Int. J. Communications, Network and System Sciences, 2018, 11, 9-26
http://www.scirp.org/journal/ijcns
ISSN Online: 1913-3723
ISSN Print: 1913-3715

Anonymous and Unlinkable Membership Authentication with Illegal Privilege Transfer Detection

Sung-Ming Yen, Tsung-Min Kuo, Tzu-Yu Yang

Department of Computer Science and Information Engineering, National Central University, Taiwan
Email: yensm@csie.ncu.edu.tw, tmkuo@csie.ncu.edu.tw

How to cite this paper: Yen, S.-M., Kuo, T.-M. and Yang, T.-Y. (2018) Anonymous and Unlinkable Membership Authentication with Illegal Privilege Transfer Detection. *Int. J. Communications, Network and System Sciences*, **11**, 9-26.
https://doi.org/10.4236/ijcns.2018.112002

Received: December 28, 2017
Accepted: February 4, 2018
Published: February 7, 2018

Copyright © 2018 by authors and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/
Open Access

Abstract

Anonymous authentication schemes, mostly based on the notion of group signatures, allow a group member to obtain membership from a server and gain access rights if the member can prove their authenticity to the verifier. However, existing authentication schemes are impractical because they neglect to provide an exclusive verification of the blacklist. In addition, the schemes are unaware of malicious members who are involved in privilege transferring. In this paper, a novel membership authentication scheme providing detection of membership transfer and proof of membership exclusiveness to the blacklist is proposed.

Keywords

Anonymous Authentication Schemes, Traitor Tracing, Revocation of Membership, Dynamic Membership

1. Introduction

The rapid development of the Internet has resulted in an increase in electronic transactions that allow users to buy goods or services from online platforms provided by Internet companies, including Google, Facebook, eBay, and Twitter. Service providers must confirm whether a user is permitted to access its resource. Access control [1] [2] provides a solution by verifying either a cryptographic certificate or a username and password. However, the information provided by the user during an interaction with the service provider may undermine the user privacy: the user must risk being traced or even impersonated by corrupt service providers.

Group signatures [3] [4] [5] [6] allow group members with privilege to sign a signature under the group secret key. On the basis of an auxiliary cryptographic technique called zero-knowledge proof [7], the verifier can check the member's access rights by using the group public key without knowing the member's identity. The member obtains access to the service if their presented signature is verified to be valid. If necessary, the signature can be opened by a specific group manager to identify the signature's originator in case of a dispute. Several provably secure authentication schemes [8] [9] [10] [11] have been proposed to create anonymous memberships in which any member of a group can prove to the service provider (*i.e.*, the verifier) that they are qualified to access a service or file; however, these schemes are impractical because they do not provide exclusive verification of revoked memberships. As proven by the fact that some members have been revoked, in contrast to the use of fixed time periods [12] by employing a one-way chain, Ateniese *et al.* [13] require group members to prove that their membership does not appear on the current certificate revocation list (CRL). However, in their scheme, the verifier must check whether the member's membership fits any of the revocation information on the CRL in turn by using their "REVOKE" algorithm every time a member requests membership authentication. The cost to the group manager is proportional to the number of revoked group members because issuing new memberships to non-revoked members is required every time a membership is revoked. Clearly, the scheme performs inefficiently and has not been improved to date.

Additionally, state-of-the-art authentication schemes provide few revocation methods without describing how to detect malicious members' illegal behavior; in other words, such schemes are unaware of malicious members who have been involved in privilege transfer. This is known as impersonation or an illegal privilege transfer attack and is a priority for prevention because it regularly occurs in the aforementioned schemes and is difficult to trace. In addition, the modern authentication schemes are becoming more complicated to ensure security. However, this is not a favorable development because it will obstruct the development of membership authentication schemes, resulting in research becoming impractical and unattractive. In summary, a robust authentication scheme should contain two components: a membership authentication approach that can withstand members who engage in membership transfer and proof of membership that is exclusive to the current CRL.

In this paper, a novel membership authentication scheme is proposed that provides a simple solution for membership authentication and revocation. The proposed scheme may suffer from the disadvantage of illegal privilege transfer; however, this problem can easily be solved by employing the traitor tracing technique [14] [15] [16] [17] [18]. Traitor tracing was first suggested by [14], which discusses how to identify a traitor in a public key cryptographic scheme and proposes some approaches for revoking access rights for at least one of the traitors involved in illegal privilege transfer. To evade accountability, a traitor may attempt to modify their secret key to avoid being traced. Traitor tracing schemes ensure that no such strategy can succeed, and the schemes guarantee that the traitor's identity is revealed. Typical CRL approaches are not directly applicable to our proposed scheme because the memberships are anonymous and unlinkable. Instead of compiling a typical CRL, the dynamic accumulator technique [19] [20] [21] is employed in the proposed scheme to enable an eligible member to prove the exclusiveness of their membership on the CRL.

Organization of the Thesis

This paper is organized as follows. Section 2 describes the anonymous authentication scheme and its requirements as well as the dynamic accumulators. An anonymous and unlinkable membership authentication scheme with illegal privilege transfer detection is proposed in Section 3. Security and performance analysis of the proposed scheme is detailed in Section 4. Finally, Section 5 presents the conclusion.

2. Background

2.1. Anonymous Authentication Schemes

Group signatures [12] [13] with membership revocation are typically defined using the following algorithms:

Setup:

A probabilistic algorithm that outputs the group public key and group secret key for the group manager, given a security parameter as the input.

Join:

A protocol between the group manager and a user that results in the user becoming a group member and receiving a group signing key.

Sign:

A probabilistic algorithm that outputs an anonymous membership for a member, with some necessary parameters (including the member's group signing key) as the input.

Verify:

An algorithm for examining the validity of an alleged membership with respect to a group public key.

Open:

An algorithm, which can only be implemented by the group manager, used to determine the originator's identity.

Some authentication schemes [10] [11] have been proposed for creating anonymous memberships by extending the idea of group signatures. Three parties are involved in generic authentication schemes, namely the prover (*i.e.*, the member), issuer (*i.e.*, the key generation center [**KGC**]), and verifier (*i.e.*, the application server [**AS**]). The issuer is assumed to be a trusted third party responsible for generating unique and anonymous memberships for eligible provers. A prover with membership can prove to the verifier that they have been given an appropriate membership. The verifier can verify the validity of memberships, but knows nothing about the prover's real identity. The scheme must guarantee that different authentication messages submitted by the same prover cannot be linked.

Additionally, the following security requirements, which have been identified and discussed in the literature, should be inspected.

Unforgeability:

Only an eligible prover can obtain a unique valid membership. An adversary cannot feasibly forge a membership that can obtain verification.

