Enhanced Mutual Authenticated Key Agreement Protocol for Anonymous Roaming Service in Global Mobility Networks

Hyunsung Kim
Dept. of Cyber Security, Kyungil University, Kyungbuk, Korea
Dept. of Mathematical Sciences, University of Malawi, Zomba, Malawi

***Abstract*—With the rapid development of mobile intelligent terminals, users can enjoy ubiquitous life in global mobility networks (GLOMONET). It is essential to secure user information for providing secure roaming service in GLOMONET. Recently, Xu et al. proposed a mutual authentication and key agreement (MAKA) protocol as the basic security building block. The purpose of this paper is not only to show some security problems in Xu et al.'s MAKA protocol and but also proposes an enhanced MAKA protocol as a remedy protocol for Xu et al.'s MAKA protocol. The proposed protocol ensures higher security compared to the well-known authentication and key agreement protocols but has a bit computational overhead than them due to the security enhancements.**

***Keywords*—Information security; roaming security; anonymity; authenticated key agreement; cryptanalysis**

I. Introduction

GLOMONET provides global roaming service to users moving from one network to another [1-2]. Users can enjoy rich and colorful services, such as online shopping, social entertainment, bank transfer and security exchange, with the help of GLOMONET network entities. Roaming service enables mobile user (MU) to use the services extended by home agent (HA) in a foreign agent (FA). Thus, user authentication and key agreement protocol for roaming service plays the very important role in GLOMONET [3-5]. In particular, the authentication and key agreement protocol for roaming service enables a MU and a FA authenticate each other and agree on a common session key to establish a secure channel over GLOMONET with the help of the HA. During roaming process in GLOMONET, privacy protection, especially focused on user anonymity, is a challenging and essential requirement that the identity of MU is protected against adversaries. Mutual authentication is also a very important security aspect. It requires that MU, FA and HA prove their authenticity to each other before offering any application services in GLOMONET.

To support roaming facility, several authentication and key agreement protocols [6-] have been proposed in GLOMONET. However, many of them have been proved to be insecure against known attacks. Zhu et al. proposed a two-factor authentication scheme but Lee et al. showed that Zhu et al.'s scheme does not achieve mutual authentication and is vulnerable to impersonation attack [6-7]. Furthermore, Lee et al. proposed a remedy scheme for Zhu et al.'s scheme. But Wu et al. showed that Lee et al.'s scheme fails to provide user anonymity [8]. Wang et al. also introduced a new authentication scheme but Jeon et al. pointed out that Wang et al.'s scheme cannot withstand against forgery attacks and fails to achieve anonymity [9-10]. Independently, Chang et al. proved Lee et al.'s scheme fails to achieve user anonymity and proposed a new authentication scheme [11]. Unfortunately, Youn et al. found that Change et al.'s scheme cannot provide anonymity [12]. Recently, Zhou et al. proposed a MAKA protocol based on the decisional Diffie–Hellman assumption [13]. While Gope et al. pointed out that Zhou et al.'s protocol is vulnerable to reply attacks and insider attack and proposed a new protocol [14]. However, Xu et al. showed that Gope et al.'s protocol is susceptible to replay attack and have a large storage burden with some more problems and proposed a new novel efficient MAKA protocol with desynchronization for anonymous roaming service in GLOMONET [15].

There are two purposes of this paper, to show deficiencies of Xu et al.'s protocol and to propose a new remedy MAKA protocol. Xu et al.'s protocol is lightweight but has a protocol flaw and is susceptible to off-line identifier and password guessing attack, stolen verifier attack and denial of service (DoS) attack. We utilize symmetric cryptosystem to implement pseudonym identifier in each session, which can achieve anonymity. Therefore, the proposed protocol could achieve more secure properties compared to the other well-known MAKA protocols but has a bit more overhead to draw some more functions to be secure enough.

The rest of the paper is organized as follows: in Section II, we provide a brief overview of GLOMONET and Xu et al.'s MAKA protocol. Section III provides an attack model and security flaws in Xu et al.'s MAKA protocol. Sections IV and V propose an enhanced MAKA protocol to solve the weaknesses in Xu et al.'s protocol with the security and performance analysis. Finally, Section VI provides the conclusion.

II. Backgrounds

This section provides an overview of the target network and Xu et al.'s MAKA protocol [15]. The purpose of this section is to withdraw security flaws in Xu et al.'s MAKA protocol.

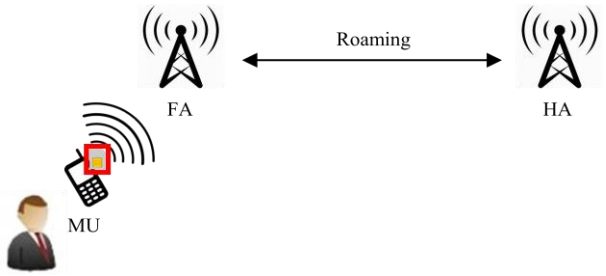

Fig. 1. Network Configuration for GLOMONET.

A. Global Mobility Network

Increased use of digital communication systems including cellular phones has led to support the roaming service in GLOMONET. Mobility is a function, which enables a MU to access the services of foreign network (FN) while roaming [16]. In GLOMONET, MUs can access their home network (HN) services from remote places with the help of FA. Authenticity of the MUs plays a crucial role to gain the access to the network services. In roaming scenario, there are three entities, MU, FA and HA. MUs in GLOMONETs visit FN, the role of FN is to authenticate MU with the help of HA as shown in Fig. 1.

