12-2018

Privacy-preserving remote user authentication with K-times untraceability

Yangguang TIAN
Singapore Management University, ygtian@smu.edu.sg

Yingjiu LI
Singapore Management University, yjli@smu.edu.sg

Binanda SENGUPTA
Singapore Management University, binandas@smu.edu.sg

Robert H. DENG
Singapore Management University, robertdeng@smu.edu.sg

Albert CHING
i-Sprint Innovations

See next page for additional authors

Citation

TIAN, Yangguang; LI, Yingjiu; SENGUPTA, Binanda; DENG, Robert H.; CHING, Albert; and LIU, Weiwei. Privacy-preserving remote user authentication with K-times untraceability. (2018). *Information Security and Cryptology: 14th International Conference, Inscrypt 2018, Fuzhou, China, December 14-17: Proceedings*. 11449, 647-657.

Available at: https://ink.library.smu.edu.sg/sis_research/4365

Yangguang TIAN, Yingjiu LI, Binanda SENGUPTA, Robert H. DENG, Albert CHING, and Weiwei LIU

Privacy-Preserving Remote User Authentication with *k*-Times Untraceability

[1]Yangguang Tian, [1]Yingjiu Li, [1]Binanda Sengupta, [1]Robert H. Deng, [2]Albert Ching and [3]Weiwei Liu

[1]School of Information Systems, Singapore Management University, Singapore
[2]i-Sprint Innovations, Singapore
[3]School of Mathematics and Statistics,
North China University of Water Resources and Electric Power,
Zhengzhou, Henan, China
[1]{ygtian,yjli,binandas,robertdeng}@smu.edu.sg,[2]albert.ching@i-sprint.com,
[3]liuweiwei@ncwu.edu.cn

Abstract. Remote user authentication has found numerous real-world applications, especially in a user-server model. In this work, we introduce the notion of *anonymous remote user authentication with k-times untraceability* (k-RUA) for a given parameter k, where authorized users authenticate themselves to an authority (typically a server) in an anonymous and k-times untraceable manner. We define the formal security models for a generic k-RUA construction that guarantees user authenticity, anonymity and user privacy. We provide a concrete instantiation of k-RUA having the following properties: 1) a third party cannot impersonate an authorized user by producing valid transcripts for the user while conversing during a session; 2) a third party having access to the communication channel between the user and the authority cannot identify the session participants; 3) the authority can trace the real identities of dishonest users who have authenticated themselves for more than k times; 4) our k-RUA construction avoids using expensive pairing operations — which makes it efficient and suitable for devices having limited amount of computational resources.

Keywords: Remote User Authentication, Anonymity, User Privacy, k-times Untraceability

1 Introduction

User authentication is typically the first line of defense in most of the secure information systems. In the well-known user-server setting, a user has to authenticate herself to a (possibly remote) authentication server before opting for the services. Moreover, in order to protect a legitimate user's privacy, anonymous user authentication is widely studied in the literature and is deployed in numerous real-world applications. Researchers have come up with several solutions that exploit cryptographic techniques, such as group signatures [2], blind signatures and ring signatures [16], to ensure (or enhance) privacy. On the other hand, in some applications, it is required that a legitimate user can authenticate herself (and benefit from the services) for a limited number of times. For example, systems like e-cash, e-coupon and e-voting need such privacy guarantees. In such scenarios, k-times anonymous authentication (k-TAA) [14, 13, 3] serves the purpose. It is a fine-grained approach for privacy protection which ensures that a legitimate user can be authenticated anonymously only up to k number of times (for a threshold parameter k). On the other hand, if a user tries to authenticate herself beyond the threshold k, then her anonymity is compromised.

Although k-TAA schemes address the issue of restricting a user to bounded number of authentications, k-TAA is not suitable for building an authentication system for a mobile platform due to the following reasons. First, the traditional (and more generic) user authentication system involves an authentication server and multiple independent users, whereas k-TAA requires an extra (trusted) group manager. For example, it is cumbersome for a mobile device user if she has to consult a third party every time she enrolls to (or logs into) a server. Second, a mobile-platform-based system usually employs devices with low-power and limited resources, whereas k-TAA requires certain computation-intensive operations such as pairings (bilinear maps) and proofs of knowledge. Thus, to achieve both anonymity and traceability in a secure mobile setting is a non-trivial task.

