Stuart G. Stubblebine, and Paul F. Syverson. Authentic Attributes with Fine-Grained Anonymity Protection, Financial Cryptography 2000, LNCS Series, Springer-Verlag, 2000.

Authentic Attributes with Fine-Grained Anonymity Protection

Stuart G. Stubblebine* Paul F. Syverson†

Abstract

Collecting accurate profile information and protecting an individual's privacy are ordinarily viewed as being at odds. This paper presents mechanisms that protect individual privacy while presenting accurate—indeed authenticated—profile information to servers and merchants. In particular, we give a pseudonym registration scheme and system that enforces unique user registration while separating trust required of registrars, issuers, and validators. This scheme enables the issuance of global unique pseudonyms (GUPs) and attributes enabling practical applications such as authentication of accurate attributes and enforcement of "one-to-a-customer" properties.

We also present a scheme resilient to even pseudonymous profiling yet preserving the ability of merchants to authenticate the accuracy of information. It is the first mechanism of which the authors are aware to guarantee recent validity for group signatures, and more generally multi-group signatures, thus effectively enabling revocation of all or some of the multi-group certificates held by a principal.

1 Introduction

The Internet has provided an excellent opportunity for target marketing. In target marketing, sellers distinguish the major market segments, target one or more of those segments, and develop products and marketing programs tailored to each segment. Sellers focus their resources on the buyers to whom they have the greatest chance of selling. Thus, sellers try to obtain segmentation information about users such as geographic, demographic, psychographic, and behavioral information.

Buyers are typically concerned about privacy. Users may even object to the distribution of collective information about user groups. Recently, Amazon

*CertCo, 55 Broad St. - Suite 22, New York, NY 10004, USA, stuart@stubblebine.com, http://www.stubblebine.com

†Center for High Assurance Computer Systems, Naval Research Laboratory, Washington, DC 20375, USA, syverson@itd.nrl.navy.mil

introduced a service that let people see who was buying what. The intent was to do so in a manner that would not compile and post data for groups of less than 200 people. Nonetheless, privacy advocates expressed concern as did representatives of some of the profiled organizations [17]. The cited article and other early reports indicated that Amazon had no intentions of allowing customers to opt out of this profiling. Evidence of the strength of customer concern is that Amazon quickly reversed itself and allowed individuals as well as whole companies or organizations to opt out.

On the other hand, buyers are willing to provide marketing information in exchange for something of value as made evident by the success of a number of such commercial schemes giving away cash [7], Internet access [19], and computers [12]. Complementing any concerns about individual privacy, such value incentives provide motivation to defraud merchants. For example, if a merchant is offering a one-to-a-customer or one-per-address incentive he needs to authenticate that the same people are not collecting multiple times under different claimed identities. Of course this problem did not originate on-line. In some coupon scams, a few individuals would obtain a cash register to generate receipts and mail in numerous rebate coupons. These schemes were made largely impractical through software that identifies by zip code and name where funds are being sent. However, in the on-line case, unauthenticated identities and locations are even easier to produce.

Incentive programs are not the only marketing area where security is at issue. We have already mentioned concerns people have felt over forced profiling even of a fairly nonspecific nature. Still, buyers are often willing to provide personal marketing information in exchange for nothing more than convenience. Of greater concern than the ability to profile customers at a single merchant is the consolidation of the ability to gather and profile individuals across the entirety of there on-line activity Microsoft's Passport [21] is essentially a single-sign-on scheme that allows one to visit multiple sites using a single name and password. Passwords are stored only at a central Passport site and profile information is shared with other sites provided that the user gives consent. Passport thus provides some profile protection and control. Nonetheless, at least one report linked one of the recent Hotmail bugs, which generally left Hotmail account passwords exposed to an easy attack, to its integration with Passport. And, the Passport site is trusted to protect profiling information (and trusted not to abuse that information itself). Further, customers still share personal profile information albeit at their discretion.

In addition to portal based profiling it may be possible to consolidate profile information which is not explicitly centralized and even match on-line with off-line ("real world") information. The recent merger of on-line advertising firm DoubleClick and consumer data company Abacus Direct was "the most dangerous assault against anonymity on the Internet since the Intel Processor Serial Number" according to Junkbusters President Jason Catlett. "By synchronizing cookies with name and address from email, registrations and ecommerce trans-

actions, the merged company would have a surveillance database of Orwellian proportions."

