An Anonymous Authentication Scheme for Identification Card

He Ge

Microsoft Corporation
One Microsoft Way, Redmond, WA 98052
hege@microsoft.com

Abstract. This paper presents the concept of anonymous identification card, a technique enabling a card holder to demonstrates his/her authenticity without disclosing real identity. Anonymous identification card can be used in settings in which people need to demonstrate their eligibility to do certain things, meanwhile they are sensitive to their privacy, not hoping to disclose their identity information to a verifier. We proposed an efficient anonymous authentication scheme for this anonymous identification card, with the support of rogue card revocation. The most advantage of our scheme is its simplicity and efficiency such that all computation can be carried out by a resource-limited identification card. We proved our scheme is secure under the strong RSA assumption and the decisional Diffie-Hellman assumption.

Keywords: Privacy, Anonymous Identification, Identity Management, Group Signature, Cryptographic Protocol.

1 Introduction

Consider the following scenarios: people carry identification cards to prove their authenticity when accessing restricted buildings, using pay parking lots, or driving through tollgates. This identification card is embedded with a cryptographic chip that can carry out computation for authentication. If authentication is associated with a unique identifier (e.g., person's name), transactions by the same user at different places can be tracked and analyzed. To protect a user's privacy, it is desirable to deploy anonymous authentication scheme in such scenarios. That is, a system can verify a user holding a valid card without being able to obtain this person's identity information. A similar scenario happens in Trusted Computing Platforms [18], in which a computer can attest that it holds an original cryptographic chip, Trusted Computing Module (TPM), to a remote server without revealing information for this special chip.

The underlying mechanisms of these applications are cryptographic protocols related to so-called group signature schemes. A group signature is a privacy-preserving signature scheme introduced by Chaum and Heyst [12]. In such a scheme, there are two basic entities: the group manager and certain number of group members. The group manager issues group membership certificate/ credential for each group member. Later, based on its own group membership

P. Ning, S. Qing, and N. Li (Eds.): ICICS 2006, LNCS 4307, pp. 238–248, 2006.

certificate, a group member can sign a message on behalf of the group without revealing its identity. That is, a third party can only identify the signature is produced by one group member without being able to find which particular one. Only the group manager can open a signature and reveal its originator. Besides, signatures signed by the same group member 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., [10,9,1,7,4,5]).

In our target application, an identification card has extremely limited resource, either computing capability, or memory space. It is desirable that a cryptographic protocol should be lightweight. Trusted Computing Platforms deploy an anonymous authentication technique called "Direct Anonymous Attestation" (DAA), which has been introduced in [6]. The current solution for DAA is a computing intensive construction. To complete all cryptographic calculations in real time, the computation has to be distributed among the TPM and the its host, typically a personal computer. DAA works fine for powerful computing devices, however, it is a too expensive construction for an identification card.

The contribution of this paper includes two parts: (1) we propose a statistical zero-knowledge proof of knowledge protocol, through which a prover can convince a verifier that he knows two integers that are relatively prime without revealing any knowledge for these two integers; (2) based on the result in (1), we devised a lightweight anonymous authentication scheme for anonymous identification card. The new scheme is simple and efficient. As a result, all cryptographic computation can be completed by the card alone. Therefore, the new scheme is more suitable for our target scenarios, i.e., identification card.

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

2 The Model

In this section, we define a model which captures security requirements for our target application.

Definition 1 (The Model). *A trusted card issuer takes responsibility for issuing anonymous identification card (AID). The issuer and AIDs form a group in which the issuer holds a group master key, while AIDs hold group member keypairs. The system should satisfy the following security requirements.*

1. *(Forgery-resistance) The keypair in the AID can only be produced using the issuer's master key.*
2. *(Anonymous Authentication) The AID can anonymously attest its authenticity to a verifier. It is infeasible to extract AID's identity information, or link transactions by the same AID.*

3. *(Rogue AID Revocation) Under certain security policy, the issuer can recover an AID's identity and reveal all malicious behaviors associated with this AID. The revoked keypairs should be published on the revocation list to exclude rogue AIDs by all verifiers.*