Strong/weak unlinkability:

Strong unlinkability ensures that the pseudonym and real identity of a prover cannot be linked during multiple uses of the membership. Conversely, weak unlinkability allows only a pseudonym but not the prover's real identity to be linked when the prover uses the membership more than once.

Nontransferability:

Even though the verifier knows nothing about the prover's real identity during the interactions; however, a sound authentication scheme must guarantee that membership transfer behavior can be detected and abused memberships can be revoked.

Excludability:

Neither a group member nor the group manager can sign on behalf of other group members.

For efficient exclusive verification of the CRL and detection of illegal privilege transfer, the following attractive security properties are necessary:

Dynamic membership:

The membership can easily be updated by any eligible member of the group when inserting (deleting) a new (abusive) member rather than issuing a new membership or requiring the verifier to refer to the CRL.

Traitor detection:

The scheme must be able to determine the real identity of the malicious member.

2.2. Dynamic Reversed Accumulator

Accumulators were first proposed by Benaloh and de Mare [22] for combining a set of members' specific values into one accumulator. Each corresponding member is assigned a unique witness, which is used to prove the validity of their membership. However, the computational complexity of the Benaloh-de Mare scheme increases linearly, either according to the number of group members or the number of revoked members. In 2002, Camenisch and Lysyanskaya [19] proposed an efficient dynamic accumulator scheme in which members can update their witness dynamically without the authority's help. Additionally, the computational complexity of inserting and deleting a member as well as updating members' witnesses is independent of the number of accumulated values. In 2009, Camenisch *et al.* proposed an additional accumulator scheme [20] that involves using a bilinear cryptography technique. However, the schemes in [19] [20] were later proved insecure by Kuo *et al.* [21] later. Although other accumulators [23] [24] [25] [26] utilize bilinear cryptography, these schemes are either vulnerable to collusion attacks or inefficient.

In this section, we review the dynamic reversed accumulator scheme of Kuo *et al.* [21], which relies on the strong RSA assumption [7] [27]. To the best of our knowledge, their scheme is the most efficient and secure accumulator scheme for state-of-the-art dynamic accumulators and is highly applicable for the granting and revoking of privileges.

Initialization:

Let the modulus $n = p \times q$, with p and q safe primes; U be a set of t eligible members, each with an identity x_u ($u = 1, \cdots, t$); and $\tilde{U}$ be a set of members being revoked. All identities are assumed to be pairwise relatively prime, and the authority maintains the sets U and $\tilde{U}$, which are initially empty. The authority chooses an element $g \in QR_n$ and a prime z (which can be 2); computes the accumulator as $ACC = f(g,z) = g^z \bmod n$, where $g \neq 1$; and publishes (ACC, g). Here, $f(\cdot)$ is a public quasi–commutative function [21]. It holds that

- $f(f(g,x_1),x_2) = f(f(g,x_2),x_1) = g^{\prod_{u=1}^{2} x_u} \bmod n$;
- $f(g,U) = f(f(\cdots f(g,x_1)\cdots),x_t) = g^{\prod_{u=1}^{t} x_u} \bmod n.$

Member insertion:

To include a new member x_w, the authority examines whether $x_w \notin U$, and if so, adds x_w to the set U (new set of eligible members as $U' = U \cup \{x_w\}$) and updates the aforementioned archive. The new member is given a witness $wit_w = f(ACC, x_w^{-1}) = g^{z \times x_w^{-1} \bmod \phi(n)} \bmod n$ and a value x_w for $\gcd(x_w, \phi(n)) = 1$. Here, the accumulator ACC is not changed; therefore, the group members do not need to update their witnesses.

Witness verification:

Only an eligible member $x_u \in U$ can prove the validity of their system access to a verifier, that their unique value x_u is included in the public accumulator ACC, and that they know the corresponding witness wit_u on the basis of the zero-knowledge proof technique. The verifier can verify the correctness by using the online public information ACC maintained by the authority, and the group member x_u is granted access rights if the following Equation (1) holds for their claim:

$$wit_u^{x_u} \equiv g^{(z \times x_u^{-1}) \times x_u} \equiv ACC (\bmod n). \tag{1}$$

Member deletion:

When the membership of group member x_v is revoked, the authority deletes x_v from the set U and moves the value x_v into the set $\tilde{U}$. The authority computes the new accumulator $ACC' = f(ACC, x_v^{-1}) = ACC^{x_v^{-1}} \bmod n$, updates the archive, and publishes the revocation information (ACC', x_v). Knowledge of p, q is required for computing x_v^{-1}. Kuo *et al.* called their scheme a dynamic reversed accumulator because the value $x_v \in \tilde{U}$ here is subtracted from the accumulator and the accumulator decreases gradually. Additionally, each member in U must update their witness to reflect the result of the updated accumulator.

On the basis of the extended Euclidean algorithm, the eligible members $x_u \in U$ can compute the integers *a* and *b* satisfying $a \times x_u + b \times x_v = 1$ and update their witnesses as $wit'_u = ACC'^a \times wit_u^b \bmod n$, such that $\left(wit'_u\right)^{x_u} = ACC'$. Computing the witness update does not require knowledge of (*p*, *q*) and can be performed only by the eligible members $x_u \in U$. It is infeasible for the revoked member x_v to update their witness because $\gcd\left(x_v, x_v\right) \neq 1$. Crucially, the computational costs of both updating the group accumulator and each individual member's witness are independent of the size of $\tilde{U}$.

The scheme of Kuo *et al.* features a substantial computational cost reduction compared with the existing methods because renewing the accumulator and valid members' witnesses is required only when revoking violating members (not including new members).

3. Proposed Anonymous Authentication Scheme

In this section, a basic scheme of anonymous membership authentication with anonymity, unlinkability, and efficiency is proposed. Furthermore, we discuss its security. Subsequently, an enhanced version of the scheme is accordingly proposed, and this scheme is analyzed in the next section. The member must additionally establish a secure channel with the verifier in contrast to the aforementioned authentication schemes; in other words, a lower layer node-to-node secure channel with randomized encryption is assumed.

3.1. Bilinear Groups and Security Assumptions

The following definition of a bilinear map comes from [28] and is a fundamental building block of our proposed scheme. Let $\left(\mathbb{G}_1, +\right)$, $\left(\mathbb{G}_2, +\right)$, and $\left(\mathbb{G}_T, \cdot\right)$ be three groups of the same prime order *q*, and let *P*, *Q* be two generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. We say that $\left(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T\right)$ are *asymmetric* bilinear map groups if $\mathbb{G}_1 \neq \mathbb{G}_2$ and the bilinear map $\hat{e}: \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ satisfies the following properties:

- Bilinearity: $\forall \left(P, Q\right) \in \mathbb{G}_1 \times \mathbb{G}_2$ and $\forall a, b \in \mathbb{Z}_q^*$, $\hat{e}\left(aP, bQ\right) = \hat{e}\left(P, Q\right)^{ab}$.
- Nondegeneracy: $\hat{e}\left(P, Q\right) \neq 1$.
- Computability: $\forall \left(P, Q\right) \in \mathbb{G}_1 \times \mathbb{G}_2$, $\hat{e}\left(P, Q\right)$ is efficiently computable.