In this work, we aim to design an efficient anonymous remote user authentication system suitable for mobile devices with a guarantee that a dishonest user deviating from the correct execution of the protocol can be traced. We explore whether we can exploit an e-coupon[1]/e-cash system to construct an anonymous remote user authentication system with traceability. We observe that an e-coupon system is more suitable than an e-cash system due to the following reasons. Unlike a bank, an e-coupon system, in general, does not involve central authority (group manager), which is required in an e-cash system in order to generate coins for the users. Moreover, e-cash system usually uses more expensive algorithms/protocols than e-coupon system. The deployment of mobile e-coupon systems [9] has showed their viability in practice. In the e-coupon system, the issue[2] protocol between a user and the vendor (service provider) can be applied to the enrollment phase of user authentication. On the other hand, the redeem protocol involves checking the authenticity of coupons, and certain services are redeemed in case coupons are valid. We do not consider whether the services are provided or not; we only focus on checking the authenticity of coupons in the user authentication setting as the goal of authentication server is to authenticate a legitimate (or authorized) user only.

We note that a secure and anonymous remote user authentication with traceability cannot be simply built upon existing e-coupon systems. The main concerns of existing e-coupon systems [6, 12, 5, 1, 10] can be listed as follows: unforgeability, double-redemption detection, unlinkability and unsplittability. The

[1] An e-coupon is also sometimes named as a multi-coupon as such a coupon can be redeemed more than once [6].

[2] An e-coupon system is usually comprised of issue and redeem protocols [6].

e-coupon system proposed by Liu et al. [11] has a new property: "k-times redemption detection" while the basic security requirements mentioned above are also met. Specifically, the real identity of a dishonest user can be traced by the service provider if the user tries to redeem the coupon more than k times — which aligns with our design goal. However, contrary to their claims, their e-coupon system fails to achieve traceability since a dishonest user in their system can misuse coupon without being detected.

1.1 This work

In this work, we introduce the notion of anonymous remote user authentication with k-times untraceability (k-RUA) that enables authorized users to authenticate themselves to a remote authentication server anonymously and ensures the traceability to detect dishonest users. Our contributions can be summarized as follows.

- We present the formal security definitions for privacy-preserving remote user authentication. In particular, we propose a user authenticity model to capture impersonation attacks, an anonymity model to address an honest-but-curious[3] authentication server and a user privacy to ensure the privacy of protocol participants.
- We present the *first* generic construction of k-RUA, which is built upon a secure e-coupon system. We prove it can achieve user authenticity, anonymity and user privacy. In particular, k-times untraceability enables an authorized user to authenticate herself to an authentication server up to k times without being traced. The real identity of a dishonest user is revealed to the authentication server in case the user tries to authenticate for more than k times.
- We show that the e-coupon system proposed in [11] fails to achieve their claimed k-times redemption detection. We fix their e-coupon system in our proposed k-RUA. In addition, we show that the same attacks are applicable to their previous work [10] and we also fix it accordingly.

1.2 Related Work

k-Times Anonymous Authentication. Teranishi et al. [14] proposed the first authentication scheme which allows users to anonymously perform the authentication at most k times (k-TAA). In particular, a user's identity is fully protected within the k-times authentication, while anyone is able to trace a dishonest user trying to authenticate herself beyond the allowable k times. Later on, dynamic k-TAA (denote k-TAA$'$) schemes were proposed in the literature [13, 3] that allow the service provider to independently grant/revoke a user from his access group in order to have better control over their clients. We note that some constructions [13, 3] were based on expensive pairings (bilinear maps), which are not suitable for devices with limited resources.

[3] The authentication server is assumed to execute the protocol as specified, just try to learn additional information from the transcript during protocol execution.