B. Xu et al.'s MAKA Protocol

Xu et al. proposed a MAKA protocol as a remedy scheme of Gope et al.'s protocol [15]. This subsection reviews Xu et al.'s MAKA protocol. Table 1 shows the notations used in this paper.

Xu et al.'s MAKA protocol is consisted of four phases, registration phase, mutual authentication and key agreement phase, password renewal phase and shared key renewal phase.

[Registration phase] In this phase, MU uses real identity to register in HA through secure channel. After registration, MU gets a SC, which stores the authentication information. The details are

Step 1: MU sends his/her identity ID_M to HA through the secure channel.

Step 2: After receiving ID_M, HA randomly generates two numbers n_h and n_0 and then computes $K_{uh} = h(ID_M||n_h)$ and $EID = E_k(ID_M||n_0)$. Hereafter, HA stores ID_M and K_{uh}, forms a message $\{EID, K_{uh}, h()\}$ and sends it to MU through a secure channel.

Step 3: MU chooses a password PSW_M upon receiving the message sending from HA. And MU computes $EID^* = EID \oplus h(ID_M||PSW_M)$, $K_{uh}^* = K_{uh} \oplus h(ID_M||PSW_M)$. Finally, MU replaces EID with EID^* and K_{uh} with K_{uh}^*. Now SC contains $\{EID^*, K_{uh}^*, h()\}$.

[Message authentication and key agreement phase] In this phase, MU and FA authenticate and establish a session key each other with the assistance of HA. The details are

Step 1: MU generates a random number N_m and submits his/her identity ID_M and password PSW_M to SC. MU derives $K_{uh} = K_{uh}^* \oplus h(ID_M||PSW_M)$, $EID = EID^* \oplus h(ID_M||PSW_M)$ and computes $N_x = h(ID_M||K_{uh}) \oplus N_m$ and $V_1 = h(EID||N_x||T_1||ID_M||K_{uh})$. Finally, MU forms a message $M_{A1} : \{EID, N_x, ID_h, V_1, T_1\}$ and sends it to FA.

Step 2: After receiving M_{A1}, FA first checks whether the current time is within T_1. If not, the protocol terminates immediately. Otherwise, FA generates a random number N_f and computes $N_y = h(K_{fh}) \oplus N_f$ and $V_2 = h(EID||N_x||N_y||T_2||K_{fh}||N_f)$. Finally, FA forms a message M_{A2}: $\{EID, N_x, ID_f, V_1, T_1, N_y, V_2, T_2\}$ and sends it to HA.

Step 3: When HA receives M_{A2}, it checks whether the current time is within T_2. If not, the protocol terminates immediately. Otherwise, HA computes $N_f = h(K_{fh}) \oplus N_y$, $V_2^* = h(EID||N_x||N_y||T_2||K_{fh}||N_f)$ and then it checks whether V_2^* is equal to V_2. If not, it will terminate the connection. Otherwise, HA decrypts EID through $ID_M||n_0 = D_k(EID)$. Next, it computes $V_1^* = h(EID||N_x||T_1||ID_M||K_{uh})$ and checks whether V_1^* is equal to V_1. If not, it terminates the connection. Otherwise, HA generates a random number n_1 and computes $D = E_k(ID_M||n_1)$ and $FID^* = FID \oplus h(ID_M||K_{uh})$. Hereafter, it derives $N_m = h(ID_M||K_{uh}) \oplus N_x$, $N_x' = h(K_{uh}||ID_M||N_m) \oplus N_f \oplus n_0$, $N_y' = h(K_{fh}||ID_f||N_f) \oplus N_m \oplus n_0$, $V_3 = h(N_y'||N_f) \oplus K_{fh}$, and $V_4 = h(N_x'||FID^*||N_m) \oplus K_{uh}$. At last, HA forms a response message M_{A3}:$\{N_x', N_y', V_3, V_4, FID^*\}$ and sends it to FA.

Step 4: Upon receiving M_{A3}, FA computes $V_3^* = h(N_y'||N_f) \oplus K_{fh}$ and checks whether it is equal to V_3. If so, it derives $N_m \oplus n_0 = h(K_{fh}||ID_f||N_f) \oplus N_y'$ and computes a session key $SK = N_m \oplus n_0 \oplus N_f$. Finally, it sends the message $M_{A4} : \{N_x', V_4, FID^*\}$ to MU.

Step 5: Upon receiving M_{A4}, MU computes $V_4^* = h(N_x'||FID^*||N_m) \oplus K_{uh}$ and checks whether it is equal to V_4. If the verification is successful, he/she computes $N_f \oplus n_0 = h(K_{uh}||ID_M||N_m) \oplus N_x'$ and derives the session key $SK = N_m \oplus n_0 \oplus N_f$ and then, he/she computes $FID = FID^* \oplus h(ID_M||K_{uh})$ and replaces EID with FID.

[Password renewal phase] To change the password, MU needs to use his/her old password PSW_M and enter the new password PSW_M^*. After that, MU computes $K_{uh} = K_{uh}^* \oplus h(ID_M||PSW_M)$, $EID = EID^* \oplus h(ID_M||PSW_M)$, $K_{uh}^{**} = K_{uh} \oplus h(ID_M||PSW_M^*)$ and $EID^{**} = EID \oplus h(ID_M||PSW_M^*)$. MU replaces K_{uh}^* with K_{uh}^{**} and EID^* with EID^{**} in SC.

