INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS
Int. J. Commun. Syst. 2013; **26**:1028–1037
Published online 4 January 2012 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/dac.1387

Strong roaming authentication technique for wireless and mobile networks

Daojing He[1,*,†], Chun Chen[1], Sammy Chan[2] and Jiajun Bu[1]

[1]*College of Computer Science, Zhejiang University, China*
[2]*City University of Hong Kong, Hong Kong SAR, China*

SUMMARY

When one considers the broad range of wirelessly connected mobile devices used today, it is clear that integrating such network-enabled devices into secure roaming over wireless networks is of essential importance. Over the years, many authentication protocols have been suggested to address this issue. Among these protocols, the recently proposed privacy-preserving universal authentication protocol, *Priauth*, exceeds the security and efficiency of other authentication techniques. This paper studies the existing roaming authentication protocols and shows that they are not strong enough to provide secure roaming services in three aspects. Further, using Priauth as an example, we propose efficient remedies that fix the weaknesses. The experimental results show that the proposed approaches are feasible in practice. Copyright © 2012 John Wiley & Sons, Ltd.

Received 5 August 2011; Revised 6 October 2011; Accepted 28 October 2011

KEY WORDS: authentication; security and privacy; roaming service; wireless and mobile networks

1. INTRODUCTION

With the fast development of wireless technology, various wireless and mobile networks have been deployed and used in our daily life, including mobile telecommunication systems (e.g., Global System for Mobile Communications, 3G), roadside-to-vehicle communication systems, wireless local area networks (e.g., 802.11) for local area, wireless metropolitan access network (e.g., WiMAX) for wide area, and satellite network for worldwide coverage. This trend shows that the world has been turning into a ubiquitous computing environment, where people can have 'anywhere, anytime' network access services using their mobile devices (e.g., vehicle, laptop PC, personal digital assistant, and wireless phone) without being limited by the geographical coverage of their own home networks. To ensure persistent connectivity for users traveling from one network to another network, which is possibly of a different type, roaming services should be provided.

Regardless of the types of networks involved, as shown in Figure 1, a typical roaming scenario involves three parties: a roaming user U, a visited foreign server F, and a home server H of which U is a subscriber. Normally, F and H have a roaming agreement, so that U can access its subscribed services through F when U is in a foreign network administered by F. Before U can access resources provided by F, an appropriate authentication process between U and F must be carried out. This process should be efficient enough to support resource-restricted mobile devices and demanding applications, such as multimedia content delivery [1], and be secure enough because mobile users and different network providers are involved as well. Obviously, without

*Correspondence to: Daojing He, College of Computer Science, Zhejiang University, China.
†E-mail: hedaojinghit@gmail.com

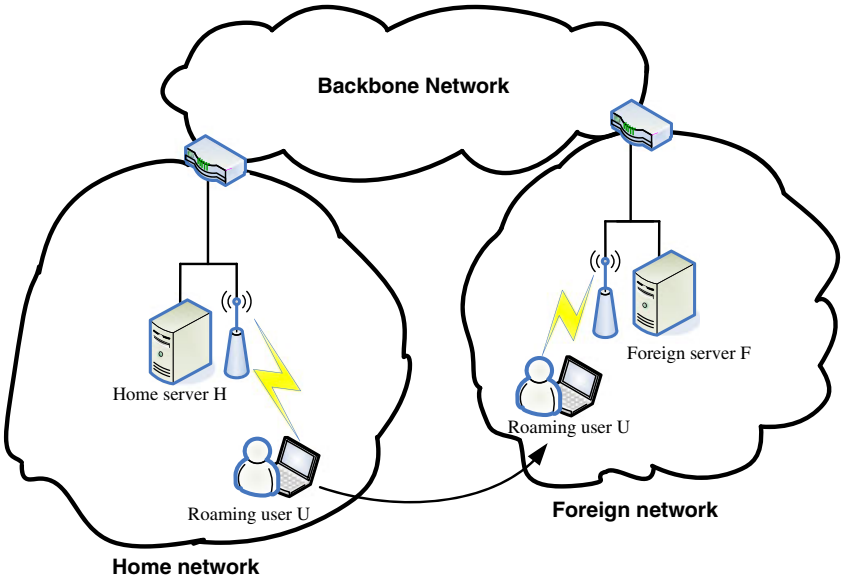

Figure 1. Roaming services overview.

appropriate security and efficiency guarantees, users and network providers are reluctant to accept roaming services.

