Times Limited Accountable Anonymous Online Submission Control System from Single-Verifier k-times Group Signature

Xingwen Zhao and Fangguo Zhang
School of Information Science and Technology, Sun Yat-Sen University, Guangzhou 510275, P.R.China
Guangdong Key Laboratory of Information Security Technology, Guangzhou 510275, P.R.China
E-mail: sevenzhao@hotmail.com, isszhfg@mail.sysu.edu.cn

Keywords: privacy, anonymous authentication, accountability, group signature

Received: May 26, 2010

People in authority may want to submit some messages anonymously on a famous website, while the maintainers may want to limit the times each person can submit messages on the website so as to save the storage space. More over, when people abuse the system, the maintainers want to find ways to identify their identities. To realize such a system, what we need are some methods that can protect users' privacy, control their access times, and at the same time can identify malicious users when abuses are found. Current signature schemes or credential systems cannot fully achieve above purpose. A single-verifier k-times group signature scheme is proposed, adding times limited property to the group signature scheme. It allows a user to issue group signatures to the only verifier up to k_i times for period T_i. We use online tracing method to restrict each user to k_i signatures strictly, and use the tracing ability of group signature to identify those who abuse the system. Based on it, we can construct times limited accountable anonymous online submission control system for websites. Within allowed times, people can submit articles anonymously, even website maintainers cannot identify two articles are from the same person. When a person posts more than the allowed times, his submission will be rejected. When abuse is found, website maintainers can send the signature to the corresponding open authority to find out the identity.

Povzetek: Članek predlaga metodo podpisovanja spletnih sporočil, ki zagotavlja anonimnost le pri omejenem številu uporab.

1 Introduction

Nowadays people may want to post articles anonymously. Imaging there is a famous website, and people from several authority organizations are allowed to post articles on it. People read the articles and know they are from a person of the authority organization, but they do not know who indeed posts the messages. Even the website maintainers cannot tell two articles are from the same person. At the same time, the maintainers of the website may want these authorities to be concise to save the storage space, so they may want to limit the times each person can post messages on the website. When a person attempts to post more than allowed times, he will be found immediately and his submission is rejected, while his anonymity is still protected. Moreover, when people abuse the system, the maintainers can find ways to identify those abusers. Can we have some methods to protect users' privacy, control their access times, and at the same time identify malicious users when abuses are found?

Related Works. Group signature [7], ring signature [15] and anonymous credential [6] can be used to protect people's privacy. However, users in these protocols can show their signatures and credentials as many times as they want.

In the linkable ring signature [12], signatures from the same signer can be linked so as that multiple signing behaviors can be controlled. However, the signers are always anonymous and abusers can never be found.

The k-times anonymous authentication (k-TAA)[16, 18, 11, 17] was introduced to protect the privacy while limiting the authentications. Each user can only authenticate anonymously up to k times, with k determined by each application provider (AP) and fixed for all users. When a user authenticates more than k times to an AP, its privacy is compromised. Some articles [14, 13, 1] described dynamic k-TAA, enabling APs to grant or revoke users independently. However, the property that enables AP to grant users may compromise users' privacy in a sense, because AP knows the identity of each granted user. Again, the value k is determined by each AP and fixed for all users. Camenisch et al. [4] brought forward a periodic n-times anonymous authentication scheme. In their scheme, each user can authenticate anonymously up to n times in each period, no matter how many APs there exist. However, in schemes listed above, the authentications are fully anonymous if they are issued within allowed times. If some users abuse the system within allowed times, they cannot be identified and punished. Therefore, the existing variants of k-TAA are not applicable to our case.

Emura et al. [10] presented a selectable k-TAA scheme,

allowing each user has different allowed number of authentication for different AP. In their scheme, the user chooses the allowed number anonymously, and AP decides to accept or reject. However, the computation cost for granting phase is high, which is linear to the allowed number k. The anonymity is weaken since authentications between the same user and the same AP are linkable to AP (AP knows they are from the same user).

Our Contribution. In this paper, a single-verifier k-times group signature scheme is proposed as building block, where all the group signatures are verified by the only verifier, and the signatures from the same person are limited to k_i times during time period T_i times. Based on it, times limited accountable anonymous online submission control system for websites is constructed. Within allowed times, people can post articles anonymously with their signatures, even website maintainers cannot identify two articles are from the same person. When a person posts more than the allowed times, his post will be rejected. When abuse is found, website maintainers can send the signature to open authority of the group to find out the identity.

Paper Outline. The rest of this paper is organized as follows: In Section 2 we introduce some preliminaries. In Section 3, we describe the proposed single-verifier k-times group signature scheme, the building tool for the submission control system. In Section 4, we briefly describe the times limited accountable anonymous online submission control scheme for websites. In Section 5, system attributes and comparison with related previous schemes are presented. Finally, conclusion is given in Section 6.

