CERTIFIED PSEUDONYMS COLLIGATED WITH MASTER SECRET KEY

Vijayakrishnan Pasupathinathan, Josef Pieprzyk
ACAC, Department of Computing, Macquarie University, Sydney, Australia
krishnan@ics.mq.edu.au, josef@ics.mq.edu.au

Huaxiong Wang
Division of Mathematical Sciences, Nanyang Technological University, Singapore
HXWang@ntu.edu.sg

Keywords: Anonymity, Identification, Colligated Pseudonyms, TPM

Abstract: A pseudonym provides anonymity by protecting the identity of a *legitimate user*. A user with a pseudonym can interact with an unknown entity and be confident that his/her identity is secret even if the other entity is dishonest. In this work, we present a system that allows users to create pseudonyms from a trusted master public-secret key pair.
The proposed system is based on the intractability of factoring and finding square roots of a quadratic residue modulo a composite number, where the composite number is a product of two large primes. Our proposal is different from previously published pseudonym systems, as in addition to standard notion of protecting privacy of an user, our system offers colligation between seemingly independent pseudonyms. This new property when combined with a trusted platform that stores a master secret key is extremely beneficial to an user as it offers a convenient way to generate a large number of pseudonyms using relatively small storage.

1 INTRODUCTION

The use of pseudonyms have been proposed as a mechanism to hide a user's identity by providing anonymity, while being still suitable to authenticate the holder of the pseudonym in a communication system (Chaum, 1985). David Chaum argued that using pseudonyms provides a way that allows a user to work anonymously, with multiple organisations, by allowing the user to obtain a credential from one organisation using his/her pseudonym and obtain services using that credential from another organisation without revealing his/her true identity (Chaum, 1981; Chaum, 1985). To this end, Chaum and Evertse developed a pseudonym system and proposed an RSA-based implementation while relying on a trusted centre who must sign all credentials (Chaum and Evertse, 1986). Chen extended the scheme from (Chaum, 1985) and presented its discrete-logarithm version that relies on a trusted centre (Chen, 1995). An advantage of these schemes is that, they allow the user to generate pseudonyms, giving n user greater degree of control over his/her identity. However, these schemes have a common weakness. Although the identity of the user is hidden, the credentials (such as certificates of his/her public key) or pseudonyms can be easily shared (unauthorised transfer) with other users.

Based on security of preserving a high-value (*master*) secret key, Canettie et al. (Canetti et al., 2000) and Lysayanskaya et al. (Lysyanskaya et al., 1999) independently proposed non-transferable pseudonym systems. Though credentials obtained on pseudonyms can be used anonymously, the authors of (Canetti et al., 2000) assume that, certification authority (CA) grants credentials only when each user reveals their true identity to them. This makes their scheme prone to collusion between a CA and a verifier, as they can deduce the real identity associated with the user pseudonym. The scheme from (Lysyanskaya et al., 1999) protects against unauthorised transfer of the user credentials, by forcing a user to reveal the master secret key if they choose to share their credentials. But the scheme shares the same weakness as in (Canetti et al., 2000), during the registration phase, users are required to disclose their true identity (master public key) to a CA.

1.1 Scope and Contribution

This paper presents a pseudonym system which is based on the public key cryptosystem. The main idea is to use a single trusted master secret key with many matching public keys (pseudonyms). The proposed system gives users the ability to generate multiple pseudonyms (that are independent of the master public key) from a trusted master secret key. An important property of the system is that, it provides users the ability to generate signatures using the master secret key, which are verifiable using certificates that were issued against pseudonyms.

Let us consider an example. Consider a TPM (Trusted Platform Module) chip that is integrated into a computing platform (such as mobile phones, laptops, etc.). The chip contains a certified public-secret key pair. The public key is certified by its manufacturer and recorded on the TPM chip at the time of manufacturing. The certified public key of the chip can be used to authenticate the machine with the TPM. The TPM is used to further certify public keys of users associated with the machine. A verifier can authenticate a user based on the certificate chain consisting of the user certificate, the TPM certificate and the manufacturer certificate. But, revealing the identity of the machine to every verifier would not only compromise the anonymity of the machine but also the anonymity of user(s) of the machine. It is possible to identify a user using their pseudonyms but, the verifier trusts only the TPM chip's certified public key and not the operating system of the machine or any newly generated pseudonyms. Therefore, we require a system that gives a user the ability to generate and control the usage of multiple identities based on a trusted master identity (TPM's certified public key), where the pseudonyms should not only be independent of the master identity (anonymity), but also there is a relation between all pseudonyms generated[1] and the trusted master secret key stored in the chip (we call this relation *colligation*).

