Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 858579, 11 pages
http://dx.doi.org/10.1155/2013/858579

*Research Article*

# Authenticated Blind Issuing of Symmetric Keys for Mobile Access Control System without Trusted Parties

**Shin-Yan Chiou**

*Department of Electrical Engineering, Chang Gung University, 259 Wen-Hwa 1st Road, Kwei-Shan, Taoyuan 333, Taiwan*

Correspondence should be addressed to Shin-Yan Chiou; ansel@mail.cgu.edu.tw

Received 12 April 2013; Revised 25 May 2013; Accepted 26 May 2013

Academic Editor: Wang Xing-yuan



Mobile authentication can be used to verify a mobile user's identity. Normally this is accomplished through the use of logon passwords, but this can raise the secret-key agreement problem between entities. This issue can be resolved by using a public-key cryptosystem, but mobile devices have limited computation ability and battery capacity and a PKI is needed. In this paper, we propose an efficient, non-PKI, authenticated, and blind issued symmetric key protocol for mobile access control systems. An easy-to-deploy authentication and authenticated key agreement system is designed such that empowered mobile devices can directly authorize other mobile devices to exchange keys with the server upon authentication using a non-PKI system without trusted parties. Empowered mobile users do not know the key value of the other mobile devices, preventing users from impersonating other individuals. Also, for security considerations, this system can revoke specific keys or keys issued by a specific user. The scheme is secure, efficient, and feasible and can be implemented in existing environments.

## 1. Introduction

Authentication enables authorized persons to use specific services. A password authentication scheme to achieve user authentication [1] was first proposed in 1981. Later, several schemes [2–6] were proposed to remedy some of the security weakness in [1] and many password-based authentication schemes [7, 8] have since been proposed. Besides authentication, key establishment protocols are an important cryptographic primitive. The first unauthenticated key agreement protocol based on asymmetric cryptographic techniques was proposed by Diffie and Hellman [9, 10].

The rapid development of electronic technologies has resulted in various mobile devices which increase the convenience of everyday tasks, and an increasing number of applications for mobile devices are now used on the Internet or in wireless networks, raising the importance of mobile authentication in insecure channels. Many authentication protocols for wireless networks have been proposed. Some of these protocols are used in portable communication systems (PCSs) such as the global system for mobile (GSM) protocol [11], maintain the architecture of GSM (MGSM) protocols [12–16], and public-key system protocols [17–20].

Generally, mobile authentication [21–24] can be implemented via traditional public-key cryptography [6, 13]. However, mobile devices suffer from limited computation and battery capacity. Thus, traditional public-key cryptograph, which requires the computation of modular exponentiation, cannot be widely used in mobile devices.

Compared with other public-key cryptography methods, the elliptic curve cryptosystem (ECC) [4, 17, 25, 26] has significant advantages including smaller key sizes and faster computation, making ECC-based authentication protocols [27, 28] more suitable for use in mobile devices.

However, like other public-key cryptography methods, ECC also needs a public key infrastructure (PKI) to maintain certificates for users' public-keys. As the number of users increases, PKI needs a large storage space to store the users' public keys and certificates.

In addition, implementing an authenticated key agreement protocol requires the corresponding party's authenticated public key. For example, for Alice and Bob to execute the NIST recommended MQV key agreement protocol [14, 29], Alice needs an authenticated public key $PK_B$ from Bob and Bob needs an authenticated public key $PK_A$ from Alice.

Given the difficulty of deploying a PKI system, easy-to-deploy authentication and authenticated key agreement systems are preferable, such as identity-based cryptosystems and key agreement systems.

Sakai et al. [30] proposed an identity ID-based public-key cryptosystem. Unlike traditional certificate-based public-key systems, ID-based public-key cryptosystems do not need to store users' public keys and certificates, thus simplifying certificate management. However, the user's private key must be generated by the Key Generator Center.

On the other hand, several identity-based key agreement protocols [17, 30–36] have been proposed. Smart [35], Chen and Kudla [32], Sakai et al. [30], Shim [34], and McCullagh and Barreto [17] designed identity-based and authenticated key agreement protocols based on Weil and Tate pairing techniques. However, Chen and Kudla [32] pointed out that Smart's protocol is not secure in several respects. Cheng et al. [26] showed that Chen-Kudla's protocol is not secure against unknown key share attacks. Scott's protocol is not secure against man-in-the-middle attacks. Sun and Hsieh [37] pointed out that Shim's protocol is insecure against key compromise impersonation attacks or man in the middle attacks. Choo [38] showed that McCullagh and Barreto's protocol is insecure against key revealing attacks. Although McCullagh and Barreto [17] revised their protocol, the revised protocol does not achieve the weak perfect forward secrecy property. Therefore, most identity-based key agreement protocols are impractical or fail to meet all required security properties.

Wang and Zhao [39] proposed an enhanced key agreement protocol based on the semigroup property of the Chebyshev chaotic map. Their method is similar to the Diffie-Hellman algorithm. Compared with other key agreement protocols based on chaos [40–44] their method provides not only mutual authentication but also security resistance to replay attacks, man-in-the-middle attacks, and Bergamo et al.'s attack [45]. Later, Niu and Wang [46] proposed an anonymous key agreement protocol based on chaotic maps to improve the security of Tseng et al.'s protocol [47].

In 2010, Sun and Hsieh [37] present a client authentication and key exchange protocol using bilinear pairings for mobile client-server environments. Although both mutual authentication and key exchange are provided, the relative computation cost of pairing is approximately 20 times higher than that of the scalar multiplication [13]. Therefore it is not sufficiently efficient for mobile client server environments.

Haist and Osten [48] proposed a secure key distribution method using classical light. Between communicating parties, this method relies on interferometric measurements of white light (by one of the parties) and the random choice of delays (by the others). Their system requires a third party, and all the communicating parties must ensure their devices are appropriately equipped and that measurements are conducted at a particular time and place. In 2012, Álvarez-Bermejo et al. [49] proposed a key multicasting protocol based on the use of orthogonal systems in vector spaces. Their protocol helps a server facing a huge number of users to exchange a small number of messages.

This paper describes the design of an easy-to-deploy authentication and authenticated key agreement system such that empowered mobile devices can directly authorize other mobile devices to exchange authenticated keys with the server using a non-PKI system. We discuss the problem of *key reproducing*, which means a new authenticated key can be directly generated from another empowered key. New authenticated keys should be different from the empowered key to prevent spoofing.

We address the problem of mobile access key issuing (key reproduction) from other mobile keys for role-based access control models, which is essential in the daily authenticating environment. For example, mobile device can be used to unlock electronic door locks or to access computers. A homeowner can directly issue an access key to his parents, friends, or guests. An empowered mobile device $D_A$ can reproduce new authenticated keys to other devices $D_F$ or $D_U$ so that (1) $D_F$ and $D_U$ can pass system verification, (2) the system can precisely identify who ($D_A$, $D_F$, or $D_U$) is accessing the system, and (3) the key issuer $A$ does not know the values of the keys issued to $D_F$ or $D_U$. Our scheme provides a secure key-reproducing method which does not require a third party.

We design a *master key* and a *visitor key*. A *master key* owner is empowered to issue keys to another user, where the issued key can be a *master key* or a *visitor key*. The *newly issued master key* is permitted to issue other keys while a visitor key is not. For authentication purpose, the system database keeps some authentication data. Specifically, we provide a *storage-free* choice for visitor keys, such that the authentication data of the visitor keys do not have to be stored in the system database. For security considerations, this system can revoke specific keys or keys issued by a specific person.

The contributions of the proposed system are as follows. The proposed system

(1) authorizes mobile devices to directly exchange authenticated keys with the server from empowered mobile devices;

(2) ensures key issuers are blind to the secret keys provided to key requesters;

(3) reduces requirements for trusted parties;

(4) reduces requirements for public-key cryptography or public-key infrastructure (PKI);

(5) provides confidentiality of key values among mobile devices;

(6) allows for the revocation of specific keys or keys issued by a specific user;

(7) provides a *storage-free* choice to reduce database storage requirements;

(8) reduces vulnerability to replay attacks and man-in-the-middle attacks;

(9) can be implemented in existing environments.

The rest of this paper is organized as follows. In Section 2, we provide a technical description of the proposed protocol including notations, model, protocol goals, and security requirements. The construction details of the proposed protocol are presented in Section 3. Security, efficiency, correctness analysis, and property comparison for the proposed protocol are given in Section 4 and we draw conclusions in Section 5.

## 2. Preliminaries

Our scheme assumes two grades of keys, a *master key* ($\mathcal{MK}$) and a *visitor key* ($\mathcal{VK}$). We also assume there are two rules, a *key owner* and a *key verifier*. A *key owner* possesses either $\mathcal{MK}$ or $\mathcal{VK}$, which passes the authentication of *key verifiers*. A *key verifier* is a guard system (such as electronic door locks or computers) that owns its database. Each $\mathcal{MK}$ or $\mathcal{VK}$ has a validity time. Only $\mathcal{MK}$ is authorized to generate another new $\mathcal{MK}$ or $\mathcal{VK}$ for other users.

