Self-Certified Sybil-Free Pseudonyms

Leonardo A. Martucci
Karlstad University
Universitetsgatan 2
651 88 Karlstad, Sweden

Markulf Kohlweiss
Katholieke Universiteit Leuven
Kasteelpark Arenberg 10
Leuven-Heverlee, Belgium

Christer Andersson
Combitech
Regementsgatan 17
653 40 Karlstad, Sweden

Andriy Panchenko
RWTH Aachen University
Informatik IV Ahornstr. 55
D-52074 Aachen, Germany

ABSTRACT

Accurate and trusted identifiers are a centerpiece for any security architecture. Protecting against Sybil attacks in a privacy-friendly manner is a non-trivial problem in wireless infrastructureless networks, such as mobile ad hoc networks. In this paper, we introduce *self-certified Sybil-free pseudonyms* as a means to provide privacy-friendly Sybil-freeness without requiring continuous online availability of a trusted third party. These pseudonyms are self-certified and computed by the users themselves from their cryptographic long-term identities. Contrary to identity certificates, we preserve location privacy and improve protection against some notorious attacks on anonymous communication systems.

Categories and Subject Descriptors: C.2.2 [Network Protocols]: Applications; Protocol Architecture.

General Terms: Security.

Keywords: Identities, Privacy, Sybil Attack.

1. INTRODUCTION

Today, many users cooperate in networked environments. Often, these systems depend on a majority of users being honest for tasks like voting in virtual community, reputation computation, or traffic mixing. Unless such systems implement expensive countermeasures they fall prey to the Sybil Attack [5], which entails a single attacker controlling arbitrarily many user accounts (Sybil identities). Further, both identity certificates and advanced non-centralized Sybil defence mechanisms [7, 15] are highly privacy-invasive.

We define an *identity domain* as a domain uniquely defining the context in which a set of user identifiers can be used. The context can include parameters such as validity time, location, and application. Ideally, an identity domain should provide a Sybil-free environment (i. e., absent of Sybil identities) in which applications can be deployed. This paper shows how, given one initial Sybil-free identity domain, we can propagate the Sybil-freeness to arbitrary many identity domains, where in every identity domain each user is known under a different and unique pseudonym, and further there is no need for any continuous involvement of a Trusted Third Party (TTP), as access to a Certificate Authority (CA) is required only for bootstrapping a Sybil-free domain[1]. These pseudonyms are part of a cryptographic framework that we call *self-certified Sybil-free pseudonyms*.

To participate in our scheme, a user $\mathcal{U}$ must first enroll with the CA to acquire one *membership certificate* $cert_{\mathcal{U}}$. Using the membership certificate, the user can now create one *self-certified pseudonym* $P_{(\mathcal{U},ctx)}$ per newly created identity domain (where each domain has a publicly announced identifier ctx). Membership certificates can be used for issuing pseudonyms for arbitrarily many identity domains, but the pseudonyms are only valid within the domain they were issued for. Pseudonyms issued for different domains are mutually unlinkable, and they cannot be linked to their underlying membership certificate even by the CA.

Self-certified pseudonyms can be useful for applications such as anonymous communication systems, e-voting, Byzantine fault tolerance, distributed double spending detection, file sharing, reputation systems, chat rooms or forums, instant messaging, and e-commerce platforms. They are particularly suitable for infrastructureless wireless networks, as in these environments the online availability of a CA cannot be guaranteed. Therefore, we describe an application scenario in which we augment Crowds [12] with our scheme to provide Sybil-free anonymous communication in an ad hoc network. We also provide a general security analysis and a discussion on the security of the application scenario.

The paper is organized as follows. Related work is given in Section 2, while Section 3 presents our solution. Section 4 analyzes the security properties of the self-certified pseudonyms while our application scenario is described in Section 5. Finally, Section 6 concludes the paper.

WiSec'08, March 31–April 2, 2008, Alexandria, Virginia, USA.

[1]Identity domains can be trivially constructed assuming the continuous availability of a TTP. The main problems with this approach are its privacy implications, as the TTP can link all pseudonyms to the issuing user, and the need for a continuous availability of a TTP, which cannot be guaranteed in environments such as mobile ad hoc networks.

