PRIVACY-PRESERVING MACHINE AUTHENTICATED KEY AGREEMENT FOR INTERNET OF THINGS

Beaton Kapito[1,3], Mwawi Nyirenda[1] and Hyunsung Kim[1,2]

[1]Department of Mathematical Sciences, University of Malawi, Zomba, Malawi
[2]School of Computer Science, Kyungil University, Kyungbuk, Korea
[3]Malawi Adventist University, Ntcheu, Malawi

ABSTRACT

Internet of things (IoT) is the integration of computer-based systems and the physical world in which things interact with each other. Due to heterogeneity and resource-constrained feature of IoT devices, there are many privacy and security challenges resulting in many threat vulnerabilities in IoT environments. After reviewing and analyzing the recent IoT security, privacy, and authentication protocols, we will withdraw research gaps focused on the elimination of human factors in IoT authentication. In order to fill these research gaps, this paper proposes a privacy-preserving machine authenticated key agreement based on IoT, denoted as IoT_{MAKA}. IoT_{MAKA} uses dynamic identity and machine fingerprint to provide security and privacy. Security analysis shows that IoT_{MAKA} provides anonymity and untraceability, provides freshness, and is secure against passive and active attacks. IoT_{MAKA} reduces communication overheads by 20% and computational overheads by 25% on average as compared to the previous related works.

KEYWORDS

Internet of Things, Authenticated Key Agreement, Privacy, Machine Authentication, Cryptography.

1. INTRODUCTION

With the advance in information communication technology, the Internet has developed to link computers around the world. The communication between and among physical things using the Internet is referred to as the Internet of things (IoT) [26, 34]. IoT involves extending Internet connectivity beyond standard devices to any range of traditionally non-Internet-enabled physical devices and everyday objects [1, 4, 13, 27, 38]. IoT objects collect and analyze data regularly to initiate action, providing intelligence for planning, decision making, and management [22]. Potential applications of the IoT are numerous and diverse, permeating into all areas of every-day life of individuals, enterprises, and society as a whole [5, 16, 31].

However, like all emerging technologies, IoT faces challenges that need to be overcome to ensure that the technology is successfully deployed on a large scale [41]. The challenge concerning privacy and security is of particular importance, as IoT technology unobtrusively collect information about the environment. The unique characteristics of IoT environments that make it vulnerable to many privacy and security challenges are: (i) IoT devices are usually resource-constrained, battery-driven, fault-prone systems and generally have lower processing power and memory. So fulfilling the requirements for implementing appropriate security and anonymization services is difficult because this requires a sufficient amount of processing and memory resources. Hence, IoT devices become an easy target for attackers [21]. (ii) IoT devices are heterogeneity. IoT integrates a multitude of various devices from different manufacturers,

software platforms and communication protocols. Any weak point could be a gate to leak data to attackers. (iii) Sensors and other devices are located everywhere, and therefore exposed to theft. Attackers use this increased physical accessibility of devices to extract information from them. (iv) IoT is ubiquitous and pervasive such that connected devices are worn, carried or seamlessly embedded in the world around us. These devices collect data, communicate and interact with other devices, without users' knowledge and permission thus compromising users' privacy [33]. (v) IoT has a dynamic characteristic because pervasive devices such as wearables join and leave the IoT network anytime. These features make the traditional information security measures insufficient for the IoT environment [7].

Other new challenges that come along with IoT due to its unique characteristics include high installation, operation, and maintenance costs, along with a host of environmental and integration challenges such as: (i) Interoperability-the functionality of various interconnected devices should not be prevented by relevant security solutions in IoT network system. (ii) Scalability-large numbers of nodes are there in an IoT network so a scalable security mechanism should be proposed for IoT. (iii) Autonomic control-IoT network must be spontaneous and devices must adapt themselves for some kind of self-management, self-configuring, self-healing, self-protecting, and self-optimizing [17]. Further, we also have security challenges such as: (i) Vulnerability in hardware and software. (ii) Easy exposure of devices to attackers. (iii) Physical damages by natural disasters such as earthquakes and fires. (iv) Cyber reconnaissance where the attacker uses cracking technique to access secret information or just sabotage the existing systems. (v) Brute force attacks on passwords and controlled attacks using denial of service [3, 6, 9, 11, 15, 29, 43].

Privacy threats of the IoT are more challenging because the data collection process is more passive, more pervasive, and less intrusive. This feature results in users being unaware that they are being tracked [17, 20]. Major privacy threats in IoT are (i) Identification-the threat of associating an identifier with private data about an individual [21]. (ii) Localization and tracking-the threats of determining and recording a person's location through time and space by using Internet traffic. (iii) The profiling-the practice of collecting and processing data about individuals' activities such as sites visited, pages viewed, and emails sent over long periods in order to categorize them according to some key features [46]. (iv) Linkage-the disclosure of information due to a combination of separated data sources and linking different systems. This occurs because aggregating information creates synergies [4].

Anonymity and untraceability play an important role in ensuring privacy. Anonymity is the concept of decoupling or removing the connection to a particular entity from the data collected. Untraceability is the idea of completely removing an item or piece of data from the digital world [41]. Due to the resource-constrained feature of IoT environments, the challenge is how to establish a shared cryptographic key in a secure manner, between the IoT device and the service server (*SS*). Mutual authentication [2, 32, 36] is required for such a scenario because all communicating entities need to be sure of the legitimacies of the other entities involved. This necessitates the use of the central server (*CS*) as a link and trusted third party.

In 2006, Wong et al. presented a user authentication in IoT and it spawned many subsequent kinds of research in authentication protocols over IoT [10, 12, 18, 23, 28, 32, 37, 45]. After reviewing Wong et al.'s protocol, Das found out that Wong et al.'s protocol was vulnerable to attacks such as many logged-in users with the same login-ID attack, stolen verifier attack, etc. [12]. In order to improve Wong et al.'s protocol, Das proposed an authentication protocol for wireless sensor networks (WSNs) in IoT using the smart card. Das's work preceded many subsequent reviews. The works of Khan and Alghathbar, He et al. and Yeh et al. found that Das's protocol had lack of features including key agreement, user anonymity and mutual authentication

and also was prone to password guessing, gateway bypassing, denial-of-service (DoS) and sensor node capture attacks [18, 25]. Vaidya et al. showed that Khan and Alghathbar's protocol was also vulnerable to several security attacks and proposed an improved version of Khan and Alghathbar's protocol [37]. Their protocol provided security features like mutual authentication, password protection, key agreement and resilience against several attacks. In 2011, Yeh et al. proposed two factor authentication protocols for WSNs by using elliptic curve cryptography (ECC) [45]. They chose ECC to provide security features with higher efficiency on computational overheads as compared with the other protocols. After reviewing Yeh et al.'s protocol, Shi and Gong found out that the protocol failed to achieve mutual authentication contrary to their claim, further to this, they also pointed out that Yeh et al.'s protocol does not support the features of key agreement and user anonymity [32]. Motivated by the weaknesses of Yeh et al.'s protocol, Shi and Gong presented an improved ECC-based authentication protocol for WSNs, which they claimed was efficient and provided more features than Yeh et al.'s protocol [32]. Nevertheless, Choi et al. reviewed Shi and Gong's protocol and found out that it is prone to stolen smart card and unknown key share attacks [10]. They consequently presented an enhanced protocol for WSNs. Xue et al. reviewed Choi et al.'s protocol and improved it by proposing a user authentication protocol for WSNs using temporal credentials [44]. The protocol has high efficiency due to the use of hash function and exclusive-OR operations only as compared to ECC or Rivest, Shamir, and Adleman. Despite the claimed superiority of Xue et al.'s protocol, He et al. reviewed the protocol and found that it is weak against impersonation, modification, and off-line password guessing attacks [18]. He et al. presented an improved protocol to remedy shortcomings of Xue et al.'s protocol. Jiang et al. reviewed He et al.'s protocol and found that it was prone to tracking, user impersonation and stolen smart card attacks [23]. While maintaining all the virtues of He et al.'s protocol, Jiang et al. proposed an untraceable user authentication protocol using ECC. In their protocol, user and gateway node performed ECC point multiplication operations, while sensor node just needed hash function operations [23]. However, Li et al. reviewed the protocols of Xue et al., He et al., and Jiang et al. and found some common weaknesses of all the three previous protocols. The common weaknesses of the three protocols were that all of them lack wrong password detection and password change mechanisms [28]. Secondly, all the three protocols are unsuited for IoT environments because the user exchanges messages directly with sensor nodes instead of passing through a gateway node. Lastly, all the three protocols are vulnerable to known session-specific temporary information attack and clock synchronization problem. In order to address the weaknesses of the three protocols above, Li et al. proposed a three-factor anonymity authentication protocol for WSNs in IoT environments by using user biometrics. Li and Niu also adopted fuzzy commitment scheme and error correcting codes to help in handling the user's biometric information [28]. After analysis and comparison with the three protocols, they show that their protocol is most efficient computationally. Furthermore, they argued that their protocol also achieves more security and functional features comparatively. However, our critical analysis finds that Li and Niu's protocol is still not suitable for IoT environments as they claimed because it is dependent on human involvement in its authentication process. We also observe that the communication and computational overhead costs can reduce further.