Anonymity and colligation are in some sense contradictory. Anonymity requires that, it is impossible (at least computationally) for an entity with knowledge of a pseudonym, to link that pseudonym with either the master identity or any other generated pseudonym. Whereas, colligation requires that the prover is guaranteed that there is an underlying link that exists between all pseudonyms (that appear to be unrelated to each other) was generated from the trusted master secret key. Previously published proposals like, (Damgard, 1988; Lysyanskaya et al., 1999; Camenisch and Lysyanskaya, 2002; Chen, 1995; Canetti et al., 2000; Chaum, 1985) that achieved anonymity have considered a user's identity that consists of public-secret key pair as a single unified structure. Under a such assumption it is unfeasible to obtain both anonymity and colligation. We aim to segregate the structure and provide anonymity to a user but still maintain colligation between pseudonyms generated using the user's master secret key. The implication of this structure is that, a user's master secret key becomes highly valuable, as all his pseudonyms are linked directly to the secret key.

Based on the security requirement of non revealable master public key in a TPM, Brickell et al. proposed a method for direct anonymous attestation (DAA) (Brickell et al., 2004) that provides anonymity to a user based on the Camenisch-Lysyanskaya credential system (Camenisch and Lysyanskaya, 2002). Unfortunately, the scheme (Brickell et al., 2004) does not provide secret key linkability for identities that are generated. Consequently, in their scheme, the TPM needs to maintain a database of those identities and associated secret keys. The database can get quite large if the TPM serves a large group of users. Also, their DAA scheme does not support identity transfer among machines. In this paper we limit ourselves to the problem of achieving anonymity and colligation, and we do not address the issue of identity transfer.

1.2 Organisation

Section 2 provides the background on anonymous certification system and cryptographic techniques employed. In Section 3 we provide our construction, and in Section 4, we discuss its security. In Section 5, we discuss integration of our proposal in a TPM based setting and conclude in Section 6.

2 Background

User anonymity and colligation between master secret key and user generated identities is of paramount importance. To provide anonymity to user generated identities (pseudonyms) our proposal will make use of an anonymous certification scheme, such as, a scheme with blind signatures. An anonymous certification system is necessary to provide anonymity to a user and to prevent collusion between a certi-

[1] To a certifier it is essential that the system provides guarantee that, all pseudonyms from a particular TPM can be traced back to a single secret key, but a verifier needs proof of this binding between the master secret key and only the pseudonym that he/she is currently presented with. We do not make this distinction here.

fier and a verifier. To this end, we will employ a modified blind signature scheme (refer Section 3.3) proposed by Pointcheval (Pointcheval, 2000). Note that any anonymous certification scheme that supports non-transferability and revocation of anonymity can be employed with some necessary modifications. To provide colligation between the generated pseudonyms and master secret key we can use any one-way function. In our construction we use squaring modulo a composite integer. In this section, first we describe the model of an anonymous certification scheme that is going to provide certificates for user generated identities (pseudonyms). In the remaining of this section we summarise the main cryptographic building blocks that we use in our constructions.

2.1 Anonymous Certification System

Anonymous certification system (ACS) represents the certification process of a public key by a certifier who does not know the public key. This is essentially a Chaum blind signature (Chaum, 1982) on the public key of the user, *i.e.* it provides anonymity to the receiver[2].

A typical ACS consists of four entities and three protocols. The entities are: a user $\mathcal{U}$, a verifier $\mathcal{V}$, a certifier $\mathcal{C}$ and a trustee (tracer) $\mathcal{T}$. The protocol suites include: a *certification protocol*, where $\mathcal{U}$ interacts with $\mathcal{C}$ to obtain a certified pseudonym *i.e.* the pseudonym is blindly signed. An *identification protocol*, where $\mathcal{V}$ interacts with $\mathcal{U}$ to authenticate $\mathcal{U}$'s credential and provide services. A *trace protocol*, where $\mathcal{T}$ participates and is invoked to trace the real identity associated with $\mathcal{U}$'s pseudonym.

2.1.1 System setting

The user, $\mathcal{U}$, chooses a modulus N_i, such that a $N_i = p_1^{(i)} p_2^{(i)}$, is a product of two distinct large primes each congruent to 3 (mod 4), ($p_1^{(i)}, p_2^{(i)}$ are Blum integers (Blum et al., 1986)), an element $g \in \mathbb{Z}_{N_i}$ whose order is $\phi(N_i) = (p_1^{(i)} - 1)(p_2^{(i)} - 1)$ and where i is the number of pseudonyms. We also require the modulus for pseudonyms to be different, otherwise anonymity can be compromised trivially by just maintaining a list of modulus. The user chooses a master secret key $SK_{\mathcal{U}_0} \in \mathbb{Z}_{N_0}$ and publishes the master public key $PK_{\mathcal{U}_0} = g^{SK_{\mathcal{U}_0}} \mod N_0$ (which represents the user's true and public identity). The certifier $\mathcal{C}$ publishes its public key $PK_{\mathcal{C}} = g^{SK_{\mathcal{C}}} \mod N_c$ while keeping the corresponding secret key private. The certifier also publishes the public key of the Trustee $\mathcal{T}$, (for tracing and revocation) which would be of the form $PK_{\mathcal{T}} = g_1^{SK_{\mathcal{T}}} \mod N_T$, where $g_1 \in \mathbb{Z}_{N_T}$. Every user registers with a certification authority to obtain a certificate of the form $\mathtt{CERT}_{\mathcal{C}}\langle PK_{\mathcal{U}_0}\rangle$.