2. RELATED WORK

Below follows related work on Sybil attack protection and techniques for creating unlinkable and unique pseudonyms.

2.1 The Sybil Attack

Levine *et al.* [7] have recently provided a comprehensive survey on countermeasures against the Sybil attack for a wide range of applications. On the basis of this survey, the following strategies against the Sybil attacks were found:

- *Trusted certification*: This strategy is based on a central authority establishing a Sybil-free identity domain by binding each entity to a single cryptographic identifier (often, an identity certificate). Douceur has shown that this approach is the only approach fully capable of preventing Sybil attacks [5]. This strategy usually has an expensive setup as the initial assignment of identifiers is usually done by manual intervention. Another challenge is privacy as identity certificates can be used to profile users (multiple certificate shows are linkable).
- *Resource testing*: This strategy tries to identify Sybil nodes by distributing tasks to all network nodes, such as computing or storage capability tests. The underlying assumption is that an attacker does not possess enough resources to perform the additional tests imposed on each Sybil node. Some drawbacks regarding resource testing are listed in [5].
- *Recurring costs and fees*: A variant of resource testing where identities are periodically re-validated using resource tests. Often, this entails that each participating identity is periodically charged with a certain fee.
- *Trusted devices*: This strategy, which can be combined with trusted certification, binds one hardware device to one network entity. The main problem with this approach is that there exist no efficient way to prevent one entity from obtaining multiple hardware devices other than manual intervention.
- *Domain specific*: Some of the surveyed countermeasures are application-domain specific. For example, Piro *et al.* [11] have proposed a detection mechanism for ad hoc networks based on the fact that Sybil nodes in mobile ad hoc networks normally move in clusters.

Our proposal builds on trusted certification. The solutions analyzed in [7] belonging to this category are either only applicable for specific scenarios, requiring online interaction with a TTP, or failing to preserve user privacy. Our proposal is more general and does not suffer from these limitations. It can be compared to traditional identity certificates, but it also offers unlinkability between multiple certificate shows.

2.2 Cryptographic Related Work

Different cryptographic systems can be used to create unlinkable and unique pseudonyms. As long as the identification of "double-spent" pseudonyms is not an issue, such pseudonyms can be realized based on the so-called epoch number of direct anonymous attestation [1]. Schemes that support identification were presented in [2] and [3]. By binding a different tag to every identity domain, k-times anonymous authentication [13] can be used to create unique pseudonyms. Our scheme uses the cryptographic techniques of Camenisch *et al.* [2] (i. e., e-tokens), but can be seen as a more general systems framework that could also be instantiated using other cryptographic techniques.

3. SOLUTION DESCRIPTION

This section presents the technical description of how self-certified Sybil-free pseudonyms are constructed and used. We assume that (i) the CA is capable of establishing the initial Sybil-free domain[2]; (ii) identity domain identifiers ctx are unique; and, (iii) devices are capable of performing the necessary cryptographic functions. Regarding the attacker model, we consider attackers seeking to (i) deploy a Sybil attack in a given identity domain and (ii) identify a relationship between two pseudonyms generated for different identity domains to find out if those pseudonyms belong to the same user. We assume that attackers are able to eavesdrop on all network communication, but each attacker has at most one membership certificate $cert_{\mathcal{U}}$.

3.1 E-Token Signatures

We use a special signature scheme for creating pseudonym certificates: Camenisch *et al.* have proposed a protocol for periodically spendable e-tokens [2]. In their scenario, sensors spend an e-token whenever they report some data. Yet, it is only possible to compute k different e-tokens per time period. Consequently, sensors can file at most k reports per time period anonymously. Otherwise the sensors have to spend some e-token twice, which allows everyone to compute the sensor's identity from these two e-token show transcripts.

While k-spendable e-tokens provide the necessary main functionality for our proposal, we adapt their solution in several ways: (i) while their show protocol is interactive we require non-interactive publicly verifiable shows for signature verification; (ii) we bind a temporal public key to the e-token show – the public key is the message that is signed; (iii) instead of time periods we limit the number of generated e-tokens per signing context – while a context has a validity period, it may also have a name and other characteristics; and, (iv) we use a version optimized for $k = 1$. The first two properties are obtained by applying the Fiat-Shamir heuristic [6], a cryptographic trick that turns certain interactive identification protocols into signature schemes. Instead of a time period t, we use an arbitrary context identifier ctx. The value ctx can be seen as identifying the context in which a signer is allowed to sign only once.