Over the past years, many authentication protocols have been suggested for ensuring security in roaming services [2–12]. Very recently, He *et al.* [12] proposed a privacy-preserving universal authentication protocol, named *Priauth*, which exceeds the security and efficiency of other authentication techniques [2–11]. Particularly, it provides an efficient approach to tackle the problem of user revocation while supporting strong user untraceability. In this paper, we show that most previously reported studies on roaming authentication have not focused on denial-of-service (DoS) resistance, conditional privacy preservation, and supporting large-scale user revocation. Therefore, they are not strong enough to be deployed for the real-world applications without further development. Furthermore, we use Priauth as an example to illustrate some feasible approaches for enhancing the security and efficiency of these schemes. Note that there are many other desirable features (e.g., user authentication, low communication cost and computation complexity, and session key establishment) that existing authentication protocols [2–12] already possess, and thus, they are omitted in this paper.

The remainder of this paper is organized as follows. Section 2 reviews the related work. Section 3 analyzes and improves the security and efficiency of current authentication protocols. Experimental results and performance analysis of the proposed approaches are given in Section 4. Finally, Section 5 concludes the paper.

2. RELATED WORK AND REVIEW OF PRIAUTH PROTOCOL

All existing roaming authentication works can be classified into two categories, namely three-party and two-party approaches. As shown in Figure 2(a), in the three-party approach, upon receiving an access request from a roaming user U, the foreign server F sends an authentication quest to U's home server H. After receiving a response from H, F uses the secret information provided by H to perform authentication and key establishment with U. The conventional roaming authentication approaches [2–9] follow three-party roaming structure. Alternatively, as shown in Figure 2(b), the two-party approach is that without the help of H, F performs mutual authentication and session key establishment with U. Compared with the three-party roaming approach, the advantages of two-party roaming technique include the following. First, it avoids some problems such as the connection loss between the foreign server and the home server and the single point failure of the home

DOI: 10.1002/dac

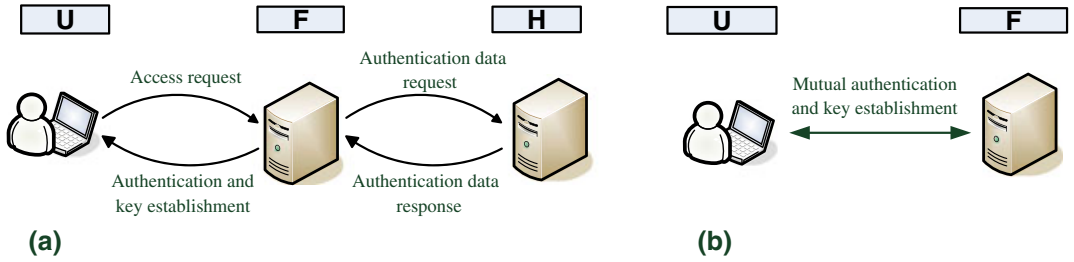

Figure 2. The structure of roaming authentication: (a) three-party roaming structure and (b) two-party roaming structure.

server, which are inevitable in the three-party roaming structure. Second, it requires less communication rounds. In the three-party roaming structure, a communication round between the foreign server and the home server is required. When the home server is many hops away from the foreign server, this communication delay becomes more crucial. These advantages together have led to the recent increasing popularity of two-party roaming authentication [10–12].