The proposed membership authentication scheme can be operated in both symmetric and asymmetric settings. For greater efficiency, the symmetric setting is more appropriate, whereas the asymmetric setting has greater security. Here, we directly use the asymmetric setting to enrich our cryptanalysis content in Section 4 and demonstrate the flexibility of our proposed scheme.

The security of our scheme relies on the hardness of the following problems, which were introduced in [29].

Definition 1 (**Fixed Argument Pairing Inversion Problems**). Let $\hat{e}$ be an asymmetric pairing. The *fixed argument pairing inversion* 1 (**FAPI-1**) problem is as follows: Given $\mathcal{U}_1 \in_R \mathbb{G}_1$ and a value $z \in \mathbb{G}_T$, compute $\mathcal{U}_2 \in \mathbb{G}_2$ such that $\hat{e}\left(\mathcal{U}_1, \mathcal{U}_2\right) = z$. The *fixed argument pairing inversion* 2 (**FAPI-2**) problem is as follows: Given $\mathcal{U}_2 \in_R \mathbb{G}_2$ and a value $z \in \mathbb{G}_T$, compute $\mathcal{U}_1 \in \mathbb{G}_1$ such that $\hat{e}(\mathcal{U}_1, \mathcal{U}_2) = z$.

Both problems FAPI-*j* (for *j* = 1 or 2) have a unique solution for each given pair $(\mathcal{U}_j, z) \in \mathbb{G}_j \times \mathbb{G}_T$ because the pairing is non-degenerate and the groups $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ are cyclic of order *q*. Finally, a general case of the pairing inversion problem is presented in the following definition.

Definition 2 (**Generalized Pairing Inversion** (**GPI**) **Problem**)**.** The problem is to find two values $\mathcal{U}_1 \in \mathbb{G}_1$ and $\mathcal{U}_2 \in \mathbb{G}_2$ such that $\hat{e}(\mathcal{U}_1, \mathcal{U}_2) = z$ when a pairing $\hat{e}$ as above and a value $z \in \mathbb{G}_T$ are given.

3.2. Basic Scheme

In this section, we first introduce a basic anonymous authentication scheme comprising three parties, namely the group member, **KGC**, and **AS**. A **KGC** is a trusted third party responsible for issuing private keys to all valid members, and an **AS** provides services to any eligible member with proof of valid membership. The basic scheme comprises the following algorithms:

Setup.

As mentioned in Section 3.1, $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ are three bilinear cyclic groups of prime order *q*; $\hat{e}: \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a bilinear mapping with underlying groups of same order *q*; and $P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$ are two generators. Let x_j be *N* prime numbers chosen from the field $\mathbb{Z}_q^*$, for $1 \le j \le N$. The **KGC** selects a large even integer *k* with $k < N$ and computes the group secret key as $X = \prod_{j=1}^{k} x_j$ and the corresponding group public key as $Y = \hat{e}(P, Q)^X$. The **KGC** then publishes the system parameters as

$$\mathcal{G} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, \hat{e}, P, Q, Y).$$

Join.

For each legitimate member U_i of a group, the **KGC** randomly selects $k/2$ elements of x_j (the components of this subset are denoted as $\hat{x}_j$) and computes $a_i = \prod_{j=1}^{k/2} \hat{x}_j$ and $b_i = X/a_i$. Subsequently, U_i is given their private key ($a_i P$, $b_i Q$). Clearly, we have $\hat{e}(a_i P, b_i Q) = \hat{e}(P, Q)^X = Y$, but $a_i P$ and $b_i Q$ cannot be used directly as proof of membership, otherwise any two application service requests are easily linked and the member's privacy is threatened. Here, an archive is required for maintaining the tuple (a_i, b_i, U_i) in which the **KGC** can reveal the real identity of a malicious member who has been recognized as a traitor.

Sign.

When the member U_i requests service from an **AS**, U_i selects two random numbers $\alpha, \beta \in \mathbb{Z}_q^*$ and computes $\mathbb{P} = a_i P + \alpha P$ and $\mathbb{Q} = b_i Q + \beta Q$. U_i also computes $A = \hat{e}(a_i P, Q)^\beta$, $B = \hat{e}(P, b_i Q)^\alpha$, and $C = \alpha\beta P$. Here, the tuple $\{\mathbb{P}, \mathbb{Q}, A, B, C\}$ is the membership of U_i for obtaining access to application services provided by a specific **AS**.

Verify.

1) $U_i \Rightarrow$ **AS**$: \{\mathbb{P}, \mathbb{Q}, A, B, C\}$ over a secure channel.

2) **AS** verifies the membership proof by checking whether

$$\hat{e}(\mathbb{P},\mathbb{Q}) = Y \times A \times B \times \hat{e}(C,Q). \tag{2}$$

Because blinding factors α and β are used, U_i can prove their membership multiple times to the same or to a different **AS**; by contrast, all the authentication messages $\{\mathbb{P},\mathbb{Q},A,B,C\}$ cannot be linked to reveal that they are all generated by the same member U_i.

Correctness of the scheme.

The correctness of the verification is shown as follows. Given the group public key $Y = \hat{e}(P,Q)^X = \hat{e}(a_iP,b_iQ)$ and the membership proof $\{\mathbb{P},\mathbb{Q},A,B,C\}$, U_i gains access to **AS**'s services if the scheme works correctly and Equation (3) holds:

$$\begin{aligned}\hat{e}(\mathbb{P},\mathbb{Q}) &= \hat{e}((a_i+\alpha)P,(b_i+\beta)Q)\\ &= \hat{e}(P,Q)^{(a_i+\alpha)\times(b_i+\beta)}\\ &= \hat{e}(P,Q)^{a_ib_i} \times \hat{e}(a_iP,Q)^{\beta} \times \hat{e}(P,b_iQ)^{\alpha} \times \hat{e}(P,Q)^{\alpha\beta}\\ &= Y \times A \times B \times \hat{e}(C,Q).\end{aligned} \tag{3}$$

U_i cannot compute and send $C = \hat{e}(P,Q)^{\alpha\beta}$ directly to the **AS**. Otherwise, the scheme becomes insecure if it is designed in the aforementioned approach. Because the verification equation would become

$$\hat{e}(\mathbb{P},\mathbb{Q}) = Y \times A \times B \times C,$$

and any attacker could select two random $\mathbb{P}'$ and $\mathbb{Q}'$ and then compute $C' = \hat{e}(\mathbb{P}',\mathbb{Q}')/(Y \times A' \times B')$, where A' and B' are also randomly selected. The attack can thusly pass the verification procedure with the forged membership proof $\{\mathbb{P}',\mathbb{Q}',A',B',C'\}$.