After the literature review on authentication, privacy, and security in IoT, we found three main research gaps. The first gap is the necessity to eliminate the human factor in IoT authenticated key agreement (AKA). We will propose machine biometrics in form of machine fingerprint as an authenticating factor. The second gap is to propose a protocol that prioritizes privacy provision instead of always prioritizing security, which results in compromising the privacy of communicating entities. The third gap is reduction of computation and communication overheads to improve on the efficiency of authentication protocols. Therefore, this paper dares to fill these three research gaps by proposing a new privacy-preserving machine AKA (MAKA) for IoT denoted as IoT_{MAKA}. IoT_{MAKA} will consider two privacy goals and four security goals for the AKA

in IoT. The two privacy goals are anonymity and untraceability and four additional security goals are mutual authentication, session key agreement, the freshness of message and provision of security against active and passive attacks. IoT_{MAKA} achieves efficiency by minimizing computational and communicational overhead.

2. BACKGROUND

This section provides network configuration and some cryptographic concepts used in this paper where we consider privacy-preserving AKA for IoT. First, we discuss the network configuration together with the roles of each of three entities used in the protocol. Furthermore, we review security and privacy basis of the proposed protocol, which are; symmetric key cryptosystem, one-way hash function, challenge response mechanism and fuzzy commitment scheme. After that, we will do a literature review of related works on privacy, security and AKA in IoT.

2.1. Network Model

The targeting network environment is a machine-to-machine IoT environment, which requires a form of data communication that involves three entities that do not require human interaction or intervention in the process of communication. In this paper, we aim at privacy-preserving architecture in IoT. Since IoT uses the internet, then we need to use smart gargets, represented by an IoT device in our protocol. The IoT device will have an active role in initiating communication by sending requests to a service server through a central server. This IoT device must have knowledge of available service providers and the services they provide. The need of privacy necessitates the use of a trusted *CS* that authenticates all entities in a network. We also use *CS* in order to save energy as per the research findings in Heinzelman et al., they found out that energy consumption of nodes in IoT network is directly proportional to the distance between them, so using *CS* between IoT device and *SS* minimizes energy consumption [19]. Communications in IoT are about seeking for services after one undergoes authentication by *CS*. Therefore, there must be an entity to offer these services. *SS* represents this entity. The environment consists of an IoT device with sensors and a memory chip (*MC*), a *CS,* and a *SS* as shown in Figure 1.

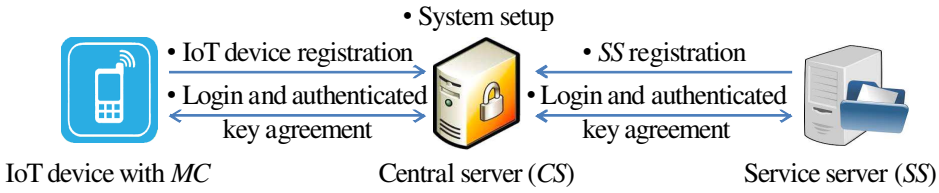


Figure 1. Network configuration

The roles of each entity are:

- IoT device with *MC*: It consists of software and hardware for generating sensing data, computing meta-information, sending reports, receiving instructions, and acting accordingly. The main role of the IoT device is to collect data and send the data to *SS* through *CS* or directly to *CS* so that *SS* can take the necessary actions in real time. IoT device comes with some sensors and an *MC*. The sensors collect environmental data required for the target services. *MC* is for secure data storage and acting like a smart card in IoT device.

- *CS*: It consists of software and hardware for identification and credentials, it stores unique identification and secret credentials such as keys. This is a fully trusted server responsible for login and AKA between IoT device and *SS*. It is responsible for system setup, authentication, and key agreement of IoT device and *SS*. It facilitates communication and data exchanges between the IoT device and *SS*.

- *SS*: It consists of software and hardware for receiving instructions, and acting accordingly. This supports various rich and convenient services to the IoT device. However, *SS* does not have credentials to communicate with IoT device directly but through *CS*. For the security and privacy reasons, *SS* does not communicate directly with IoT device even after successful authentication from *CS*

The relationship of the entities is that when IoT device collects data and sends it to *SS* through *CS* for appropriate action and service. *CS* authenticates both the IoT device and *SS*. This authentication assures both IoT device and *SS* that they are communicating with a legally acceptable entity not an adversary in a masquerading attack.

2.2. Cryptographic Basic Functions

This section discusses some mathematical preliminaries used as the security basis of the proposed protocol. They are one-way hash function, fuzzy commitment scheme, challenge-response, and symmetric key cryptography.

[**One-way Hash Function**]. A one-way hash function $h(.)$: $\{0, 1\}^* \rightarrow \{0, 1\}^n$ is an algorithm where $\{0, 1\}^*$ and $\{0, 1\}^n$ denote binary strings of arbitrary length, *, and fixed length, n, respectively [39]. It takes an arbitrary length binary string $x \in \{0, 1\}^*$ as input but outputs a binary string of fixed length n, say $y \in \{0, 1\}^n$ such that $y = h(x)$ and is called digest or hash value. This is a function for which finding the inverse of any random input is computationally infeasible [21].

[**Fuzzy Commitment Scheme**] A fuzzy commitment scheme is cryptographic primitive that allows one to commit a chosen value while keeping it hidden to others with the ability to reveal the committed value later [24]. The committed value is binding thus cannot be changed by either party. Suppose $h(.)$: $\{0, 1\}^* \rightarrow \{0, 1\}^n$ is a secure hash function which can commit a code word $c \in C$ using an n bit witness y as $F(c, y) = \{\alpha, \delta\}$, where $\alpha = h(c)$ and $\delta = y \oplus c$. The commitment $F(c, y) = \{\alpha, \delta\}$ can be opened using witness y', which is relatively close to y, but no need to be the same as y [24, 28]. To open the commitment using y', the receiver computes $c' = f(y' \oplus \delta) = f(c \oplus (y' \oplus y))$, and checks whether $\alpha = h(c')$. If they are equal, the commitment is successfully open [35]. Otherwise, the witness y' is not valid. This paper uses a fuzzy commitment scheme due to the noisy characteristic of biometrics (i.e. the input the biometric information is not exactly the same as the template). In this scenario, biometric template can be seen as the witness y, and c can be opened by the input biometric y', which is close to y.

[**Symmetric Key Cryptography**] Symmetric key cryptography is a cryptographic algorithm in which the sender and receiver of a message share a single, common key that encrypts and decrypts the message. Advanced encryption standard (AES) is a standard, which has a variable key length of 128, 192, or 256 bits [40]. In this paper, we will use AES with a 128-bit key for our symmetric key cryptography. The use of symmetric key cryptography is to provide privacy of IoT device to *CS* based on dynamic identity (*DID*).

3. PRIVACY-PRESERVING MACHINE AUTHENTICATED KEY AGREEMENT

The purpose of this section is to propose a new privacy-preserving machine authenticated key agreement, (IoT_{MAKA}) in IoT environments. The security and privacy of IoT_{MAKA} relies on symmetric-key cryptography, hash function, and fuzzy commitment scheme. IoT_{MAKA} emphasizes on three features, which are; it relies on machine factor authentication, it provides privacy-preserving and efficiency in communication and computational overheads. First, we define a new concept of machine factor for authentication, which removes human factor to fit it to IoT environment. Based on the machine fingerprint, IoT_{MAKA} will be designed.

3.1. Basic Features for Designing IoT_{MAKA}

This sub-section discusses the basic requirement for machine-oriented AKA for IoT environments. It first compares and contrasts machine oriented authentication factors and human-oriented authentication factors. Next, we look at IoT device fingerprinting and lastly look at authentication of IoT devices by using device fingerprinting. After that, we discuss the design goals of IoT_{MAKA} by considering of IoT features.

[**Machine authentication factors**] Machine to machine communication is a form of data communication that involves one or more entities that do not necessarily require human interaction or intervention in the process of communication. It should be different from human to human or human to machine communication models. Human-oriented authenticating factors include what one can remember such as password, what one is such as biometrics and what one has such as smart card. On the other hand, machine authenticating factors only include two factors: what machine is like machine fingerprint and what machine has like memory chip. Machines authentication factors do not include passwords because machines cannot remember anything. Table 1 shows a comparison of authentication factors between human-oriented and machine-oriented factors.

Table 1. Authentication factors comparison.

Human-oriented factors	Machine-oriented factors
· What you remember - password · What you are - biometric feature · What you have - smart card	· What the machine is - machine fingerprint · What machine has - memory chip

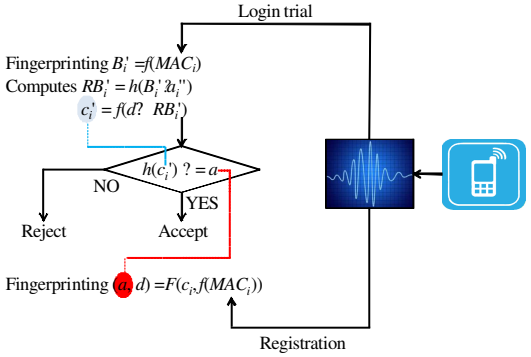


Figure 2. Machine fingerprinting using a fuzzy commitment scheme

[**Machine Fingerprinting**] Machine fingerprinting is a technique to authenticate devices using unique features extracted from the machines' distinctive characteristics. We use a machine's radio frequency (RF) emission as their fingerprint for the authentication of the device [11, 30]. Authentication of devices is one of the main security concerns in the IoT for device authentication. IoT_{MAKA} uses a fuzzy commitment scheme as the basic machine fingerprinting operation. $F(\{0, 1\}^n, \{0, 1\}^n) \rightarrow (\{0, 1\}^l, \{0, 1\}^n)$ is a fuzzy commitment scheme, which can commit a codeword $c \in C$ using an n bit witness y as $F(c, y) = (\alpha, \delta)$, where $C \in \{0, 1\}^n$, $\alpha=h(c)$ and $\delta = y \oplus c$. The commitment $F(c, y) = (\alpha, \delta)$ can be opened using witness y', which is relatively close to y, but no need to be the same as y. To open the commitment using y', the receiver computes $c' = f(y' \oplus \delta) = f(c \oplus (y' \oplus y))$, and checks whether α is equal to $h(c')$. If they are equal, the commitment opens successfully. Otherwise, the witness y' is not valid. Here, $F(.)$ is the commitment scheme while $f(.)$ is a decoding function of the commitment scheme. Therefore, IoT_{MAKA} uses a fuzzy commitment function $F(.)$ at registration of machine fingerprint while the decoding $f(.)$ is for the login trials as shown in Figure 2.