TABLE I. NOTATIONS

Symbol	Description
MU	Mobile user
FA	Foreign agent
HA	Home agent
SC	Smartcard
SK	Session key
ID_M	Identity of MU
ID_h	Identity of HA
ID_f	Identity of FA
PWS_M	Password of MU
K_{uh}	Shared key between MU and HA
K_{fh}	Shared key between FA and HA
r_i, N_i	Random numbers
T_i	Timestamp
$E_k(.)$, $D_k(.)$	Symmetric key encryption/decryption with key k
EID	Dynamic identity of MU
$h(.)$	One-way hash function
$\|\|$	Bitwise concatenation
$\oplus$	Bitwise exclusive-or

[Shared key renewal phase] This phase is to reestablish the shared key between MU and HA after the shared key is suspected of disclosure. Firstly, MU sends his/her real identity ID_M to HA through secure channel and HA computes the new shared key $K_{uh} = K_{uh}^{*} \oplus h(ID_M || n_h)$ and sends it to MU through the secure channel. After receiving the message, MU updates the shared key in SC.

III. Cryptanalysis on Xu et al.'s MAKA Protocol

This section provides cryptanalysis on Xu et al.'s MAKA protocol based on Dolev-Yao security model in [17]. We will show that Xu et al.'s MAKA protocol is weak against off-line identifier and password guessing attack, stolen verifier attack and denial of service attack with a protocol flaw.

A. Dolev-Yao Attack Model

The motivation of Dolev-Yao model is to verify public key protocols against active attacks with considerable power [17]. In their model, following attacker assumptions are

- Adversary has complete control over the entire network
- Adversary acts as a legitimate user and can obtain any message from any party
- Adversary can initiate the protocol with any party and can be a receiver to any party in the network.

Furthermore, we add two more assumptions to Dolev-Yao model that are for the proper cryptanalysis of MAKA protocol as follows

- Adversary may obtain all the sensitive parameters stored in SC's by monitoring the power consumption of it if adversary could steal MU's SC [18]
- Adversary can steal the verification table from HA.

B. Security Weakness in Xu et al.'s MAKA Protocol

This section shows the security weaknesses of Xu et al.'s MAKA protocol, which will shows that adversary can mount different types of attacks on the MAKA protocol based on Dolev-Yao attack model with two additional assumptions described in the subsection 3.1. Firstly, we will show a flaw in Xu et al.'s MAKA protocol and will show three security weaknesses in it.

[Protocol Flaw] A security protocol is a concrete protocol that performs a security related function and applies cryptographic methods. It should be a sufficiently detailed protocol, which can be used to implement multiple and interoperable versions of a program [19]. However, Xu et al.'s MAKA protocol is incomplete because it does not define *FID* properly but just used to form FID^{*} in step 3 of the message authentication and key agreement phase. That is the reason why we would like to change *D* into *FID* for the proper protocol run.

[Off-Line Identifier and Password Guessing Attack] Since the message authentication and key agreement phase of Xu et al.'s MAKA protocol is executed in the open network environment, an attacker can eavesdrop the communication channels among MU, FA and HA before the start of this attack. Moreover, we assumed that the attacker stole MU's SC. Thus, the attacker could get the messages, M_{A1} : {*EID*, N_x, ID_h, V_1, T_1}, M_{A2} : {*EID*, N_x, ID_f, V_1, T_1, N_y, V_2, T_2}, M_{A3} : {N_x', N_y', V_3, V_4, FID^{*}} and M_{A4} : {N_x', V_4, FID^{*}} from the communication channels. Furthermore, the attacker could get the important information on the memory of SC of MU, {EID^{*}, K_{uh}^{*}, $h()$}. By using the acquired information, the attacker could compute $EID \oplus K_{uh} = EID^{*} \oplus K_{uh}^{*}$ from the memory of SC and get $K_{uh}' = EID \oplus K_{uh} \oplus EID$ by using *EID* in M_{A1}. After that, the attacker could perform the off-line identifier and password guessing attack as follows. First of all, the attacker tries to perform the identifier guessing attack by using V_1 with the related information. (1) The attacker guesses an identifier candidate ID_{Mi} and computes $V_1' = h(EID || N_x || T_1 || ID_{Mi} || K_{uh}')$ in an off-line manner. (2) The attacker checks whether V_1' is equal to V_1 or not. If they are the same, the identifier guessing is successful. Otherwise, the attacker repeats Steps (1) and (2) until the correct one is withdrawn. After that with the properly derived ID_{Mi}, the attacker tries the password guessing attack by using EID^{*} or K_{uh}^{*} with the related information. (1) The attacker guesses a password candidate PSW_{Mi} and computes $EID^{*'} = EID \oplus h(ID_{Mi} || PSW_{Mi})$ in an off-line manner. (2) The attacker checks whether $EID^{*'}$ is equal to EID^{*} or not. If they are the same, the password guessing is successful. Otherwise, the attacker repeats Steps (1) and (2) until the correct password is withdrawn.