2 Preliminaries

2.1 Bilinear Pairings and q-SDH Problem

We first review a few concepts related to bilinear pairings. We follow the notation of [3]:

Definition 1 (Bilinear Pairings)**.** *Let $\mathbb{G}$ be a (multiplicative) cyclic group of prime order p and g is a generator of $\mathbb{G}$. A one-way map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a bilinear pairing if the following conditions hold.*

- **Bilinear:** *For all $u, v \in \mathbb{G}$, and $a, b \in \mathbb{Z}_p$, $e(u^a, v^b) = e(u, v)^{ab}$.*
- **Non-degeneracy:** *$e(g, g) \neq 1$, i.e., if g generates $\mathbb{G}$, then $e(g, g)$ generates $\mathbb{G}_T$.*
- **Computability:** *There exists an efficient algorithm for computing $e(u, v)$, $\forall u, v \in \mathbb{G}$.*

The q-SDH problem was introduced by Boneh and Boyen [2] to construct short signatures without random oracles.

Definition 2 (q-SDH Problem)**.** *The q-SDH problem in $(\mathbb{G}_1, \mathbb{G}_2)$ is defined as follows: given a (q+2)-tuple $(g_1, g_2, g_2^{\gamma}, g_2^{(\gamma^2)}, \ldots, g_2^{(\gamma^q)})$ as input, output a pair $(g_1^{\frac{1}{\gamma+x}}, x)$ where $x \in \mathbb{Z}_p^*$. We say that the q-SDH is (q, t, ϵ)-hard if for all t-time adversaries A, we have*

$$Pr\left[A(g_1, g_2, g_2^{\gamma}, g_2^{(\gamma^2)}, \ldots, g_2^{(\gamma^q)}) = (g_1^{\frac{1}{\gamma+x}}, x)\right] < \epsilon.$$

2.2 Proofs of Knowledge of Discrete Logarithms

We will use the notation introduced by Camenisch and Stadler [5] for various proofs of knowledge of discrete logarithms. For instance,

$$PK\{(\alpha, \beta, \gamma) : y = g^{\alpha}h^{\beta} \wedge z = g'^{\alpha}h'^{\gamma}\};$$

is used for proving the knowledge of integers α, β and γ such that $y = g^{\alpha}h^{\beta}$ and $z = g'^{\alpha}h'^{\gamma}$ holds. Here y, g, h, z, g' and h' are elements of some groups $\mathbb{G} =< g >=< h >$ and $\mathbb{G}_T =< g' >=< h' >$.

3 The Building Tool: Single-verifier k-times Group Signature

The building tool is a single-verifier k-times group signature scheme, in which authorized people can issue group signatures up to k_i times for time T_i. Signatures are all verified by a same verifier so that each user are limited to k_i signatures strictly.

3.1 The Model

A single-verifier k-times group signature scheme is similar to group signature scheme, while the signing times are limited during each period. It consists of the following algorithm:

- Key Generation: The algorithm generates the secret key for the group manager, the secret key for the open authority, and the public parameters. There may be several groups.
- Member Joining: User registers with the group manager to join the group. After that, user obtains group member certificate, while group manager obtains user's identification and tracing information, which will be used to identify the abuser.
- Times Announcing: The verifier announces the signing times allowed for each future period. They can be the same or different, depending on the applications.
- Sign: The user signs the message to show that he is a member of a certain group. Each member certificate can be used to sign up to k times during each period.
- Verify: The verifier checks if the signature is valid and it is within the allowed times. If accepted, some tracing information is recorded into a log file. If not, the signature will be rejected.

– Open: When necessary, the open authority finds out user's identification from the signature.

3.2 Security Notions

A single-verifier k-times group signature scheme should fulfill the following security notions:

– Correctness: The signature from an honest user within the allowed times should be accepted by the honest verifier. And the open algorithm should correctly identify the signer.

– Unforgeability: It is computationally impossible to produce a valid signature, without the knowledge of a membership certificate.

– Anonymity: Only the open authority can identify which user provided the signatures.

– Traceability: It must be hard to produce a valid signature such that either the honest open authority is unable to identify the signer, or the open authority believes it has identified the origin but is unable to produce a correct proof of its claim.

– Detectability: Suppose k is the number of times the verifier allows each user to sign during a single period. Detectability means that an adversary, who colludes with w users, is unable to issue more than kw signatures without being detected the same verifier.

3.3 Online k-times limitation

Our idea of online k-times limitation is developed from Damgård et al. [8]. Each user holds a secret key SK which is different from the others. A hash function H_1 maps string str to elements in a group where decisional Diffie-Hellman (DDH) problem is hard. If each user is allowed to sign only once during period T, then the user needs to show $H_1(T)^{SK}$, the commitment to SK, as the tracing tag, along with the proof of knowledge of owning the group member certificate. If $H_1(T)^{SK}$ appears twice, the verifier rejects the signature. If each user is allowed to sign k times during period T, $H_1(T, k_i)^{SK}$ is used, with $k_i = 1, \ldots, k$.

3.4 Single-verifier k-times Group Signature

Our single-verifier k-times group signature is built upon the group signature scheme by Delerablée and Pointcheval [9].