Our model could be seen as a simplified group signature model (e.g. [1]). However, we adjust the revocation mechanism in the original model to satisfy our security requirement. In the classical group signature model, the revocation is implemented by opening all anonymous signatures in the pool by the group manager. Just as pointed out by Kiayias *et al.* in [16], this mechanism is either inefficient (centralized operation by the group manager), or unfair (unnecessarily identifying all innocent group member's signatures). To overcome this shortcoming, they introduced a variant scheme of group signature called traceable signature, which we refer to as the KTY scheme. However, the new revocation mechanism in the KTY scheme violates a security requirement called "backward unlinkability" in group signature: disclosing the secret of a group member should not reveal all this group member's previous behaviors. This conflicting issue shows in anonymous authentication, suitability of certain feature is more application oriented, and no sole definition could accommodate all conditions. In this paper we adopt the revocation mechanism in the KTY scheme since it is more appropriate for our target application.

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 [8]). *An RSA modulus $n = pq$ is called special if $p = 2p' + 1$ and $q = 2q' + 1$ where p' and q' also are prime numbers. Special RSA modulus is also called safe RSA modulus in some literature [1].*

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 $|QR_n| = \frac{1}{4}|Z_n^*|$.*

We list two properties about QR_n which will be be used in section 5 for the security proof.

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 .*

Property 2. *If g is a generator of QR_n, then $g^a \bmod n$ is a generator of QR_n if and only if $GCD(a, |QR_n|) = 1$.*

The security of our techniques relies on the following security assumptions which are widely accepted in the cryptography literature. (for example, [2,14,9,1]).

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 $v^e \equiv u \pmod{n}$. 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 over 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), $x, y, z \in_R Z_n$, 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 [11,15,9]. They include protocols for things such as knowledge of a discrete logarithm, knowledge of equality of two discrete logarithms, knowledge of a discrete logarithm that lies in an interval, etc. We introduce one of them here. Readers may refer to the original papers for more details.

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. α, l, l_c are security parameters that are all greater than 1. X is a constant number. A prover Alice knows x, the discrete logarithm of T_1, and $x \in [X - 2^l, X + 2^l]$. Alice demonstrates her knowledge of $x \in [X - 2^{\alpha(l+l_c)}, X + 2^{\alpha(l+l_c)}]$ as follows.*

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 $w = t - c(x - X)$, and $w \in \pm\{0,1\}^{\alpha(l+l_c)+1}$. Alice sends w to Bob.*
4. *Bob checks $w \in \pm\{0,1\}^{\alpha(l+l_c)+1}$ and*

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

If the equation holds, Alice proves knowledge of the discrete logarithm of T_1 lies in the range $[X - 2^{\alpha(l+l_c)+1}, X + 2^{\alpha(l+l_c)+1}]$.

Remark 1. It should be emphasized that while Alice knows a secret x in $[X - 2^l, X + 2^l]$, the protocol only guarantees that x lies in the extended range $[X - 2^{\alpha(l+l_c)+1}, X + 2^{\alpha(l+l_c)+1}]$.

Next, we propose a zero-knowledge protocol to show co-primality of two discrete logarithms. That is, a prover demonstrates its knowledge of discrete logarithms of two elements T_1, T_2 in QR_n being relatively prime. The method is based on the following theorem.

Theorem 1. *Let n be an RSA modulus. For a random element $u \in Z_n^*$, if one can find a tuple (T_1, T_2, x, y) such that $T_1^x T_2^y \equiv u \pmod{n}$, then x, y must be relatively prime.*

Proof. By contradiction. If x, y are not co-prime, we assume $GCD(x, y) = e$, $x = k_1 e$, $y = k_2 e$. Then we have $T_1^x T_2^y \equiv (T_1^{k_1} T_2^{k_2})^e \equiv u \pmod{n}$. Thus, we find a pair (v, e) such that $v^e \equiv u \pmod{n}$, where $v \equiv T_1^{k_1} T_2^{k_2} \pmod{n}$, to solve a flexible RSA problem. This contradicts the strong RSA assumption. Therefore x, y must be relatively prime. □

