A Group Signature Scheme with Signature Claiming and Variable Linkability

He Ge Stephen R. Tate
Department of Computer Science and Engineering
University of North Texas
Denton, TX 76203

Abstract

In this paper we present a new system for anonymous authentication, which allows a user to prove possession of an authenticated credential while not revealing his or her identity. Our system is an extension of an existing group signature scheme, providing additional features drawn from work on traceable signatures and direct anonymous attestation. Our first extension, a property taken from traceable signatures, allows a signer to later "claim" a signature, proving that they are indeed the individual who created this signature. Our second extension, taken from the work on direct anonymous attestation, allows a prover and verifier to agree on the degree of linkability of transactions, ranging from completely unlinkable transactions, to being able to link transactions made within a fixed time period (like a day), to completely linkable transactions. Our scheme is efficient enough to be practical, and is proved to be secure under the strong RSA assumption and the decisional Diffie-Hellman assumption.

Keywords: Group Signature, Privacy, Anonymous Authentication, Cryptographic Protocol.

1 Introduction

In this paper, we present new techniques for performing authentication, one of the five fundamental security services in information assurance (the others being confidentiality, integrity, availability, and non-repudiation) [16]. The CNSS Information Assurance Glossary [16] defines authentication as a "security measure designed to establish the validity of transmission, message, or originator, or a means of verifying an individual's authorizations to receive specific categories of information." Most authentication services in use today work by using authentication to link a user to an identity, and then looking up authorizations for that individual in an access control database. However, as the CNSS definition makes clear, the goal is to determine authorization — determining identity is often not necessary and in some cases revealing this information is undesirable. With this in mind, there has been a large amount of work in the cryptographic research community on anonymous authentication systems, in which authenticated users receive credentials from a designated group manager, and in later interactions a user can prove possession of such a credential in a privacy-preserving manner.

The most heavily studied type of anonymous authentication system is the "group signature scheme," which provides a well-defined set of services and security guarantees that we describe in more detail below (recent prominent work in group signatures includes [12, 11, 1, 2, 8, 5, 6][1]). However, several authors have identified various desirable properties not provided by the group signature definition, and have introduced variants of this basic scheme including work on "anonymous credential systems" [9, 10], "traceable signatures" [21] and a system designed for trusted computing platforms called "direct anonymous attestation" [7]. Our contribution in this paper is to show how the group signature scheme of Camenisch and Michels [11] can be modified so that it supports two particularly useful extensions, one from traceable signatures that allows a signer to later claim a particular signature, and one from the work on direct anonymous attestation that allows a prover and verifier to agree on a variable degree of signature linkability. Our modifications to the Camenisch and Michels scheme replace operations with modified formulas that have the same computational complexity, so our system preserves the efficiency of the Camenisch and Michels scheme while providing a unique set of extended features which is useful in many situations.

1.1 Background

A group signature is a privacy-preserving signature scheme introduced by Chaum and Heyst in 1991 [15]. In such a scheme, a group member can sign a message on behalf of the group without revealing his identity. Only the the specified open authority (who may or may not be the group

[1] An extensive bibliography of group signature literature can be found at http://www.i2r.a-star.edu.sg/icsd/staff/guilin/bible/group-sign.htm

manager) can open a signature and find its originator. Signatures signed by the same user cannot be identified as from the same source, i.e, "linked". Recently, the study of group signature schemes has attracted considerable attention, and many solutions have been proposed in the literature (e.g., [12, 11, 1, 2, 8, 5, 6]). Creating an anonymous authentication scheme from a group signature is simple: the group is simply the set of authorized users, and authentication is performed by a group member placing a group signature on a challenge (nonce) sent by the service requiring authentication. From the properties of group signatures, all the service or an attacker can learn is that the signature was made by a valid group member (i.e., an authorized user).

Of course, anonymous authentication has a dangerous side effect: anonymous corrupted users. Since all users are anonymous, a mechanism is needed to identify corrupted users, effectively and fairly. To do so, Kiayias *et al.* have recently proposed a variant of group signatures, called traceable signatures [21]. They define "traceability" as the ability to identify signatures signed by a specified group member (based on some per-user secret information kept by the group manager) without requiring the open authority to open them. Tracing can be done by "trace agents" distributively and efficiently. With a group signature, this cannot be done fairly since opening all signatures violates the privacy of innocent group members. Kiayias *et al.* also introduced the concept of "self-traceability," or "claiming." That is, a group member himself can stand out, claiming a signature signed by himself without compromising his other signatures and secrets.

