applied sciences

Article

Formal Security-Proved Mobile Anonymous Authentication Protocols with Credit-Based Chargeability and Controllable Privacy [†]

Chun-I Fan [1,*,‡] and Vincent Shi-Ming Huang [2,‡]

[1] Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung 80424, Taiwan

[2] Big Data Platform Team, CTO Office, Foxconn, Kaohsiung 80661, Taiwan; vincent.sm.huang@gmail.com

[*] Correspondence: cifan@mail.cse.nsysu.edu.tw; Tel.: +886-7-525-2000 (ext. 4346)

[†] A partial result of this research was presented at International Workshop on Security, Nara, Japan, 29–31 October 2007.

[‡] These authors contributed equally to this work.

Academic Editor: Christos Verikoukis

Received: 9 April 2016; Accepted: 9 June 2016; Published: 17 June 2016

Abstract: Smart mobile phones are widely popularized and advanced mobile communication services are provided increasingly often, such that ubiquitous computing environments will soon be a reality. However, there are many security threats to mobile networks and their impact on security is more serious than that in wireline networks owing to the features of wireless transmissions and the ubiquity property. The secret information which mobile users carry may be stolen by malicious entities. To guarantee the quality of advanced services, security and privacy would be important issues when users roam within various mobile networks. In this manuscript, an anonymous authentication scheme will be proposed to protect the security of the network system and the privacy of users. Not only does the proposed scheme provide mutual authentication between each user and the system, but also each user's identity is kept secret against anyone else, including the system. Although the system anonymously authenticates the users, it can still generate correct bills to charge these anonymous users via a credit-based solution instead of debit-based ones. Furthermore, our protocols also achieve fair privacy which allows the judge to revoke the anonymity and trace the illegal users when they have misused the anonymity property, for example, if they have committed crimes. Finally, in this paper, we also carry out complete theoretical proofs on each claimed security property.

Keywords: mutual authentication; anonymity; fair privacy; mobile networks; ubiquitous computing

1. Introduction

Recently, mobile communication is becoming more and more popular such that many applications and services are provided in the mobile network environments [1]. Moreover, some countries have constructed wireless network architectures of 4G (4th Generation) mobile networks. There is also smart mobile equipment that has been produced in order for people to enjoy mobile services anywhere and anytime. It is obvious that mobile computing will penetrate people's lives in the near future. Convenient mobile network services and powerful mobile equipment will make people all around the world become willing to join the society of mobile communications.

Mobile users may process important documents or secret personal information in their mobile equipment when they roam around the mobile networks. They might worry about whether it is secure for them to carry their important data to the mobile networks. When mobile users exchange messages in the mobile networks, they will face lots of security threats. The eavesdroppers may try to obtain their transmitted messages, their real identities, and even their locations where they are roaming around [2]. The more information the eavesdroppers know, the less security and privacy the mobile users preserve. Sometimes the vicious insiders of the system operator would disclose the classified information of mobile users. Any system without maintaining user privacy will not be acceptable in the future [3–5].

There exist some weaknesses on user privacy in the existent 2G mobile network system. Each mobile user's alias, TMSI, can be linked to her/his real identity, IMSI, by attackers when the VLR requests her/him to retransmit her/his IMSI. The 2G mobile network also has no design for satisfying mutual authentication and protecting the users' privacy against the system operator. A mobile user may be cheated by some fake base stations in a mobile network system due to lack of mutual authentication. Although the 3G system has provided mutual authentication, the privacy or anonymity of mobile users has not been sufficiently considered yet.

Most of the proposed authentication schemes [6–12] which emphasize the privacy of mobile users usually assign an anonymous identity to each user. A mobile user will obtain an anonymous identity after she/he is successfully authenticated by the system operator, and she/he will take this valid alias to roam over the mobile networks. The eavesdroppers do not know the relation between her/his real identity and alias, but the system operator can derive the relation. To protect the user's privacy perfectly, we hope that anyone else, even the system operator, cannot derive such relations either. Owing to the unlinkability property, the technique of *blind signatures* [13] can help us with realizing complete anonymity for mobile users.

Another problem is that once a mobile user gets anonymity, how can the system operator charge her/him when she/he requests the mobile network services via an anonymous identity? Especially, how can the system charge the user via a credit-based way, which is the most commonly-used billing solution and has been accepted by almost all of the mobile users? Further more, if there is some mobile user who misused the anonymity property to commit crimes, how can the judge handle it? All of the current solutions cannot cope with all of the above problems at the same time.

In our solution, every mobile user is anonymous from the system operator and any other person's point of view when she/he is accessing the mobile network resources. Furthermore, the system operator can charge the mobile user according to the communication time the user consumed via a credit-based way. Moreover, we also consider the issue of fair privacy. The privacy of the mobile users who misused the anonymity property can be revoked by the judge, and the police can trace the criminals who have gotten anonymity. This is the property of *fair privacy*. We simultaneously realize the *anonymity*, *credit-based chargeability*, and *fair privacy* (revokeability and traceability) in our proposed authentication protocols for mobile communications.

We produced a related work [14] which introduced the basic idea of this research. In this manuscript, we proposed more security features: Unlinkability, Unforgeability, Tamper Resistance, Swindling Resistance, Secure Mutual Authentication, and Secure Authenticated Key Exchange. Furthermore, the formal security proofs guarantee the security strength of the proposed system. Besides, we also did implementation to show the practical computation cost on cellphone.

2. Some Requirements for Anonymous Authentication

In mobile network environments, we need the following requirements for anonymous authentication.

1. Dynamic anonymous identity: When an anonymous user uses the same anonymous identity to roam over the mobile network for all sessions, her/his identity may be exposed by analyzing her/his behavior. We think that an anonymous user should use different anonymous identities for different sessions when she/he roams over the mobile network.
2. No relation between any two aliases: The privacy of a mobile user will be broken if the relations between any two aliases of the user are disclosed.
3. No mapping table, which contains the mapping between each real identity and its corresponding anonymous identity, stored in the system operator: The system operator authenticates an

anonymous user directly without maintaining a database to record a mapping between all of the user's anonymous identities and the user's real identity. This will make it possible for the user to gain her/his privacy against the system, and the system can save its storage space.

4. Authenticated key agreement: After anonymous authentication between the system and an anonymous user, a shared session key will be established. If the user shares a long-term key with the system in advance and derives the session key via the shared long-term key, the system can trace her/him by recognizing the long-term key embedded in the session key. Hence, in order to preserve user anonymity, the system and the user have to establish their session key without sharing any key or information in advance. Besides, all session keys should be mutually independent from each other.
5. Traceability in some situations: If there exist malicious users, a trusted third party must be able to revoke their privacy. An anonymous authentication protocol should own the feature of revokable anonymity in order to deal with the above situation.
6. Credit-based chargeability: When a user conceals her/his identity from the system operator, it will be hard for the system to charge the anonymous user after she/he utilizes the network services. An anonymous authentication scheme for mobile communications should allow the system operator to charge anonymous mobile users via the popularized credit-based way without revealing their identities. The credit-based chargeability in the proposed system means that any user does not need to pay, even pay with a credit card, before she/he uses the services of the system. Every consumption of the user will be accumulated in her/his ticket. The ticket has a life cycle and the user should return the ticket to the system at the end of the cycle, say the end of each month. Finally, the system will send the user a bill which includes the total amount of the consumption retrieved from the ticket.

3. The Proposed Protocols

First, we define and explain some notations as follows:

1. MS, H, V: These are three participants in our protocols. MS is a mobile user, H is the server of the home network, and V is the server of a visiting network.
2. ID_{MS}: the real identity of MS.
3. E_x, D_x: E_x is a semantic secure encryption function [15] and D_x is the decryption function corresponding to E_x where x can be an input symmetric key or public/private key.
4. $k_{ms\_h}, k_{ms\_v}, k_{v\_h}$: The shared session keys between MS and H, MS and V, and V and H, respectively.
5. (pk_J, sk_J) and (pk_V, sk_V): (pk_J, sk_J) is the public/private key pair of the judge and (pk_V, sk_V) is the public/private key pair of V.
6. l_r: a security parameter.
7. F_1, F_2, and F_3: three one-way hash functions.
8. A judge device: The judge issues a tamper-resistant device which contains {a random-number generator, a symmetric-key cryptosystem, a public-key cryptosystem, a public-private key pair of the judge, F_1, F_2}. This device will be integrated into the system of H. It is impossible to steal or modify any information embedded in the device. In our scheme, the judge is an off-line party, *i.e.*, the judge does not need to keep connection with H in our protocols, but the judge device does. In practice, the judge device can be implemented by the technique of TPM (Trusted Platform Module) [16] which is maintained by the Trusted Computing Group [17]. Nowadays, TPMs are also embedded in mobile phones and notebook computers [18].
9. γ: This is a due date. As shown in Figure 1, if a mobile user requests a ticket for communication in time slot P_i, H will assign her/him the due date γ_{i+1} where γ_{i+1} is the last day of next time slot P_{i+1}. H assigns the same γ to each mobile user who requests a ticket in the same time slot. All time slots are equally long.

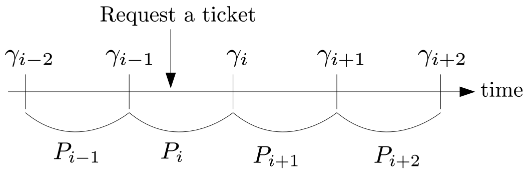


Figure 1. The time slots.

Our scheme consists of four protocols which are described in Sections 3.3–3.6, respectively. In our scheme, a mobile user requests an anonymous ticket by performing the protocol in Section 3.3. Then she/he can use the anonymous ticket for network services by executing the protocol in Section 3.4. After she/he performs the protocol in Section 3.4 for network services, H can charge her/him on the due date via the protocol in Section 3.5. Especially, if she/he does something illegal, the judge and the police can revoke her/his privacy or trace her/him through the protocol in Section 3.6.

3.1. Overview of Our Proposed Scheme

In this section, we describe how a mobile user obtains anonymity, how the system charges an anonymous user via a credit-based method, and how the judge revokes the anonymity from an anonymous user who does something malicious.

In our scheme, a mobile user has to request an anonymous ticket first and then uses it for authentication. As shown in Figure 2, when the mobile user request a ticket, the system, V and H, will send her/him a blinded ticket with her/his the identity ID and an initial value $w = 0$. The mobile user gets anonymity by unblinding the obtained ticket.

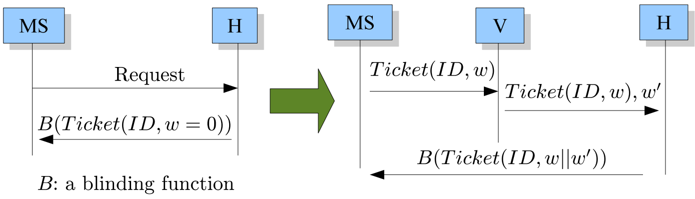


Figure 2. Overview of the proposed scheme.

The system charges the mobile user by a credit-based way as follows. Each time the mobile user consumes her/his ticket for mobile network services, the system will return her/him a new one which contains an updated value $(w + w')$ where w' is the value of the money H wants to charge the user for this time of communication service. Finally, the user must return her/his current unused ticket to the system on the due date of the ticket and the system will send her/him a bill which contains the accumulated value retrieved from the returned ticket of the user. During the services, the user is anonymous to the system under the protection from our proposed anonymity mechanism.

However, if the user does something malicious, the judge can revoke her/his anonymity by extracting her/his identity from the ticket and the police can trace the user via the embedded identity.

3.2. Key Generation

H chooses two distinct large primes p_H and q_H and computes $n_H = p_H q_H$. H also selects its public key e_H and the private key d_H such that $e_H d_H \equiv 1 \pmod{\phi(n_H)}$ where $\phi(n_H) = (p_H - 1)(q_H - 1)$. Finally, H publishes $\{n_H, e_H, F_1, F_2, F_3\}$ and keeps $\{p_H, q_H, d_H\}$ secret. Besides, H also publishes all time slots P_is, $i \in \{1, 2, 3, ...\}$, *i.e.*, all of the due dates γ's are published.

3.3. The Protocol for Requesting an Initial Anonymous Ticket

In our scheme, the mobile user, MS, can request an anonymous ticket by running the protocol in this section after she/he performs any existing secure mutual authentication protocol with the system, V and H. There exists a secure channel between V and H where the shared encryption key is k_{v_h}. This protocol contains the following steps and it is also shown in Figure 3.

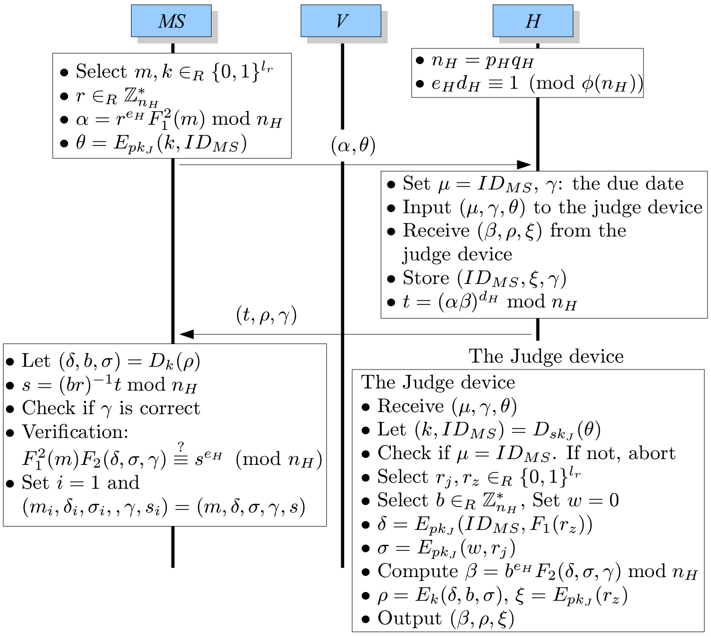

MS
V
H
• Select $m, k \in_R \{0,1\}^{l_r}$
• $r \in_R \mathbb{Z}^*_{n_H}$
• $\alpha = r^{e_H} F_1^2(m) \bmod n_H$
• $\theta = E_{pk_J}(k, ID_{MS})$
• $n_H = p_H q_H$
• $e_H d_H \equiv 1 \pmod{\phi(n_H)}$
(α, θ)
• Set $\mu = ID_{MS}$, γ: the due date
• Input (μ, γ, θ) to the judge device
• Receive (β, ρ, ξ) from the judge device
• Store (ID_{MS}, ξ, γ)
• $t = (\alpha\beta)^{d_H} \bmod n_H$
(t, ρ, γ)
• Let $(\delta, b, \sigma) = D_k(\rho)$
• $s = (br)^{-1} t \bmod n_H$
• Check if γ is correct
• Verification: $F_1^2(m) F_2(\delta, \sigma, \gamma) \overset{?}{\equiv} s^{e_H} \pmod{n_H}$
• Set $i = 1$ and $(m_i, \delta_i, \sigma_i, , \gamma, s_i) = (m, \delta, \sigma, \gamma, s)$
The Judge device
The Judge device
• Receive (μ, γ, θ)
• Let $(k, ID_{MS}) = D_{sk_J}(\theta)$
• Check if $\mu = ID_{MS}$. If not, abort
• Select $r_j, r_z \in_R \{0,1\}^{l_r}$
• Select $b \in_R \mathbb{Z}^*_{n_H}$, Set $w = 0$
• $\delta = E_{pk_J}(ID_{MS}, F_1(r_z))$
• $\sigma = E_{pk_J}(w, r_j)$
• Compute $\beta = b^{e_H} F_2(\delta, \sigma, \gamma) \bmod n_H$
• $\rho = E_k(\delta, b, \sigma)$, $\xi = E_{pk_J}(r_z)$
• Output (β, ρ, ξ)

Figure 3. The protocol for requesting an initial anonymous ticket.

1. $MS \rightarrow H : (\alpha, \theta)$.

   First, MS randomly generates two l_r-bit strings (m, k) and an integer $r \in \mathbb{Z}^*_{n_H}$. Then MS computes

   $$\alpha = r^{e_H} F_1^2(m) \bmod n_H \tag{1}$$

   and $\theta = E_{pk_J}(k, ID_{MS})$. Finally, MS submits (α, θ) to H.
2. $H \rightarrow$ The judge device : (μ, γ, θ).

   In this step, H knows that MS, whose real identity is ID_{MS}, wants to request a ticket. Let $\mu = ID_{MS}$ and γ be the last day of next time slot. Then H inputs (μ, γ, θ) into the judge device. H also records that ID_{MS} has bought a ticket in the current time slot and she/he will have to return an unused ticket on the due date γ for billing.
3. The judge device $\rightarrow H : (\beta, \rho, \xi)$.

   First, the judge device decrypts θ by computing $D_{sk_J}(\theta)$ and parses the result as (k, ID_{MS}). Then it checks if $\mu = ID_{MS}$. If true, it randomly generates two l_r-bit strings (r_j, r_z) and an integer $b \in \mathbb{Z}^*_{n_H}$. Then it sets $w = 0$ and computes $\delta = E_{pk_J}(ID_{MS}, F_1(r_z))$, $\sigma = E_{pk_J}(w, r_j)$, and

   $$\beta = b^{e_H} F_2(\delta, \sigma, \gamma) \bmod n_H \tag{2}$$

   Finally, it computes $\rho = E_k(\delta, b, \sigma)$ and $\xi = E_{pk_J}(r_z)$ and returns (β, ρ, ξ) to H.
4. $H \rightarrow MS : (t, \rho, \gamma)$.

   After receiving (β, ρ, ξ), H records (ID_{MS}, ξ, γ) and computes $t = (\alpha\beta)^{d_H} \bmod n_H$. Then it sends (t, ρ, γ) to MS.

5. Unblinding.

   After receiving (t, ρ, γ), *MS* checks if γ is the last day of next time slot. Then she/he decrypts ρ by computing $D_k(\rho)$ and parses the result as (δ, b, σ). She/He also computes

$$s = (br)^{-1} t \bmod n_H \tag{3}$$

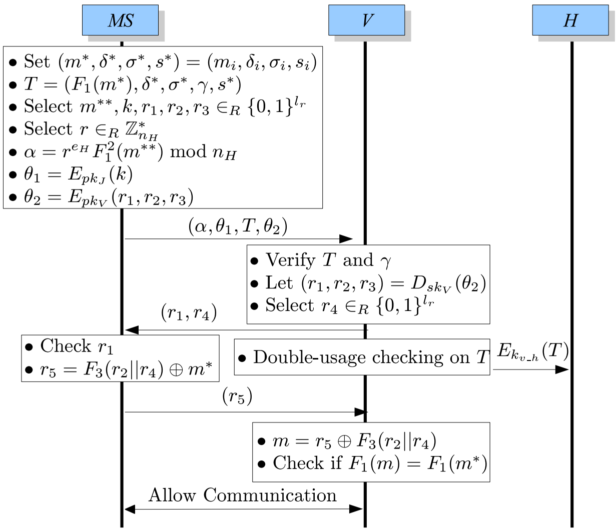
   Then she/he obtains a ticket $(m, \delta, \sigma, \gamma, s)$ and can verify it by examining if the following formula is true:

$$F_1^2(m) F_2(\delta, \sigma, \gamma) \equiv s^{e_H} \pmod{n_H} \tag{4}$$

   Finally, *MS* sets $i = 1$ and $(m_i, \delta_i, \sigma_i, \gamma, s_i) = (m, \delta, \sigma, \gamma, s)$ and then goes to the protocol of Section 3.4 when she/he decides to use the ticket to roam the mobile networks.

