{XHGWN}, <D_3>_{Pj}, <D_4>_{Pj}, <D_5>_{Pj}, <T_2>)$

According to Step 10, assumption A6 and the believe rule, we get:

Step 11. $SN_j|\equiv HGWN|\equiv (HGWN \overset{P_j}{\leftrightarrow} SN_j)$

According to the jurisdiction rule, we get:

Step 12. $SN_j|\equiv (SN_j \overset{P_j}{\leftrightarrow} HGWN)$

According to Step 8, Step 9, Step 10 and the nonce verification rule, we get:

Step 13. $SN_j|\equiv U_i|\equiv (U_i \overset{SK}{\leftrightarrow} SN_j)$

(Goal 4)

According to assumption A8 and the jurisdiction rule, we get:

Step 14. $SN_j|\equiv (SN_j \overset{SK}{\leftrightarrow} U_i)$

(Goal 2)

Based on Message 3, we derive

Step 15. $HGWN \lhd <D_6>_{Pj}, <D_7>_{Pj}, <T_3>$

According to assumption A6 and the message meaning rule, we get:

Step 16. $HGWN|\equiv SN_j|\sim(<D_6>_{Pj}, <D_7>_{Pj}, <T_3>)$

According to assumption A3 and the freshness concatenation rule, we get:

Step 17.$HGWN|\equiv \#(<D_6>_{Pj}, <D_7>_{Pj}, <T_3>)$

According to Step 16, Step 17 and the nonce verification rule, we get:

Step 18. $HGWN|\equiv SN_j|\equiv (<D_6>_{Pj}, <D_7>_{Pj}, <T_3>)$

According to Step 18, assumption A7 and the believe rule, we get:

Step 19. $HGWN|\equiv SN_j|\equiv (SN_j \overset{P_j}{\leftrightarrow} HGWN)$

According to Step 16, Step 17, Step 18 and the nonce verification rule, we get:

Step 20. $HGWN|\equiv SN_j|\equiv (SN_j \overset{SK}{\leftrightarrow} HGWN)$

According to assumption A10 and the jurisdiction rule, we get:

Step 21. $HGWN|\equiv (HGWN \overset{SK}{\leftrightarrow} SN_j)$

Based on Message 4, we derive

Step 22. $U_i \triangleleft <D_8>_{Ki}, <D_9>_{Ki}, <D_{11}>_{Ki}, <D_{13}>_{Ki}, <D_{14}>_{Ki}, <T_2>$

According to assumption A4 and the message meaning rule, we get:

Step 23. $U_i|\equiv HGWN|\sim(<D_8>_{Ki}, <D_9>_{Ki}, <D_{11}>_{Ki}, <D_{13}>_{Ki}, <D_{14}>_{Ki}, <T_2>)$

According to assumption A2 and the freshness concatenation rule, we get:

Step 24.$U_i|\equiv \#(<D_8>_{Ki}, <D_9>_{Ki}, <D_{11}>_{Ki}, <D_{13}>_{Ki}, <D_{14}>_{Ki}, <T_2>)$

According to Step 23, Step 24 and the nonce verification rule, we get:

Step 25. $U_i|\equiv HGWN|\equiv (<D_8>_{Ki}, <D_9>_{Ki}, <D_{11}>_{Ki}, <D_{13}>_{Ki}, <D_{14}>_{Ki}, <T_2>)$

According to Step 25, assumption A5 and the believe rule, we get:

Step 26. $U_i|\equiv HGWN|\equiv (HGWN \overset{K_i}{\leftrightarrow} U_i)$

According to Step 23, Step 24, Step 25 and the nonce verification rule and the jurisdiction rule, we get:

Step 27. $U_i|\equiv SN_j|\equiv (SN_j \overset{SK}{\leftrightarrow} U_i)$

(Goal 3)

According to assumption A8 and the jurisdiction rule, we get:

Step 28. $U_i|\equiv (U_i \overset{SK}{\leftrightarrow} SN_j)$

(Goal 1)

According to Steps 14 and 28, UAKA successfully achieves both goals (Goals 1 and 2). Both U_i and SN_j believes that they share a common session key $SK=h(DID_i||r_i||r_j||r_h||ID_{SNj})$.