[2]Whereas, group signature schemes as employed by (Brickell et al., 2004) provide anonymity to the source.

2.1.2 Protocol Certify

The certification involves two steps: certification of the master public key and certification of pseudonyms. In an TPM based setting the master public key is certified by the manufacturer, and the following describes the certification of the pseudonyms.

The user, $\mathcal{U}$, generates pseudonyms of the form ($PK_{\mathcal{U}_1}$, ..., $PK_{\mathcal{U}_i}$) using the identity generation process described in Section 3.2. The user then identifies himself/herself (using the master public key) to the certifier and engages in a *certify* protocol to obtain a certificate on a pseudonym $PK_{\mathcal{U}_i}$. The value of $PK_{\mathcal{U}_i}$ is never revealed to the certifier. We shall express this phase as

$$(PK_{\mathcal{U}_i}, \mathtt{CERT}_{\mathcal{C}}\langle PK_{\mathcal{U}_i}\rangle) \leftarrow Certify(\mathcal{U}, \mathcal{C}, \mathtt{CERT}_{\mathcal{C}}\langle PK_{\mathcal{U}_0}\rangle)$$

i.e. " $\mathcal{U}$ engages in the certify protocol with $\mathcal{C}$ using $\mathtt{CERT}_{\mathcal{C}}\langle PK_{\mathcal{U}_0}\rangle$ to obtain a certificate on $PK_{\mathcal{U}_i}$, $\mathtt{CERT}_{\mathcal{C}}\langle PK_{\mathcal{U}_i}\rangle$".

2.1.3 Protocol Identify

A user $\mathcal{U}$ who wishes to avail services offered by a verifier $\mathcal{V}$, engages in a identification protocol to convince that he/she possess the necessary credentials. We shall express this phase as

$$\langle \mathtt{PROOF}_{\mathcal{U}_i}\rangle \leftarrow Identify(\mathcal{U}, \mathcal{V}, PK_{\mathcal{U}_i}, \mathtt{CERT}_{\mathcal{C}}\langle PK_{\mathcal{U}_i}\rangle, PK_{\mathcal{T}})$$

i.e. " $\mathcal{U}$ engages in an identification protocol with a verifier $\mathcal{V}$ using the psuedonymn $PK_{\mathcal{U}_i}$ and $\mathtt{CERT}_{\mathcal{C}}\langle PK_{\mathcal{U}_i}\rangle$ and which contains the encryption of the identity under the public key $PK_{\mathcal{T}}$".

2.1.4 Protocol Trace

A verifier who needs to trace the identity of the user contacts the trustee $\mathcal{T}$ by providing with the transcript from an identification protocol $\langle \mathtt{PROOF}_{\mathcal{U}_i}\rangle$. We shall express this phase as

$$(PK_{\mathcal{U}_0}) \leftarrow Trace(\mathcal{V}, \mathcal{T}, PK_{\mathcal{U}_i}, \mathtt{CERT}_{\mathcal{C}}\langle PK_{\mathcal{U}_i}\rangle, \langle \mathtt{PROOF}_{\mathcal{U}_i}\rangle)$$

i.e. " $\mathcal{V}$ engages in the tracing protocol with $\mathcal{T}$ using the values $PK_{\mathcal{U}_i}, \mathtt{CERT}_{\mathcal{C}}\langle PK_{\mathcal{U}_i}\rangle$ and proof of identity use $\langle \mathtt{PROOF}_{\mathcal{U}_i}\rangle$ to obtain the master identity $PK_{\mathcal{U}_0}$".

2.2 Assumptions

Our system relies on the following assumptions:

- **Assumption 1** (*Factoring*) A probabilistic polynomial-time algorithm $\mathcal{G}$ exists which on input $1^{|N|}$ outputs N, where N is a composite of two prime number, p_1 and q_1, such that for any probabilistic polynomial time algorithm $\mathcal{A}$, the probability that $\mathcal{A}$ can factor N is negligible *i.e.* the probability of success is smaller than $\frac{1}{poly(|N|)}$.
- **Assumption 2** (*Square Root*) A probabilistic polynomial-time algorithm $\mathcal{A}$ which on input N and a, where N is a composite of two prime numbers, p_1 and q_1 and $a \in \mathrm{QR}_N$ is a quadratic residue, the probability that $\mathcal{A}$ can output b, such that $b^2 \equiv a \mod N$ is negligible, *i.e.* the probability of success is smaller than $\frac{1}{poly(|N|)}$.
- **Assumption 3** (*Square Decisional Diffie-Hellmann*) The square decisional Diffie-Hellman (SDDH) problem is defined as follows. Distinguish between distributions of the form (g, g^a, g^{a^2}) from (g, g^a, g^r), where r is random and uniformly chosen integer from $\{1, \ldots, N-1\}$. We assume that there is no probabilistic polynomial-time algorithm $\mathcal{G}$ that can solve a random instance of the SDDH problem with probability $\frac{1}{2} + \frac{1}{poly(|N|)}$.