We now briefly review the Priauth of He *et al.* [12]. It is built on the verifier-local revocation group signature with backward unlinkability (VLR-GS-BU) technique. Let $\mathbb{G}$ be a cyclic additive group of order p, where p denotes a large prime number. Consider that there are multiple servers, each server manages a set of subscribers and each subscriber could be a roaming user. In the system setup phase, as the group manager, each server randomly selects a generator $g \in \mathbb{G}$ and then sets integers $N, T \in \mathbb{N}$ indicating the number of subscribers (i.e., users) and the number of time intervals, respectively. Through the VLR-GS-BU technique, each server has a master public key mpk, a vector of N subscribers' secret keys $usk = (usk[1], \ldots, usk[N])$, and a vector of $N \times T$ revocation tokens $urt = (urt[1][1], \ldots, urt[1][T], urt[2][1], \ldots, urt[2][T], \ldots, urt[N][1], \ldots, urt[N][T])$, where $urt[i][j]$ denotes the revocation token of user U_i at time interval j. The master public key mpk of each server is publicly known to all other servers. For each subscriber of a server H, say U_i, U_i secretly obtains a user secret key $usk[i]$ from H during the registration phase whereas the vector of $N \times T$ revocation tokens is kept by H. Here, H is called the home server of the subscriber U_i. Each server also has a signing/verification key pair (sk, pk) of a conventional digital signature method, for example, ECDSA [14]. Additionally, each server can set the interval unit as τ, for example, one day. Thus, at the beginning of each day, say j, all servers except H download the latest revocation list (RL) $RL_j = \{urt[k_1][j], \ldots, urt[k_i][j], \ldots, urt[k_l][j]\}$ from H, where $1 \leqslant k_i \leqslant N$. Thus, F can look up the RL to find out whether a roaming user is revoked or not without actually knowing who the roaming user is, and the whole process can be performed without any real-time involvement of H. Also, the ID and pk of each server are publicly known to all the users who are within the network controlled by the server. This could be realized by adding these information into *beacon messages* that are periodically broadcasted to declare service existence. Next, we describe the protocol that is carried out between a roaming user U_i (whose home server is H) and a visited foreign server F.

(1) U_i first chooses a random number $R_u \in \mathbb{G}$, and a temporary identity $alias$, and generates a group signature σ_U on message $(mpk_H, usk[i], j, H \| F \| alias \| R_u \cdot g \| ts)$ and then sends access request message $\{H, alias, R_u \cdot g, ts, \sigma_U\}$ to F. Here, a timestamp ts is added by U_i to counter replay attacks. Note that instead of signing a message m directly, most signature schemes should sign the one-way hash result of m. For simplicity, we will ignore the one-way hash function throughout this paper.

(2) Upon receiving the message, F checks whether U_i is a subscriber of H and then checks whether U_i is revoked by H. If the result is negative, F rejects it; otherwise, F chooses a random number $R_F \in \mathbb{G}$ and computes $\sigma_F = ECDSA.Sig(sk_F, m_F)$, where $m_F = H \| F \| alias \| R_u \cdot g \| R_F \cdot g$. Then F sends $\{F, R_F \cdot g, \sigma_F\}$ back to U_i. Subsequently, F computes the session key $SK = R_F \cdot (R_u \cdot g)$ and erases R_F from its memory.

DOI: 10.1002/dac

(3) After receiving $\{F, R_F{\cdot}g, \sigma_F\}$, U_i verifies σ_F by running ECDSA.Ver(pk_F, m_F, σ_F). If ECDSA.Ver returns 1, U_i generates the session key $SK = R_u{\cdot}(R_F{\cdot}g)$ and erases R_u from its memory. After that, U_i generates $(H\|F\|alias\|R_u{\cdot}g\|R_F{\cdot}g)_{SK}$ and then sends it to F. Here, $(X)_K$ indicates encrypting a message X using a symmetric key K. After receiving the message, F decrypts and then verifies it. If the message is valid, F concludes that U_i has established a session key; otherwise, F rejects the connection.

Obviously, for $\forall j \in [1, 2, \ldots, T]$, if H wants to revoke a particular user U_i, he simply puts the revocation token $urt[i][j]$ into RL_j. Otherwise, for $\forall j \in [1, 2, \ldots, T]$, if H allows U_i to access the global network, H does not put $urt[i][j]$ into RL_j. We refer the readers to He *et al.* [12] for a detailed description of Priauth.

3. SECURITY ANALYSIS AND IMPROVEMENT

As reported in [3], designing a secure roaming protocol is a difficult task. There are so many details involved (e.g., the complicated interactions with the environment) that the designer can only try his best to make sure his protocol is infallible. On the other hand, until now, a simple, efficient, and convincing formal methodology for correctness analysis of the protocols of this kind is still an important subject of research and an open problem. In reality, the degree of confidence accompanying a mechanism increases with time only if the underlying algorithms can survive many years of public scrutiny. In this section, we show that existing roaming authentication techniques do not meet the three main security and efficiency properties, namely DoS-resistance, conditional privacy preservation, and supporting large-scale user revocation. Also, using Priauth as an example, we suggest some remedies to address these issues.