4.1.2 Informal security analysis

Although it is important to provide a formal security proof on any cryptographic protocol, the formal security proof of protocols remains one of the most challenging issues for cryptography research. Until now, a simple, efficient and convincing formal methodology

for correctness analysis on security protocols is still an important subject of research and an open problem. Because of these reasons, most protocols have been demonstrated with a simple proof. This section follows the security analysis approaches used in (Kim, 2014). As shown in Table 1, the security analysis is focused on verifying the overall security requirements for UAKA, including passive and active attacks.

Proposition 1. *UAKA provides anonymity and unlinkability*.

Proof: Anonymity is a property of network security. An entity in a system has anonymity if no other entity can identify the first entity, nor is there any link back to the first entity that can be used, nor any way to verify that any two anonymous acts are performed by the same entity. As shown in **proposition 6**, it is clear from UAKA that an attacker has no way to obtain or guess the identity ID_i of U_i as it is not only protected by symmetric key cryptography but also using pseudo-identity. Thereby, UAKA provides anonymity and also unlinkability.

Proposition2. *UAKA is secure against HGWN masquerading attack.*

Proof: By definition, this is the attack in which an attacker pretends to be a legitimate *HGWN* and plays in between U_i and SN_j with the assumption that the attacker could obtain any messages transmitted in the previous sessions. In UAKA, the attacker could try to form M_2=<DID_i, D_3, D_4, D_5, T_2> or M_4=<D_8, D_9, D_{11}, D_{13}, D_{14}, T_4> right after receiving M_1=<DID_i, ID_{SNj}, D_1, D_2, T_1> from U_i for the trial of this attack. However, they are impossible to the attacker in UAKA because they require knowledge of the important secret key, X_{HGWN} of *HGWN* for them to send M_2=<DID_i, D_3, D_4, D_5, T_2>. Again to U_i, the

attacker needs to form a correct M_4=<D_8, D_9, D_{11}, D_{13}, D_{14}, T_4>, which requires the knowledge of K_i'' where $K_i''=h(D_{12}'||X_{HGWN})$. Without the knowledge of X_{HGWN}, the attacker could not form the proper message M_4. In the other hand, against to SN_j, the attacker needs to form M_2=<D_3, D_4, D_5, T_2>, which requires the knowledge of P_j where $P_j=h(ID_{SNj}\oplus X_{HGWN})$. The attacker could not do anything to form the proper message with the same reason for U_i. There is no feasible way the attacker knows X_{HGWN} or P_j. Hence we can confirm that UAKA resists *HGWN* masquerading attack.

Proposition 3. *UAKA is secure against SN_j masquerading attack.*

Proof: With the similar definition of the attack on *HGWN* and the assumption, to masquerade as a legitimate SN_j, an attacker needs to form a proper response message M_3=<D_6, D_7, T_3> to *HGWN*. For the attacker to do this, he (or she) must have the knowledge of $P_j=h(ID_{SNj}\oplus X_{HGWN})$. However, it is not possible in UAKA as the attacker does not have the knowledge of the secret key X_{HGWN} of the involved parties. Thus, it will be impossible for him (or her) to compute the message M_3 correctly. Therefore, UAKA can resist the *SN* masquerading attack.