We also use the Chaum and Pederson construction (Chaum and Pedersen, 1992) as a sub-protocol for a interactive proof of knowledge for the discrete log problem (DL-EQ). Their protocol (Chaum and Pedersen, 1992) was designed for the case when group of the exponents has prime order, whereas in our protocol the group of the exponents have composite order. But as suggested by (Camenisch and Michels, 1999), the proof of knowledge of discrete logarithm from different groups (DL-EQ) holds even when working over a cyclic sub-group of $\mathbb{Z}_N^*$. We combine the DL-EQ with El-Gamal encryption over a composite modulus (Franklin and Haber, 1993) to encrypt the master identity of the user under the public key of the trustee, verifiable by the certification authority.

3 PROTOCOLS

We shall now present our scheme that consists of four phases: identity generation, certification, identification and trace.

3.1 System Setting

The system involves four entities. A user $\mathcal{U}$ who holds a long term certified public key $PK_{\mathcal{U}_0}$ (we shall call it the master public key), and wishes to hide his identity from a verifier $\mathcal{V}$. The public keys are certified by a certification authority $\mathcal{C}$ and a trustee $\mathcal{T}$ responsible for tracing the pseudonym used by the user.

The $\mathcal{U}$ master public-secret key-pair is generated as in Section 2.1.1. $\mathcal{U}$ then obtains a certificate on the master public key $PK_{\mathcal{U}_0}$ from a certification authority $\mathcal{C}$, which represents the $\mathcal{U}$'s true identity.

The public key of the certification authority is $PK_{\mathcal{C}} = g^{SK_{\mathcal{C}}}$ and the trustee is $PK_{\mathcal{T}} = g_1^{SK_{\mathcal{T}}}$, where $SK_{\mathcal{C}}$ and $SK_{\mathcal{T}}$ are the corresponding secret keys for the certification authority and the trustee respectively.

3.2 Identity Generation

$\mathcal{U}$ generates new identities using the following key generation process, which takes the inputs, N_j, g, a counter value i (indicating the total number of new identities being generated), identity level l (number of identities generated previously) and the master secret key $SK_{\mathcal{U}_0}$.

`I-Generation(`g,i,l,$SK_{\mathcal{U}_0}$`)`
`For` $j = l,\ldots,i$ `do` $PK_{\mathcal{U}_j} = g^{SK_{\mathcal{U}_0}^{2^j}} \mod N_j$ `EndFor`
`Return(`$PK_{\mathcal{U}_l},\ldots,PK_{\mathcal{U}_j}$`)`

During the first run the value of identity level l would be 1 and counter value i is the number of new identities $\mathcal{U}$ requires. Further calls to the key generation, the identity level would be the counter value that was used during the previous run ($l' = i$). An implicit requirement is that, $\mathcal{U}$ should keep track of the values i and l as long as the master public key remains valid.

We could (and do) treat the identities generated as public keys, that are of the form $(PK_{\mathcal{U}_l},\ldots,PK_{\mathcal{U}_i}) = (g^{SK_{\mathcal{U}_0}^{2^l}},\ldots,g^{SK_{\mathcal{U}_0}^{2^i}})$

3.3 Certification

The newly generated public keys ($PK_{\mathcal{U}_1}$, ..., $PK_{\mathcal{U}_l}$) are required to be certified by $\mathcal{C}$ before they can be used. It is possible to use a normal certification procedure as currently employed in public key cryptosystems, where the public key $PK_{\mathcal{U}_i}$ is signed by $\mathcal{U}$ using the master secret key $SK_{\mathcal{U}_0}$ and sent to $\mathcal{C}$ for certification. $\mathcal{C}$ verifies the signature using the master public key $PK_{\mathcal{U}_0}$, on a successful verification $\mathcal{C}$ digitally signs using his private key $SK_{\mathcal{C}}$ and sends the certificate to $\mathcal{U}$. This method is quite straightforward,