3.1. Denial-of-service resistance

In DoS attacks against roaming services, the adversary may flood a large number of illegal access request messages to network servers. The purpose is to exhaust their resources and render them less capable of serving legitimate users. Obviously, a practical authentication mechanism should maintain service availability despite of DoS attacks. Such attacks can be classified into three categories.

First, referring to Figure 2(a), because three-party roaming approaches [2–9] require a foreign server to unconditionally forward any access request, valid or invalid, to the home server, the adversary can easily launch the DoS attack on a home server through a foreign server. Obviously, this problem is inevitable in the three-party roaming structure. However, two-party roaming techniques only require the roaming user and the foreign server to be involved in each protocol run; the DoS attack on home servers is thus not applicable.

Second, because of some design considerations (e.g., establishing a session key or recording a cookie), some proposals (e.g., [10, 11]) require each foreign server to use a challenge-response approach with a roaming user before the foreign server authenticates the user. An adversary can easily send a large volume of forged access requests to exhaust the storage, processing, and bandwidth resources of foreign servers. Obviously, to thwart this attack, similar to Priauth described previously, upon receiving an access request message, each foreign server first verifies this message.

Third, in most roaming authentication techniques (e.g., [5–12]), for each access request message, the foreign server needs to perform expensive cryptographic operations (e.g., pairing computation in [10–12]) to check the validity of the sender. In other words, a foreign server cannot authenticate an access request message immediately after receiving it. Particularly, to enable the foreign server to locally check the validity of roaming users, some complex cryptography techniques (e.g., hierarchical identity-based encryption, identity-based signature, group signature) must be used in two-party roaming authentication techniques, which usually result in high computation overhead on foreign server. This limitation can be easily exploited by the adversary. That is, it can inject bogus access request messages into the networks, forcing the foreign servers that receive such messages to perform expensive verifications and eventually exhausting their resources. Despite the necessity

DOI: 10.1002/dac

and importance, no research has been conducted to address this attack in roaming authentication until now.

To prevent the attack, we adopt message-specific puzzle or client puzzles of [15] into current roaming authentication protocols (e.g., [5–12]). The idea is summarized as follows. When there is no evidence of such an attack, each foreign server processes access requests normally, that is, indiscriminately. However, when a foreign server comes under a suspected DoS attack, it performs expensive verification on access requests selectively. In particular, the server attaches a unique puzzle into the *beacon messages* and requires the puzzle solution to be attached in each access request message. The server commits resources to process an access request only when the solution is correct. In general, solving a puzzle requires a brute-force search in the solution space, while solution verification is very fast. Additionally, puzzles are deployed in conjunction with conventional timeouts on server resources. Thus, to create an interruption in service, an adversary must have abundant resources to be able to promptly compute a large enough number of puzzle solutions in line with his sending rate of illegal access requests. In contrast, although puzzles slightly increase legitimate users' computational load when the server is under attack, they are still able to obtain network accesses regardless the existence of the attack.

Here, we use Priauth and message-specific puzzle as an example. If a network server, say F, is not under attack, it attaches 'No' into the *beacon messages*. It indicates to the roaming users that no puzzles are being distributed and the Priauth protocol is executed normally. On the other hand, if F is under attack, it adds 'Yes' and a puzzle (i.e., a timestamp T_F, a random number a and an integer l) into the *beacon messages*. To initiate a connection with F, a roaming user must solve the puzzle within a specified time interval. A valid solution S is such a value that after applying the hash function $h()$ to $(H\|alias\|R_u{\cdot}g\|\sigma_U\|T_F\|a\|S)$, the first l bits of the resulting image are all '0', as illustrated in Figure 3. The parameter l determines the strength of the puzzle. Before transmitting the access request message, a user first tries to solve the puzzle by finding the puzzle solution S. Subsequently, the user sends the final access request message $\{H\|alias\|R_u{\cdot}g\|\sigma_U\|T_F\|a\|S\}$ to F. Obviously, the puzzle solution in every access request can be efficiently verified by F via a hash function operation and comparison. Only if this verification is successful, F performs expensive verification on the access request.