As shown in Figure 2, we use an RF signal from the IoT device as an input of the fuzzy commitment, expressed as $F(c_i, f(MAC_i))$ and $f(MAC_i)$, respectively. The media access control (MAC) address of a device is a unique identifier assigned to a network interface controller. For communications within a network segment, we use a device's MAC address as its network address (MAC_i). Note that $f(MAC_i)$ is not applying MAC_i to the function $f(.)$ but means to extract the distinctive feature of IoT device with MAC_i's RF signal. For IoT devices, CS chooses $\alpha = h(c_i)$ and $\delta = c_i \oplus RB_i$ where $c_i \in C$ stored in the database of CS. On IoT device login trial, MC checks the machine fingerprint $B_i' = f(MAC_i)$ and MC derives $a_i' = G_i \oplus B_i'$, computes $RB_i' = h(B_i' \| a_i'')$ and $c_i' = f(\delta \oplus RB_i') = f((c_i \oplus RB_i) \oplus RB_i')$, and checks the validity of $h(c_i')^? = \alpha$. If the validation is not successful, MC rejects the trial. Otherwise, the ownership check is accepted.

3.2. Design Goal

Our design goal is to propose a new AKA protocol for IoT, which preserves privacy and provides required security on IoT based on the predefined machine factors. Specifically, this paper dares to achieve the following objectives.

[***Privacy Goal* 1**] *Anonymity*: The property, which prevents an attacker who has recorded past communications from discovering the identities of the participants. Although achieving anonymity can be an important design criterion in cryptographic systems, it comes at a cost. Our goal is to develop mathematical techniques that enable anonymity in IoT_{MAKA} without compromising security.

[***Privacy Goal* 2**] *Untraceability*: The ability to disable the adversary from tracing the source of captured data or unable to be found or discovered. It means an adversary cannot tell what messages belong to what entity.

[***Security Goal* 1**] *Mutual authentication*: Mutual authentication, also called two-way authentication, is a process or technology in which both entities in a communications link authenticate each other. In a network environment, the client authenticates the server and vice-versa. In this way, the network assures users of exclusively communicating with legitimate entities and servers. Mutual authentication is gaining acceptance as a tool that can minimize the risk of online fraud.

[***Security Goal* 2**] *Session Key agreement*: A key agreement is one of a key establishment technique in which an agreed key is derived by two (or more) parties as a function of information contributed by, or associated with, each of these, such that no party can predetermine the

resulting value. In this process, the key generation is done in a collaborative manner, resulting in both parties having the key.

[***Security Goal* 3**] *Freshness of message*: This is to provide certainty to a protocol of detection of replayed messages in the replay attack. Therefore, freshness is the assurance that the received message is most recent or fresh, not replayed old message. The provision of freshness helps to counter replay attacks.

[***Security Goal* 4**] *Provide Security against passive and active attacks*: The protocol should be secure against both active and passive attacks.

Table 2. Notations.

Notation	Description
CS, SS	Central server and service server
MAC_i	Media access control address of IoT device i
MC	Secure memory chip of i
ID_{CS}, ID_{SS}	Identities of CS and SS
DID_i	Dynamic identity of i
K_{CS}	Master secret key of CS
$K_{CS\text{-}SS}$	Secret key established between CS and SS
SK	Session key established between entities
a_i, r_i, r_{CS}, r_{SS}	Random numbers
$h(.)$	One way hash function
$\gamma(.)$	Fuzzy commitment
$f(.)$	Machine fingerprint mechanism
$E_K(.)$, $D_K(.)$	Symmetric encryption and decryption with the key K
$\oplus$	Exclusive OR operation
$\|$	Message concatenation operation

3.3. IoT_{MAKA}-The Proposed Protocol

IoT_{MAKA} uses machine fingerprint as an authentication factor based on IoT device's RF signal. We adopt a fuzzy commitment scheme to verify the validity of the machine fingerprint. IoT_{MAKA} has three parties, which are the IoT device with MC, CS and SS. IoT_{MAKA} has four phases including system setup, IoT device registration, service server registration and login and AKA.

[**System Setup**] System setup of IoT_{MAKA} is the preliminary system arrangement to ensure that the execution of the protocol completes successfully. Before the execution of IoT_{MAKA}, some parameters should be defined by CS as follows; First, a group Z_n is selected where n is very large for maximum security, and a code set $C \in \{0, 1\}^{n.}$ Then CS generates a random number $K_{CS} \in Z_n$ as its private key and defines a hash functions $h(.)$, two fuzzy commitment functions $F(.)$ and $f(.)$ and two asymmetric key functions $E(.)$ for encryption and $D(.)$ for decryption based on AES. Next, CS publishes the parameters $\{Z_n, h(.), f(.), F(.), E(.), D(.)\}$ to the targeted network.

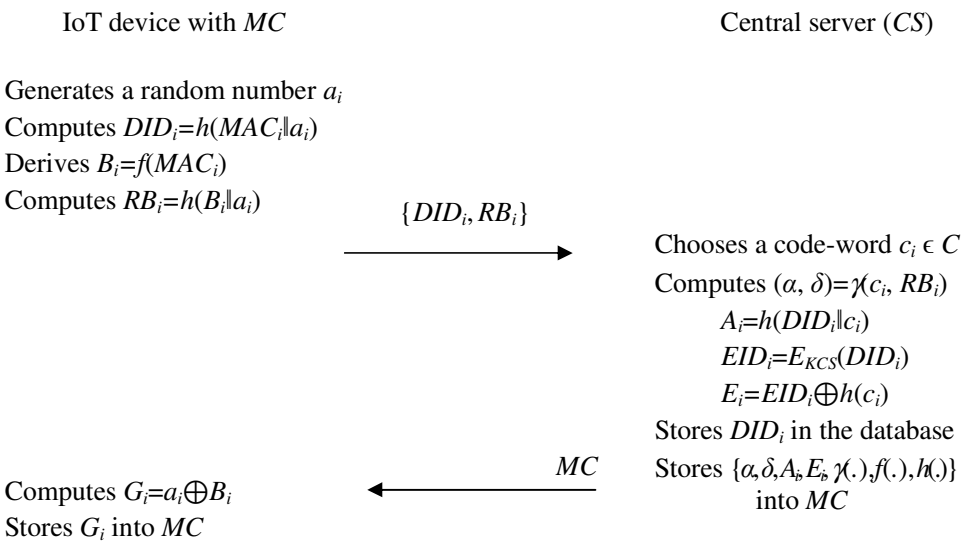


Figure 3. IoT device registration phase of IoT_{MAKA}

[**IoT Device Registration**] Before an IoT device communicates with *SS*, it needs to register with *CS* so that it becomes part of the system. Figure 3 shows the flow of this phase and the description of this phase is as follows:

R1. IoT device generates a random number a_i, and computes an amplified dynamic identity $DID_i = h(MAC_i \| a_i)$. After that, it derives its machine fingerprint $B_i = f(MAC_i)$, amplifies the fingerprint to become $RB_i = h(B_i \| a_i)$ and then sends the registration message $\{DID_i, RB_i\}$ to *CS* via a secure channel.

R2. Upon receipt of the registration request from IoT device, *CS* chooses a random code word $c_i \in C$ for IoT device and computes $(\alpha, \delta) = F(c_i, RB_i)$, where $\alpha = h(c_i)$ and $\delta = c_i \oplus RB_i$. Then, *CS* computes $A_i = h(DID_i \| c_i)$, $EID_i = E_{KCS}(DID_i)$ and $E_i = EID_i \oplus h(c_i)$ where $E(.)$ is the symmetric encryption function and K_{CS} is the master secret key of *CS*. After that, *CS* stores $\{\alpha, \delta, A_i, E_i, F(.), f(.), h(.)\}$ into a secure *MC* and sends it to IoT device via a secure offline manner. Finally, *CS* stores DID_i in its database and deletes the other information.

R3. After *MC* installation, IoT device computes $G_i = a_i \oplus B_i$ and stores G_i into *MC*. Now *MC* contains parameters $\{\alpha, \delta, A_i, E_i, G_i, F(.), f(.), h(.)\}$.

[**Service Server Registration**] Like IoT device, *SS* also needs to register with *CS* before providing any service to the IoT device. *SS* selects an identifier ID_{SS} and sends it to *CS* via secured channel, which is established based on the pre-relationship with *CS*. After receiving the registration request from *SS*, *CS* computes a secret key $K_{CS\text{-}SS} = h(ID_{SS} \| K_{CS})$ for *SS*. *CS* sends $\{ID_{SS}, K_{CS\text{-}SS}\}$ to *SS* securely.