[Stolen Verifier Attack] The legitimacy of user in Xu et al.'s MAKA protocol is determined based on the verifier. As we mentioned in the attack model, an attacker can steal the verifier {ID_M and K_{uh}} stored in HA for this attack. Even if the verifier does not include the secret key of HA, the attacker could pretend to be an honest HA for MU by forming a legitimate message M_{A4}, which needs to be send to MU. The attacker could perform the FA masquerading attack based on the stolen verifier attack as follows. (1) The attacker performs a dictionary attack to find the proper identifier ID_{Mi} by using $V_1' = h(EID || N_x || T_1 || ID_{Mi} || K_{uh}')$ based on the verifier with the request message M_{A1}: {*EID*, N_x, ID_h, V_1, T_1} from MU in an off-line manner. (2) The attacker forms a legal message M_{A4} : {N_x', V_4, FID^{*}} after selecting two random numbers N_x' and FID^{*}, deriving $N_m' = N_x \oplus h(ID_{Mi} || K_{uh})$ and computing $V_4 = h(N_x' || FID^{*} || N_m') \oplus K_{uh}$. (3) The attacker derives a session key as $SK = N_m' \oplus h(K_{uh} || ID_{Mi} || N_m') \oplus N_x'$, which will be the same with MU's computation.

[Denial of Service Attack] This attack is a cyber-attack in which the perpetrator seeks to make a resource unavailable to its intended users by disrupting services of a host. The password renewal phase only changes without checking the ownership of MU. That is the reason why any attacker could try to perform that phase with any PSW_{Mi} and PSW_{Mi}^{*} pair when MU temporarily vacate his/her system with SC. The attacker performs denial of service attack as follows. (1) The attacker uses two random numbers for passwords PSW_{Mi} and PSW_{Mi}^{*}. (2) The attacker computes $K_{uh}' = K_{uh}^{*} \oplus h(PSW_{Mi})$, $EID' = EID^{*} \oplus h(PSW_{Mi})$, $K_{uh}'' = K_{uh}' \oplus h(PSW_{Mi}^{*})$ and $EID'' = EID' \oplus h(PSW_{Mi}^{*})$. (3) The attacker replaces K_{uh}^{*} with K_{uh}^{**} and EID' with EID'' in SC. After this, MU cannot use the service from FA based on SC.

IV. Enhanced MAKA Protocol

This section proposes an enhanced MAKA protocol to overcome the weaknesses of Xu et al.'s MAKA protocol. We need to design a new protocol, which does not use verification table in HA side with the other aspects to resist various attacks. The design goals of our enhanced MAKA protocol are as follows

- To achieve mutual authentication with the provision of anonymity
- To establish the session key fairly
- To resist common attacks, such as guessing attack, lost smart card attack, denial of service attack and so on
- To provide user friendliness of password change
- To achieve computational and communicational efficiency.

Enhanced MAKA protocol is composed of three phases, registration phase, mutual authenticated key agreement phase and password renewal phase. Enhanced MAKA protocol does not need to have the shared key renewal phase because the key is updated once in the mutual authenticated key agreement phase run. In the registration phase, MU registers any specific services to HA by using real identity through secure channel. Unlike Xu et al.'s MAKA protocol, enhanced MAKA protocol does not need to use a verifier table in HA, which improves the security of the protocol. The mutual authenticated key agreement phase provides mutual authentication and key agreement. In this phase, MU and FA can authenticate each other with the assistance of HA with a proper session key establishment. The password renewal allows MU to update the password without the supervision of HA only after the proper MU authentication.

A. Registration Phase

In this phase, MU registers his/her identity to HA and HA issues MU a SC to be used in the further phases. The whole processes of this phase require to be processed through a secure channel. Fig. 2 depicts the processes of this phase, which are given in detail as follows

Step 1: MU selects and sends his/her real identity ID_M to HA.

Step 2: After receiving ID_M, HA generates a random number n_0 and computes $K_{uh} = h(ID_M||n_0)$ and $EID = E_k(ID_M||n_0)$, which k is the master key only known by HA and EID is the dynamic identity of MU. HA issues a SC by writing $\{EID, K_{uh}, h()\}$ in the memory of it and sends it to MU.

Step 3: MU chooses a password PSW_M upon receiving the message sending from HA. MU computes $EID^* = EID \oplus h(ID_M||PSW_M)$, $K_{uh}^* = K_{uh} \oplus h(PSW_M||ID_M)$ and $AV = h(EID||K_{uh})$. Finally, MU replaces EID with EID^* and K_{uh} with K_{uh}^*. Now SC contains $\{EID^*, K_{uh}^*, AV, h()\}$.

The important feature in this phase is that it does not need to keep ID_M and n_0 in HA side for the further processing of the protocol, which could enhance the security of the protocol.

B. Mutual Authenticated Key Agreement Phase

In this phase, MU and FA can establish a session key only after mutual authentication is successful with the assistance of HA. It uses the dynamic identity to achieve anonymity of MU. Fig. 3 depicts the processes of this phase, which are given in detail as follows

Step 1: MU inputs ID_M and PSW_M to SC. SC derives $K_{uh} = K_{uh}^* \oplus h(PSW_M||ID_M)$ and $EID = EID^* \oplus h(ID_M||PSW_M)$ and computes $AV' = h(EID||K_{uh})$. If AV' is not equal to AV, SC terminates the protocol. Otherwise, SC generates a random number N_m and computes $N_x = h(ID_M||K_{uh}) \oplus N_m$ and $V_1 = h(EID||N_x||T_1||ID_M||K_{uh})$. Finally, SC forms M_{A1} : { EID, N_x, ID_h, V_1, T_1} where T_1 is a timestamp of SC and sends it to FA.