Note that in some application scenarios, the adversary and a legal mobile user may be incomparable in computation power. To prevent a powerful adversary, the parameter l should be considerably large; however, this would bring extra burden to the low-power mobile devices in a normal authentication process. To solve this issue, we propose some modifications on the aforementioned message-specific puzzle. For current roaming authentication protocols (e.g., [5–12]), in system initiation phase, each network server chooses a random number as the puzzle key of himself and then distributes the puzzle key to the other servers and its subscribers using a secure transmission protocol (e.g., wired transport layer security protocol). When a foreign server comes under a suspected DoS attack, the server requires the solution of the puzzle key of the corresponding home server to be attached in each access request message. The server commits resources to process an access request only when the solution is correct. More specifically, this example is modified as follows. If F is under attack, it adds 'Yes', a timestamp T_F, and an integer l into the *beacon messages*. A valid solution S is such a value that after applying the hash function $h()$ to $(H\|alias\|R_u{\cdot}g\|\sigma_U\|T_F\|k\|S)$,

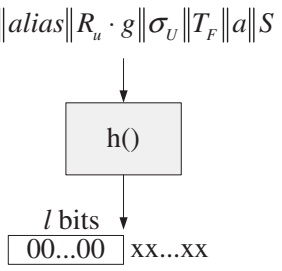

Figure 3. Message-specific puzzle of Priauth.

DOI: 10.1002/dac

the first l bits of the resulting image are all '0', where k is the puzzle key of server H. Before transmitting the access request message, with the knowledge of the puzzle key k, a user first tries to solve the puzzle by finding the puzzle solution S. Subsequently, the user sends the final access request message $\{H\|alias\|R_u{\cdot}g\|\sigma_U\|T_F\|S\}$ to F. Upon receiving the message, F uses the puzzle key k to check the validity of the message via a hash function operation and comparison. On the other hand, an adversary first finds the puzzle key through a brute-force search in the key space and then solves the puzzles. This is because if an adversary keeps sending a large number of bogus access request messages, the visiting server will trace such an adversary. Compared with the previous message puzzle, here the modified message-specific puzzle can more effectively mitigate DoS attacks. This is because an authorized user has a clear advantage over the adversary because of the prior knowledge of the puzzle key.

Note that the first two attacks were already addressed in [12], and so the attacks are not the main contributions of this paper. One main contribution of this paper is to show and resist the third attack.

3.2. Two-party roaming authentication with conditional privacy preservation

The anonymous roaming authentication in wireless networks should be conditional, such that the home server (or legitimate law enforcement authorities) can find a way to track a targeted user and collect its privacy information (e.g., user identity and current location), even though the user is not traceable by the public. To our knowledge, until now only three two-party roaming authentication protocols have been presented [10–12]. However, we observe that all of them do not achieve conditional privacy preservation. To solve this issue, for the schemes of [10–12], an ideal approach is that according to the roaming agreement, the visited foreign server notifies the home server of the authentication result after performing roaming authentication. Because this step is carried out after the authentication process, it does not affect the authentication time. Upon receiving the authentication result, the home server can use its secret key to obtain the privacy information of the user. Here, we use Priauth as an example. Once the foreign server F finishes roaming authentication, it forwards the access request message $\{H, alias, R_u{\cdot}g, ts, \sigma_U\}$ to the home server H. For the purpose of nonrepudiation, F uses its signing key to sign on the access request under ECDSA [14]. Suppose that H wants to track a user (say U_i). With the knowledge of all revocation tokens $\{urt[i][1], \ldots, urt[i][j], \ldots, urt[i][T]\}$ of the user, H takes over all group signatures σ_U $(= (T_1, T_2, T_3, T_4, V))$ from mobile users during the time interval j and verifies them by checking the Signature from zero-knowledge Proofs of Knowledge (SPK) V and $T_3 = e(T_4, urt[i][j])$. If it is valid, it can be concluded that the signature is signed by the traced user. Thus, H can trace the user across server authentication boundaries. For simplicity, the detailed description of SPK and the computation of the group signature (T_1, T_2, T_3, T_4, V) are omitted in this paper. The reader can refer to He *et al.* [12].