The e-token based signature scheme consists of the algorithms IKg, UKg, $Obtain$, $Issue$, $Sign$, $Verify$, $Identify$, and $Revoke$. These algorithms are executed by the issuer $\mathcal{I}$ of e-token dispensers, the user $\mathcal{U}$, and the signature verifiers:

- $IKg(1^k)$ and $UKg(1^k, pk_{\mathcal{I}})$ – creates the issuers key pair $(pk_{\mathcal{I}}, sk_{\mathcal{I}})$ and the user's key pair $(pk_{\mathcal{U}}, sk_{\mathcal{U}})$, respectively. The value k is the security parameter;
- $Obtain(pk_{\mathcal{I}}, sk_{\mathcal{U}}) \leftrightarrow Issue(pk_{\mathcal{U}}, sk_{\mathcal{I}})$ – at the end of this protocol between a user and the e-token issuer, the user obtains an e-token dispenser D that can be used to create one e-token based signature per ctx. I

[2]This assumption is not exclusive for our scheme. In any certificate-based scheme, be it a fully centralized or completely distributed model (or something in between), some entity/entities *must* be trusted not to hand out more than one credential per identity.

Table 1: A summary of the notation used on the conceptual and the solution level.

Conceptual Level	Solution Level
membership certificate $cert_{\mathcal{U}}$	dispenser D
pseudonym certificate $cert_{(\mathcal{U},ctx)}$	transcript τ
pseudo-random pseudonym $P_{(\mathcal{U},ctx)}$	serial number S
public key $pk_{(\mathcal{U},ctx)}$	message m
domain identifier, context descriptor ctx	

stores $pk_{\mathcal{U}}$ and revocation information r_D under the user's identity;

- $Sign(m, D, pk_{\mathcal{I}}, ctx)$ – shows an e-token from dispenser D in context ctx to sign a message m. The outputs are a token serial number (TSN) S, a transcript τ, and an updated e-token dispenser D';
- $Verify(m, S, \tau, pk_{\mathcal{I}}, ctx)$ – checks that S and τ were created by a valid dispenser D to sign a message m in context ctx;
- $Identify(pk_{\mathcal{I}}, S, \tau, \tau', m, m')$ – given two records (S, τ) and (S, τ') created by a dispenser D when signing m and m', $m \neq m'$, respectively, $Identify$ computes the public key $pk_{\mathcal{U}}$ of the owner of D;
- $Revoke(sk_{\mathcal{I}}, pk_{\mathcal{I}}, r_D)$ – takes as input $sk_{\mathcal{I}}$ and $pk_{\mathcal{I}}$ and the revocation information r_D that corresponds to a particular user (see $Obtain$). It outputs an updated issuer public key $pk'_{\mathcal{I}}$. The dispenser D is revoked and can no longer be used to create signatures that verify under this updated issuing key.

In the rest of the paper, we assume that all parties use the most up-to-date issuer key for signing and verification. For details on the cryptographic constructions, see [8].

3.2 Instantiation based on E-Token Signatures

This section describes how to implement Sybil-free self-certified pseudonyms using e-token signatures. The pseudonym certificates $cert_{(\mathcal{U},ctx)}$ that come with the self-certified pseudonyms provide three main functions: (*i*) binding of a freshly generated public key to the pseudonym (as with identity certificates); (*ii*) verification of the pseudonym and the binding; and (*iii*) disclosure of the user identity and revocation of her certificates should the same membership certificate be used to create two different pseudonym certificates for the same identity domain.

The interaction model of our proposal consists of two "phases", one *enrollment* phase in which an initial Sybil-free identity domain is established, and one *identity domain buildup and use* phase where users create and maintain identity domains derived from the original identity domain. See Table 1 for a summary of our notation for the conceptual and the solution level, while the entities of our system and the roles they may assume are listed in Table 2.

