SHORT COMMUNICATION

On the security of an anonymous roaming protocol in UMTS mobile networks

Shuhua Wu 1,2,*, Qiong Pu 2,3 and Ji Fu 1

1 Department of Network Engineering, Information Engineering University, Zhengzhou, China
2 State Key Laboratory of Information Security, Graduate University of Chinese Academy of Sciences, Beijing, China
3 CIMS Research Centre, Tongji University, Shanghai, China

* Corresponding author, e-mail: pqwsh@yahoo.com.cn

Received: 21 April 2011 / Accepted: 4 February 2012 / Published: 10 February 2012

Abstract: In this communication, we first show that the privacy-preserving roaming protocol recently proposed for mobile networks cannot achieve the claimed security level. Then we suggest an improved protocol to remedy its security problems.

Keywords: cryptanalysis, anonymous roaming, authentication, UMTS

INTRODUCTION

With the advancement and tremendous development of computer networks and telecommunications, user mobility has become a highly desirable network feature nowadays, especially in wireless networks (e.g. cellular networks [1-3]). This technology enables users to access services universally and without geographical limitations. In other words, they can go outside the coverage zone of their home networks, travel to foreign networks and access services provided by the latter as a visiting user or a guest. This capability is usually called roaming. Security is one of the major requirement in roaming networks. In addition to authentication, user’s privacy is equally important in such networks. To preserve this feature, not only should the user’s identity be protected (anonymity requirement), but also his location and the relation between his activities should be kept secret (untraceability requirement). The violation of either of the mentioned requisites can seriously endanger the user’s privacy. Samfat et al. [1] have proposed a comprehensive classification for different levels of privacy protection according to the knowledge of different entities about the user’s identification information. The classification is as follows:
C1: Each user is anonymous to eavesdroppers and his activities are unlinkable to them.

C2: In addition to C1, each user is anonymous to the foreign servers and his activities are unlinkable to them.

C3: In addition to C2, the relationship between the user and servers (the home server and the foreign servers) is anonymous for eavesdroppers.

C4: In addition to C3, the home server of the user is anonymous to the foreign servers.

C5: In addition to C4, each user is anonymous and his activities are unlinkable to his home server.

In the standard universal mobile telecommunication system (UMTS) [2], the home server must be always aware of the mobile user’s location in order to route the incoming calls towards the user. Moreover, the foreign server should know the identity of the home server for billing purpose. Therefore, it seems that the admissible level of privacy protection in this scenario is C3. In the last decades, several schemes addressed the privacy of users in mobile networks [3-15]. However, the most perfect and practical scheme that has been proposed so far only achieves the C2 class of anonymity and the possible C3 class has not been provided in UMTS yet. To fill this blank, Fatemi et al. [2] recently proposed a privacy-preserving roaming protocol based on hierarchical identity-based encryption (IBE) [16] for mobile networks. This protocol was claimed to achieve the acceptable C3 level of privacy. In this communication, we first show that it has some security weakness and thus the claimed security level is not achieved. Finally, we propose an enhanced protocol to remedy the existing security loopholes.

**PRELIMINARY**

In this section we recall the concept of Identity-Based Encryption (IBE) and hierarchical IBE (HIBE) schemes, upon which Fatemi et al.’s scheme builds. Here we just follow their description [2]. At first, we introduce the concept of a bilinear map between two groups, which will be used in the IBE scheme. Let $G_1$ be an additive group and $G_2$ be a multiplicative group, both of order $q$ ($q$ should be some large prime, e.g. 160 bits). We say that a map $e: G_1 \times G_1 \rightarrow G_2$ is an admissible bilinear map if all the three following conditions are satisfied:

1) $e(aP, bQ) = e(P, Q)^{ab}$ for all $a, b \in Z_q$ and $P, Q \in G_1$ (bilinear condition);

2) the map does not send all elements of $G_1 \times G_1$ to the identity element of $G_2$ (non-degeneracy condition); and

3) there is an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in G_1$ (computability condition).