To understand one problem with signature claiming, consider that many group signature schemes work by having the signer include their identity in the signature, encrypted using a semantically secure encryption algorithm and the open authority's public key. For example, consider a system in which the open authority uses El Gamal encryption over a cyclic group with generator g that provides semantic security (such as a subgroup of Z_p^* of prime order), where the open authority has public key y. Then the signature created by a party with identity I could include the El Gamal encryption of the identity (Iy^r, g^r), where r is a random exponent. The open authority can clearly open this to reveal the identity if necessary. Furthermore, if the signer keeps a record of all the random r values that she used, then a signature can be claimed by simply revealing the r value so that any user can decrypt the identity. However, keeping the complete list of all used r values is both inefficient and a security risk, so our challenge is to support signature claiming without requiring the group member to keep a record of its random values.

Another variant of group signatures has been adopted by the Trusted Computing Group [22], an industry group developing standards for "trusted computing platforms." A trusted computing platform is a computing device integrated with a cryptographic chip called the trusted platform module (TPM). The TPM is designed and manufactured in a specific way such that all parties can trust cryptographic computing results from this TPM. A trusted computing platform can implement many security related features based on TPM, such as secure boot, sealed storage, and software integrity attestation.

However, deployment of TPMs introduces privacy concerns. Each TPM is loaded with its own unique public key pair called its "endorsement key," and the most straightforward use of a TPM would reveal this unique public key to remote parties during attestation operations. Thus, different servers could cooperate with each other to link the transactions made by the same TPM. To protect the privacy of a TPM owner, it is desirable to carry out anonymous authentication, i.e, a TPM can prove its authenticity to a remote server without disclosing its identifier. Such a mechanism has been implemented by the technique called Direct Anonymous Attestation (DAA) in [7]. DAA is basically a group signature scheme; however, since it is important for a user to have trust in such an anonymous attestation scheme, there is no open authority or capability for signatures to be opened.

In addition, DAA introduces the notion of "variable anonymity," which is conditionally linkable anonymous authentication: the same TPM will produce linkable signatures for a certain period of time. The period of time during which signatures can be linked can be determined by the parties involved and can vary from an infinitesimally short period (leading to completely unlinkable signatures) to an infinite period (leading to completely linkable signatures). Signatures made by the same user in different periods of time or to different servers cannot be linked. By setting the linkability period to a moderately short time period (a day to a week) a server can potentially detect if a key has been compromised and is being used by many different users, while still offering some amount of unlinkability.

1.2 Our Results

In the previous section we briefly introduced some of the available techniques for anonymous authentication. Numerous constructions with different features have been proposed to accommodate different settings. This raised the question which we address in this paper: Can we devise a construction which combines the features from different authentication primitives? More specifically, can we have a group signature scheme which also supports signature claiming and variable anonymity? So far as we know, no such scheme has been proposed to work in this manner, probably because variable anonymity is quite a new identified feature in anonymous authentication.

We consider variable anonymity to be particularly important when group signatures are used for anonymous authentication, since it is the only way key sharing violations can be detected. More specifically, while the standard group signature scheme can use the open authority to identify a user that performs malicious actions, consider what happens when one authorized user shares his authentication credential with a set of co-conspirators. For example, a large set of users could share a single subscription to some pay web site. Since all authentications are completely unlinkable in a group signature scheme, it would be impossible to determine whether 1000 requests coming in during a day are from 1000 different valid users or from 1000 people sharing a single valid credential. Introducing linkability for a limited time period is the only way to detect this, and if an unusually high number of requests using the same credential come in from different IP addresses during the same day, then this could be flagged as potentially malicious behavior, and the open authority could then open the signatures to determine the real owner of this credential for further investigation.

In this paper, we present our construction for a group signature scheme that supports both signature claiming and variable linkability. Our construction is built up from the group signature scheme due to Camenisch and Michels [11], which we will refer to in this paper as the "CM signature scheme." The added capabilities do not adversely affect the time complexity of the various operations, so we preserve the efficiency of the CM signature scheme while providing these additional features.

The rest of this paper is organized as follows. The next section introduces a concrete model for our signature scheme. Section 3 reviews some definitions, cryptographic assumptions, and building blocks of our proposed scheme. Section 4 presents the proposed scheme. Security properties are considered in Section 5. Finally, we summarize and give conclusions in section 6.