3.2.1 Enrollment

This phase involves several users and one issuer – the certificate authority $\mathcal{I}$. Initially $\mathcal{I}$ generates an e-token issuing key pair $(pk_{\mathcal{I}}, sk_{\mathcal{I}})$ using IKg. To enroll, a user $\mathcal{U}$ creates a membership key pair $(pk_{\mathcal{U}}, sk_{\mathcal{U}})$ using UKg. She transfers $pk_{\mathcal{U}}$ to $\mathcal{I}$ and authenticates under her identity for the Sybil-free identity space. In turn, $\mathcal{U}$ and $\mathcal{I}$ interact using the $Obtain(pk_{\mathcal{I}}, sk_{\mathcal{U}}) \leftrightarrow Issue(pk_{\mathcal{U}}, sk_{\mathcal{I}})$ protocol. In this way $\mathcal{U}$ obtains an e-token dispenser D. It is used as her membership certificate $cert_{\mathcal{U}}$ (see Table 1).

3.2.2 Identity domain buildup and use

In this phase, users collectively buildup and participate in identity domains. It consists of three subphases, during which a subset of the users may take the roles of domain controller and / or verifier.

1. *Identity domain context creation.* To create a context for an identity domain, a domain controller publishes a domain identifier ctx. As a heuristic, a long-lived ctx should follow some kind of URI-like (Uniform Resource Identifier) scheme and a short-lived ctx should include the identity domain's validity time. The uniqueness of the domain identifiers used by $\mathcal{U}$ can be guaranteed under three conditions: (*i*) $\mathcal{U}$ never turns back her clock; (*ii*) $\mathcal{U}$ keeps a list of all her used domain identifiers and removes records from the list only if the corresponding identity domains have expired; and (*iii*) $\mathcal{U}$ only joins domains that have not yet expired and whose domain identifiers are not already in her list.

   In addition, ctx may contain the name of the domain, the public key of the domain controller, or even a contract that all of the users who join the domain should agree on. From a practical point of view, there is no hard limit on the size of ctx. It can be hashed down to a constant size value before being used in the cryptographic algorithms. Appending the hash to the validity time makes the uniqueness of ctx independent from the collision resistance of the hash function.

   As the identity domain controller does not need to be trusted, any user (or several users) could perform this role. Besides managing ctx, the identity domain controller will often be responsible for distributing pseudonym certificates.

2. *Pseudonym certificate creation and verification.* Registration at an identity domain is done using the triplet $(pk_{(\mathcal{U},ctx)}, P_{(\mathcal{U},ctx)}, cert_{(\mathcal{U},ctx)})$, generated as follows: a user $\mathcal{U}$ with a membership certificate $cert_{\mathcal{U}}$ wants to certify a new application specific and hitherto uncertified public/private key pair, which we will from now on call $(pk_{(\mathcal{U},ctx)}, sk_{(\mathcal{U},ctx)})$. She creates a pseudo-random pseudonym $P_{(\mathcal{U},ctx)}$ for a given ctx using the e-token to sign $pk_{(\mathcal{U},ctx)}$. The $Sign(pk_{(\mathcal{U},ctx)}, cert_{\mathcal{U}}, pk_{\mathcal{I}}, ctx)$ algorithm outputs an e-token-based signature (S, τ). $\mathcal{U}$ uses the e-token's serial number S as her pseudo-random pseudonym $P_{(\mathcal{U},ctx)}$, and the transcript τ as her pseudonym certificate $cert_{(\mathcal{U},ctx)}$ (see Table 1).

   Everyone can now verify the correctness of $cert_{(\mathcal{U},ctx)}$ using $Verify(pk_{(\mathcal{U},ctx)}, P_{(\mathcal{U},ctx)}, cert_{(\mathcal{U},ctx)}, pk_{\mathcal{I}}, ctx)$.

Table 2: A summary of the system entities and their respective roles.

Entities	Trusted	Possible Roles
Certificate Authority	Yes	Issuer
User	No	User, verifier, domain controller

Afterwards, the uniqueness of the pseudonym can be checked by comparing $P_{(\mathcal{U},ctx)}$ with the pseudonyms of the other certificates for this domain.