[**Login and AKA**] Login is the act of logging into a system to get some services. IoT_{MAKA} uses the scenario that IoT device requests services to *SS*, while *CS* checks the authenticity of the

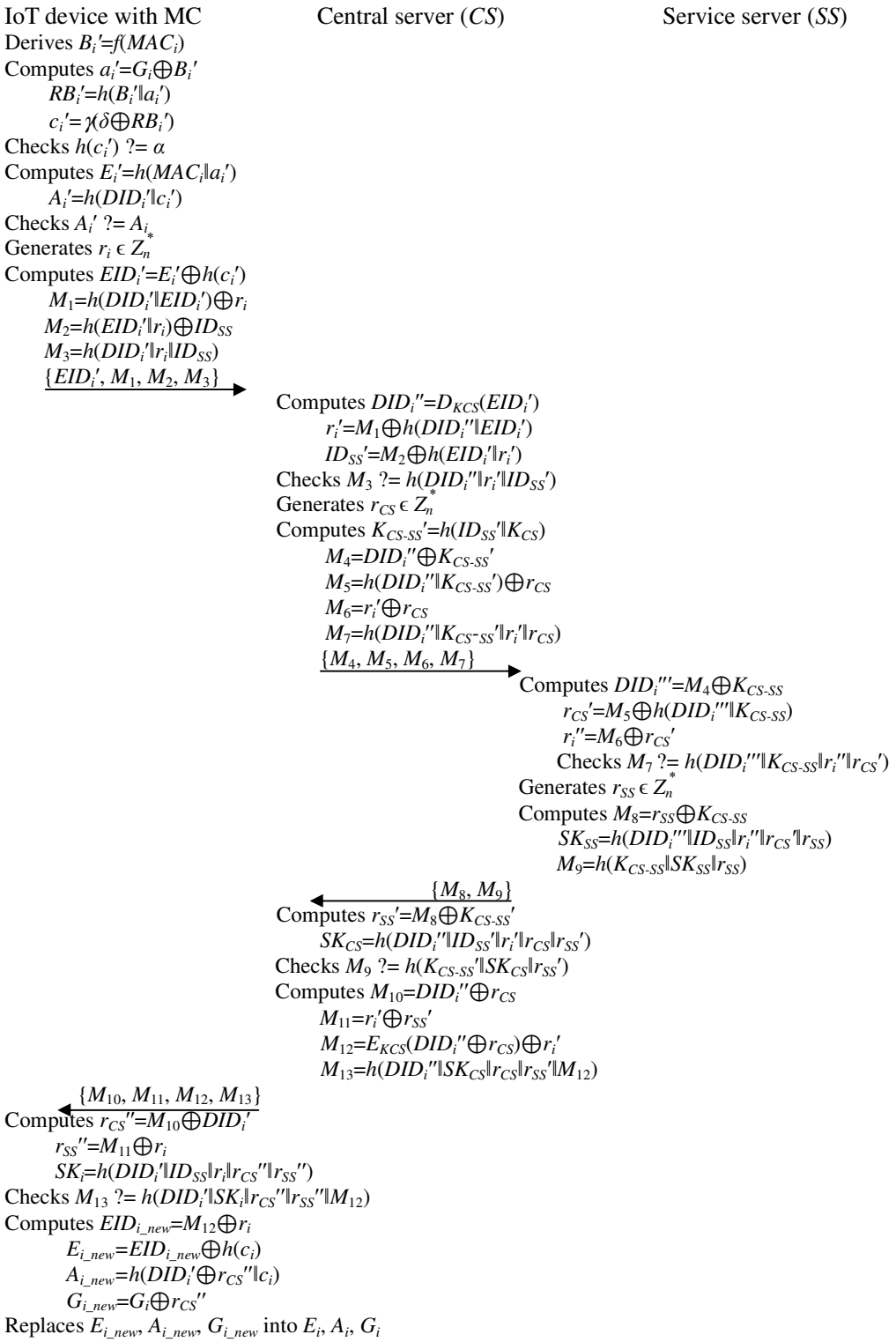


Figure 4. Login and AKA phase of IoT_{MAKA}

IoT device before giving it authority to *SS*. When the IoT device wants to access or send data to *CS* or *SS*, IoT device should pass the ownership check by *MC* first. Figure 4 shows the detailed conceptual flow of this phase as follows:

A1. IoT device imprints its machine fingerprint $B_i' = f(MAC_i)$ and *MC* derives $a_i' = G_i \oplus B_i'$, computes $RB_i' = h(B_i' \| a_i'')$ and $c_i' = f(\delta \oplus RB_i')$, and checks $h(c_i')^? = \alpha$. *MC* terminates the session if they are not equal. Otherwise, IoT device passes the fingerprint verification and *MC* computes $E_i' = h(MAC_i \| a_i')$ and $A_i' = h(DID_i' \| c_i')$, and checks $A_i'^? = A_i$. *MC* terminates the session if they are not equal. Otherwise, IoT device is verified by *MC*. *MC* chooses a random number r_i and c $\in Z_n$, and computes $EID_i' = E_i \oplus h(c_i')$, $M_1 = h(DID_i' \| EID_i') \oplus r_i$, $M_2 = h(EID_i' \| r_i) \oplus ID_{SS}$, and $M_3 = h(DID_i' \| r_i \| ID_{SS})$. After that, *MC* sends the login request message $\{EID_i', M_1, M_2, M_3\}$ to *CS*.

A2. On receiving the login request, *CS* computes $DID_i'' = D_{KCS}(EID_i')$, $r_i' = M_1 \oplus h(DID_i'' \| EID_i')$ and $ID_{SS}' = M_2 \oplus h(EID_i' \| r_i')$, and checks $M_3^? = h(DID_i'' \| r_i' \| ID_{SS}')$. *CS* terminates the session if they are not equal. Otherwise, *CS* generates a random number r_{cs}, and computes $K_{CS\text{-}SS}' = h(ID_{SS}' \| K_{CS})$, $M_4 = DID_i'' \oplus K_{CS\text{-}SS}'$, $M_5 = h(DID_i'' \| K_{CS\text{-}SS}') \oplus r_{cs}$, $M_6 = r_i' \oplus r_{cs}$ and $M_7 = h(DID_i'' \| K_{CS\text{-}SS}' \| r_i' \| r_{cs})$. After that, *CS* sends the message $\{M_4, M_5, M_6, M_7\}$ to *SS*.

A3. On receiving the message, *SS* computes $DID_i''' = M_4 \oplus K_{CS\text{-}SS}$, $r_{cs}' = M_5 \oplus h(DID_i''' \| K_{CS\text{-}SS})$ and $r_i'' = M_6 \oplus r_{CS}'$, and checks $M_7^? = h(DID_i''' \| K_{CS\text{-}SS} \| r_i'' \| r_{CS}')$. *SS* terminates the session if the equation does not hold. Otherwise, *SS* generates a random number r_{ss}, and computes $M_8 = K_{CS\text{-}SS} \oplus r_{ss}$, $SK_{SS} = h(DID_i''' \| ID_{SS} \| r_i'' \| r_{CS}' \| r_{SS})$ and $M_9 = h(K_{CS\text{-}SS} \| SK_{SS} \| r_{SS})$. *SS* responds to *CS* with the message $\{M_8, M_9\}$.

A4. After getting the message from *SS*, *CS* computes $r_{ss}' = M_8 \oplus K_{CS\text{-}SS}'$ and $SK_{CS} = h(DID_i'' \| ID_{SS}' \| r_i' \| r_{CS} \| r_{SS}')$, and checks $M_9^? = h(K_{CS\text{-}SS}' \| SK_{CS} \| r_{SS}')$. *CS* terminates the session is rejected if they are not equal. Otherwise, *CS* computes $M_{10} = DID_i'' \oplus r_{CS}$, $M_{11} = r_i' \oplus r_{SS}'$, $M_{12} = E_{KCS}(DID_i'' \oplus r_{CS}) \oplus r_i'$ and $M_{13} = h(DID_i'' \| SK_{CS} \| r_{CS} \| r_{SS}' \| M_{12})$. Finally, *CS* sends the message $\{M_{10}, M_{11}, M_{12}, M_{13}\}$ to IoT device. A5. Upon receiving the message from *CS*, *MC* computes $r_{cs}'' = M_{10} \oplus DID_i'$, $r_{ss}'' = M_{11} \oplus r_i$ and $SK_i = h(DID_i' \| ID_{SS} \| r_i \| r_{CS}'' \| r_{SS}'')$, and checks $M_{13}^? = h(DID_i' \| SK_i \| r_{CS}'' \| r_{SS}'' \| M_{12})$. *MC* terminates the session if they are not equal. Otherwise, the authentication process is completed. Only if the process is successful, *MC* computes $EID_{i\,new} = M_{12} \oplus r_i$, $E_{i\_new} = EID_{i\_new} \oplus h(c_i')$, $A_{i\_new} = h(DID_i' \oplus r_{CS}'' \| c_i')$ and $G_{i\_new} = G_i \oplus r_{CS}''$, and replaces E_{i_new}, A_{i_new} and G_{i_new} into E_i, A_i and G_i, respectively. Finally, IoT device can access *SS* on *MC* for any communication via *CS*, and a session key $SK_i = (SK_{CS} = SK_{SS})$ is shared among IoT device, *CS* and *SS*.

4. SECURITY AND PERFORMANCE ANALYSIS

This section provides security and performance analysis. We base the analyses on Dolev and Yao threat model [14]. The purpose of this analysis is to show IoT_{MAKA} achieves design goals defined in subsection 3.2. Furthermore, performance analysis will show that IoT_{MAKA} has good aspects of computational and communicational overheads as compared to recent related protocols.