Selection of parameter *k*.

Let k = 100 and 200; it yields $C(100,50) = 10^{29}$ and $C(200,100) = 10^{59}$ combinations of the value a_i, respectively, where $C(\cdot)$ is a combination function.

Remarks and Discussion

Impersonation or illegal privilege transfer attack.

A sound anonymous membership authentication scheme should consider how to counteract a forged membership duplication to others from a valid member. That is, a valid member U_i may attempt to share their private key $\{a_iP,b_iQ\}$ with their untrusted friend U_x. We assume that collusion among the **AS**, **KGC**, and U_x is possible. With knowledge of both a_iP and b_iQ (and the related identity of its owner revealed by the U_x), the **AS** can obtain both $\alpha P = \mathbb{P} - a_iP$ and $\beta Q = \mathbb{Q} - b_iQ$ when the original member U_i logs in to the **AS**. To check whether a service request is made by U_i, the **AS** verifies whether

$$\hat{e}(C,Q) = \hat{e}(\alpha P,\beta Q). \tag{4}$$

Clearly, this "private key revelation" forces U_i not to share their private key and privilege with others; otherwise, any two of U_i's service requests can be linked and their anonymity will be ruined. In this attack, the original member U_i also risks privilege revocation by the **KGC** (here, a typical blacklist is required) after an unauthorized privilege transfer is confirmed. The privilege is thus *nontransferable*.

In the following, we show another privilege transfer approach launched by the member U_i, but this approach does not undermine U_i's anonymity. Let α and β be two blinding factors as before; U_i uses a third blinding factor γ and sends the "transformed" private key $\{\gamma a_i P, \gamma^{-1} b_i Q\}$ to their friend U_x, who can be either trusted or untrusted. On the basis of this transformed private key, the unprivileged U_x can prove their membership to the **AS** through the same anonymous authentication scheme by computing $\mathbb{P} = \gamma a_i P + \alpha P$, $\mathbb{Q} = \gamma^{-1} b_i Q + \beta Q$, $A = \hat{e}(a_i \gamma P, Q)^{\beta}$, $B = \hat{e}(P, b_i \gamma^{-1} Q)^{\alpha}$, and $C = \alpha\beta P$. The **AS** also verifies the membership proof by checking whether Equation (2) holds. In this attack, collusion between the **AS** and U_x to threaten the original member U_i's privacy is impossible. Nevertheless, the untrusted friend U_x can disclose the fact of illegal privilege transfer to the **KGC** by providing $\{\gamma a_i P, \gamma^{-1} b_i Q\}$. Recall that the **KGC** keeps the a_i and b_i selected for each member U_i and can therefore check whether a member U_i is involved in an unauthorized privilege transfer as follows:

$$\hat{e}\left(\gamma a_i P, \gamma^{-1} b_i Q\right)^{a_i^{-1} b_i^{-1}} = \hat{e}(P, Q). \tag{5}$$

If Equation (5) holds, the original member U_i is revoked, and this forces U_i to not share their transformed private key and privilege with others.

In addition, if U_x would never betray U_i, the trusted **KGC** can be consulted online to recognize this privilege transfer as follows. Assume that the **KGC** can compute $\hat{e}(P, Q)^{a_i + b_i}$ in advance for all registered members. If the **AS** provides a suspicious $\{\mathbb{P}, \mathbb{Q}, A, B\}$ to the **KGC** for investigating potential privilege transfers, the **KGC** attempts to use all registered members' information (a_i, b_i), for $1 \le i \le N$, and computes both

$$\pi_i = A^{a_i^{-1}} B^{b_i^{-1}} = \hat{e}(P, Q)^{\gamma\beta + \gamma^{-1}\alpha} \tag{6}$$

and

$$\begin{aligned} \lambda_i &= \hat{e}(\mathbb{P}, Q) \times \hat{e}(P, \mathbb{Q}) \\ &= \hat{e}(P, Q)^{\gamma a_i + \alpha}\, \hat{e}(P, Q)^{\gamma^{-1} b_i + \beta} \\ &= \hat{e}(P, Q)^{\gamma a_i + \gamma^{-1} b_i + \alpha + \beta}. \end{aligned} \tag{7}$$

The **KGC** then tests whether any $\hat{e}(P, Q)^{a_i + b_i} \times \pi_i$ equals λ_i to verify whether the received authentication message was generated by privileged member U_i. Clearly, the authentication message generated from a transformed private key with $\gamma \neq 1$ will fail to pass the verification, and we can conclude that someone has transferred their membership to someone else. The **KGC** knows nothing regarding the malicious member's real identity, which is known as "weak unlinkability." In a medium-sized setting with a moderately large number of members, the described online investigation might be possible if not performed frequently. However, this method cannot completely prevent illegal privilege transfer attack.

Replay attack.

This basic scheme cannot withstand the replay attack.

3.3. Enhanced Version of the Proposed Scheme

Consider the potential illegal privilege transfer attack and unpreventable replay attack mentioned in Section 3.2.1. An enhanced scheme is proposed in this section. The scheme features anonymity and unlinkabilty and guarantees security against the aforementioned attacks. Because some algorithms are identical to those defined in Section 3.2, including **Setup** and **Join**, this section describes only the differences. The algorithms of the enhanced scheme are detailed as follows:

Sign.

Let T_i be a timestamp and $h(\cdot):\{0,1\}^* \to \mathbb{G}_1$ be a collision-free one-way hash function. When the member U_i requests service from an **AS**, U_i selects two random numbers $\alpha, \beta \in \mathbb{Z}_q^*$ and computes $A = \alpha\beta P$, $B = \alpha P$, $\mathbb{P} = \beta^{-1} a_i P + t\alpha P$, and $\mathbb{Q} = \beta b_i Q$, where $t = h(T_i)$. Here, the tuple $\{\mathbb{P}, \mathbb{Q}, A, B, T_i\}$ is the membership of U_i for obtaining access to the application services provided by a specific **AS**.

Verify.

1) $U_i \Rightarrow$ **AS** $: \{\mathbb{P}, \mathbb{Q}, A, B, T_i\}$ over a secure channel.