E-coupon System. The privacy-preserving e-coupon system was first proposed by Chen et al. [6] that allows a user purchase e-coupons and redeem them unlinkably. Furthermore, the number of redemptions remaining can be hidden from the vendor (i.e., coupon issuer). To reduce the cost for issuing and redeeming coupons, Nguyen [12] proposed an efficient e-coupon system which has constant communication and computation costs (that does not scale with the redemption limit k). Nguyen's e-coupon system also allows the coupon issuer to revoke an e-coupon. In an independent work [5], Canard et al. proposed an e-coupon system that is more efficient than [6]. They added new features to an e-coupon system that include the following: a user can choose the number of coupons she wants to issue; a user can choose the value of each coupon from a set of pre-defined values.

Armknecht et al. [1] proposed an e-coupon system that takes into account multiple vendors. Specifically, a user can redeem multiple coupons anonymously with different vendors in an arbitrary order. This system prevents double-spending by maintaining a trusted database that records the transaction of each redeemed coupon. Liu et al. [10] proposed a pairing-free e-coupon system that achieves both traceability against dishonest users and anonymity (i.e., untraceable) for honest users without involving any trusted *third* party.

In a recent work, Liu et al. [11] introduced a new notion called "strong user privacy", i.e., the privacy of the service chosen by a user during the redemption process (user redemption privacy). To meet strong user privacy requirements, they rely on an existing oblivious transfer scheme [7]. We also notice that the vendor can easily *link* two redemptions since a single coupon is issued by each user. However, the vendor cannot trace the real identities of honest users as long as the number of redemptions does not exceed k.

2 Security Model

Notation. We define a system with n users. We denote the i-th session established by a user U as Π_U^i, and identities of all the users recognised by Π_U^i during the execution of that session by partner identifier pid_U^i. We define sid_U^i as the unique session identifier belonging to the session i established by the user U. Specifically, $\mathsf{sid}_U^i = \{m_j\}_{j=1}^n$, where $m_j \in \{0,1\}^*$ is the message transcript among users.

We say an oracle Π_U^i may be *used* or *unused.* The oracle is considered as unused if it has never been initialized. The oracle is initialized as soon as it becomes part of a group. After the initialisation the oracle is marked as used and turns into the *stand-by* state where it waits for an invocation to execute a protocol operation. Upon receiving such invocation the oracle Π_U^i learns its partner identifier pid_U^i and turns into a *processing* state where it sends, receives and processes messages according to the description of the protocol. During that phase, the internal state information $state_U^i$ is maintained by the oracle. The oracle Π_U^i remains in the processing state until it collects enough information to finalise the user authentication. As soon as the authentication is accomplished Π_U^i *accepts* and *terminates* the protocol execution meaning that it would not send or receive further messages. If the protocol execution fails then Π_U^i terminates without being accepted.

2.1 System Model

A remote user authentication with k-times untraceability (k-RUA) involves two types of entities: multiple enorlled users and an authentication server. We define a k-RUA protocol that consists of the following algorithms/protocols:

- Setup: The authentication server $\mathbb{S}$ takes the security parameter λ as input, outputs the master public/secret key pair $(\texttt{mpk}, \texttt{msk})$.
- KeyGen: User takes master public key mpk as input, outputs a public/secret key pair $(\texttt{pk}, \texttt{sk})$.
- Enrollment: This is an interactive protocol that runs between an enrolled user and an authentication server $\mathbb{S}$ over a public channel. The enrolled user will generate a credential and become an authorized user after enrollment.
- Authentication: This is an interactive protocol between an authorized user and an authentication server $\mathbb{S}$ over a public channel. An authorized user sends her credential and k-size commitments to $\mathbb{S}$, while $\mathbb{S}$ accept it if and only if the credential send is valid.
- k-Times Untraceability: The authentication server $\mathbb{S}$ takes $k+1$ authentication transcripts of one user as input, outputs the user's secret key sk.

2.2 Security Model

User Authenticity. Informally, an adversary $\mathcal{A}$ attempts to impersonate an authorized user and authenticate to an authentication server[4]. We define a formal authenticity game between a probabilistic polynomial-time (PPT) adversary $\mathcal{A}$ and a simulator (i.e., challenger) $\mathcal{S}$ below.