What we describe in this paper is no less than an attempt to address all of the above issues on a technical level. This paper's contributions primarily fall in the category of "systems" contributions. That is, we carefully architect a system and protocols using well established cryptographic mechanisms. In particular, we propose an infrastructure for *globally unique pseudonyms* or GUPs. These are used to provide better authenticated market segmentation information than is typically available. They also protect merchants against attacks on incentive programs that can occur when recipients are not authenticated. At the same time, they can be used in various ways to protect the privacy of individuals. For example, in the profiling done by Amazon described above, it would not be necessary to opt out of the profiling; a customer could simply choose not to share employer or group information when purchasing. And, unlike Passport, there is no single site trusted with the customer's profile information and its link to the customer. Another advantage of GUPs is that they complicate the ability for multiple individuals to cooperate to produce a pseudonym and/or profile that corresponds to no one individual.

The other main innovation of this paper is the addition of the ability to show recency and do revocation for multi-group signatures. In ordinary group signatures one can prove membership in a given group, which makes them natural to use for anonymous attributes. In multi-group signatures, it is possible for a prover to show that the same principal has signed to show membership in several groups (without revealing which individual). Thus, a principal can show that he has multiple attributes together, without revealing anything else. As is common for group signature schemes, a major limitation is the inability to do revocation (or equivalently show validity more recent than in the issued certificate). We add to multi-group signatures the capacity to show recent validity using tickets. This effectively permits revocation of any or all attributes because the revoked individual will not be able to obtain fresh tickets. Although issuing short expiration periods in the tickets is the primary method of revocation, we also provide a means for revoking individual tickets.

Multi-group attributes can be instituted in conjunction with the GUPs of the first half of the paper, or can be built on top of traditional key certificates, albeit with less privacy protection than when used in combination with GUPs.

In Section 2 of the paper, we give an overview of the systems and protocols presented in the paper. In Section 3 we will describe some of the background and related work. In Section 4 we set out assumptions underlying our protocols. In Section 5 we set out a protocol for the basic issuance of GUP certificates. In Section 6 we set out a protocol for the issuance and use of GUP associated attribute certificates. In Section 7 we set out a protocol to show recent validity for multi-group signatures. In Section 8 we set out desirable security properties and discuss which of them is satisfied by which of our systems. We summarize our contributions and make some concluding observations in Section 9.

2 High Level Overview

In the basic issuance of a GUP, an individual will present a registrar with proof of his identity. The registrar contacts an issuer who can confirm whether or not that customer has ever received a global unique pseudonym (GUP) previously. Assuming not, the issuer will provide the customer with a GUP certificate binding that pseudonym to public keys for signature and encryption.

This is a very simple description of the basic system. In fact the registrar is not a single entity but is a group of principals for which the customer must contact a threshold number of them. This protects against rogue registrars allowing customers to obtain multiple GUPs. The issuers are likewise threshold entities. This will protect against disclosure of individual information as well as preventing multiple registration (by returning a false OK on the double-registration check).

GUP certificates can be used in at least two ways. First, the individual can also get attribute certificates, e.g., indicating state of residence, level of income, etc. by providing proof to a registrar and going through a similar process. These are associated with the GUP and can be useful in situations where maintaining a somewhat global pseudonymous profile is important, e.g., when trying to establish credit. (For the attributes a different threshold of registrars may be required, perhaps only a single one depending on the attribute.) Customers can also go to a validator to get a ticket indicating that an attribute certificate (or GUP certificate) is still current, e.g., that he has not registered a move since the time of certificate issuance. This is presented to a merchant, who can then be assured of the accuracy of a customer's profile. Also, the merchant can be sure that a single customer is not returning multiple times under different guises, e.g., to take advantage of one-time promotional offers.

The above design is compatible with both on-line and physical systems. For example, the customer might be providing the registrar with proof of a unique public digital ID as might be manifested in a protocol involving a current certificate from some commercial certificate authority. On the other hand, it might involve going into a bank (and a post office, etc.) and presenting a driver's license and birth certificate. When the registration process is complete, certificates could be on a smart card that the customer is carrying.

The second half of our paper concerns the issuance of multi-group user certificates and attribute memberships based on the Ateniese-Tsudik multi-group signatures [1]. As noted above we use a ticket issuance protocol to guarantee recency and to permit revocation.

3 Background and Related Work

Digital pseudonyms have been investigated for some time. The seminal work in this area is by David Chaum [5]. A customer might authenticate himself

to a merchant and then obtain a pseudonym for use with that merchant. The pseudonym is typically issued via a blind signature so that is not linkable to its owner. Thus, it is also not linkable across merchants or institutions at which the customer might have pseudonymous accounts. It is however, linkable with respect to transactions performed at the merchant, so that the merchant is able to develop and maintain a local profile and history associated with that customer. For a discussion of the 'pseudonym-like' UST mechanism that nonetheless provides locally unlinkable transactions see [25].