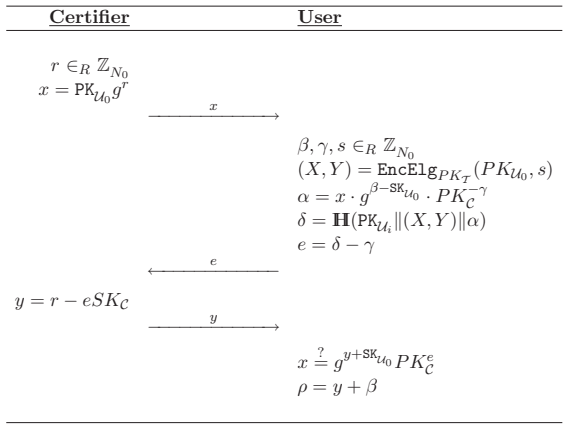

Figure 1: Modified Blind Certification Protocol of (Pointcheval, 2000) - The signature on $PK_{\mathcal{U}_i}$ is (α, δ, ρ) and a receiver can verify using the relation $\alpha \stackrel{?}{=} g^\rho \mathtt{PK}_\mathcal{C}^\delta$

but certain applications (e.g. applications based on TPM) require the new identities to be protected even from the certifier. So, we propose a modification to the certification scheme based on a blind signature scheme using a composite modulus by Pointcheval (Pointcheval, 2000). The blind signature scheme now includes the master public key of the user which is used by the certifier to form the commitment and is later verified by the user.

The certification process is represented by:

$$(PK_{\mathcal{U}_i}, \mathtt{CERT}_\mathcal{C}\langle PK_{\mathcal{U}_i}\rangle) \leftarrow Certify(\mathcal{U}, \mathcal{C}, \mathtt{CERT}_\mathcal{C}\langle PK_{\mathcal{U}_0}, (X,Y)\rangle)$$

where, $\mathtt{CERT}_\mathcal{C}\langle PK_{\mathcal{U}_i}\rangle$ is the valid blind signature $(PK_{\mathcal{U}_i}, \alpha, \delta, \rho)$ by $\mathcal{C}$ on $PK_{\mathcal{U}_i}$ and (X,Y), accomplished by the three-pass protocol depicted in Figure 1. The security proof of the modified protocol trivially follows the proof presented in Pointcheval's paper (Pointcheval, 2000).

3.4 Identification

The Identification protocol (Figure 2) is based on Pointcheval optimised identification scheme (Pointcheval, 2000) of Girault's identification scheme (Girault, 1991), but it now also includes the DL-EQ $\log_g X = \log_{\mathtt{PK}_\mathcal{T}} Y$. In this protocol a user $\mathcal{U}$ uses his certified pseudonym to identify himself/herself with a verifier $\mathcal{V}$ and at the end of the protocol the verifier obtains an undeniable proof of $\mathcal{U}$ participation in the protocol. The identification process is represented by

$$\langle \mathtt{PROOF}_{\mathcal{U}_i}\rangle \leftarrow Identify(\mathcal{U}, \mathcal{V}, PK_{\mathcal{U}_i}, \mathtt{CERT}_\mathcal{C}\langle PK_{\mathcal{U}_i}\rangle, PK_\mathcal{T})$$

3.5 Tracing

The trace protocol (Figure 3) is invoked by a verifier $\mathcal{V}$ after $\mathcal{U}$ has misused a pseudonym and runs

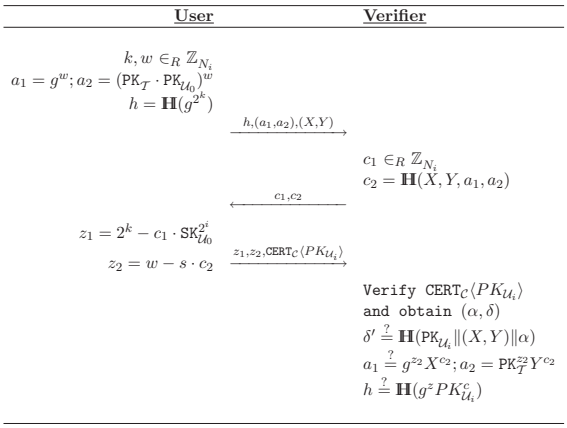

Figure 2: Identification Protocol

between the verifier $\mathcal{V}$ and the trustee $\mathcal{T}$. To trigger the protocol $\mathcal{V}$ has to provide proof of protocol participation by $\mathcal{U}$. We shall express this phase as

$$(PK_{\mathcal{U}_0}) \leftarrow Trace(\mathcal{V}, \mathcal{T}, PK_{\mathcal{U}_i}, \mathtt{CERT}_\mathcal{C}\langle PK_{\mathcal{U}_i}\rangle, \langle \mathtt{PROOF}_{\mathcal{U}_i}\rangle)$$

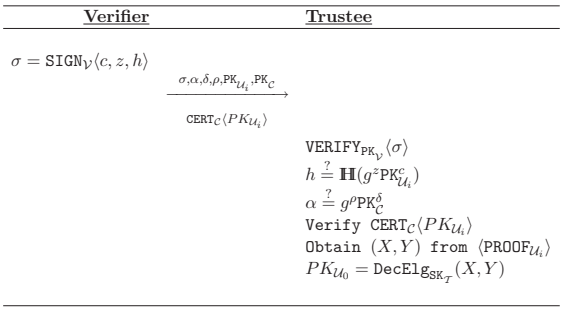

Figure 3: Tracing Protocol

4 SECURITY

4.1 Adversary Goals

We assume an active adversary $\mathcal{A}$, who is capable of eavesdropping and injecting messages in the communication medium. We also assume that an adversary may be also be a legitimate (but dishonest) participant in a protocol, *i.e.* either the certifier or the verifier or both may be dishonest.