Proposition 4. *UAKA is secure against replay attack.*

Proof: Replay attack is an attack where the attacker captures the previously transmitted messages and uses them during UAKA execution to make the receiver of the message believe that the transmitted message is from a legal entity. In order to justify UAKA resist from the replay attack, we assume that the attacker has captured the previous session messages of UAKA and later tries to transmit the same message to the targeted entity. In a

replay attack, it does not matter if the attacker who intercepted the original message can read or decipher the key. All he (or she) has to do is capture and resent the entire thing - message and key - together. To counter this possibility, UAKA uses random session key, which is a type of code that is only valid for one transaction and cannot be used again. For an example, when preparing M_1=<DID_i, ID_{SNj}, D_1, D_2, T_1>, UAKA uses r_i as its random number while in when preparing M_2=<D_3, D_4, D_5, T_2>, *HGWN* generate r_h as its random number and SN_j generates r_j as its random number to send a message to *HGWN*. Another preventative measure for this type of attack is using time stamps on all messages as we can see each and every message in UAKA. This prevents hackers from resenting messages sent longer ago than a certain period of time, thus reducing the window of opportunity for an attacker to eavesdrop, siphon off the message, and resent it. In specific, time stamp and random number are used together to guarantee the freshness of each message. Following this, we can conclude that UAKA is strong against replay attack.

Proposition 5. *UAKA could withstand trace attack.*

Proof: Trace attack is an attack against unlinkability where the attacker can distinguish the messages communicated between entities by eavesdropping on a communication. For an attacker to achieve this, he (or she) intercepts two or more messages from two or more different sessions and checks whether they have something in come that can be computed by the attacker. If it happens, the attacker believes that these two messages belong to the same source, either from U_i or *HGWN*. However, the attacker cannot trace U_i, *HGWN* and SN_j after intercepting the communicating messages because UAKA updates DID_i and K_i apart from that he (or she) uses the one-way hash function and the symmetric key

cryptography, which are infeasible for an attacker to compute important parameters such as X_{HGWN}.

Proposition 6. *UAKA could withstand privileged insider attack.*

Proof: An insider attack is defined as a malicious attack perpetrated on a network or computer system by a person with the authorized system access. Practically, in UAKA, it is assumed that *HGWN* is trusted. So, *HGWN* provides confidentiality to U_i's credential, where leakage of any confidential parameters of the user is not permitted. But, it is observed that due to the presence of an insider, systems can get hacked. Therefore, U_i's information such as identity and password should be kept secret such that the insider of the *HGWN* cannot gain control over U_i's information. In UAKA, during user registration phase, instead of the original ID_i and PW_i the masked identity $TID_i=h(ID_i||u)$ and password $RPW_i=h(PW_i||u)$ were used. Hence, extracting U_i's password or identity by the insider of *HGWN* is computationally infeasible due to the non-invertible property of the one-way hash function and the symmetric key cryptography. Therefore, UAKA can resist privileged insider attack.

Proposition 7. *UAKA could withstand password guessing attack.*

Proof: A password guessing attack is an attack that consists of an attacker trying many passwords or pass phrases with the hope of eventually guessing correctly. We suppose that U_i's *SC* was stolen by an attacker, then the attacker can extract the information stored on

SC <Y_i, DID_i, $h(\cdot)$, ID_{SNj}, C_i, V_i> by using the method of power analysis, where $V_i=h(ID_i||PW_i||u)$, $C_i=u\oplus h(ID_i||PW_i)$ and $Y_i=K_i\oplus RPW_i$ (Eisnbarth et al., 2008). The attacker needs to know u, ID_i and PW_i, where this information are known only to U_i, and both user ID_i and PW_i are unknown to the attacker because they are well protected by the one-way hash function and the symmetric key cryptography. So, the attacker has no way to guess or exact U_i's ID_i and PW_i at the same time, as it is computationally infeasible to guess the two parameters at the same time. Hence, there is nowhere for an attacker to update PW_i of U_i. Therefore, UAKA is free from the stolen smart card attack.