Protocol 2. (Knowledge of Co-Primality of Two Discrete Logarithms) *(Sketch) Suppose Alice knows a, c are relatively prime. She first uses GCD algorithm to compute b, d, such that $ab + cd = 1$. Then Alice computes*

$$T_1 = g^b \pmod{n},\ T_2 = T_1^a \pmod{n},$$

$$T_3 = g^d \pmod{n},\ T_4 = T_3^c \pmod{n}.$$

Alice sends (T_1, T_2, T_3, T_4) to Bob, and proves she knows discrete logarithms of T_2, T_4 with base T_1, T_3 respectively. Finally, $T_2 T_4 = g \pmod{n}$, this shows that discrete logarithms of T_2, T_4 are relatively prime.

4 The Authentication Protocol for Anonymous Identification Card

The card issuer, the producer of AIDs, sets various parameters, the lengths of which depend on a *security parameter*, which we denote by σ.

4.1 System Parameter Setting

The system parameters are set by the issuer, these values are:

- n, g, h: n is a special RSA modulus such that $n = pq$, $p = 2p' + 1$, and $q = 2q' + 1$, where p and q are each at least σ bits long (so $p, q > 2^\sigma$), and p' and q' are prime. $g, h \in_R QR_n$ are random generators of the cyclic group QR_n. n, g, h are public values while p, q are kept secret by the issuer.
- α, l_c, l_s: security parameters that are greater than 1.
- X: a constant value. $X > 2^{\alpha(l_c + l_s)+2}$.

4.2 Key Generation for the AID

The key generation method is straightforward. The card issuer picks a random prime number $s \in_R [X - 2^{l_s}, X + 2^{l_s}]$ and computes

$$E = g^{s^{-1}} \pmod{n},$$

where s^{-1} is the inverse of s modulo $|QR_n| = p'q'$. (E, s) is the keypair of the AID. The issuer feeds (E, s) into the AID, and records (E, s) in its database.

4.3 Anonymous Authentication

The idea of our method to implement anonymous authentication is: the AID generates a random blinding integer b, computes $T_1 = E^b \pmod{n}$, $T_2 = g^b \pmod{n}$. Then the AID sends (T_1, T_2) to a verifier. The AID proves that $T_1^s \equiv T_2 \pmod{n}$ and s lies in the correct interval; $T_2 \equiv g^b \pmod{n}$, and s, b are co-prime.

Protocol 3. (Anonymous Authentication)

1. *The AID picks random* $b \in_R [X - 2^{l_s}, X + 2^{l_s}]$, $t_1, t_2 \in_R \pm\{0,1\}^{\alpha(l_s+l_c)}$. *It uses GCD algorithm to solve* $sa + bd = 1$. *The AID computes (all computations done modulo n):*

$$T_1 = E^b,\ T_2 = g^b,\ T_3 = T_1^{t_1},$$
$$T_4 = h^a,\ T_5 = T_4^s,\ T_6 = T_4^{t_1},$$
$$T_7 = h^d,\ T_8 = T_7^b,\ T_9 = T_7^{t_2},\ T_{10} = g^{t_2}.$$

   $(T_1, T_2, T_3; T_4, T_5, T_6)$ *are used to prove equality of discrete logarithms of* T_2, T_5 *with base* T_1, T_4 *respectively. Also, they are served to prove* s *lies in the correct range.* $(g, T_2, T_{10}; T_7, T_8, T_9)$ *are used to prove equality of discrete logarithms of* T_2, T_8 *with the base* g, T_7, *respectively. The AID sends* $(T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8, T_9, T_{10})$ *to the verifier.*
2. *The verifier picks number* $c \in \{0,1\}^{l_c}$, *and sends it to the AID.*
3. *The AID computes*

$$w_1 = t_1 - c(s - X),\ w_2 = t_2 - c(b - X),$$

   and sends (w_1, w_2) *to the verifier.*
4. *The verifier checks* $w_1, w_2 \in \pm\{0,1\}^{\alpha(l_s+l_c)+1}$, *and checks (all computations done modulo n):*

$$T_1^{w_1 - cX} T_2^c =? T_3,\ T_4^{w_1 - cX} T_5^c =? T_6,$$
$$g^{w_2 - cX} T_2^c =? T_{10},\ T_7^{w_2 - cX} T_8^c =? T_9,\ T_5 T_8 =? h.$$

If all these equations hold, this completes the AID's anonymous authentication.