For a general scenario, we define three roles: *administrator* ($\langle\mathcal{A}\rangle$), *super user* ($\langle\mathcal{S}\rangle$), and *normal user* ($\langle\mathcal{U}\rangle$). We assume both the administrator and the super users have $\mathcal{MK}$ and the normal users hold $\mathcal{VK}$. Although all key owners have their own validity times, the validity time of the administrator is designed to never expire. Furthermore, we design a *short-term visitor key authentication* (SVA) which is used for rarely used $\mathcal{VK}$s. This reduces system storage space requirements because the $\mathcal{VK}$s' data is not stored in the system database.

*2.1. Notations, Model, and Protocol Goals.* In this section we first list notations, our model, and our protocol goals. The notations are shown in Table 1. The model and the protocol goals are shown in Table 2. The scheme has two rules, *key owners* and *key verifier* (i.e., *the guard system*). Key owners can be administrator ($\langle\mathcal{A}\rangle$), super user ($\langle\mathcal{S}\rangle$), and normal user ($\langle\mathcal{U}\rangle$). The key grades of ($\langle\mathcal{A}\rangle$) and ($\langle\mathcal{S}\rangle$) are $\mathcal{MK}$ and the key grade of $\langle\mathcal{U}\rangle$ is $\mathcal{VK}$. Each user $\mathcal{U}_i$ in any status $\langle\mathcal{A}\rangle$, $\langle\mathcal{S}\rangle$, and $\langle\mathcal{U}\rangle$ has a validity time $\mathrm{VT}_{\mathcal{U}_i}$. The guard system $\mathcal{G}_j$ is a key verifier (or access control) system and its status is $\langle\mathcal{G}\rangle$. The guard system verifies the validity and validity times of keys, and store the related data of key owners in its database.

Our protocol goals are (1) only $\mathcal{MK}$s are permitted to issue (generate) other new $\mathcal{MK}$s or $\mathcal{VK}$s; (2) the key issuers (generators) do not know the values of their issued (generated) keys; (3) only legitimate $\mathcal{MK}$ and $\mathcal{VK}$ can pass the verification of the guard system; (4) $\mathcal{G}$ can precisely identify who is accessing the system; (5) short-term visitor key authentication (SVA) (i.e., a *storage-free* choice) eliminates the need for system storage space; (6) keys can be revoked and keys generated from one specified user can be revoked; (7) PKI or public-key cryptosystem is not required; (8) authentication is efficient for users; (9) the system provides replay attack resistance; and (10) the system provides man-in-the-middle attack resistance. Of course, the first item can be designed arbitrarily.

Table 1: Notations.

| Notation | Meaning |
|---|---|
| $\langle\mathcal{A}\rangle$ | Administrator status |
| $\langle\mathcal{S}\rangle$ | Super user status |
| $\langle\mathcal{U}\rangle$ | Normal user status |
| $\langle\mathcal{G}\rangle$ | Guard system status |
| $\mathcal{MK}$ | Master grade authentication key |
| $\mathcal{VK}$ | Visitor grade authentication key |
| $\mathrm{ID}_U$ | Entity identity $U$ |
| $T_U$ | Time stamp used for an entity $U$ |
| $\mathrm{VT}_U$ | Validity time of an entity $U$ |
| $E_k(x)$ | Encryption of message $x$ using symmetric key $k$ |
| $D_k(x)$ | Decryption of message $x$ using symmetric key $k$ |
| $H_k(x)$ | Hash value of message $x$ using hash function $H$ and key $k$ |
| $\lvert x\rvert$ | Bit length of message $x$ |

Table 2: The models and protocol goals.

| Status | Key grade | Issued-key grade |
|---|---|---|
| $\langle\mathcal{A}\rangle$ | $\mathcal{MK}$ | $\mathcal{MK}, \mathcal{VK}$ |
| $\langle\mathcal{S}\rangle$ | $\mathcal{MK}$ | $\mathcal{MK}, \mathcal{VK}$ |
| $\langle\mathcal{U}\rangle$ | $\mathcal{VK}$ | none |

*2.2. Security Requirements.* The security requirements of our proposed protocol are as follows.

(1) *Authentication*. Guard can authenticate users.

(2) *Confidentiality of key values*. Users can only obtain their own keys and have no access to any information about others.

(3) *Replay attack resistance*. An adversary (or the originator) who intercepts or eavesdrops the data and retransmits it is prevented from successfully obtaining private information or posing as a party running an authentication protocol.

(4) *MITM attack resistance*. An adversary who intercepts the data between the Guards and users, makes independent connections with them, and relays messages between them is prevented from successfully making the guards and users believe that they are talking directly to each other.

## 3. The Proposed Scheme

In our protocol, there are two basic roles, *key owner* and *key verifier*, where key owners have their key storage space and key verifiers have their own database. Table 3 shows the possible contents in a key owner's storage space. From the table, we can see that the key owner has three different statuses $\mathcal{A}$, $\mathcal{S}$, and $\mathcal{U}$ for different guard systems $\mathcal{G}_1$, $\mathcal{G}_2$, $\mathcal{G}_3$, and $\mathcal{G}_4$. He or she uses different keys $\mathcal{K}_{\mathcal{U}_1}$, $\mathcal{K}_{\mathcal{U}_2}$, $\mathcal{K}_{\mathcal{U}_3}$, and $\mathcal{K}_{\mathcal{U}_4}$, respectively, for guard systems $\mathcal{G}_1$, $\mathcal{G}_2$, $\mathcal{G}_3$, and $\mathcal{G}_4$. In addition, the various guard systems have different

Table 3: Possible storage of a key owner $\mathcal{U}_i$.

| Status | Guard system | Key | Valid time | Issuer | Key status |
|---|---|---|---|---|---|
| $\langle \mathcal{A} \rangle$ | $\mathrm{ID}_{\mathcal{G}_1}$ | $\mathcal{K}_{\mathcal{U}_1}$ | $\mathrm{VT}_{\mathcal{U}_1}$ | N/A | *ok* |
| $\langle \mathcal{S} \rangle$ | $\mathrm{ID}_{\mathcal{G}_2}$ | $\mathcal{K}_{\mathcal{U}_2}$ | $\mathrm{VT}_{\mathcal{U}_2}$ | $\mathrm{ID}_{\mathcal{U}_1}$ | *ok* |
| $\langle \mathcal{U} \rangle$ | $\mathrm{ID}_{\mathcal{G}_3}$ | $\mathcal{K}_{\mathcal{U}_3}$ | $\mathrm{VT}_{\mathcal{U}_3}$ | $\mathrm{ID}_{\mathcal{U}_1}$ | $\mathrm{EMK}_{\mathcal{U}_3}$ |
| $\langle \mathcal{U} \rangle$ | $\mathrm{ID}_{\mathcal{G}_4}$ | $\mathcal{K}_{\mathcal{U}_4}$ | $\mathrm{VT}_{\mathcal{U}_4}$ | $\mathrm{ID}_{\mathcal{U}_2}$ | *ok* |

Table 4: Possible database contents of a Guard $\mathcal{G}_j$.

| Status | User ID | Key | Valid time | Issuer |
|---|---|---|---|---|
| $\langle \mathcal{A} \rangle$ | $\mathrm{ID}_{\mathcal{U}_1}$ | $\mathcal{K}_{\mathcal{AG}}$ | $\mathrm{VT}_{\mathcal{U}_1}$ | 00 |
| $\langle \mathcal{S} \rangle$ | $\mathrm{ID}_{\mathcal{U}_2}$ | $\mathcal{K}_{\mathcal{U}_2}$ | $\mathrm{VT}_{\mathcal{U}_2}$ | $\mathrm{ID}_{\mathcal{U}_1}$ |
| $\langle \mathcal{S} \rangle$ | $\mathrm{ID}_{\mathcal{U}_3}$ | $\mathcal{K}_{\mathcal{U}_3}$ | $\mathrm{VT}_{\mathcal{U}_3}$ | $\mathrm{ID}_{\mathcal{U}_2}$ |
| $\langle \mathcal{U} \rangle$ | $\mathrm{ID}_{\mathcal{U}_4}$ | $\mathcal{K}_{\mathcal{U}_4}$ | $\mathrm{VT}_{\mathcal{U}_4}$ | $\mathrm{ID}_{\mathcal{U}_1}$ |
| $\langle \mathcal{U} \rangle$ | $\mathrm{ID}_{\mathcal{U}_5}$ | $\mathcal{K}_{\mathcal{U}_5}$ | $\mathrm{VT}_{\mathcal{U}_5}$ | $\mathrm{ID}_{\mathcal{U}_2}$ |
| $\langle \mathcal{U} \rangle$ | $\mathrm{ID}_{\mathcal{U}_6}$ | $\mathcal{K}_{\mathcal{U}_6}$ | $\mathrm{VT}_{\mathcal{U}_6}$ | $\mathrm{ID}_{\mathcal{U}_3}$ |

validity times, issuers, and key status, where *key status* will be explained later.