2 The Model

This section introduces the model for our signature scheme, which is a variant of the group signature model (e.g. [1]). Both these two models support procedures Setup, Join, Sign, Verify, and Open. Our signature scheme supports linkability identifiers in the sign protocol and supports additional procedures Claim (Self-trace) and Claim-Verify.

Definition 1 *A group signature scheme with signature claiming and variable linkability is a digital signature scheme with three types of participants: A group manager, an open authority, and group members. It consists of the following procedures:*

- **Setup***: For a given security parameter σ, the group manager produces system-wide public parameters and a group manager master key for group membership certificate generation.*
- **Join***: An interactive protocol between a user and the group manager. The user obtains a group membership certificate to become a group member. The public certificate and the user's identity information are stored by the group manager in a database for future use.*
- **Sign***: Using his group membership certificate and his private key, a group member creates an anonymous group signature for a message.*
- **Verify***: A signature is verified to make sure it originates from a legitimate group member without the knowledge of which particular one.*
- **Open***: Given a valid signature, an open authority discloses the underlying group membership certificate.*
- **Claim (Self-trace)***: A group member creates a proof that he created a particular signature.*
- **Claim_Verify***: A party verifies the correctness of the claiming transcript.*

Similar to a group signature, our signature scheme should satisfy the following properties:

- **Correctness**: Any valid signature can be correctly verified by the Verify protocol and a valid claiming proof can be correctly verified.
- **Forgery-Resistance**: A valid group membership certificate can only be created by a user and the group manger through Join protocol.
- **Anonymity**: It is infeasible to identify the real signer of a signature except by the open authority or if the signature has been claimed.
- **Unlinkability**: It is infeasible to link two different signatures of the same group member.
- **Non-framing**: No one (including the group manager) can sign a message in such a way that it appears to come from another user if it is opened.
- **Non-appropriation**: No one (including the group manager) can make a valid claim for a signature which they did not create.

3 Definitions and Preliminaries

This section reviews some definitions, widely accepted complexity assumptions, and building blocks that we will use in this paper.

3.1 Number-Theoretic Assumption

Definition 2 (Special RSA Modulus) *An RSA modulus $n = pq$ is called special if $p = 2p' + 1$ and $q = 2q' + 1$ where p' and q' also are prime numbers.*

Definition 3 (Quadratic Residue Group QR_n) *Let Z_n^* be the multiplicative group modulo n, which contains all positive integers less than n and relatively prime to n. An element $x \in Z_n^*$ is called a* quadratic residue *if there exists an $a \in Z_n^*$ such that $a^2 \equiv x \pmod{n}$. The set of all quadratic residues of Z_n^* forms a cyclic subgroup of Z_n^*, which we denote by QR_n. If n is the product of two distinct primes, then $|QR_n| = \frac{1}{4}|Z_n^*|$.*

Our signature protocol requires us to compute random generators of QR_n, and the following property shows that almost all elements of QR_n are in fact generators.

Property 1 *If n is a special RSA modulus, with p, q, p', and q' as in Definition 2 above, then $|QR_n| = p'q'$ and $(p'-1)(q'-1)$ elements of QR_n are generators of QR_n.*

Proof: Consider the group Z_p^* and corresponding subgroup QR_p. Since p is prime, exactly half of the elements of Z_p^* are in QR_p, so $|QR_p| = \frac{1}{2}(p-1) = p'$. Since p' is prime, any element of QR_p generates a subgroup of size 1 or of size p', and since only the identity element generates a subgroup of size 1, the remaining $p'-1$ elements of QR_p have order p'. Similarly, $q'-1$ elements of QR_q have order q'. By the Chinese Remainder Theorem, if element x_p has order m_p in Z_p^* and element x_q has order m_q in Z_q^*, then the unique element $x \in Z_n^*$ with $x_p = x \bmod p$ and $x_q = x \bmod q$ has order LCM(m_p, m_q) in Z_n^*. Furthermore, $x \in QR_n$ if and only if $x_p \in QR_p$ and $x_q \in QR_q$. Therefore, $(p'-1)(q'-1)$ elements of QR_n have order LCM$(p', q') = p'q' = |QR_n|$ in QR_n, and so are generators of QR_n. □

The security of our techniques relies on the following security assumptions which are widely accepted in the cryptography literature. (see, for example, [3, 19, 12, 13, 1]).