Step 2: After receiving M_{A1}, FA first checks whether the current time is within T_1. If not, the protocol terminates immediately. Otherwise, FA generates a random number N_f and computes $N_y = h(K_{fh}) \oplus N_f$ and $V_2 = h(EID||N_x||N_y||T_2||K_{fh}||N_f)$. After that, FA forms M_{A2} : { EID, N_x, ID_f, V_1, T_1, N_y, V_2, T_2} where T_2 is a timestamp of FA and sends it to HA.

Step 3: When HA receives M_{A2}, it checks if the current time is within T_2. If not, the protocol terminates immediately. Otherwise, HA computes $N_f = h(K_{fh}) \oplus N_y$, $V_2^* = h(EID||N_x||N_y||T_2||K_{fh}||N_f)$ and then it checks if V_2^* is equal to V_2. If not, it terminates the connection. Otherwise, HA decrypts EID through $ID_M||n_0 = D_k(EID)$ and computes $K_{uh}' = h(ID_M||n_0)$. After that, it computes $V_1^* = h(EID||N_x||T_1||ID_M||K_{uh}')$ and checks if V_1^* is equal to V_1. If not, it terminates the connection. Otherwise, HA generates a random number n_1 and computes $FID = E_k(ID_M||n_1)$, $FID^* = FID \oplus h(ID_M||K_{uh}')$ and $K_{uh}^{**} = h(ID_M||n_1) \oplus h(ID_M||K_{uh}'||N_m)$. After that, it derives $N_m = h(ID_M||K_{uh}) \oplus N_x$, $N_x' = h(K_{uh}||ID_M||N_m) \oplus N_f \oplus n_0$, $N_y' = h(K_{fh}||ID_f||N_f) \oplus N_m \oplus n_0$, $V_3 = h(N_y'||N_f||T_3) \oplus K_{fh}$, and $V_4 = h(N_x'||FID^*||K_{uh}^{**}||N_m||T_3) \oplus K_{uh}'$. At last, HA forms a response message M_{A3} : { N_x', N_y', V_3, V_4, FID^*, K_{uh}^{**}, T_3 } where T_3 is a timestamp of FA and sends it to FA.

Step 4: Upon receiving M_{A3}, FA checks whether the current time is within T_3. If not, the protocol terminates immediately. Otherwise, FA computes $V_3^* = h(N_y'||N_f||T_3) \oplus K_{fh}$ and checks whether it is equal to V_3. If so, it derives $N_m \oplus n_0 = h(K_{fh}||ID_f||N_f) \oplus N_y'$ and computes the session key $SK = N_m \oplus n_0 \oplus N_f$. Finally, it sends the message M_{A4} : { N_x', V_4, FID^*, K_{uh}^{**}, T_3 } to MU.

Step 5: Upon receiving M_{A4}, SC checks whether the current time is within T_3. If not, the protocol terminates immediately. Otherwise, SC computes $V_4^* = h(N_x||FID^*||K_{uh}^{**}||N_m||T_3) \oplus K_{uh}$ and checks whether it is equal to V_4. If the verification is successful, SC computes $N_f \oplus n_0 = h(K_{uh}||ID_M||N_m) \oplus N_x'$, derives the session key $SK = N_m \oplus n_0 \oplus N_f$. After that, SC computes $FID' = FID^* \oplus h(ID_M||K_{uh})$ and $K_{uh}'' = K_{uh}^{**} \oplus h(ID_M||K_{uh}||N_m)$, and updates $EID^* = FID' \oplus h(ID_M||PSW_M)$, $K_{uh}^* = K_{uh}'' \oplus h(PSW_M||ID_M)$ and $AV = h(EID'||K_{uh}'')$ on it.

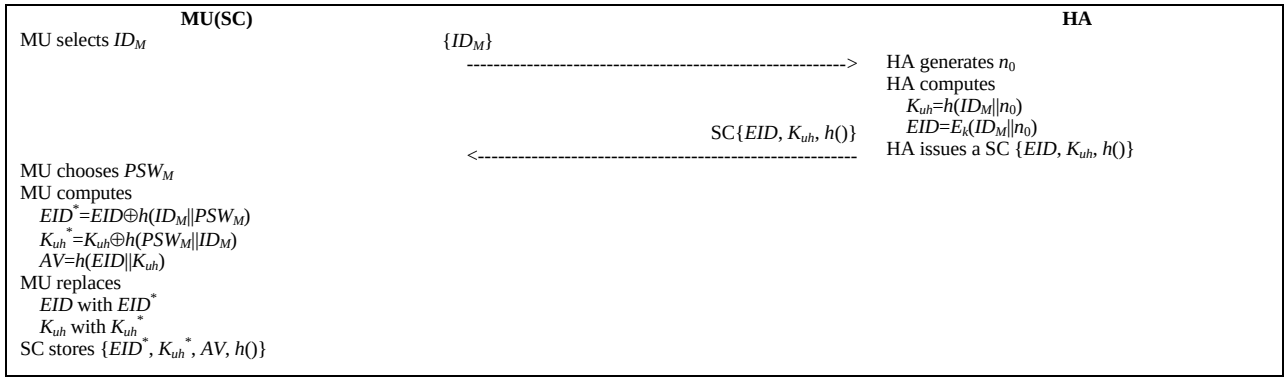

Fig. 2. The Registration Phase of Enhanced MAKA Protocol.

