2004－ＣＳＥＣ－26　(52)
2004／7／21

部分的なlinkabilityを付加したRefreshable Tokens

繁富　利恵[†]　大塚　玲[††]　KeithMartin[†††]　今井　秀樹[†]

† 東京大学生産技術研究所 〒153-8505 東京都目黒区駒場 4-6-1
†† 情報処理推進機構 〒113-6591 東京都文京区本駒込 2-28-8
††† Royal Holloway, University of London
E-mail: †sigetomi@imailab.iis.u-tokyo.ac.jp, ††a-otsuka@ipa.go.jp, †††keith.martin@rhul.ac.uk, ††††imai@iis.u-tokyo.ac.jp

あらまし　現状のサービスにおいて、サービス提供者における利用履歴の収集は、よりきめ細やかなサービスを行う上においては、非常に重要な位置を占める。しかし、現状のプライバシ保護の匿名性における研究は、ユーザのlinkabilityを持たない匿名性のみを含んだものが多く、サービス提供者における情報収集を加味している研究が少ない。そこで本稿では、ユーザが認めた場合にのみ、サービス提供者が、匿名性を保ったままlinkabilityをつけることのできる方式について提案を行う。

キーワード　電子マネー, プライバシ保護, 匿名認証

Refrehshable Tokens with Optional Linkability

Rie SHIGETOMI[†], Akira OTSUKA[††], Keith MARTIN[†††], and Hideki IMAI[†]

† Institute of Industrial Science, University of Tokyo　4-6-1 Komaba, Meguro-ku, Tokyo 153-8505 JAPAN
†† Information-technology Promotion Agency, Japan　2-28-8, Hon-Komagome, Bunkyo-ku, Tokyo, 113-6591, JAPAN
††† Royal Holloway, University of London　Egham Hill, Egham, Surrey, TW20 0EX, United Kingdom
E-mail: †sigetomi@imailab.iis.u-tokyo.ac.jp, ††a-otsuka@ipa.go.jp, †††keith.martin@rhul.ac.uk, ††††imai@iis.u-tokyo.ac.jp

Abstract　As more services are provided digitally and digital service providers collect more information about users, potential privacy problems are on the increase. This is particularly true when users have to present some form of electronic token in exchange for a digital service. One way to address privacy problems is to provide digital services anonymously. The problem with many previous anonymous token services is that once a token is issued, for a particular value, it cannot be modified in any way other than to decrease the amount of credit associated with the token as it is spent. In this paper we propose a system for "refreshing" tokens anonymously, which allows the holder of a token to request that the issuing organization reissues a modified token in an anonymous way. The new (refreshed) token includes the same embedded user identification as the original token. Unlike most previous schemes, this allows anonymous tokens to be used for applications such as foreign exchange, where tokens may need to be exchanged for refreshed tokens issued in a different monetary value. This scheme also allows anybody to check the validity of the token (not just the token issuer). We also demonstrate how refreshable tokens can be optionally linked, without compromising the anonymity of the owner.

Key words　Electronic cash, Privacy Protection, Anonymous Authentication

1. Introduction

More and more consideration has been shown to privacy problems as the use of digitized information and services has become more and more of the part of our everyday life [6] [7]. To address this problem, each user (customer) should be permitted to control the accumulation of digital records relating to their digital activities. While legislation and privacy policies go a long way towards providing a sympathetic infrastructure, they can only discourage service providers from accumulating personal data. One way of actually preventing service providers from being able to keep records of user services is to enable service to be provided anonymously. Anonymous authentication has thus received a lot of recent attention in the form of various anonymous credential schemes.

Most credential schemes are based on users obtaining signed credentials from organizations and then later demonstrating possession of these credentials in exchange for services. Such systems are said anonymous when the organization verifying the credential cannot determine the identity of the presenter of the credential. Some authors have consider a fuller notion of anonymity to mean that transactions carried out by the same user cannot be related [2] [3] [4].