- Setup. $\mathcal{S}$ generates a master public/secret key pair $(\texttt{mpk}, \texttt{msk})$ for authentication server $\mathbb{S}$ and public/secret key pairs $(\texttt{pk}_i, \texttt{sk}_i)$ for n users by running the corresponding KeyGen algorithms. In addition, $\mathcal{S}$ honestly generates credential s_i for n users by running the Enrollment protocol. Eventually, $\mathcal{S}$ sends user's identity/credential $\{ID_i, s_i\}$ and server's identity $ID_{\mathbb{S}}$ to $\mathcal{A}$.
- Training. $\mathcal{A}$ can make the following queries in arbitrary sequence to $\mathcal{S}$.
  - Send: If $\mathcal{A}$ issues a send query in the form of (U, i, m) to simulate a network message for the i-th session of user U, then $\mathcal{S}$ would simulate the reaction of instance oracle Π_U^i upon receiving message m, and return to $\mathcal{A}$ the response that Π_U^i would generate; If $\mathcal{A}$ issues a send query in the form of $(U'$, '*start*'), then $\mathcal{S}$ creates a new instance oracle $\Pi_{U'}^i$ and returns to $\mathcal{A}$ the first protocol message.

[4] This is a modified version of user authenticity model [15] in the remote user authentication setting.

  - Secret Key Reveal: If $\mathcal{A}$ issues a secret key reveal (or corrupt, for short) query to user i, then $\mathcal{S}$ will return the secret key $\mathtt{sk}_i$ to $\mathcal{A}$.
  - Master Secret Key Reveal: If $\mathcal{A}$ issues a master secret key reveal query to $\mathbb{S}$, then $\mathcal{S}$ returns the master secret key $\mathtt{msk}$ to $\mathcal{A}$.
  - State Reveal: If $\mathcal{A}$ issues a state reveal query to (possibly unaccepted) instance oracle $\Pi^j_{U_i}$ ($j \neq i$), then $\mathcal{S}$ will return all internal state values contained in $\Pi^j_{U_i}$ at the moment the query is asked.
- Challenge. $\mathcal{A}$ wins the game if all of the following conditions hold.
  1. $\mathbb{S}$ accept user i; It implies $\mathsf{pid}^s_{\mathbb{S}}$ and $\mathsf{sid}^s_{\mathbb{S}}$ exist.
  2. $\mathcal{A}$ did *not* issue Master Secret Key Reveal query to $\mathbb{S}$;
  3. $m \in \mathsf{sid}^s_{\mathbb{S}}$, *but* there exists *no* $\Pi^s_{U_i}$ which has sent m (m denotes the message transcript from user i).

  Note that $\mathcal{A}$ is allowed to reveal all user's secret keys. We define the advantage of an adversary $\mathcal{A}$ in the above game as

$$\mathtt{Adv}_{\mathcal{A}}(\lambda) = |\Pr[\mathcal{A}\ wins]|.$$

Definition 1. *We say a k-RUA protocol has user authenticity if for any PPT $\mathcal{A}$, $\mathtt{Adv}^{k\text{-}RUA}_{\mathcal{A}}(\lambda)$ is a negligible function of the security parameter λ.*

Anonymity. Informally, an adversary (e.g., authentication server) is not allowed to identify who are the authenticated users, with the condition that authorized users authenticate themselves to authentication server within k times. We define a game between an *insider* adversary $\mathcal{A}$ and a simulator $\mathcal{S}$ below.