2) Let T_V be the current timestamp of the **AS** and T_{td} be an appropriate tolerance in the time delay. Given the group public key *Y*, the **AS** can verify the membership proof presented by a member U_i, and U_i is granted access rights if the following Equations ((8) and (9)) hold; otherwise, the **AS** rejects the request of U_i:

$$T_V - T_i \le T_{td} \tag{8}$$

$$\hat{e}(\mathbb{P}, \mathbb{Q}) \equiv Y \times \hat{e}(tB, \mathbb{Q}) \tag{9}$$

The scheme enables the **AS** to validate U_i's claim while learning nothing about their real identity, even if it colludes with the **KGC**. For the purpose of anonymous authentication and *strong unlinkability*, two blinding factors and a timestamp are employed so that a member can prove their membership multiple times to the same or to a different **AS**. All the authentication messages $\{\mathbb{P}, \mathbb{Q}, A, B, T_i\}$ cannot be linked to reveal that they are all generated by the same member. Cryptanalysis of this enhanced scheme is presented in Section 4.

3.3.1. Detection of Illegal Privilege Transfer

The member U_i is assumed to be able to send the transformed private key $\{\gamma a_i P, \gamma^{-1} b_i Q\}$ to their friend U_x, who can be either a trusted or an untrusted individual, where γ is a random number and $\gamma \ne 1$. After obtaining the transformed private key, U_x computes $A = \alpha\beta P$, $B = \alpha P$, $\mathbb{P} = \beta^{-1}\gamma a_i P +$

$t\alpha P$, and $\mathbb{Q} = \beta\gamma^{-1}b_i Q$, where $t = h(T_x)$. The unprivileged party U_x can prove their membership to the **AS** through the aforementioned improved anonymous authentication scheme and obtain access to the resource on the **AS** if Equations ((8) and (9)) hold. Here, collusion between the **AS** and U_x that threatens the original member U_i's privacy is impossible. As mentioned in Section 3.2.1, two approaches exist for detecting whether a member U_i is involved in an unauthorized privilege transfer. The first is to let U_x disclose U_i's illegal privilege transfer by providing $\{\gamma a_i P, \gamma^{-1} b_i Q\}$ to the **KGC**; however, this approach is passive and impractical for preventing private keys from being transformed. The second is that the trusted **KGC** can be consulted online to recognize the privilege transfer as follows. Recall that the **KGC** retains a_i and b_i when generating private keys for each member U_i. If the **AS** provides a suspicious $\{\mathbb{P}, \mathbb{Q}, A, B, T_i\}$ to the **KGC** for investigating potential privilege transfer, the KGC attempts to use all values of b_i, for $1 \le i \le N$, of registered members and computes both

$$\pi_i = \hat{e}\left(B, \mathbb{Q}^{b_i^{-1}}\right) = \hat{e}(P, Q)^{\alpha\beta\gamma^{-1}} \tag{10}$$

and

$$\lambda_i = \hat{e}(A, Q) = \hat{e}(P, Q)^{\alpha\beta}. \tag{11}$$

The **KGC** checks whether any π_i equals λ_i to determine whether the received authentication message was generated by privileged member U_i. Clearly, the aforementioned authentication message generated from a transformed private key with $\gamma \ne 1$ fails to pass the verification. We can conclude that the received authentication message is an impersonated membership and that someone has shared their private key and privilege with another, but the **KGC** does not know who is the traitor at this stage because the proposed authentication scheme provides "anonymity". Consequently, the **KGC** uses the information (a_i, b_i) to compute θ_i, for $1 \le i \le N$, as follows:

$$\begin{aligned} \theta_i &= \hat{e}\left((\mathbb{P} - tB)^{a_i^{-1}}, \mathbb{Q}\right) = \hat{e}\left(P^{\left(\beta^{-1}\gamma a_i\right) \times a_i^{-1}}, Q^{\beta\gamma^{-1}b_i}\right) \\ &= \hat{e}(P, Q)^{\left(\beta^{-1}\gamma\right) \times \left(\beta\gamma^{-1}b_i\right)} = \hat{e}(P, Q)^{b_i}. \end{aligned} \tag{12}$$

The **KGC** subsequently examines all values of b_i, for $1 \le i \le N$, to determine whether any θ_i equals $\hat{e}(P, b_i Q)$ and thus discover the real identity and revoke the membership of the traitor U_i who is involved in an unauthorized privilege transfer. This online detection approach can be performed regularly in case the **AS** has noticed that an unauthorized privilege transfer has occurred in the system.

3.3.2. Exclusiveness of the Membership

By employing the dynamic reversed accumulator of Kuo *et al.*, which is described in Section 2.2, a member Alice who has been included in the set U receives a membership (wit_A, x_A) and can therefore prove to the **AS** that her identity x_A is not on the CRL; this is called "exclusiveness of the membership". Of course, a dynamic public archive is required for storing information regarding joined and revoked members as well as the current accumulator ACC. Each member is assumed to read the archive regularly and update their witness when ACC has been changed. This accumulator performs more efficiently than existing methods because renewing the accumulator and valid members' witnesses is required only when revoking violating members but not when adding new members. The **AS** thus does not have to verify whether a member is on the CRL in contrast to those presented in [6] [13]. Additionally, forging of the witness by an adversary is infeasible according to the strong RSA assumption [7] [27].

In addition, the accumulator of Kuo *et al.* provides efficient multiwitness verification in which a group member may access multiple services or files simultaneously and the **AS** can verify the member's qualifications simultaneously. This property is outstanding and has not been demonstrated in previous studies. Suppose that m services exist, namely $S_1, S_2, \cdots, S_m$, provided by various **AS**. The **KGC** must generate m accumulators in advance as $ACC_j = f\left(g, z_j\right) = g^{z_j} \bmod n$ for service S_j, where $j = 1, \cdots, m$. If Alice is permitted to access the services S_1 and S_2 both provided by the same **AS**, she may be assigned the witnesses $wit_{A_j} = f\left(ACC_j, x_A^{-1}\right) = g^{z_j \times x_A^{-1}} \bmod n$ ($j = 1, 2$). Thus, Alice can convince the **AS** of the validity of her membership and gains access to services S_1 and S_2 by providing the corresponding witnesses wit_{A_1} and wit_{A_2} on the basis of the zero- knowledge proof technique. The **AS** must obtain the current accumulators ACC_1 and ACC_2 of services S_1 and S_2 and verify whether Alice is qualified to access these services through Equation (13):

$$\left(wit_{A_1} \times wit_{A_2}\right)^{x_A} \equiv \left(g^{(z_1+z_2) \times x_A^{-1}}\right)^{x_A} \equiv g^{(z_1+z_2)} \equiv ACC_1 \times ACC_2 \left(\bmod n\right) \qquad (13)$$

4. Performance and Security Analysis

This section verifies our claim of an efficient, anonymous, and unlinkable membership authentication scheme. We first detail the security properties provided by our scheme. Note that some properties have been detailed in the aforementioned sections.