In many credential schemes an organization needs to check whether a user has got the right to be provided a service by the organization only once, at the beginning of the transaction. Such tokens are essentially "one-purpose" tokens that do not change in any way during their period of validity. This type of token is enough for many situations, such as a token granting access to an event (such as an online film presentation) or demonstrating the validity of an attribute (such as student status). In other situations, however, this type of token is not sufficient. There are many situations where it might be desirable to be able to "refresh" a previously issued token:

- **Renewable credit:** In certain applications a spent token is automatically renewable. For example, in a loan service such as an electronic library it may be the case that once a book has been returned the token used to borrow it should be renewed. In this type of application a user might want to use digital tokens to anonymously borrow books, while the library may want to restrict the number of books out on loan (and hence tokens issued) without compromising the anonymity of the user.
- **Exchangeable value:** A user may wish to anonymously change the representation of value contained in some digital tokens. A foreign exchange service, for example, may take digital tokens issued in one monetary system and convert them into digital tokens in another enumeration. This type of application might also arise with prepaid money tokens and store loyalty points.
- **Anonymous change:** A special case of the above example is when an organization receives an excess of tokens in one "currency" from a user who is paying for a service. If the organization wishes to refund the user then they could return the excess tokens, however this would result in the possibility that the organization could note the token details and link the current service provision to a future payment by the same user using the returned tokens.
- **Change of token details:** Values relating to an issued token may be subject to change. For example a token may have an expiry date that, under certain conditions, can be extended. In such cases a user will need to literally "refresh" the token by submitting it to the issuing body and having the current token renewed.
- **Anonymous termination of service:** By requiring periodic refreshing of tokens, it becomes possible for a server to terminate the provision of service to a user anonymously, without revealing the user's identity, simply be refusing to refresh tokens to users who have acted maliciously.

Note that several of the above situations can be addressed simply be issuing new tokens to users. The problem with this solution is of course that most credential schemes can only issue new tokens to identified users. The applications that we have in mind are those where users wish to user the service anonymously, and hence be able to refresh tokens without revealing their identity.

1.0.1 Our Contribution

In this paper we propose a new refreshable token scheme, where each token contains an embedding of the user's identification. This embedded identity is not revealed to anyone (including the issuer) when the token is verified. Further, such tokens can be refreshed anonymously. It is thus possible for an organization to receive a token from an anonymous user and refresh that token in such a way that the user's identification remains embedded in the refreshed token even though the organization does not know who the identity of the user. Any application using this *refreshable anonymous token* scheme operates broadly as follows:

(1) When a user wants to be provided a service from an organization, the organization issues a token to the user.

(2) The user immediately refreshes the token in such a way that the issuing organization can no longer identify the user from the token.

(3) When the user requests an anonymous service from the organization, the user pays for the service by submitting the token to the organization anonymously.

(4) If the user wishes to "refresh" the token for any reason (for any of the reasons discussed previously) then the

user is able to do so anonymously.

(5) If the same token is used twice then the user's identification can be revealed, even if the token has been refreshed.

Our scheme also has the following optional extension. From a privacy perspective, unlinkability of electronic tokens prevents a service provider from linking transactions by the same user and conducting profiling of user data. However it is possible that certain users might choose to allow token spending to be linked, possibly in exchange for discounted service. Our scheme can be modified in such a way that users can optionally choose to make their refreshable tokens linkable, in the sense that the service provider will be able to tell that a refreshed token is related to a previous token, even though they will still not be able to identify which user these tokens are associated with. For this reason we refer to this extension property as *optional linkability.*

We have already suggested "Refreshable" in some papers [13] [14] [12]. This paper show the new definition of "Refreshable" and the balance of "Full Anonymity" and "Optional Linkability".

2. Preliminaries

In this section we define the Vector-DDH Problem [11], which is equivalent to the well-known Decision Diffie-Hellman (DDH) problem. The Vector-DDH problem is a multiple term extension of the DDH problem. This result is extensively used in this paper to blind a vector of elements in a multiplicative group with large prime order.

The Vector-DDH Problem has already explained in [14] [13]. Then, we will show the definition of the proposition which is the most important point of "refresh".

[Proposition 1] **(ZKP-EQDH: Zero-knowledge proof of the equality of DH exponent)** Let $G =< g >$ be a cyclic group generated by g of prime order q. Given $(g^a, g^b, g^{a'}, g^{b'})$ as the common inputs to a prover and a verifier, there exists an efficient protocol such that a prover with witness a, b, a', b' can prove to a polynomially bounded verifier that $g^{ab} = g^{a'b'}$ without revealing a, b, a', b' or g^{ab}.