Table 4 shows the possible database contents of a guard system $\mathcal{G}_j$. From the table, we can see user status, user identity, user access key, the key's validity time, and the issuers of user access key. We will show how to establish the data from our protocol later. Our protocols can be separated into seven phases, (1) *initialization*, (2) *authentication*, (3) *key generation*, (4) *registration*, (5) *short-term visitor key authentication (SVA)*, (6) *key revocation*, and (7) *scheme designs*. Aside from registration in the initialization phase, which is assumed to take place in a secure environment, all entities are communicated through insecure channels. (The assumption is reasonable in that users apply and register digital certificates personally in banks or local registration offices).

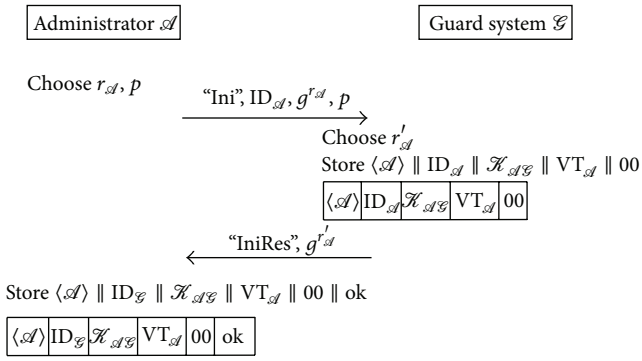


Figure 1: Initialization (under a secure environment).

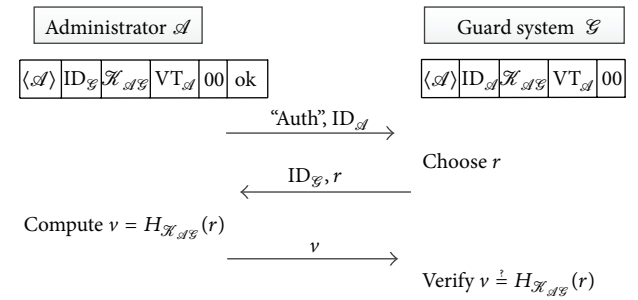


Figure 2: Authentication.

*3.1. Initialization.* Basically, each guard system $\mathcal{G}$ matches at least one administrator $\mathcal{A}$. Therefore, for each guard system, an administrator must execute an initialization phase to *switch on* a new guard system.

We assume the administrator $\mathcal{A}$ and guard system $\mathcal{G}$ share a secret key $\mathcal{K}_{\mathcal{AG}}$ after the initialization phase. A validity time $\mathrm{VT}_{\mathcal{A}}$ is also recorded in the data base of $\mathcal{G}$ along with the storage space (memory or hard disc) of $\mathcal{A}$. Normally, an administrator's $\mathrm{VT}_{\mathcal{A}}$ is recorded as "*never expired*."

There are many methods for exchanging a secret key $\mathcal{K}_{\mathcal{AG}}$ between $\mathcal{A}$ and $\mathcal{G}$. For example, the $\mathcal{A}$ can choose a secret key $\mathcal{K}_{\mathcal{AG}}$, encrypt it using $\mathcal{G}$'s public key, and send the encrypted secret key to $\mathcal{G}$. Alternatively, $\mathcal{A}$ and $\mathcal{G}$ may use key exchange protocols to share an exchanged key. Here, we use the Diffie-Hellman key exchange protocol as a simple example.

In our scheme, we assume $\mathcal{A}$ and $\mathcal{G}$ exchange secret key $\mathcal{K}_{\mathcal{AG}}$ in a secure offline environment, such as through encrypted Wi-Fi Direct [50], Bluetooth [51, 52] or Infrared Data Association (IrDA) [53]. As shown in Figure 1, $\mathcal{A}$ chooses a random number $r_{\mathcal{A}}$, prime $p$ and then sends $g^{r_{\mathcal{A}}}$, $p$, its identity $\mathrm{ID}_{\mathcal{A}}$, and initialization request "`Ini`" to $\mathcal{G}$. Next, $\mathcal{G}$ chooses a random number $r'_{\mathcal{A}}$, stores the information $\langle \mathcal{A} \rangle || \mathrm{ID}_{\mathcal{A}} || \mathcal{K}_{\mathcal{AG}} || \mathrm{VT}_{\mathcal{A}} || 00$ in its database, and sends a response "`IniRes`" to $\mathcal{A}$, where "||" means concatenation and $\mathcal{K}_{\mathcal{AG}} = g^{r_A r'_{\mathcal{A}}} \bmod p$. A stores the information $\langle \mathcal{A} \rangle || \mathrm{ID}_{\mathcal{G}} || \mathcal{K}_{\mathcal{AG}} || \mathrm{VT}_{\mathcal{A}} || 00 || ok$ in its database and finishes this phase.

*3.2. Authentication.* There are two grades of authentication key, *master key* ($\mathcal{MK}$) and *visitor key* ($\mathcal{VK}$). The key owners use their keys to pass the guard system's verification via the *authentication* phase. Since a key owner and a guard system should have shared a secret key, they can use message authentication methods to authenticate each other. There are many message authentication methods, but three-way challenge-response authentication is used here.

Take administrator $\mathcal{A}$, for example (as shown in Figure 2). A key owner first sends its identity ($\mathrm{ID}_{\mathcal{A}}$) and a request "`Auth`" to a guard system $\mathcal{G}$. Next, $\mathcal{G}$ checks whether $\mathrm{VT}_{\mathcal{A}}$ is valid. If it is not, $\mathcal{G}$ sends the message "`Key expired`" and terminates this phase. Else, $\mathcal{G}$ chooses a random number $r$ and sends $r$ and its identity $\mathrm{ID}_{\mathcal{G}}$ to $\mathcal{A}$.

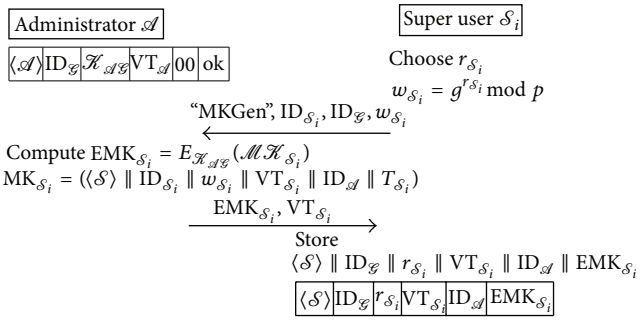


Figure 3: Key generation (master key).

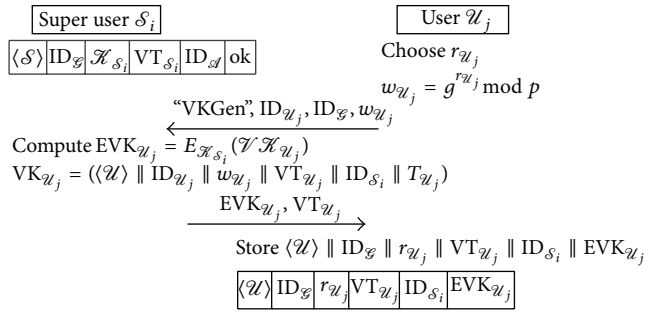


Figure 4: Key generation (visitor key).

After receiving $ID_{\mathcal{G}}$ and $r$, $\mathcal{A}$ computes $v = H_{\mathcal{K}_{\mathcal{AG}}}(r)$ and sends $v$ to $\mathcal{G}$. To verify whether $v$ is a valid authentication value, $\mathcal{G}$ checks whether the equation $v = H_{\mathcal{K}_{\mathcal{AG}}}(r)$ holds. If it does not hold, $\mathcal{G}$ sends the message "`The authentication is invalid`" and terminates the phase. Otherwise, $\mathcal{G}$ takes $\mathcal{A}$ as a legitimate user and lets $\mathcal{A}$ pass the verification.

*3.3. Key Generation.* Except for the current administrator in $\langle\mathcal{A}\rangle$, who shares a secret key with the guard system, a new super user in status $\langle\mathcal{S}\rangle$ may request a new key in grade $\mathcal{MK}$. (Other new administrators in status $\langle\mathcal{A}\rangle$ may need to do the same thing.) Also, a new normal user in status $\langle\mathcal{U}\rangle$ may request a new key in grade $\mathcal{VK}$. He can ask a $\mathcal{MK}$ owner to generate a new key for him. This key generation phase lets an authentication key in grade $\mathcal{MK}$ generate a new key in grade $\mathcal{MK}$ or $\mathcal{VK}$. For security considerations, this phase should be carried out face-to-face via mobile devices. That is, we assume the key requester and key issuer run the key generation phase in a secure environment, such as via encrypted Wi-Fi Direct, Bluetooth, or IrDA. Otherwise, a basic authentication for communicated messages is advised.