- Setup: $\mathcal{S}$ generates a master public/secret key pair $(\mathtt{mpk}, \mathtt{msk})$ for authentication server $\mathbb{S}$ and public/secret key pairs $(\mathtt{pk}_i, \mathtt{sk}_i)$ for n users by running the corresponding KeyGen algorithms. In addition, $\mathcal{S}$ honestly generates a k-size set of credentials $\{s_i\}$ for each user by running the Enrollment protocol. Eventually, $\mathcal{S}$ sends user's identities/credential sets $\{ID_i, s_i\}$ and server's master public/secret key pairs $(\mathtt{mpk}, \mathtt{msk})$ to $\mathcal{A}$. $\mathcal{S}$ also tosses a random coin b which will be used later in the game.
- Training: $\mathcal{A}$ interacts with all users via a set of oracle queries (as defined in the user authenticity model). Eventually, $\mathcal{A}$ outputs two new distinct users (ID_0, ID_1), while $\mathcal{S}$ generates two credential sets $\{s_0\}, \{s_1\}$ for users (ID_0, ID_1) by running the Enrollment protocol.
- Challenge: $\mathcal{A}$ is given one of challenge credential sets $\{s_b\}$, and $\mathcal{A}$ continues to interact with all users (include two new users ID_0, ID_1) via all oracle queries until it terminates and outputs bit b'.

  Note that $\mathcal{A}$ is allowed to activate at most k sessions for ID_0 or ID_1 during Challenge stage, and $\mathcal{A}$ is *not* allowed to reveal the secret keys of ID_0 and ID_1. We define the advantage of $\mathcal{A}$ in the above game as

$$\mathtt{Adv}_{\mathcal{A}}(\lambda) = |\Pr[\mathcal{S} \rightarrow 1] - 1/2|.$$

Definition 2. *We say a k-RUA protocol has anonymity if for any PPT $\mathcal{A}$, $\mathtt{Adv}_{\mathcal{A}}(\lambda)$ is a negligible function of the security parameter λ.*

User Privacy. Informally, an adversary (e.g., non-authorized user) is not allowed to identify who are the session participants. We define a game between an *outsider* adversary $\mathcal{A}$ and a simulator $\mathcal{S}$ below:

- Setup: $\mathcal{S}$ generates a master public/secret key pair $(\texttt{mpk}, \texttt{msk})$ for $\mathbb{S}$ and public/secret key pairs $(\texttt{pk}_i, \texttt{sk}_i)$ for n users by running the corresponding KeyGen algorithms. In addition, $\mathcal{S}$ honestly generates credential s_i for each user by running the Enrollment protocol. Eventually, $\mathcal{S}$ sends user's identity/credential $\{ID_i, s_i\}$ and server's identity $ID_{\mathbb{S}}$ to $\mathcal{A}$. $\mathcal{S}$ also tosses a random coin b which will be used later in the game. We denote the original n users set as $\mathcal{U}$.
- Training: $\mathcal{A}$ is allowed to issue Send, State Reveal queries and at most n-2 Secret Key Reveal queries to $\mathcal{S}$. In particular, $\mathcal{A}$ is *not* allowed to issue Master Secret Key Reveal query to $\mathbb{S}$. We denote the honest (i.e., uncorrupted) user set as $\mathcal{U}'$.
- Challenge: $\mathcal{S}$ randomly selects two users $ID_0, ID_1 \in \mathcal{U}'$ as challenge candidates, and $\mathcal{S}$ removes them from $\mathcal{U}'$ and simulates ID_b^* by either $ID_b^* = ID_1$ if $b = 1$ or $ID_b^* = ID_0$ if $b = 0$.

  Let authentication server $\mathbb{S}$ interact with user ID_b^*. $\mathcal{A}$ can access all the communication transcripts among them.

$$\mathbb{S} \leftrightarrow ID_b^* = \begin{cases} ID_1 & b = 1 \\ ID_0 & b = 0 \end{cases}$$

  Finally, $\mathcal{A}$ outputs b' as its guess for b. If $b' = b$, then $\mathcal{S}$ outputs 1; otherwise, $\mathcal{S}$ outputs 0.