3. Informal Definition

Now, we will explain informal definition of our scheme.

Refreshable Tokens Scheme is a token scheme which is an expansion of electronic cash scheme and the two-party protocol, the user and the organization.

"Refreshable Token Scheme" $RT = (RT_{KeyGeneration}, RT_{Issue}, RT_{Refresh}, RT_{Verify}, RT_{Present}, RT_{Trace})$ consists of six polynomial-time algorithms. We will explain about these algorithms informally. $RT_{KeyGeneration}$ is a key generation algorithm for the organization. RT_{Issue} is a registration algorithm for the organization and one user, that does not have to do with all users for this protocol. Thus, the organization could issue the user's ID each time when the user or the organization promises this user can be a member of this protocol. Also, after RT_{Issue}, the user gets a pseudo tokens t_0, which is not anonymous(,and then should do $RT_{Refresh}$ for getting anonymous token). $RT_{Refresh}$ algorithm is on input a token t_i which is for getting to be provided the service by the organization, and on output a next token t_j. If $t_i = t_0$, and then the organization might not be provided the service because t_0 is not anonymous so a privacy information of the service is unveiled. Thus, this is depends on a contract between the organization and the user's. t_i and t_j are elements of T which is a set of all tokens and no linkability. Also, another outputs of $RT_{Refresh}$, which is t_j, is *transcript* for double-use traceability. $RT_{Present}$ algorithm is that the user finishes the service provided by the organization on outputs *transcript*. RT_{Trace} is for tracing algorithm, if the user used same token twice, then the organization gets two transcripts. RT_{Verify} is a verification algorithm that check the token is valid or not on input t and on outputs 0 for valid and 1 for invalid.

We will explain about the difference between $RT_{Refresh}$ and $RT_{Present}$. That is $RT_{Refresh}$ to continue to provide the service, and $RT_{Present}$ to finish providing the service.

3.1 Requirements

We will show the security requirements of our scheme in the following:

- Full Anonymity: Let be $t_i, t_j \in T$ tokens controlled by the service provider Bob, then it is computational infeasible for the adversary Bob *Ad* to decide without the user Alice's help weather t_i, t_j are related by same person Alice.
- Double-Use Traceability: Let be $t \in T$ a token, it is efficient unveiled the adversary's name $i \in [n]$ to use same t twice: $i = RT_{Trace}(t, transcript_0, transctipt_1)$ even if some adversaries without Bob colludes and correct their all tokens.

We will discuss about the difference between previous scheme and our scheme.

All previous schemes are able to check whether the user has got the right to be provided the service by the organization only at the beginning of transaction. This property is not suitable for some situations. We have already discussed some application for our procedure in section 1. Thus, we will show how this property is possible from technical view.

Our scheme properties are Full Anonimity and Double-use Traceability, then as same as most previous anonymous credential schemes such as electronic cash scheme [1] [5] [8] [9] [10], multi-show credential scheme [2] [3] [4]. The difference is that our main point is the token is able to *refresh* holding with double-use traceability, however, the token holder, that is the user, does not have to show the user's identification

when refreshing the token. In off-line electronic cash scheme, when the bank, that means the organization in our scheme, issues the new coin, the user has to show the user's identification. That is because the bank has to check that the coin is included the user's identification to the coin for double-use traceability. In our scheme, the organization, who is called Bob, could issue new token with the same hidden privacy information as the previous token, hence, the user, who is called Alice, can get the new token related to previous with Bob's help anonymously. However, Bob cannot know the previous token and the new token are related, unlinkability.

4. Definition

In this section, we will notate and define our scheme: "Full-Anonymity", and "Double Use Traceability".

If $k \in \mathbb{N}$ then 1^k denotes the string of k ones. If n is an integer then $[n] = 1, \ldots, n$. If Ad is a randomized algorithm then $[Ad(x, y, \ldots)]$ denotes the set of all points having positive probability of being output by Ad on inputs $x, y, \ldots$ and by $z \leftarrow Ad(x, u, \ldots)$ the result of running Ad on the same inputs with freshly generated coins.