*3.3.1. $\mathcal{MK}$ Generation.* In the phase, assume a new super user $\mathcal{S}_i$ requests the administrator $\mathcal{A}$ for a new key which is in grade $\mathcal{MK}$ and is used in guard system $\mathcal{G}$. As shown in Figure 3, $\mathcal{S}_i$ first chooses a random number $r_{\mathcal{S}_i}$ and computes $w_{\mathcal{S}_i} = g^{r_{\mathcal{S}_i}} \bmod p$. Then $\mathcal{S}_i$ sends $w_{\mathcal{S}_i}$, its identity $ID_{\mathcal{S}_i}$, the guard system's identity $ID_{\mathcal{G}}$, and $\mathcal{MK}$ generation request "`MKGen`" to $\mathcal{A}$. Next, $\mathcal{A}$ computes $EMK_{\mathcal{S}_i} = E_{\mathcal{K}_{\mathcal{AG}}}(\mathcal{MK}_{\mathcal{S}_i})$ and then sends $EMK_{\mathcal{S}_i}$ and its validity time value $VT_{\mathcal{S}_i}$ to $\mathcal{S}_i$, where $MK_{\mathcal{S}_i} = \langle\mathcal{S}\rangle||ID_{\mathcal{S}_i}||w_{\mathcal{S}_i}||VT_{\mathcal{S}_i}||ID_{\mathcal{A}}||T_{\mathcal{S}_i}$ and $T_{\mathcal{S}_i}$ is a time stamp. After receiving $EMK_{\mathcal{S}_i}$ and $VT_{\mathcal{S}_i}$, $\mathcal{S}_i$ stores the information $\langle\mathcal{S}\rangle||ID_{\mathcal{G}}||r_{\mathcal{S}_i}||VT_{\mathcal{S}_i}||ID_{\mathcal{A}}||EMK_{\mathcal{S}_i}$ in its storage space and finishes this phase.

*3.3.2. $\mathcal{VK}$ Generation.* $\mathcal{VK}$ generation lets a valid key in grade $\mathcal{MK}$ generate a new key in the grade $\mathcal{VK}$. The steps are similar to those in $\mathcal{MK}$ generation. Assume a new normal user $\mathcal{U}_j$ requests super user $\mathcal{S}_i$ to provide a new key which is in grade $\mathcal{VK}$ and is used in guard system $\mathcal{G}$. As shown in Figure 4, $\mathcal{U}_j$ chooses a random number $r_{\mathcal{U}_j}$, computes $w_{\mathcal{U}_j} = g^{r_{\mathcal{U}_j}} \bmod p$, and sends $w_{\mathcal{U}_j}$, $ID_{\mathcal{U}_j}$, $ID_{\mathcal{G}}$, and "`VKGen`" to $\mathcal{S}_i$. Then, $\mathcal{S}_i$ computes $EMK_{\mathcal{U}_j} = E_{\mathcal{K}_{\mathcal{S}_i}}(\mathcal{VK}_{\mathcal{U}_j})$ and sends $EMK_{\mathcal{U}_j}$ and $VT_{\mathcal{U}_j}$ to $\mathcal{U}_j$, where $VK_{\mathcal{U}_j} = \langle\mathcal{U}\rangle||ID_{\mathcal{U}_j}||w_{\mathcal{U}_j}||VT_{\mathcal{U}_j}||ID_{\mathcal{S}_i}||T_{\mathcal{U}_j}$. After receiving $EVK_{\mathcal{U}_j}$ and $VT_{\mathcal{U}_j}$, $\mathcal{U}_j$ stores $\langle\mathcal{U}\rangle||ID_{\mathcal{G}}||r_{\mathcal{U}_j}||VT_{\mathcal{U}_j}||ID_{\mathcal{S}_i}||EMK_{\mathcal{U}_j}$ in its storage space and finishes this phase.

*3.4. Registration.* After finishing the key generation phase, each new $\mathcal{MK}$ or $\mathcal{VK}$ requester gets a *pseudo key*, which is invalid and unusable. He or she has to run the *registration* phase to produce a real and valid key. The guard system can then store the real key in its database.

*3.4.1. $\mathcal{MK}$ Registration.* In this phase, assume a super user $\mathcal{S}_i$ wants to register a generated pseudo key to a guard system $\mathcal{G}$. As shown in Figure 5, $\mathcal{S}_i$ sends $EMK_{\mathcal{S}_i}$, $ID_{\mathcal{S}_i}$, $VT_{\mathcal{S}_i}$, the key issuer $ID_{\mathcal{A}}$, and the registration request "`MKRgst`" to $\mathcal{G}$. Next, $\mathcal{G}$ checks whether the status of $ID_{\mathcal{A}}$ is in status $\langle\mathcal{A}\rangle$ or $\langle\mathcal{S}\rangle$ and is thus permitted to issue keys. If it is not, $\mathcal{G}$ sends the message "`the key issuer is illegal`" and terminates this phase. Otherwise, $\mathcal{G}$ computes $MK'_{\mathcal{S}_i} = D_{\mathcal{K}_{\mathcal{AG}}}(EMK_{\mathcal{S}_i})$ and verifies time stamp $T_{\mathcal{S}_i}$. If $T_{\mathcal{S}_i}$ is inappropriate, $\mathcal{G}$ sends the message "`the time stamp is inappropriate`" and terminates this phase. Otherwise, $\mathcal{G}$ checks whether $ID_{\mathcal{S}_i}$, $VT_{\mathcal{S}_i}$, and $ID_{\mathcal{A}}$ match the values in $MK'_{\mathcal{S}_i}$. If they are not, $\mathcal{G}$ sends the message "`Data inconsistent`" and terminates this phase. If they are, $\mathcal{G}$ chooses $r'_{\mathcal{S}_i}$, computes $w'_{\mathcal{S}_i} = g^{r'_{\mathcal{S}_i}} \bmod p$ and $\mathcal{K}_{\mathcal{S}_i} = w_{\mathcal{S}_i}{}^{r'_{\mathcal{S}_i}} \bmod p$, and stores the information $\langle\mathcal{S}\rangle||ID_{\mathcal{S}_i}||\mathcal{K}_{\mathcal{S}_i}||VT_{\mathcal{S}_i}||ID_{\mathcal{A}}$ in its database. Then $\mathcal{G}$ sends the command "`ok`" and $w'_{\mathcal{S}_i}$ back to $\mathcal{S}_i$. After receiving them, $\mathcal{S}_i$ computes $\mathcal{K}_{\mathcal{S}_i} = w'_{\mathcal{S}_i}{}^{r_{\mathcal{S}_i}} \bmod p$, changes $r_{\mathcal{S}_i}$ to $\mathcal{K}_{\mathcal{S}_i}$, alters the value $EMK_{\mathcal{S}_i}$ to "`ok`", and finishes this phase.

Following this phase, $\mathcal{K}_{\mathcal{S}_i}$ is a valid $\mathcal{MK}$. $\mathcal{S}_i$ can use $\mathcal{K}_{\mathcal{S}_i}$ to authenticate itself to guard system $\mathcal{G}$ via *authentication* phase.

*3.4.2. $\mathcal{VK}$ Registration.* $\mathcal{VK}$ registration is similar to $\mathcal{MK}$ registration. Assume a normal user $\mathcal{U}_j$ registers a generated pseudo key to a guard system $\mathcal{G}$. As shown in Figure 6, $\mathcal{U}_j$ sends $EVK_{\mathcal{U}_j}$, $ID_{\mathcal{U}_j}$, $VT_{\mathcal{U}_j}$, the key issuer $ID_{\mathcal{S}_i}$, and "`VKRgst`" to $\mathcal{G}$. Next, $\mathcal{G}$ checks whether the status of $ID_{\mathcal{S}_i}$ is in status $\langle\mathcal{A}\rangle$ or $\langle\mathcal{S}\rangle$. $\mathcal{G}$ computes $VK'_{\mathcal{U}_j} = D_{\mathcal{K}_{\mathcal{S}_i}}(EVK_{\mathcal{U}_j})$

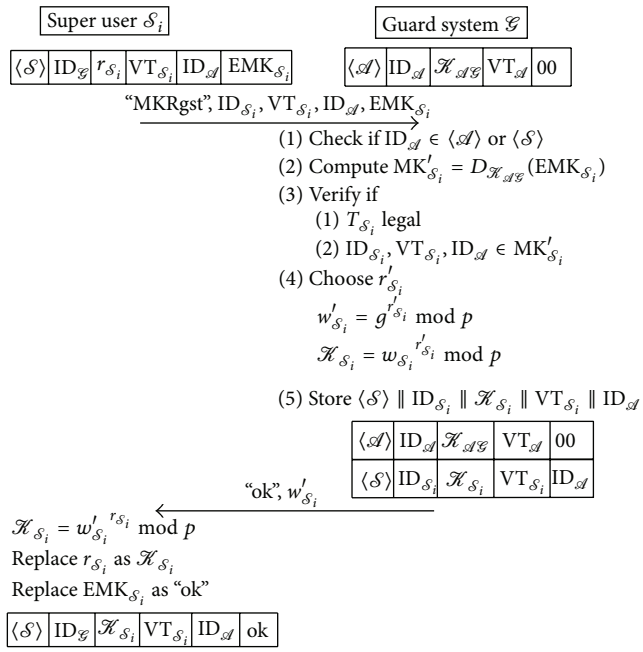


Figure 5: Registration (master key).

and verifies $T_{\mathcal{U}_j}$. $\mathcal{G}$ checks whether $ID_{\mathcal{U}_j}$, $VT_{\mathcal{U}_j}$, and $ID_{\mathcal{S}_i}$ match the values in $VK'_{\mathcal{U}_j}$. $\mathcal{G}$ chooses $r'_{\mathcal{U}_j}$, computes $w'_{\mathcal{U}_j} = g^{r'_{\mathcal{U}_j}} \bmod p$ and $\mathcal{K}_{\mathcal{U}_j} = w_{\mathcal{U}_j}{}^{r'_{\mathcal{U}_j}} \bmod p$, and stores $\langle\mathcal{U}\rangle||ID_{\mathcal{U}_j}||\mathcal{K}_{\mathcal{U}_j}||VT_{\mathcal{U}_j}||ID_{\mathcal{S}_i}$ in its database. Then $\mathcal{G}$ sends the command "ok" and $w'_{\mathcal{U}_j}$ to $\mathcal{U}_j$. Then, $\mathcal{U}_j$ computes $\mathcal{K}_{\mathcal{U}_j} = w'_{\mathcal{U}_j}{}^{r_{\mathcal{U}_j}} \bmod p$, replaces $r_{\mathcal{U}_j}$ to $\mathcal{K}_{\mathcal{U}_j}$, alters $EMK_{\mathcal{U}_j}$ to "ok," and finishes this phase.