4.1. Security Analysis

In this subsection, we will discuss the security and privacy threat model according to [14]. We do formal analysis according to BAN logic [8]. The informal analysis is by cryptanalysis by using design goals in subsection 3.2 and lastly, we discuss the security features analysis.

4.1.1. Threat Model

A threat model is an imperative module of the designing of an AKA protocol. The threat model is a process for enhancing security by classifying vulnerabilities and objectives and then defining preventive measures of threats to the system. In this framework, a threat is a potential malicious attack from an adversary that can cause damage to the assets. We base the threat model on the following assumptions;

- Any IoT device may be corrupted and turned into a device controlled by the adversary. We refer to this as a malicious device. We assume that all cryptographic keys of the malicious device are known to the adversary
- An adversary can extract the information from *MC* or any device by examining power consumption and leaked information
- An adversary is able to eavesdrop on all the communications between the entities involved in the communication chancel over a public channel
- An adversary has the potential to modify a message, delete, redirect and resend the eavesdropped transmitted messages
- An adversary can be a legal user or an outsider in any system
- An adversary can guess low entropy secret and identity individually easily but guessing two secret parameters is computationally infeasible in polynomial time
- It is assumed that the protocol used in the AKA system is known to the attacker
- Kerckhoffs's principle: A cryptosystem should be secure even if everything about the system, except the session key, is public knowledge [14].

4.1.2. Formal Analysis

In this subsection, we analyze IoT_{MAKA} using BAN logic. BAN logic analyses protocols by using axioms to verify message origin, message freshness and trustworthiness of the origin of the message [8]. We use the following notations in formal security analysis using the BAN logic:

- $Q \models X$: Principal Q believes the statement X.
- $\#(X)$: Formula X is fresh.
- $Q \Rightarrow X$: Principal Q has jurisdiction over the statement X.
- $\overset{K}{\mapsto} Q$: Principal Q has a public key K.
- $Q \triangleleft X$: Principal Q sees the statement X.
- $Q \sim X$: 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 with each other. The session key SK is good, in that any principal except Q and R. will never discover it.

In addition, we use the following BAN logic rules to prove that IoT_{MAKA} provides a secure mutual authentication among IoT device, *CS* and *SS*:

1. **Message-meaning rule**: $\dfrac{R \mid \equiv R \overset{Y}{\leftrightarrow} S, \quad R \triangleleft \langle X \rangle_Y}{R \mid \equiv S \mid \sim X}$

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

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

4. **Freshness rule**: $\frac{R|\equiv\ \#(X)}{R|\equiv\ \#(XY)}$

In order to show that IoT_{MAKA} provides secure mutual authentication among IoT device with *MC*, *CS* and *SS*, we need to achieve the following goals:

Goal 1: $\text{IoT device}|\equiv(\text{IoT device} \overset{SK}{\leftrightarrow} SS)$

Goal 2: $SS|\equiv(SS \overset{SK}{\leftrightarrow} \text{IoT device})$

Goal 3: $\text{IoT device}|\equiv SS|\equiv(SS \overset{SK}{\leftrightarrow} \text{IoT device})$

Goal 4: $SS|\equiv\text{IoT device}|\equiv(\text{IoT device} \overset{SK}{\leftrightarrow} SS)$

Idealized form: The arrangement of the transmitted messages among IoT device with *MC*, *CS* and *SS* in IoT_{MAKA} to the idealized forms is as follows:

Message 1. $\text{IoT device} \rightarrow CS$: $<EID_i'>_{KCS}$, $<M_1>_{KCS}$, $<M_2>_{KCS}$, $<M_3>_{KCS}$
Message 2. $CS \rightarrow SS$: $<M_4>_{KCS\text{-}SS}$, $<M_5>_{KCS\text{-}SS}$, M_6, $<M_7>_{KCS\text{-}SS}$
Message 3. $SS \rightarrow CS$: $<M_8>_{KCS\text{-}SS}$, $<<M_9>_{KCS\text{-}SS}>_{SK}$
Message 4. $CS \rightarrow \text{IoT device}$: M_{10}, M_{11}, $<M_{12}>_{KCS}$, $<<M_{13}>_{KCS}>_{SK}$

Assumptions: The following are the initial assumptions of IoT_{MAKA}:

A1: $\text{IoT device}|\equiv\#(r_i, a_i)$
A2: $CS|\equiv\#(r_{cs})$
A3: $SS|\equiv\#(r_{ss})$
A4: $\text{IoT device}|\equiv(\text{IoT device} \overset{(K_{CS})}{\longleftrightarrow} CS)$
A5: $CS|\equiv(CS \overset{(K_{CS})}{\longleftrightarrow} \text{IoT device})$
A6: $CS|\equiv(CS \overset{K_{CS-SS}}{\longleftrightarrow} SS)$
A7: $SS|\equiv(SS \overset{K_{CS-SS}}{\longleftrightarrow} CS)$
A8: $\text{IoT device}|\equiv SS|\Longrightarrow \text{IoT device} \overset{SK}{\leftarrow} SS$
A9: $SS|\equiv\text{IoT device}|\Longrightarrow SS \overset{SK}{\leftarrow} \text{IoT device}$

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. $CS \triangleleft <EID_i'>_{KCS}$, $<M_1>_{KCS}$, $<M_2>_{KCS}$, $<M_3>_{KCS}$
According to assumption A5 and the message-meaning rule, we get:
Step 2. $CS|\equiv\text{IoT device}|\sim(<EID_i'>_{KCS}, <M_1>_{KCS}, <M_2>_{KCS}, <M_3>_{KCS})$
According to assumption A1 and the freshness concatenation rule, we get:
Step 3: $CS|\equiv\#(<EID_i'>_{KCS}, <M_1>_{KCS}, <M_2>_{KCS}, <M_3>_{KCS})$
According to Step 2, Step 3 and the nonce verification rule, we get:

Step 4. $CS|\equiv \mathbf{IoT\ \ device}|\equiv(<EID_i'>_{KCS}, <M_1>_{KCS}, <M_2>_{KCS}, <M_3>_{KCS})$
According to Step 4, assumption A4 and the believe rule, we get:
Step 5. $CS|\equiv \mathbf{IoT\ \ device}|\equiv(\mathbf{IoT\ \ device} \overset{(K_{CS})}{\longleftrightarrow} \boldsymbol{CS})$
According to the jurisdiction rule, we get:
Step 6. $CS|\equiv(\boldsymbol{CS} \overset{(K_{CS})}{\longleftrightarrow} \mathbf{IoT\ \ device})$
Based on Message 2, we derive
Step 7. $SS \triangleleft <M_4>_{KCS\text{-}SS}, <M_5>_{KCS\text{-}SS}, M_6, <M_7>_{KCS\text{-}SS}$
According to assumption A7 and the message-meaning rule, we get:
Step 8. $SS|\equiv CS|\sim<M_4>_{KCS\text{-}SS}, <M_5>_{KCS\text{-}SS}, M_6, <M_7>_{KCS\text{-}SS})$
According to assumption A2 and the freshness concatenation rule, we get:
Step 9: $SS|\equiv\#(<M_4>_{KCS\text{-}SS}, <M_5>_{KCS\text{-}SS}, M_6, <M_7>_{KCS\text{-}SS})$
According to Step 8, Step 9 and the nonce verification rule, we get:
Step 10. $SS|\equiv CS|\equiv(<M_4>_{KCS\text{-}SS}, <M_5>_{KCS\text{-}SS}, M_6, <M_7>_{KCS\text{-}SS})$
According to Step 10, assumption A6 and the believe rule, we get:
Step 11. $SS|\equiv CS|\equiv(\boldsymbol{CS} \overset{K_{CS-SS}}{\longleftrightarrow} \boldsymbol{SS})$
According to the jurisdiction rule, we get:
Step 12. $SS|\equiv(\boldsymbol{SS} \overset{K_{CS-SS}}{\longleftrightarrow} \boldsymbol{CS})$
According to Step 8, Step 9, Step 10 and the nonce verification rule, we conclude:
Step 13. $SS|\equiv \mathbf{IoT\ \ device}|\equiv(\mathbf{IoT\ \ device} \overset{SK}{\leftarrow} \boldsymbol{SS})$ **(Goal 4)**
According to assumption A8 and the jurisdiction rule, we get:
Step 14. $SS|\equiv(\boldsymbol{SS} \overset{SK}{\leftarrow} \mathbf{IoT\ \ device})$ **(Goal 2)**
Based on Message 3, we derive
Step 15. $CS \triangleleft <M_8>_{KCS\text{-}SS}, <<M_9>_{KCS\text{-}SS}>_{SK}$
According to assumption A6 and the message-meaning rule, we get:
Step 16. $CS|\equiv SS|\sim(<M_8>_{KCS\text{-}SS},<<M_9>_{KCS\text{-}SS}>_{SK})$
According to assumption A3 and the freshness concatenation rule, we get:
Step 17: $CS|\equiv\#(<M_8>_{KCS\text{-}SS}, <<M_9>_{KCS\text{-}SS}>_{SK})$
According to Step 16, Step 17 and the nonce verification rule, we get:
Step 18. $CS|\equiv SS|\equiv(<M_8>_{KCS\text{-}SS}, <<M_9>_{KCS\text{-}SS}>_{SK})$
According to Step 18, assumption A7 and the believe rule, we get:
Step 19. $CS|\equiv SS|\equiv (\boldsymbol{SS} \overset{K_{CS-SS}}{\longleftrightarrow} \boldsymbol{CS})$
According to Step 16, Step 17, Step 18 and the nonce verification rule, we get:
Step 20. $CS|\equiv SS|\equiv(\boldsymbol{SS} \overset{SK}{\leftarrow} \boldsymbol{CS})$
According to assumption A6 and the jurisdiction rule, we get:
Step 21. $CS|\equiv(\boldsymbol{CS} \overset{SK}{\leftarrow} \boldsymbol{SS})$
Based on Message 4, we could derive
Step 22. $\mathbf{IoT\ \ device} \triangleleft M_{10}, M_{11}, <M_{12}>_{KCS}, <<M_{13}>_{KCS}>_{SK}$
According to assumption A4 and the message-meaning rule, we get:
Step 23. $\mathbf{IoT\ \ device}|\equiv CS|\sim(M_{10}, M_{11}, <M_{12}>_{KCS}, <<M_{13}>_{KCS}>_{SK})$
According to assumption A2 and the freshness concatenation rule, we get:
Step 24: $\mathbf{IoT\ \ device}|\equiv\#(M_{10}, M_{11}, <M_{12}>_{KCS}, <<M_{13}>_{KCS}>_{SK})$
According to Step 23, Step 24 and the nonce verification rule, we get:
Step 25. $\mathbf{IoT\ \ device}|\equiv CS|\equiv(<M_8>_{KCS\text{-}SS}, <<M_9>_{KCS\text{-}SS}>_{SK})$
According to Step 25, assumption A5 and the believe rule, we get:
Step 26. $\mathbf{IoT\ \ device}|\equiv CS|\equiv(\boldsymbol{CS} \overset{K_{CS}}{\longleftrightarrow} \mathbf{IoT\ \ device})$