Also, we will define a function $f : \mathbb{N} \rightarrow \mathbb{N}$ which is polynomially bounded and polynomial-time computable. If $f : \mathcal{A}_x \times \mathcal{B}_y \rightarrow \mathcal{C}_x \times \mathcal{D}_y$, then f is a function that $\mathcal{A}$ denotes x's a set of input, $\mathcal{B}$ denotes y's a set of input, $\mathcal{C}$ denotes x's a set of output, and $\mathcal{D}$ denotes y's a set of output.

"Refreshable Token Scheme" is in its basic from two-party protocol between a user, Alice, and an organization, Bob.

4.1 Security Definition

Let $\mathcal{Y}$ be a set of public key, $\mathcal{S}$ be a set of secret key, T be a set of possible tokens that is message and signature pairs, W be a set of witnesses, and TR be a set or transcripts.

"Refreshable Token Scheme" $RT = (RT_{KeyGeneration}, RT_{Issue}, RT_{Refresh}, RT_{Verify}, RT_{Present}, RT_{Trace})$ consists of six polynomial-time algorithms:

[Definition 1] **(Refreshable Tokens Scheme)**

$RT_{KeyGeneration}$: $1^k \rightarrow \{\mathcal{Y}, \mathcal{S}, [n]\}_{Bob}$ On input a security parameter 1^k, this probabilistic polynomial-time algorithm outputs the initial public key $\mathcal{Y}$ (including all system parameters), the secret key $\mathcal{S}$ for Bob, and the number of members.

RT_{Issue}:$\{\mathcal{Y}, \mathcal{S}\,[n]\}_{Bob} \rightarrow \{(i, T, W)\}_{Alice,Bob}$ A protocol, on input $\mathcal{Y}$, $\mathcal{S}$, $[n]$. Alice's outputs are T which is a set of Tokens, and W which is a set of witness for T.

$RT_{Refresh}$:$\{\mathcal{Y}, (T, W)\}_{Alice} \times \{\mathcal{Y}, \mathcal{S}\}_{Bob} \rightarrow \{T', W'\}_{Alice} \times \{TR\}_{Bob}$ A probabilistic polynomial-time algorithm that takes T which is a set of tokens, and W which is a set of witness for $T[i, k]$, $\mathcal{Y}$ and , $\mathcal{S}$ which is known by only Bob, and outputs (T', W) for a set of new tokens, and TR for Bob which is a set of transcripts of T.

RT_{Verify}:$\{\mathcal{Y}, t \in T\} \rightarrow \{0, 1\}$ A probabilistic polynomial-time algorithm that on inputs $\mathcal{Y}$, $t \in T$ and outputs valid or invalid.

$RT_{Present}$:$\{\mathcal{Y}, (T, W)\}_{Alice} \times \{\mathcal{Y}\}_{Bob} \rightarrow \{TR\}_{Bob}$ A probabilistic polynomial-time algorithm that on inputs $T[i, k]$, $W[i, k]$ and outputs $TR[i]$ for Alice which is a set of transcripts for $T[i, k]$.

RT_{Trace}:$\{t \in T, transctipt_0, transcript_1 \in TR\} \rightarrow i \in [n]$ A polynomial time algorithm that on inputs $(t \in T[i])$ and $(transcript_0, transcript_1 \in TR[i])$ and outputs i.

In using $RT_{Refresh}$ algorithm, inputing data $t \in T$ and outputting data $t' \in T$ are related the same Alice's witness W, on the other hand, Bob cannot see witness information W. These token data t, t' are constructed $(msg, sig), (msg', sig')$ pairs, which msg is a message and sig is a signature for msg. Also, This sig should be blinded because t and t' are unlinkable. Thus, Alice creates by herself msg' using her witness, then Bob signs for msg' blinded. However, before being signed by Bob, Bob has to check whether msg and msg' has been including same witness W. This is because Alice could cheat to Bob that msg' has the witness. Then, after Bob sign msg blindly, Alice gets the new token t'.

This concludes Zero-knowledge proof of the equality of DH exponent1 and Blind signature such as Schnorr signature [1].

We will show the new property for optional linkability in section 6..