3.4. The Protocol for Using an Anonymous Ticket in the ith Round before the Due Date

This protocol makes them possible for the anonymous mobile user *MS* to perform mutual authentication with *V* and use her/his ticket for mobile network services. It contains the following steps and also is illustrated in Figure 4.



MS
V
H
• Set $(m^*, \delta^*, \sigma^*, s^*) = (m_i, \delta_i, \sigma_i, s_i)$
• $T = (F_1(m^*), \delta^*, \sigma^*, \gamma, s^*)$
• Select $m^{**}, k, r_1, r_2, r_3 \in_R \{0,1\}^{l_r}$
• Select $r \in_R \mathbb{Z}^*_{n_H}$
• $\alpha = r^{e_H} F_1^2(m^{**}) \bmod n_H$
• $\theta_1 = E_{pk_J}(k)$
• $\theta_2 = E_{pk_V}(r_1, r_2, r_3)$
$(\alpha, \theta_1, T, \theta_2)$
• Verify T and γ
• Let $(r_1, r_2, r_3) = D_{sk_V}(\theta_2)$
• Select $r_4 \in_R \{0,1\}^{l_r}$
(r_1, r_4)
• Check r_1
• $r_5 = F_3(r_2 || r_4) \oplus m^*$
• Double-usage checking on T
$E_{k_{v\_h}}(T)$
(r_5)
• $m = r_5 \oplus F_3(r_2 || r_4)$
• Check if $F_1(m) = F_1(m^*)$
Allow Communication

Figure 4. The protocol for using an anonymous ticket in the *i*-th round before the due date.

1. $MS \rightarrow V : (\alpha, \theta_1, T, \theta_2)$.

   First, *MS* sets $(m^*, \delta^*, \sigma^*, s^*) = (m_i, \delta_i, \sigma_i, s_i)$ and then prepares $T = (F_1(m^*), \delta^*, \sigma^*, \gamma, s^*)$ and randomly generates 5 l_r-bit strings $(m^{**}, k, r_1, r_2, r_3)$ and an integer $r \in \mathbb{Z}^*_{n_H}$. Furthermore, *MS* computes $\alpha = r^{e_H} F_1^2(m^{**}) \bmod n_H$, $\theta_1 = E_{pk_J}(k)$, and $\theta_2 = E_{pk_V}(r_1, r_2, r_3)$. Finally, *MS* submits $(\alpha, \theta_1, T, \theta_2)$ to *V*.
2. $V \rightarrow MS : (r_1, r_4)$.

   After receiving $(\alpha, \theta_1, T, \theta_2)$, *V* first verifies *T* by examining if

$$(s^*)^{e_H} \equiv F_1(F_1(m^*)) F_2(\delta^*, \sigma^*, \gamma) \pmod{n_H} \tag{5}$$

   and γ is not expired. If true, *V* decrypts θ_2 to get (r_1, r_2, r_3) and randomly generates an l_r-bit string r_4. Then *V* sends (r_1, r_4) to *MS*.
3. $V \rightarrow H : E_{k_{v\_h}}(T)$.

   After *V* sends (r_1, r_4) to *MS*, it also immediately submits $E_{k_{v\_h}}(T)$ to *H* in order to perform the double-using checking on *T*. If *T* is doubly used, the connection will be terminated.

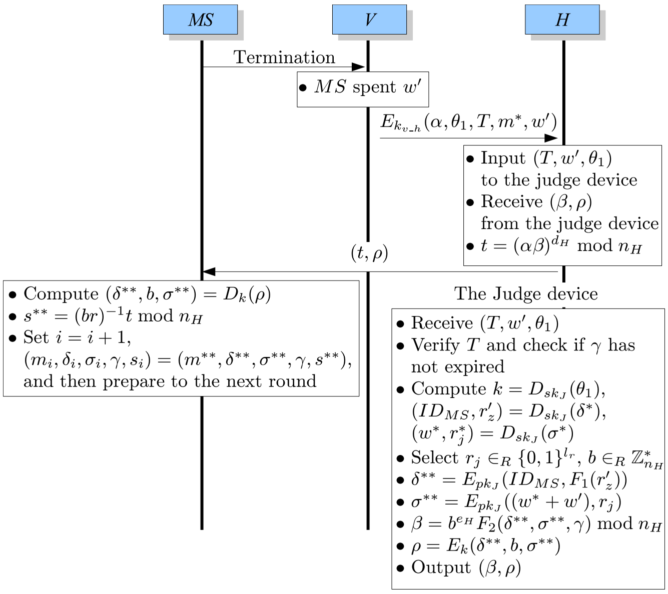
4. $MS \rightarrow V : (r_5)$.

   After receiving (r_1, r_4), MS checks if r_1 is the same as the one which was chosen by herself/himself. Then MS computes $r_5 = F_3(r_2||r_4) \oplus m^*$ and sends r_5 to V.

5. Allowing Communication:

   After receiving r_5, V computes $m = r_5 \oplus F_3(r_2||r_4)$ and checks if $F_1(m) = F_1(m^*)$ where $F_1(m^*)$ is retrieved from T. If true, V ensures that MS is the real owner of T. Therefore, V allows MS to communicate with it. During the communication, they can encrypt/decrypt their messages via the session key r_3.

After MS terminates her/his communication, she/he will get a returned ticket which will be used for the next round of authentication. As shown in Figure 5, she/he has to perform the following procedures with the system to obtain the returned ticket.



MS
V
H
Termination
• MS spent w'
$E_{k_{v\_h}}(\alpha, \theta_1, T, m^*, w')$
• Input (T, w', θ_1) to the judge device
• Receive (β, ρ) from the judge device
• $t = (\alpha\beta)^{d_H} \bmod n_H$
(t, ρ)
• Compute $(\delta^{**}, b, \sigma^{**}) = D_k(\rho)$
• $s^{**} = (br)^{-1} t \bmod n_H$
• Set $i = i + 1$, $(m_i, \delta_i, \sigma_i, \gamma, s_i) = (m^{**}, \delta^{**}, \sigma^{**}, \gamma, s^{**})$, and then prepare to the next round
The Judge device
• Receive (T, w', θ_1)
• Verify T and check if γ has not expired
• Compute $k = D_{sk_J}(\theta_1)$, $(ID_{MS}, r'_z) = D_{sk_J}(\delta^*)$, $(w^*, r^*_j) = D_{sk_J}(\sigma^*)$
• Select $r_j \in_R \{0,1\}^{l_r}$, $b \in_R \mathbb{Z}^*_{n_H}$
• $\delta^{**} = E_{pk_J}(ID_{MS}, F_1(r'_z))$
• $\sigma^{**} = E_{pk_J}((w^* + w'), r_j)$
• $\beta = b^{e_H} F_2(\delta^{**}, \sigma^{**}, \gamma) \bmod n_H$
• $\rho = E_k(\delta^{**}, b, \sigma^{**})$
• Output (β, ρ)

Figure 5. The protocol for terminating the communication and getting a returned ticket for the next round of communication.

1. $MS \rightarrow V$: (Termination).

   MS notifies V that she/he wants to terminate her/his communication.

2. $V \rightarrow H : (E_{k_{v\_h}}(\alpha, \theta_1, T, m^*, w'))$.

   After receiving the termination request from MS, V computes the spending value w' of MS according to the communication time or services utilized by MS. Then V sends $E_{k_{v\_h}}(\alpha, \theta_1, T, m^*, w')$ to H.

3. $H \rightarrow$ The judge device: (T, w', θ_1).

   H decrypts the message received from V and stores (T, m^*, w') into its database. Then H inputs (T, w', θ_1) into the judge device.

4. The judge device $\rightarrow H : (\beta, \rho)$.

   When receiving $(T,\ w',\ \theta_1)$, the judge device will verify T by (5) first and verify whether the due date γ embedded in T has expired or not. If one of the above verifications fails, the judge device will return an aborting signal. Otherwise, the judge device computes $k = D_{sk_J}(\theta_1)$, $(ID_{MS}, r'_z) = D_{sk_J}(\delta^*)$, and $(w^*, r^*_j) = D_{sk_J}(\sigma^*)$ where δ^* and σ^* are retrieved from T. Furthermore, it randomly selects a string $r_j \in \{0,1\}^{l_r}$ and an integer $b \in \mathbb{Z}^*_{n_H}$ and prepares $\delta^{**} = E_{pk_J}(ID_{MS}, F_1(r'_z))$, $\sigma^{**} = E_{pk_J}((w^* + w'), r_j)$, and $\beta = b^{e_H} F_2(\delta^{**}, \sigma^{**}, \gamma) \bmod n_H$. Finally, it computes $\rho = E_k(\delta^{**}, b, \sigma^{**})$ and outputs (β, ρ) to H.

5. $H \rightarrow MS : (t, \rho)$.

   After receiving (β, ρ), H computes $t = (\alpha\beta)^{d_H} \bmod n_H$ and returns (t, ρ) to MS.
6. Unblinding

   After receiving (t, ρ), MS computes $(\delta^{**}, b, \sigma^{**}) = D_k(\rho)$ and $s^{**} = (br)^{-1} t \bmod n_H$. MS obtains a new ticket as $(m^{**}, \delta^{**}, \sigma^{**}, \gamma, s^{**})$ which can be verified by checking whether $F_1^2(m^{**}) F_2(\delta^{**}, \sigma^{**}, \gamma) \equiv (s^{**})^{e_H} \pmod{n_H}$ is true or not. If true, MS sets $i = i + 1$ and $(m_i, \delta_i, \sigma_i, \gamma, s_i) = (m^{**}, \delta^{**}, \sigma^{**}, \gamma, s^{**})$, which is the new unused (fresh) ticket of the user. Thus, she/he can use the fresh ticket for the next round of communication before the due date, γ.

3.5. The Protocol for Charging Mobile Users

For each mobile user, MS, the system operator, H, calculates her/his bill through the following steps on the due date, γ:

1. MS returns her/his real identity and unused ticket, $(m^*, \delta^*, \sigma^*, \gamma, s^*)$, to H before the due date.
2. H checks that the ticket does not exist in its database and sends the ticket to the judge device.
3. The judge device verifies if the ticket is valid via (4) and checks if the γ has expired. If true, it computes $(w, r_j) = D_{sk_J}(\sigma^*)$ and returns the spending value w to H.
4. H adds w to the bill of MS and deletes the record which indicates that MS has ever requested a ticket.
5. Send the bill to MS.

Besides, if the mobile user wants to request a ticket after the due date, γ, she/he should perform the protocol of Section 3.3 again.

Our scheme adopts credit-based charging, i.e. the system charges each mobile user after it has finished a sequence of services for the user, just as the practical situation in the real world. It is different from the others which provided approaches of debit-based charging, i.e. each mobile user has to purchase payment token(s) before she/he starts accessing the services provided by the system [6,12]. What are the differences between charging mobile users in advance and charging them after the services? The followings are the reasons why we design our scheme to charge mobile users via a credit-based way.

1. Adaptability. In current GSM services, almost all of the systems adopt credit-based ways to charge users.
2. Reducing the relations between any two rounds of communication with one token only. There are two possible ways to charge a mobile user in advance (debit-based ways), which are described as follows:

   (a) The mobile user purchases a set of payment tokens from the system previously where each of the tokens is with a unit of value. In each round of communication, the mobile user sends a proper number of tokens to the system for payment. In this case, it is difficult for the system to derive the relation between any two rounds of communication since the tokens are independent one another. However, this will consume much storage and communication cost for recording and transmitting these tokens.

   (b) The mobile user purchases only one payment token from the system previously where the token is with a specific value w. In the following round of communication, the mobile user sends the token to the system for payment and then the system returns a new token with value $(w - w_1)$ if the user consumes w_1 value of that token. In this mechanism, the mobile user just needs to store one token. However, this will cause defective privacy. When the system returns one token with value $(w - w_1)$ to the user, the system knows that the user will use the token with value $(w - w_1)$ in the next round. There exists a relation between these two rounds of communication.

Our scheme allows a user to store one token and greatly reduces the relations between any two rounds of communication from the system's point of view. All of the users return their unused tickets to the system for charging and thus the system knows the total spending value of every user in the previous time slot. However, it is difficult for the system to trace a specific user by finding out all of her/his spending values from the spending value pool which contains all spending values of all users in the previous time slot. This is the **subset sum problem**, shown below, which is NP-Hard [19]. The proposed system makes it computationally infeasible to link any two rounds of communication with the assumption of large subset sizes.

Definition 1. *Given a vector over integers* $A = (a_1, a_2, \ldots, a_n)$ *and a positive integer* s*, called the sum, compute a solution vector* $X = (x_1, x_2, \ldots, x_n)$ *where* $x_i \in \{0, 1\}$ *such that* $AX = a_1x_1 + a_2x_2 + \cdots + a_nx_n = s$.

The integer s can be regarded as the total spending value of a mobile user and the vector A contains all spending values in the spending value pool.

3. Free from the problem of overspending. In debit-based charging methods (both of the above two ways (a) and (b)), when a mobile user shows her/his token(s) to the system for communicating, her/his communication will be terminated if the tokens or the token's value are used up. It will cause inconvenience for the mobile user. If the system does not terminate the communication, the mobile user will overspend the token(s) and the system must perform extra procedures to deal with the situation. In our scheme, based on a credit-based method, the above problem can be avoided.

3.6. The Protocol for Privacy Revoking

In some situations, H or the judge needs to disclose the identity of an anonymous mobile user. For example, some user commits a crime; the police want to trace some criminals; or some mobile users who do something harmful for H. Our scheme supports two ways to trace illegal anonymous mobile users.

1. Tracing the mobile user by a designated ticket: Once an anonymous user imposes on anonymity to commit a crime, her/his ticket will be reported to the judge. Assume that the ticket is $(m', \delta', \sigma', \gamma', s')$. The judge will extract δ' from the ticket and parse $D_{pk_J}(\delta')$ to get ID_{MS}.
2. Tracing the tickets by a designated mobile user: If the police want to trace a criminal (whose real identity is ID_{MS}) in the time slot P_i, the police can send (ID_{MS}, γ_{i+1}) to H and ask H and the judge to disclose the privacy of the criminal. In this case, H will retrieve ξ from its stored records according to (ID_{MS}, γ_{i+1}) and send ξ to the judge. After decrypting ξ and obtaining r_z, the judge computes

$$\begin{cases} \delta_1 = E_{pk_J}(ID_{MS}, F_1^1(r_z)) \\ \delta_2 = E_{pk_J}(ID_{MS}, F_1^2(r_z)) \\ \delta_3 = E_{pk_J}(ID_{MS}, F_1^3(r_z)) \\ \vdots \\ \delta_i = E_{pk_J}(ID_{MS}, F_1^i(r_z)) \end{cases} \tag{6}$$

Then, it sends $\{\delta_1, \delta_2, \delta_3, \ldots, \delta_i\}$ to H, and H can help the police to trace the mobile user in time slot P_i via the above set. In our scheme, the mobile user takes the anonymous ticket containing δ_1 for her/his first round of communication, the ticket containing δ_2 for the second round, and so forth. According to this order, H can trace the communication activities of the criminal from the first round to the ith round via $\{\delta_1, \delta_2, \delta_3, \ldots, \delta_i\}$.

3.7. Exceptions

In addition to the above issues, there are three exceptions that may happen in our scheme. One is that the mobile user denies returning her/his ticket for billing on the due date. Another is that the mobile user lost her/his ticket (or lost her/his mobile device), and the other one is that the mobile user's communication is terminated abnormally.

1. The mobile user denies returning her/his ticket for billing on the due date: After the due date γ, if there is any mobile user who has not returned her/his unused ticket yet, H will send a list $\mathscr{L}$ to the judge where $\mathscr{L}$ contains the identities of the mobile users who did not return their unused tickets. According to $\mathscr{L}$, the judge sends a payment notification to each mobile user on $\mathscr{L}$ and announces another due date $\gamma^{'}$. If a mobile user, say ID_{MS}, has not returned her/his unused ticket on the new due date $\gamma^{'}$, the judge will compute the set $\{\delta_1, \delta_2, \delta_3, \ldots\}$ according to ID_{MS} via Equation (6) and then sends it to H. Let T_i denote the ith ticket, *i.e.*, the ticket containing δ_i. Assume that the mobile user denied returning T_{i+1}. H can find (T_i, w') from its database via δ_i. When H finds (T_i, w'), the judge can help H with extracting the spending value w^* from T_i, and then H computes $w'' = w^* + w'$ and adds w'' to the bill of the mobile user ID_{MS}.
2. The mobile user lost her/his ticket: When a mobile user, say ID_{MS}, lost her/his unused ticket T_i, she/he must ask H to freeze her/his unused ticket or it may be used by a malicious user. After an authorization process, for example, the mobile user signs a document to show that she/he agrees H to ask the judge to compute $\{\delta_1, \delta_2, \ldots\}$ where the mobile user authorizes H to reveal her/his privacy, H sends (ID_{MS}, ξ) to the judge to compute $\{\delta_1, \delta_2, \delta_3, \ldots\}$ by Equation (6). Assume that the mobile user lost T_i. H must deny the services for $T_i, T_{i+1}, T_{i+2}, \ldots$ by $\delta_i, \delta_{i+1}, \delta_{i+2}, \ldots$, respectively, where $i \in \mathbb{N}$. Besides, H finds (T_{i-1}, w') from its database via δ_{i-1} and sends T_{i-1} to the judge to extract the accumulated spending value w^* from T_{i-1}. After the judge returns w^* to H, H adds $(w^* + w')$ to the bill of the mobile user.