3. *Misuser identification and revocation.* By executing $Identify$, it is possible to extract the membership public key $pk_{\mathcal{U}}$ of a user from two pseudonym registrations $(pk_{(\mathcal{U},ctx)}, P_{(\mathcal{U},ctx)}, cert_{(\mathcal{U},ctx)})$ and $(pk'_{(\mathcal{U},ctx)}, P'_{(\mathcal{U},ctx)}, cert'_{(\mathcal{U},ctx)})$ if $P_{(\mathcal{U},ctx)} = P'_{(\mathcal{U},ctx)}$ (assured by the system) and $pk_{(\mathcal{U},ctx)} \neq pk'_{(\mathcal{U},ctx)}$. $Identify(pk_T, P_{(\mathcal{U},ctx)}, cert_{(\mathcal{U},ctx)}, cert'_{(\mathcal{U},ctx)}, pk_{(\mathcal{U},ctx)}, pk'_{(\mathcal{U},ctx)})$ will output $pk_{\mathcal{U}}$. Then, $cert_{\mathcal{U}}$ can be revoked using *Revoke*. Note that we do not view a user who reuses the same public key $pk_{(\mathcal{U},ctx)}$ as a Sybil attacker. She is just using the same short term identity again.

3.3 Efficiency

The overall costs of our system are linear in the size of the identity domain with respect to users joining the domain, and quadratic with respect to the verification of the Sybil property: every user needs to execute the *Sign* algorithm for herself, and the *Verify* algorithm for all other users. The construction in [2] requires 10 multi-base exponentiations for pseudonym certificate creation and a similar number of multi-exponentiations for verification. See [8] for a writeup of the protocol details. Using multi-base exponentiation tricks, multi-base exponentiations can be made almost as efficient as normal exponentiations. This compares to schemes that do not support identification with about half the number of multi-exponentiations, and ordinary CA-issued pseudonym certificates with one or two exponentiations. Verification may not be needed in all cases, e. g., if users trust the domain controller to verify users on their behalf, or if the application bases its security properties on the assumption that only a set of key users are not Sybil nodes, rather than every single user.

4. SECURITY ANALYSIS

This section discusses how our proposal ensures the Sybil-freeness of identity domains and the unlinkability of self-certified pseudonyms. We also discuss sharing and theft of membership certificates.

4.1 The Sybil-Proof & Unlinkability Property

Sybil-Proof Property: the cryptographic properties of e-token signatures ensure that for each valid membership certificate there can exist only one unique pseudonym $P_{(\mathcal{U},ctx)}$ per identity domain (see Section 3.1). However, as there is no inherent trust in any user in the identity domain (including the domain controller), users have to check the correctness of the pseudonym certificate $cert_{(\mathcal{U},ctx)}$ of all other users in the domain by locally running $Verify$. After an honest user has finished this verification and has checked the uniqueness of $P_{(\mathcal{U},ctx)}$, she is assured that her communication partner is a real user with public key $pk_{(\mathcal{U},ctx)}$, provided that she authenticated with $sk_{(\mathcal{U},ctx)}$.

Unlinkability Property: our approach has strong unlinkability properties as the cryptographic properties of the e-token signatures ensure the algorithmic unlinkability of two pseudonym certificates generated for different domains (see Section 3.1). However, should the users violate precautions on the network or application layers, the attacker may still be able to make an educated guess on whether two arbitrary pseudonym certificates from different identity domains are related or not. In a real word scenario, a variety of different information could help the attacker to make such a guess, for instance, the location property of the identity domain or the location of the user. A traffic analysis of each setting is required to assess the concrete threats to the users' privacy.

4.2 Membership Certificate Sharing / Theft

In order to commit an identity-based attack, an attacker must either forge a membership certificate, create multiple pseudonym certificates for the same ctx, or misuse other users' membership certificates through theft or sharing. The first two options are infeasible, as they would force the attacker to break the underlying e-token scheme (see Section 3.1). The remaining viable strategies are sharing and theft:

- By sharing c membership certificates among c corrupt users, an attack can be launched. Still, in contrast to a Sybil attack where one attacker injects c forged identities into a network, an attacker must now inject c certified (yet misused) pseudonyms in one identity domain – notably more difficult (and less effective because of its boundness) than a Sybil attack. Sharing can be hindered by equipping the users' devices with Trusted Platform Modules (TPM) or a similar hardware token, and then storing the secrets related to the membership certificate in the TPM. Another option is to include personal information in the membership certificate (e.g., a credit card number) that is automatically disclosed in case of sharing.