– Key Generation: Selects bilinear pairings $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ as required in Section 2. Randomly selects generator $g_2 \in_R \mathbb{G}_2$, so $g_1 \leftarrow \psi(g_2)$ is the generator of $\mathbb{G}_1$. Randomly selects another generator $h \in_R \mathbb{G}_1$, and $\xi_1, \xi_2 \in_R \mathbb{Z}_p^*$, then calculates $u \in \mathbb{G}_1$, satisfying $u^{\xi_1} = h$ and $u^{\xi_2} = g_1$. ξ_1, ξ_2 are the secret keys for the open authority. Selects $\gamma \in_R \mathbb{Z}_p^*$ as the secret key for the group manager, and calculates $w = g_2^{\gamma}$. Two collision resistant hash functions $H_1 : \{0,1\}^* \to \mathbb{G}_T$, $H_2 : \{0,1\}^* \to \mathbb{Z}_p$ are selected. The public parameters for the group are gpk=($\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$, e, ψ, u, $h = u^{\xi_1}$, $g_1 = u^{\xi_2}$, g_2, $w = g_2^{\gamma}$, H_1, H_2). User generates public key upk and private key usk for itself.

– Member Joining: User interacts with the group manager to join the group. The steps are described as follows.

1. User U selects $y \in_R \mathbb{Z}_p$ to calculate $C = h^y$, and generates non-interactive zero knowledge proof π to prove knowing y. U sends C and π to GM.
2. GM checks if C is ever used by other users before. If used, GM requires U to run the Member Joining algorithm again. If C is never used, GM checks whether π is valid. If invalid, GM rejects the joining.
3. GM selects $x \in_R \mathbb{Z}_p$ and calculates $A = (g_1 C)^{\frac{1}{\gamma+x}}$, $B = e(g_1 C, g_2)/e(A, w)$, $D = e(A, g_2)$. GM generates non-interactive zero knowledge proof V to prove knowing the discrete logarithm of B in basis D. GM sends A and V to U. In fact, B = $e(A^{\gamma+x}, g_2)/e(A, w)$ = $e(A^{\gamma}, g_2)e(A^x, g_2)/e(A, g^{\gamma})$ = $e(A^x, g_2)$ = $e(A, g_2)^x$, which means GM only needs to prove knowing x.
4. After User U obtains A and V, he calculates $B = e(g_1 C, g_2)/e(A, w)$, $D = e(A, g_2)$, checks if V is valid. If valid, U signs A with his key usk to obtain S, and sends S to GM.
5. GM verifies S with upk and A. If valid, GM records (upk,A,x,S), and sends x to U. The joining records are sent to open authority.
6. User obtains x and verifies the following equation

$$e(A, g_2)^x e(A, w) e(h, g_2)^{-y} = e(g_1, g_2).$$

If the equation holds, the user joins the group successfully and the group member certificate is (A, x, y). The equation is expressed as followed:

$$\begin{aligned} & e(A, g_2)^x e(A, w) e(h, g_2)^{-y} \\ = \; & e(A, g_2)^x e(A, g_2^{\gamma}) e(h, g_2)^{-y} \\ = \; & e(A, g_2^{x+\gamma}) e(h, g_2)^{-y} \\ = \; & e((g_1 h^y)^{\frac{1}{x+\gamma}}, g_2^{x+\gamma}) e(h, g_2)^{-y} \\ = \; & e(g_1, g_2) e(h^y, g_2) e(h, g_2)^{-y} \\ = \; & e(g_1, g_2). \end{aligned}$$

– Times Announcing: The verifier publishes a list to show the signing times allowed for each future period. The list is as follows, (T_1,k_1),...,(T_n,k_n). Or the verifier can publish only a k indicating the users can sign messages up to k time during each period.

– Sign: Suppose user U with certificate (A, x, y) wants to sign a message m during period T and each user is allowed to sign k times during period T. Suppose it is the ith ($0 < i \leq k$) time user U shows to the verifier, he behaves as follows. User U calculates h_i=$H_1(T, i)$ with the commitment E_i=h_i^y, and generates a standard non-interactive zero-knowledge proof π_i=$ZKP\{(A, x, y) : E_i$=$h_i^y \wedge e(A, g_2)^x \cdot e(A, w) \cdot e(h, g_2)^{-y}$=$e(g_1, g_2)\}$, which is also the signature on message m. User U sends (i, E_i, π_i) to the verifier. Technical details of zero-knowledge proof are as follows.