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

Step 27. $\text{IoT device}|\equiv SS|\equiv(SS \overset{SK}{\leftrightarrow} \text{IoT device})$ **(Goal 3)**

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

Step 28. $\text{IoT device}|\equiv(\text{IoT device} \overset{SK}{\leftrightarrow} SS)$ **(Goal 1)**

According to Steps 14 and 28, IoT_{MAKA} successfully achieves both goals (Goals 1 and 2). Both IoT device with *MC* and *SS* believes that they share a common session key $SK = h(DID_i'\|ID_{SS}\|r_i\|r_{CS}''\|r_{SS}'')$.

4.1.3. Informal Analysis

Although it is important to provide a formal security proof on any cryptographic protocol, the informal security proof of protocols remains one of the most challenging issues for cryptography research [28]. 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. The security analysis focuses on verifying the overall security requirements for IoT_{MAKA}, including passive and active attacks, as follows.

***Proposition* 1.** *IoT_{MAKA} provides entity anonymity.*

Proof: In IoT_{MAKA}, we obtain the anonymity of the entity by applying the hash function and relying on the symmetric key cryptosystem. Two phases in IoT_{MAKA}, the registration phase, and the login and AKA phase, use encrypted amplified identities with the one-way hash function. *CS* only gets the real identity of the IoT device. There is no way for an attacker to know the real identity, even if the attacker could capture the messages $\{EID_i', M_1, M_2, M_3\}$, $\{M_4, M_5, M_6, M_7\}$, $\{M_8, M_9\}$ and $\{M_{10}, M_{11}, M_{12}, M_{13}\}$ during the protocol run of IoT_{MAKA}.

***Proposition* 2.** *IoT_{MAKA} provides untraceability.*

Proof: In IoT_{MAKA}, we obtain the *untraceability* of the entity by using a dynamic identity DID_i, formed by applying a one-way hash function on IoT device's identity concatenated with a random number. In all the phases of IoT_{MAKA}, the real identity of the IoT device is hidden except to *CS* alone. Furthermore, the registration phase and the login and AKA phase of IoT_{MAKA}, use encrypted amplified dynamic identities EID_i with the one-way hash function. *CS* only gets the real identity of IoT device. There is no way for an attacker can trace any relationship between different sessions even if the attacker could capture the messages $\{EID_i', M_1, M_2, M_3\}$, $\{M_4, M_5, M_6, M_7\}$, $\{M_8, M_9\}$ and $\{M_{10}, M_{11}, M_{12}, M_{13}\}$ during the protocol run of IoT_{MAKA}.

***Proposition* 3.** *IoT_{MAKA} cannot reveal the private key set or the generated session key to outsiders.*

Proof: The security of the private key relies on the combinations of the amplified identities and the secret values. This indicates that an attacker has to know both of them to retrieve the private key set. However, there is no way that the attacker could derive the secret values or the amplified identities from the private key set due to the one wayness of the hash function and difficulty of the symmetric key cryptosystem. For the concern of revealing the session key *SK*, the attacker needs to have the power to analyze and get the necessary information from the intercepted messages $\{EID_i', M_1, M_2, M_3\}$, $\{M_4, M_5, M_6, M_7\}$, $\{M_8, M_9\}$ and $\{M_{10}, M_{11}, M_{12}, M_{13}\}$. However, there is no way that the attacker could know the session key due to the hash function and the symmetric key cryptosystem in IoT_{MAKA}.

***Proposition* 4.** *IoT_{MAKA} provides session key freshness, therefore strong against replay attack.*

Proof: The random numbers r_i, r_{cs} and r_{ss} used to establish the session key in the login and AKA phase guarantees the freshness of the session key. An adversary cannot get any information to know the session key due to the hash function and the symmetric key cryptosystem. Furthermore, IoT_{MAKA} is strong against the replay attack due to the session key freshness support with M_3, M_7, M_9 and M_{13} in the messages.

***Proposition* 5.** *IoT_{MAKA} is secure against passive attacks.*

Proof: We assume that an attacker is successful if the attacker knows any useful information from the intercepted messages. We show that the probability of success for learning them is negligible due to the difficulty of the underlying mathematical problems, the one wayness of the hash function, and the secrecy on the symmetric key cryptosystem.

- We prove the completeness of IoT_{MAKA} by describing the run of the protocol in Section 3.
- If the attacker is passive, all the attacker can gather are the intercepted messages $\{EID_i', M_1, M_2, M_3\}$, $\{M_4, M_5, M_6, M_7\}$, $\{M_8, M_9\}$ and $\{M_{10}, M_{11}, M_{12}, M_{13}\}$. However, it is impossible to find the key related information from them due to the difficulty of the one wayness of the hash function and secrecy on the symmetric key cryptosystem.

Finally, we conclude that IoT_{MAKA} is secure against passive attack.

***Proposition* 6.** *IoT_{MAKA} is secure against active attacks.*

Proof: An attacker is successful if he/she finds the session key SK or knows any of the secrets K_{CS} and $K_{CS\text{-}SS}$. Therefore, we will show that the probability of the success of finding them is negligible due to the difficulty of the one wayness of the hash function and secrecy on the symmetric key cryptosystem.

- The acceptance by all entities means that they successfully verify each M_3, M_7, M_9, and M_{13} in the corresponding messages. Successful verification of each M_3, M_7, M_9 and M_{13} imply the use of correct session key SK. In case of acceptance, the probability that the attacker could modify the messages is negligible. Additionally, the only way for the attacker to find the session key or the private key information is to solve the difficulty of the underlying mathematical problems, the one wayness of the hash function and secrecy on the symmetric key cryptosystem.

- Now, we consider the active attacker with the following cases.

  (1) An attacker cannot get the private keys K_{CS} and $K_{CS\text{-}SS}$ due to the difficulty of the one wayness of the hash function and secrecy on the symmetric key cryptosystem.
  (2) An attacker cannot masquerade as IoT device to cheat neither CS nor SS. This is mainly because the attacker cannot generate valid messages without deriving the correct session key SK and the private keys K_{CS} and $K_{CS\text{-}SS}$. Furthermore, the attacker could not compute the proper M_3, M_7, M_9 and M_{13}, which is required for the verification of the related secret information to the counterparty.
  (3) An attacker cannot impersonate CS to cheat either IoT device or SS. Only the legal CS could form the legal messages, which matches properly with the information from the counter party in the protocol run. Even if the attacker could pass the verifications at the protocol steps, the attacker still cannot get any useful information from $\{EID_i', M_1, M_2, M_3\}$ due to the difficulty of the underlying mathematical problems, and cannot generate the

consequent valid messages. Finally, we conclude that IoT_{MAKA} is secure against active attack.

4.1.4. Features Analysis

Features analysis is a detailed examination of the protocol to see how much it satisfies the protocol design goals. The results of the analysis of IoT_{MAKA} are also compared with similar analysis results of four earlier protocols as shown in Table 3 [10, 18, 22, 28]. The comparison results show that Choi et al.'s protocol lacks the features of user anonymity and untraceability. He et al.'s protocol and Jiang et al.'s protocol lacks the detection mechanism for unauthorized login. Furthermore, He et al.'s protocol does not provide the features of mutual authentication and untraceability and is vulnerable to user impersonation attack. Compared to the four protocols, IoT_{MAKA} achieves more ideal functional features and resists most of the privacy and security attacks as shown in Table 3.

Table 3. Features comparison.

Feature / Protocol	F1	F2	F3	F4	F5	F6	F7	F8	F9
[10] protocol	No	No	Yes	No	Yes	No	Yes	Yes	Yes
[18] protocol	Yes	No	Yes	No	No	No	Yes	Yes	No
[22] protocol	Yes	Yes	Yes	No	Yes	No	Yes	Yes	No
[28] protocol	Yes	Yes	Yes	No	Yes	No	Yes	Yes	Yes
IoT_{MAKA}	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

F1: Provides anonymity, F2: Provides untraceability, F3: Provides session key agreement, F4: Eliminates human factor, F5: Provides mutual authentication, F6: Suitable for IoT, F7: Resistant to replay attack, F8: Resistant to impersonation attack, F9: Detection mechanism for unauthorized login.