   In order to handle this exception, the privacy of $T_1, T_2, T_3, \ldots, T_{i-1}$ of the mobile user will be revealed. However, if the mobile user remembered how many tickets she/he has used, she/he can still preserve her/his privacy. For example, a mobile user lost her/his unused ticket, and she/he remembers that she/he has consumed 4 tickets. Then the judge just needs to compute $\{\delta_4, \delta_5, \delta_6, \ldots\}$ for H, and $\{\delta_1, \delta_2, \delta_3\}$ are still kept secret for the mobile user. H will check if δ_j' exists in its database where $j' = \{4, 5, 6, \ldots\}$. If $\delta_{j'}$ exists in its database and $\delta_{j'+1}$ does not, H will retrieve $(T_{j'}, w')$, which will be used for charging the mobile user, from the database via $\delta_{j'}$. After the mobile user freezes her/his lost ticket, she/he can perform the protocol in Section 3.3 again to request a new ticket.
3. The communication is terminated abnormally: Consider the case that the communication of Step 5 in Section 3.4 is abnormally terminated, *i.e.*, the mobile user does not receive a renewed ticket. We assume that each time when the mobile user receives (t, ρ) successfully, she/he will return an ACK to H. Once H does not receive ACK, it will store (t, ρ) and $(m^*, \delta^*, \sigma^*, \gamma, s^*)$ into an unsuccessful communication record. Thus, the mobile user can retransmit $(m^*, \delta^*, \sigma^*, \gamma, s^*)$ to H, and H can re-send (t, ρ) to the mobile user.

   Even though the mobile user lost all information in the abnormal termination, *i.e.*, the mobile user cannot unblind t and decrypt ρ when H retransmits them to her/him, she/he can notify H that she/he lost her/his ticket and then go back to the protocol of requesting an anonymous ticket (Section 3.3) to request a new one. In such a case, H can still correctly charge the mobile user and the mobile user can still use the new ticket for the following communications.

4. Security Proofs

4.1. Security Requirements

- Unlinkability: No one except the judge can trace a user when she/he is using her/his ticket for roaming the mobile networks.
- Ticket Unforgeability: None can forge a ticket without performing the requesting ticket protocol of Section 3.3 with the system.
- Tamper Resistance: The triple (δ, σ, γ) in a ticket cannot be modified.
- Ticket Swindling Resistance: Anyone else cannot consume an eavesdropped ticket for communication services where the ticket is owned by some user.
- Mutual Authentication: Neither a mobile user without a valid ticket nor an illegal system can pass the authentication.
- Secure Authenticated Key Exchange: After mutual authentication, a mobile user and V can share a common session key unknown to any eavesdropper.

4.2. Unlinkability

In our scheme, a mobile user gets an initial anonymous ticket by running the requesting ticket protocol in Section 3.3 and obtains a renewed one when running the using ticket protocol in Section 3.4. In either Sections 3.3 or 3.4, the mobile user performs the similar operations to get an anonymous ticket. Here, we define a game as follows.

Definition 2. *Let k be a security parameter, MS_0 and MS_1 be two honest mobile users, and $\mathcal{J}$ be the judge. The game is shown below.*

Step 1. According to our proposed scheme, H sets up the system parameters which contain H's public key (e_H, n_H), secret key (d_H, p_H, q_H), and hash functions (F_1, F_2, F_3). $\mathcal{J}$ generates its key pair (pk_J, sk_J).
Step 2. H generates and outputs two messages m_0 and m_1.
Step 3. Randomly pick a bit $b \in \{0, 1\}$ and place m_b and m_{1-b} on the private input tapes of MS_0 and MS_1, respectively. The bit b will not be revealed to H.
Step 4. H performs the protocol (Sections 3.3 or 3.4) of our scheme with MS_0 and MS_1, respectively, to issue blinded tickets to them.
Step 5. If MS_0 and MS_1 output two tickets which are $(m_b, \delta_b, \sigma_b, \gamma_b, s_b)$ and $(m_{1-b}, \delta_{1-b}, \sigma_{1-b}, \gamma_{1-b}, s_{1-b})$ on their private tapes, respectively, give the two 5-tuples in a random order to H; Otherwise, $\perp$ is given to H.
Step 6. H outputs $b' \in \{0, 1\}$ as the guess of b. H wins the game if $b = b'$. Define the advantage of H as

$$Adv_H^{Linkability}(k) = |2P[b' = b] - 1|$$

where $P[b' = b]$ denotes the probability of $b' = b$.

Definition 3 (Unlinkability)**.** *In our scheme, the protocols in Sections 3.3 and 3.4 satisfy the unlinkability property if the advantage $Adv_H^{Linkability}(k)$ in the game of Definition 2 is negligible.*

Theorem 1. *If E_{pk_J} and E_k are two semantic secure encryption functions, our proposed protocols in Sections 3.3 and 3.4 satisfy the unlinkability property.*

Proof. In Step 5 of Definition 2, if H is given $\perp$, it will determine b with probability $\frac{1}{2}$ which is exactly the same as a random guess of b.

We assume that H gets $(m_b, \delta_b, \sigma_b, \gamma_b, s_b)$ and $(m_{1-b}, \delta_{1-b}, \sigma_{1-b}, \gamma_{1-b}, s_{1-b})$. Let $(\alpha_i, \theta_i, \gamma_i, \mu_i, \beta_i, \rho_i, \xi_i, t_i)$ and $(\alpha_i, \theta_{1_i}, T_i, m_i^*, w_i', \beta_i, \rho_i, t_i)$ be the view of H to the protocol of Section 3.3 and the protocol of Section 3.4, respectively, where $i \in \{0, 1\}$ and $\gamma_0 = \gamma_1$.

Consider $(\theta_i, \gamma_i, \mu_i, \rho_i, \xi_i)$ in Section 3.3 where $\theta_i = E_{pk_J}(k_i, ID_{MS_i})$, $\mu_i = ID_{MS_i}$, $\rho_i = E_{k_i}(\delta_i, b_i, \sigma_i)$, and $\xi_i = E_{pk_J}(r_{z_i})$, $(\theta_{1_i}, T_i = \{F_1(m_i^*), \delta_i^*, \sigma_i^*, \gamma_i, s_i^*\}, m_i^*, w_i', \rho_i)$ in Section 3.4 where $\theta_{1_i} = E_{pk_J}(k_i)$,

$\rho_i = E_{k_i}(\delta_i^{**}, b_i, \sigma_i^{**})$, and w_i' is encrypted in $\sigma_i = E_{pk_J}((w_i^* + w' + i), r_{j_i})$, and (δ_i^*, σ_i^*) in T_i where $\delta_i^* = E_{pk_J}(ID_{MS_i}, F_1(\bar{r}'_{z_i}))$ and $\sigma_i^* = E_{pk_J}((\bar{w}_i^* + \bar{w}_i'), \bar{r}'_{j_i})$. Since E_{pk_J} and E_{k_i} are semantically secure encryption functions, the information encrypted in the above ciphertexts will not be revealed.

In both Sections 3.3 and 3.4, (α_i, β_i, t_i) can be considered as follows. For $(m, \delta, \sigma, \gamma, s) \in \{(m_0, \delta_0, \sigma_0, \gamma_0, s_0), (m_1, \delta_1, \sigma_1, \gamma_1, s_1)\}$ and (α_i, β_i, t_i), $i \in \{0,1\}$, there always exists a pair (r_i', b_i') such that H can compute $r_i' = (\alpha_i F_1^2(m)^{-1})^{d_H} \bmod n_H$ via (1) and $b_i' = (\beta_i F_2(\delta, \sigma, \gamma)^{-1})^{d_H} \bmod n_H$ via (2). Thus, (3) is satisfied owing to $t_i = (\alpha_i \beta_i)^{d_H} \bmod n_H$ and $s \equiv (F_1^2(m) F_2(\delta, \sigma, \gamma))^{d_H} \pmod{n_H}$.

From the above, given any $(m, \delta, \sigma, \gamma, s) \in \{(m_0, \delta_0, \sigma_0, \gamma_0, s_0), (m_1, \delta_1, \sigma_1, \gamma_1, s_1)\}$ and (α_i, β_i, t_i), $i \in \{0,1\}$, there always exists a corresponding pair (r_i', b_i') such that Equations (1)–(3) are satisfied.

Hence, considering Step 6 of the game, H successes in determining b with probability $\frac{1}{2}$. We have that $P[b' = b] = \frac{1}{2}$ and $Adv_H^{Linkability}(k) = 0$. Therefore, the proposed scheme satisfies the unlinkability property. □

4.3. Ticket Unforgeability

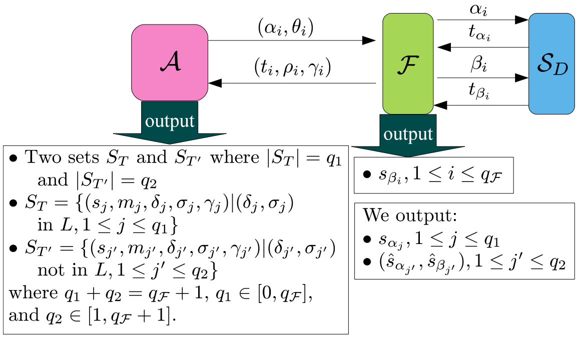
In 2003, Bellare *et al.* introduced a problem called the RSA Chosen Target Inversion (RSA-CTI) Problem [20]. Then they proved that the Full Domain Hash RSA (FDH-RSA) blind signature is unforgeable as long as the RSA-CTI problem is hard. In this section, we will show that the ticket requesting protocol of Section 3.3 and the ticket using protocol of Section 3.4 satisfy unforgeability as long as the FDH-RSA blind signature is with unforgeability.

Theorem 2. *If an attacker $\mathcal{A}$ can forge an unused ticket in the proposed scheme (Sections 3.3 or 3.4) with probability at least $\epsilon_{\mathcal{A}}$ in time $t_{\mathcal{A}}$, there exists a forger $\mathcal{F}$ that can break the unforgeability of the FDH-RSA blind signature with probability at least ϵ' in time t' such that*

$$\begin{cases} \epsilon' \geq \epsilon_{\mathcal{A}} \\ t' \approx t_{\mathcal{A}} + q_{\mathcal{F}} t_{\mathcal{F}} + 2 q_{\mathcal{F}} t_{S_D} \end{cases} \tag{7}$$

where $q_{\mathcal{F}}$ is the number of queries $\mathcal{A}$ makes to $\mathcal{F}$, $t_{\mathcal{F}}$ is the time for $\mathcal{F}$ to deal with a query, and t_{S_D} is the time for the FDH-RSA blind signing oracle to process a signing query.

Proof. The model of this proof is shown as Figure 6. Let S_D be the FDH-RSA blind signing oracle. The public key of S_D is (e_H, n_H). First, $\mathcal{F}$ initializes the environment by generating the public/private key pair (pk_J, sk_J) of the judge and selecting three hash functions (F_1, F_2, F_3). Then $\mathcal{F}$ publishes $(pk_J, e_H, n_H, F_1, F_2, F_3)$. $\mathcal{F}$ utilizes (e_H, n_H) as the public key of H of the system. $\mathcal{F}$ will simulate the system such that $\mathcal{A}$ can query $\mathcal{F}$ to get tickets. If $\mathcal{A}$ can output $q_{\mathcal{F}} + 1$ tickets after querying $\mathcal{F}$ $q_{\mathcal{F}}$ times, we can succeed in one-more forgery to break the unforgeability of the FDH-RSA blind signature scheme. Here, we just show how to simulate the ticket requesting protocol of Section 3.3. The simulation of the ticket using protocol in Section 3.4 is similar to that of the protocol in Section 3.3.



- Two sets S_T and $S_{T'}$ where $|S_T| = q_1$ and $|S_{T'}| = q_2$
- $S_T = \{(s_j, m_j, \delta_j, \sigma_j, \gamma_j) | (\delta_j, \sigma_j)$ in $L, 1 \leq j \leq q_1\}$
- $S_{T'} = \{(s_{j'}, m_{j'}, \delta_{j'}, \sigma_{j'}, \gamma_{j'}) | (\delta_{j'}, \sigma_{j'})$ not in $L, 1 \leq j' \leq q_2\}$

where $q_1 + q_2 = q_{\mathcal{F}} + 1$, $q_1 \in [0, q_{\mathcal{F}}]$, and $q_2 \in [1, q_{\mathcal{F}} + 1]$.

- $s_{\beta_i}, 1 \leq i \leq q_{\mathcal{F}}$

We output:
- $s_{\alpha_j}, 1 \leq j \leq q_1$
- $(\hat{s}_{\alpha_{j'}}, \hat{s}_{\beta_{j'}}), 1 \leq j' \leq q_2$

Figure 6. The model of the proof for unforgeability.

When $\mathcal{A}$ submits a query (α_i, θ_i) to $\mathcal{F}$, $\mathcal{F}$ will return (t_i, ρ_i, γ_i) to $\mathcal{A}$. $\mathcal{F}$ is depicted in Figure 7. Finally, $\mathcal{A}$ outputs $q_{\mathcal{F}} + 1$ tickets $(s_i, m_i, \delta_i, \sigma_i, \gamma_i)$ where $m_i \neq m_{i'}$, $\delta_i \neq \delta_{i'}$, $\sigma_i \neq \sigma_{i'}$, and $1 \leq i \neq i' \leq q_{\mathcal{F}} + 1$. The outputted tickets can be categorized into two subsets S_T and $S_{T'}$ where $|S_T| = q_1$ and $|S_{T'}| = q_2$. For each ticket $(s_j, m_j, \delta_j, \sigma_j, \gamma_j)$ in S_T, (δ_j, σ_j) is in L where $1 \leq j \leq q_1$, *i.e.*, the tickets in S_T are queried from $\mathcal{F}$. On the other hand, each ticket in $S_{T'}$ is forged by $\mathcal{A}$. We say that $\mathcal{A}$ successfully breaks our scheme if (1) $q_1 + q_2 = q_{\mathcal{F}} + 1$; (2) $q_1 \in [0, q_{\mathcal{F}}]$; and (3) $q_2 \in [1, q_{\mathcal{F}} + 1]$. In the followings, we will show that we can obtain $(q_{\mathcal{F}} + q_1 + 2q_2)$ signatures by querying S_D $(2q_{\mathcal{F}} + q_2)$ times where $(q_{\mathcal{F}} + q_1 + 2q_2) - (2q_{\mathcal{F}} + q_2) = q_1 + q_2 - q_{\mathcal{F}} = 1$.

```
$\mathcal{F}$ $(\alpha_i, \theta_i)$
1. Send $\alpha_i$ to $S_D$ and get $t_{\alpha_i} (= \alpha_i^{d_H} \bmod n_H)$;
2. Let $(k_i, ID_i) = D_{sk_J}(\theta_i)$;
3. Select $b_i \in_R \mathbb{Z}^*_{n_H}$ and $r_{j_i}, r_{z_i} \in_R \{0,1\}^{l_r}$;
4. Set $w_i = 0$ and $\gamma_i$ as the due date;
5. Compute $\sigma_i = E_{pk_J}(w_i, r_{j_i})$;
6. Compute $\delta_i = E_{pk_J}(ID_i, F_1(r_{z_i}))$;
7. Compute $\beta_i = b_i^{e_H} F_2(\delta_i, \sigma_i, \gamma_i) \bmod n_H$;
8. Send $\beta_i$ to $S_D$ and get $t_{\beta_i} (= \beta_i^{d_H} \bmod n_H)$;
9. Unblind: $s_{\beta_i} = b_i^{-1} t_{\beta_i} \bmod n_H$;
10. Compute $t_i = t_{\alpha_i} t_{\beta_i} \bmod n_H$;
11. Store $(s_{\beta_i}, \delta_i, \sigma_i, \gamma_i)$ in $L$;
12. Compute $\rho_i = E_{k_i}(\delta_i, b_i, \sigma_i)$;
13. Return $(t_i, \rho_i, \gamma_i)$;
```

Figure 7. The forger $\mathcal{F}$.

First, according to the above simulation, we can get $q_{\mathcal{F}}$ signatures $(s_{\beta_i}, \delta_i, \sigma_i, \gamma_i)$'s retrieved from L where $\mathcal{F}$ computed $s_{\beta_i} = (b_i)^{-1} t_{\beta_i} \bmod n_H$ and $s_{\beta_i}^{e_H} \equiv F_2(\delta_i, \sigma_i, \gamma_i) \pmod{n_H}$ with $1 \leq i \leq q_{\mathcal{F}}$ during the simulation. For the tickets $(s_j, m_j, \delta_j, \sigma_j, \gamma_j)$'s in S_T, we can obtain q_1 signatures $(s_{\alpha_j}, F_1(m_j))$ by retrieving s_{β_j} from L via (δ_j, σ_j) and then computing $s_{\alpha_j} = s_j (s_{\beta_j})^{-1} \bmod n_H$ where $s_{\alpha_j}^{e_H} \equiv F_1^2(m_j) \pmod{n_H}$ with $1 \leq j \leq q_1$. For the tickets $(s_{j'}, m_{j'}, \delta_{j'}, \sigma_{j'}, \gamma_{j'})$'s in $S_{T'}$, we can get q_2 signatures $(\hat{s}_{\alpha_{j'}}, F_1(m_{j'}))$'s and q_2 signatures $(\hat{s}_{\beta_{j'}}, \delta_{j'}, \sigma_{j'}, \gamma_{j'})$'s by the following procedure where $1 \leq j' \leq q_2$. We first randomly select $\hat{b}_{j'} \in \{0,1\}^{l_r}$ and compute $\hat{\beta}_{j'} = \hat{b}_{j'}^{e_H} F_2(\delta_{j'}, \sigma_{j'}, \gamma_{j'}) \bmod n_H$. Then we send $\hat{\beta}_{j'}$ to S_D and obtains $\hat{t}_{\beta_{j'}}$. Finally, we compute $\hat{s}_{\beta_{j'}} = \hat{t}_{\beta_{j'}} (\hat{b}_{j'})^{-1} \bmod n_H$ and $\hat{s}_{\alpha_{j'}} = s_{j'} (\hat{s}_{\beta_{j'}})^{-1} \bmod n_H$ where $\hat{s}_{\alpha_{j'}}^{e_H} \equiv F_1^2(m_{j'}) \pmod{n_H}$ and $\hat{s}_{\beta_{j'}}^{e_H} \equiv F_2(\delta_{j'}, \sigma_{j'}, \gamma_{j'}) \pmod{n_H}$ for each j' with $1 \leq j' \leq q_2$. Consequently, we query S_D $(2q_{\mathcal{F}} + q_2)$ times and obtain $(q_{\mathcal{F}} + q_1 + 2q_2)$ signatures. We succeed in one-more forgery to break the FDH-RSA blind signature scheme.

□

4.4. Tamper Resistance

In our scheme, the information (δ, σ, γ) of a ticket is used for anonymity control, charging, and recording the due date of the ticket, respectively. In this subsection, we will show that none can tamper (δ, σ, γ) of a ticket. First, we introduce a problem called the alternative formulation of RSA Known-Target Inversion (RSA-AKTI) Problem [20] which has been proved being hard by Bellare *et al.*

Definition 4 (RSA AKTI)**.** *Let $k \in \mathbb{N}$ be the security parameter. Let $\mathcal{A}$ be an adversary which can access the RSA-inversion oracle $\mathcal{O}_{inv}$ and the challenge oracle $\mathcal{O}_N$. The challenge oracle $\mathcal{O}_N$ will randomly return $y_i \in \mathbb{Z}^*_{n_H}$ when it is queried. Consider the following experiment:*

Experiment $Exp_{\mathcal{A}}^{RSA-AKTI}(k)$
- $(n_H, e_H, d_H) \leftarrow KeyGen(k)$.
- $(x_1, \ldots, x_m) \leftarrow A^{\mathcal{O}_{inv}, \mathcal{O}_N}(n_H, e_H, k)$ *where m is the number of queries to* $\mathcal{O}_N$.
- Let $y_1, \ldots, y_m$ *be the challenges returned by* $\mathcal{O}_N$.
If the followings are both true, return 1; else return 0.
1. $\forall i \in \{1, \ldots, m\} : x_i^{e_H} \equiv y_i \pmod{n_H}$.
2. $\mathcal{A}$ *made strictly fewer than m queries to* $\mathcal{O}_{inv}$.