3.3. Supporting large-scale user revocation

To achieve two-party roaming authentication, group signature techniques have been introduced into the design of the protocols [11, 12]. However, we observe that they cannot support large-scale user revocation. The detailed analysis is given as follows. When a user U_i is revoked for interval j because of some reasons (e.g., expiration of service subscription), the corresponding home server simply adds U_i's trace key tk_i of Yang *et al.* [11] (i.e., revocation token $urt[i][j]$ of He *et al.* [12]) to RL, and the updated RL is downloaded by the foreign servers. Note that the size of RL is linear to the accumulated number of trace keys being revoked, which can potentially grow fairly large as time elapses. That is, the delay incurred in these protocols to verify each access request is linearly proportional to the number of revoked users. Therefore, these protocols may not achieve good performance in a large-scale network, where the number of revoked users may be large.

As stated in [13], user revocations are mainly due to two reasons: one is expiration of service subscription and the other is violation of network access policy. Because of the nature of the network access service, user revocations due to the former reason usually happen periodically and are prescheduled; this is the major reason causing the size growth of RL. On the other hand, user revocations due to the latter is often random and sporadic. With this observation, a hybrid membership

DOI: 10.1002/dac

maintenance approach can be employed in Priauth to minimize the size of RL. In the system setup phase, each server can set the minimum subscription period of the network service as δ time unit, for example, 1 month. For the duration of each minimum subscription period, each server generates a new master public key mpk, a new vector of N subscribers' secret keys, and a new vector of $N \times \frac{\delta}{\tau}$ revocation tokens. Also, the server arranges the usage of these keys and revocation tokens in a sequential manner. That is, the server delivers the current mpk to other servers. A mobile user who subscribes the network service for $x\delta$ time units through the corresponding home server will obtain x secret keys. Each of these secret keys will only be valid for δ time unit and expires automatically afterward. Additionally, if a user is revoked because of the violation of network access policy, the home server simply follows the procedure previously described to update RL. Now the size of RL will not grow very large because RL does not involve the revocation of the users whose service subscriptions are expired. Because the server can generate the secret keys for each user offline (or in the system setup phase) and often the server is resource rich, the overhead for generating the secret keys for each user can be omitted.

On the basis of the aforementioned analysis, Table I summarizes the functionality of the proposed improvement and make comparisons with that of related works [2–12].

4. PERFORMANCE AND IMPLEMENTATION

As described in Section 3, only the first and third properties need real experiments to show that they are feasible in practice.

4.1. Denial-of-service resistance

We implement the message-specific puzzle and the modified message-specific puzzle and incorporate them into Priauth to show their efficiency in practice. In our implementation, we use the Miyaji–Nakabayashi–Takano (MNT) family of curves [17], where p is 170 bits and elements of G are 171 bits. Thus, the length of the total group signature σ_U is 362 bytes. Note that such key lengths are considered secure enough for now and immediate future. Here, the lengths of $H, alias, T_F, a$, and S are set to 1, 2, 4, 2, and 4 bytes, respectively. We perform the same experiment five hundred times and take an average over them.

Table II gives the time required to solve a message-specific puzzle on a Linux PC with 3.1 GHz processor and 4 GB memory (using OpenSSL library [16]) when the parameter l varies. For example, the time required to solve a message-specific puzzle is 6.79 s when the parameter l is set to 22. As previously described, this time consumption is enough to defeat DoS attacks. Although puzzles slightly increase legitimate users' computational load when the server is under attack, they are still able to obtain network accesses regardless the existence of the attack.

Table I. Functionality comparison between the related protocols and our proposal.

Protocols	DoS resistance	Conditional privacy preservation	SLSUR
Three-party approaches [2–9]	No	Yes	Yes
Proposed improvement on three-party approaches [2–9]	No	Yes	Yes
The protocol of Wan *et al.* [10]	No	No	Yes
Proposed improvement on protocol of Wan *et al.* [10]	Yes	Yes	Yes
Protocols of Yang *et al.* [11] and He *et al.* [12]	No	No	No
Proposed improvement on protocols of Yang *et al.* [11] and He *et al.* [12]	Yes	Yes	Yes

DoS, denial-of-service; SLUR, supporting large-scale user revocation.

DOI: 10.1002/dac

Table II. The execution time for solving a message-specific puzzle in Priauth.

The parameter l	12	14	16	18	20	22	24
The execution time (s)	0.0069	0.0272	0.1060	0.4168	1.7184	6.7851	27.571

Table III. The execution time for solving a modified message-specific puzzle with knowing the puzzle key.