4.2 Requirements

We say that t is a true token such that $t = RT_{Refresh}(\mathcal{Y}, \mathcal{S}, t, w)$. We say that $t \in T$ is a valid token with respect to $\mathcal{Y}$ if $RT_{Verify}(\mathcal{Y}, t) = 1$. However, after RT_{Issue}, Alice has got a set of tokens which has no anonymity because only Bob creates by himself the token. Thus Alice has to do $RT_{Refresh}$ procedure to get a new token which has anonymity. Then Alice's anonymity is protected.

The scheme must satisfy the following correctness requirements: For all $k \in \mathbb{N}$, all $(\mathcal{Y} \in [RT_{KeyGeneration}(1^k)])$, all $n \in \mathbb{N}$, all $i \in [n]$, and all $t \in [RT_{Issue}(\mathcal{Y}, \mathcal{S}, i)]$

$$RT_{Verify}(\mathcal{Y}, RT_{Refresh}(\mathcal{Y}, \mathcal{S}, t)) = 1$$

and,

$$RT_{Trace}(RT_{Refresh}(\mathcal{Y}, \mathcal{S}, t), RT_{Refresh}(\mathcal{Y}, \mathcal{S}, t)) = i$$

. (If $RT_{Refresh}$ is substitute for $RT_{Present}$, RT_{Trace} returns i.)

We will discuss about "Full Anonymity" and "Double Use Traceability". Informally, in our protocol, "Full Anonymity" is that it is computationally infeasible for a probabilistic-time adversary to find Alice's identification from any token even if the token has done polynomial-times $RT_{Refresh}$ oracle. On the other hand, "Double Use Traceability" is

$Exp^{anony-b}_{RT,Ad}(k)$

$(\mathcal{S}, \mathcal{Y}) \leftarrow RT_{KeyGeneration}(1^k)$

$(St, t_0, witness_0, t_1, witness_1) \leftarrow Ad^{RT_{Refresh}(\mathcal{Y},\cdots)}(choose, \mathcal{Y}, \mathcal{S})$

$((transcript'_0), (t'_0, witness'_0)) \leftarrow RT_{Refresh}((\mathcal{S}), (t_0, witness_0))$

$((transcript'_1), (t'_1, witness'_1)) \leftarrow RT_{Refresh}((\mathcal{S}), (t_1, witness_1))$

$d \leftarrow Ad^{RT_{Refresh}}(guess, St, transcript'_0, transcript'_1, t'_b)$

If Ad did not query its oracle with t in the guess stage then return d EndIf

Return 0

Fig. 1 Experiments: to define Full Anonymity of Refreshable Tokens Scheme $RT = (RT_{KeyGeneration}, RT_{Issue}, RT_{Refresh}, RT_{Verify}, RT_{Present}, RT_{Trace})$.

that a probabilistic-time adversary is efficiently unveiled by a double-used token even if the token has done polynomial-times $RT_{Refresh}$ oracle.

4.3 Full Anonymity

In our protocol, Full Anonymity is it is computationally infeasible for an adversary Bob who choose two tokens and transcripts to decide whether t engaged b using $RT_{Refresh}$ oracle.

4.3.1 Anonymity Experiments

To use Refreshable Tokens Scheme $RT = (RT_{KeyGeneration}, RT_{Issue}, RT_{Refresh}, RT_{Verify}, RT_{Present}, RT_{Trace})$, adversary Ad and a bit b we associate the first experiments $Exp^{anony-b}_{RT,Ad}(k)$ given in Figure 1. In the *choose* stage, Ad takes as input public key $\mathcal{Y}$ and Bob's secret key $\mathcal{S}$. During this stage, it can also query $RT_{Refresh}$ oracle on a token of Ad's choice, and it is required that at the end of the stage Ad outputs two valid tokens and witness $(t_0, witness_0)$ and $(t_1, witness_1)$, $t_0, t_1 \in T$ and $witness_0, witness_1 \in W$. In the next stage, Ad gets next token and witness $t'_0, witness'_0$ from $t_0, witness_0$ and $t'_1, witness'_1$ from $t_1, witness_1$. The goal is to guess t'_0 or t'_1 which of previous token t_0 or t_1 has refreshed $RT_{Refresh}$.