Assumption 1 (Strong RSA Assumption) *Let n be an RSA modulus. The* Flexible RSA Problem *is the problem of taking a random element $u \in Z_n^*$ and finding a pair (v, e) such that $e > 1$ and $v^e = u \pmod{n}$. The* Strong RSA Assumption *says that no probabilistic polynomial time algorithm can solve the flexible RSA problem with non-negligible probability.*

Assumption 2 (Decisional Diffie-Hellman Assumption for QR_n) *Let n be a special RSA modulus, and let g be a generator of QR_n. For two distributions (g, g^x, g^y, g^{xy}), (g, g^x, g^y, g^z), $x, y, x \in_R Z_n$, there is no probabilistic polynomial-time algorithm that distinguishes them with non-negligible probability.*

3.2 Building Blocks

Our main building blocks are *statistical honest-verifier zero knowledge proofs of knowledge* related to discrete logarithms over QR_n [14, 20, 13]. They include protocols for things such as the knowledge of a discrete logarithm, the knowledge of the equality of two discrete logarithms, the knowledge of the discrete logarithm that lies in certain interval, etc. We introduce one of them here. Readers may refer to the original papers for more details.

Definition 4 (Protocol 1) *Let n be a special RSA modulus, QR_n be the quadratic residue group modulo n, and g be a generator of QR_n. Let α, l, and l_c be security parameters that are all greater than 1, and let X be a constant number. In the following protocol, Alice knows x, the discrete logarithm of T_1 (so $g^x \equiv T_1 \pmod{n}$), where $x \in [X - 2^l, X + 2^l]$. After the protocol is executed, Bob is convinced that Alice knows the discrete log x of T_1 such that $x \in [X - 2^{\alpha(l+l_c)+1}, X + 2^{\alpha(l+l_c)+1}]$.*

1. *Alice picks a random $t \in \pm\{0,1\}^{\alpha(l+l_c)}$ and computes $T_2 = g^t \pmod{n}$. Alice sends (T_1, T_2) to a verifier Bob.*
2. *Bob picks a random $c \in \{0,1\}^{l_c}$ and sends it to Alice.*
3. *Alice computes*

$$w = t - c(x - X),$$

which she sends to Bob. Notice that an honest Alice knows a value of $x \in [X - 2^l, X + 2^l]$, so given the range in which t and c were selected, an honest Alice will produce a w that satisfies $w \in [-2^{\alpha(l+l_c)+1}, 2^{\alpha(l+l_c)+1}]$ (actually in a slightly smaller interval than this, but this is a sufficiently tight bound for our purposes).

4. *Bob checks that $w \in [-2^{\alpha(l+l_c)+1}, 2^{\alpha(l+l_c)+1}]$ and*

$$g^{w-cX} T_1^c \equiv T_2 \pmod{n}.$$

If both tests pass, then Bob is convinced that Alice knows the discrete logarithm of T_1 and that it lies in the range $[X - 2^{\alpha(l+l_c)+1}, X + 2^{\alpha(l+l_c)+1}]$.

Remark 1: The parameter $\alpha > 1$ is used since we do not know the size of the group QR_n, and determines the statistical closeness of our actual distribution to the ideal one. In other words, α determines the statistical zero-knowledge property of this protocol. For a more in-depth discussion and analysis, we refer the reader to [11].

Remark 2: The asymmetry of this proof — that Alice must know a value in a more restrictive range that Bob will be

convinced of — seems unusual at first, but as we will see in later sections this is sufficient to prove the properties that we need.

Remark 3: Using the Fiat-Shamir heuristic [18], the protocol can be turned into a non-interactive "signature of knowledge," which is secure in the random oracle model [4]. We will introduce our new signature scheme in the manner of a "signature of knowledge" in next section.

4 Our Group Signature Scheme

In this section, we describe our implementation of a group signature scheme that supports signature claiming and variable linkability. As mentioned earlier, this scheme is an enhanced version of the Camenisch and Michels group signature scheme [11], which we call the CM group signature scheme.