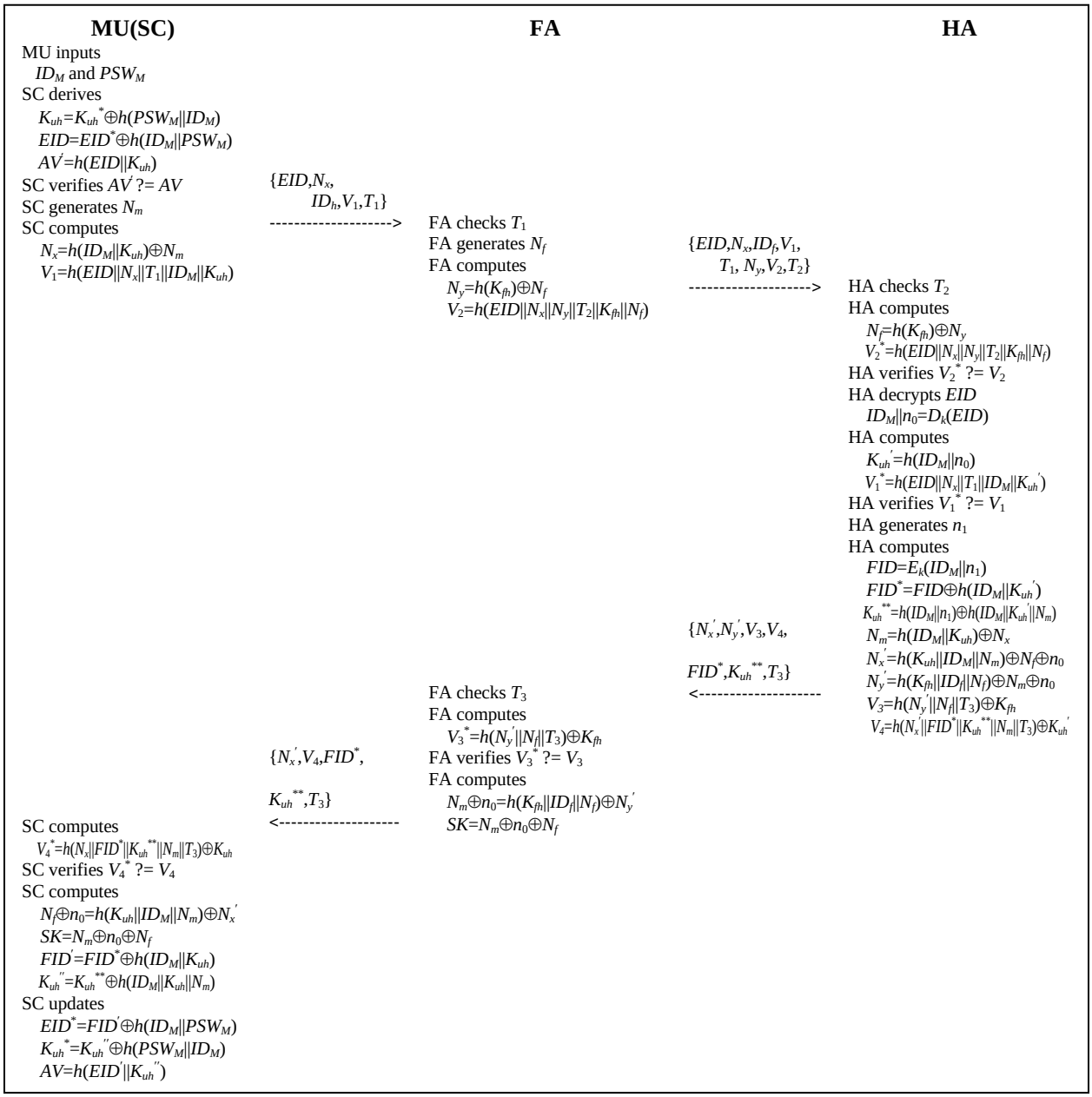

Fig. 3. The Mutual Authenticated Key Agreement Phase of Enhanced MAKA Protocol.

This phase regularly updates MU's dynamic identity and secret key between MU and HA. These features could enhance anonymity of user and security of the protocol.

C. Password Renewal Phase

MU can change his/her password without the supervision of HA. To change the password, MU needs to pass the ownership of SC first. For this, MU inputs ID_M and PSW_M to SC. SC derives $K_{uh} = K_{uh}^* \oplus h(PSW_M||ID_M)$ and $EID = EID^* \oplus h(ID_M||PSW_M)$ and computes $AV' = h(EID||K_{uh})$. If AV' is not equal to AV, SC terminates the protocol. Otherwise, SC asks MU to input a new password PSW_M^*. SC updates $EID^* = EID \oplus h(ID_M||PSW_M^*)$ and $K_{uh}^* = K_{uh} \oplus h(PSW_M^*||ID_M)$ on it.

V. Analysis

This section provides analysis of security and performance of enhanced MAKA protocol by comparing it with Gope et al.'s protocol in [14] and Xu et al.'s MAKA protocol in [15].

A. Security Analysis

The security analysis is performed based on the Dolev-Yao model with two more assumptions as described in Section 3.1. We solved the issues in Xu et al.'s MAKA protocol mentioned in Section 3.2. Unlike Xu et al.'s MAKA protocol and Gope et al.'s protocol, the proposed protocol does not need to consider the stolen verifier attack. Thereby, as shown in Table 2, the proposed enhanced MAKA protocol provides more secure and efficient properties.