Throughout this communication, the Bilinear Diffie-Hellman (BDH) in $\langle G_1, G_2, e \rangle$ is believed to be hard (i.e. it is hard to compute $e(P, P)^{abc} \in G_2$, given $\langle P, aP, bP, cP \rangle$ for some $a, b, c \in Z_q$). Since the BDH problem is not harder than the computational Diffie-Hellman (CDH) problem in $G_1$ or $G_2$, the CDH problem in $G_1$ or $G_2$ is also believed to be hard. The CDH problem in $G_1$ is as follows: given random $\langle P, aP, bP \rangle$ for $a, b \in Z_q$, compute $abP$; the CDH problem in $G_2$ is defined similarly. In addition, for a point $Q \in G_1^*$, the isomorphism $f_Q: G_1 \rightarrow G_2$ by $f_Q(P) = e(P, Q)$ is considered as a one-way function ($P$ cannot be inferred from $e(P, Q)$ and $Q$).
since an efficient algorithm for inverting $f_Q$ for some $Q$ results in an efficient algorithm for solving CDH problem in $G_2$.

Now we begin to introduce the IBE system. An IBE is a public key cryptosystem in which the public key takes any arbitrary string such as a name or an e-mail address, and the private key generator (PKG) can produce a private key corresponding to each string. Hence, one can encrypt a message by a public key even if the public key’s owner has not yet set up his private key. An efficient IBE is presented [17], which is called a Boneh-Franklin scheme. Let $P$ be a generator of $G_i$ and $s \in Z_q^*$ be the PKG’s master key. Then in the Boneh-Franklin scheme, each user’s identity-based private key should be computed as $k_U = sH_l(U)$, where $H_l : \{0,1\}^* \rightarrow G_i$ is a cryptographic hash function and $U$ is the user’s identity. Then one can encrypt a message using the public key $U$, and $U$ can decrypt the ciphertext using the private key $k_U$. The BF scheme is resistant to the chosen ciphertext attack, assuming the hardness of the BDH problem [17].

Similar to the public key cryptosystems, a hierarchy of PKGs is desirable in an IBE system to reduce the workload of the master servers. A two-level HIBE (2-HIBE) is presented [16]. There are three entities involved in a 2-HIBE scheme: a root PKG which possesses a master key $s$, the domain PKGs which gain their domain keys from the root PKG, and the users with private keys generated by their domain PKGs. The 2-HIBE scheme benefits from a linear one-way function $h : G_i \times Z_q^* \rightarrow G_i$ with the following properties:

1. For all $P \in G_i, a, x \in Z_q^*, h(aP, x) = ah(P, x)$.
2. Given $x, x_i \in Z_q^*, P \in G_i$ and $<x, h(aP, x_i)>$ for $i = 1, \cdots, n$, $h(aP, x)$ cannot be computed with any probabilistic polynomial-time algorithm.

The function $h$ defined above is a one-way function with respect to its first argument, i.e. $P$ cannot be inferred from $h(P, x)$ and $x$. Then the key for domain $S$ is $k_S = sH_l(S) \in G_i$ and the key for user $U$ in domain $S$ is $k_U = h(k_S, H_2(S \oplus U)) \in G_i$, where $H_2 : \{0,1\}^* \rightarrow Z_q^*$ is a cryptographic hash function and $\oplus$ denotes concatenation. Finally, one can encrypt a message by a public key $<S, U>$ and $U$ can decrypt the ciphertext using $k_U$.