  1. User U selects $\alpha, \beta \in_R \mathbb{Z}_p$, calculates $T_1 = u^\alpha$, $T_2 = Ah^\alpha$, $T_3 = u^\beta$, $T_4 = Ag^\beta$.
  2. In order to sign the message m, user U selects r_α,r_β,r_x,r_y,$r_{x\alpha} \in_R \mathbb{Z}_p$, calculates $R_1 = u^{r_\alpha}$, R_2=$e(T_2, g_2)^{r_x} \cdot e(T_2, w)^{\cdot} e(h, g_2)^{-r_{x\alpha}} \cdot e(h, w)^{-r_\alpha} \cdot e(h, g_2)^{-r_y}$, $R_3 = u^{r_\beta}$, $R_4 = h^{r_\alpha} g^{-r_\beta}$, $h_i = H_1(T, i)$, E_i'=$h_i^{r_y}$, and c=$H_2(m, T_1, T_2, T_3, T_4, R_1, R_2, R_3, R_4)$. R_2 can be written as $R_2 = e(A, g_2)^{r_x} \cdot e(A, w)^{\cdot} e(h, g_2)^{\alpha r_x - r_{x\alpha} - r_y} \cdot e(h, w)^{\alpha - r_\alpha}$, so that user U can obtain R_2 with fewer computations, since all these pairings can be pre-computed.
  3. User U calculates $s_\alpha = r_\alpha + c\alpha$, $s_\beta = r_\beta + c\beta$, $s_x = r_x + cx$, $s_y = r_y + cy$, $s_{x\alpha} = r_{x\alpha} + cx\alpha$.
  4. so π_i=$(T_1, T_2, T_3, T_4, E_i', c, s_\alpha, s_\beta, s_x, s_y, s_{x\alpha})$

– Verify: The verifier maintains a tracing log $TLOG_T$ for time period T. On receiving (i, E_i, π_i), the verifier checks if $1 \leq i \leq k$, and makes sure that E_i does not exist in $TLOG_T$. Else, the verifier rejects the execution. If both of them hold, the verifier checks whether the zero-knowledge proof is valid as follows.

  1. The verifier calculates $R_1 = u^{s_\alpha} T_1^{-c}$, $R_3 = u^{s_\beta} T_3^{-c}$, $R_4 = h^{s_\alpha} g^{-s_\beta} T_2^{-c} T_4^c$, $R_2 = e(T_2, g_2)^{s_x} \cdot e(T_2, w)^{\cdot} \cdot e(h, g_2)^{-s_{x\alpha}} \cdot e(h, w)^{-s_\alpha} \cdot e(h, g_2)^{-s_y} \cdot e(T_2, w)^c \cdot e(g_1, g_2)^{-c}$ = $e(T_2, g_2^{s_x} w^{c+1}) \cdot e(h, g_2)^{-s_{x\alpha} - s_y} \cdot e(h, w)^{-s_\alpha} \cdot e(g_1, g_2)^{-c}$. We notice that the pairing computation we need to obtain R_2 is only one, and other pairings can be pre-computed.
  2. The verifier calculates $h_i = H_1(T, i)$, and checks if $h_i^{s_y}$=$E_i' \cdot (E_i)^c$ holds.
  3. The verifier checks if c is equal to $H_2(m, T_1, T_2, T_3, T_4, R_1, R_2, R_3, R_4)$. If all the verifications pass, the verifier accepts the signature, and records E_i into tracing log $TLOG_T$. Else, the verifier rejects the execution.

– Open: When necessary, the open authority obtains T_1, T_2, T_3, T_4 from the signature, and calculates A=$T_2(T_1)^{\xi_1}$=$T_4(T_3)^{\xi_2}$. By searching A in Member Joining records, the open authority can find out the corresponding upk, the public key of the user who has issued that signature.

3.5 Security Analysis

Theorem 3.1. *The proposed scheme is correct, assuming GM, user, verifier and open authority are all honest.*

Proof. The proof is straightforward. If user U_i and GM are honest, U_i will carefully select a secret key y_i and GM will make sure that g^{y_i} is different from others' public keys. So the secret key y_i is also different from the others. In the Member Joining protocol, U_i obtains a member certificate $(A = (g_1 h^{y_i})^{\frac{1}{\gamma+x}}, x, y_i)$ from the honest GM. For each unused $k_i \in [1, k]$ during period T, there will not exist a value the same as $E_i = h_i^{y_i} = (H_1(T, k_i))^{y_i}$ for index k_i during period T, because y_i is different from others' secret keys. Then the verifier will not reject such an execution. On the knowledge of $(A = (g_1 h^{y_i})^{\frac{1}{\gamma+x}}, x, y_i)$, U_i is able to generate a valid proof π_i=$ZKP\{(A, x, y_i) : E_i$=$h_i^{y_i} \wedge e(A, g_2)^x \cdot e(A, w) \cdot e(h, g_2)^{-y_i}$=$e(g_1, g_2)\}$, which will then be accepted by the verifier as a successful execution. Since π_i is honestly generated, the open authority can extract A and find out the corresponding user public key, when it is necessary. □

Theorem 3.2. *If the group signature scheme by Delerablée and Pointcheval [9] is unforgeable, the proposed single-verifier k-times group signature is unforgeable.*

Proof. (**Sketch**) Suppose that an algorithm $\mathcal{A}$ can forge the proposed group signature with non-negligible probability. Our scheme is combination the online k-times limitation with Delerablée et al's group signature [9], both of which are linked by a zero-knowledge proof of a secret value y. Given an instance of forged single-verifier k-times group signature, we can extract from it an instance of Delerablée et al's group signature.