4.1 System Parameter Setting

The group manager picks a security parameter σ, and generates the system parameters as follows:

- n, g, h: n is a special RSA modulus such that $n = pq$, where p and q are each at least σ bits long (so $p, q > 2^{\sigma}$), and $p = 2p' + 1$, and $q = 2q' + 1$, with p' and q' both being prime. g, h are random generators of the cyclic group QR_n. n, g, h are public values while p and q are kept secret by the administrator. The bit-length of g, $l_g > 2\sigma$, is also publicly available.
- α, l_c, l_s: Security parameters that are greater than 1.
- X: A constant integer, $X > 2^{\alpha(l_s+l_c)+1}$.
- Two strong collision-resistant hash functions: $\mathcal{H}_1 : \{0,1\}^* \rightarrow Z_n^*$, and $\mathcal{H}_2 : \{0,1\}^* \rightarrow \{0,1\}^{l_c}$.

An illustration of the system parameters is the setting of $\sigma = 1024$ (so n is 2048 bits), $l_g = 2046$, $\alpha = 9/8$, $X = 2^{860}$, $l_s = 600$ and $l_c = 160$.

The open authority creates his ElGamal public keypair [17], i.e., a random private key x and corresponding public key y such that $y = h^x \mod n$. Note that the open authority may or may not be the same as the group manager. Furthermore, we can support multiple open authorities at the cost of increasing the length of our signatures or by having the open authorities share a key.

4.2 Join Protocol

We adopts the same Join protocol as in CM group signature. A group member obtains its group membership certificate as a keypair (E, s), such that s is prime, $s \in [X, X + 2^{l_s}]$, and

$$E^s \equiv g \pmod{n}.$$

s is the group member's private key and is kept secret by the group member. For further details on how the Join protocol works, see [11].

4.3 Sign Protocol

1. Compute a random element $j \in_R QR_n$ by squaring the hash of the verifier's base name bsn and the current linkability interval counter ct, so

   $$j = (\mathcal{H}_1(bsn_v||ct))^2 \mod n.$$

   For different verifiers (with different base names) or different linkability intervals (different values of ct), different and independent j values will be computed. Due to Property 1, the probability that j is not a generator of QR_n is negligible.

2. Generate a random blinding integer $b \in_R \{0,1\}^{l_g}$ and compute:

   $$T_1 = Ey^b \pmod{n},\ T_2 = h^b \pmod{n},$$
   $$T_3 = j^s \pmod{n}.$$

3. Randomly choose $t_1 \in_R \{0,1\}^{\alpha(l_s+l_c)}, t_2 \in_R \{0,1\}^{\alpha(l_g+l_s+l_c)}$, and $t_3 \in_R \{0,1\}^{\alpha(l_s+l_c)}$ and

   - Compute (all computations done modulo n) $d_1 = T_1^{t_1}/y^{t_2}$, $d_2 = T_2^{t_1}/h^{t_2}$, and $d_3 = j^{t_1}$;
   - $c = \mathcal{H}_2(g||h||y||j||T_1||T_2||T_3||d_1||d_2||d_3||m)$, where m is a message to be signed.
   - $w_1 = t_1 - c(s - X),\ w_2 = t_2 - csb$.

4. Output the signature $(c, w_1, w_2, T_1, T_2, T_3)$ on message m.

Remark: Note that the main difference between our method and CM group signatures is the computed value T_3 — the corresponding value in the CM group signature scheme (denoted d in their notation) is computed as $h^s j^b$, where j is a fixed generator in their scheme.

4.4 Verify Protocol

1. Compute the same generator j, and

   $$c' = \mathcal{H}_2(g||h||y||j||T_1||T_2||T_3||g^c T_1^{w_1-cX}/y^{w_2}||$$
   $$T_2^{w_1-cX}/h^{w_2}||j^{w_1-cX}T_3^c||m)$$

2. Accept the signature if and only if $c = c'$ and $w_1 \in \pm\{0,1\}^{\alpha(l_s+l_c)+1}$, $w_2 \in \pm\{0,1\}^{\alpha(l_g+l_s+l_c)+1}$.

4.5 Claim and Claim_Verify

A group member uses Protocol 1 (see Definition 4) to claim his signature by proving knowledge of the discrete logarithm s of T_3 w.r.t. base j, and proving that s lies in the range of valid private keys.

4.6 Open Protocol

For a valid signature, an open authority can open the signature to find its originator as follows

$$E = T_1/T_2^x \pmod{n}.$$

For the non-framing property, the open authority must also issue a proof that it correctly revealed the group member, which can be done identically to the method used by the CM group signature.