The use of pseudonyms we develop is complementary to that set out by Chaum. One can use our globally unique pseudonyms and any necessary attribute certificates when contacting a merchant. If it is desired to obtain a Chaumian local pseudonym[1] at this point, it should be as easy to do so as without our system. Note however that such local pseudonyms are typically transferable. Consider someone who lives in Maryland and this person wants to make use of a site open only to New Jersey residents. He can have a legitimate New Jersey resident register at the site. The legitimate resident can then give (sell?) the pseudonymous account to the Maryland resident. Local deterrents against such sharing are easily circumvented. Deterrents that tie local pseudonyms to something that the owner is globally averse to or limited from sharing/selling (e.g., as in identity escrow [15]) are more involved to implement. A mechanism that ties such deterrents into documents to prevent unauthorized publishing is described in [10]. We will return to such mechanisms later in Sections 6.3 and 7.3.[2] Attributes that are explicitly associated with GUPs do allow cross merchant profiling, although they are not directly linkable back to the customer. At the same time transfer of GUPs is limited at least to those with whom one would share all the responsibilities associated with the GUP signature. And, in our system, the GUP is tied to all the merchants with whom one pseudonymously associates, not just one (with whom one may have no interest other than, e.g., to obtain an account for an unqualified friend). Finally, GUP associated attributes allow individuals to prove pseudonymous profile information, e.g., in the establishing of credit. As originally presented, Chaum's pseudonyms allow this only on a per merchant basis. We do provide our own server specific pseudonym scheme. Unlike the original Chaum scheme, the individual is identified only by GUP when obtaining his server specific pseudonym.

There are other pseudonym management sites and services [20, 22, 29]. These provide various privacy protections for various applications—in some ways more than the systems proposed in this paper. Although, like basic Chau-

[1] Here and below we will use 'Chaumian pseudonym' to refer to the use of pseudonyms as Chaum set out in [5]. When we describe limitations on Chaumian pseudonyms, we mean only to imply areas that were not addressed rather than any limitation of the technical mechanisms.

[2] UST pseudonym tokens can be connected to global customer information, e.g., a signature key associated with the customer's publically known ID, more easily because a transaction that authenticates customer ID does not associate him with other transactions at the same merchant. In this way they are a more natural complement to GUPs. Cf. [25].

mian pseudonyms, some of their goals are complementary to ours. A nymserver like that of `nym.alias.net` is essentially an infrastructure supporting pseudonymous email communication via anonymous channels. ProxyMate provides a single sign-on pseudonym and password management system for accessing Web sites. It dynamically generates login names and passwords for sites based on the Proxymate user name and password and the address of the destination Web site. In this way it does not require the storage of user information, as opposed to Passport. The Freedom product from Zero-Knowledge Systems is an "Internet Identity Management System". Like the original Chaum design, Freedom nyms can be created for multiple separate purposes, e.g., one for each merchant the user contacts. However, local client software manages the various nyms and interfaces with their anonymous communications network. Whatever the advantages or similarities of any of these to the system design herein, none of these provides any means for guaranteeing unique identities or for authenticating property attributions, two of our main security goals. Recent work on pseudonym systems that reduces the use of a trusted center and that discourages identity sharing is presented in [16]. This work also has many of the same security goals as ours, including the two just mentioned. However, its focus is more on provability of security for theoretical systems while ours is on practical realisability of systems. Also, not all of the goals are the same. In [16], effectively even collaborating registrars and issuers cannot compromise a GUP. On the other hand, except in the case of double spending a single-use certificate, there is no provision for escrowing identity so as to be able to reveal the GUP and/or public identity of misbehaving principals.

In Section 7, we describe how to effectively permit revocation of Ateniese-Tsudik multi-group signature certificates by adding a validation ticket that must be used for the multi-group signature to be considered still valid. Multi-group signatures are themselves based on the group signature approach of Camenisch and Stadler [4]. This work made improvements over previous work in the size of group public keys and of group signatures as well as in the easy addition of new group members. The concept of a group signature, in which the signature is anonymous (relative to the size of the group) unless opened by some group manager or trusted third party, was introduced by Chaum and van Heyst [6]. A direct advance on the revocation problem was made by Boneh and Franklin in [3]. That paper presented a scheme that permitted relatively efficient revocation by permitting queries with respect to arbitrary subgroups of the original signature group. The basic, efficient scheme is limited in that any two group members can conspire to produce an unopenable and nonrevocable group member. Other schemes are presented that overcome this limitation, albeit with increased cost. Because of the focus of this paper on being able to demonstrate various attributes to various merchants or others our discussion will be in terms of the Ateniese-Tsudik multi-group signatures. But, our techniques should apply to any group signature scheme, for example, any of the above.