Resistance of replay attack

An adversary may attempt to resend a stolen membership tuple to pass verification. Recall that **AS** accepts a membership proof if Equation (8) holds (one of the necessary conditions); thus, resending a stolen membership tuple would increase the time of ($T_V - T_i$) and therefore the adversary cannot pass the verification.

Membership nontransferability

Recall that a valid member U_i can send the transformed private key to their friend U_x by adding a random number γ as $\left\{\gamma a_i P, \gamma^{-1} b_i Q\right\}$. The unprivileged party U_x can prove their validity to the **AS** by computing $\left\{\mathbb{P}, \mathbb{Q}, A, B, T_x\right\}$ with the transformed private key. Through detection of illegal privilege transfer, as described in Section 3.3.1, the membership of any traitor who is involved in unauthorized privilege transfer will be revoked by the **KGC**. This can force the member U_i not to share their private key and privilege with others; otherwise, any two of U_i's service requests can be linked and their anonymity will be ruined.

The following two lemmas from [29] [30] must be given before demonstrating theorems related to our scheme.

Lemma 1. The GPI is not harder than either FAPI-1 or FAPI-2.

Proof.

(**FAPI-1** $\Rightarrow$ **GPI**:)

Given a GPI instance Y and an element $\mathcal{U}_1 \in_R \mathbb{G}_1$ as input, call the FAPI-1 oracle and output $\mathcal{U}_2 \in \mathbb{G}_2$; the FAPI-1 solver can solve the GPI problem.

(**GPI** $\nRightarrow$ **FAPI-1**:)

By contrast, given an FAPI-1 instance as input, the GPI solver cannot solve the FAPI-1 problem.

We can similarly prove that GPI ≤ FAPI-2.

Lemma 2. If FAPI-*j* is solvable, then the computational Diffie-Hellman (CDH) problem in $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ is solvable.

Proof.

Let $\mathcal{O}_j$ be the oracle to solve FAPI-*j*, for *j* = 1 or 2. Recall that $\mathcal{O}_1(\mathcal{U}_1, z)$ returns $\mathcal{U}_2 \in \mathbb{G}_2$ and $\mathcal{O}_2(\mathcal{U}_2, z)$ returns $\mathcal{U}_1 \in \mathbb{G}_1$ such that $\hat{e}(\mathcal{U}_1, \mathcal{U}_2) = z$. Suppose that ($\mathcal{U}_1$, $a\mathcal{U}_1$, $b\mathcal{U}_1$) is a CDH input in $\mathbb{G}_1$. Choose a $\mathcal{U}_2 \in \mathbb{G}_2$ and compute $z' = \hat{e}(a\mathcal{U}_1, \mathcal{U}_2)$. Call FAPI-1 solver $\mathcal{O}_1(\mathcal{U}_1, z')$ to obtain $a\mathcal{U}_2$. Finally, compute $z'' = \hat{e}(b\mathcal{U}_1, a\mathcal{U}_2)$ and call FAPI-2 solver $\mathcal{O}_2(\mathcal{U}_2, z'')$ to obtain $ab\mathcal{U}_1$.

The other two cases (solving CDH in $\mathbb{G}_2$ and $\mathbb{G}_T$) are similar.

Theorem 3 (Private key forgery freeness)**.** Let $\mathcal{A}_f$ be a polynomial-time adversary who is not in the group and who is assigned a parameter $\mathcal{G} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, q, \hat{e}, P, Q, Y)$. The proposed scheme can withstand $\mathcal{A}_f$ from private key forgery through the following manners of attack:

1) computing $\mathcal{U}_1 \in \mathbb{G}_1$ and $\mathcal{U}_2 \in \mathbb{G}_2$ with $\hat{e}(\mathcal{U}_1, \mathcal{U}_2) = Y$

2) choosing an element $\mathcal{U}_1 \in_R \mathbb{G}_1$ and finding $\mathcal{U}_2 \in \mathbb{G}_2$ with $\hat{e}(\mathcal{U}_1, \mathcal{U}_2) = Y$

3) choosing an element $\mathcal{U}_2 \in_R \mathbb{G}_2$ and finding $\mathcal{U}_1 \in \mathbb{G}_1$ with $\hat{e}(\mathcal{U}_1, \mathcal{U}_2) = Y$

4) extracting the group secret key *X* from the group public key *Y*.

Proof.

We discuss these cases of possible forgery in the following.

Case 1:

An adversary $\mathcal{A}_f$ may attempt to find two elements $\mathcal{U}_1 \in_R \mathbb{G}_1$ and $\mathcal{U}_2 \in \mathbb{G}_2$ such that $\hat{e}(\mathcal{U}_1, \mathcal{U}_2) \equiv \hat{e}(P, Q)^X \equiv Y$, for an unknown *X*. For $\mathcal{A}_f$ to forge an eligible private key pair $(\mathcal{U}_1, \mathcal{U}_2)$ with a nonnegligible probability $v(k)$ as follows is computationally infeasible:

$$\begin{aligned} Pr\Big[& P \leftarrow \mathbb{G}_1; Q \leftarrow \mathbb{G}_2; (\mathcal{U}_1, \mathcal{U}_2) \leftarrow \mathcal{A}_f\left(P, Q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{Z}_q^*\right) : \mathcal{U}_1 \in \mathbb{G}_1, \\ & \mathcal{U}_2 \in \mathbb{G}_2; \hat{e}(\mathcal{U}_1, \mathcal{U}_2) = \hat{e}(P, Q)^X = Y \Big] = v(k). \end{aligned} \tag{14}$$

The success of $\mathcal{A}_f$ can be used to solve the GPI problem defined in Section 3.1.

Cases 2 & 3:

From *Lemmas* 1 and 2, we know that 1) the GPI problem is not harder than either FAPI-1 or FAPI-2 and 2) if FAPI-*j* (where *j* = 1 or 2) is solvable, then the CDH problem is solvable. That is, an $\mathcal{A}_f$ who can succeed in cases 2 and 3 can be used to solve the problems of GPI and CDH.