- A corrupted user can also try to steal membership certificates from honest users. The magnitude of such an attack would be the same as with sharing, and here, as well as in the former attack, there is a possibility for $\mathcal{I}$ to revoke the stolen membership certificates given that the attack is detected and the membership public keys are recovered through $Identify$.

Corrupted users that share membership certificates can be identified if they register two different pseudonym certificates (with different public keys) for the same identity domain. Users could agree on using the same public key, for instance by generating $pk_{(\mathcal{U},ctx)}$ according to some deterministic function f, i.e., $pk_{(\mathcal{U},ctx)} = f(ctx)$. Attackers applying this strategy can still be detected and banned from the current identity domain, but they cannot be identified[3]. However, this means that attackers have to trust each other, as all of them will know every $sk_{(\mathcal{U},ctx)}$. Therefore, the advantage for the attackers appears dubitable, when compared to the online-coordinated use of multiple identities, which is of course always possible.

4.3 Corrupt Domain Controllers

A malicious domain controller may do the following:

- She may choose a preexisting ctx. Should a user create a second pseudonym certificate for ctx, an attacker can run the $Identify$ algorithm to de-anonymize the user. This attack is thwarted by keeping a list of already

[3]Protection against this attack requires interactivity. In such a protocol the domain controller (or other online parties) could contribute randomness to the generation of the public / private key or the pseudonym certificate.

used *ctx*'s for the validity period specified by *ctx*. If a user sees an already used *ctx* when joining a domain, it is most probably a reused *ctx*.

- In cases where the identity domain controller is responsible for distributing the pseudonym certificates, she can help users that share an identity and unexpectedly joined the same identity domain to avoid identification by only publishing one of the pseudonym certificates.

- An attacker acting as a gateway between the identity domain controller and a subset of honest users may cause a partitioning of the network. Doing so, she can merely prevent sharing attackers from being identified.

While in practice all of these attacks should be considered relevant, none of them allow the attacker to break the *Sybil-proof* and *unlinkability properties*.

5. MOBILE AD HOC CROWDS

This section elaborates on a scenario for Sybil-free anonymous communication in mobile ad hoc networks (MANETs) using Crowds [12]. An anonymous communication should ideally make sure that nodes on virtual paths between senders and receivers belong to different entities, or else the anonymity properties may be compromised by an attacker controlling a large portion of the nodes in the user group [9]. Yet, in the context of infrastructureless networks, current mechanisms either fail to enforce this Sybil-freeness in a privacy-preserving manner or are not compatible with wireless infrastructureless networks, such as ad hoc networks[4]. Below, we describe how to apply our self-certified pseudonyms to augment Crowds with privacy-friendly admission control features. Before the scenario walkthrough, the chosen anonymous communication mechanism and network environment are briefly introduced.

- *Crowds* transfers messages anonymously through virtual paths created randomly based on a global probability of forwarding [12]. The user base is denoted a *crowd*, and all users in the crowd run a local *jondo* application. Also, a central *blender* application handles user memberships, i. e., enables users to join the crowd. Some of the inherent roles and concepts in Crowds fit well with those of our system (e.g., the responsibilities of the blender). The features provided by our proposal allow us to go beyond Crowds, as traditional Crowds neither protects against the Sybil attack nor provides reliable long-term identifiers.

- In *mobile ad hoc networks* it is natural to assume that users take turns in performing crucial roles. Thus, the role of the domain controller (i. e., the blender) can be passed on among the network users. This should not be seen as a disadvantage, as the domain controller does not have to be trusted (corrupt domain controllers can temporarily disrupt the network, but cannot break the Sybil-free/unlinkability properties). Also, collectively anonymizing network traffic mainly involves traffic routing, a standard procedure in ad hoc networks. Moreover, it is usually regarded acceptable to assume that users in a mobile ad hoc network occasionally have intermittent connectivity with the Internet, where they then may interact with a CA. The favorable peer-to-peer nature and adaptation of Crowds to an ad hoc scenario have been previously shown [9].