  We define the advantage of $\mathcal{A}$ in the above game as

$$\texttt{Adv}_{\mathcal{A}}(\lambda) = \Pr[\mathcal{S} \to 1] - 1/2.$$

Definition 3. *We say a k-RUA protocol has anonymity if for any PPT $\mathcal{A}$, $\texttt{Adv}_{\mathcal{A}}(\lambda)$ is a negligible function of the security parameter λ.*

3 Security Risks of E-coupon Systems [10, 11]

We notice that the privacy-preserving e-coupon systems in [10, 11] include two important primitives: a new blind signature scheme and an existing oblivious transfer scheme [7]. The blind signature aims to achieve user's anonymity (i.e., untraceability) with respect to service provider, which is the *target* of subsequent attacks. That is, the dishonest users may misuse a valid coupon and successfully avoid the Reveal algorithm. To show the potential security risks of [10] and its extension [11], we just review the extended e-coupon system [11]. Note that the detailed description of extended e-coupon system is referred to [11], and the notation below will mostly *follow* the notation in [10, 11].

Concrete Attacks. By summarising the security risks in [11], we classify two types of adversaries. The goal for both of them is to avoid their Reveal algorithm and misuse a valid coupon. Below we present the detailed attacks respectively.

- Type one. The target is user's secret key. In the issue stage, user receives values $(\delta_1 = pk_{\mathcal{U}}^{k'}, \delta_2 = g^{k'})$ from $\mathbb{S}$. However, a dishonest user $\mathcal{U}$ can replace her secret key x to x' and ask $\mathbb{S}$ to blindly sign it. Specifically, a dishonest user computes $\alpha = (g^{x'y})^{x_1}, \beta = (g^{x'})^{x_1}, \lambda = g^{x_1}$, and $m = \mathtt{H}_1(\alpha, \beta, \lambda), r = m \cdot \beta^a \cdot \delta_1'^{b \cdot x_1/a}$ where $\delta_1' = g^{k \cdot x'}$. Eventually, user stores $(\alpha, \beta, \lambda, r, s)$ as a valid coupon after interaction with $\mathbb{S}$. Note that α, β, r are generated using the new secret key x'.
  Notice that a dishonest user is allowed to modify the value δ_1 (to δ_1') and pass the verification of blinded signature successfully: $\mathtt{H}_1(\alpha, \beta, \lambda) \overset{?}{=} \beta^{-s} \cdot \alpha^{\mathtt{H}_2(\mathtt{H}_1(\alpha,\beta,\lambda),r)} \cdot r$. In the reveal stage, the secret value x_1 will be revealed with regard to a misbehaving user. However, $\mathbb{S}$ could not determine the identity of dishonest user in its database since $pk_{\mathcal{U}}^{x_1} \neq \beta (= g^{x_1 \cdot x'})$.
- Type two. The target is the chosen randomness. In the issue stage, a dishonest user computes $\lambda = g^{x_1'}$ (rather than $\lambda = g^{x_1}$) using a different randomness and generates other parameters honestly using the randomness x_1. In the reveal stage, a secret value x_1' will be revealed with regard to a misbehaving user $\mathcal{U}$. However, $\mathbb{S}$ could not determine the identity of dishonest user in its database since $pk_{\mathcal{U}}^{x_1'} \neq \beta (= g^{x_1 \cdot x})$. Note that the randomness in λ is different from the randomness in α, β, r.
  Same attack can be applied to [10]. If a dishonest user redeems a coupon twice, $(R_1 = x_1' + c_1 \cdot x_1 \cdot x, R_2 = x_1' + c_2 \cdot x_1 \cdot x)$, then $\mathbb{S}$ is not able to obtain the secret key x (what $\mathbb{S}$ can obtain is a value $x_1 \cdot x$).

4 Proposed Construction

A user obtains her credential after interacting with an authentication server $\mathbb{S}$ during Enrollment stage. Later, $\mathbb{S}$ acknowledges an authorized user's authenticity during Authentication stage if and only if the user authenticates with a valid credential. In particular, $\mathbb{S}$ is able to *link* the authorized user's credential with commitments at most k times. If an authorized user authenticates herself to $\mathbb{S}$ for k+1 times, then $\mathbb{S}$ can *identify* the real identity of the user.