Case 4:

Similarly, if $\mathcal{A}_f$ can succeed in extracting the group secret key *X* from the group public key $Y = \hat{e}(P,Q)^X$ without knowledge of *k* prime numbers, then it can be used directly to solve the discrete logarithm (DL) problem in $\mathbb{G}_T$. Moreover, such an adversary $\mathcal{A}_f$ can create any private key pair by computing $a_f \in \mathbb{Z}_q^*$ and $b_f = X/a_f$. That is, pairing inversion can be computed efficiently by $\mathcal{A}_f$ in known pairing groups. This case has been discussed in [31] [32] and is known as the MOV reduction in that the DL in $\mathbb{G}_1$ and $\mathbb{G}_2$ can be solved in polynomial time if a DL oracle exists for $\mathbb{G}_T$. Breaking the DL in $\mathbb{G}_1$, $\mathbb{G}_2$, or $\mathbb{G}_T$ is clearly harder than FAPI-*j* because intuitively breaking FAPI-*j* (where *j* = 1 or 2) involves only computing a group element in $\mathbb{G}_1$ or $\mathbb{G}_2$ and does not directly provide a method for recovering an exponent in $\mathbb{G}_1$, $\mathbb{G}_2$, or $\mathbb{G}_T$. □

From this reasoning, we know GPI ≤ FAPI-*j* = CDH ≤ $\mathrm{DL}_{\mathbb{G}_T}$. A straightforward approach for $\mathcal{A}_f$ is to forge an eligible private key by using the method of Case 1. However, no efficient algorithm is available that can break the GPI with a nonnegligible probability $v(k)$. In summary, we conclude that our proposed scheme is secure against any possible private key forgery. Additionally, forged memberships generated using the aforementioned approaches will be recognized by the proposed scheme for detecting illegal privilege transfer described in Section 3.3.1, if the **AS** reports suspicious membership to the **KGC** for investigation.

Theorem 4 (Resistance against membership impersonation). Let $\mathcal{A}_m$ be a polynomial-time adversary with a valid private key. Given the system parameter $\mathcal{G}$, the proposed scheme can withstand an adversary $\mathcal{A}_m$ engaging in a private key impersonation aimed at the other eligible members U_i or the new member U_k.

Proof.

If the adversary $\mathcal{A}_m$ has ever joined the group, they may attempt to add a random number γ to the private key $(a_m P, b_m Q)$ they received previously to find the private key $(a_i P, b_i Q)$ of an eligible member U_i such that $\gamma a_m P = a_i P$ and $\gamma^{-1} b_m Q = b_i Q$. Clearly, this attack cannot be recognized by the proposed illegal privilege transfer detection mechanism. This problem can be reduced to the selection of parameter *k* described in Section 3.2. We assume that *k* = 200, which yields $C(200,100) = 10^{59}$ combinations of the value a_i. The probability that two assignments of the value a_i for different members are identical is approximately 1/10^{59}, which is also the probability that the adversary

$\mathcal{A}_m$ can succeed in computing the valid private keys of eligible members $x_v \in Y$ and is negligible. □

Finally, Table 1 illustrates the substantial improvement of our proposed scheme compared with other schemes for the security concerns that we mentioned and defined in Section 2.1. Our proposed scheme features strong unlinkability, nontransferability, dynamic membership, and traitor detection. Table 2 shows the main enhancement in efficiency achieved by our scheme. The computational cost of our scheme comprises that of the presented anonymous authentication scheme and the dynamic reversed accumulator of Kuo *et al.* [21].

5. Conclusion

We propose a novel membership authentication scheme through which anonymity, strong unlinkability, and illegal privilege transfer detection are

Table 1. Comparison of security requirements.

Security requirements	[12]	[13]	Proposed scheme
Anonymity	Yes	Yes	Yes
Unforgeability	Yes	Yes	Yes
Strong/weak unlinkability	Weak unlinkability	Weak unlinkability	Strong unlinkability
Nontransferability	No	No	Yes
Exculpability	Yes	Yes	No
Dynamic membership	Yes	No	Yes
Traitor detection	No	No	Yes

Table 2. Comparison of computational costs.

Algorithms	[12]	[13]	Proposed scheme
Setup	$1E_n + 1M_n$	$1E_n + 1M_n$	$1E_T + E_n + (t-1)M_q$ $+1M_n$
Join	$(2\hat{j}+7)E_n + (\hat{j}+2)I_e$ $+(3\hat{j}+3)M_e + 3M_n$ $+3A_e + 2S_e + 2PK$	$6E_n + 1I_e + 1E_e + 1M_e$ $+3M_n + 2A_e + 2PK$	$1E_n + 1I_e + 1I_q$ $+\left(\frac{t}{2}-1\right)M_q + 1M_e$
Sign	$3E_n + 1M_n + 1PK$	$6E_n + 1E_e + 1I_e$ $+2M_n + 1PK$	$4E_1 + 1E_2 + 1I_e$ $+4M_q + 1H$
Verify	$6E_n + 2I_n + 8S_n + 3M_n$	$4E_e + (\tilde{t}+11)E_n + 2I_n + 4M_e$ $+7M_n + 4S_e + 1H + 1PK$	$2P + 1E_1 + 1M_T + 1PK$
Revoke	$1E_n$	$1E_n$	$1E_n + 1I_e$
Membership update			
member insertion	None	Reissuing membership	None
member revocation	None	Reissuing membership	$1Eu + 2E_n + 1M_n$

* $t/\tilde{t}$: number of eligible/revoked members, $\hat{j}$: time period, *H*: hash operation, $E_1/E_2/E_T/E_n$: exponentiation in $\mathbb{G}_1/\mathbb{G}_2/\mathbb{G}_T/\mathbb{Z}_n$, $I_n/I_e/I_q$: inverse modulo $n/\phi(n)/q$, M_e/M_q : multiplication modulo $\phi(n)/q$, $M_1/M_T/M_n$: multiplication in $\mathbb{G}_1/\mathbb{G}_T/\mathbb{Z}_n$, A_e/S_e : addition/subtraction over exponent, *P*: pairing operation, S_n : squaring operation modulo *n*, *Eu*: extended Euclidean, and *PK*: proof of knowledge.

provided. As aforementioned discussion, our proposed scheme can perform more efficiently if the symmetric setting of bilinear map groups is applied. By employing an efficient dynamic reversed accumulator, system members can prove their membership exclusiveness of the CRL to the verifier. Additionally, the practicality and attractiveness of our proposed scheme is supported.

References

[1] Malik, Federal Financial Institutions Examination Council (2001) Authentication of Internet Banking Environment. http://www.ffiec.gov

[2] Hu, V.C., Ferraiolom, D., Schnitzer, A., Sandlin, K., Miller, R. and Scarfone, K. (2013) Guide to Attribute Based Access Control (ABAC) Definition and Considerations. NIST Special Publication.

[3] Chaum, D. and van Heyst, E. (1991) Group Signatures. *Advances in Cryptology—EUROCRYPT*'91, **547**, 257-265.

[4] Camenisch, J. (1997) Efficient and Generalized Group Signatures. *Advances in Cryptology—EUROCRYPT*'97, **1233**, 465-479.