That means:

- $(\mathcal{S}, \mathcal{Y})$ are sellected by $RT_{KeyGeneration}$ according to its internal coin flip.
- An Adversary Ad chooses two tokens t_0 and t_1 and their witnesses w_0 and w_1 interacting with RT oracles.
- Two tokens t_0, and t_1 are refreshed $RT_{Refresh}$.
- Ad guess d after seeing t_0 and t_1 by Ad.

We denote by

$$Adv^{anon}_{RT,Ad} = Pr[Exp^{anon-1}_{RT,Ad}(\mathcal{Y}) = 1] - Pr[Exp^{anon-0}_{RT,Ad}(\mathcal{Y}) = 1]$$

the advantage of adversary Ad in breaking the Full Anonymity of RT.

4.4 Double Use Traceability

Double-Use Traceable is it is efficiency unveiled to use same t twice even if some adversary without Bob colludes and correct their all tokens.

4.5 Traceability Experiments

We will define Full Traceability using Traceable Experiments of Figure 2. Here, Ad On input $\mathcal{Y}$ which is a public key, Ad starts its attack by adaptively corrupting a set C of members. The tokens of the members that are corrupted and their number is entirely up to it. At the end of the choose stage, the set C contains the identities of the corrupted members. St contains all tokens of C. In the guess stage, the adversary attempts to produce (t, w), and we say it wins, if t is a valid token, but RT_{Trace} algorithm return some valid user identity i such that $i \notin C$. Otherwise, the experiment return 0.

That means:

- $(\mathcal{S}, \mathcal{Y})$ are sellected by $RT_{KeyGeneration}$ according to its internal coin flip.
- Let C be a set of Ad's members, then adding new member j to C for joining polynomial-time numbers of Adversary.
- Let K be a set of tokens by j, then adding K to St as many adversarys as possible.
- An Adversary Ad chooses new token (t', w') interacting with RT oracles.
- RT_{Trace} is impossible to truely return Ad's name using wrong (t', w').

We define the advantage of adversary Ad in defeating Double Use Traceability of the RT by:

$$Adv^{trace}_{RT,Ad}(k) = Pr[Exp^{trace}_{RT,Ad}(k) = 1]$$

and say that RT is Double Use Traceable if for any polynomial-time adversary Ad, the two-argument function $Adv^{trace}_{RT,Ad}(.,.)$ is negligible.

$Exp^{trace}_{RT,Ad}(k)$

$(\mathcal{S}, \mathcal{Y}) \leftarrow RT_{KeyGeneration}(1^k)$

$St \leftarrow (\mathcal{Y}); C \leftarrow 0; K \leftarrow \epsilon; Cont \leftarrow true;$

While$(Cont = true)$do

$(Cont, St, j.l) \leftarrow Ad^{RT_{Refresh}(\mathcal{Y},\mathcal{S},.)}(choose, St, K)$

If $Cont = true$ then

$((t, w), (.)) \leftarrow RT_{Issue}((j), \mathcal{S})$

$\mathcal{C} \leftarrow \mathcal{C} \cup j$

$K \leftarrow (t', w')$EndIf

EndWhile

$(t', w') \leftarrow Ad^{RT_{Refresh}(\mathcal{Y},\mathcal{S},.)}(guess, St)$

If$RT_{Verify}(\mathcal{Y}, t) =$ 0then return 0;

$((.), (transcrypt_0)) = RT_{Refresh}((\mathcal{S}), (t', w'))$

$((.), (transcrypt_1)) = RT_{Refresh}((\mathcal{S}), (t', w'))$

If there exists$i \in [n]$such that the following are true then return 1 else return 0 :

1. $RT_{Trace}(t\, transcript_0, transcript_1) = i \notin \mathcal{C}$
2. $(t\, l')$was not queried by Ad to its oracle

Fig. 2 Experiments: to define Double Use Traceability of Refreshable Tokens Scheme $RT = (RT_{KeyGeneration}, RT_{Issue}, RT_{Refresh}, RT_{Verify}, RT_{Present}, RT_{Trace})$.

5. Main Construction

We will show the main points of construction of our protocol because we have already showed detail construction in [14] [13].