- Setup: The authentication server $\mathbb{S}$ takes the security parameter λ as input and outputs the master secret key $\mathtt{msk} = (y, e, f)$ and the master public key $\mathtt{mpk} = (g^y, g_1 = g^e, g_2 = g^f, g_1^y, g_2^y)$. $\mathbb{S}$ also generates the hash functions $\mathtt{H}_1 : \{0,1\}^* \to \mathbb{G}, \mathtt{H}_2 : \mathbb{G} \to \mathbb{Z}_q$. $\mathbb{S}$ chooses a public key encryption (PKE) scheme (e.g., [8]) for the system.
- KeyGen: The user i chooses the secret key $\mathtt{sk}_i = x \in \mathbb{Z}_q$ and computes the public key $\mathtt{pk}_i = g_1^x$.
- Enrollment: The user i and the authentication server $\mathbb{S}$ interact with each other as described below
  - Upon receiving a request from the user i, $\mathbb{S}$ chooses a random element $\mathcal{K} \in_R \mathbb{Z}_q$, computes $(\delta_1 = g^{\mathcal{K}}, \delta_2 = (g_1^x \cdot g_2)^{\mathcal{K}})$ and sends them to the user i;
  - The user i chooses $x_1, a, b \in_R \mathbb{Z}_q$ and computes $\alpha = (g_1^x \cdot g_2)^{y \cdot x_1}, \beta = (g_1^x \cdot g_2)^{x_1}$. Then, the user i computes $m = \mathtt{H}_1(\alpha || \beta), r = m \cdot \beta^a \cdot \delta_2^{b \cdot x_1}, m' = \mathtt{H}_2(m||r)/b$ and sends m' to $\mathbb{S}$;

  - $\mathbb{S}$ computes the *blinded* signature $s' = \mathcal{K} + y \cdot m'$ and sends it to the user i;
  - The user i verifies whether $g^{s'} \overset{?}{=} g^{y \cdot m'} \cdot \delta_1$. If verification fails, it outputs $\texttt{abort}$; otherwise, the user computes $s = s' \cdot b + a$ and stores (α, β, r, s) as a valid credential.

- $\mathsf{Authentication}$: The authorized user i and the authentication server $\mathbb{S}$ interact with each other as described below

  - The user i computes two k-size sets of commitments $(S_1, S_2, \cdots, S_k) = (g_1^{x \cdot s_1}, g_1^{x \cdot s_2}, \cdots, g_1^{x \cdot s_k})$ and $(\overline{S_1}, \overline{S_2}, \cdots, \overline{S_k}) = (g_2^{s_1}, g_2^{s_2}, \cdots, g_2^{s_k})$, where $s_i \in \mathbb{Z}_q$ for each $1 \leq i \leq k$;
  - The user i generates the ciphertext $C_i = \mathsf{Enc}_{\texttt{mpk}}(\{S_i, \overline{S_i}\})$ and sends it to the authentication server $\mathbb{S}$ as an authentication request;
  - Upon receiving a request from the user i, $\mathbb{S}$ chooses a challenge nonce c_i and sends it to the user i;
  - The user i computes $R_1 = x_1 + s_1 \cdot c_i + s_2 \cdot c_i^2 + \cdots s_k \cdot c_i^k, R_2 = x \cdot R_1$ and sends message $m_i = (R_1, R_2, \alpha, \beta, r, s)$ to $\mathbb{S}$;
  - $\mathbb{S}$ checks whether $\mathtt{H}_1(\alpha || \beta) \overset{?}{=} \beta^{-s} \cdot \alpha^{\mathtt{H}_2(\mathtt{H}_1(\alpha||\beta)||r)} \cdot r$ and $g_1^{R_2} \cdot g_2^{R_1} \overset{?}{=} \beta \cdot S_1^{c_i} \cdot S_2^{c_i^2} \cdots S_k^{c_i^k} \cdot \overline{S_1}^{c_i} \cdot \overline{S_2}^{c_i^2} \cdots \overline{S_k}^{c_i^k}$. If either of them fails, it outputs $\texttt{abort}$; otherwise, it outputs accept.

- Trace: We assume that a specific credential (α, β, r, s) is used by a dishonest user for k+1 times. Then, $\mathbb{S}$ gets $k + 1$ shares about the secret x_1 and $x_1 \cdot x$, respectively. Once $\mathbb{S}$ obtains the values of x_1 and $x_1 \cdot x$, $\mathbb{S}$ can successfully compute the user's secret key x.