Proposition 8. *UAKA could withstand DoS attack.*

Proof: DoS attack is an attack in which the perpetrator seeks to make a machine or network resource unavailable to its intended users by temporarily or indefinitely disrupting services of a host connected to Internet. In UAKA, we have three possibilities where a registered user could encounter DoS. However, UAKA is efficient to resist DoS attack in all scenarios as follows: In first situation when a user inputs incorrect credentials unknowingly during login phase, however, SC can correctly verify the login credentials using the condition $V_i?=h(ID_i||PW_i||u)$. This ensures that only with the correct input of user credentials a login message M_1=<DID_i, ID_{SNj}, D_1, D_2, T_1>will be executed. Thus, there will not be occurrence of DoS. Adversary may also try to engage sensors by replaying the messages so that valid user login attempt may deny or delayed. However, the transmitted message M_2=<DID_i, D_3, D_4, D_5, T_2> includes the time stamp. The sensor verifies the freshness of time stamp before professing the request. This shows that a sensor can efficiently encounter the fake request in UAKA, which shows the security of UAKA against DoS attack. The third

situation is where an adversary can mount an application layer DoS attack. This is a form of DoS attack, where attackers target the application layer of the open systems interconnection model. The attack over-exercises specific functions or features of a website with the intention to disable those functions or features. This application layer attack is different from an entire network attack. However this is not feasible in UAKA since UAKA involves the use of one-way hash function which very difficult for an attacker to compute it.

Table 1: Comparison of security features

Features \ Protocols	Farash et al.'s protocol	Amin and Biswas's protocol	Srinivas et al.'s protocol	UAKA
Anonymity and unlinkability	Not	Not	Not	Provide
Masquerading attack	Weak	Weak	Weak	Strong
Replay attack	Strong	Strong	Strong	Strong
Trace attack	Weak	Weak	Weak	Strong
Insider attack	Strong	Strong	Strong	Strong
Password guessing attack	Weak	Weak	Weak	Strong
DoS attack	Strong	Strong	Weak	Strong

4.2 Performance analysis

This section provides performance analysis focused on computational overhead and communicational overhead of UAKA. Furthermore, we provide two comparisons among

UAKA and the related protocols (Farash et al., 2016; Amin and Biswas, 2016; Srinivas et al., 2017). This gives an insight into the effectiveness of UAKA.

First of all, computational overhead of UAKA is considered based on basic operations we used, which is hash function and symmetric key encryption. This is the same as the other related protocols since they also used hash function and symmetric key cryptosystem. That is the reason why we need to consider those whole operations for the proper comparison. For the proper computational overhead measure, we use the result from Srinivas et al., which is based on MIRACL library with 32-bit Windows 7 operating systems and Visual C++ 2008.Symmetric key cryptosystem operation and hash function require 0.1303 ms and 0.0004 ms, respectively, if AES and SHA-1 are used. Table 2 shows the comparison of computational overhead among UAKA and the related protocols.

In login phase, UAKA requires $5T_h$ for U_i only. Authenticated key agreement requires $10T_h$, $16T_h+2T_{SE}$ and $5T_h$ for U_i, *HGWN* and SN_j, respectively. The other operations that we have used are XOR and concatenation. However, these operations are comparably negligible to hash function and symmetric key cryptography. Hence, we have not included the computation cost of them in computation overhead analysis. UAKA has 0.273 ms, which is a bit higher computation overhead compared to the other related protocols, which requires 0.0124 ms of Farash et al.'s, 0.0080 ms of Amin and Biswas's and 0.0116/0.0140 ms of Srinivas et al.'s. However, UAKA provide higher security and privacy features as shown in Table 1.