4.2. Performance Analysis

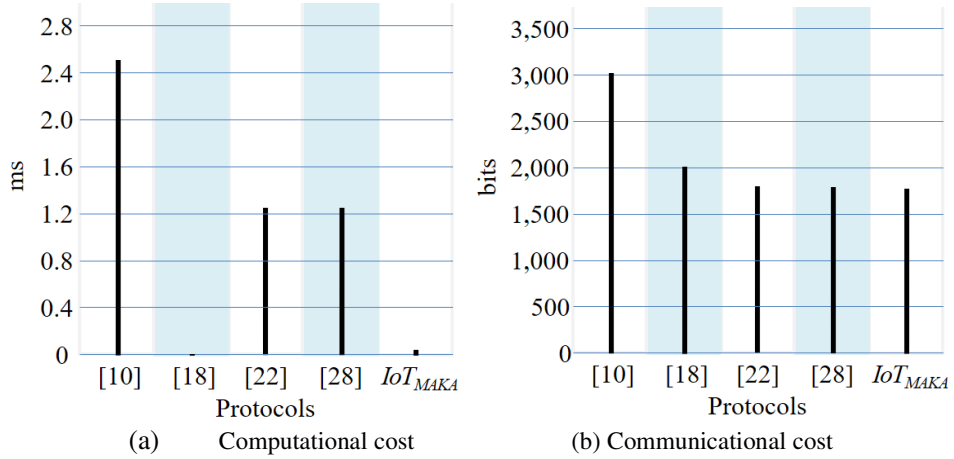
In this subsection, we analyze the computation and communication overheads of IoT_{MAKA} and provide comparisons against the four related protocols [10, 18, 22, 28]. We will provide performance and communication cost comparisons of IoT_{MAKA} against the protocols.

4.2.1. Computational Overhead Analysis

In this subsection, we analyze the computational overheads in terms of time taken for each step in the protocol run. To facilitate the evaluation of computation costs, we use a scale provided by Wu et al. [42]. They provided computation costs for a symmetric key operation (T_h), an asymmetric key operation (T_A), and an ECC point multiplication (T_E) as 0.0000328 *ms*, *0.0214835 ms* and 0.427576 *ms*, respectively. Although we used two fuzzy commitment functions in IoT_{MAKA}, we consider them as the same as the hash function because of their similar nature. Table 4 and Figure 5 (a) show the computational cost comparisons of IoT_{MAKA} and the related four protocols. Results from Table 4 show that IoT_{MAKA} is more efficient than the protocols of Choi et al; Jiang et al; and Li et al. He et al.'s protocol is slightly more efficient than IoT_{MAKA} but their protocol lacks many functional features as shown in Table 3. He et al.'s protocol is also vulnerable to many attacks. IoT_{MAKA} keeps the efficiency of computations by reducing computational overheads by 25% on average and achieves most functional, security, and privacy features.

Table 4. Computational cost comparison.

Protocol \ Entity	IoT device	CS	SS	Total
[10] protocol	$3T_E+9T_h$	$1T_E+5T_h$	$2T_E+6T_h$	$6T_E+20T_h$=2.566112 *ms*
[18] protocol	$8T_h$	$9T_h$	$6T_h$	$23T_h$=0.0007544 *ms*
[22] protocol	$2T_E+8T_h$	$1T_E+9T_h$	$6T_h$	$3T_E+23T_h$=1.2834824 *ms*
[28] protocol	$2T_E+8T_h$	$1T_E+9T_h$	$4T_h$	$3T_E+21T_h$=1.2834168 *ms*
IoT_{MAKA}	$10T_h$	$2T_A + 11T_h$	$4T_h$	$2T_A + 25T_h$=0.0437870*ms*



(a) Computational cost (b) Communicational cost

Figure 5. Performance comparisons among related protocols

4.2.2. Communicational Overhead Analysis

In this subsection, we analyze the communication overhead costs in terms of bit-length of a random number, timestamp, etc. The length of a random number, fuzzy commitment function, timestamp, one-way hash function digest, secret key, identity, and password are 128 bits and the length of ECC point multiplication is 160 bits [28]. Table 5 and Figure 5 (b) list the comparison result of communication costs analysis between IoT_{MAKA} and the related four protocols [10, 18, 22, 28]. The required bits for the communication in the protocols of Choi et al., He et al., Jiang et al. Li et al. and IoT_{MAKA} are 3,072 bits, 2,048 bits, 1,856 bits, 1,856 and 1,792 bits, respectively. IoT_{MAKA} requires a smaller number of bits for the communication aspect than all the four other related protocols. Therefore, IoT_{MAKA} minimizes communication costs by 20% on average.

Table 5. Communicational cost comparison.

Protocol \ Entity	IoT device	*CS*	*SS*	Total
[10] protocol	5*128+160=800 bits	3*128=384 bits	11*128+3*160=1,888 bits	3,072 bits
[18] protocol	6*128=768 bits	6*128=768 bits	4*128=512 bits	2,048 bits
[22] protocol	2*160+3*128=704 bits	5*128=640 bits	4*128=512 bits	1,856 bits
[28] protocol	2*160+3*128=704 bits	7*128=986 bits	2*128=256 bits	1,856 bits
IoT_{MAKA}	4*128=512 bits	8*128=1,024 bits	2*128=256 bits	1,792 bits

5. CONCLUSION

In this paper, we have designed new privacy-preserving MAKA for IoT environments, denoted as IoT_{MAKA}. In IoT_{MAKA}, there are three entities, IoT device, *CS* and *SS*. The IoT device has an active role in initiating communication by sending requests to *SS* through *CS*. The IoT device must have knowledge of available service providers and the services they provide. Because of the need of privacy, there is *CS* as a trusted central controlling unit and authenticating entity. *SS* is the service provider and actuator. In IoT_{MAKA}, we had three major aims namely:

- To eliminate the human factor in IoT AKA
- To propose a protocol that prioritizes privacy provision by providing entity anonymity and untraceability
- To maximize efficiency in computational and communication costs.

Firstly, we drew the required features that AKA should satisfy by reviewing and analyzing some previous protocols. The required features are drawn the privacy of communicating entity, the security of communicated data by achieving major security goals. We designed privacy-preserving MAKA protocol, IoT_{MAKA}. After the design and run of IoT_{MAKA}, we provided analyses in three ways namely; formal security analysis by using BAN logic, informal analysis by using cryptanalysis, and performance analysis comprising of communication and computational overhead analysis. Analyses results showed that IoT_{MAKA} is better on privacy-preservation than in earlier protocols. Moreover, in terms of communication and computation overheads, IoT_{MAKA} reduces communication overheads by 20% and computational overheads by 25% on average as compared to the four earlier protocols. IoT_{MAKA} can be useful in privacy preserving applications in real life. IoT_{MAKA} provides privacy alongside the security of the communicated message. IoT_{MAKA} can ensure user confidence to promote wide acceptance and reap the potentials of IoT. For example, IoT_{MAKA} can assure the privacy of sensitive information of patients monitored in smart health. We leave practical implementations IoT_{MAKA} and its usability in restricted areas and areas with no internet coverage for future works.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

ACKNOWLEDGMENTS

The results in this paper are part of Beaton Kapito's Master's degree thesis. The corresponding author is Hyunsung Kim. This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2017R1D1A1B04032598).

REFERENCES

[1] Aarika, K., Bouhlal, M., Abdelouahid, R. A., Elfilali, S. & Benlahmar, E., (2020) "Perception layer security in the internet of things", *Procedia Computer Science*, Vol. 175, pp. 591-596.

[2] Alzahrani, B. & Fotiou, N., (2020) "Enhancing Internet of Things Security using Software-Defined Networking", *Jourlan of Systems Architecture*, Vol. 110, No. 101779.

[3] Azarmehr, M., Ahmadi, A. & Rashidzadeh, R., (2017) "Secure authentication and access mechanism for IoT wireless sensors", in *Proc. of ISCAS*, pp. 1-4.

[4] Aziz, A. & Singh, K., (2019) "Lightweight Security Scheme for Internet of Things", *Wireless Personal Communications*, Vol. 104, No. 2, pp. 557-593.

[5] Bagay, D., (2020) "Information security of Internet things", *Procedia Computer Science*, Vol. 169, pp. 179-182.

[6] Barbosa, G., Endo, P. T. & Sadok, D., (2019) "An internet of things security system based on grouping of smart cards managed by field programmable gate array, *Computers & Electrical Engineering*, Vol. 74, pp. 331-348.

[7] Bugeja, J., Jacobsson, A. & Davidsson, P., (2016) "On privacy and security challenges in smart connected homes", in *Proc. of European Intelligence and Security Informatics Conference*, pp. 172-175.

[8] Burrows, M., Abadi, M. & Needham, R., (1989) "A logic of authentication", *Royal Society of London Mathematical, Physical and Engineering Sciences*, Vol. 426, pp. 233-271.

[9] Chandan, R. R. & Mishra, P. K., (2020) "Consensus Routing and Environmental Discrete Trust Based Secure AODV in MANETs", *International Journal of Computer Networks & Communications*, Vol. 12, No. 3, pp. 1-20.

[10] Choi, Y., Lee, D., Kim, J., Jung, J., Nam, J. & Won, D., (2014) "Security enhanced user authentication protocol for wireless sensor networks using elliptic curves cryptography", *Sensors*, Vol. 14, pp. 10081-10106.