In the ticket requesting protocol (Section 3.3) and the ticket using protocol (Section 3.4), if λ tickets are requested, the system side (the judge device) will generate $(\delta_i, \sigma_i, \gamma_i)$ for each ticket where $1 \leq i \leq \lambda$. Here, we define ticket tampering below.

Definition 5 (Ticket Tampering)**.** *There exists an attacker* $\mathcal{A}_{TR}$ *who runs the ticket requesting protocol in Section 3.3 or the ticket using protocol in Section 3.4* λ *times. Let* $S_{\mathcal{A}} = \{(\delta_1, \sigma_1, \gamma_1), \ldots, (\delta_\lambda, \sigma_\lambda, \gamma_\lambda)\}$ *where* $(\delta_i, \sigma_i, \gamma_i)$ *is generated for* $\mathcal{A}_{TR}$ *and thus* δ_i *contains* $\mathcal{A}_{TS}$*'s identity and* σ_i *contains an accumulated value which* $\mathcal{A}_{TR}$ *spent for* $i = 1, \ldots, \lambda$. $\mathcal{A}_{TR}$ *can output a tampered ticket* $(s, m, \delta', \sigma', \gamma')$ *where* $(\delta', \sigma', \gamma') \notin S_{\mathcal{A}}$.

Theorem 3. *The proposed scheme is secure against Ticket Tampering if the RSA-AKTI problem is hard.*

Proof. The model of this proof is shown in Figure 8. There exist a simulator $\mathcal{S}$ and an attacker $\mathcal{A}_{\mathcal{TR}}$ in this model. $\mathcal{S}$ will simulate the environment of our proposed scheme in the random oracle model. $\mathcal{S}$ engages in the proposed scheme to generate the key pair (pk_J, sk_J) of the judge and creates two oracles $\mathcal{O}_{F_1}$ and $\mathcal{O}_{F_2}$. $\mathcal{S}$ can query the oracles $\mathcal{O}_N$ and $\mathcal{O}_{inv}$ defined in Definition 4. $\mathcal{A}_{TR}$ will query $\mathcal{O}_{F_1}$ and $\mathcal{O}_{F_2}$ for the hashed values of the hash functions F_1 and F_2, respectively. There are two lists L_{F_1} and L_{F_2}. L_{F_1} will be used to store (m, τ) where $F_1(m) = \tau^{e_H} \bmod n_H$ and L_{F_2} will be used to record $(\delta, \sigma, \gamma, \pi)$ where $F_2(\delta, \sigma, \gamma) = \pi$. $\mathcal{A}_{TR}$ can query $\mathcal{S}$ at most λ_1 times, $\mathcal{O}_{F_1}$ at most λ_2 times, and $\mathcal{O}_{F_2}$ at most λ_3 times. $\mathcal{S}$, $\mathcal{O}_{F_1}$, and $\mathcal{O}_{F_2}$ are described in Figure 9. Before $\mathcal{A}_{TR}$ queries $\mathcal{S}$, $\mathcal{S}$ initializes the environment by publishing (n_H, e_H, pk_J), setting $i_{guess} = 0$, and guessing a number λ' where $1 \leq \lambda' \leq \lambda_3$. $\mathcal{S}$ guesses that $\mathcal{A}_{TR}$ will successfully output a tampered ticket as $(s, m, \delta, \sigma, \gamma)$ such that 3-tuple (δ, σ, γ) is not produced by $\mathcal{S}$, *i.e.*, $(\delta, \sigma, \gamma) \notin (\delta_i, \sigma_i, \gamma_i)$ for each i with $1 \leq i \leq \lambda_1$, where the value of $F_2(\delta, \sigma, \gamma)$ is obtained from $\mathcal{O}_{F_2}$ at the λ'th query to $\mathcal{O}_{F_2}$.

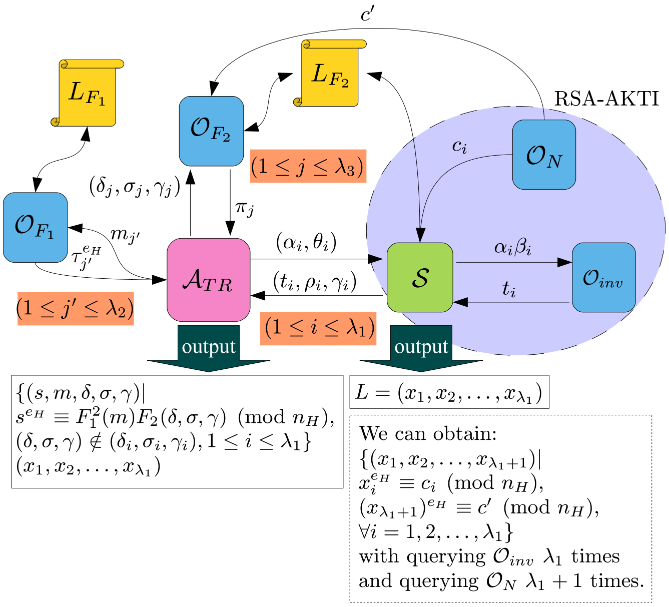

c'
L_{F_1}
L_{F_2}
RSA-AKTI
$\mathcal{O}_{F_2}$
c_i
$\mathcal{O}_N$
$(1 \leq j \leq \lambda_3)$
$(\delta_j, \sigma_j, \gamma_j)$
π_j
$\mathcal{O}_{F_1}$
$m_{j'}$
$\tau_{j'}^{e_H}$
(α_i, θ_i)
$\alpha_i\beta_i$
$\mathcal{A}_{TR}$
(t_i, ρ_i, γ_i)
$\mathcal{S}$
t_i
$\mathcal{O}_{inv}$
$(1 \leq j' \leq \lambda_2)$
$(1 \leq i \leq \lambda_1)$
output
output
$\{(s, m, \delta, \sigma, \gamma)|$
$s^{e_H} \equiv F_1^2(m)F_2(\delta, \sigma, \gamma) \pmod{n_H}$,
$(\delta, \sigma, \gamma) \notin (\delta_i, \sigma_i, \gamma_i), 1 \leq i \leq \lambda_1\}$
$(x_1, x_2, \ldots, x_{\lambda_1})$
$L = (x_1, x_2, \ldots, x_{\lambda_1})$
We can obtain:
$\{(x_1, x_2, \ldots, x_{\lambda_1+1})|$
$x_i^{e_H} \equiv c_i \pmod{n_H}$,
$(x_{\lambda_1+1})^{e_H} \equiv c' \pmod{n_H}$,
$\forall i = 1, 2, \ldots, \lambda_1\}$
with querying $\mathcal{O}_{inv}$ λ_1 times
and querying $\mathcal{O}_N$ $\lambda_1 + 1$ times.

Figure 8. The model of the proof for Tamper Resistance.

```
$\mathcal{S}(\alpha_i, \theta_i)$
Let $(k_i, ID_i) = D_{sk_J}(\theta_i)$;
Select $b_i \in_R \mathbb{Z}^*_{n_H}$;
Set $w_i = 0$ and $\gamma_i$ as the due date;
Do {
    Select $r_{j_i}, r_{z_i} \in_R \{0,1\}^{l_r}$;
    Compute $\sigma_i = E_{pk_J}(w_i, r_{j_i})$;
    Compute $\delta_i = E_{pk_J}(ID_i, F_1(r_{z_i}))$;
} Until $((\delta_i, \sigma_i, \gamma_i)$ is not in $L_{F_2})$
Query $\mathcal{O}_N$ and obtain $c_i \in \mathbb{Z}^*_{n_H}$;
Store $(\delta_i, \sigma_i, \gamma_i, \alpha_i^{-1} c_i \bmod n_H)$ in $L_{F_2}$;
Compute $\beta_i = b_i^{e_H} \alpha_i^{-1} c_i \bmod n_H$;
Query $\mathcal{O}_{inv}$ with $\alpha_i \beta_i \bmod n_H$ and
    then get $t_i = (\alpha_i \beta_i)^{d_H} \bmod n_H$;
Compute $x_i = t_i b_i^{-1} \bmod n_H$;
Store $(x_i, c_i)$ in $L$;
Compute $\rho_i = E_{k_i}(\delta_i, b_i, \sigma_i)$;
Return $(t_i, \rho_i, \gamma_i)$
```

```
$\mathcal{O}_{F_1}(m)$
If $(m$ is in $L_{F_1})$
    Retrieve $\tau$ from $L_{F_1}$;
Else {
    Do {
        Select $\tau \in_R \mathbb{Z}^*_{n_H}$;
    } Until $(\tau$ not in $L_{F_1})$
    Store $(m, \tau)$ in $L_{F_1}$;}
Return $(\tau^{e_H} \bmod n_H)$;
```

```
$\mathcal{O}_{F_2}(\delta, \sigma, \gamma)$
$i_{guess}$++;
If $((\delta, \sigma, \gamma)$ is in $L_{F_2})$
    Retrieve $\pi$ from $L_{F_2}$ by
    $(\delta, \sigma, \gamma)$;
Else {
    If $(i_{guess} == \lambda')${
        Query $\mathcal{O}_N$ and get
    $c' \in \mathbb{Z}^*_{n_H}$;
        Set $\pi = c'$; }
    Else {
        Do {
            Select $\pi \in_R \mathbb{Z}^*_{n_H}$;
        } Until $(\pi$ not in $L_{F_2})$
        Store $(\delta, \sigma, \gamma, \pi)$ in $L_{F_2}$; }
Return $\pi$;
```

Figure 9. The oracles in the proof of Tamper Resistance.

As shown in Figure 9, when $\mathcal{A}_{TR}$ submits a query (α_i, θ_i) to $\mathcal{S}$, $\mathcal{S}$ will send $\alpha_i \beta_i \equiv b_i^{e_H} c_i \pmod{n_H}$ to $\mathcal{O}_{inv}$ and get $x_i \equiv c_i^{d_H} \equiv t_i b_i^{-1} \pmod{n_H}$ where $1 \leq i \leq \lambda_1$. $\mathcal{S}$ stores (x_i, c_i)'s in a list L. If $\mathcal{A}_{TR}$ successfully outputs a tampered ticket $(s, m, \delta, \sigma, \gamma)$ after λ_1 queries to $\mathcal{S}$ with probability at least ϵ_{TR}, we can obtain $x_{\lambda_1+1} \equiv (c')^{d_H} \pmod{n_H}$ with probability at least $\epsilon'_{TR} \geq \frac{\epsilon_{TR}}{\lambda_3}$ by the following procedure.

1. Search L_{F_1} by m and get entry (m, τ').
2. Search L_{F_1} by $(\tau')^{e_H} \bmod n_H$ and get entry $((\tau')^{e_H} \bmod n_H, \tau)$ where $F_1^2(m) = \tau^{e_H} \bmod n_H$, *i.e.*, $\tau = (F_1^2(m))^{d_H} \bmod n_H$.
3. Compute $x_{\lambda_1+1} = s\tau^{-1} \bmod n_H$ and thus $x_{\lambda_1+1} \equiv (c')^{d_H} \pmod{n_H}$.

Consequently, $\mathcal{S}$ queries $\mathcal{O}_N$ $(\lambda_1 + 1)$ times, and $\mathcal{O}_{inv}$ λ_1 times, and then we can obtain $(x_1, \ldots, x_{\lambda_1})$ from L and x_{λ_1+1} such that $x_i^{e_H} \equiv c_i \pmod{n_H}$ and $x_{\lambda_1+1}^{e_H} \equiv c' \pmod{n_H}$ where $1 \leq i \leq \lambda_1$. We successfully solve the RSA-AKTI problem with non-negligible probability at least ϵ'_{TR}. □

4.5. Ticket Swindling Resistance

In our scheme, a mobile user has to show $T = (F_1(m^*), \delta^*, \sigma^*, \gamma, s^*)$ for authentication. In this subsection, we will prove that none can successfully pass authentication via an eavesdropped T. We call this Ticket Swindling Resistance. In order to prove this, we first introduce the communication model and some definitions as follows.

The Communication Model. We briefly describe the communication model [21,22] of our distributed environment. Oracle $\Pi^u_{MS_i,V_j}$ models that a mobile user MS_i performs the anonymous authentication protocol of Section 3.4 with the entity V_j in the uth session of MS_i. Oracle $\Pi^v_{V_j,MS_i}$ models that a system entity V_j performs the protocol with the mobile user MS_i in the vth session of V_j. An adversary E is a probabilistic polynomial-time Turing machine that is allowed to make the following queries.

- $Execute(\Pi^u_{MS_i,V_j}, \Pi^v_{V_j,MS_i})$: This query models all kinds of passive attacks. MS_i and V_j will carry out the protocol of Section 3.4 and the adversary E can eavesdrop all messages transmitted between MS_i and V_j.

- $Send(\Pi^u_{MS_i,V_j}, \mathcal{M})$ or $Send(\Pi^v_{V_j,MS_i}, \mathcal{M})$: This query models all kinds of active attacks. The adversary E can send any message $\mathcal{M}$ to $\Pi^u_{MS_i,V_j}$ or $\Pi^v_{V_j,MS_i}$ which will give responses to E according to the protocol of Section 3.4. E can make the query $Send(\Pi^u_{MS_i,V_j}, \mathcal{N})$ to get a response of the first flow where $\mathcal{N}$ is an empty string.
- $Reveal(\Pi^u_{MS_i,V_j})$ or $Reveal(\Pi^v_{V_j,MS_i})$: This query allows the adversary to get the session key of $\Pi^u_{MS_i,V_j}$ or $\Pi^v_{V_j,MS_i}$ after $\Pi^u_{MS_i,V_j}$ and $\Pi^v_{V_j,MS_i}$ have successfully finished mutual authentication and established a common session key.
- $Reveal(T)$: This query allows the adversary to obtain the secret value m if T has been successfully consumed for authentication where $T = (F_1(m), \delta, \sigma, \gamma, s)$.
- $Corrupt(V_j)$: This query reveals V_j's long-term key sk_{V_j}.

In our protocol, once a mobile user consumes her/his T for authentication, T will be kept in the system's database for double-using checking. Hence, a successfully-used T cannot be consumed again by any eavesdropper. Any attacker can just try to swindle an eavesdropped T which has not been successfully used, *i.e.*, the attacker has to interfere the authentication process after she/he obtains T in the first flow of the authentication protocol in Section 3.4. We define Ticket Swindling below.

Definition 6 (The Ticket Swindling Game). *Let $k \in \mathbb{N}$ be the security parameter. $\mathcal{A}_{TS}$ is a polynomial time adversary who tries to swindle an eavesdropped T. Consider the following experiment:*

Experiment $Exp^{TS}_{\mathcal{A}_{TS}}(k)$
- $(n_H, e_H, F_1, F_2, F_3, pk_J, pk_V) \leftarrow Setup(k)$
- $\mathcal{A}^{Execute,Send,Reveal,Corrupt}_{TS}(n_H, e_H, F_1, F_2, F_3, pk_J, pk_V)$
If the followings are true return 1 else return 0
1. $\mathcal{A}_{TS}$ *makes* $Send(\Pi^v_{V_j,MS_i}, (\alpha, \theta_1, T, \theta_2))$ *and* $Send(\Pi^v_{V_j,MS_i}, r_5)$ *queries and then* $\Pi^v_{V_j,MS_i}$ *accepts.*
2. T *is outputted from* $\Pi^u_{MS_i,V_j}$
3. r_5 *has never been outputted by* $\Pi^u_{MS_i,V_j}$
4. $\mathcal{A}_{TS}$ *has never made* $Reveal(T)$ *and* $Corrupt(V_j)$ *queries*

The advantage of $\mathcal{A}_{TS}$ is $Adv^{TS}_{\mathcal{A}_{TS}}(k) = Pr[Exp^{TS}_{\mathcal{A}_{TS}}(k) = 1]$. We say that our scheme satisfies Ticket Swindling Resistance if $Adv^{TS}_{\mathcal{A}_{TS}}(k)$ is negligible.

Besides, we define the following Indistinguishability Game under the Chosen-Ciphertext Attack (IND-CCA) based on [23].

Definition 7 (IND-CCA). *Let $k \in \mathbb{N}$ be the security parameter. $\mathcal{C}$ is a challenger and $\mathcal{F}$ is a polynomial time adversary. $\mathcal{P}$ is an asymmetric cryptosystem with semantic security where the public-private key pair is (pk, sk). There are two oracles $\mathcal{O}_E$ and $\mathcal{O}_D$. $\mathcal{F}$ can query $\mathcal{O}_E$ to encrypt a plaintext by pk and query $\mathcal{O}_D$ to decrypt a ciphertext by sk. Consider the following experiment:*

Experiment $Exp^{IND-CCA}_{\mathcal{F}}(k)$
- $(pk, sk) \leftarrow Setup(k)$
- $(M_0, M_1) \leftarrow \mathcal{F}^{\mathcal{O}_E,\mathcal{O}_D}$
- $E_{pk}(M_b) \overset{b \in_R \{0,1\}}{\longleftarrow} \mathcal{C}(M_0, M_1)$
- $b' \leftarrow \mathcal{F}^{\mathcal{O}_E,\mathcal{O}_D}(E_{pk}(M_b))$
If the followings are both true return 1 else return 0
1. $\mathcal{F}$ *never submits the query* $E_{pk}(M_b)$ *to* $\mathcal{O}_D$
2. $b' = b$

We define the advantage of $\mathcal{F}$ is $Adv^{IND-CCA}_{\mathcal{F}}(k) = |Pr[Exp^{IND-CCA}_{\mathcal{F}}(k) = 1] - \frac{1}{2}|$.

We also introduce the RSA Single-Target Inversion Problem (RSA-STI) [20] as follows.

Definition 8 (RSA-STI). *Let $k \in \mathbb{N}$ be the security parameter. Let $\mathcal{A}$ be a polynomial time adversary. Consider the following experiment:*

$$\begin{array}{l} \textit{Experiment } Exp_{\mathcal{A}}^{RSA-STI}(k) \\ \text{- } (N, e, d) \overset{R}{\leftarrow} KeyGen(k) \\ \text{- } y \overset{R}{\leftarrow} \mathbb{Z}_N^*; x \leftarrow \mathcal{A}(N, e, k, y) \\ \textit{If } x^e \equiv y \pmod N \textit{ return 1 else return 0} \end{array}$$

We define the advantage of $\mathcal{A}$ as $Adv_{\mathcal{A}}^{RSA-STI}(k) = Pr[Exp_{\mathcal{A}}^{RSA-STI}(k) = 1]$.