*3.5. Short-Term Visitor Key Authentication* (SVA). We propose *short-term visitor key authentication* (SVA), which is used for rarely or briefly used $\mathcal{VK}$s. SVA does not have to execute the *registration* phase, followed by authentication. Rather, it can directly proceed the SVA phase to pass the guard system's verification using the generated *pseudo key*. Although its authentication time takes a little longer, it does not require storing any data about the $\mathcal{VK}$ owner in the guard system's database. SVA combines the *registration* and *authentication* phases. Assume a normal user $\mathcal{U}_j$ wants to proceed SVA to a guard system $\mathcal{G}$. As shown in Figure 7, $\mathcal{U}_j$ sends $EVK_{\mathcal{U}_j}$, $ID_{\mathcal{U}_j}$, $VT_{\mathcal{U}_j}$, the key issuer $ID_{\mathcal{S}_i}$, and "SVA" to $\mathcal{G}$. Next, $\mathcal{G}$ checks the legal status of the key issuer $ID_{\mathcal{S}_i}$. Then $\mathcal{G}$ checks validation of $VT_{\mathcal{U}_j}$, computes $VK'_{\mathcal{U}_j} = D_{\mathcal{K}_{\mathcal{S}_i}}(EVK_{\mathcal{U}_j})$, and verifies $T_{\mathcal{U}_j}$. $\mathcal{G}$ then checks whether $ID_{\mathcal{U}_j}$, $VT_{\mathcal{U}_j}$, and $ID_{\mathcal{S}_i}$ match the values in $VK'_{\mathcal{U}_j}$. Later, $\mathcal{G}$ chooses a random number $r'$, computes $w' = g^{r'} \bmod p$, and sends $w'$ and $ID_{\mathcal{G}}$ to $\mathcal{U}_j$. $\mathcal{U}_j$ then computes $v = w'^{r_{\mathcal{U}_j}} \bmod p$ and sends $v$ to $\mathcal{G}$. $\mathcal{G}$ then checks whether the equation $v = w_{\mathcal{U}_j}{}^{r'} \bmod p$ holds. If all the verifications above are legitimate, $\mathcal{U}_j$ passes to the SVA verification.

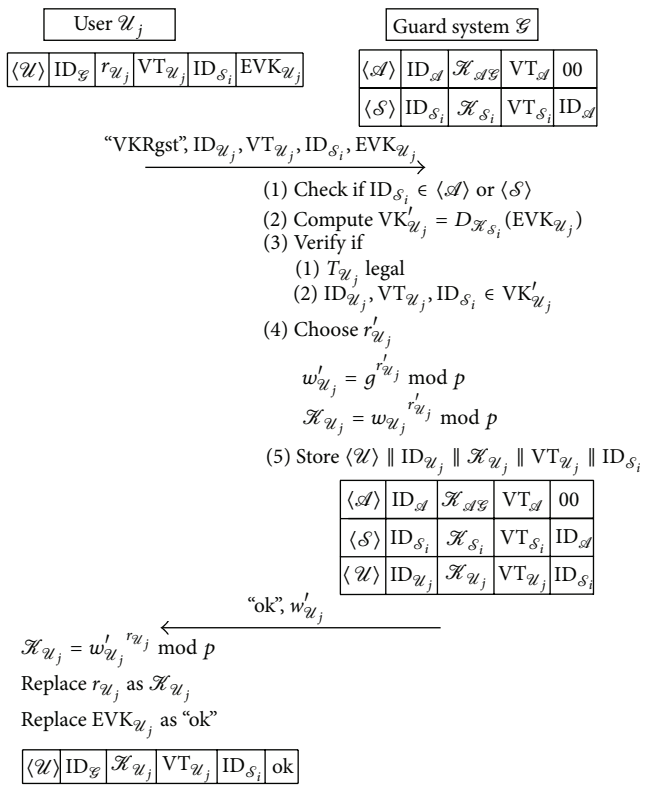


Figure 6: Registration (visitor key).

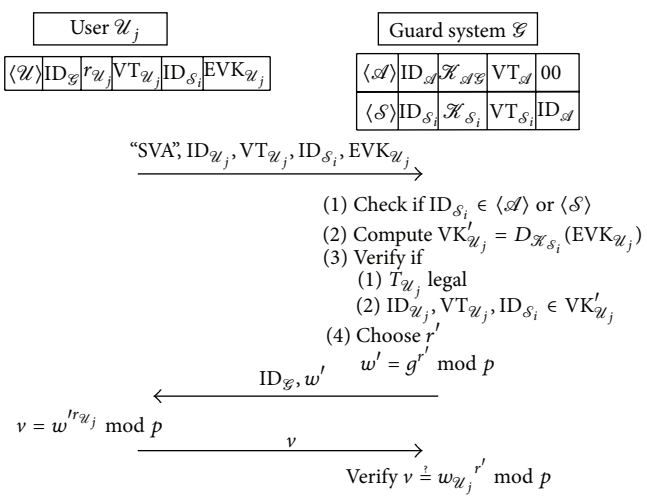


Figure 7: Short-term visitor key authentication (SVA).

*3.6. Key Revocation.* For a variety of reasons, individual keys, or all keys directly or indirectly generated by a particular, sometimes need to be revoked. For example, assume a super user $A$ generates a key for a super user $B$, and $B$ generates a key for a normal user $C$. If $A$'s key is compromised, we should revoke the keys of both $B$ and $C$ for possible dissimulations. To achieve this goal, all the issued keys can be figured in a tree (or directed graph) structure. Put the administrator on the root vertex, the key generator on the parent vertex, and the key requester on the child vertex. Create a (directed) edge

from a parent $u$ to a child $v$ when $u$ generates a key for $v$. When we revoke the key of one user in a parent vertex, we should also revoke all the keys of users in its child vertexes.

*3.7. Scheme Designs.* Aside from the designed model shown in Table 2, the proposed scheme can also be used to design other models. For example, as shown in Table 5, we can design four different statuses (administrator, super user, power user, and normal user) with different key grades and issued-key grades. In registration phase, guard systems can judge the ability of key generators according to their statuses or levels. Therefore, the database of guard system may look like the list in Table 6.

Table 5: Protocol goals of other scenarios.

| Status | Key grade | Issued-key grade | Level |
|---|---|---|---|
| Administrator | $\mathcal{MK}$ | $\mathcal{SK}, \mathcal{PK}, \mathcal{UK}$ | 3 |
| Super user | $\mathcal{SK}$ | $\mathcal{PK}, \mathcal{UK}$ | 2 |
| Power user | $\mathcal{PK}$ | $\mathcal{UK}$ | 1 |
| Normal user | $\mathcal{UK}$ | None | 0 |

Table 6: Guard system database for Table 5.

| Status | User ID | Key | Valid time | Issuer |
|---|---|---|---|---|
| $\langle \mathcal{A} \rangle$ | $\mathrm{ID}_{\mathcal{U}_1}$ | $\mathcal{K}_{\mathcal{AG}}$ | $\mathrm{VT}_{\mathcal{U}_1}$ | 00 |
| $\langle \mathcal{S} \rangle$ | $\mathrm{ID}_{\mathcal{U}_2}$ | $\mathcal{K}_{\mathcal{U}_2}$ | $\mathrm{VT}_{\mathcal{U}_2}$ | $\mathrm{ID}_{\mathcal{U}_1}$ |
| $\langle \mathcal{P} \rangle$ | $\mathrm{ID}_{\mathcal{U}_3}$ | $\mathcal{K}_{\mathcal{U}_3}$ | $\mathrm{VT}_{\mathcal{U}_3}$ | $\mathrm{ID}_{\mathcal{U}_2}$ |
| $\langle \mathcal{P} \rangle$ | $\mathrm{ID}_{\mathcal{U}_4}$ | $\mathcal{K}_{\mathcal{U}_4}$ | $\mathrm{VT}_{\mathcal{U}_4}$ | $\mathrm{ID}_{\mathcal{U}_1}$ |
| $\langle \mathcal{U} \rangle$ | $\mathrm{ID}_{\mathcal{U}_5}$ | $\mathcal{K}_{\mathcal{U}_5}$ | $\mathrm{VT}_{\mathcal{U}_5}$ | $\mathrm{ID}_{\mathcal{U}_2}$ |
| $\langle \mathcal{U} \rangle$ | $\mathrm{ID}_{\mathcal{U}_6}$ | $\mathcal{K}_{\mathcal{U}_6}$ | $\mathrm{VT}_{\mathcal{U}_6}$ | $\mathrm{ID}_{\mathcal{U}_4}$ |
| $\langle \mathcal{U} \rangle$ | $\mathrm{ID}_{\mathcal{U}_7}$ | $\mathcal{K}_{\mathcal{U}_7}$ | $\mathrm{VT}_{\mathcal{U}_7}$ | $\mathrm{ID}_{\mathcal{U}_1}$ |

## 4. Analysis of Proposed Scheme

*4.1. Security Analysis.* We analyze the security of our protocols according to the requirements defined in Section 2.2 as follows.