$RT_{KeyGeneration}$

$g, p, q \leftarrow < 1^k >$

Choose x using q

$[n] := < p >$

$z \leftarrow < g, x >$

$\mathcal{Y} := < g, p, q, z >$

$\mathcal{S} := < x >$

return$(\mathcal{Y}, \mathcal{S}, [n])$

RT_{Issue}

$i \leftarrow [n]$

Parse (g, p, q, z) as $\mathcal{Y}$

$\mathcal{Y}, \mathcal{S} \leftarrow$ parameter(1^k)

return$(i, T[i, 0], W[i, 0])$

RT_{Verify}

Parse(g, p, q, z) as $\mathcal{Y}$

Parse(msg, σ) as t

Verify$(0, 1) \leftarrow (msg, \sigma, g, z)$

return(0 or 1)

$RT_{Refresh}$

Parse(g, p, q, z) as $\mathcal{Y}$

If$(1 \leftarrow RT_{Verify}(T, W, TR)))$ then

$TR \leftarrow (T, W)$

$i, W' := < W >$

$msg \leftarrow T, g, p, q, z, i$

Check (msg, T) has same i

Parse x as $\mathcal{S}$

$\sigma \leftarrow msg, \mathcal{S}$

$T' \leftarrow (msg, \sigma)$

$W' \leftarrow (msg, \sigma)$

$TR' \leftarrow (T, W)$

return(T', W', TR')

$RT_{Present}$

Parse(g, p, q, z) as $\mathcal{Y}$

If$(1 \leftarrow RT_{Verify}(T, W, TR)))$ then

$TR \leftarrow (T, W)$

return(TR)

RT_{Trace}

$i \leftarrow (transcript_0, transcript_1, t)$

return(i)

6. Optional Linkability

In this section we briefly describe an extension to our basic refreshable token scheme that allows a user to optionally surrender the unlinkability of their token spending.

Service providers collect usage record so that they can profile the user needs and monitor the service that they provide. There are many applications where users are happy to surrender some degree of anonymity in order to gain a perceived enhancement of service. One obvious example is points schemes based on store loyalty cards, where users agree to allow their profile to be developed through linked transactions in order to qualify for discounts and access to special offers.

The following enhancement to our refreshable token scheme allows a user to choose to permit linking of transactions, while maintaining a degree of anonymity (meaning in this case that the organization can link token spends without knowing the identity of the user whose token spends are being linked). This property can be achieved while continuing to offer double spending traceability. As this property is "opt in" from a user perspective, we refer to it as *optional linkability* and informally define it as follows:

- Optional-Linkability: If $t_i, t_j \in T$ are tokens that are embedded with the same user identity then it is efficient for Bob to establish that these tokens ontain the same user identity, but Bob is unable to decide the identity of the user embedded in the two tokens. Further, if Alice agrees to link her tokens, then Alice is unable to refresh tokens that are unlinkable without Bob's agreement.

6.1 Solution for this protocol

Optional linkability is easily introduced into the protocol $RT_{refresh}$ in the following way:

- If Alice and Bob decides "linked", then Bob creates one-time secret key and public key pair $(x, z = g^x)$ and shows z to Alice.
- When Alice creates her token t, she has to use z.
- Bob is able to check Alice using z, and sign to t.

If Alice's transaction t_0, that is the same meaning of using

$RT_{OPRefresh}$

Parse$(g, p, q) as \mathcal{Y}$

If$(1 \leftarrow RT_{Verify}(T, W, TR)))$ then

$x', z' = g^{x'} \leftarrow \mathcal{Y}$

$\mathcal{S} \leftarrow x'$

$TR \leftarrow (T, W)$

$i, W' :=< W >$

$msg \leftarrow T, g, p, q, z', i$

Check (msg, T) hassame i

Parse x' *as* $\mathcal{S}$

$\sigma \leftarrow msg, \mathcal{S}$

$T' \leftarrow (msg, \sigma)$

$W' \leftarrow (msg, \sigma)$

$TR' \leftarrow (T, W)$

return(T', W', TR')

Fig. 3 Construction: Optional Linkability

t_0 token, and t_1 are linked, then Bob create key pair: secret key and public key at the beggining of the transaction. That means, before $RT_{Refresh}$, Bob creates x_0 and $z_0 = g^{x_0}$ and Bob and Alice do $RT_Refresh$ using x_0, z_0.