Theorem 4. *The proposed anonymous authentication protocol satisfies Ticket Swindling Resistance.*

Proof. The proof model is illustrated in Figure 10. A simulator $\mathcal{S}_{TS}$ will simulate the communication environment and help us to solve the RSA-STI problem. First, $\mathcal{S}_{TS}$ obtains the parameters (N, e, y) from the RSA-STI problem. $\mathcal{S}_{TS}$ initializes the system parameters, which are the public-private key pairs of H, V's, and the judge, and constructs the oracles $\Pi^u_{MS_i,V_j}$ and $\Pi^v_{V_j,MS_i}$. $\mathcal{S}_{TS}$ also controls three hash oracles $\mathcal{O}_{F_1}$, $\mathcal{O}_{F_2}$, and $\mathcal{O}_{F_3}$ to simulate the hash functions F_1, F_2, and F_3, respectively. When $\mathcal{O}_{F_1}$ is queried with m, it will return r_{f_1} retrieved from L_{F_1} via m if m exists in L_{F_1} or return $r_{f_1} = (m^e \bmod N)$ and records (m, r_{f_1}) in L_{F_1}. If $\mathcal{O}_{F_2}$ is queried with (δ, σ, γ), it will return r_{f_2} retrieved from L_{F_2} via (δ, σ, γ) if (δ, σ, γ) exists in L_{F_2} or return a randomly-selected string $r_{f_2} \in \{0,1\}^{l_r}$ and record $(\delta, \sigma, \gamma, r_{f_2})$ in L_{F_2}. When $\mathcal{O}_{F_3}$ is queried with $(r_2||r_4)$, it will return r_{f_3} retrieved from L_{F_3} if $(r_2||r_4)$ exists in L_{F_3} or a randomly-chosen string $r_{f_3} \in \{0,1\}^{l_r}$ and record $((r_2||r_4), r_{f_3})$ in L_{F_3}. There is also an oracle $\mathcal{O}_T$ which plays the role of H in the protocol of Section 3.3 (or Section 3.4) to issue tickets. $\mathcal{O}_T$ will return (t, ρ, γ) when it is queried with (α, θ).

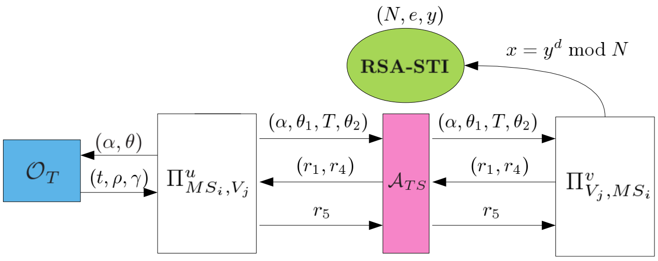


Figure 10. The model of the proof of Ticket Swindling Resistance.

Assume that an attacker $\mathcal{A}_{TS}$ performs at most q_1 times of $Execute(\Pi^u_{MS_i,V_j}, \Pi^v_{V_j,MS_i})$ queries, q_2 times of $Send(\Pi^u_{MS_i,V_j}, \mathcal{N})$ queries, and q_3 times of $Send(\Pi^v_{V_j,MS_i}, \mathcal{M})$ queries. $\mathcal{A}_{TS}$ can also submit a $Reveal(T)$ query to get the secret m of T where T must have been consumed for authentication.

$\mathcal{S}_{TS}$ initializes four global parameters which are $i_{guess} = 0$, $s_{guess} = 0$, $T_{guess} = \mathcal{N}$, and $r_{guess} = \mathcal{N}$. Then $\mathcal{S}_{TS}$ guesses that the attacker $\mathcal{A}_{TS}$ will swindle T which is returned from the λth $Send(\Pi^u_{MS_i,V_j}, \mathcal{N})$ query. When $Execute(\Pi^u_{MS_i,V_j}, \Pi^v_{V_j,MS_i})$ is queried, the oracle $\Pi^u_{MS_i,V_j}$ will run the protocol of Section 3.3 (or Section 3.4) with $\mathcal{O}_T$ to get a ticket $(m, \delta, \sigma, \gamma, s)$ and prepares $T = (F_1(m), \delta, \sigma, \gamma, s)$. Then it takes T to perform the protocol of Section 3.4 with $\Pi^v_{V_j,MS_i}$ under the presence of $\mathcal{A}_{TS}$. When $Send(\Pi^u_{MS_i,V_j}, \mathcal{M})$ and $Send(\Pi^v_{V_j,MS_i}, \mathcal{M})$ are queried, $\Pi^u_{MS_i,V_j}$ and $\Pi^v_{V_j,MS_i}$ will act according to Figure 11. There are five lists in Figure 11 where $L_{T_{MS}}$ and L_{T_V} are used to record the transcript of $\Pi^u_{MS_i,V_j}$ and $\Pi^v_{V_j,MS_i}$, $L_{K_{MS}}$ and L_{K_V} store the session keys of $\Pi^u_{MS_i,V_j}$ and $\Pi^v_{V_j,MS_i}$, and L_{usedT} records all used T's.

```
$\Pi^u_{MS_i,V_j}$ : $Send(\Pi^u_{MS_i,V_j}, \mathcal{M})$
1. If $(\mathcal{M} == \mathcal{N})$ {
2.   $i_{guess}$++;
3.   Select $m, k' \in_R \{0,1\}^{l_r}$;
4.   Select $r' \in_R \mathbb{Z}^*_{n_H}$;
5.   Set $m' = F_1(m)$;
6.   If $(i_{guess} == \lambda)$ {
7.     Reset $m' = y$;
8.     Set $s_{guess} = u$; }
9.   Compute $\alpha' = (r')^{e_H} F_1(m') \bmod n_H$;
10.  Compute $\theta' = E_{pk_J}(k', ID_{MS_i})$;
11.  Send $(\alpha', \theta')$ to $\mathcal{O}_T$ and get $(t, \rho, \gamma)$;
12.  Let $(\delta, b, \sigma) = D_{k'}(\rho)$;
13.  Compute $s = (br')^{-1} t \bmod n_H$;
14.  Prepare $T = (m', \delta, \sigma, \gamma, s)$;
15.  Select $m^{**}, k, r_1, r_2, r_3 \in_R \{0,1\}^{l_r}$;
16.  Select $r \in_R \mathbb{Z}^*_{n_H}$;
17.  Compute $\alpha = r^{e_H} F_1^2(m^{**}) \bmod n_H$;
18.  Compute $\theta_1 = E_{pk_J}(k)$;
19.  Compute $\theta_2 = E_{pk_V}(r_1, r_2, r_3)$;
20.  Store $(u, r_1, r_2, r_3, m)$ in $L_{T_{MS}}$;
21.  If $(i_{guess} == \lambda)$ { Let $T_{guess} = T$; }
22.  Output $(\alpha, \theta_1, T, \theta_2)$; } //the first flow
23. Else {
24.  Parse $\mathcal{M}$ as $(r_1', r_4)$; //the second flow
25.  Retrieve $(r_1, r_2, r_3, m)$ from $L_{T_{MS}}$
     via $u$;
26.  If $(r_1 == r_1')$ {
27.    If $(u == s_{guess})$ {
28.      Select $r_5 \in_R \{0,1\}^{l_r}$;
29.      Let $r_{guess} = r_5$;
30.      Store $((r_2||r_4), \perp)$ in $L_{F_3}$; }
31.    eles {
32.      Submit $(r_2||r_4)$ to $\mathcal{O}_{F_3}$ and
         get $r_{f_3}$;
33.      Let $r_5 = r_{f_3} \oplus m$; }
34.    Store $(r_3, u)$ in $L_{K_{MS}}$;
35.  Else {
36.    Let $r_5$ = "auth-failure"; }
37.  Output $r_5$; } //the third flow
```

```
$\Pi^v_{V_j,MS_i}$ : $Send(\Pi^v_{V_j,MS_i}, \mathcal{M})$
1. If ($\mathcal{M}$ can be parsed as $(\alpha, \theta_1, T, \theta_2)$) {
2.   If ($T$ not in $L_{usedT}$) {
3.     Let $(r_1, r_2, r_3) = D_{sk_V}(\theta_2)$;
4.     Do {
5.       Select $r_4, r_{f_3} \in_R \{0,1\}^{l_r}$;
6.     } Until ($(r_2||r_4)$ and $r_{f_3}$ not in $L_{F_3}$)
7.     Store $((r_2||r_4), r_{f_3})$ in $L_{F_3}$;
8.     Store $(v, r_2, r_3, r_4, T)$ in $L_{T_V}$;
9.     Output $(r_1, r_4)$; } //the second flow
10.  Else
11.    Output "double-using"; }}
12. Else {
13.  Parse $\mathcal{M}$ as $r_5$;
14.  Retrieve $(r_2, r_3, r_4, T)$ from
     $L_{T_V}$ via $v$;
15.  Parse $T$ as $(h_{m'}, \delta, \sigma, \gamma, s)$;
16.  Send $(r_2||r_4)$ to $\mathcal{O}_{F_3}$ and get $r_{f_3}$;
17.  If $(r_{f_3} \neq \perp)$ {
18.    Compute $m = r_5 \oplus r_{f_3}$;
19.    Submit $m$ to $\mathcal{O}_{F_1}$ and get $h_m$;
20.    If $(h_m == h_{m'})$ {
21.      Store $(r_3, v)$ in $L_{K_V}$;
22.      Store $(m, T)$ in $L_{usedT}$;
23.      Output "auth-success"; }
24.    Else {
25.      Output "auth-failure"; }}
26.  Else {
27.    If $(r_5 == r_{guess})$ {
28.      Store $(\perp, T)$;
29.      Output "auth-success"; }
30.    Else {
31.      Output "auth-failure"; }}
32. }
```

Figure 11. The oracle $\Pi^u_{MS_i,V_j}$ in the proof of Ticket Swindling Resistance.

In Figure 11, s_{guess} denotes the session u of $\Pi^u_{MS_i,V_j}$. $\Pi^u_{MS_i,V_j}$ checks if the current session u is corresponding to the λth $Send(\Pi^u_{MS_i,V_j}, \mathcal{N})$ query (line 27 to line 30). If true, it randomly selects $r_5 \in \{0,1\}^{l_r}$. Here, $\Pi^u_{MS_i,V_j}$ does not know x and sets $F_3(r_2||r_4) = \perp$. The simulation will fail if $\mathcal{A}_{TS}$ sends a query $(r_2||r_4)$ to $\mathcal{O}_{F_3}$. However, we will show that the probability of the above failure is negligible in Appendix.

In Figure 11, $\Pi^v_{V_j,MS_i}$ checks if r_{f_3} is equal to $\perp$. If true, this means that the current session v matches the λth $Send(\Pi^u_{MS_i,V_j}, \mathcal{N})$ query.

After finishing the simulation, $\mathcal{S}_{TS}$ can retrieve (m, T) from L_{usedT} via T_{guess}. If $m \neq \perp$, $\mathcal{S}_{TS}$ has that $T_{guess} = T = (m' = y, \delta, \sigma, \gamma, s)$ where $y = F_1(m) = m^e \bmod N$. Thus, $\mathcal{S}_{TS}$ solves the RSA-STI problem. Therefore, $\mathcal{A}_{TS}$, with non-negligible probability at least ϵ_{TS}, can consume an eavesdropped T to successfully perform the anonymous authentication protocol of Section 3.4 with $\Pi^v_{V_j,MS_i}$, $\mathcal{S}_{TS}$ can solve the RSA-STI problem with non-negligible advantage at least $\frac{\epsilon_{TS}}{q_2}$.

□

4.6. Secure Mutual Authentication

In order to prove the security of mutual authentication in the proposed scheme, we first introduce *Matching Conversations* and *No Matching*$^E(k)$ [21,22] as follows.

Definition 9 (Matching Conversations). *Fix a number of flows $R = 2\rho - 1$ and an R-flow protocol $P = (\Pi, \mathcal{G})$ where Π specifies how players behave and $\mathcal{G}$ generates key pairs for each entity. Run P in the presence of an adversary E and consider two oracles $\Pi^u_{MS_i,V_j}$ and $\Pi^v_{V_j,MS_i}$, that engage in conversations K and K′ respectively.*

1. K' is a matching conversation to K if there exist $\tau_0 < \tau_1 < \cdots < \tau_{R-1}$ such that K is prefixed by

$$(\tau_0, \mathcal{N}, \alpha_1), (\tau_2, \beta_1, \alpha_2), \ldots, (\tau_{2\rho-2}, \beta_{\rho-1}, \alpha_\rho)$$

and K' is prefixed by

$$(\tau_1, \alpha_1, \beta_1), (\tau_3, \alpha_2, \beta_2), \ldots, (\tau_{2\rho-3}, \alpha_{\rho-1}, \beta_{\rho-1})$$

2. K is a *matching conversation* to K' if there exist $\tau_0 < \tau_1 < \cdots < \tau_R$ such that K' is prefixed by

$$(\tau_1, \alpha_1, \beta_1), (\tau_3, \alpha_2, \beta_2), \ldots, (\tau_{2\rho-3}, \alpha_{\rho-1}, \beta_{\rho-1}), (\tau_{2\rho-1}, \alpha_\rho, *)$$

and K is prefixed by

$$(\tau_0, \mathcal{N}, \alpha_1), (\tau_2, \beta_1, \alpha_2), \ldots, (\tau_{2\rho-2}, \beta_{\rho-1}, \alpha_\rho)$$

Finally, $\Pi^u_{MS_i,V_j}$ and $\Pi^v_{V_j,MS_i}$ are said to have had *matching conversations* if K is a matching conversation to K' and K' is a matching conversation to K.

Definition 10 (No Matching$^E(k)$)**.** *Let $k \in \mathbb{N}$ be the security parameter. No Matching$^E(k)$ is that when protocol P is run against an adversary E, there exists an oracle $\Pi^u_{MS_i,V_j}$ with $MS_i, V_j \notin S_C$ (where S_C denotes the set of entities corrupted by E) which accepted but there is no oracle $\Pi^v_{V_j,MS_i}$ which has had a matching conversation to $\Pi^u_{MS_i,V_j}$, or vice versa.*

Definition 11. *A protocol P is a secure mutual authentication protocol if for every polynomial-time adversary E:*

1. *If $\Pi^u_{MS_i,V_j}$ and $\Pi^v_{V_j,MS_i}$ have matching conversations, then both oracles accept;*
2. *The probability of No Matching$^E(k)$ is negligible.*

Theorem 5. *The protocol of Section 3.4 is a secure mutual authentication protocol.*

Proof. Our authentication protocol satisfies the first condition of Definition 11, if the the adversary acts as a wire. Hence, we concentrate on the proof for the second condition.

When we carry out the experiment of the communication model against E, E may succeed in the following two cases. Case 1 is that there exists an oracle Π^u_{MS,V_j} which accepted, where $MS_i, V_j \notin S_C$ and S_C is the set of corrupted entities, but there is no oracle $\Pi^v_{V_j,MS_i}$ has a matching conversation to $\Pi^u_{MS_i,V_j}$. Case 2 is that there exists an oracle $\Pi^v_{V_j,MS_i}$ which accepted but there is no oracle Π^u_{MS,V_j} has a matching conversation to $\Pi^v_{V_j,MS_i}$. Suppose that E has probability ϵ_1 in Case 1 and ϵ_2 in Case 2. Thus, we conclude that if *No Mathing*$^E(k)$ is non-negligible, ϵ_1 or ϵ_2 must be non-negligible.

In Case 1, E has to make $Send(\Pi^u_{MS_i,V_j}, \mathcal{N})$ query at some time τ_0 and make $Send(\Pi^u_{MS_i,V_j}, (r_1, r_4))$ query at some time $\tau_2 > \tau_0$. If (r_1, r_4) are valid, the state of $\Pi^u_{MS_i,V_j}$ will be changed as "accepted". The proof model of this case is depicted in Figure 12. In the proof model, we will construct a simulator $\mathcal{S}_{MA}$ who will simulate the communication environment to E and try to break the IND-CCA defined in Definition 7. Assume that there are q_1 entities MS_i's and q_2 entities V_j's in the communication environment and E will perform $Send(\Pi^u_{MS_i,V_j}, \mathcal{N})$ at most q_3 times with $i \in \{1, \ldots, q_1\}$ and $j \in \{1, \ldots, q_2\}$ where q_1, q_2, and q_3 are polynomials of security parameter k. There also exists an oracle $\mathcal{O}_T$ who will play the role of H to run the protocol of Section 3.3 to issue tickets.

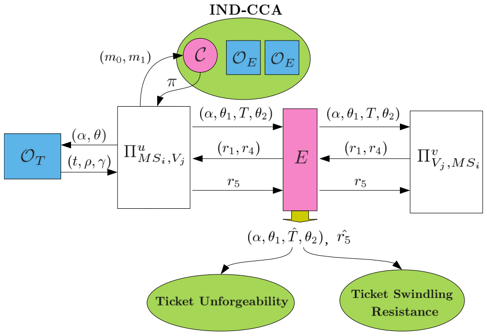


Figure 12. The proof model of Case 1.

In order to set up the communication environment, $\mathcal{S}_{MA}$ first randomly selects four strings, $(\tilde{r}_1, \tilde{r}'_1, \tilde{r}_2, \tilde{r}_3)$, and sets $m_0 = (\tilde{r}_1, \tilde{r}_2, \tilde{r}_3)$ and $m_1 = (\tilde{r}'_1, \tilde{r}_2, \tilde{r}_3)$. Then, $\mathcal{S}_{MA}$ sends (m_0, m_1) to $\mathcal{C}$ and gets $\pi = E_{pk}(m_b)$ from $\mathcal{C}$ where $b \in_R \{0, 1\}$ and pk is the public key in the IND-CCA game. S_R denotes the set of the outputs of $Send(\Pi^v_{V_j,MS_{i'}}, (\alpha, \theta_1, T, \theta_2))$ queries, *i.e.*, the second flow (r_1, r_4)'s. $\mathcal{S}_{MA}$ randomly chooses two integers λ and j' and guesses that $\Pi^u_{MS_i,V_{j'}}$ will accept after E makes $Send(\Pi^u_{MS_i,V_{j'}}, (r_1, r_4))$ query where $(r_1, r_4) \notin S_R$ and the session u was started by the λth $Send(\Pi^u_{MS_i,V_{j'}}, \mathcal{N})$ query. Then, $\mathcal{S}_{MA}$ sets up the public/private keys (pk_{V_j}, sk_{V_j}) for entity V_j, where $j = \{1, \ldots, q_2\}$ and $j \neq j'$, and assigns the public key pk to entity $V_{j'}$ and generates public/private keys (e_H, n_H, d_H) and (pk_J, sk_J) for H and the judge.