Table 2: Comparison of computational overhead at login and authenticated key agreement

Protocol / Entity	Farash et al.'s protocol	Amin and Biswas's protocol	Srinivas et al.'s protocol	UAKA
GWN	$14T_h$	$8T_h/7T_h$	$13T_h/16T_h$	$16T_h+ 2T_{SE}$
SN	$7T_h$	$5T_h/ 5T_h$	$6T_h/5T_h$	$5T_h$
U_i	$11T_h$	$7T_h/8T_h$	$10T_h/14T_h$	$10T_h$
Total	$32T_h$	$20T_h/20T_h$	$29T_h /35T_h$	$31T_h+ 2T_{SE}$
Time (ms)	0.0128 ms	0.0080/0.0080 ms	0.0116/0.0140 ms	0.273 ms

In Table 3, we have compared the communication overhead that is required for the login and authenticated key agreement among UAKA and the related protocols. We assumed to use SHA-1 with 160 bits and each timestamp, random number and *ID* of user or of *SN* with 152 bits. Login request message $M_1 =<TID_i, ID_{SNj}, D_1, D_2, T_1>$ requires 97 bytes. During the authenticated key agreement, messages $M_2 =<TID_i, D_3, D_4, D_5, D_6, T_2>$, $M_3 =<D_7, D_8, T_3>$ and $M_4=<D_9, D_{10}, D_{11}, T_4>$require 118 bytes, 59 bytes and 79 bytes, respectively. Thus during the login and authenticated key agreement in UAKA requires 97+98+59+119 = 373 bytes. In contrary, the communication overhead for Farash et al.'s protocol, Amin and Biswas's protocol and Srinivas et al.'s protocol requires 434 bytes, 373/642 bytes and 353/547 bytes, respectively. It is important to mention here that UAKA requires less communicational overhead than the other related protocols.

Table 3: Comparison of communicational overhead at login and authenticated key agreement

Protocol / Overhead	Farash et al.'s protocol	Amin and Biswas's protocol	Srinivas et al.'s protocol	UAKA
Total number of messages	4	4/8	4/7	4
Total bytes	434 bytes	373/642 bytes	353/547bytes	373bytes

CHAPTER 5

CONCLUSION

For the past decades, WSN has imposed a very big impact on the lives of people all over the world. Since WSN is affecting each and every side of the human's life convenience and has applications that are very necessary for all stakeholders.*SN* has limited resources such as bandwidth, storage, processing capability and energy. Therefore, once *SN* is compromised by adversaries, information of it has no privacy and security. Hence, security and privacy mechanism in WSN is particularly important. Authenticated key agreement is the most important security building blocks for WSN.

We have shown in this paper that the related protocol failed to provide unlinkability and anonymity and also we have shown that they are weak against most known attacks. In order to overcome the previous protocols security vulnerabilities in multi-*GWN* WSNs, we have proposed an unlinkable user authenticated key agreement, named as UAKA, for multi-*GWN*WSNs. Security and privacy on UAKA are based on one-wayness of hash function and secrecy of symmetric key cryptography, which has lightweight property especially for WSNs. UAKA supports dynamic node addition and provides user friendly password change. Security validation of UAKA has been done using BAN logic and informal cryptanalysis. It preserves all the original merits of the related protocols and provides security and privacy, which are unlinkability and anonymity, *GWN* and *SN* masquerading attacks, replay attack, trace attack, insider attack, password guessing attack and denial of service attack.

It is also important to note that UAKA has one weakness and that is, it has a bit higher computational overhead compared to the related protocols due to providing security and privacy. Thus we recommend that other researchers may think of how we can resolve this challenge while making sure that security and privacy is not compromised.

BIBILIOGRAPHY

Al-Janabi, S., Al-Shourbaji, I., Shojafar, M. & Shamshirband, S. (2017). Survey of main challenges (security and privacy) in wireless body area networks for healthcare applications. *Egyptian Informatics Journal*, 18(2), 113-122.

Al-Mousawi, A. J. & Al-Hassani, H. K. (2018). A survey in wireless sensor network for explosives detection, *Computers & Electrical Engineering*, 72, 682-701.

Amin R. & Biswas G. (2016). A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks. *Ad Hoc Networks*, 36, 58-80.

Asimi, Y., Asimi, A., Guezzaz, A., Tbatou, Z. & Sadqi, Y. (2018). Unpredictable cryptographic primitives for the robust wireless network security. *Procedia Computer Science*, 134, 316-321.