As in (Damgard, 1988; Lysyanskaya et al., 1999), we want our pseudonym system to be secure against the following attacks, *i.e.* an adversary's goal is to mount any of following attacks:

- *Pseudonym forgery*: An adversary tries to forge a pseudonym for some user, possibly in associa-

tion with other participants, including the certifier. That is the attack can be either:

1. An adversary in possession of a valid proof tuple $(PK_{\mathcal{U}_i}, \mathtt{CERT}_{\mathcal{C}}\langle PK_{\mathcal{U}_i}\rangle)$ issued to another user or for a tuple of the form $(PK_{\mathcal{U}_i}, \mathtt{CERT}_{\mathcal{C}}\langle PK_{\mathcal{A}}\rangle)$ is successfully able to execute an identification protocol with a verifier identifying as $\mathcal{U}_i$.
2. An adversary successfully identifying himself/herself by executing an identification protocol with a tuple of the form $(PK_{\mathcal{A}}, \mathtt{CERT}_{\mathcal{C}}\langle PK_{\mathcal{U}_i}\rangle)$.

- *Identity compromise*: An adversary in association with other participants tries to obtain information regarding the user's master public-secret key-pair, *i.e.* and adversary with the knowledge of all user generated public keys ($PK_{\mathcal{U}_1}, \ldots, PK_{\mathcal{U}_l}$), it should be computationally infeasible for an adversary to either obtain the master public key $PK_{\mathcal{U}_0}$.
- *Pseudonym linking and colligation*: An adversary tries to obtain information that links a pair of pseudonyms to the same user or to a user's master public key. The goal is that even with the knowledge of all user generated public keys ($PK_{\mathcal{U}_1}, \ldots, PK_{\mathcal{U}_l}$), it should be computationally infeasible for an adversary to prove that any of the PK's in the set ($PK_{\mathcal{U}_1}, \ldots, PK_{\mathcal{U}_l}$), are related.

We now present our claims on the security of our proposal.

Claim 4.1 *If the Square Decisional Diffie-Hellman (SDDH) problem is hard, then public keys generated from the master public key are indistinguishable.*

The public keys generated are of the form $g^{SK^{2^i}}$ where, $i \in 0, \ldots, l$. For an adversary $\mathcal{A}$ to distinguish between a newly generated public key from a master or another newly generated public key, $\mathcal{A}$ should solve the square Diffie-Hellman decision problem, i.e., efficiently distinguish between two distributions of the form (g, g^{SK}, g^{SK^2}) and (g, g^{SK}, g^c), which is assumed to be hard.

Claim 4.2 *It is computationally infeasible to obtain the master public key of a user by an adversary even with the knowledge of all newly generated public key.*

Proof (Sketch) : For an adversary to obtain the master public key ($\mathtt{PK}_{\mathcal{U}_0}$) from a pseudonym ($\mathtt{PK}_{\mathcal{U}_i}$) presented, the adversary needs to solve, first the discrete log problem to obtain $\mathtt{SK}_{\mathcal{U}_i}$ and then solve the square root problem to obtain the value i. This violates our security assumptions. It is also a well known fact that, assuming the factoring of Blum Integers is intractable, the function $f_N = \mathtt{SK}_{\mathcal{U}_0}^{2^i} \mod N_i$ is a trapdoor (one-way) permutation (Goldreich, 1999).

Claim 4.3 *It is computationally infeasible to obtain the master public key of a user by a verifier or a certifier even if the certifier and verifier collude.*

Proof (Sketch) : Both $\mathcal{C}$ and $\mathcal{V}$ have knowledge of the public parameters. In addition, $\mathcal{C}$ has the knowledge of the user's master public key $\mathtt{PK}_{\mathcal{U}_0}$, whereas, a verifier has the knowledge of the cipher-text obtain from the El-Gamal encryption (X,Y), the pseudonym of the user $\mathtt{PK}_{\mathcal{U}_i}$, the signature value on both the pseudonym and the cipher-text $(\alpha, \rho, \delta, \mathtt{PK}_{\mathcal{C}})$.

For a dishonest certifier $\widehat{\mathcal{C}}$ and a dishonest verifier $\mathcal{V}$ to obtain the master public key $\mathtt{PK}_{\mathcal{U}_0}$ of a user, either independently or in collusion, any one of the following cases need to be satisfied.

Case 1 The blind signature protocol during the certification process leaks information about the identity of the user.

Case 2 A verifier is able to deduce the master identity from the pseudonym presented during the identification protocol.

Case 3 The certifier and a verifier with their combined knowledge, are able to identify the colligation that exists between a pseudonym and the master secret key.