4.7 Variable Linkability

Variable linkability is controlled by the random generator j. If each time a group member chooses a different value for j, complete unlinkability is realized. If certain values are fixed for a period of time (for example, verifier may require time value ct be fixed for a day, then j keeps unchanged in the same day), the same group member will always produce the same T_3. As a result, linkability is achieved in this time interval. If ct never changes, thus always giving the same j value for a given verifier, then a pseudo-anonymity system results.

To provide further flexibility, we can extend the current construction to a DAA-like scheme for trusted computing platforms [7]. To do this, we simply use a value y for which no one knows the discrete log, effectively disabling the open capability. This mirrors the DAA requirement in which opening is forbidden. Furthermore, a variant of the claiming ability can be used for rouge TPM tagging. The current TPM revocation method supported by DAA includes the use of a revocation list that includes any discovered private keys of compromised TPMs. Later, a verifier can identify a rouge TPM by checking the list. For our construction, the private key s should be published on the list. Verifiers can check whether

$$j^s =? T_3 \pmod{n}$$

for all revoked s on the list to identify rouge TPMs (group members).

5 Security Properties

Our scheme uses the same certificate as in CM group signature, so properties that depend only on the form of the membership certificate such as forgery resistance and non-framing are unaffected by our changes. We have changed their Sign and Verify protocols, so in this section we prove that the basic security properties provided by these protocols are met.

For the security of certificates, we have the theorem as follows.

Theorem 1 (Forgery-resistance) *Under the restriction $X > 2^{\alpha(l_s+l_c)+1}$, a pair (E, s), such that $s \in [X - 2^{\alpha(l_s+l_c)+1}, X + 2^{\alpha(l_s+l_c)+1}]$ and $E^s = g \pmod{n}$ can only be produced by Join protocol in [11] under the strong RSA assumption. A valid group membership (E, s) falls in this range.*

Next, we introduce two lemmas which will be used shortly for the proof the scheme.

Lemma 1 *Let n be an integer. Given values $u, v \in Z_n^*$ and $x, y \in Z$ such that $GCD(x, y) = r$, and $v^x \equiv u^y \pmod{n}$, there is an efficient way to compute a value z such that $z^k \equiv u \pmod{n}$, where $k = x/r$.*

Proof: Since $GCD(x, y) = r$, using the extended Euclidean GCD algorithm, we can obtain values α and β such that $\alpha x/r + \beta y/r = 1$. Then we have

$$u \equiv u^{\alpha x/r+\beta y/r} \equiv u^{\alpha x/r} u^{y\beta/r} \equiv u^{\alpha x/r} v^{\beta x/r}$$

$$\equiv (u^\alpha v^\beta)^{x/r} \pmod{n}.$$

Therefore, setting $k = x/r$ and $z = u^\alpha v^\beta$, we have $z^k \equiv u \pmod{n}$. □

Lemma 2 *Under the strong RSA assumption, if there exists a probabilistic polynomial-time algorithm that takes an RSA modulus n and a value u and succeeds with non-negligible probability in finding values v, x, and y, such that $v^x \equiv u^y \pmod{n}$, then x divides y.*

Proof: By contradiction. Assume that there exists a probabilistic polynomial-time algorithm that takes an RSA modulus n and a value u and succeeds with non-negligible probability in finding values v, x, and y, such that $v^x \equiv u^y \pmod{n}$, but for which x does *not* divide y. Let $r = GCD(x, y)$. Since x does not divide y we have $r < x$, and so $x/r > 1$. By Lemma 1 we can find a z such that $z^k = u \pmod{n}$, with $k = x/r > 1$, which is a solution to the flexible RSA problem. However, this contradicts the strong RSA assumption, which says that no such algorithm can exist. □

We address the security of Sign and Verify protocol in the following theorem.

Theorem 2 *Under the strong RSA assumption, the interactive protocol underlying our signature scheme is a statistical honest-verifier zero-knowledge proof of knowledge of a valid group certificate (E, s) such that $E^s = g \pmod n$, and s lies in the correct interval.*

Proof: [Sketch] The proof for correctness is straightforward. The zero-knowledge property of our protocol relies on two aspects: the random choice of our blinding number b, and random generation of a generator j. The zero-knowledge related to T_1, T_2 follows the method in original paper. T_3 is constructed from a random generator j produced from a strong collision-resistant hash function $\mathcal{H}_1$. Since j is uniformly distributed over QR_n, and the length of s guarantees that $GCD(s, p'q') = 1, T_3$ is also uniformly distributed over QR_n, so a simulator can generate this same distribution. Then the zero-knowledge property for s in the form $T_3 = j^s \pmod n$ follows from the decisional Diffie-Hellman assumption over QR_n. A full proof giving an explicit simulator for the protocol appears in the full version of this paper.