Atzori, L., Lera, A. & Morabito, G. (2010). The Internet of Things: A Survey. *Computer Networks*, 54(15), 2787-2805.

Burrows, M., Abadi, M. & Needham, R. (1990). A logic of authentication. *ACM transactions on Computer Systems*, 8(1), 18-36.

Chen, T.H. & Shih, W.K. (2010). A robust mutual authentication protocol for wireless sensor networks. *ETRI Journal*, 32(5),704-712.

Das, M. L. (2009). Two-factor user authentication in wireless sensor networks. *IEEE Transactions on Wireless Communications*, 8(3), 1086-1090.

[9] Das, A. K., Sharma, P., Chatterjee, S. & Sing, J. K. (2012). A dynamic password-based user authentication scheme for hierarchical wireless sensor networks. *Journal of Network and Computer Applications*, 35(5), 1646-1656.

Debnath, A., Singaravelu, P. and Verma, S. (2014). Privacy in wireless sensor networks using ring signature. *Journal of King Saud University – Computer and Information Sciences*, 26(2), 228-236.

Deebak, B. D. (2016). Secure and efficient mutual adaptive user authentication scheme for heterogeneous wireless sensor networks using multimedia client–server systems. *Wireless Personal Communications*, 87(3), 1013-1035.

Eisenbarth, T., Kasper, T.,Maradi, A.,Paar, C.,Salmasizadeh, M. & Shalmani, M.T.M. (2008). On the power of power analysis in real world: a complete break of the keeloq code hopping scheme. *Lecture Notes in Computer Science*, 5157, 203-220.

Farash, M. S., Turkanovic, M., Kumari, S. & Hölbl, M. (2016). An efficient user authentication and key agreement scheme for heterogeneous wireless sensor net- work tailored for the internet of things environment. *Ad Hoc Networks*, 36, 152-176.

Finogeev, A. G. & Finogeev, A. A. (2017). Information attacks and security in wireless sensor networks of industrial SCADA systems. *Journal of Industrial Information Integration*, 5, 6-16.

Gao, Y., Ao, H., Feng, Z., Zhou, W. & Tang, W. (2018). Mobile network security and privacy in WSN. *Procedia Computer Science*, 129, 324-330.

Gandotra, P. & Jha, R. K. (2017). A survey on green communication and security challenges in 5G wireless communication networks. *Journal of Network and Computer Applications*, 96, 39-61.

He, D., Gao, Y., Chan, S., Chen, C. & Bu, J., (2010). An enhanced two factor user authentication scheme in wireless sensor networks. *Ad Hoc & Sensor Wireless Networks*, 10(4), 361-371.

Huang, H., Savkin, A. V., Ding, M. & Huang, C. (2019). Mobile robots in wireless sensor networks: A survey on tasks. *Computer Networks*, 148, 1-19.

Jadhav, R. & Vatsala. (2017). Security Issues and Solutions in Wireless Sensor Networks. *International Journal of Computer Applications*,162(2), 14-19.

Kerckhoffs A., La (1883). Cryptographiemilitaire. *Journal des sciences militaries*, 9, 161-191.

Kim, H. (2014). Freshness-Preserving Non-Interactive Hierarchical Key Agreement Protocol over WHMS. *Sensors*, 14, 23742-23757.

Kocher, P., Jaffe, J. & Jun, B. (1999). Differential power analysis. *Lecture Notes in Computer Science*, 1666, 388-397.

Khan M. K. & Alghathbar, K. (2010). Cryptanalysis and security improvements of two–factor user authentication in wireless sensor networks. *Sensors*, 10(3), 2450-2459.

Kuonga, S., Ali, P., Eneya, L. & Kim, H. (2019). Unlinkable User Authenticated Key Agreement for Multi-Gateway Wireless Sensor Networks. *Current Analysis on Communication Engineering*, 2, 1-11.