When E makes $Execute(\Pi^u_{MS_i,V_j}, \Pi^v_{V_j,MS_i})$ query, $\Pi^u_{MS_i,V_j}$ will perform the protocol of Section 3.3 with $\mathcal{O}_T$ and get (t, ρ, γ). Then, $\Pi^u_{MS_i,V_j}$ prepares $T = (F_1(m), \delta, \sigma, \gamma, s)$ and runs the protocol of Section 3.4 with $\Pi^v_{V_j,MS_i}$ in the presence of E. If $V_{j'}$ is involved in the execution query, S_{MA} can simulate it by running the protocol of Section 3.4 with querying $\mathcal{O}_E$ for encryption and querying $\mathcal{O}_D$ for decryption. Besides, $\mathcal{S}_{MA}$ initializes two global parameters $i_{guess} = 0$ and $f_{guess} = 0$ and empties four lists $L_{T_{MS}}$, $L_{K_{MS}}$, L_{T_V}, and L_{K_V}. When E makes $Send(\Pi^u_{MS_i,V_{j'}}, \mathcal{M})$ and $Send(\Pi^v_{V_j,MS_i}, \mathcal{M})$ queries, the actions of $\Pi^u_{MS_i,V_j}$ and $\Pi^v_{V_j,MS_i}$ are defined in Figure 13.

```
$\Pi^u_{MS_i,V_j}$ : $Send(\Pi^u_{MS_i,V_j}, \mathcal{M})$
$i_{guess}$++;
If ($\mathcal{M} == \mathcal{N}$) {
    Perform the protocol of Section 3.3
    with $\mathcal{O}_T$ and get $(m, \delta, \sigma, \gamma, s)$;
    Prepare $T = (F_1(m), \delta, \sigma, \gamma, s)$;
    Select $m^{**}, k, r_1, r_2, r_3 \in_R \{0,1\}^{l_r}$;
    Seelct $r \in_R \mathbb{Z}^*_{n_H}$;
    Compute $\alpha = r^{e_H} F_1^2(m^{**}) \bmod n_H$;
    Compute $\theta_1 = E_{pk_J}(k)$;
    If ($V_j == V_{j'}$) {
        $\theta_2 = E_{pk}(r_1, r_2, r_3)$; }
    Else {
        $\theta_2 = E_{pk_{V_j}}(r_1, r_2, r_3)$; }
    If ($i_{guess} == \lambda$) { Reset $\theta_2 = \pi$; }
    Output $(\alpha, \theta_1, T, \theta_2)$;
    If ($i_{guess} == \lambda$) {
        Store $(MS_i, u, m, \perp, r_2, r_3)$ in $L_{T_{MS}}$;}
    Else {
        Store $(MS_i, u, m, r_1, r_2, r_3)$ in $L_{T_{MS}}$;}}
Else {
    Parse $\mathcal{M}$ as $(r'_1, r'_4)$;
    Retrieve $(m, r_1, r_2, r_3)$ from $L_{T_{MS}}$
    via $(MS_i, u)$;
    If ($V_j == V_{j'}$ and $r_1 == \perp$) {
        Try to break IND-CCA;}
    Else {
        If ($r_1 == r'_1$){
            Store $(u, r_3)$ in $L_{K_{MS}}$;
            Compute $r_5 = F_3(r_2||r'_4) \oplus m$;
            Output $r_5$;}
        Else {Output "auth-failure";}}
```

```
$\Pi^v_{V_j,MS_i}$ : $Send(\Pi^v_{V_j,MS_i}, \mathcal{M})$
If ($\mathcal{M}$ can be parsed as $(\alpha, \theta_1, T, \theta_2)$){
    Parse $T$ as $(h_m, \delta, \sigma, \gamma, s)$;
    Select $r_4 \in_R \{0,1\}^{l_r}$;
    If ($V_j == V_{j'}$){
        If ($\theta_2 == \pi$) {
            Set $f_{guess} = 1$;
            Terminate; }
        Else {
            Send $\theta_2$ to $\mathcal{O}_D$ and
            get $(r_1, r_2, r_3)$; }
    Else{
        $(r_1, r_2, r_3) = D_{sk_{V_j}}(\theta_2)$;}
    Store $(V_j, v, r_2, r_3, r_4, h_m)$ in $L_{T_V}$;
    Output $(r_1, r_4)$};
Else{
    Set $r_5 = \mathcal{M}$;
    Retrieve $(r_2, r_3, r_4, h_m)$ from $L_{T_V}$
    via $(V_j, v)$;
    Compute $m = r_5 \oplus (r_2||r_4)$;
    If ($F_1(m) == h_m$) {
        Output "auth-success";
        Store $(v, r_3)$ in $L_{K_V}$; }
    Else {
        Output "auth-failure"; }
```

Figure 13. The actions of $\Pi^u_{MS_i,V_j}$ and $\Pi^v_{V_j,MS_i}$ for $Send(\Pi^u_{MS_i,V_j}, \mathcal{M})$ and $Send(\Pi^v_{V_j,MS_i}, \mathcal{M})$ queries, respectively.

In Line 22 of $\Pi^u_{MS_i,V_j}$ in Figure 13, $\mathcal{S}_{MA}$ tries to break IND-CCA as follows. If $f_{guess} = 0$, $\mathcal{S}_{MA}$ will guess $b' = 0$ when $r'_1 = \tilde{r}_1$, guess $b' = 1$ when $r'_1 = \tilde{r}'_1$, and randomly guess $b' \in \{0,1\}$ when $r'_1 \neq \tilde{r}_1$ and $r'_1 \neq \tilde{r}'_1$. If $f_{guess} = 1$, $\mathcal{S}_{MA}$ will randomly guess $b' \in \{0,1\}$. If ϵ_1 is non-negligible, $\mathcal{S}_{MA}$ has non-negligible advantage $Adv^{IND-CCA}_{\mathcal{S}_{MA}}(k)$ to output a guess bit b' such that $b' = b$ where $Adv^{IND-CCA}_{\mathcal{S}_{MA}}(k) = \epsilon'_1 - \frac{1}{2}$ and

$$\begin{aligned}
\epsilon'_1 &= \frac{\epsilon_1 + (1-\epsilon_1)\frac{1}{2^{l_r}}}{q_2 q_3} + \frac{(1-\epsilon_1)\frac{2^{l_r}-1}{2^{l_r}}}{2q_2 q_3} + \frac{(q_2 q_3 - 1)}{2q_2 q_3} \\
&> \frac{2\epsilon_1}{2q_2 q_3} + \frac{(1-\epsilon_1)\frac{1}{2^{l_r}} + (1-\epsilon_1)\frac{2^{l_r}-1}{2^{l_r}}}{2q_2 q_3} + \frac{(q_2 q_3 - 1)}{2q_2 q_3} \\
&= \frac{\epsilon_1}{2q_2 q_3} + \frac{1}{2}.
\end{aligned}$$

In Case 2, E has to send a valid 4-tuple $(\alpha, \theta_1, \hat{T}, \theta_2)$ to $\Pi^v_{V_j,MS_i}$ first and then respond a valid string $\hat{r}_5$ after receiving $(\hat{r}_1, \hat{r}_4)$ from $\Pi^v_{V_j,MS_i}$. Let S_T be the set of T's obtained by $\Pi^u_{MS_i,V_j}$. The followings are two sub-cases when E successfully impersonates $\Pi^u_{MS_i,V_j}$.

1. $\hat{T} \notin S_T$: Assume that $\hat{T} = (F_1(\hat{m}), \hat{\delta}, \hat{\sigma}, \hat{\gamma}, \hat{s})$. Thus, $\mathcal{S}_{MA}$ can obtain $\hat{m} = \hat{r}_5 \oplus F_3(r_2||r_4)$ and forge a ticket $(\hat{m}, \hat{\delta}, \hat{\sigma}, \hat{\gamma}, \hat{s})$. However, we have proved the security of ticket unforgeability in Section 4.3. Hence, the probability of that E is successful in this sub-case is negligible.
2. $\hat{T} \in S_T$: In the sub-case, E successfully swindles $\hat{T}$ which is owned by $\Pi^u_{MS_i,V_j}$. We have proved the security of ticket swindling resistance in Section 4.5. Consequently, the probability of that E is successful in this sub-case is also negligible.

Therefore, the probability ϵ_2 is negligible. We conclude that $No\ Matching^E(k)$ is negligible because ϵ_1 and ϵ_2 are both negligible. □

4.7. Secure Authenticated Key Exchange

First, we introduce a new query, $Test(\Pi^u_{MS_i,V_j})$. An adversary E can ask $Test(\Pi^u_{MS_i,V_j})$ query after $\Pi^u_{MS_i,V_j}$ has established a session key r_3 with another oracle $\Pi^v_{V_j,MS_i}$. To answer this query, the oracle flips a fair coin $b' \leftarrow \{0,1\}$ and then returns $r_k = r_3$ if $b' = 0$ and $r_k \in_R \{0,1\}^{l_r}$ if $b' = 1$. In the following, we define an experiment which was also introduced in [15,21].

Definition 12. *Let $k \in \mathbb{N}$ be a security parameter. In this experiment, the adversary E will try to guess that the returned value r_k from $Test(\Pi^u_{MS_i,V_j})$ query is a random string or the real session key.*

Experiment $Exp^{GoodGuess}_E(k)$
- $b'' \leftarrow E^{Execute,Send,Reveal,Corrupt}(r_k = Test(\Pi^u_{MS_i,V_j}))$
If the followings are true, return 1; else return 0.
1. $b' = b''$
2. E has never submitted $Reveal(\Pi^u_{MS_i,V_j})$ and
$Corrupt(V_j)$ queries.

We define the advantage of E is $Adv^{GoodGuess}_E(k) = Pr[Exp^{GoodGuess}_E(k) = 1] - \frac{1}{2}$.

Definition 13. *A protocol $P = (\Pi, \mathcal{G})$ is a secure authenticated key exchange protocol if*

1. *P is a secure mutual authentication protocol;*
2. *Both oracles $\Pi^u_{MS_i,V_j}$ and $\Pi^v_{V_j,MS_i}$ always accept and hold the same session key r_3 if E is a benign adversary; and*
3. *For any adversary E, $Adv^{GoodGuess}_E(k)$ is negligible.*

Theorem 6. *The authentication protocol of Section 3.4 is a secure authenticated key exchange protocol if the encryption E_{pk_V} is semantic secure.*

Proof. We have shown that the proposed scheme satisfies the first and second conditions of Definition 13. We consider the third condition and assume that E is an adversary who has probability $(\epsilon + \frac{1}{2})$ in outputting b'' such that $b' = b''$ where ϵ is non-negligible. The proof model is depicted as Figure 14.

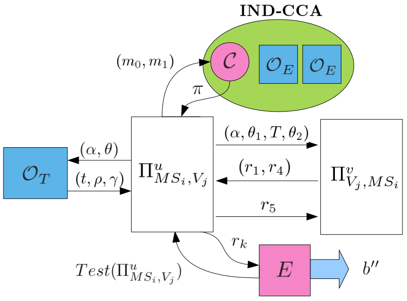


Figure 14. The proof model of Theorem 6.

$\mathcal{S}$ is a simulator who will simulate the communication environment for E. $\mathcal{S}$ first randomly selects four strings, $(\tilde{r}_1, \tilde{r}_2, \tilde{r}_3, \tilde{r}'_3)$, and prepares $m_0 = (\tilde{r}_1, \tilde{r}_2, \tilde{r}_3)$ and $m_1 = (\tilde{r}, \tilde{r}_2, \tilde{r}'_3)$. $\mathcal{S}$ then sends (m_0, m_1) to $\mathcal{C}$ and obtains $\pi = E_{pk}(m_b)$ from $\mathcal{C}$ where $b \in_R \{0, 1\}$. Assume that there are q_1 entities MS_i's and q_2 entities V_j's. E is allowed to submit $Execute(\Pi^u_{MS_i,V_j}, \Pi^v_{V_j,MS_i})$ queries at most q_3 times where q_1, q_2, and q_3 are polynomials of security parameter k. $\mathcal{S}$ guesses two numbers λ and j' where E will return the guess bit b'' after making $Test(\Pi^u_{MS_i,V_{j'}})$ which is corresponding to the λth $Execute(\Pi^u_{MS_i,V_{j'}}, \Pi^v_{V_{j'},MS_i})$ query. Then, $\mathcal{S}$ initializes $i_{guess} = 0$ and generates all public/private key pairs (pk_{V_j}, sk_{V_j}) for all entities V_j's except the entity $V_{j'}$ whose public key will be set as pk received from $\mathcal{C}$ in Definition 7. When E makes $Send(\Pi^u_{MS_i,V_j}, \mathcal{M})$ and $Send(\Pi^v_{V_j,MS_i}, \mathcal{M})$ queries, $\mathcal{S}$ can deal with them and output the corresponding messages by running the protocols of Sections 3.3 and 3.4 with generated public/private keys (pk_{V_j}, sk_{V_j})'s and encryption/decryption oracles $(\mathcal{O}_E, \mathcal{O}_D)$. The operations of $Execute(\Pi^u_{MS_i,V_{j'}}, \Pi^v_{V_{j'},MS_i})$, $Test(\Pi^u_{MS_i,V_{j'}})$, $Reveal(T)$, and $Reveal(\Pi^u_{MS_i,V_j})$ queries are depicted in Figure 15.

```
$\Pi^u_{MS_i,V_j}$ : $Execute(\Pi^u_{MS_i,V_j}, \Pi^v_{V_j,MS_i})$
$i_{guess}$++;
Run the protocol of Section 3.:
  with $\mathcal{O}_T$ to get $(m, \delta, \sigma, \gamma, s)$;
Prepare $T = (F_1(m), \delta, \sigma, \gamma, s)$;
Store $(T, m)$ in $L_T$;
Select $m^{**}, k, r_1, r_2, r_3 \in_R \{0,1\}^{l_r}$;
Compute $\alpha = r^{e_H} F_1^2(m^{**})$;
Compute $\theta_1 = E_{pk_J}(k)$;
If $(V_j == V_{j'})$ {
    If $(i_{guess} \neq \lambda)$ {
        $\theta_2 = E_{pk}(r_1, r_2, r_3)$; }
    Else {
        $\theta_2 = \pi$; }
Else {
    $\theta_2 = E_{pk_{V_j}}(r_1, r_2, r_3)$; }
Send $(\alpha, \theta_1, T, \theta_2)$ to $\Pi^v_{V_j,MS_i}$;
Wait until receive $(r_1, r_4)$ from $\Pi^v_{V_j,MS_i}$;;
Compute $r_5 = F_3(r_2||r_4) \oplus m$;
Send $r_5$ to $\Pi^v_{V_j,MS_i}$;
If $(V_j = V_{j'}$ and $i_{guess} = \lambda)$ {
    Record $(u, MS_i, V_j, \tilde{r}_3)$ in $L_{K_{MS}}$;
Else {
    Record $(u, MS_i, V_j, r_3)$ in $L_{K_{MS}}$;
```

```
$Reveal(\Pi^u_{MS_i,V_j})$
Retrieve $r_3$ from $L_{K_{MS}}$ via $(u, MS_I, V_j)$;
If $(r_3 == m_0)$ {
    Abort; }
Else {
    Return $r_3$; }
```

```
$\Pi^v_{V_j,MS_i}$ : $Execute(\Pi^u_{MS_i,V_j}, \Pi^v_{V_j,MS_i})$
Receive $(\alpha, \theta_1, T, \theta_2)$ from $\Pi^u_{MS_i,V_j}$;
Parse $T$ as $(h_m, \delta, \sigma, \gamma, s)$;
If $(V_j == V_{j'})$ {
    If $(\theta_2 == \pi)$ {
        Set $(r_1, r_2) = (\tilde{r}_1, \tilde{r}_2)$
        Set $r_3 = \tilde{r}_3$; }
    Else {
        Send $\theta_2$ to $\mathcal{O}_D$ and;
        get $(r_1, r_2, r_3)$;
Else {
    Compute $(r_1, r_2, r_3) = D_{sk_{V_j}}(\theta_2)$
Select $r_4 \in_R \{0,1\}^{l_r}$;
Send $(r_1, r_4)$ to $\Pi^u_{MS_i,V_j}$;
Wait until receive $r_5$ from $\Pi^u_{MS_i,V_j}$;
Record $(v, MS_i, V_j, r_3)$ in $L_{K_V}$;
```

```
$Test(\Pi^u_{MS_i,V_j})$
Retrieve $r_3$ from $L_{K_{MS}}$
  via $(u, MS_i, V_j)$;
If $(r_3 == m_0)$ {
    Set $r_k = m_0$; }
Else {
    Select $b' \in_R \{0,1\}$;
    If $(b' == 0)$ {
        Set $r_k = r_3$; }
    Else {
        Select $r_k \in_R \{0,1\}^{l_r}$;}
Return $r_k$; }
```

```
$Reveal(T)$
Retrieve $m$ from $L_T$ via $T$;
Return $m$;
```

Figure 15. The oracles in the proof model of Theorem 6.

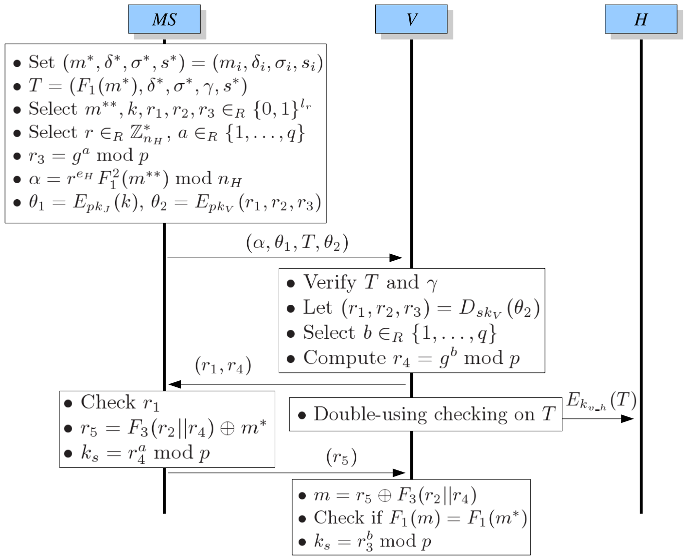
In Line 3 of $Reveal(\Pi^{u}_{MS_i,V_j})$ in Figure 15, $\mathcal{S}$ will abort the simulation because it cannot return the established session key. If the simulation is aborted, $\mathcal{S}$ will randomly guess $b' \in \{0,1\}$. Otherwise, after E performs $Test(\Pi^{u}_{MS_i,V_j})$ and outputs b'', $\mathcal{S}$ will set $b' = b''$. Thus, $\mathcal{S}$ has probability ϵ' in outputting a correct bit b' $(= b)$ to $\mathcal{C}$ where $\epsilon' \geq \frac{\epsilon+\frac{1}{2}}{q_2q_3} + \frac{q_2q_3-1}{2q_2q_3} = \frac{2\epsilon+1+q_2q_3-1}{2q_2q_3} = \frac{\epsilon}{q_2q_3} + \frac{1}{2}$. Hence, if ϵ is non-negligible, $Adv^{IND-CCA}_{\mathcal{F}}(k) = \epsilon' - \frac{1}{2} \geq \frac{\epsilon}{q_2q_3}$ is also non-negligible.