FATEMI ET AL.’S ROAMING PROTOCOL

Review of Protocol

Here we just follow the description of Fatemi et al [2]. Like Wan et al.’s scheme [18], they also assume that a 2-HIBE is implemented in the system and the domain servers have received their private keys \{ $K_{S_i} = sH_l(S_i)$ \} from a root server. Also, they suppose that the user $U$ obtains his private key $K_U = h(K_{HS}, H_2(HS \oplus U))$ during the registration at his home domain server $HS$. In addition, a temporary key $K = e(h(h(H_1(HS), H_2(HS \oplus Nym)), H_2(HS)), sH_1(HS))$ corresponding to a pseudonym $Nym$ will be computed by the user during the roaming protocol and will be used for the authentication and key agreement purposes when he enters a foreign network domain.

As shown in Figure 1, the protocol is as follows:
Step 1. When the foreign server (FS) detects a new user in his domain, it generates a nonce $N_s$ and a random number $r_s$ (both from $Z_q^*$) and computes $r_sP$. Then it stores the values $N_s$ and $r_sP$ in his database and sends the first message including his identity $ID_{FS}, N_s$ and $r_sP$ to the user.

![Diagram](image.png)

Figure 1. Fatemi et al.’s roaming protocol [2]

Step 2. Similarly, the user $U$ generates a nonce $N_u$ and a random number $r_u$ and computes $k'_u = r_u r_sP$. Then he fetches the only unused pair of $(Nym, K)$ from his memory and computes the session key to be shared with the foreign server as $sk_u = H_4(K \oplus k'_u \oplus FS \oplus Nym \oplus N_u \oplus N_s \oplus 1)$ and a verifier $mac_u = H_4(K \oplus k'_u \oplus FS \oplus Nym \oplus N_u \oplus N_s \oplus 0)$, where $H_4$ is a hash function which maps $\{0,1\}^*$ to $\{0,1\}^l$ for some security parameter $l$. After that, he selects an arbitrary $Nym_{next}$ to be used in the next execution of the roaming protocol (either in the current FS or another FS). In order to compute the corresponding key $K_{next}$ with the help of $FS$ and $HS$, the user selects a random number $a^* \in Z_q^*$ and computes the following values: $x_1^* = h(h(a^* H_1(HS), H_2(HS \oplus Nym_{next})), H_3(HS)), \quad x_2^* = h(h(a^* H_1(HS), H_2(HS \oplus U)), H_3(HS))$. Also, he chooses random numbers $b, a_1, a_2 \in Z_q^*$ and computes $a_1 x_1^*, a_2 x_2^*$ and $E_{HS}(b, U, E(K_U, U), ID_{FS})$, where $E_S(M)$ denotes the ID-based encryption of message $M$ with the public key $S$ (e.g. $HS$ or $FS$), and $E(K_U, U)$ denotes the symmetric encryption of $U$ with the key $K_U$. Next, he sends the values $E_{FS}(Nym, ID_{HS}), N_u, r_uP, N_s, r_sP, mac_u, x_1^*, a_1 x_1^* + a_2 x_2^*$ and $E_{HS}(b, U, E(K_U, U), ID_{FS})$ to the foreign server.

Step 3. Upon receiving the above values, the foreign server checks if $N_s$ and $r_sP$ exist in its database and aborts the connection if it does not find such values. Otherwise, it decrypts $E_{FS}(Nym, ID_{HS})$ with its private key $sH_4(FS)$ and obtains the $Nym$ and $ID_{HS}$. Then it generates a random number $c \in Z_q^*$ and computes $z = h(h(c H_1(HS), H_2(HS \oplus Nym)), H_3(HS))$. Subsequently, the $FS$ sends $z$ and $E_{HS}(b, U, E(K_U, U), ID_{FS})$ to the $HS$. 

---

© 2012 Maejo Int. J. Sci. Tech. 6(01), 62-69
Step 4. The home server decrypts the message $E_{HS}(b, U, E(K_U, U), ID_{FS})$ with its private key $sH_1(HS)$ and checks whether it has received the messages from the server with the identity $ID_{FS}$. Then it authenticates the user $U$ by verifying the correctness of $E(K_U, U)$. The home server terminates the connection if any of these verifications fails. Otherwise, it computes \[ y = e(y, sH_1(HS)) (s - b)H_i(HS) = sH_i(HS) - bH_i(HS) \] and sends them back to the $FS$.