Remark 2. Using the Fiat-Shamir heuristic[13], the authentication scheme can be turned into a non-interactive "signature of knowledge" scheme, which is secure in the random oracle model [3].

4.4 Rogue AID Revocation

If certain behaviors have been identified suspicious, the transaction transcripts should be submitted to the card issuer to reveal the identity. The issuer checks all s in its database

$$T_1^{s_i} =? T_2 \pmod{n}$$

to reveal this AID's identity. Then s_i is published on the revocation list for all the verifiers.

Later, a verifier can check

$$T_1^{s_i} =? T_2 \pmod{n}$$

for all s_i on the revocation list to identify a rouge AID.

4.5 Performance Analysis

Since the computation cost in the protocol is dominated by the modular squaring and multiplication, we can estimate computation cost by counting total modular squarings and multiplications. Let k_1 be the bit length of the binary representation of exponent, and k_2 be the number of 1's in the binary representation, the total computation cost can be treated as k_1 squarings and k_2 multiplications. For example, if $y = g^x \pmod{n}$, and $x \in_R \{0,1\}^{160}$. We assume half of 160 bits of s will be 1. Then the total computation includes 160 squarings and 80 expected multiplications.

In practice, we can choose $\sigma = 512$, then n is 1024 bits long. Suppose we let $\alpha = 9/8$, $l_c, l_s = 160$, and $X = 2^{367}$(46 bytes). Since $s, b \in [X - 2^{l_s}, X + 2^{l_s}]$. We should notice that the actual random part of s, b is the lower 160 bits, and the leading 268 bits are all $0's$ except the first bit. This would save lots of computation as well as memory space. Then computation related to exponents b, s are 368 squarings and 81 expected multiplications. We treat other exponents as 368-bit long. The AID needs at most 3680 (368×10) squarings and 1428 ($81 \times 4 + 184 \times 6$) multiplications. Therefore, total computation cost is 5108 modular multiplication, which can be completed by the AID alone according to the experiment result for the TPM in Trusted Computing Platform [6].

5 Security Properties

We prepare two lemmas that will be used shortly. The first lemma is due to Shamir [17].

Lemma 1. *Let n be an integer. For given values $u, v \in Z_n^*$ and $x, y \in Z_n$ such that $GCD(x, y) = 1$ and $v^x \equiv u^y \pmod{n}$, there is an efficient way to compute the value z such that $z^x \equiv u \pmod{n}$.*

Proof. Since $GCD(x, y) = 1$, we can use the Extended GCD algorithm to find a and b such that $ay + bx = 1$, and let $z = v^a u^b$. Thus

$$z^x \equiv v^{ax} u^{bx} \equiv u^{ay+bx} \equiv u \pmod{n}. \qquad \square$$

We introduce the second lemma for the security of the AID's keypair.

Lemma 2. *If* $X > 2^{\alpha(l_s+l_c)+2}$, $\alpha, l_s, l_c > 1$, *then* $(X - 2^{\alpha(l_s+l_c)+1})^2 > X + 2^{\alpha(l_s+l_c)+1}$.

Proof.

$$\begin{aligned}&(X - 2^{\alpha(l_s+l_c)+1})^2 - (X + 2^{\alpha(l_s+l_c)+1})\\ &= X^2 - X2^{\alpha(l_s+l_c)+2} + 2^{2\alpha(l_s+l_c)+2} - X - 2^{\alpha(l_s+l_c)+1}\\ &= X(X - 2^{\alpha(l_s+l_c)+2} - 1) + 2^{2\alpha(l_s+l_c)+2} - 2^{\alpha(l_s+l_c)+1}\end{aligned}$$

Since $\alpha, l_s, l_c > 1$, and $X > 2^{\alpha(l_s+l_c)+2}$, the equation is greater than 0. □

Now, we discuss the security of our scheme. First, we address the issue of keypair forgery. We consider an attack model in which an attacker can obtain a set of legitimate keypairs. A successful attack is one in which a new keypair is generated that is valid and different from current keypairs. The following theorem shows that, assuming the strong RSA Assumption, it is intractable for an attacker to forge such a keypair.