[11] Dabbagh, Y. & Saad, W., (2016) "On the Authentication of Devices in the Internet of Things", in *Proc. of IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks*, pp. 21-24.

[12] Das, M., (2009) "Two-factor user authentication in wireless sensor networks", *IEEE Trans. Wireless. Communication*, Vol. 8, pp. 1086-1090.

[13] Dawy, Z., Saad, W., Ghosh, A., Andrews, J. & Yaacoub, E., (2017) "Towards massive machine type cellular communications", *IEEE Wireless Communications Magazine*, Vol. 24, No. 1, pp. 120-128.

[14] Dolev, D. & Yao, A. C., (1983) "On the security of public key protocols", *IEEE transactions on information theory*, Vol. 29, pp. 198-208.

[15] Farash, M., Turkanović, M., Kumaric, S. & Hölbl, M., (2016) "An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment", *Ad Hoc Networks*, Vol. 36, No. 1, pp. 152-176.

[16] Gazis, V., Görtz, M., Huber, M., Leonardi, A., Mathioudakis, K., Wiesmaier, A., Zeiger, F. & Vasilomanolakis, E., (2015) "A survey of technologies for the Internet of Things", in *Proc. of International Wireless Communications and Mobile Computing Conference*, pp. 1090-1095.

[17] Gubbi, J., Buyya, R., Marusic, S. & Palaniswami, M., (2013) "Internet of Things (IoT): A vision, architectural elements, and future directions", *Future Generation Computer Systems*, Vol. 29, No. 7, pp. 1645-1660.

[18] He, D., Kumar, N. & Chilamkurti, N., (2015) "A secure temporal-credential-based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks", *Information Science*, Vol. 321, pp. 263-277.

[19] Heinzelman, W., Chandrakasan, A. & Balakrishnan, H., (2002) "An application-specific protocol architecture for wireless micro sensor networks", *IEEE Trans. Wireless Communication*, Vol. 1, pp. 660–670.

[20] Hustinx, P., (2010) "Privacy by design: delivering the promises", *Identity in the Information Society*, Vol. 3, No. 2, pp. 253–255.

[21] Jaychand, A. & Behar, N., (2017) "A Survey on IoT Security Threats and Solutions", *International Journal of Innovative Research in Computer and Communication Engineering*, Vol. 5, No. 3, pp. 5187-5193.

[22] Jiang, Q., Ma, J., Lu, X. & Tian, Y., (2015) "An efficient two-factor user authentication scheme with Unlinkability for wireless sensor networks", *Peer-to-Peer Network Application*, Vol. 8, pp. 1070-1081.

[23] Jiang, Q., Ma, J., Wei, F., Tian, Y., Shen, J. & Yang, Y., (2016) "An untraceable temporal credential based two-factor authentication scheme using ECC for wireless sensor networks", *Journal of Network Computer Applications*, Vol. 76, pp. 37-48.

[24] Juels, A. & Wattenberg, M., (1999) "A fuzzy commitment scheme", in *Proc. of 6th ACM conference on Computer and communications security*, pp. 28-36.

[25] Khan, M. & Alghathbar, K., (2010) "Cryptanalysis and security improvements of two-factor user authentication in wireless sensor networks", *Sensors*, Vol. 10, pp. 2450-2459.

[26] Kim, H., (2020) "Privacy Preserving Authentication Protocol over VANET", *The Journal of Korean Institute of Communications and Information Scciences*, Vol. 45, No. 6, pp. 941-950.

[27] Koshy, P., Babu, S. & Manoj, B. S., (2020) "Sliding Window Blockchain Architecture for Internet of Things", *IEEE Internet of Things Journal*, Vol. 7, No. 4, pp. 3338–3348.

[28] Li, X. & Niu, J., (2018) "A three-factor anonymous authentication scheme for wireless sensor networks in internet of things environments", *Journal of Network and Computer Applications*, Vol. 103, pp. 194-204.

[29] Medaglia, C. & Serbanati, A., (2010) "An overview of privacy and security issues in the internet of things", *The Internet of Things*, pp. 389-395.

[30] Patel, H., (2015) "Non-parametric feature generation for RF-fingerprinting on ZigBee devices", in *Proc. of IEEE Symposium on Computational Intelligence for Security and Defence Applications*, pp. 1-5.

[31] Patel, K. & Patel, S., (2016) "Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges", *International Journal of Engineering Science and Computing*, Vol. 6, No. 5, pp. 6122-6131.

[32] Shi, W. & Gong, P., (2013) "A new user authentication protocol for wireless sensor networks using elliptic curves cryptography", *International Journal Distribitive Sensor Network*, Vol. 12, No. 3, pp. 42-49.

[33] Sicari, S., Rizzardi, A., Grieco, L. & Coen-Porisini, A., (2015) "Security, Privacy and Trust in Internet of Things: The Road Ahead", *Computer Networks*, Vol. 76, No. 15, pp. 146-164.

[34] Skarmeta, A. & Moreno, M., (2014) "Internet of things", *Lecture Notes in Computer Science*, Vol. 8425, pp. 48-53.

[35] Sweeney, L., (2002) "k-anonymity: A model for protecting privacy", *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, Vol. 10, No. 5, pp. 557-570.

[36] Turkanovi´c, M. & Hölbl, M., (2013) "An improved dynamic password-based user authentication scheme for hierarchical wireless sensor networks", *Electron Electrics*, Vol. 6, pp. 109-116.

[37] Vaidya, B., Makrakis, D. & Mouftah, H., (2010) "Improved two-factor user authentication in wireless sensor networks", in *Proc. of IEEE 6th Wireless and Mobile Computing, Networking and Communications conference*, pp. 600-606.

[38] Vasilomanolakis, E., Daubert, J., Luthra, M., Gazis, V., Wiesmaier, A. & Kikiras, P., (2015) "On the security and privacy of Internet of Things architectures and systems", in *Proc. of International Workshop on Secure Internet of Things*, pp. 49-57.

[39] Wang, D., Wang, N., Wang, P. & Qing, S., (2015) "Preserving privacy for free: efficient and provably secure two-factor authentication scheme with user anonymity", *Information Science*, Vol. 321, pp. 162-178.

[40] Weber, R., (2010) "Internet of Things – New security and privacy challenges", *Computer Law & Security Review,* Vol. 26, No. 1, pp. 23-30.

[41] Whitmore, A., Agarwal, A. & Da, L., (2014) "The internet of things-A survey of topics and trends", *Information Systems Frontiers*, pp. 1-14.

[42] Wu, F., Xu, L., Kumari, S., Li, X., Das, A., Khan, M., Karuppiah, M. & Baliyan, R., (2016) "A novel and provably secure authentication and key agreement scheme with user anonymity for global mobility networks", *Secure Communication Networks*, Vol. 9, pp. 3527–3542.

[43] Xu, Q., Zheng, R., Saad, W. & Han, Z., (2016) "Device fingerprinting in wireless networks: Challenges and opportunities", *IEEE Communications Surveys & Tutorials*, Vol. 1, pp. 94-104.

[44] Xue, K., Ma, C., Hong, P. & Ding, R., (2013) "A temporal-credential-based mutual authentication and key agreement scheme for wireless sensor networks", *Journal of Network Computing Applications*, Vol. 36, pp. 316-323.

[45] Yeh, H., Chen, T., Liu, P., Kim, T.-H. & Wei, H., (2011) "A secured authentication protocol for wireless sensor networks using elliptic curves cryptography", *Sensors*, Vol. 11, pp. 4767–4779.

[46] Ziegeldorf, J., Morchon, O. & Wehrle, K., (2014) "Privacy in the Internet of Things: Threats and challenges", *Security and Communication Networks*, Vol. 7, No. 12, pp. 2728-2742.

AUTHORS

Beaton Kapito received a B.E. degree in Mathematics from Chancellor College of the University of Malawi and is currently a Masters Degree student with the Department of Mathematics, Chancellor College, University of Malawi. He is also working as a part-time lecturer at Chancellor College, the University of Malawi since 2017. He is also an adjunct lecturer at Malawi Adventist University, an affiliate of The University of Eastern Africa, Baraton. He has been a Mathematics teacher at Chikwawa Secondary School, Soche Adventist Secondary School, and Chileka Mission Secondary School. His research interest is in Cryptography: His Masters thesis proposal is "Privacy-Preserving Authenticated Key Agreement for Internet of Things."

Mwawi Nyirenda received the Ph.D. degree from Royal Holloway, University of London, the United Kingdom in 2018 and the M.Sc in Information Theory, Coding and Cryptography from Mzuzu University, Malawi in 2009. She is a senior lecturer at the Department of Mathematical Sciences, Chancellor College, University of Malawi and is the current Head of Department. Her research focus is considering how cryptography can be applied to improve the security and privacy of patient information communicated wirelessly in healthcare applications.

Hyunsung Kim received the M.Sc. and Ph.D. degrees in computer engineering from Kyungpook National University, Korea, in 1998 and 2002, respectively. He is a Professor at the School of Computer Science, Kyungil University, Korea from 2012. Furthermore, he is currently a visiting professor at the Department of Mathematical Sciences, Chancellor College, University of Malawi, Malawi from 2015. He also was a visiting researcher at Dublin City University in 2009. From 2000 to 2002, he worked as a senior researcher at Ditto Technology. He had been an associate professor from 2002 to 2012 with the Department of Computer Engineering, Kyungil University. His research interests include cryptography, VLSI, authentication technologies, network security, ubiquitous computing security, and security protocol.