The security of Case 1 trivially follows the proof of security of blind signature protocol by Pointcheval in (Pointcheval, 2000). For Case 2, a verifier can obtain the master public key if the proof of DL-EQ $\log_g X = \log_{\mathtt{PK}_{\mathcal{T}}} Y$ leaks any information regarding the master public key. A way of proving the security of the scheme is via the oracle replay technique formalised by Pointcheval and Stem (Pointcheval and Stern, 1996). In particular, the Schnorr signature with composite modulus has been proved secure in the random oracle model (Bellare and Rogaway, 1993) by Poupard and Stem (Poupard and Stern, 1998). They showed that if an adversary is able to forge a signature under an adaptively chosen message attack, then he/she is able to compute discrete logarithms in G.

The security of Case 3 is based on the inability of $\widehat{\mathcal{V}}$ and $\widehat{\mathcal{C}}$ to obtain any information that links the user when he/she interacts in the identification and the certification protocols. There are only two possibilities which can identify a user that he/she participated in both the protocols. (a) The pseudonym leaks value about the true identity and (b) the El-Gamal cipher-text (X,Y) which is used in both the certification and identification protocols can be linked to $\mathtt{PK}_{\mathcal{U}_0}$. If $\widehat{\mathcal{C}}$ and $\widehat{\mathcal{V}}$ in collusion are able to identify that the same (X,Y) which was presented in the identification protocol was used in the certification protocol, then they can positively establish a connection

between the pseudonym presented during the identification protocol with the master identity used in the certification protocol.

From Theorem 4.2, we can conclude that it is computationally infeasible for $\widehat{\mathcal{V}}$ or $\widehat{C}$ to obtain the master identity from a given pseudonym. As for possibility two, the hash value δ computed with cipher-text (X,Y), the pseudonym $\mathtt{PK}_{\mathcal{U}_i}$ and the value α as inputs, is blindly signed and never revealed to the certifier.

Claim 4.4 *If the El-Gamal encryption is secure, then only the corresponding trustee can obtain information about the user from the encrypted cipher-text.*

The proof of this theorem directly follows the proof in (Franklin and Haber, 1993). The authors showed that the security of the composite El-Gamal reduces to computing quadratic residue over a composite modulus that is a product of two primes. And since the master public key is encrypted using the pubic key of the trustee, only the trustee can successfully decrypt the cipher-text.

Remark 1 The protocol also provide guarantees of honest participation of a user. The cipher-text containing the master public key is signed (blindly) by the certifier. A verifier computes the hash value of the cipher-text (X,Y), the pseudonym and α again to verify against the signed hash value in the blind certificate, thus confirming that the user has performed an El-Gamal encryption over the same values, which was used during the certification process.

5 Application

In this section, we present a brief summary about how the protocols can be applied in an Trusted Platform Module (TPM) based setting. We are considering a TPM setting because of tamper resistant protection offered to the master secret key, but the protocols can be applied to other structures like directory based services (e.g. active directory, LDAP).

The Trusted Platform Module (TPM) is the basis of trusted computing, promoted by the Trusted Computing Group. A TPM consists of an unique *endorsement key* (EK) pair, that is built into the hardware module during manufacturing. The public part of EK is certified by the manufacturer and the secret part is sealed inside the TPM and is never revealed to the outside. A primary function of the TPM is attestation *i.e.* the TPM provides guarantees to a remote service that the platform is not tampered with and therefore secure. A TPM can also provide other security services like secure boot and sealed storage. We refer the reader to (TCG, 2001; TCG, 2007) for more information about TPMs.

The deployment of TPM raises some valid privacy concerns. Authentication based on directly using the TPM's EK, will compromise anonymity of the module as all transactions performed by the same TPM can be linked. Further more, it will compromise the anonymity of the user associated with the module. Privacy protection in TPM currently involves two mechanism: Privacy CA based attestation (TPM v1.1) (TCG, 2001) and Direct Anonymous Attestation (TPM v1.2)(Brickell et al., 2004; TCG, 2007). We do not propose a replacement to current TPM authentication standards. We merely wish to highlight the use of TPM as an application for our proposal, and as mentioned before our protocols can be integrated into other systems like directory based services.

TPM BASED SETTING: The endorsement key (EK) in a TPM will be of the form $(\mathtt{PK}_{\mathcal{U}_0},\mathtt{SK}_{\mathcal{U}_0})$. The EK is certified by the manufacturer and embedded into the TPM. A user who wishes to obtain services from an application software on a machine generates a pseudonym of the form $(\mathtt{PK}_{\mathcal{U}_i},\mathtt{SK}_{\mathcal{U}_i})$ as described in Section 3.2. The application software and the TPM then perform an identification protocol as in Section 3.4. At the end of the identification protocol the application software is provided a guarantee on the identity of the user and the associated TPM, but the system still protects the identity of both the TPM and the user associated with it.