The technique is briefly descripted here. The single-verifier k-times group signature is of this form: $(i, E_i, (T_1, T_2, T_3, T_4), E_i', (R_1, R_2, R_3, R_4), c, (s_\alpha, s_\beta, s_x, s_y, s_{x\alpha}))$. When the algorithm $\mathcal{A}$ is about to generate the forged signature, we take control of the random oracle for the challenge and rewind the process. Then we can extract two related signatures, with the same hash-query but different challenges. $(i, E_i, (T_1, T_2, T_3, T_4), E_i', (R_1, R_2, R_3, R_4), c, (s_\alpha, s_\beta, s_x, s_y, s_{x\alpha})\ c', (s_\alpha', s_\beta', s_x', s_y', s_{x\alpha}')$. Thereafter, simply applying the same technique as the one used to prove the soundness of zero-knowledge proof, one gets a valid certificate (A, x, y) and then generates a successful forgery for Delerablée and Pointcheval's group signature scheme. □

Theorem 3.3. *If the group signature scheme by Delerablée and Pointcheval [9] is anonymous, DDH problem is hard in $\mathbb{G}_T$, and the proof of knowledge technique is zero-knowledge, the proposed single-verifier k-times group signature is anonymous.*

Proof. A user's signature issued in the Sign protocol can be divided into two parts. One is tracing tag $E_i = h_i^{y_i}$, with $h_i \in \mathbb{G}_T$, the other part is zero knowledge proof of knowing $(A = (g_1 h^{y_i})^{\frac{1}{y_i+x}}, x, y_i)$ which satisfying $E_i = h_i^{y_i}$

and $e(A, g_2)^x \cdot e(A, w) \cdot e(h, g_2)^{-y_i}$=$e(g_1, g_2)$ at the same time. Suppose there exists an adversary $\mathcal{A}$ can determine which user is running the signing execution in the Anonymity game, then we can use $\mathcal{A}$ to solve DDH problem in $\mathbb{G}_T$. In this case, $\mathcal{A}$ can serve as an algorithm that shows connections between elements in $\mathbb{G}_T$ and $\mathbb{G}$. Suppose $\mathcal{A}$ inverts elements in $\mathbb{G}_T$ to those in $\mathbb{G}$ with probability at least ϵ, we denotes it as $\mathcal{A}(g, x)$, with $x \in \mathbb{G}_T$. We are given a DDH problem instance, that is a quadruple $\{y, y^a, y^b, y^c\}$ of elements of $\mathbb{G}_T$, and we are asked to determine if $c = ab \pmod p$. Define algorithm $\mathcal{B}$ as follows.

1. Choose a random $g \in \mathbb{G}$, and compute $q_1 = \mathcal{A}(g, y)$, $q_2 = \mathcal{A}(g, y^a)$, $q_3 = \mathcal{A}(g, y^b)$, $q_4 = \mathcal{A}(g, y^c)$.
2. Compute $e(q_1, q_4)$ and $e(q_2, q_3)$. If the two are equal output 1; else output 0.

Suppose all four outputs of algorithm $\mathcal{A}$ are correct. Then $q_2 = q_1^a$, $q_3 = q_1^b$, and $q_4 = q_1^c$. We therefore have $e(q_1, q_4) = e(q_1, q_1)^c$ and $e(q_2, q_3) = e(q_1, q_1)^{ab}$. The two elements are equal if and only if $c = ab \pmod p$. Thus if all four outputs are correct $\mathcal{B}$ gives a correct output to the Decision Diffie-Hellman problem. The probability that all four outputs are correct is at least ϵ^4.

We use standard proof of knowledge skill for the signature. If the adversary $\mathcal{A}$ can figure out which user is running the signing execution, we derive a contradiction for the zero-knowledge proof of knowledge.

We notice that if two users simultaneously select the same index k_i during period T, the verifier can determine that two signing executions are from two different users. However, this will not weaken the anonymity of the users, because the verifier still cannot determine which user is running the execution and cannot find out that two signatures are from the same user. □

Theorem 3.4. *The proposed single-verifier k-times group signature is traceable assuming the group signature scheme by Delerablée and Pointcheval [9] is traceable.*

Proof. (**Sketch**) Our single-verifier k-times group signature is of this form: $(i, E_i, (T_1, T_2, T_3, T_4), E_i', (R_1, R_2, R_3, R_4), c, (s_\alpha, s_\beta, s_x, s_y, s_{x\alpha}))$. With any instance of single-verifier k-times group signature, one can extract an instance of Delerablée et al's group signature, which is of the form (T_1, T_2, T_3, T_4), (R_1, R_2, R_3, R_4), c, $(s_\alpha, s_\beta, s_x, s_{x\alpha}))$. If single-verifier k-times group signature is untraceable, the extracted group signature is untraceable for Delerablée et al's scheme. Then we obtain a contradiction for the traceability of Delerablée et al's group signature scheme. □

Theorem 3.5. *Suppose an adversary colludes with w users and k is the number of times the verifier allows each user to sign on period T. If the adversary issues signatures more than kw times, it must be detected by the honest verifier.*

Proof. For each user on period T, only k bases can be used for generating traceable tags during the execution, namely $h_1 = H_1(T, 1), \ldots, h_k = H_1(T, k)$. If the user uses additional base, it can be easily detected by the verifier, because the user has to tell the verifier which base he is using in the execution.