We now address the proof of knowledge part. We demonstrate that a knowledge extractor is able to recover the group certificate when it has found two accepting tuples under the same commitment and different challenges from a verifier.

Let $(T_1, T_2, T_3, d_1, d_2, d_3, c, w_1, w_2)$ and $(T_1, T_2, T_3, d_1, d_2, d_3, c', w_1', w_2')$ be such tuples.

Since $d_3 \equiv j^{w_1 - cX} T_3^c \equiv j^{w_1' - c'X} T_3^{c'} \pmod n$, we have

$$j^{(w_1' - w_1) + (c - c')X} \equiv T_3^{c - c'} \pmod n.$$

By Lemma 2, under the strong RSA assumption, $c - c'$ has to divide $(w_1' - w_1) + (c - c')X$. Then we obtain

$$\tau_1 = (w_1' - w_1)/(c - c') + X. \quad (1)$$

Similarly, $d_2 \equiv T_2^{w_1 - cX}/h^{w_2} \equiv T_2^{w_1' - c'X}/h^{w_2'} \pmod n$, so we have

$$h^{w_2' - w_2} \equiv T_2^{(w_1' - w_1) + (c - c')X} \equiv (T_2^{\tau_1})^{c - c'} \pmod n,$$

where Lemma 2 again applies to show that $c - c'$ must divide $(w_2' - w_2)$. Therefore, we can compute

$$\tau_2 = (w_2' - w_2)/(c - c'),$$

such that $T_2^{\tau_1} \equiv h^{\tau_2} \pmod n$. Applying Lemma 2 once again to this formula, we see that τ_1 must divide τ_2, so we can compute $\tau_3 = \tau_2/\tau_1$.

Since $d_1 \equiv g^c T_1^{w_1 - cX}/y^{w_2} \equiv g^{c'} T_1^{w_1' - c'X}/y^{w_2'} \pmod n$, we have

$$g^{c - c'} \equiv T_1^{w_1' - w_1 + (c - c')X} y^{w_2 - w_2'} \pmod n.$$

Thus we have

$$g \equiv T_1^{\tau_1}/y^{\tau_2} \equiv T_1^{\tau_1}/y^{\tau_1 \tau_3} \equiv (T_1/y^{\tau_3})^{\tau_1} \pmod n.$$

Since $\tau_1 = (w_1' - w_1)/(c - c') + X$, $w_1, w_1' \in \pm\{0,1\}^{\alpha(l_s + l_c) + 1}$, and $c, c' \in \{0,1\}^{l_c}$, it follows that $\tau_1 \in \pm\{0,1\}^{\alpha(l_s + l_c) + 1}$. Therefore τ_1 must be in the acceptable range for private key values in our system.

Finally, let $E = T_1/y^{\tau_3} \pmod n$ and $s = \tau_1$. We have demonstrated the existence of a knowledge extractor that can find (E, s), such that $E^s = g \pmod n$ and s lies in the appropriate range, so (E, s) is a valid membership certificate. □

Unlinkability follows the same argument as the CM group signature for T_1, T_2. Since we define a new T_3 in our traceable signature, we need to show this change still keeps the unlinkability property. Similar to the case in CM group signature, the problem of linking two tuples (j, T_3), (j', T_3') with $j \neq j'$ is equivalent to deciding the equality of the discrete logarithms of T_3, T_3' with bases j, j' respectively. This is assumed to be infeasible under the decisional Diffie-Hellman assumption over QR_n. Therefore, we have the following result.

Theorem 3 (Unlinkability) *Under the decisional Diffie-Hellman assumption over QR_n, and the strong collision property of hash function, there exists no probabilistic polynomial-time algorithm that can make the linkability decision for any two arbitrary tuples $(j, T_3), (j', T_3')$ with non-negligible probability.*

6 Conclusion

In this paper we have presented a group signature scheme which is an enhancement of CM group signatures [11] that supports additional features. The new construction supports signature claiming, in which a group member can voluntary remove the cloak of anonymity from one of their signatures. Our scheme also supports the important property of variable linkability, a property which comes from work on anonymous authentication for trusted computing platforms, and is vital for detecting key sharing in an anonymous authentication system. The "variable" part of this can be adjusted to provide a wide range of linkability properties, from completely unlinkable signatures, to signatures linkable within a fixed time period, to completely linkable signatures (giving what is essentially a fixed pseudonym system). In practice, the amount of linkability would be determined by a risk analysis of the application, balancing the goal of protecting a user's privacy against a providers goal of detecting inappropriate uses of keys. As our scheme supports the full range of linkability options, it provides the best available flexibility to users as well as providers. Finally, we have

proved that our new signature scheme is secure under the strong RSA assumption and the Decisional Diffie-Hellman assumption over QR_n.