□

5. The Forward Secrecy Extension

Forward secrecy is an advanced security feature which makes the past session keys still secure even though the long-term key of a system was stolen by attackers. If a scheme is not with forward secrecy, an attacker, who has gotten the long-term key by some means, can compute all past session keys which were derived from the long-term key.

Our anonymous authentication protocol can be easily extended to own the feature of forward secrecy by adopting Diffie-Hellman key exchange protocol [24]. The extended protocol is given in Figure 16. Let p be a prime and g be a generator with order q in $\mathbb{Z}^*_p$ where q is also prime and $q|(p-1)$. In the extended authentication protocol, when MS is preparing θ_2, she/he randomly chooses an integer $a \in \{1,\ldots,q\}$ and compute $r_3 = g^a \bmod p$. Then, MS sets $\theta_2 = E_{pk_V}(r_1, r_2, r_3)$. V prepares $r_4 = g^b \bmod p$ where b is randomly selected from $\{1,\ldots,q\}$ and sends (r_1, r_4) to MS. Finally, MS computes the session key $k_s = r_4^a \bmod p$ and V computes $k_s = r_3^b \bmod p$.



MS
V
H
• Set $(m^*, \delta^*, \sigma^*, s^*) = (m_i, \delta_i, \sigma_i, s_i)$
• $T = (F_1(m^*), \delta^*, \sigma^*, \gamma, s^*)$
• Select $m^{**}, k, r_1, r_2, r_3 \in_R \{0,1\}^{l_r}$
• Select $r \in_R \mathbb{Z}^*_{n_H}$, $a \in_R \{1,\ldots,q\}$
• $r_3 = g^a \bmod p$
• $\alpha = r^{e_H} F_1^2(m^{**}) \bmod n_H$
• $\theta_1 = E_{pk_J}(k)$, $\theta_2 = E_{pk_V}(r_1, r_2, r_3)$
$(\alpha, \theta_1, T, \theta_2)$
• Verify T and γ
• Let $(r_1, r_2, r_3) = D_{sk_V}(\theta_2)$
• Select $b \in_R \{1,\ldots,q\}$
• Compute $r_4 = g^b \bmod p$
(r_1, r_4)
• Check r_1
• $r_5 = F_3(r_2||r_4) \oplus m^*$
• $k_s = r_4^a \bmod p$
• Double-using checking on T
$E_{k_{v\_h}}(T)$
(r_5)
• $m = r_5 \oplus F_3(r_2||r_4)$
• Check if $F_1(m) = F_1(m^*)$
• $k_s = r_3^b \bmod p$

Figure 16. The proposed anonymous authentication protocol with forward secrecy.

In the extended version, the mobile user has to pay two more exponentiation computations, *i.e.*, $r_3 = g^a \bmod p$ and $k_s = r_4^b \bmod p$ for completing her/his authentication. The mobile user can pre-compute $r_3 = g^a \bmod p$ before the communication.

5.1. The Security Proof for the Forward Secrecy Extension

First, we define Decisional Diffie-Hellman (DDH) Assumption which was introduced in [25].

Definition 14 (DDH). *Let p be a prime and g be a generator with prime order q in $\mathbb{Z}^*_p$ where $q|(p-1)$. Given $(p, q, g, g^a \bmod p, g^b \bmod p, g^c \bmod p)$, it is computationally indistinguishable to decide if $c \equiv ab \pmod{q}$.*

We modify the experiment of Definition 12 as follows.

Definition 15. *Let $k \in \mathbb{N}$ be a security parameter. In this experiment, the adversary E will try to guess that the returned value r_k from $Test(\Pi^u_{MS_i,V_j})$ query is a random string or the real session key.*

Experiment $Exp_E^{GoodGuessFS}(k)$
$- b'' \leftarrow E^{Send,Execute,Reveal,Corrupt}(r_k = Test(\Pi^u_{MS_i,V_j}))$
If the followings are true, return 1; else return 0.
1. $b' = b''$
2. E *has never submitted* $Reveal(\Pi^u_{MS_i,V_j})$.
3. E *makes* $Corrupt(V_j)$ *query when the session* u' *has been finished where* $u < u'$.

We define the advantage of E is $Adv_E^{GoodGuessFS}(k) = Pr[Exp_E^{GoodGuessFS}(k) = 1] - \frac{1}{2}$.

Definition 16. *A protocol $P = (\Pi, \mathcal{G})$ is with forward secrecy if P is a secure authenticated key exchange protocol and $Adv_E^{GoodGuessFS}(k)$ is negligible.*

Theorem 7. *The extension of the authentication protocol in Section 5 is a secure authentication protocol with forward secrecy.*

Proof. The proof model is illustrated in Figure 17. We will construct a simulator $\mathcal{S}_{FS}$ who obtains $(p, q, g, g^{\bar{a}} \bmod p, g^{\bar{b}} \bmod p, g^{\bar{c}} \bmod p)$ and simulates the communication environment under the presence of an adversary E. There are q_1 entities MS_i's and q_2 entities V_j's in the communication environment. Assume that E makes $Execute(\Pi^u_{MS_i,V_j}, \Pi^v_{V_j,MS_i})$ queries at most q_3 times. $\mathcal{S}_{FS}$ generates the public/private keys (pk_{V_j}, sk_{V_j}) for V_j where $j = \{1, \ldots, q_2\}$. $\mathcal{S}_{FS}$ guesses a number λ where E will output correct bit b'' for the $Test(\Pi^u_{MS_i,V_j})$ query and $\Pi^u_{MS_i,V_j}$ is involved in the λth $Execute(\Pi^u_{MS_i,V_j}, \Pi^v_{V_j,MS_i})$ query. When E makes $Send(\Pi^u_{MS_i,V_j}, \mathcal{M})$ and $Send(\Pi^v_{V_j,MS_i}, \mathcal{M})$ queries, $\mathcal{S}_{FS}$ can return the corresponding response messages by performing the protocol of Sections 3.3 and 3.4 with the generated public/private keys (pk_{V_j}, sk_{V_j})'s. Besides, $\mathcal{S}_{FS}$ resets $i_{guess} = 0$ and $Reveal(T)$, $Reveal(\Pi^u_{MS_i,V_j})$, $Test(\Pi^u_{MS_i,V_j})$, and $Execute(\Pi^u_{MS_i,V_j}, \Pi^v_{V_j,MS_i})$ queries are defined in Figure 18.

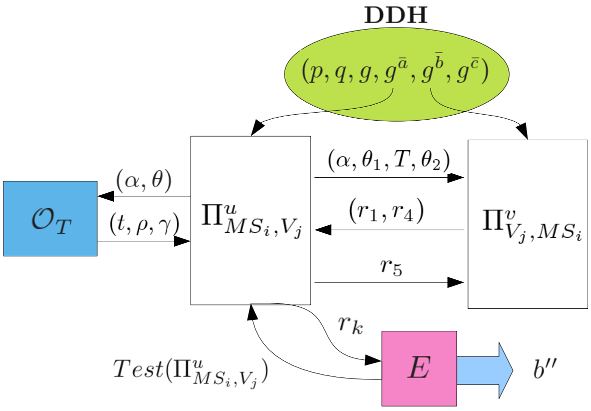
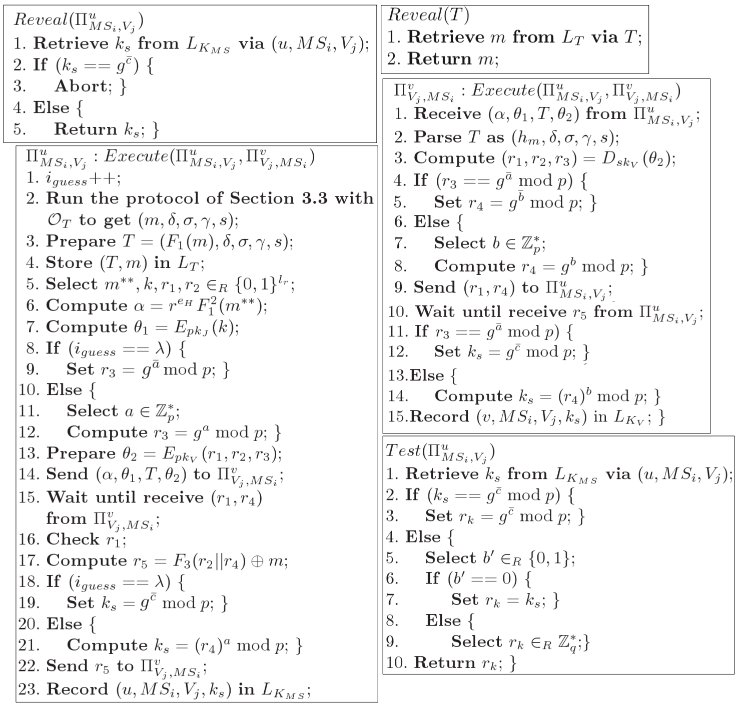


Figure 17. The proof model of Theorem 7.



```
$Reveal(\Pi^u_{MS_i,V_j})$
Retrieve $k_s$ from $L_{K_{MS}}$ via $(u, MS_i, V_j)$;
If $(k_s == g^{\bar{c}})$ {
    Abort; }
Else {
    Return $k_s$; }
```

```
$\Pi^u_{MS_i,V_j} : Execute(\Pi^u_{MS_i,V_j}, \Pi^v_{V_j,MS_i})$
$i_{guess}$++;
Run the protocol of Section 3.3 with
  $\mathcal{O}_T$ to get $(m, \delta, \sigma, \gamma, s)$;
Prepare $T = (F_1(m), \delta, \sigma, \gamma, s)$;
Store $(T, m)$ in $L_T$;
Select $m^{**}, k, r_1, r_2 \in_R \{0,1\}^{l_r}$;
Compute $\alpha = r^{e_H} F_1^2(m^{**})$;
Compute $\theta_1 = E_{pk_J}(k)$;
If $(i_{guess} == \lambda)$ {
    Set $r_3 = g^{\bar{a}} \bmod p$; }
Else {
    Select $a \in \mathbb{Z}_p^*$;
    Compute $r_3 = g^a \bmod p$; }
Prepare $\theta_2 = E_{pk_V}(r_1, r_2, r_3)$;
Send $(\alpha, \theta_1, T, \theta_2)$ to $\Pi^v_{V_j,MS_i}$;
Wait until receive $(r_1, r_4)$
  from $\Pi^v_{V_j,MS_i}$;
Check $r_1$;
Compute $r_5 = F_3(r_2 || r_4) \oplus m$;
If $(i_{guess} == \lambda)$ {
    Set $k_s = g^{\bar{c}} \bmod p$; }
Else {
    Compute $k_s = (r_4)^a \bmod p$; }
Send $r_5$ to $\Pi^v_{V_j,MS_i}$;
Record $(u, MS_i, V_j, k_s)$ in $L_{K_{MS}}$;
```

```
$Reveal(T)$
Retrieve $m$ from $L_T$ via $T$;
Return $m$;
```

```
$\Pi^v_{V_j,MS_i} : Execute(\Pi^u_{MS_i,V_j}, \Pi^v_{V_j,MS_i})$
Receive $(\alpha, \theta_1, T, \theta_2)$ from $\Pi^u_{MS_i,V_j}$;
Parse $T$ as $(h_m, \delta, \sigma, \gamma, s)$;
Compute $(r_1, r_2, r_3) = D_{sk_V}(\theta_2)$;
If $(r_3 == g^{\bar{a}} \bmod p)$ {
    Set $r_4 = g^{\bar{b}} \bmod p$; }
Else {
    Select $b \in \mathbb{Z}_p^*$;
    Compute $r_4 = g^b \bmod p$; }
Send $(r_1, r_4)$ to $\Pi^u_{MS_i,V_j}$;
Wait until receive $r_5$ from $\Pi^u_{MS_i,V_j}$;
If $r_3 == g^{\bar{a}} \bmod p)$ {
    Set $k_s = g^{\bar{c}} \bmod p$; }
Else {
    Compute $k_s = (r_4)^b \bmod p$; }
Record $(v, MS_i, V_j, k_s)$ in $L_{K_V}$; }
```

```
$Test(\Pi^u_{MS_i,V_j})$
Retrieve $k_s$ from $L_{K_{MS}}$ via $(u, MS_i, V_j)$;
If $(k_s == g^{\bar{c}} \bmod p)$ {
    Set $r_k = g^{\bar{c}} \bmod p$; }
Else {
    Select $b' \in_R \{0,1\}$;
    If $(b' == 0)$ {
        Set $r_k = k_s$; }
    Else {
        Select $r_k \in_R \mathbb{Z}_q^*$;}
Return $r_k$; }
```

Figure 18. The oracles of the proof of Theorem 7.

After the λth $Execute(\Pi^u_{MS_i,V_j}, \Pi^v_{V_j,MS_i})$ query, E makes $Test(\Pi^u_{MS_i,V_j})$ query and outputs a guess bit b''. Then $\mathcal{S}_{FS}$ can try to solve the DDH problem as follows. If $b'' = 0$, $\mathcal{S}_{FS}$ decides $c \equiv ab \pmod q$. If $b'' = 1$, $\mathcal{S}_{FS}$ decides $c \neq ab \pmod q$. Besides, if E does not make $Test(\Pi^u_{MS_i,V_j})$ query for the λth $Execute(\Pi^u_{MS_i,V_j}, \Pi^v_{V_j,MS_i})$, $\mathcal{S}_{FS}$ randomly chooses $b' \in \{0,1\}$. Assume that E has probability at least $(\epsilon + \frac{1}{2})$ with non-negligible ϵ to output correct bit b''. Thus, $\mathcal{S}_{FS}$ has non-negligible advantage at least $\frac{\epsilon+\frac{1}{2}}{q_3} + \frac{q_3-1}{2q_3} - \frac{1}{2} = \frac{\epsilon}{q_3}$ to solve the DDH assumption, *i.e.*, $Adv_E^{GoodGuessFS}(k) \geq \frac{\epsilon}{q_3}$.
□

6. Comparisons and Performance Evaluation

6.1. Comparisons

First, we describe some features as follows where these features are required for mobile users when they roam around the mobile networks.

1. Hiding identity: Mobile users hide their real identities from the system operator, H and V, and eavesdroppers.
2. No relation: It is difficult for the system to derive the relation between any two rounds of the communication of the same mobile user.
3. Secure channels: After performing mutual authentication between an anonymous mobile user and the system operator, they must establish a shared session key for the following communication activities.
4. Fair privacy: Fair privacy contains traceability and revokeability. If a crime happens, the police can trace the identities of related anonymous mobile users or the judge can revoke their privacy.
5. Credit-based chargeability: As mentioned in Section 3.5, the credit-based charging method is better than the debit-based one since the former (1) is the same as the practical situation in current GSM services; (2) can greatly reduce the relations between any two rounds of communication; and (3) is free from the problem of overspending.

The comparisons between our proposed scheme and the others are summarized in Table 1. In Table 1, the authors of [6] also mentioned untraceability and revokeability, but they did not realize them in their scheme. We believe that realizing untraceability and revokeability is not trivial.

Table 1. Comparisons.

	Privacy							**Property**
Scheme	**Hiding ID from:**							**Credit-Based**
	H	**V**	**E**	**NoR**	**S**	**T**	**R**	**Chargeability**
Ours	○	○	○	○	○	○	○	○
[6]	○	○	○	×	○	△	△	×
[7]	○	○	○	×	×	×	×	×
[8]	×	○	○	○	○	×	×	×
[9]	×	×	○	○	○	×	×	×
[10]	×	○	○	×	○	×	×	○
[11]	○	○	○	○	×	×	×	×
[12]	○	○	○	×	×	×	×	×
[26]	×	○	○	×	×	×	×	×
[27]	×	○	○	×	○	×	×	×
[28]	×	○	○	×	○	○	○	×

H: Home domain; V: Visiting domain; E: Eavesdroppers; NoR: Hard to derive relation between any two rounds; S: Secure channel; T: Traceability (Tracing a criminal user); R: Revokeability (Revoking the privacy of a user when necessary); ○: Achieving the feature; ×: Not achieving the feature; △: Not realizing the feature.

6.2. Performance Evaluation

In Table 2, we summarize the computation cost of the proposed protocols where E denotes the cost of a modulo exponentiation computation.

Table 2. Computation evaluation.

Operation	**Mobile User** (MS)	**The System** (V+H)
Requesting a ticket	4E	6E
Using a ticket	3E	2E
Termination	3E	8E

Besides, we show the benchmark of Crypto++, which is a C++ class library of cryptographic computations, in Table 3 [29]. The benchmark is measured by running Crypto++ on a machine with Intel Celleron 450MHz CPU under Windows 2000. Furthermore, we also list the hardware specifications of some recently popular mobile devices in Table 4 [30]. In Table 4, we also implemented RSA Cryptography system to check if our proposed system is practically efficient. According to Tables 3 and 4, we can objectively say that our protocols can be implemented and efficiently executed in the present mobile devices. Consequently, our anonymous authentication protocols can be performed in a reasonable time when a mobile user takes her/his mobile device to roam over the mobile network.

Table 3. The benchmark of Crypto++.

CPU: Intel Celleron 450 MHz, OS: Windows 2000			
RSA Operation	**Iterations**	**Total Time**	**Milliseconds/Operation**
1024 Encryption	41,051	30 s	0.73
1024 Decryption	1,084	30 s	27
2048 Encryption	13,912	30 s	2
2048 Decryption	164	30 s	183
1024 Signature	1,086	30 s	27
1024 Verification	43,061	30 s	0.69
2048 Signature	165	30 s	181
2048 Verification	14,187	30 s	2

Table 4. Some popular mobile devices.

Mobile Device	**CPU**	**Memory**	**Execution Time**
Mac iphone 3G	Samsung S5L8900 620 MHz	128 MB	**E**: 3 ms, **D**: 21 ms
HTC magic	Qualcomm MSM 7201A 528 MHz	192 MB	
Noika N95	Dual ARM 11 332 MHz	128 MB	
Sony Ericsson X1	Qualcomm MSM 7200 528 MHz	256 MB	

E: encrypting 256 bits of data; D: decrypting 256 bit of data.

7. Conclusions

We have proposed a mobile authentication scheme which can authenticate mobile users anonymously. When a mobile user enters the anonymity mode, she/he can perform a mutual authentication process with the system operator. The system operator can charge the anonymous user correctly according to the time she/he consumed by a credit-based method. Furthermore, if some mobile user misuses the anonymity property, the judge can revoke her/his privacy and trace her/him.