Theorem 2 (Forgery-resistance). *If there exists a probabilistic polynomial time algorithm which takes a list of valid keypairs,* $(s_1, E_1), (s_2, E_2), \ldots, (s_k, E_k)$ *and with non-negligible probability produces a new keypair* (s, E) *such that* $s \in [X - 2^{\alpha(l_s+l_c)+1}, X + 2^{\alpha(l_s+l_c)+1}]$, $E^s = g \pmod n$ *and* $s \neq s_i$ *for* $1 \leq i \leq k$, *then we can solve the flexible RSA problem with non-negligible probability.*

Proof. Suppose there exists a probabilistic polynomial-time algorithm $\mathcal{A}$ which computes a new legitimate keypair based on the available keypairs, and succeeds with some non-negligible probability. We can construct an algorithm for solving the flexible RSA problem, given a random input (u, n), as follows:

1. We pick random prime numbers $s_1, s_2, \ldots, s_k$ in the required range $[X - 2^{l_s}, X + 2^{l_s}]$, and compute
$$r = s_1 s_2 ... s_k,$$
$$g = u^r = u^{s_1 s_2 \cdots s_k} \pmod n.$$
For a random input (u, n), the probability of $u \in QR_n$ is $\frac{1}{4}$. Due to Property 1, u will be a generator of QR_n with probability nearly $\frac{1}{4}$. Since the s_i values are primes strictly less than either p' or q', it must be the case that $\mathrm{GCD}(r, |QR_n|) = 1$. Property 2 says that g is a generator of QR_n if and only u is a generator of QR_n. Then g is a generator of QR_n with non-negligible probability.
2. Next, we create k group keypairs, using the s_i values and E_i values calculated as follows:
$$\begin{aligned}E_1 &= u^{s_2 \cdots s_k} \pmod n\\ E_2 &= u^{s_1 s_3 \cdots s_k} \pmod n\\ &\vdots\\ E_k &= u^{s_1 s_2 \cdots s_{k-1}} \pmod n\end{aligned}$$
Note that for all $i = 1, \ldots, k$, $E_i^{s_i} = u^{s_1 s_2 \cdots s_k} = u^r = g \pmod n$.

3. We use the assumed forgery algorithm $\mathcal{A}$ for creating a new valid keypair (E, s), where $s \in [X - 2^{\alpha(l_s+l_c)+1}, X + 2^{\alpha(l_s+l_c)+1}]$, and $E^s = g = u^r \pmod{n}$.
4. If the forgery algorithm succeeds, then s will be different from all the s_i's. By Lemma 2, s cannot be the product of s_i, s_j, $1 \leq i, j \leq k$. Therefore, either $\mathrm{GCD}(s, s_1 s_2 \cdots s_k) = 1$, or $GCD(s, s_1 s_2 \cdots s_k) = s_i$, $1 \leq i \leq k$. In the first case, due to Lemma 1, we can find a pair (y, s) such that

$$y^s = u \pmod{n}$$

so the pair (y, s) is a solution to our flexible RSA problem instance. In the second case, assume $s = v \times s_i$, then $v < X - 2^{\alpha(l_s+l_c)+1}$, and $GCD(v, s_1 s_2 \cdots s_k) = 1$ (or $GCD(v, r) = 1$). We then have

$$E^s \equiv E^{vs_i} \equiv (E^{s_i})^v \equiv u^r \pmod{n}.$$

Again by Lemma 1, we can find a pair (y, v) such that

$$y^v = u \pmod{n}.$$

so the pair (y, v) is a solution to our flexible RSA problem instance.

Through the above steps, assuming the existence of algorithm $\mathcal{A}$, we have solved a random instance flexible RSA problem (u, n) with non-negligible probability. However, this is infeasible under the strong RSA assumption. Therefore, such algorithm $\mathcal{A}$ should not exist under the same assumption. □

Next, we address the security of anonymous authentication scheme which is described as the following theorem.