The parameter l	10	11	12	13	14	15
The execution time (ms)	1.625	3.108	6.766	13.758	23.350	35.733

Table IV. The execution time for solving a modified message-specific puzzle without knowing the puzzle key.

The parameter l	10	11	12	13	14	15
The execution time (s)	1.638842	5.043591	9.562766	12.205008	13.079033	13.287425

Tables III and IV give the time required to solve a modified message-specific puzzle on a Linux PC with 3.1 GHz processor and 2 GB memory (using OpenSSL library [16]) for legitimate users and adversaries, respectively. In our implementation, the lengths of the puzzle key k and solution S are set to 1 and 2 bytes, respectively. We perform the same experiment 1200 times and take an average over them. According to these results, it is clear that compared with legitimate users, the adversary needs much more time to find the puzzle key through a brute-force search in the key space before solving puzzles. For example, when the parameter l is set to 13, the time required to solve a modified message-specific puzzle are 13.8 ms and 12.2 s for legitimate users and adversaries, respectively. Obviously, this time consumption is enough to effectively defeat DoS attacks.

4.2. Two-party roaming authentication with conditional privacy preservation

To evaluate our proposed approach, the home server side programs have been implemented in C and executed in laptop PCs with different computational power (a single CPU). The running time of Pairing operation using PBC [20] library and Elliptic Curve Scalar Multiplication (ECSM) operation using MIRACL [21] library are summarized in Table V. We perform the same experiment 10,000 times and take an average over them. In our implementation, we use MNT curve [17] with order of 160 bits and embedding degree $d = 6$. To trace a user, verifying a group signature takes the home server 7.5 ECSM and two Pairing operations (plus four pairing operations that can be pre-computed). We assume the home server runs on a 1.6 GHz PC; thus, it takes 14.6 ms (plus 15.4 ms pre-computed). Moreover, the home server is often a powerful server (i.e., mainframe). Therefore, the proposed approach is efficient to be employed on most home servers.

He *et al.* [12] has shown that Priauth exceeds the efficiency of computation and communication of other authentication techniques [2–11]. As described in Section 3, when there is no evidence of DoS attacks, the computation and communication complexity of the proposed improved version on Priauth is the same as that of Priauth.

Table V. Running time of ECSM and Pairing operations.

	800 MHz processor		1.2 GHz processor		1.6 GHz processor		2 GHz processor	
	ECSM	Pairing	ECSM	Pairing	ECSM	Pairing	ECSM	Pairing
Time (ms)	1.83	5.734	1.547	3.841	0.916	2.872	0.672	2.134

ECSM, Elliptic Curve Scalar Multiplication.

DOI: 10.1002/dac

5. CONCLUSION

In this paper, we have presented a security analysis of current roaming authentication techniques. The analysis has led us to pinpoint critical security and efficiency weaknesses in them. We have then suggested some simple and efficient patches that fix the vulnerabilities. The experimental results show that the proposed approaches are feasible for real applications.

ACKNOWLEDGEMENTS

This work was supported by the National Science Foundation of China (Grant No. 61070155), the Program for New Century Excellent Talents in University (NCET-09-0685), and a grant from the Research Grants Council of the Hong Kong SAR, China (Project No. City U 111208).