[5] Camenisch, J. (1998) Group Signature Schemes and Payment Systems Based on the Discrete Logarithm Problem. Ph.D. Dissertation, Swiss Federal Institute of Technology, Zürich.

[6] Boneh, D., Boyen, X. and Shacham, H. (2004) Short Group Signatures. *Advances in Cryptology—CRYPTO*'04, **3152**, 41-55.

[7] Fujisaki, E. and Okamoto, T. (1997) Statistical Zero Knowledge Protocols to Prove Modular Polynomial Relations. *Advances in Cryptology—CRYPTO*'97, **1294**, 16-30.

[8] Lee, Y.K., Lee, S., Lee, S.J., Hwang, J.Y., Chung, B.H. and Lee, D.G. (2010) Anonymous Access Control Framework Based on Group Signature. *Proceedings of the* 2*nd International Conference on Information Technology Convergence and Services*, Cebu, 11-13 August 2010, 1-5.

[9] Zheng, H., Zhao, Z. and Zhang, X. (2012) Access Control Based on Group Signatures in Cloud Service. *IEEE International Conference on Computer Science and Automation Engineering*, **2**, 316-320.

[10] Hu, X. (2014) Cost-Effective Scalable and Anonymous Certificateless Remote Authentication Protocol. *IEEE Transactions on Information Forensics and Security*, **9**, 2327-2339.

[11] He, D., Zeadally, S., Kumar, N. and Lee, J.H. (2016) Anonymous Authentication for Wireless Body Area Networks with Provable Security. *IEEE Systems Journal*, **11**, 2590-2601.

[12] Song, D. (2001) Practical Forward-Secure Group Signature Schemes. *Proceedings of ACM Symposium on Computer and Communication Security*, Philadelphia, 5-8 November 2001, 225-234. https://doi.org/10.1145/501983.502015

[13] Ateniese, G., Song, D. and Tsudik, G. (2002) Quasi-Efficient Revocation of Group Signatures. *Proceedings of the* 6*th International Conference on Financial Cryptography*, Southampton, 11-14 March 2002, 183-197. https://doi.org/10.1007/3-540-36504-4_14

[14] Chor, B., Fiat, A. and Naor, M. (1994) Tracing Traitors. *Advances in Cryptology*, Wollongong, 28 November-1 December 1994, 257-270.

[15] Mitsunari, S., Sakai, R. and Kasahara, M. (2002) A New Traitor Tracing. *IEICE*

Transactions on Fundamentals, **E85-A**, 481-484.

[16] Tô, V.D., Safavi-Naini, R. and Zhang, F. (2003) New Traitor Tracing Schemes using Bilinear Map. *Proceedings of ACM Workshop on Digital Rights Management*, Washington DC, 27 October 2003, 67-76.

[17] Boneh, D., Sahai, A. and Waters, B. (2006) Fully Collusion Resistant Traitor Tracing with Short Ciphertexts and Private Keys. *Advances in Cryptology*, Santa Barbara, 20-24 August 2006, 573-592.

[18] Boneh, D. and Naor, M. (2008) Traitor Tracing with Constant Size Ciphertext. *Proceedings of* 15*th ACM Conference on Computer and Communications Security*, Alexandria, 27-31 October 2008, 501-510. https://doi.org/10.1145/1455770.1455834

[19] Camenisch, J. and Lysyanskaya, A. (2002) Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials. *Advances in Cryptology*, Santa Barbara, 18-22 August 2002, 61-76.

[20] Camenisch, J., Kohlweiss, M. and Soriente, C. (2009) An Accumulator Based on Bilinear Maps and Efficient Revocation for Anonymous Credentials. *Proceedings of the* 12*th International Conference on Practice and Theory in Public Key Cryptography*, Irvine, 18-20 March 2009, 481-500. https://doi.org/10.1007/978-3-642-00468-1_27

[21] Kuo, T.M., Yen, S.M. and Han, M.C. (2017) Dynamic Reversed Accumulator. *International Journal of Information Security*, 1-9. https://doi.org/10.1007/s10207-017-0360-6

[22] Benaloh, J. and de Mare, M. (1993) One-Way Accumulators: A Decentralized Alternative to Digital Signatures. *Advances in Cryptology*, Santa Barbara, 22-26 August 1993, 274-285.

[23] Nguyen, L. (2005) Accumulators from Bilinear Pairings and Applications. *Proceedings of the Cryptographers' Track at the RSA Conference* 2009 *on Topics in Cryptology*, San Francisco, 20-24 April 2009, 275-292. https://doi.org/10.1007/978-3-540-30574-3_19

[24] Li, J., Li, N. and Xue, R. (2007) Universal Accumulators with Efficient Non-Membership Proofs. *Proceedings of the* 5*th International Conference on Applied Cryptography and Network Security*, Zhuhai, 5-8 June 2007, 253-269. https://doi.org/10.1007/978-3-540-72738-5_17

[25] Au, M.H., Tsang, P.P., Susilo, W. and Mu, Y. (2009) Dynamic Universal Accumulators for DDH Groups and Their Application to Attribute-Based Anonymous Credential Systems. *Proceedings of the Cryptographers' Track at the RSA Conference* 2009 *on Topics in Cryptology*, San Francisco, 20-24 April 2009, 295-308. https://doi.org/10.1007/978-3-642-00862-7_20

[26] Mashatan, A. and Vaudenay, S. (2013) A Fully Dynamic Universal Accumulator. *Proceedings of the Romanian Academy*, **14**, 269-285.

[27] Barić, N. and Pfitzmann, B. (1997) Collision-Free Accumulators and Fail-Stop Signature Schemes without Trees. *Advances in Cryptology*, Santa Barbara, 17-21 August 1997, 480-494.

[28] Miller, V. (2004) The Weil Pairing, and Its Efficient Calculation. *Journal of Cryptology*, **17**, 235-261.

[29] Galbraith, S., Hess, F. and Vercauteren, F. (2008) Aspects of Pairing Inversion. *IEEE Transactions on Information Theory*, **54**, 5719-5728.

[30] Kiraz, M.S. and Uzunkol, O. (2016) Still Wrong Use of Pairings in Cryptography. Cryptology ePrint Archive, Report 2016/223. https://eprint.iacr.org/2016/223

[31] Frey, G. and Rück, H.G. (1994) A Remark Concerning m-divisibility and the Discrete Logarithm in the Divisor Class Group of Curves. *Mathematics of Computation*, **62**, 865-874.

[32] Menezes, A., Okamoto, T. and Vanstone, S. (1993) Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field. *IEEE Transactions on Information Theory*, **39**, 1639-1646.