*Authentication.* In the `Authentication` phase, take $\mathcal{A}$, for example (as shown in Figure 2). $\mathcal{G}$ can authenticate $\mathcal{A}$ from his response $v$ since $r$ is a random number chosen from $\mathcal{G}$ and is hashed via the secret key $\mathcal{K}_{\mathcal{AG}}$ between $\mathcal{A}$ and $\mathcal{G}$, where $v = H_{\mathcal{K}_{\mathcal{AG}}}(r)$. In the `Registration` phase, take $\mathcal{S}_i$, for example (as shown in Figure 5). $\mathcal{G}$ can authenticate $\mathcal{S}_i$ from $\mathrm{EMK}_{\mathcal{S}_i}$ since $\mathcal{G}$ computes $\mathrm{MK}'_{\mathcal{S}_i} = D_{\mathcal{K}_{\mathcal{AG}}}(\mathrm{EMK}_{\mathcal{S}_i})$, verifies $T_{\mathcal{S}_i}$, and checks whether $\mathrm{ID}_{\mathcal{S}_i}$, $\mathrm{VT}_{\mathcal{S}_i}$, and $\mathrm{ID}_{\mathcal{A}}$ match the values in $\mathrm{MK}'_{\mathcal{S}_i}$.

*Confidentiality of Key Values.* Take $\mathcal{S}_i$, for example (as shown in Figures 3 and 5). The secret key $\mathcal{K}_{\mathcal{S}_i}$ between $\mathcal{S}_i$ and $\mathcal{G}$ is confidential because $r_{\mathcal{S}_i}$ and $r'_{\mathcal{S}_i}$ are chosen by $\mathcal{S}_i$ and $\mathcal{G}$, and an adversary (or $\mathcal{A}$) cannot evaluate $\mathcal{K}_{\mathcal{S}_i}$ from $w_{\mathcal{S}_i}$ and $w'_{\mathcal{S}_i}$, where $w_{\mathcal{S}_i} = g^{r_{\mathcal{S}_i}} \bmod p$, $w'_{\mathcal{S}_i} = g^{r'_{\mathcal{S}_i}} \bmod p$, and $\mathcal{K}_{\mathcal{S}_i} = {w'_{\mathcal{S}_i}}^{r_{\mathcal{S}_i}} \bmod p = {w_{\mathcal{S}_i}}^{r'_{\mathcal{S}_i}} \bmod p$. Therefore, the key values between the users and $\mathcal{G}$ are kept private.

*ReplayAttack Resistance.* In the `Authentication` phase, take $\mathcal{A}$, for example (as shown in Figure 2). $\mathcal{G}$ transmits $r$ to $\mathcal{A}$ and $\mathcal{A}$ transmits $v = H_{\mathcal{K}_{\mathcal{AG}}}(r)$ to $\mathcal{G}$. Since the value $r$ changes, the value $v$ is different in different authentication phases. Therefore, the `Authentication` phase can resist replay attacks. In the `Key generation` phase (as shown in Figures 3 and 4), the key requester and key issuer should interact face-to-face via mobile devices in a secure environment, such as encrypted Wi-Fi Direct, Bluetooth, or IrDA. Therefore, the `Key generation` phase can resist replay attacks. In the `Registration` phase, take $\mathcal{S}_i$, for example (as shown in Figure 5). $\mathrm{EMK}_{\mathcal{S}_i}$ cannot be modified or reused since $\mathrm{EMK}_{\mathcal{S}_i}$ is authenticated and $\mathrm{ID}_{\mathcal{S}_i}$ is stored in $\mathcal{G}$'s database. Therefore, `Registration` phase can resist replay attacks.

*MITM Attack Resistance.* In the `Registration` phase, take $\mathcal{S}_i$, for example (as shown in Figure 5). An adversary cannot make $\mathcal{G}$ believe that he is talking directly to $\mathcal{S}_i$ since $\mathrm{EMK}_{\mathcal{S}_i}$ has to be authenticated and therefore $w_{\mathcal{S}_i}$ in $\mathrm{MK}_{\mathcal{S}_i}$ is prevented from changing. Also, an adversary can not make $\mathcal{S}_i$ believe that he is talking directly to $\mathcal{G}$ since $w_{\mathcal{S}_i}$ cannot be known from $\mathrm{EMK}_{\mathcal{S}_i}$ without secret key $\mathcal{K}_{\mathcal{AG}}$ or from the `Key generation` phase (as shown in Figure 3) which should be run in a mobile, face-to-face, and secure environment, such as through encrypted Wi-Fi Direct or Bluetooth. Even if $w_{\mathcal{S}_i}$ is known via cryptanalysis, the value $w_{\mathcal{S}_i}$ is unchanged because it is difficult for an adversary to intercept the data between $\mathcal{A}$ and $\mathcal{S}_i$ and make independent connections with $\mathcal{A}$ and $\mathcal{S}_i$ in mobile and face-to-face environments. Therefore, this system can resist man-in-the-middle attacks.

*4.2. Protocol Efficiency.* In this subsection, we analyze the performance of our proposed methods. First, we show the symbols used in analysis in Table 7. We then compare the cost, including computational cost and communication cost, of each phase: `key generation`, `registration`, `authenticate`, and `SVA` in Tables 8 and 11, respectively. Finally, the storage space cost is discussed in Table 12.

Table 8 shows that, in the key generation phase, the computational costs of key requester and key issuer are $T_E$ and $T_X$, and the communication costs are $3 \cdot l_N + l_W$ and $l_N + l_E$. As per the assumption of bit length in Table 7, the communication costs are 140 bytes and 152 bytes. Similarly, Table 9 shows the computational costs and communication costs of user and guard in the registration phase.

In both Tables 10 and 11, the computational costs of user are the same ($T_M$). The other costs of `SVA` are higher than those of general authentication, because users, if `SVA` is run, do not have to run the registration phase. Actually, aside from the computational cost of user, the other costs of `SVA` are approximate to the sum of the registration cost and the authentication cost. Note that, for both general authentication or `SVA`, the computational cost of User is just $T_M$, which means a user only has to run an MAC algorithm to complete authentication.

Table 7: Symbols used in performance analysis.

| Symbol | Meaning |
|---|---|
| $l_N$ | Length of $\langle \mathcal{U} \rangle$, $\lvert \mathrm{ID}_{\mathcal{U}_j} \rvert$, $\lvert T \rvert$, $\lvert \mathrm{VT} \rvert$, and each command (e.g., 32 bits) |
| $l_H$ | Length of the output of hash function (e.g., 128 bits) |
| $l_K$ | Length of a symmetric key $\mathcal{K}_{\mathcal{U}_j}$ (e.g., 128 bits) |
| $l_W$ | Length of $\lvert w_{\mathcal{S}_i} \rvert$ and $\lvert w_{\mathcal{U}_j} \rvert$ (e.g., 1024 bits) |
| $l_E$ | Length of $\lvert \mathrm{MK}_{\mathcal{U}_j} \rvert$ and $\lvert \mathrm{EMK}_{\mathcal{U}_j} \rvert$ (e.g., $6 \cdot l_N + l_W = 1184$ bits) |
| $T_M$ | The cost of running MAC algorithm |
| $T_C$ | The cost of running the comparison or element-existence checking |
| $T_E$ | The cost of running the symmetric-key encryption function |
| $T_D$ | The cost of running the symmetric-key decryption function |
| $T_X$ | The cost of running the module exponential computation |

Note: normally, $T_M < T_C < T_E \approx T_D < T_X$, and $l_N < l_H \approx l_K < l_W < l_E$.