4.1 Security Analysis

Theorem 4. *The proposed k-RUA achieves user authenticity if the OMDL assumption [4] holds over the underlying group* $\mathbb{G}$*.*

Due to the page limit, the detailed security proof and the subsequent proofs are deferred to the full version of this work.

Theorem 5. *The proposed k-RUA achieves anonymity if the DDH assumption [8] holds over the underlying group* $\mathbb{G}$*.*

Theorem 6. *The proposed k-RUA achieves user privacy if the underlying public key encryption scheme [8] is IND-CCA secure.*

Acknowledgements

The work is supported by the Singapore National Research Foundation under NCR Award Number NCR2016NCR-NCR002-022. It is also supported by AXA Research Fund.

5 Conclusion

In this work, we have proposed a generic construction of anonymous remote user authentication with k-times untraceability. We have also defined the formal security models to achieve certain security requirements that include user authenticity, anonymity and user privacy. We leave the construction of anonymous and traceable remote user authentication with designated verifier (where authorized users can be authenticated by a designated authentication server only) as a future work.

References

1. F. Armknecht, H. Löhr, M. Manulis, A.-R. Sadeghi, et al. Secure multi-coupons for federated environments: privacy-preserving and customer-friendly. In *IPSEC 2008*, pages 29–44. Springer, 2008.
2. G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *CRYPTO 2000*, pages 255–270. Springer, 2000.
3. M. H. Au, W. Susilo, Y. Mu, and S. S. M. Chow. Constant-size dynamic k -times anonymous authentication. *IEEE Systems Journal*, 7(2):249–261, 2013.
4. M. Bellare, C. Namprempre, D. Pointcheval, and M. Semanko. The one-more-rsa-inversion problems and the security of chaum's blind signature scheme. *Journal of Cryptology*, 16(3), 2003.
5. S. Canard, A. Gouget, and E. Hufschmitt. A handy multi-coupon system. In *ACNS 2006*, pages 66–81. Springer, 2006.
6. L. Chen, M. Enzmann, A.-R. Sadeghi, M. Schneider, and M. Steiner. A privacy-protecting coupon system. In *FC 2005*, pages 93–108. Springer, 2005.
7. C.-K. Chu and W.-G. Tzeng. Efficient k-out-of-n oblivious transfer schemes with adaptive and non-adaptive queries. In *PKC 2005*, pages 172–183. Springer, 2005.
8. R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *CRYPTO '98*, pages 13–25, 1998.
9. M. F. Hinarejos, A.-P. Isern-Deyà, J.-L. Ferrer-Gomila, and L. Huguet-Rotger. Deployment and performance evaluation of mobile multicoupon solutions. *International Journal of Information Security*, pages 1–24, 2018.
10. W. Liu, Y. Mu, and G. Yang. An efficient privacy-preserving e-coupon system. In *Inscrypt 2014*, pages 3–15. Springer, 2014.
11. W. Liu, Y. Mu, G. Yang, and Y. Yu. Efficient e-coupon systems with strong user privacy. *Telecommunication Systems*, 64(4):695–708, 2017.
12. L. Nguyen. Privacy-protecting coupon system revisited. In *FC 2006*, pages 266–280. Springer, 2006.
13. L. Nguyen and R. Safavi-Naini. Dynamic k-times anonymous authentication. In *ACNS 2005*, pages 318–333, 2005.
14. I. Teranishi, J. Furukawa, and K. Sako. k-times anonymous authentication (extended abstract). In *ASIACRYPT 2004*, pages 308–322, 2004.
15. Y. Tian, G. Yang, Y. Mu, K. Liang, and Y. Yu. One-round attribute-based key exchange in the multi-party setting. In *ProvSec 2016*, pages 227–243. Springer, 2016.
16. F. Zhang and K. Kim. ID-based blind signature and ring signature from pairings. In *ASIACRYPT 2002*, pages 533–547. Springer, 2002.