Theorem 3. *Under the strong RSA assumption, the anonymous authentication protocol is a statistical zero-knowledge honest-verifier proof of a keypair* (E, s) *such that* $E^s \equiv g \pmod{n}$ *and* s *lies in the correct interval.*

Proof (Sketch). Our protocol directly deploys the standard building blocks to accomplish anonymous authentication.

In the protocol, $(g, T_2, T_{10}; T_7, T_8, T_9)$ are used to prove equality of the discrete logarithms of T_2 with base g, and T_8 with base T_7. This is the statistical honest-verifier zero-knowledge protocol that its security has been proved in the literature under the strong RSA assumption.

$(T_1, T_2, T_3; T_4, T_5, T_6)$ are used to prove equality of discrete logarithms of T_2 with base T_1, and T_5 with base T_4. Also, they are served to prove this discrete logarithm $s \in [X - 2^{\alpha(l_s+l_c)+1}, X + 2^{\alpha(l_s+l_c)+1}]$. This is a generalized version of knowledge protocol of equality of discrete logarithms introduced in [16]. The protocol is also secure under the strong RSA assumption.

Since $T_5 T_8 \equiv h \pmod{n}$, by Theorem 1, discrete logarithms of T_5, T_8, s and b, respectively, are co-prime.

Putting the above together, the AID demonstrates that it knows s, b such that $T_1^s \equiv g^b \pmod{n}$, and s, b are relatively prime. Due to Lemma 1, the AID can solve this equation and obtain $E^s \equiv g \pmod{n}$, which is a valid keypair. □

Finally, we propose the theorem for anonymity property of the scheme. The problem of linking two tuples (T_1, T_2), (T_1', T_2') is equivalent to deciding equality of discrete logarithms of T_2, T_2' with bases T_1, T_1' respectively. This is infeasible under the decisional Diffie-Hellman assumption over QR_n. Therefore, we have the following result.

Theorem 4 (Anonymity). *Under the decisional Diffie-Hellman assumption, the protocol implements anonymous authentication such that it is infeasible link transactions by the same AID.*

6 Conclusion

In this paper, we have presented an efficient protocol to implement authentication for anonymous identification card (AID) with supporting rogue AID revocation. The proposed scheme is simple and efficient enough to be deployed in an identification card to protect user's privacy. The new scheme is proved to be secure under the strong RSA assumption and the decisional Diffie-Hellman assumption.

Theorem 1 is an interesting result in the paper, which is a corollary of the strong RSA assumption. Based on this result, we devised a knowledge proof of co-primality of discrete logarithms. This theorem might be used in other cryptographic construction.

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. N. Baric and B. Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In *Advances in Cryptology — Eurocrypto*, pages 480–494, 1997.
3. 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.
4. D. Boneh, X. Boyen, and H.Shacham. Short group signatures. In *Advances in Cryptology — Crypto'04, LNCS 3152*, pages 41–55, 2004.
5. 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.
6. E. Brickell, J. Camenisch, and L. Chen. Direct anonymous attestation. In *ACM Conference on Computer and Communications Security*, pages 132–145, 2004.
7. 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.
8. J. Camenisch and A. Lysyanskaya. A signature scheme with efficient protocols. In *SCN'02, LNCS 2576*, pages 268–289, 2002.

9. 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.
10. J. Camenisch and M. Stadler. Efficient group signature schemems for large groups. In *Advances in Cryptology — Crypto'97, LNCS 1294*, pages 410–424, 1997.
11. 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.
12. D. Chaum and E. van Heyst. Group signature. In *Advances in Cryptology — Eurocrypt*, pages 390–407, 1992.
13. 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.
14. E. Fujisaki and T. Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In *Advances in Cryptology — Crypto*, pages 16–30, 1997.
15. 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.
16. A. Kiayias, Y. Tsiounis, and M. Yung. Traceable signatures. In *Advances in Cryptology—Eurocypt, LNCS 3027*, pages 571–589. Springer-Verlag, 2004.
17. A. Shamir. On the generation of cryptograpically strong psedorandom sequences. *ACM Transaction on computer systems*, 1, 1983.
18. TCG. http://www.trustedcomputinggroup.org.