[4]It is generally easier to mount Sybil attacks in ad hoc settings as the common countermeasures in fixed networks, such as selecting nodes with IP addresses from different jurisdictions or subnetworks, are not applicable.

```
<ctx>
  <application>Crowds<\application>
  <valid_fr>2007-02-10 16:00 GMT<\valid_fr>
  <valid_to>2007-02-11 16:00 GMT<\valid_to>
  <loc>Wondercompany, Wonderland<\loc>
  <ran_nonce>0F59765E7ED3E67C<\ran_nonce>
  <issuer>pk_(U1,ctx)<\issuer>
<\ctx>
```

Figure 1: Example of a *ctx* for mobile ad hoc Crowds.

5.1 Scenario Walkthrough

- *Acquiring a membership certificate:* A prerequisite for joining an identity domain (the crowd) is that a user $\mathcal{U}$ obtained a membership certificate $cert_{\mathcal{U}}$ in beforehand during a time when $\mathcal{U}$ had connectivity with $\mathcal{I}$.

- *Creating an identity domain:* To buildup an identity domain, a user, say $\mathcal{U}_1$, becomes the domain controller, to which other users (possessing a valid $cert_{\mathcal{U}}$) may register. The domain controller chooses a *ctx* (see Fig. 1) and signs it with $sk_{(\mathcal{U}_1,ctx)}$. The context *ctx* is broadcasted and users approving of it register by sending $cert_{(\mathcal{U},ctx)}$ back to the domain controller.

- *Domain participation and verification:* In turn, enrolled users participate in anonymous communication over Crowds. Within the identity domain's validity period (specified in *ctx*), new users may join by registering at the domain controller, and existing users may leave. To inform users about newly enrolled users, the domain controller periodically broadcaststhe certified pseudonyms of enrolled users (including temporal network addresses, see Fig. 2). To ensure that the identity domain is Sybil-free (in particular, the virtual paths), all users assert that the other users possess valid certified pseudonyms by locally running $Verify$ – at least once for each pseudonym during the identity domain's validity period. Full verification may hamper scalability, especially for low-end devices. Yet, as mentioned in Section 3.3, there are strategies that could be applied to improve scalability. Further, probabilistic protection against having a Sybil node in the vulnerable first position in the path may be ensured by only verifying the correctness of the pseudonym certificate of the first user in the path and a subset of the other users. This strategy saves computational power, but it may give a false notion regarding the anonymity set size.

$$\{cert_{(\mathcal{U}_1,ctx)}, P_{(\mathcal{U}_1,ctx)}, pk_{(\mathcal{U}_1,ctx)}, \{IP, MAC\}_{\mathcal{U}_1}\};$$
$$\{cert_{(\mathcal{U}_2,ctx)}, P_{(\mathcal{U}_2,ctx)}, pk_{(\mathcal{U}_2,ctx)}, \{IP, MAC\}_{\mathcal{U}_2}\};$$
$$\ldots$$
$$\{cert_{(\mathcal{U}_n,ctx)}, P_{(\mathcal{U}_n,ctx)}, pk_{(\mathcal{U}_n,ctx)}, \{IP, MAC\}_{\mathcal{U}_n}\}$$

Figure 2: Example of a pseudonym list disseminated by the domain controller.

- *Dissolving an identity domain:* Deciding on the validity period of an identity domain is a trade-off between usability and privacy. Long validity periods reduce privacy, as users can be profiled under a certain pseudonym for a longer time. This opens the door for long-term intersection/disclosure attacks. Yet, brief validity periods increase the rate at which paths and long-lived connections in Crowds must be torn down, resulting in usability problems and performance loss. After the validity period of *ctx* has expired, the pseudonym certificates stored at the domain controller automatically become invalid. Now, to preserve unlinkability, the users should change their {IP, MAC}-pairs before entering a new domain[5].

5.2 Security Properties of the App. Scenario

Our proposal protects Crowds against the Sybil attack as a single user trying to register twice can be detected. Moreover, the domain controller (or other users) can expose the misbehaving user through $Identify$, and the CA can revoke the membership certificate with *Revoke*. Further, as described below, our scheme improves protection regarding predecessor attacks [10, 12, 14] and intersection/disclosure attacks [4] against Crowds (see [8] for more details):

- The predecessor attack is made more difficult as the Sybil-proof property helps to reduce the ratio of honest vs. corrupted users and the unlinkability property hampers the inherent assumption regarding long-lived communications.
- The validity period of *ctx* can be adjusted so that the required number of observations for an intersection/disclosure attacker is never reached by a realistic attacker. The unlinkability property of the pseudonym certificate and the {IP, MAC}-pair changes ensure that the attack has to be redone for new identity domains.