When Bob needs to check which tokens are linked, Bob is able to search using $(x_0, z_0), \ldots$.

We will discuss the balance between Full Anonymity and Optional Linkability.

Full Anonymity is that let be $t_i, t_j \in T$ tokens which are related by same user, then it is computertional infeasible for the adversary Bob Ad to decide without the user Alice's help. Therefore, in the definition of Full Anonymity, if Alice help Bob for link of two tokens by Alice's transaction, it is possible to link Alice's tokens. Then, "Optional Linkability" means that if Alice decides to show the linkable to Bob, that means Alice help the linkable to Bob, then Bob can link these tokens.

7. Conclusion

In this paper, we suggested the definition of "Refreshable Tokens" and showed one construction for this definition. Nowadays, the service provider needs more consideration of understanding of consumer's tastes for providing the service because of diversity in individual value. However, to collect and acumulate the consumer's tastes are threaten to invade the provacy of consumer. These ballance have got a lot of problems. Our solution is just one solution so we needs more another solution depends on situations.

References

[1] S. Brands. Untraceable off-line cash in wallets with observers. In *Proc. of CRYPTO '93*, pages pp 302–318.

Springer, 1994.

[2] J. Camenisch and I. Damgard. Verifiable encryption and applications to group signatures and signature sharing. In *Tech. Rep. RS-98-32. Department of Computer Science, University of Aarhus.* Brics, Dec. 1998.

[3] J. Camenisch and A. Lysyanskaya. Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation. In *EUROCRYPT2001*, volume 1976 of *Lecture Notes in Computer Science*, pages pages 331–345. Springer, 2001.

[4] J. Camenisch and M. Michels. A group signature scheme based on an rsa-varian. In *Tech. Rep. RS-98-27. Department of Computer Science.* Brics, University of Aarhus, 1998.

[5] D. Chaum. Blind signatures for untraceable payments. In *Proc. of CRYPTO'82*, Lecture Notes in Computer Science, pages pp.199–203. Springer, 1982.

[6] D. Chaum. Group signatures. In *Advances in Cryptology - EUROCRYPT '91*, volume 547 of *Lecture Notes in Computer Science*, pages pp.257–265. Springer, 1991.

[7] D. Chaum. Security without identification: Card computers to make big brother obsolete. In *Communications of the ACM, v. 28, n. 10*, pages pp. 1030–1044. ACM, Oct 1985.

[8] D. Chaum, A. Fiat, and M. Naor. Untraceable electronic cash. In *Proc. of CRYPTO '88*, Lecture Notes in Computer Science, pages pp. 319–327. Springer, 1988.

[9] D. Chaum and T. Pedersen. Wallet databases with observers. In *Proc. of CRYPTO '92*, Lecture Notes in Computer Science, pages pp.89–105. Springer, 1993.

[10] D. Chaum, R. Rivest, and A. T. Sherman. Blind signatures for untraceable payments. In *Advances in Cryptology Proceedings of Crypto 82, Plemum*, Lecture Notes in Computer Science, pages pp. 199–203. Springer, 1983.

[11] A. Otsuka, G. Hanaoka, J. Shikata, and H. Imai. An unconditionally secure electronic cash scheme with conputational untraceable. In *IEICE, The Institute of Electronics, Information and Communication Engineers*, pages pp. 140–148, 2002.

[12] R. Shigetomi, A. Otsuka, and H. Imai. Anonymous authentication scheme for xml security standard with refreshable tokens. In *ACM Workshop on XML Security*, pages 86–93, October, 2003.

[13] R. Shigetomi, A. Otsuka, T. Ogawa, and H. Imai. Refreshability of tokens. In *情報理論とその応用学会シンポジウム 2002*, pages 43–46, 2002.

[14] R. Shigetomi, A. Otsuka, T. Ogawa, and H. Imai. Refreshable tokens and its application to anonymous loan. In *Symposium on Cryptography and Information Security (SCIS) 2003 浜松*, pages 5–10, 2003.