Using these k bases, each user can perform only k times successful executions, and w users can perform kw times. After kw times normal executions, if they collude together and want to sign the messages for one more time, they need a new secret key to create a different tracing tag other than the kw used tags. And they also need to prove knowing a message-signature pair for the secret key, which is not obtained from normal interaction with the GM. It cannot be fulfilled due to the unforgeability of our scheme. □

4 Times Limited Accountable Anonymous Online Submission System

Our system can be divided into two parts: organization and website. To ensure the security of our system, the security of both parts should be considered.

- **Hardware Infrastructure Security**. The Hardware infrastructures that support the system, such as servers and backup disks, should only be accessible to trusted persons.
- **Secure Channels**. There are secure channels between organization and website, in order for website to obtain and update public keys securely.
- **Reliability**. Robust design should be provided to support for backup, load balancing, and failover.

The security considerations described above are out of scope of this paper.

With the building tool provided in Section 3, we can easily construct times limited accountable anonymous online submission system suitable for various websites. Each website needs only to announce the allowed times for future periods and act as the only verifier.

4.1 System Preparation Phase

The website signs agreements with several organizations that the website allows people from these organizations to post messages anonymously on it. In addition, the organizations agree to reveal the identities of abusers when the website requires. The agreements should show what kinds of messages are allowed to be submitted. A trusted party in each organization generates the public parameters and the corresponding secret keys for GM and OA, as described in Section 3. GM and OA in each organization can be trusted by the websites. The website prepares storage space for recording the authenticated submissions. It can be a database on a hard disk, with different tables for different periods and organizations.

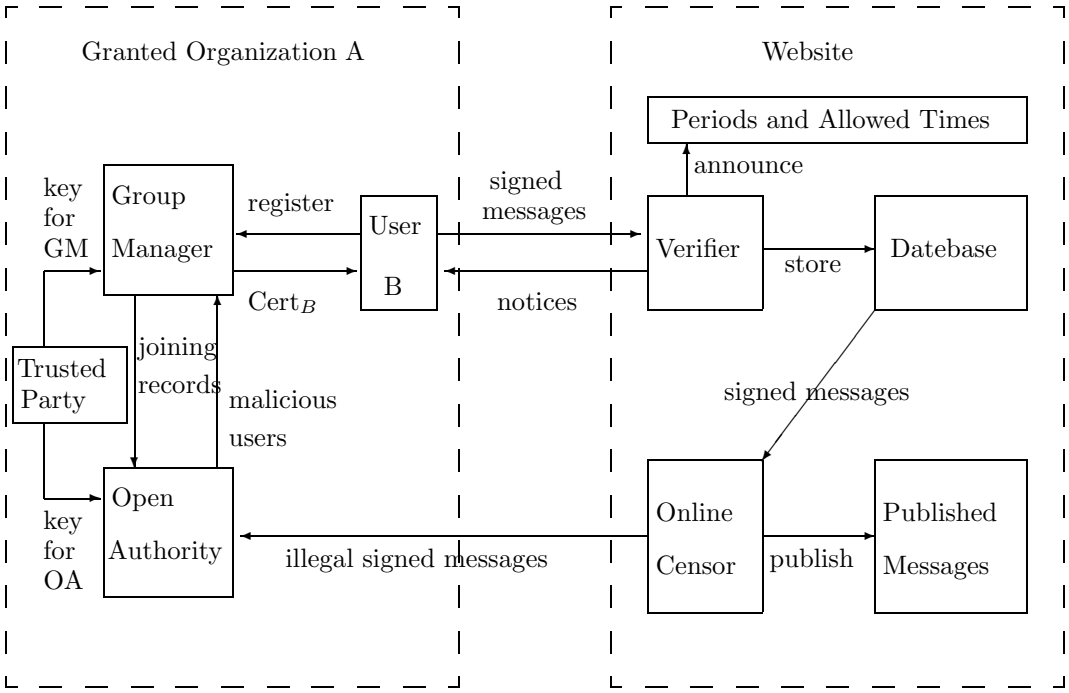

Figure 1: Times Limited Accountable Anonymous Submission System with Online Censor

4.2 Member Joining Phase

Members in each organization register with GM to obtain certificates, by running **Member Joining** algorithm as described in Section 3. Each organization should notify its members of the allowed kinds of messages to be submitted.

4.3 Times Announcing Phase

The verifier publishes a list to indicate submission times allowed for each future period. The list is as follows, $(T_1,k_1),\ldots,(T_n,k_n)$. Alternatively, the verifier can choose to publish a k for all future periods. In addition, the website should notify the users of the current period.