In the proposed scheme, the privacy of an honest mobile user might be broken by the system operator if the mobile user lost her/his ticket since the system operator must trace her/his used tickets in order to find the spending value of her/him. Finding a solution to cope with the problem would be the subject of an interesting research topic.

Acknowledgments: This work was supported in part by the Ministry of Science and Technology of the Taiwan under grants MOST 104-2221-E-110-043, MOST 105-2923-E-110-001-MY3, and ”Aim for the Top University Plan” of the National Sun Yat-sen University and Ministry of Education, Taiwan.

Author Contributions: Both of the authors worked collaboratively in the design of the scheme, proofs for its security, and analyses on its performance.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix

We define an experiment as follows.

Definition A1 (The Experiment of Hash Querying)**.** *Let $k \in \mathcal{N}$ be a security parameter and $\Pi^u_{MS_i,V_j}$ and $\Pi^v_{V_j,MS_i}$ are two oracles who play the roles of MS_i and V_j in Section 3.4, respectively. There exists an attacker $\mathcal{A}_{HQ}$ who can perform Execute$(\Pi^u_{MS_i,V_j}, \Pi^v_{V_j,MS_i})$ queries at most q_{HQ} times and observe all communication flows which are $(\alpha_\lambda, \theta_{1_\lambda}, T_\lambda, \theta_{2_\lambda})$, $(r_{1_\lambda}, r_{4_\lambda})$, and r_{5_λ} where $\lambda = \{1, \ldots, q_{HQ}\}$.*

$\mathcal{A}_{HQ}$ can submit $Reveal(T_\lambda)$ to get m_λ and perform the $Reveal(r_{5_\lambda})$ query to obtain $(r_{2_\lambda}||r_{4_\lambda})$ where $r_{5_\lambda} = F_3(r_{2_\lambda}||r_{4_\lambda}) \oplus m_\lambda$. Consider the following experiment:

Experiment $Exp^{HQ}_{\mathcal{A}_{HQ}}(k)$
- $(\hat{r}, \hat{\lambda}) \leftarrow \mathcal{A}^{Execute,Reveal}_{HQ}(all\ communication\ flows)$
If the followings are true, return 1; else return 0.
1. $\mathcal{A}_{HQ}$ has never submitted $Reveal(r_{5_{\hat{\lambda}}})$ query.
2. $r_{5_{\hat{\lambda}}} = F_3(\hat{r}) \oplus m_{\hat{\lambda}}$
3. $\hat{\lambda} \in \{1, \ldots, q_{HQ}\}$.

We define the advantage of $\mathcal{A}_{HQ}$ as $Adv^{HQ}_{\mathcal{A}_{HQ}}(k) = Pr[Exp^{HQ}_{\mathcal{A}_{HQ}}(k) = 1]$.

Theorem A1. *If the advantage $Adv^{HQ}_{\mathcal{A}_{HQ}}(k)$ is non-negligible, $Adv^{IND-CCA}_{\mathcal{F}}(k)$ is also non-negligible.*

Proof. The proof model is shown in Figure A1. We will design a simulator $\mathcal{S}_{HQ}$ who can simulate the experiment of Definition A1. First, $\mathcal{S}_{HQ}$ randomly chooses four different strings, $(\tilde{r}_1, \tilde{r}_2, \tilde{r}'_2, \tilde{r}_3)$, and sets $m_0 = (\tilde{r}_1, \tilde{r}_2, \tilde{r}_3)$ and $m_1 = (\tilde{r}_1, \tilde{r}'_2, \tilde{r}_3)$. $\mathcal{S}_{HQ}$ then sends (m_0, m_1) to the challenger $\mathcal{C}$, which was defined in Definition 7, and gets pk and $\pi = E_{pk}(m_b)$ where $b \in_R \{0, 1\}$. Then $\mathcal{S}_{HQ}$ creates q_1 mobile users MS_i's and q_2 entities V_j's and guesses two integers j' and λ' where $1 \leq j' \leq q_2$ and $1 \leq \lambda' \leq q_{HQ}$, *i.e.*, $\mathcal{S}_{HQ}$ guesses that $\mathcal{A}_{HQ}$ will output $(\hat{r}, \hat{\lambda})$ where $\lambda' = \hat{\lambda}$ and $V_{j'}$ will be involved in the λ'th $Execute(\Pi^u_{MS_i,V_{j'}}, \Pi^v_{V_{j'},MS_i})$ query. $\mathcal{S}$ also generates public/private key pairs for V_j where $j = \{1, \ldots, q_2\}$ and $j \neq j'$. The public key of $V_{j'}$ will be set as pk.

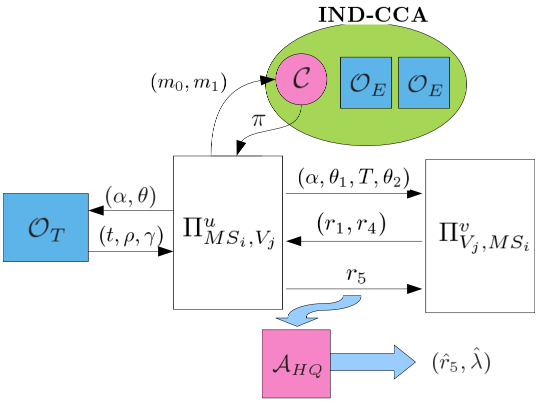


Figure A1. The proof model of Theorem A1.

During the simulation, $\mathcal{A}_{HQ}$ must send $(r_2||r_4)$ to $\mathcal{O}_{F_3}$ to request the F_3-hashed value of $(r_2||r_4)$, *i.e.*, $F_3(r_2||r_4)$. $\mathcal{S}_{HQ}$ initially sets $i_{guess} = 0$ and empties four lists $L_{K_{MS}}$, L_{K_V}, L_{F_3}, and L_T. $Reveal(r_5)$, $Reveal(T)$, $\mathcal{O}_{F_3}$, and $Execute(\Pi^u_{MS_i,V_{j'}}, \Pi^v_{V_j,MS_i})$ are defined in Figure A2.

```
$\Pi^u_{MS_i,V_j}$ : $Execute(\Pi^u_{MS_i,V_j}, \Pi^v_{V_j,MS_i})$
1. $i_{guess}$++;
2. Run the protocol of Section 3.3 with $\mathcal{O}_T$
   to get $(m, \delta, \sigma, \gamma, s)$;
3. Prepare $T = (F_1(m), \delta, \sigma, \gamma, s)$;
4. Record $(m, T)$ in $L_T$;
5. Select $m^{**}, k, r_1, r_2, r_3 \in_R \{0,1\}^{l_r}$;
6. Compute $\alpha = r^{e_H} F_1^2(m^{**}) \bmod n_H$;
7. Compute $\theta_1 = E_{pk_J}(k)$;
8. If $(V_j == V_{j'})$ {
9.     If $(i_{guess} \neq \lambda')$ {
10.        $\theta_2 = E_{pk}(r_1, r_2, r_3)$; }
11.    Else {
12.        $\theta_2 = \pi$; }}
13. Else {
14.    $\theta_2 = E_{pk_{V_j}}(r_1, r_2, r_3)$; }
15. Send $(\alpha, \theta_1, T, \theta_2)$ to $\Pi^v_{V_j,MS_i}$;
16. Wait until receive $(r_1, r_4)$ from $\Pi^v_{V_j,MS_i}$;
17. If $(V_j == V_{j'}$ and $i_{guess} == \lambda')$ {
18.    Select $r_5 \in_R \{0,1\}^{l_r}$; }
19. Else {
20.    Send $(r_2||r_4)$ to $\mathcal{O}_{F_3}$ and get $h_{f_3}$;
21.    Compute $r_5 = h_{f_3} \oplus m$; }
22. Send $r_5$ to $\Pi^v_{V_j,MS_i}$;
23. If $(V_j \neq V_{j'}$ or $i_{guess} \neq \lambda')$ {
24.    Update $(\perp, r_2||r_4, h_{f_3})$ as $(r_5, r_2||r_4, h_{f_3})$
       in $L_{F_3}$; }
25. Record $(MS_i, u, r_3)$ in $L_{K_{MS}}$;
```

```
$Reveal(T)$
1. If ($T$ exists in $L_T$) {
2.     Retrieve $m$ from $L_T$ via $T$;
3.     Return $m$; }
4. Else {
5.     Return $\perp$; }
```

```
$\Pi^v_{V_j,MS_i}$ : $Execute(\Pi^u_{MS_i,V_j}, \Pi^v_{V_j,MS_i})$
1. Receive $(\alpha, \theta_1, T, \theta_2)$ from $\Pi^u_{MS_i,V_j}$;
2. Parse $T$ as $(h_m, \delta, \sigma, \gamma, s)$;
3. If $(V_j == V_{j'})$ {
4.     If $(\theta_2 == \pi)$ {
5.         Set $(r_1, r_2, r_3) = (\tilde{r}_1, \perp, \tilde{r}_3)$; }
6.     Else {
7.         Send $\theta_2$ to $\mathcal{O}_D$ and
           get $(r_1, r_2, r_3)$; }
8. Else {
9.     $(r_1, r_2, r_3) = D_{sk_{V_j}}(\theta_2)$; }
10. Do {
11.    Select $r_4 \in_R \{0,1\}^{l_r}$;
12. } Until $r_4$ is not a suffix of any
       $(r_2||r_4)$ stored in $L_{F_3}$;
13. Send $(r_1, r_4)$ to $\Pi^u_{MS_i,V_j}$;
14. Record $(V_j, v, r_3)$ in $L_{K_V}$;
```

```
$\mathcal{O}_{F_3}(r_2||r_4)$
1. If ($(r_2||r_4)$ exists in $L_{F_3}$) {
2.     Retrieve $h_{f_3}$ from $L_{F_3}$ via $(r_2||r_4)$; }
3. Else {
4.     Select $h_{f_3} \in_R \{0,1\}^{l_r}$;
5.     Record $(\perp, r_2||r_4, h_{f_3})$ in $L_{F_3}$; }
6. Return $h_{f_3}$;
```

```
$Reveal(r_5)$
1. If ($r_5$ exists in $L_{F_3}$) {
2.     Retrieve $(r_2||r_4)$ from $L_{F_3}$ via $r_5$;
3.     Return $(r_2||r_4)$; }
4. Else {
5.     Abort; }
```

Figure A2. The oracles in the proof of Theorem A1.

Finally, if $\mathcal{A}_{HQ}$ outputs $(\hat{r}, \hat{\lambda})$, $\mathcal{S}_{HQ}$ can guess b' as follows. If $\hat{\lambda} = \lambda'$, $\mathcal{S}_{HQ}$ guesses $b' = 0$ when $\hat{r} = (\tilde{r}_2||r_{4_{\lambda'}})$, $b' = 1$ when $\hat{r} = (\tilde{r}'_2||r_{4_{\lambda'}})$, $b' \in_R \{0,1\}$ when $\hat{r} \neq (\tilde{r}_2||r_{4_{\lambda'}})$ and $\hat{r} \neq (\tilde{r}'_2||r_{4_{\lambda'}})$. If $\hat{\lambda} \neq \lambda'$, $\mathcal{S}_{HQ}$ randomly outputs $b' \in \{0,1\}$.

If $\mathcal{A}_{HQ}$ has probability ϵ_{HQ}, not less than a non-negligible probability, to output the correct string $\hat{r}$, $\mathcal{S}_{HQ}$ has probability ϵ' to output b' such that $b' = b$ where

$$\begin{aligned}
\epsilon' &= \frac{\epsilon_{HQ} + (1-\epsilon_{HQ})\frac{1}{2^{l_r}}}{q_{HQ}q_2} + \frac{(1-\epsilon_{HQ})(\frac{2^{l_r}-1}{2^{l_r}})}{2q_{HQ}q_2} + \frac{q_{HQ}q_2 - 1}{2q_{HQ}q_2} \\
&\geq \frac{2\epsilon_{HQ} + 1 - \epsilon_{HQ} + q_{HQ}q_2 - 1}{2q_{HQ}q_2} \\
&= \frac{\epsilon_{HQ}}{2q_{HQ}q_2} + \frac{1}{2}.
\end{aligned}$$

Thus, $\mathcal{S}_{HQ}$ is an adversary $\mathcal{F}$ who has non-negligible advantage $Adv_{\mathcal{F}}^{IND-CCA}(k) \geq \frac{\epsilon_{HQ}}{2q_{HQ}q_2}$ in the IND-CCA game.

□

References

1. Fragkiadakis, A.G.; Askoxylakis, L.G.; Tragos, E.Z.; Verikoukis, C.V. Ubiquitous Robust Communications for Emergency Response Using Multi-operator Heterogeneous Networks. *EURASIP J. Wirel. Commun. Netw.* **2011**, *13*, 1–16.
2. Hwang, K.F.; Chang, C.C. A Self-encryption Mechanism for Authentication of Roaming and Teleconference Services. *IEEE Trans. Wirel. Commun.* **2003**, *2*, 400–407.
3. Samfat, D.; Molva, R.; Asokan, N. Untraceability in Mobile Networks. In Proceedings of the 1st Annual International Conference on Mobile Computing and Networking, Berkeley, CA, USA, 13–15 November 1995; pp. 26–36.
4. Asokan, N. Anonymity in a Mobile Computing Environment. In Proceedings of the Workshop on Mobile Computing System and Applications, Santa Cruz, CA, USA, 1994, 8–9 December 1994; pp. 200–204.

5. Ozturk, C.; Zhang, Y.; Trappe, W.; Ott, M. Source-location Privacy for Networks of Energy-constrained Sensors. In Proceedings of the Second IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems (WSTFEUS'04), Vienna, Austria, 11–12 May 2004; pp. 68–72.
6. Karygiannis, A.; Kiayias, A.; Tsiounis, Y. A Solution for Wireless Privacy and Payments Based on E-cash. In Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks, Athens, Greece, 5–9 September 2005; pp. 206–218.
7. He, Q.; Wu, D.; Khosla, P. The Quest for Personal Control over Mobile Location Privacy. *IEEE Commun. Mag.* **2004**, *42*, 130–136.
8. Park, S.Y.; Han, M.S.; Eom, Y.I. An Efficient Authentication Protocol Supporting Privacy in Mobile Computing Environments. In Proceedings of the 5th IEEE International Conference on High Speed Networks and Multimedia Communications, Jeju Island, Korea, 3–5 July 2002; pp. 332–334.
9. Zhu, J.; Ma, J. A New Authentication Scheme with Anonymity for Wireless Environments. *IEEE Trans. Consum. Electron.* **2004**, *50*, 231–235.
10. Kesdogan, D.; Fouletier, X. Secure Location Information Management in Cellular Radio Systems. In Proceedings of the IEEE Wireless Communication System Symposium, Smithtown, NY, USA, 27–28 November 1995; pp. 35–40.
11. Lin, W.D.; Jan, J.K. A Wireless-based Authentication and Anonymous Channels for Large Scale Area. In Proceedings of the Sixth IEEE Symposium on Computers and Communications, Hammamet, Tunisia, 3–5 July 2001; pp. 36–41.
12. Tracz, R.; Wrona, K. Fair Electronic Cash Withdrawal and Change Return for Wireless Networks. In Proceedings of the 1st International Workshop on Mobile Commerce, Rome, Italy, 21 July 2001; pp. 14–19.
13. Chaum, D. Blind Signature Systems. In *Advance in Cryptology—CRYPTO '83*; Springer: Berlin/Heidelberg, Germany, 1984; p. 153.
14. Fan, C.-I.; Huang, V.-S. Anonymous Authentication Protocols with Credit-Based Chargeability and Fair Privacy for Mobile Communications. In *International Workshop on Security (IWSEC), LNCS 4752*; Springer-Verlag: Berlin/Heidelberg, Germany, 2007; pp. 412–427.
15. Canetti, R.; Halevi, S.; Katz, J. A Forward-secure Public-key Encryption Scheme. *J. Cryptol.* **2007**, *20*, 265–294.
16. Pearson, S. *Trusted Computing Platforms, the Next Security Solution*; Technical Report HPL-2002-221; Hewllet-Packard Laboratories: Bristol, UK, 2002.
17. Trusted Computing Group Website. Available online: http://www.trustedcomputinggroup.org/ (accessed on 17 January 2016).
18. Bajikar, S. *Trusted Plateform Module (TPM) Based Security on Notebook PCs-White Paper*; Mobile Platform Group, Intel Corporation: Santa Clara, CA, USA, 2002; Volume 1, p. 1.
19. Horowitz, E.; Sahni, S. Computing Partitions with Applications to the Knapsack Problem. *J. ACM* **1974**, *21*, 277–292.
20. Bellare, M.; Namprempre, C.; Pointcheval, D.; Semanko, M. The One-more-rsa-inversion Problems and the Security of Chaum's Blind Signature Scheme. *J. Cryptol.* **2008**, *16*, 185–215.
21. Bellare M.; Rogaway, P. Entity Authentication and Key Distribution. In *Advances in Cryptology—CRYPTO' 93*; Springer: Berlin/Heidelberg, Germany, 1994; Volume 773, pp. 232–249.
22. Blake-Wilson, S.; Menezes, A. Entity Authentication and Authenticated Key Transport Protocols Employing Asymmetric Techniques. In *Security Protocols*; Springer: Berlin/Heidelberg, Germany, 1998; pp. 137–158.
23. Goldwasser, S.; Micali, S. Probabilistic Encryption. *J. Comput. Syst. Sci.* **1984**, *28*, 270–299.
24. Diffie, W.; Hellman, M.E. New Directions in Cryptography. *IEEE Trans. Inf. Theory* **1976**, *22*, 644–654.
25. Boneh, D. The Decision Diffie-Hellman Problem. In Proceedings of the Third Algorithmic Number Theory Symposium, Portland, OR, USA, 21–25 June 1998; Volume 1423, pp. 48–63.
26. Mu, Y.; Varadharajan, V. A New Scheme of Credit Based Payment for Electronic Commerce. In Proceedings of the 23rd Annual Conference on Local Computer Networks, LowelI, MA, USA, 11–14 October 1998; pp. 278–284.
27. Yang, C.C.; Tang, Y.L.; Wang, R.C.; Yang, H.W. A Secure and Efficient Authentication Protocol for Anonymous Channel in Wireless Communications. *Appl. Math. Comput.* **2005**, *169*, 1431–1439.

28. Yeh, K.H. An Anonymous and Lightweight Authentication Scheme for Mobile Devices. *Inf. Technol. Control* **2015**, *44*, 206–215.
29. Crypto++. Crypto++ Benchmarks. Available online: http://www.packetstormsecurity.org/crypt/LIBS/cryptolib/benchmarks.html (accessed on 15 March 2009).
30. The GSM Phone Reviews Website. Available online: http://www.gsmar-ena.com/ (accessed on 22 May 2008).