Lee, S., Lee, J., Sin, H., Yoo, S., Lee, S., Lee, J., Lee, Y. & Kim, S. (2008). An energy-efficient distributed unequal clustering protocol for wireless sensor networks. *International Journal of Electronics and Communication Engineering*, 8(12), 2715-2719.

Lee, S. W. & Kim, H. (2014). Freshness Consideration of Hierarchical Key Agreement Protocol in WSNs. *International Journal of Security and Its Applications*, 8(1), 81-92.

[27] Li, C., Ye, M., Chen, G. & Wu, J. (2005). An energy-efficient unequal clustering mechanism for wireless sensor networks. *Proceedings on the IEEE International*

Conference on Mobile Adhoc and Sensor Systems Conference, Nov. 7, 2005 Washington DC, USA. IEEE.

Messerges, T. S., Dabbish, E. A. & Sloan, R. H. (2002). Examining smart-card security under the threat of power analysis attacks. *IEEE Transactions on Computers*, 51(5), 541-552.

MIRACL, https://www.miracl.com, accessed at July 10, 2018.

Nyang, D. & Lee, M. K.(2009). Improvement of Das's two-factor authentication protocol in wireless sensor networks. *IACR Cryptology ePrint Archive*, 631.

Pietro, R. D., Guarino, S., Verde, N. V. and Domingo-Ferrer, J. (2014). Security in wireless ad-hoc networks – A survey. *Computer Communications*, 51, 1-20.

Romer, K. & Mattern, F. (2004). The design space of wireless sensor networks. *IEEE Wireless Communications*, 11(6), 54-61.

Singh, V. K., Verma, S. & Kumar, M. (2016). Privacy Preserving In-network Aggregation in Wireless Sensor Networks. *Procedia Computer Science*, 94, 216-223.

Schneier, B. (1996). *Applied Cryptography*. John Wiley & Sons.

Song, T., Jung, J., Kang, D., Kim, H. & Won, D. (2017). *Cryptanalysis of an Authentication Scheme for Multi-Gateway Wireless Sensor Networks. Proceedings of the Twelfth International Conference on Digital Information Management*, Sep 12-14, 2017,Fukuoka, Japan, IEEE.

Srinivas, J., Mukhopadhyay, S. & Mishra, D. (2017). Secure and efficient user authentication scheme for multi-gateway wireless sensor networks. *Ad Hoc Networks*, 54, 147-169.

Turkanovic, M., Brumen, B. & Hölbl, M. (2014). A novel user authentication and key

agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the internet of things notion. *Ad Hoc Networks*, 20, 96-112.

Vaidya, B., Makrakis, D. & Mouftah, H. T. (2010). Improved two-factor user authentication in wireless sensor networks. *Proceedings on the IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications*, Oct. 11-13, 2010, ON, Canada, IEEE.

Vaidya, B., Makrakis, D. & Mouftah, H. (2012). Two-factor mutual authentication with key agreement in wireless sensor networks. *Security and Communication Networks*, 9(2), 171-183.

Wong, K. H. M., Zheng, Y., Cao, J. & Wang, S. (2006). A dynamic user authentication scheme for wireless sensor networks. *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, June 5-7, 2006*, Taichung, Taiwan, IEEE.

Xue, K., Hong, P. and Ma, C. (2014). A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture. *Journal of Computer and System Sciences*, 80(1), 195-206.

Xu, J., Liu, W., Lang, F., Zhang, Y. and Wang, C. (2010). Distance measurement model based on RSSI in WSN. *Wireless Sensor Network*, 2, 606-611.

Xu, S. & Wang, X. (2013). A new user authentication scheme for hierarchical wireless sensor networks. *International Review on Computers and Software*, 8(1), 197-203.

Yousefpoor, M. S. & Barati, H. (2019). Dynamic key management algorithms in wireless sensor networks: A survey. *Computer Communications*, 134, 52-69.