References

[1] G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *Advances in Cryptology — Crypto*, pages 255–270, 2000.

[2] G. Ateniese, D. Song, and G.Tsudik. Quasi-efficient revocation in group signatures. In *Financial Cryptography'02*, pages 183–197, 2002.

[3] N. Baric and B. Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In *Advances in Cryptology — Eurocrypto*, pages 480–494, 1997.

[4] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient procotols. In *First ACM Conference On computer and Communication Security*, pages 62–73. ACM Press, 1993.

[5] D. Boneh, X. Boyen, and H.Shacham. Short group signatures. In *Advances in Cryptology — Crypto'04, LNCS 3152*, pages 41–55, 2004.

[6] D. Boneh and H. Shacham. Group signatures with verifier-local revocation. In *Proc. of the 11th ACM Conference on Computer and Communications Security (CCS 2004)*, pages 168–177, 2004.

[7] E. Brickell, J. Camenisch, and L. Chen. Direct anonymous attestation. In *ACM Conference on Computer and Communications Security*, pages 132–145, 2004.

[8] J. Camenisch and J. Groth. Group signatures: Better efficiency and new theoretical aspects. In *Security in Communication Networks (SCN 2004), LNCS 3352*, pages 120–133, 2005.

[9] J. Camenisch and A. Lysyanskaya. Efficient nontransferable anonymous multi-show credential system with optional anonymity revocation. In *B. Pfitzmann, editor, Advances in Cryptology —EUROCRYPT, LNCS 2045*, pages 93–118. Springer-Verlag, 2001.

[10] J. Camenisch and A. Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In *Advances in Cryptology — Crypto'02, LNCS 2442*, pages 61–76, 2002.

[11] J. Camenisch and M. Michels. A group signature scheme based on an RSA-variants. Technical Report RS-98-27, BRICS, University of Aarhus, Nov. 1998.

[12] J. Camenisch and M. Stadler. Efficient group signature schemems for large groups. In *Advances in Cryptology — Crypto'97, LNCS 1294*, pages 410–424, 1997.

[13] J. Camenisch and M. Stadler. A group signature scheme with improved efficiency. In *Advances in Cryptology — ASIACRYPT'98, LNCS 1514*, pages 160–174, 1998.

[14] A. Chan, Y. Frankel, and Y. Tsiounis. Easy come - easy go divisible cash. In *K. Yyberg, editor, Advances in Cryptology – Eurocrypt'98, LNCS 1403*, pages 561 – 574. Sringer-Verlag, 1998.

[15] D. Chaum and E. van Heyst. Group signature. In *Advances in Cryptology — Eurocrypt*, pages 390–407, 1992.

[16] Committee on National Security Systems. National information assurance (IA) glossary – CNSS instruction no. 4009, May 2003.

[17] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Advances in Cryptology — Crypto*, pages 10–18, 1984.

[18] A. Fiat and A. Shamir. How to prove yourself: practical solutions to identification and signature problems. In *Advances in Cryptology — CRYPTO'86, LNCS 263*, pages 186–194. Springer-Verlag, 1987.

[19] E. Fujisaki and T. Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In *Advances in Cryptology — Crypto*, pages 16–30, 1997.

[20] E. Fujisaki and T. Okamoto. A practical and provably secure scheme for publicly verifable secret sharing and its applications. In *Advances in Cryptology – EUROCRYPTO'98*, pages 32–46, 1998.

[21] A. Kiayias, Y. Tsiounis, and M. Yung. Traceable signatures. In *Advances in Cryptology—Eurocypt, LNCS 3027*, pages 571–589. Springer-Verlag, 2004.

[22] TCG. http://www.trustedcomputinggroup.org.

[23] TCG. TPM V1.2 Specification Changes: A summary of changes with respect to the v1.1b TPM specification, 2003.

[24] TCG. TPM Main: Part 1 design principles, 2005.