6 CONCLUSION

The aim of a pseudonym is to hide the identity of *legitimate users* by providing confidentiality to the identity, thereby providing anonymity. A pseudonyms also need to be traced in case of misuse and therefore needs to provide only restricted anonymity. In this paper, we have presented an pseudonym system by using the property of preserving a high value secret key. The system not only provides restricted anonymity but also supports colligation between a trusted *high value* secret key and generated pseudonyms.

Compared to other pseudonym schemes, our scheme has an efficient identification protocol, thus computation can be carried out on a devices that are constrained of processing power. Computations may be performed on the module itself, whereas the DAA scheme (Brickell et al., 2004; TCG, 2007) requires computation to be distributed among the TPM and the host computer. Our scheme is also ideally suited for storage constraint devices. Because, there are no new secret key to be generated for each pseudonyms, only

counter values of the pseudonym, thus there is no appreciable increase in storage requirement even when the number of pseudonyms required are high.

Finally, in terms of anonymity, our proposal provides a excellent benefit to users, as not only applications on a single computer can be associated with a different pseudonym but also every web based application used by a user can be associated with a pseudonym.

REFERENCES

Bellare, M. and Rogaway, P. (1993). Random oracles are practi- cal: A paradigm for designing efficient protocols. *ACM Conference on Computer and Communications Security'93*, pages 62—73.

Blum, L., Blum, M., and Shub, M. (1986). A simple unpredictable pseudo random number generator. *SIAM J. Computing*, 15(2):364–383.

Brickell, E., Camenisch, J., and Chen, L. (2004). Direct anonymous attestation. In *11th ACM Conference on Computer and Communications Security*. ACM Press.

Camenisch, J. and Lysyanskaya, A. (2002). Dynamic accumulators and application to efficient revocation of anonymous credentials. *Advances in Cryptology - CRYPTO'02*, LNCS 2442:101–120.

Camenisch, J. and Michels, M. (1999). Separability and efficiency for generic group signature schemes. *Advances in Cryptology -CRYPTO'99*, LNCS 1666:413—430.

Canetti, R., Charikar, M. S., Rajagopalan, S., Ravikumar, S., Sahai, A., and Tomkins, A. S. (2000). Non-transferable anonymous credentials. Patent No: 7222362.

Chaum, D. (1981). Untraceable electronic mail, return addresses, and digital pseudonymns. *Communications of the ACM*, 24(2):84–88.

Chaum, D. (1982). Blind signatures for untraceable payments. *Advances in Cryptology -CRYPTO'82*, pages 199–203.

Chaum, D. (1985). Security without identification: Transaction systems to make big brother obsolete. *Communications of the ACM*, 28(10):1030–1044.

Chaum, D. and Evertse, J.-H. (1986). A secure and privacy-protecting protocol for tranmitting personal information between organisation. In *Advances in Cryptology - CRYPTO'86*, pages 118–167. Springer-Verlag.

Chaum, D. and Pedersen, T. (1992). Transferred cash grows in size. *Advances in Cryptology -EUROCRYPT'92*, LNCS 658:390–407.

Chen, L. (1995). Access with pseudonyms. In Dawson, E. and Golic, J., editors, *Cryptography: Policy and Algorithms*, number 1029, pages 232–243. Springer-Verlag.

Damgard, I. (1988). Payment systems and credential mechanisms with provable security against abuse by individuals. *Advances in Cryptology -CRYPTO'88*, LNCS 403:328–335.

Franklin, M. and Haber, S. (1993). Joint encryption and message-efficient secure computation. *Advances in Cryptology -CRYPTO'93*, LNCS 773:266 – 277.

Girault, M. (1991). Self-certified public keys. In *Advances in Cryptology - EUROCRYPT'91*, volume 547 of *Lecture Notes in Computer Science*, pages 490–497. Springer-Verlag.

Goldreich, O. (1999). *Modern Cryptoraphy, Probabilistic Proofs and Pseudo-randomness*. Springer.

Lysyanskaya, A., Rivest, R. L., Sahai, A., and Wolf, S. (1999). Pseudonym systems (extended abstract). *Selected Areas in Cryptography'99*, LNCS 1758:184–199.

Pointcheval, B. and Stern, J. (1996). Security proofs for signature schemes. *Advances in Cryptology - EUROCRYPT'96*, LNCS 1070:387–398.

Pointcheval, D. (2000). The composite discrete logarithm and secure authentication. In Imai, H. and Zheng, Y., editors, *International Workshop on Practice and Thoery in Public Key Cryptography - PKC'2000*, volume 1751 of *Lecture Notes in Computer Science*, pages 113–128, Melbourne, Australia. Springer-Verlag.

Poupard, G. and Stern, J. (1998). Security analysis of a practical "on the fly" authentication and signature generation. *Advances in Cryptology - EUROCRYPT'98*, LNCS 1403:422–436.

TCG (2001). Trusted computing group main specification v1.1.

TCG (2007). Trusted computing group main specification v1.2.