4.4 Message Submission Phase

When a user from the agreed organization wants to submit a message to the website, he selects a random unused number less than the allowed time for current period, and generates the single-verifier k-times group signature as described in Section 3. He writes down the message in a pre-designed form, indicating his organization and attaching the signature.

4.5 Signed Message Verification Phase

On receiving the submission, the website can verify it as described in Section 3. If the submission is from the granted organization and its signature passes the verification, the message and its signature are recorded into the database table for current period and the coming organization. If not, the submission is rejected and a notice is sent to the user immediately.

4.6 Message Review Phase

These verifications and recording can be automatically done by software programs. The website can choose online or offline review for the messages, according to its running policy. When online review is chosen, some people (censors) are deployed to read the messages before they can be published. Figure 1 illustrates a submission system with online censor. When in the case of offline review, the messages can be published immediately. Later, some people are deployed to browse through these published messages and remove those impropriate. No matter which kind of review is used, if a message is found impropriate for the website, the message and corresponding signature are sent to open authority for further treatment.

5 Security Attributes and Performance Comparison

5.1 Security Attributes

Our system inherits the security attributes from group signature scheme [9], in addition to times limited authentication.

– **Anonymity**. Given signatures produced by a user, no one except the open authority should be able to find

out the signer's identity. The website cannot decide two signatures are from the same user.

- **Traceability**. Given a valid signature, the open authority is bound to identify the signer.
- **Non-frameability**. Anybody, even group manager and open authority, is not able to wrongly accuse someone for having signed a message.
- **Concurrent Join** The system allows for several users to register at the same time.
- **Dynamic Revocation**. As indicated in [9], the group manager can remove a user from the organization, by publishing new public parameters and some information of the revoked user. The unrevoked users can update their certificates accordingly. Mass revocations are done one by one.
- **Times Limited Authentication**. No one can authenticate more than announced number to the honest verifier.

5.2 Performance Comparison

We evaluate our scheme in Table 1 by comparing it with several related works, including the group signature scheme by Delerablée and Pointcheval [9], dynamic k-TAA by Au et al.[1], periodic k-times anonymous authentication by Camenisch et al. [4], and the selectable k-TAA scheme by Emura et al. [10]. Because schemes in [1, 4] did not provide detailed computations of zero-knowledge proof, so the computation cost and signature length are given by us approximately.

We notice that in scheme of k-TAA and its variants, such as dynamic k-TAA [1] and periodic n-TAA [4], the signing and verifying cost is huge and the signature length is not constant, since they need to prove that the committed signing index lies in an interval $[1, k]$. Users are fully anonymous when they authenticate no more than the allowed times, or else their identities are exposed.

The selectable k-times relaxed anonymous authentication by Emura et al. [10] is efficient in signing and verifying phase. However, the computation cost for user and AP is huge in granting phase, which is linear to the allowed number k. The storage cost for user is also linear to k. The weaken anonymity is suitable for their application since the linkable authentications are needed for AP to adjust marketing strategy.

Our scheme is almost as efficient as the group signature scheme [9]. We need only a few more computations and some extra length to realize the times limited property.

6 Conclusion and Discussion

We propose single-verifier k-times group signature scheme to allow each user to authenticate up to k_i times during period T_i, without leaking the privacy of the users, while maintaining the ability of revealing the identities of abusers. We show that the scheme can be used to construct flexible anonymous online submission control system for websites.

We notice that if two users from the same organization using the same index k_i for signing during the same period, the website can know the two submissions are from two different users. However, this will not weaken the anonymity of the users, because the website still cannot determine which user is submitting and cannot tell two submissions are from the same user.

Our single-verifier k-times group signature scheme can be turned into a generic scheme, in which we can employ other group signature scheme so as to achieve different efficiency.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (No. 60773202, 61070168).

References

[1] M. H. Au, W. Susilo, and Y. Mu, (2006). Constant-size dynamic k-TAA. *SCN 2006*, LNCS 4116, Springer-Verlag, Maiori, Italy, pp. 111-125.

[2] D. Boneh, and X. Boyen, (2004). Short signatures without random oracles. *EUROCRYPT 2004*, LNCS 3027, Springer-Verlag, Interlaken, Switzerland, pp. 56-73.

[3] D. Boneh, B. Lynn, and H. Shacham, (2001). Short signatures from the Weil pairing, *ASIACRYPT 2001*, LNCS 2248, Springer-Verlag, Gold Coast, Australia, pp. 514-532.

[4] J. Camenisch, S. Hohenberger, M. Kohlweiss, A. Lysyanskaya, and M. Meyerovich, (2006). How to win the clone wars: efficient periodic n-Times anonymous authentication. *ACM CCS 2006*, ACM Press, Alexandria, VA, USA, pp. 201-210.

[5] J. Camenisch, and M. Michels, (1997). Efficient group signature schemes for large group. *CRYPTO 1997*, LNCS 1296, Springer-Verlag, Santa Barbara, California, USA, pp. 410-424.

[6] D. Chaum, (1985). Security without identification: transaction systems to make big brother obsolete. *Communications of the ACM*, 28(10), pp. 1030-1044.