REFERENCES

1. AbuAli N, Hayajneh M, Hassanein H. Congestion-based pricing resource management in broadband wireless networks. *IEEE Transactions on Wireless Communications* 2010; **9**(8):2600–2610.
2. European Telecommunications Standards Institute (ETSI), GSM 02.09: Security Aspects, 1993.
3. He D, Ma M, Zhang Y, Chen C, Bu J. A strong user authentication scheme with smart cards for wireless communications. *Computer and Communications* 2011; **34**(3):367–374.
4. Chang C-C, Lee C-Y, Chiu Y-C. Enhanced authentication scheme with anonymity for roaming service in global mobility networks. *Computer and Communications* 2009; **32**(4):611–618.
5. Zhu J, Ma J. A new authentication scheme with anonymity for wireless environments. *IEEE Transactions on Consumer Electronics* 2004; **50**(1):230–234.
6. Lee CC, Hwang MS, Liao IE. Security enhancement on a new authentication scheme with anonymity for wireless environments. *IEEE Transactions on Industrial Electronics* 2006; **53**(5):1683–1687.
7. Wu CC, Lee WB, Tsaur WJ. A secure authentication scheme with anonymity for wireless communications. *IEEE Communications Letters* 2008; **10**:722–723.
8. Yang G, Wong DS, Deng X. Anonymous and authenticated key exchange for roaming networks. *IEEE Transaction on Wireless Communications* 2007; **6**(9):3461–3472.
9. Tang C, Wu DO. An efficient mobile authentication scheme for wireless networks. *IEEE Transaction on Wireless Communications* 2008; **7**(4):1408–1416.
10. Wan Z, Ren K, Preneel B. A secure privacy-preserving roaming protocol based on hierarchical identity-based encryption for mobile networks. *Proceedings of ACM WiSec '08*, 2008.
11. Yang G, Huang Q, Wong DS, Deng X. Universal authentication protocols for anonymous wireless communications. *IEEE Transaction on Wireless Communications* 2010; **9**(1):168–174.
12. He D, Bu J, Chan S, Chen C, Yin M. Privacy-preserving universal authentication protocol for wireless communications. *IEEE Transaction on Wireless Communications* 2011; **10**(2):431–436.
13. Ren K, Yu S, Lou W, Zhang Y. PEACE: a novel privacy-enhanced yet accountable security framework for metropolitan wireless mesh networks. *IEEE Transactions on Parallel and Distributed Systems* 2010; **21**(2):203–215.
14. ANSI X9.62. Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA), 1999.
15. Juels A, Brainard J. Client puzzles: a cryptographic countermeasure against connection depletion attacks. *Proceedings of NDSS '99*, 1999.
16. OpenSSL. http://www.openssl.org.
17. Miyaji A, Nakabayashi M, Takano S. New explicit conditions of elliptic curve traces for FR-reduction. *IEICE Transactions on Fundamentals* 2001; **E84-A**(5):1234–123.
18. Zeng K, Govindan K, Mohapatra P. Non-cryptographic authentication and identification in wireless networks. *IEEE Wireless Communications* 2010; **17**(5):56–62.
19. Pang J, Greenstein B, Gummadi R, Seshan S, Wetherall D. 802.11 user fingerprinting. *Proceedings of MobiCom '07*, 2007; 99–110.
20. Pairing based cryptography benchmarks. (Available from: http://crypto.stanford.edu/pbc/times.html).
21. Michael S. Multiprecision Integer and Rational Arithmetic C/C++ Library (MIRACL). Published by Shamus Software Ltd., http://www.shamus.ie/.

DOI: 10.1002/dac

AUTHORS' BIOGRAPHIES

Daojing He received his BEng and MEng degrees in Computer Science from the Harbin Institute of Technology in 2007 and 2009, respectively. He is currently a PhD student in Zhejiang University, China. His research interests include network and systems security with focuses on wireless network security. He serves as TPC for IEEE Globecom 2011, IEEE PIMRC 2011, IEEE WCNC 2012, IEEE ICC 2012, and so on.

Chun Chen received the bachelor's degree in Mathematics from Xiamen University, China, in 1981, and the master's and PhD degrees in Computer Science from Zhejiang University, China, in 1984 and 1990, respectively. He is a professor in the College of Computer Science and the director of the Institute of Computer Software at Zhejiang University. His research activity is in image processing, computer vision, and embedded system.

Sammy Chan received his BE and MEngSc degrees in Electrical Engineering from the University of Melbourne, Australia, in 1988 and 1990, respectively, and his PhD degree in Communication Engineering from the Royal Melbourne Institute of Technology, Australia, in 1995. From 1989 to 1994, he was with Telecom Australia Research Laboratories, first as a research engineer and between 1992 and 1994 as a senior research engineer and project leader. Since December 1994, he has been with the Department of Electronic Engineering, City University of Hong Kong, where he is currently an associate professor.

Jiajun Bu received the BS and PhD degrees in Computer Science from Zhejiang University, China, in 1995 and 2000, respectively. He is currently a professor in the College of Computer Science and the deputy dean of the Department of Digital Media and Network Technology at Zhejiang University. His research interests include embedded system, mobile multimedia, and data mining.

DOI: 10.1002/dac