[Providing Mutual Authentication] Enhanced MAKA protocol uses Challenge-Response mechanism together with timestamp. The goal of enhanced MAKA protocol is to provide mutual authentication between MU and FA. However, FA does not have direct way to authenticate MU that is the reason why it should depend on HA, which has credential relationship with MU. HA authenticates MU through V_1 by checking the possession of the correct pair of ID_M and K_{uh} and FA based on V_2 for the correctness of K_{fh}. Only the attacker with the knowledge of ID_M and K_{uh} could masquerade as a legal MU and the same for FA with K_{fh}. Furthermore, MU also authenticates FA by helping of HA based on V_4. Only the legal FA could pass the correct V_4 via HA. Addition to this, FA authenticates HA based on V_3, which only the correct HA could form it by using K_{fh}. Therefore, through the help of HA, MU and FA perform the mutual authentication since an attacker from the attack model could not do anything to masquerade any party in the proposed protocol.

TABLE II. Security Properties between Protocols

Protocol	UA[a]	MA[b]	PGA[c]	PVA[d]	PDA[e]
Gope et al.	Provide	Provide	No	No	No
Xu et al.	Provide	Provide	No	No	No
Proposed	Provide	Provide	Yes	Yes	Yes

[a.] UA: User Anonymity, [b.] MA: Mutual Authentication, [c.] PGA: Provision of Guessing Attack
[d.] PVA: Prevention of Verifier Attack, [e.] PDA: Provision of DoS Attack

[Providing Key Agreement] A fair key agreement protocol is a protocol that the session key contains the contribution of each participant. In our enhanced MAKA protocol, the session key is derived based on MU and FA's session dependent random numbers N_m and N_f together with n_0, which satisfies the fair session key agreement. MU and FA perform the key agreement via HA securely since an attacker from the attack model could not do anything to know the session key in the proposed protocol.

[Providing Anonymity of User] Since wireless network is more vulnerable to several attacks and mobile terminals' computational power is limited, anonymity in protocol design is an important issue. Anonymity is the ability of an individual to seclude himself/herself or information about himself/herself. Enhanced MAKA protocol uses pseudonym, EID, for this purpose. Furthermore, the pseudonym is dynamically chanced in each session to provide anonymity. An attacker from the attack model could not do anything to know the identity of MU in the proposed protocol.

[Prevention of Off-line Identifier and Password Guessing Attack] An attacker could get the messages, M_{A1} : $\{EID, N_x, ID_h, V_1, T_1\}$, M_{A2} : $\{EID, N_x, ID_f, V_1, T_1, N_y, V_2, T_2\}$, M_{A3} : $\{N_x', N_y', V_3, V_4, FID^*, K_{uh}^{**}, T_3\}$ and M_{A4} : $\{N_x', V_4, FID^*, K_{uh}^{**}, T_3\}$ from the communication channels. Furthermore, the attacker could get the important information on the memory of SC of MU, $\{EID^*, K_{uh}^*, AV, h()\}$. To perform the attack, the attacker needs to know ID_M and PSW_M at the same time. However, it is infeasible to the attacker due to the lack of knowledge on k or K_{uh}. Furthermore, MU's pseudonym is updated in each session. Thereby, enhanced MAKA protocol could cope from the identifier and password guessing attack even with the assumption of the usage of non-tamper resistant smart card.

[Prevention of Denial of Service Attack] The password renewal phase of enhanced MAKA protocol provides authenticity check of MU. That is the reason why an attacker with the attack model could not do anything for the denial of service attack. Only after the success of the ownership check, MU could change his/her password with a new one and update related information on SM securely. Thereby, enhanced MAKA protocol could cope from the denial of service attack.

[Prevention of Replay Attack] Enhanced MAKA protocol uses timestamp mechanism together with challenge-response mechanism to prevent replay attacks. Timestamps and random numbers could present the freshness of messages. If the current time exceeds the permitted time threshold of the received message, the message is not fresh and it means that the attacker fakes and replays it. Under this circumstance, the protocol is finished immediately. Even if the attacker could forge a valid timestamp T_i, he/she does not have the ability to forge the related V_i, which provides the integrity of message. Thereby, enhanced MAKA protocol could cope from various replay attacks.

B. Perormance Analysis

This section discusses the performance analysis by considering operational cost of the related protocols. The computational analysis of an authentication and key agreement

protocol is generally conducted by focusing on operations performed by each party within the protocols. Therefore, for analysis of the computational costs, we concentrated on the operations that are conducted by the parties in the network: namely MU, HA and FA. In order to facilitate the analysis of the computational costs, we define the following notation.

- T_h: the time to execute a one-way hashing operation
- T_x: the time to execute an XOR operation
- T_s: the time to compute a symmetric key cryptosystem operation

In addition, in order to achieve accurate measurement, we performed an experiment. This experiment was performed using the Crypto++ Library [20] on a system using the 64-bits Windows 7 operating system, 3.2 GHz processor, 4 GB memory, Visual C++ 2013 Software, the SHA-1 hash function and the AES symmetric encryption/decryption function. We summarize the results as shown in Table 3.

TABLE III. COMPUTATIONAL OVERHEAD BETWEEN PROTOCOLS

Protocol	**MU**	**FA**	**HA**	**Total**
Gope et al.	$7T_h+6T_x$	$5T_h+4T_x$	$11T_h+7T_x$	$23T_h+17T_x$
Xu et al.	$6T_h+8T_x$	$4T_h+5T_x$	$8T_h+6T_x$ $+2T_s$	$18T_h+17T_x$ $+2T_s$
Proposed	$12T_h+11T_x$	$4T_h+5T_x$	$11T_h+9T_x$ $+2T_s$	$28T_h+25T_x$ $+2T_s$

From Table 3, we could know that the proposed enhanced MAKA protocol has a bit more overheads than the other two protocols. It is mainly due to provide ownership check for SC, remove the verification table for HA and renewal of the dynamic identity to MAKA, which are the security costs.