6. SUMMARY & OUTLOOK

In this paper, we presented a solution to the Sybil attack that does not require online connectivity to a TTP *and* preserves user privacy. Our scheme is particularly suitable for being deployed in infrastructureless wireless networks. We have provided a security analysis of our proposal and showed how the anonymous communications mechanism Crowds can benefit from our approach when deployed in a mobile ad hoc network. Future work includes a proof-of-concept prototype for the scenario in Section 5 as well as an experimental performance evaluation of the cryptographic primitives.

Acknowledgements

Parts of this work have been funded by the EU FP6 project PRIME and the IST FIDIS (Future of Identity in the Information Society) Network of Excellence project.

7. REFERENCES

[1] E. Brickell, J. Camenisch, and L. Chen. Direct anonymous attestation. In *CCS '04: Proceedings of the 11th ACM conference on Computer and communications security*, pages 132–145, New York, NY, USA, 2004. ACM Press.

[5]Also, the transmission power can be changed to obfuscate device location.

[2] J. Camenisch, S. Hohenberger, M. Kohlweiss, A. Lysyanskaya, and M. Meyerovich. How to win the clone wars: Efficient periodic n-times anonymous authentication. In *ACM Conference on Computer and Communications Security*. ACM, 2006.

[3] I. Damgård, K. Dupont, and M. Ø. Pedersen. Unclonable group identification. In *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 555–572. Springer, 2006.

[4] G. Danezis. Statistical disclosure attacks: Traffic confirmation in open environments. In *Proceedings of Security and Privacy in the Age of Uncertainty, (SEC2003)*, pages 421–426, Athens, May 2003. IFIP TC11, Kluwer.

[5] J. R. Douceur. The Sybil Attack. In *Peer-to-Peer Systems: Proceedings of the 1st Int Peer-to-Peer Systems Workshop (IPTPS)*, volume 2429, pages 251–260. Springer-Verlag, 7–8 Mar 2002.

[6] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO '86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer Verlag, 1987.

[7] B. N. Levine, C. Shields, and N. B. Margolin. A survey of solutions to the sybil attack. Tech report 2006-052, University of Massachusetts Amherst, Amherst, MA, Oct 2006.

[8] L. Martucci, M. Kohlweiss, C. Andersson, and A. Panchenko. Self-certified Sybil-free Pseudonyms – Introducing Privacy in Wireless Networks. Technical Report Karlstad University Studies, Karlstad University, 2008.

[9] L. A. Martucci, C. Andersson, and S. Fischer-Hübner. Chameleon and the identity-anonymity paradox: Anonymity in mobile ad hoc networks. In *Short paper proceedings of the 1st Int Workshop on Security (IWSEC 2006)*, Kyoto, Japan, 23–24 Oct 2006.

[10] A. Panchenko and L. Pimenidis. Fundamental Limits of Anonymity Provided by Crowds. In *8th Int Symposium on System and Information Security SSI'2006*, São José, Brazil, 8–10 Nov 2006.

[11] C. Piro, C. Shields, and B. N. Levine. Detecting the Sybil Attack in Mobile Ad hoc Networks. In *Proceedings of the 2nd Int Conference on Security and Privacy in Communication Networks*, Baltimore, MD, USA, 2006. IEEE Press.

[12] M. K. Reiter and A. D. Rubin. Crowds: Anonymity for Web Transactions. *ACM Transactions on Information and System Security*, pages 66–92, Apr 1998.

[13] I. Teranishi, J. Furukawa, and K. Sako. k-times anonymous authentication (extended abstract). In *ASIACRYPT*, volume 3329 of *Lecture Notes in Computer Science*, pages 308–322. Springer, 2004.

[14] M. Wright, M. Adler, B. N. Levine, and C. Shields. Defending anonymous communication against passive logging attacks. In *Proceedings of the 2003 IEEE Symposium on Security and Privacy*. IEEE, May 2003.

[15] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. SybilGuard: defending against sybil attacks via social networks. In *SIGCOMM '06*, pages 267–278, New York, NY, USA, 2006. ACM Press.