Table 8: Cost of key generation.

| Item | Key requester | Key issuer |
|---|---|---|
| Computational cost | $T_E$ | $T_X$ |
| Communication cost | $3 \cdot l_N + l_W$ (140 bytes) | $l_N + l_E$ (152 bytes) |
| Transaction number | 1 | 1 |

Table 9: Cost of Registration.

| Item | User | Guard |
|---|---|---|
| Computational cost | $T_X$ | $2 \cdot T_X + T_D + 5 \cdot T_C$ |
| Communication cost | $4 \cdot l_N + l_E$ (164 bytes) | $l_N + l_W$ (132 bytes) |
| Transaction number | 1 | 1 |

Table 10: Cost of authenticate.

| Item | User | Guard |
|---|---|---|
| Computational cost | $T_M$ | $T_M + T_C$ |
| Communication cost | $2 \cdot l_N + l_H$ (24 bytes) | $l_N + l_H$ (20 bytes) |
| Transaction number | 1 | 1 |

Table 11: Cost of SVA.

| Item | User | Guard |
|---|---|---|
| Computational cost | $T_M$ | $2 \cdot T_X + T_D + 5 \cdot T_C + T_C$ |
| Communication cost | $4 \cdot l_N + l_E + l_H$ (180 bytes) | $l_N + l_W$ (132 bytes) |
| Transaction number | 2 | 1 |

Table 12 compares storage space costs. Before registration, users keep the value $\mathrm{EMK}_{\mathcal{U}_j}$, of which the length is $l_E$ and is about 1184 bits. After registration, users replace $\mathrm{EMK}_{\mathcal{U}_j}$ to "ok," for which the length is $l_N$ and is about 32 bits. Therefore, the user's storage costs before and after registration are quite different. In SVA scheme, since users do not have to run the registration phase, the user's space costs in SVA and in general scheme (before registration) are the same.

Table 12: Storage space cost.

| Item | User | Guard |
|---|---|---|
| General scheme (before registration) | $3 \cdot l_N + l_K + l_E$ (176 bytes) | None (0 bytes) |
| General scheme (after registration) | $4 \cdot l_N + l_K$ (32 bytes) | $4 \cdot l_N + l_K$ (32 bytes) |
| SVA scheme (no registration required) | $3 \cdot l_N + l_K + l_E$ (176 bytes) | None (0 bytes) |

*4.3. Protocol Correctness.* We analyze the correctness of our protocols according to the goals stated in Section 2.1.

(1) *Only $\mathcal{MK}$s are permitted to issue (generate) other new $\mathcal{MK}$s or $\mathcal{VK}$s.* In the Registration phase, $\mathcal{G}$ checks whether the key issuer is in status $\langle \mathcal{A} \rangle$ or $\langle \mathcal{S} \rangle$. Also, $\mathcal{G}$ uses the secret key, shared with the key issuer, to decrypt the message EMK and checks the correctness of key issuer's ID, key requester's ID, and validity time. Therefore, only $\mathcal{MK}$s are permitted to issue (generate) other new keys.

(2) *The key issuers (generators) do not know the values of their issued (generated) keys.* In the Key generation phase, user $\mathcal{S}_i$ (or $\mathcal{U}_j$) chooses a random number $r_{\mathcal{S}_i}$ (or $r_{\mathcal{U}_j}$), computes $w_{\mathcal{S}_i} = g^{r_{\mathcal{S}_i}} \bmod p$ (or $w_{\mathcal{U}_j} = g^{r_{\mathcal{U}_j}} \bmod p$), and sends it to key issuer. The key issuer then encrypts $w_{\mathcal{S}_i}$ (or $w_{\mathcal{U}_j}$) into $\mathrm{EMK}_{\mathcal{S}_i}$ (or $\mathrm{EMK}_{\mathcal{U}_j}$) and sends it to the key requester. Then, in the Registration phase, the key requester sends it to $\mathcal{G}$. $\mathcal{G}$ then uses $w_{\mathcal{S}_i}$ (or $w_{\mathcal{U}_j}$) to make a Diffie-Hellman key exchange indirectly and authentically and exchange a secret key $\mathcal{K}_{\mathcal{S}_i}$ (or $\mathcal{K}_{\mathcal{U}_j}$). Therefore, key issuers do not know the values of their issued keys.

(3) *Only legitimate $\mathcal{MK}$ and $\mathcal{VK}$ can pass the guard system verification.* After running the Registration phase, the legitimate $\mathcal{MK}$ or $\mathcal{VK}$ key owner can pass the verification of $\mathcal{G}$ by running the Authentication phase. ($\mathcal{VK}$ key owner can choose to run SVA directly without running the Registration phase.) Without assistance from the key issuers, who use secret keys to generate EVK, illegitimate key users fail the verification of $\mathcal{G}$ in running the Registration, Authentication, or SVA phases.

(4) *$\mathcal{G}$ can precisely identify who is accessing the system.* $\mathcal{G}$ can use the IDs and secret keys stored in its database to identify who is accessing the system. In SVA mode, $\mathcal{G}$ can use the IDs and secret keys of key issuers to determine user identities.

(5) *Short-term visitor key authentication (SVA) (i.e., a storage-free choice) frees up system storage space.* As

shown in Figure 7, $\mathcal{VK}$ key owners, who run SVA, do not run the Registration phase. Therefore, $\mathcal{G}$ does not need to store data related to $\mathcal{VK}$ key owners in its database.

(6) *Keys can be revoked and keys generated by one specified user can be revoked.* From Section 3.6, we present a tree-structure method of key revocation allowing for the revocation of individual keys and keys generated by a specified user.

(7) *PKI or public-key cryptosystem is not required.* As we mentioned in Section 1, traditional public-key cryptograph, which requires the computation of modular exponentiation, cannot be broadly used in mobile devices. The proposed system only uses a symmetric key cryptosystem, thus obviating the need for PKI or public-key cryptosystems.

(8) *Authentication is efficient for users.* As shown in Tables 10 and 11, in our proposed system the computational cost of a user is just $T_M$, which means a user only has to run an MAC algorithm to complete authentication. In addition, the total communication cost is about 44 bytes and 312 bytes, respectively, for general authentication and SVA. Therefore, the authentication is efficient for users.

(9) *The system provides replay attack resistance.* In the Authentication phase, $\mathcal{G}$ poses a challenge using a random $r$, which is different in each authentication instance. Key owners then use their secret keys to compute the MAC value of $r$. This change and response method provides replay attack resistance.

(10) *The system provides man-in-the-middle attack resistance.* We assume the Key generation and Registration phases are run at different times, and in different places and environments. Consequently, an attacker can not simultaneously obtain or forge $w_{\mathcal{S}_i}$ (or $w_{\mathcal{U}_j}$) and $w'_{\mathcal{S}_i}$ (or $w'_{\mathcal{U}_j}$). Therefore, the system provides man-in-the-middle attack resistance. Moreover, for security considerations, the Key generation phase should be run face-to-face or provide authenticated messages, thus protecting $w_{\mathcal{S}_i}$ (or $w_{\mathcal{U}_j}$) from attack. Even if an attacker can simultaneously forge them, the key requester will eventually fail to verify $\mathcal{G}$ and become aware of the attacks.

*4.4. Property Comparison.* The properties of the proposed protocol are compared with those of the Wang-Zhao protocol [39] and Niu-Wang protocol [46] in Table 13, where "Empowered issuing" and "Specific-key revocation", respectively, denote keys issued by empowered entities and the ability to revoke specific keys or keys issued by a specific user. ($^*1$) indicates that the agreed key is blind to TTP but is not really a "key issuing" action. Their protocols require TTP and the three entities are needed to simultaneously performs key agreement protocols. Our protocol, however, does not need TTP and *independent* key generation (in the key generation phase) can be independently performed between two mobile entities. That is, our protocol perform the key generation phase (between the key issuer and key requesters) and key registration phase (between the key requesters and guard system) independently, while other protocols perform key agreement among two entities and TTP simultaneously. Therefore, in our protocol, key requesters may perform the key generation phase with key issuer and then perform the key registration phase with a guard system later at another time. In contrast with the other protocols, the proposed protocol is more efficient and effective in the blind issuing of symmetric keys for mobile systems.

Table 13: Comparison of properties.

| | Wang-Zhao protocol | Niu-Wang protocol | Proposed protocol |
|---|---|---|---|
| Authentication | √ | √ | √ |
| Blindness of secret keys | √ | √ | √ |
| Blind key issuing | ($^*1$) | ($^*1$) | √ |
| Non-PKI requirement | √ | √ | √ |
| Replay attack resistance | √ | √ | √ |
| MITM attack resistance | √ | √ | √ |
| Empowered issuing | | | √ |
| Independent key generation | | | √ |
| Non-TTP requirement | | | √ |
| Specific-key revocation | | | √ |
| Storage-free choice | | | √ |

## 5. Conclusion

In the real world, a new key can be reproduced from any old key. We adapt and improve this concept in the mobile electrical world. In our scheme, the value of the old key is irrelevant to the new key, and the old and new key users are unaware of each other's key values, thus achieving secure identification. Key revocation in a tree structure also increases system security. The system is efficient and feasible and can be used in mobile access systems.

## Acknowledgments

This work was partially supported by the National Science Council under Grant NSC 101-2221-E-182-071. The authors also gratefully acknowledge helpful comments and suggestions from the reviewers, which have improved the presentation.

## References

[1] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[2] A. K. Awasthi and S. Lal, "A remote user authentication scheme using smart cards with forward secrecy," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 4, pp. 1246–1248, 2003.

[3] A. K. Awasthi and S. Lal, "An enhanced remote user authentication scheme using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 2, pp. 583–586, 2004.

[4] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, 1987.

[5] T. Kwon, Y. H. Park, and H. J. Lee, "Security analysis and improvement of the efficient password-based authentication protocol," *IEEE Communications Letters*, vol. 9, no. 1, pp. 93–95, 2005.