VI. CONCLUSION

In this paper, we proposed an enhanced MAKA protocol in GLOMONET after showing the security problems in Xu et al.'s MAKA protocol. First of all, we showed a protocol flaw and three security weaknesses in Xu et al.'s protocol. The proposed enhanced MAKA protocol solved the problems in Xu et al.'s protocol efficiently by adopting ownership check, removing the verification table and renewing the dynamic identity periodically as shown in Table 2. However, it gets a bit of overhead due to the security provision functionalities as shown in Table 3.

ACKNOWLEDGMENT

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] M. Sauter, From GSM to LTE:An Introduction to Mobile Networks and Mobile Broadband, John Wiley & Sons, Ltd., 2011.

[2] Information Resources Management Association, Mobile Computing and Wireless Networks: Concepts, Methodologies, Tools, and Applications, IGI Global, 2015.

[3] H. Kim, "Data Centric Security and Privacy Research Issues for Intelligent Internet of Things," ICSES Interdisiplinary Transactions on Cloud Computing, IoT, and Big Data, vol. 1, no. 1, pp. 1-2, 2017.

[4] H. Kim, "P_PAKA : Privacy Preserving Authenticated Key Agreement Protocol in Smart Grid," International Journal of Security and its Applications, vol. 8, no. 6, pp. 17-24, 2014, doi:10.14257/ijsia.2014.8.6.02.

[5] H. Kim, "Remote User Authentication Scheme with Key Agreement Providing Forward Secrecy," Journal of Security Engineering, vol. 12, no. 1, pp. 1-12, 2015.

[6] J. Zhu and J. Ma, "A new authentication scheme with anonymity for wireless environments," IEEE Transactions on Consumer Electronics, vol. 50, no. 1, pp. 231-235, 2004, doi:10.1109/TIE.2006.881998.

[7] C. Lee, M. Hwang, and E. Liao, "Security enhancement on a new authentication scheme with anonymity for wireless environments," IEEE Transactions on Industrial Electronics, vol. 53, no. 5, pp. 1683-1687, 2006, doi:10.1109/TIE.2006.881998.

[8] C. C. Wu, W. B. Lee, and W. J. Tsaur, "A secure authentication scheme with anonymity for wireless communications," IEEE Communications Letters, vol. 12, no. 10, pp. 722-723, 2008, doi:10.1109/LCOMM.2008.080283.

[9] R. Wang, W. Juang, and C. Lei, "A robust authentication scheme with user anonymity for wireless environments," International Journal of Innovative Computing, Information and Control, vol. 5, no. 4, pp. 1069-1080, 2009.

[10] W. Jeon, J. Kim, Y. Lee, and D. Won, "Security analysis of authentication scheme for wireless communications with user anonymity," Lecture Notes in Electrical Engineering, vol. 180, pp. 225-231, 2012, doi:10.1007/978-94-007-5082-1_28.

[11] C. Chang, C. Lee, and Y. Chiu, "Enhanced authentication scheme with anonymity for roaming service in global mobility networks," Computer Communications, vol. 32, no. 4, pp. 611-618, 2009, doi:10.1016/j.comcom.2008.11.032.

[12] T. Youn, Y. Park, and J. Lim, "Weaknesses in an anonymous authentication scheme for roaming in global mobility networks," IEEE Communications Letters, vol. 13, no. 7, 417-473, 2009, doi:10.1109/LCOMM.2009.090488.

[13] T. Zhou and J. Xu, "Provable secure authentication protocol with anonymity for roaming service in global mobility networks," Computer Networks, vol. 55, no. 1, pp. 205-213, 2011, doi:10.1016/j.comnet.2010.08.008.

[14] P. Gope and T. Hwang, "Lightweight and energy-efficient mutual authentication and key agreement scheme with user anonymity for secure communication in global mobility networks," IEEE Systems Journal, vol. 10, no. 4, pp. 1370-1379, 2015, doi:10.1109/JSYST.2015.2416396.

[15] G. Xu, J. Liu, Y. Lu, X. Zeng, Y. Zhang, and X. Li, "A novel efficient MAKA protocol with desynchronization for anonymous roaming service in global mobility networks," Journal of Network and Computer Applications, vol. 107, pp. 83-92, 2018, doi:10.1016/j.jnca.2018.02.003.

[16] R. Madhusudhan and K. S. Suvidha, "An Efficient and Secure User Authentication Scheme with Anonymity in Global Mobility Networks," Proc. of 31st International Conference on Advanced Information networking and Applications Workshops, pp. 19-24, 2017, doi:10.1109/WAINA.2017.133.

[17] D. Dolev and A. C. Yao, "On the Security of Public Key Protocols," IEEE Transactions on Information Theory, vol. IT-29, no. 2, pp. 198-208, 1983, doi:10.1109/TIT.1983.1056650.

[18] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Examining smart-card security under the threat of power analysis attacks," IEEE Transactions on Computers, vol. 51, pp. 541–552, 2002, doi:10.1109/TC.2002.1004593.

[19] Cryptographic protocol, Wikipedia, https://en.wikipedia.org/wiki/Cryptographic_protocol.

[20] W. Dai, Crypto++ Library 5.6.1, Available online: http://www.cryptopp.com (accessed on 2 Jan. 2019).