[7] D. Chaum, and E. V. Heyst, (1991). Group signatures. *EUROCRYPT 1991*, LNCS 547, Springer-Verlag, Brighton, UK, pp. 257-265.

[8] I. Damgård, K. Dupont, and M. O. Pedersen, (2006). Unclonable group identification. *EUROCRYPT 2006*, LNCS 4004, Springer-Verlag, St. Petersburg, Russia, pp. 555-572.

Table 1: Performance Comparison with Previous Works

	[9]	[1]	[4]	[10]	Our scheme
Public Key Size	constant	constant	constant	constant	constant
Credential Size	$(1G_1+2p)$	$1G_1+1G_2+4p$	$(1G_1+2p)$	$(k+2)G_1+2P$	$(1G_1+2p)$
Granting Computations	constant	constant	constant	$4E_T + 3M_T +$ (3k+13)E + (k+3)M	constant
Signing Computations	$8M+3E_T+2M_T$	$(\log k+16)E + 9M$	$(\log k+21)E+ 14M + 3P$	3E	$9E + 2M + 3E_T + 3M_T$
Verifying Computations	$9M+3E_T+3M_T+1P$	$(\log k+27)E+19M+3P+9E_T+9M_T$	$(\log k+37)E + 24M + 3P+1M_T$	4E + 2M	$8E + 5M + 3E_T + 3M_T + 1P$
Signature Length	$(4G_1+4p+1c)$	$(21+3\log k)p + (\log k+2)G_p+4G_1+1c$	$(13+3\log k)p + \log kG_p + 8G_1 + 2G_T+3c$	$3G_1 + 1p + 1c$	$4G_1+2G_T + 5p+1c$
Action when authenticated $> k$	-	identity can be extracted	identity can be extracted	reject	reject
Anonymity when authenticated $\leq k$	-	fully anonymous	fully anonymous	linkable to AP, anonymous to others except OA	anonymous to all except OA
Different Limits for Each User	-	no	no	yes	no
AP Can Choose Users	-	yes	no	no	no

G_1: element in $\mathbb{G}_1$; G_2: element in $\mathbb{G}_2$; G_T: element in $\mathbb{G}_T$;
G_p: element in $\mathbb{G}_p$ where DDH problem is difficult; p: element in $\mathbb{Z}_p$; P: pairing in $\mathbb{G}_1 \times \mathbb{G}_2$;
M: multiplication (or division) in $\mathbb{G}_1$; M_T: multiplication (or division) in $\mathbb{G}_T$;
E: exponentiation in $\mathbb{G}_1$; E_T: exponentiation in $\mathbb{G}_T$; c: a small integer for zero-knowledge proof.

[9] C. Delerablée, and D. Pointcheval, (2006). Dynamic fully anonymous short group signatures. *VIETCRYPT 2006*, LNCS 4341, Springer-Verlag, Hanoi, Vietnam, pp. 193-210.

[10] K. Emura, A. Miyaji, and K. Omote, (2009). A Selectable k-Times Relaxed Anonymous Authentication Scheme. *WISA 2009*, LNCS 5932, Springer-Verlag, Busan, Korea, pp. 281-295.

[11] M. Layouni, and H. Vangheluwe, (2007). Anonymous k-Show credentials. *EuroPKI 2007*, LNCS 4582, Springer-Verlag, Palma de Mallorca, Spain, pp. 181-192.

[12] J. K. Liu, V. K. Wei, and D. S. Wong, (2004). Linkable spontaneous anonymous group signature for ad hoc groups (extended abstract). *ACISP 2004*, LNCS 3108, Springer-Verlag, Sydney, Australia, pp. 325-335.

[13] L. Nguyen, (2006). Efficient dynamic k-Times anonymous authentication. *VIETCRYPT 2006*, LNCS 4341, Springer-Verlag, Hanoi, Vietnam, pp. 81-98.

[14] L. Nguyen, and R. Safavi-Naini, (2005). Dynamic k-Times anonymous authentication. *ACNS 2005*, LNCS 3531, Springer-Verlag, New York, NY, USA, pp. 318-333.

[15] R. Rivest, A. Shamir, and Y. Tauman, (2001). How to leak a secret. *ASIACRYPT 2001*, LNCS 2248, Springer-Verlag, Gold Coast, Australia, pp. 552-565.

[16] I. Teranishi, J. Furukawa, and K. Sako, (2004). k-Times anonymous authentication (Extended Abstract). *ASIACRYPT 2004*, LNCS 3329, Springer-Verlag, Jeju Island, Korea, pp. 308-322.

[17] I. Teranishi, J. Furukawa, and K. Sako, (2009). k-Times Anonymous Authentication. *IEICE Transactions*, 92-A(1), pp. 147-165.

[18] I. Teranishi, and K. Sako, (2006). k-Times anonymous authentication with a constant proving cost. *PKC 2006*, LNCS 3958, Springer-Verlag, New York, NY, USA, pp. 525-542.