[6] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[7] N. M. Al-Saidi, M. R. M. Said, and W. A. M. Othman, "Password authentication based on fractal coding scheme," *Journal of Applied Mathematics*, vol. 2012, Article ID 340861, 16 pages, 2012.

[8] S. G. Yoo, K. Y. Park, and J. Kim, "A security-performance-balanced user authentication scheme for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 382810, 11 pages, 2012.

[9] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[10] C. Wang and J. Luo, "An efficient key-policy attribute-based encryption scheme with constant ciphertext length," *Mathematical Problems in Engineering*, vol. 2013, Article ID 810969, 7 pages, 2013.

[11] M. Rahnema, "Overview of the GSM system and protocol architecture," *IEEE Communications Magazine*, vol. 31, no. 4, pp. 92–100, 1993.

[12] K. Al-Tawil, A. Akrami, and H. Youssef, "A new authentication protocol for GSM networks," in *Proceedings of the 23rd Conference on Local Computer Networks (LCN '98)*, pp. 21–30, October 1998.

[13] X. Cao, W. Kou, Y. Yu, and R. Sun, "Identity-based authenticated key agreement protocols without bilinear pairings," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 91, no. 12, pp. 3833–3836, 2008.

[14] L. Law, A. Menezes, M. Qu, J. Solinas, and S. Vanstone, "An efficient protocol for authenticated key agreement," *Designs, Codes, and Cryptography*, vol. 28, no. 2, pp. 119–134, 2003.

[15] E. Chen, "Unsupervised user similarity mining in gsm sensor networks," *The Scientific World Journal*, vol. 2013, Article ID 589610, 11 pages, 2013.

[16] F. Bayrakceken and K. Yegin, "Fluorescence, decay time, and structural change of laser dye cresyl violet in solution due to microwave irradiation at gsm 900/1800 mobile phone frequencies," *International Journal of Photoenergy*, vol. 2012, Article ID 965426, 4 pages, 2012.

[17] N. McCullagh and P. S. L. M. Barreto, "A new two-party identity-based authenticated key agreement," in *Topics in Cryptology—CT-RSA 2005*, vol. 3376, pp. 262–274, Springer, Berlin, Germany, 2005.

[18] M. Aydos, B. Sunar, and C. Koc, "An elliptic curve cryptography based authentication and key agreement protocol for wireless communication," in *Proceedings of the 2nd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications Symposium on Information Theory*, 1998.

[19] M. J. Beller, L.-F. Chang, and Y. Yacobi, "Privacy and authentication on a portable communications system," *IEEE Journal on Selected Areas in Communications*, vol. 11, no. 6, pp. 821–829, 1993.

[20] C.-C. Lo and Y.-J. Chen, "Secure communication mechanisms for GSM networks," *IEEE Transactions on Consumer Electronics*, vol. 45, no. 4, pp. 1074–1080, 1999.

[21] A. Klimm, B. Glas, M. Wachs, S. Vogel, K. D. Müller-Glaser, and J. Becker, "A security scheme for dependable key insertion in mobile embedded devices," *International Journal of Reconfigurable Computing*, vol. 2011, Article ID 820454, 19 pages, 2011.

[22] W. Ren, J. Song, M. Lei, and Y. Ren, "BVS: a lightweight forward and backward secure scheme for PMU communications in smart grid," *International Journal of Digital Multimedia Broadcasting*, vol. 2011, Article ID 382147, 9 pages, 2011.

[23] Y. Kong and B. Phillips, "Revisiting sum of residues modular multiplication," *Journal of Electrical and Computer Engineering*, vol. 2010, Article ID 657076, 9 pages, 2010.

[24] X. Lu, W. Wang, and J. Ma, "Authentication and integrity in the smart grid: an empirical study in substation automation systems," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 175262, 13 pages, 2012.

[25] Certicom Research: Standard for efficient cryptography—SEC 1: Elliptic curve cryptography, 2000, http://www.secg.org/.

[26] Z. Cheng, M. Nistazakis, R. Comley, and L. Vasiu, "On the indistinguishability-based security model of key agreement protocols-simple cases," in *Proceedings of the International Conference on Applied Cryptography and Network Security (ACNS '04)*, vol. 4, 2004.

[27] W. Shi and P. Gong, "A new user authentication protocol for wireless sensor networks using elliptic curves cryptography," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 730831, 7 pages, 2013.

[28] J. Portilla, A. Otero, E. de la Torre et al., "Adaptable security in wireless sensor networks by using reconfigurable ECC hardware coprocessors," *International Journal of Distributed Sensor Networks*, vol. 2011, Article ID 740823, 12 pages, 2011.

[29] National Institute of Standards and Technology, NIST Special Publication 800-56: recommendation on key establishment schemes, Draft 2.0, 2003, http://csrc.nist.gov/groups/ST/toolkit/index.html.

[30] R. Sakai, K. Ohgishi, and M. Kasahara, "Cryptosystems based on pairing," in *Proceedings of the Symposium on Cryptography and Information Security*, pp. 135–148, Okinawa, Japan, 2000.

[31] X. Cao, W. Kou, and X. Du, "A pairing-free identity-based authenticated key agreement protocol with minimal message exchanges," *Information Sciences*, vol. 180, no. 15, pp. 2895–2903, 2010.

[32] L. Chen and C. Kudla, "Identity based authenticated key agreement protocols from pairings," in *Proceedings of the 16th IEEE Computer Security Foundations Workshop*, pp. 219–233, 2003.

[33] E. Okamoto, "Proposal for identity-based key distribution systems," *Electronics Letters*, vol. 22, no. 24, pp. 1283–1284, 1986.

[34] K. Shim, "Efficient ID-based authenticated key agreement protocol based on Weil pairing," *Electronics Letters*, vol. 39, no. 8, pp. 653–654, 2003.

[35] N. P. Smart, "Identity-based authenticated key agreement protocol based on Weil pairing," *Electronics Letters*, vol. 38, no. 13, pp. 630–632, 2002.

[36] K. Tanaka and E. Okamoto, "Key distribution system for mail systems using ID-related information directory," *Computers and Security*, vol. 10, no. 1, pp. 25–33, 1991.

[37] H. Sun and B. Hsieh, "Security analysis of shims authenticated key agreement protocols from pairings," Tech. Rep. 113, 2003.

[38] K. Choo, "Revisit of mccullagh-barreto two-party id-based authenticated key agreement protocols," *International Journal of Network Security*, vol. 1, no. 3, pp. 154–160, 2005.

[39] X. Wang and J. Zhao, "An improved key agreement protocol based on chaos," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 12, pp. 4052–4057, 2010.

[40] D. Xiao, X. Liao, and S. Deng, "A novel key agreement protocol based on chaotic maps," *Information Sciences*, vol. 177, no. 4, pp. 1136–1142, 2007.

[41] S. Han, "Security of a key agreement protocol based on chaotic maps," *Chaos, Solitons and Fractals*, vol. 38, no. 3, pp. 764–768, 2008.

[42] E. Chang and S. Han, "Using passphrase to construct key agreement. CBS-IS," Tech. Rep., Curtin University of Technology, 2006.

[43] S. Han and E. Chang, "Chaotic map based key agreement with/out clock synchronization," *Chaos, Solitons and Fractals*, vol. 39, no. 3, pp. 1283–1289, 2009.

[44] X.-Y. Wang and J.-F. Zhao, "Cryptanalysis on a parallel keyed hash function based on chaotic neural network," *Neurocomputing*, vol. 73, no. 16–18, pp. 3224–3228, 2010.

[45] P. Bergamo, P. D'Arco, A. de Santis, and L. Kocarev, "Security of public-key cryptosystems based on Chebyshev polynomials," *IEEE Transactions on Circuits and Systems I*, vol. 52, no. 7, pp. 1382–1393, 2005.

[46] Y. Niu and X. Wang, "An anonymous key agreement protocol based on chaotic maps," *Communications in Nonlinear Science and Numerical Simulation*, vol. 16, no. 4, pp. 1986–1992, 2011.

[47] H.-R. Tseng, R.-H. Jan, and W. Yang, "A chaotic maps-based key agreement protocol that preserves user anonymity," in *Proceedings of the IEEE International Conference on Communications (ICC '09)*, pp. 1–6, June 2009.

[48] T. Haist and W. Osten, "White-light interferometric method for secure key distribution," *The Journal of Supercomputing*, vol. 62, no. 2, pp. 656–662, 2012.

[49] J. Álvarez-Bermejo, N. Antequera, R. García-Rubio, and J. López-Ramos, "A scalable server for key distribution and its application to accounting," *The Journal of Supercomputing*, vol. 64, no. 1, pp. 132–143, 2013.

[50] Wikipedia, Wi-Fi Direct, 2012, http://en.wikipedia.org/wiki/Wi-Fi_Direct/.

[51] S. Bluetooth, Bluetooth specification, 2012, http://www.Bluetooth.com/.

[52] C. Bisdikian, "An overview of the Bluetooth wireless technology," *IEEE Communications Magazine*, vol. 39, no. 12, pp. 86–94, 2001.

[53] A. Specification, Infrared Data Association (IrDA) Std, 1998.