Step 5. The $FS$ computes $k'_s = r_s r_P$ and the values $K^* = y^{-1}$, $mac_u^* = H_4(K^* \| k'_s \| FS \| Nym \| N_s \| N_s \| 0)$. The $FS$ rejects the connection if the equality $mac_u = mac_u^*$ does not hold. Otherwise, it accepts $K^*$ as the user’s key corresponding to $Nym$ and authenticates the user. The computed $K^*$ together with the message $E_{HS}(b, U, E(K_U, U), ID_{FS})$ are credentials by which the foreign server will be able to request the user’s home server for service charge. Indeed, these values become a proof for payment request. In the next step the foreign server computes the session key $sk_e = H_4(K^* \| k'_s \| FS \| Nym \| N_s \| N_s \| 1)$ and the authenticator $mac_e = H_4(K^* \| k'_s \| FS \| Nym \| N_s \| N_s \| 2)$. Moreover, the foreign server calculates \[ y_1 = e(x'_1, (s - b)H_i(HS)) \] and \[ y_2 = e(a, x'_1 + a, x'_1, (s - b)H_i(HS)) \] to make the computation of $K_{next}$ feasible for the user. Finally, it returns $mac_e, y_1$ and $H_4(y_2)$ to the user.

Step 6. When the user receives the messages from the foreign server, he computes $mac_e^* = H_4(K^* \| k'_s \| FS \| Nym \| N_s \| N_s \| 2)$ and checks the equality $mac_e = mac_e^*$. If it does not hold, the user aborts the connection. Otherwise, he authenticates the foreign server and computes the following values: \[ y_1^* = y_1 \cdot e(x'_1, bH_i(HS)) \] \[ y_2^* = (y_1^*)^{a} = e(a, x'_1, -bH_i(HS)) \] and $K_{next} = (y_1^*)^{a^{-1}}$. Afterwards, the user considers whether $H_4(y_2) = H_4(y_2^*)$. If the equation holds, he accepts $K_{next}$ as the key corresponding to $Nym_{next}$. If not, he rejects the connection.

At the end of the protocol, $sk_e = sk_1$ is the key that the user and the home server have agreed upon to be used for security purpose.

Weakness of Fatemi et al.’s Protocol

We assume the adversary has totally controlled a mobile user $U$ or equivalently he has revealed the secret keys $K_U$ through side channel attacks [19]. We further assume the adversary has corrupted one of foreign networks, e.g. $FN$. Let $HS$ be the home server of $U$ and $FS$ be the server of $FN$. The adversary impersonated $U$ to visit $FN$ and initiated an execution of Fatemi et al.’s roaming protocol. He obtained from the corrupted network the message transmitted from $HA$ to $FS$ in Step 4: $(s - b)H_i(HS)$, where $b$ is the random number chosen by the adversary in Step 2. He could then compute $HS$’s secret key $K_H$ through $K_{HS} = sH_1(HS) = (s - b)H_i(HS) + bH_i(HS)$. When the adversary knows $K_{HS}$, the problem of Fatemi et al.’s roaming protocol becomes evident:

- Firstly, the adversary can reveal the real identity of any other subscriber of $HS$. When a mobile user $U (\neq U)$ performs the authentication procedure with a foreign server $FS (\neq FS)$, the adversary eavesdrops their communication and can easily get the message transmitted between...
U and FS : ID_{FS} in Step 1 and E_{HS}(b,U,E(K,U),ID_{FS}) in Step 2. Then the adversary just guesses that U is a subscriber of HS and attempts to decrypt the message E_{IFS}(b,U,E(K,U),ID_{FS}) with K_{IFS} = sH_{1}(HS). If he can retrieve ID_{FS} from the decrypted message, he confirms his guess is correct, i.e. HS = HS, because otherwise the probability that he gets any meaningful results from decryption for verification is next to zero. Then he further retrieves item U from the decrypted message and thus knows the user’s real identity U. This even contradicts the C1 security requirements. However, if U is not a subscriber of HS, his attack cannot succeed, but he will always succeed for any subscriber of HS.

• Secondly, the adversary can impersonate any other subscriber of HS (e.g. U) because he may derive K_{U} from K_{IFS} as follows: K_{U} = h(K_{IFS},H_{2}(HS\square[U])). In other words, the authentication mechanism of the protocol is completely compromised.

IMPROVED ROAMING PROTOCOL

The above demonstrated attacks show that Fatemi et al.’s protocol does not seem to achieve authentication or anonymity. In this section we present an enhanced protocol to remedy the security loopholes. As shown in Figure 2, our protocol is based on that of Fatemi et al. and it has the following changes:

- In Step 2 the computation of \( a_{1}x_{1}^{*} + a_{2}x_{2}^{*} \) is not needed any longer and finally the user U sends the values \( E_{FS}(Nym,ID_{HS}),N_{u},r,P,N_{u},r,P,mac_{u},x_{1}^{*},E_{HS}(b,U,E(K_{U},U),ID_{FS}) \) and \( E_{HS}(b,U,E(K_{U},U),ID_{FS}) \) to the foreign server.
- In Step 3 the FS also forwards \( x_{1}^{*} \) to the HS. That is, the FS sends \( z, x_{1}^{*}, E_{HS}(b,U,E(K_{U},U),ID_{FS}) \) and \( x_{1}^{*} \) to the HS.
In Step 4 the computation of \((s-b)H_i(HS) = sH_i(HS) - bH_i(HS)\) is not needed. Instead, the home server computes a new item \(w = E(K_u, e(x_i^*, sH_i(HS))\|x_i^*)\) to make the computation of \(K_{next}\) feasible for the user and finally sends \(w\) along with \(y\) back to the \(FS\).

In Step 5 the computation of \(y_1\) and \(y_2\) is not needed and the foreign server returns \(mac_x\) and \(w\) to the user.

In Step 6 the computation of \(y_1^*\) and \(y_2^*\) is not needed. After the verification of \(mac_x\) is passed and the foreign server is authenticated, the user \(U\) decrypts \(w\) using his own secret key \(K_u\) to retrieve the two items \(e(x_i^*, sH_i(HS))\|x_i^*\). If the decrypted \(x_i^*\) is the same as \(x_i^*\) computed in Step 1, he proceeds to compute \(K_{next} = (e(x_i^*, sH_i(HS)))^{(a')^{-1}}\) and accepts \(K_{next}\) as the key corresponding to \(Nym_{next}\). If not, he rejects the connection.

In our improved protocol, the item \(e(x_i^*, sH_i(HS))\) is used to make the computation of \(K_{next}\) feasible for the user. Given \(e(x_i^*, sH_i(HS))\), it is impossible for the adversary to compute \(sH_i(HS)\) since the isomorphism \(f_Q\) (here \(Q = x_i^*\)) is a one-way function. Therefore, the attacks described previously will not work any more. Although the changes introduce some computation overhead on the side of \(HS\) due to the computation of \(w\), the computation cost of \(FS\) or \(U\) is significantly reduced since both \(FS\) and \(U\) omit several costly operations (including bilinear map and exponentiation). In practice the device of the mobile user is much less powerful than the servers’. Our protocol, therefore, would be more practical.

ACKNOWLEDGEMENTS

This work was supported in part by the National Natural Science Foundation of China (Project No. 61101112) and China Postdoctoral Science Foundation (Project No. 2011M500775).

REFERENCES


© 2012 by Maejo University, San Sai, Chiang Mai, 50290 Thailand. Reproduction is permitted for noncommercial purposes.