4 System Assumptions

Our designs makes some basic assumptions. The first one applies only to GUP based systems. For the recency guarantees associated with multi-group signatures, it is not necessary except in combined use with GUPs. The others apply to all of the systems and protocols in the paper.

- *Unique Public Identification:* Each principal can be assigned a unique identity.

An example of an attempt at this is social security numbers (SSNs) in the United States. In practice, SSNs are neither perfectly universal (not all individuals have them) nor perfectly unique (some individuals have more than one, and some are held by more than one individual). However, our design assumes that this issue is solved to an adequate degree.

- *Verifiable Public Identification:* Each principal possesses proof that his public ID is indeed his.

The nature of the proof may vary depending on the system. This may be possession of a signature key or the ability to perform a zero-knowledge proof. At least initially, it might not be electronic, e.g., possession of a passport, of a driver's license and birth certificate, etc.

- *Anonymous Communication:* Principals are not identified by communications mechanisms.

In practice today, it is quite common for on-line principals to be identified, e.g., by the IP address from which they are connecting. However, there are mechanisms available to prevent or at least complicate this identification. Also, given the possibility of spoofing, etc., it is not an adequate means of authentication in any case. Authenticating information should be passed through the data stream if needed, rather than being attempted for, e.g., the IP connection itself. We assume that, if needed, all communication is via some mechanism such as Onion Routing [14] that is designed to provide this type of anonymity.

5 GUP Protocols

Before setting out the GUP registration protocol, we introduce some notation. $\{X\}_K$ indicates the encryption of X with key K. Encryption thus represented is assumed to be an atomic operation. Similarly, $\lfloor X \rfloor_{K^{-1}}$ indicates the signing of X with private key K^{-1}. $h(X)$ indicates a hash of X. These are all assumed to have the usual desired properties wrt integrity, difficulty of computing without the appropriate secret, etc. $nonce_P$ is a nonce, assumed to be generated by principal P. The normal sending of a message M from P to Q is represented

by $P \rightarrow Q\,M$. This communication does not assume guaranteed or timely delivery, and the connection of both P and Q to it is assumed to be visible to all. If the connection of P to the communication is assumed to be hidden by some anonymizing mechanism, this is represented by $P \Rightarrow_P Q\,M$. (Similarly the recipient can be assumed hidden by the delivery mechanism—even from the sender, e.g., $P \Rightarrow_Q Q\,M$.) This notation was introduced to describe anonymous communication protocols in [23], *q.v.* for further background.

C represents an individual (customer). $\mathcal{R}$ is the registrar, which is not a single entity but a threshold group entity. When a customer presents something to the registrar, he must actually present it to some threshold number of members of the registrar group. This makes it less likely that corrupt registrars will knowingly accept inadequate proof of identity since a threshold number must be corrupted. Similarly, $\mathcal{I}$ represents a threshold group of issuers. We assume a threshold communications infrastructure such as in the Intrusion Tolerance via Threshold Cryptography (ITTC) project as described in [28]. Threshold cryptography was introduced by Desmedt and Frankel in [8].) Thus, signatures and decryptions performed by these groups are all threshold group actions. It is possible to formally represent such group actions and communications within an ordinary protocol description [27]. We do not address such representation in this paper.

5.1 GUP Registration and Issuance

The following protocol describes the interaction between an individual, the registrar group and the issuer group.

M1. $C \rightarrow \mathcal{R}$: $\quad public\_name(C)$, Proof of $public\_name(C)$

M2. $\mathcal{R} \rightarrow \mathcal{C}$: $\quad nonce_{\mathcal{R}}$

M3. $C \rightarrow \mathcal{R}$: $\quad \{K(GUP(C)), \lfloor nonce_{\mathcal{R}} \rfloor_{K^{-1}_{GUP(C)}}\}_{K(\mathcal{I})}$

M4. $\mathcal{R} \rightarrow \mathcal{I}$: $\quad \{\lfloor time_R, nonce_{\mathcal{R}}, \{public\_name(C)\}_{K(\mathcal{E})}$
$\{K(GUP(C)), \lfloor nonce_{\mathcal{R}} \rfloor_{K^{-1}_{GUP(C)}}\}_{K(\mathcal{I})} \rfloor_{K^{-1}_{\mathcal{R}}}\}_{K(\mathcal{I})}$

M5. $\mathcal{I} \Rightarrow_{\mathcal{C}} \mathcal{C}$: $\quad \lfloor time_I, GUP(C), K(GUP(C)), expire\_time_I \rfloor_{K^{-1}_{\mathcal{I}}}$

In Message 1, the customer provides the registrars with proof that he is the bearer of his public identity. As we noted above in the high level overview, this proof might take the form of face-to-face presentation of valid credentials, such as a passport. Or, it might be take the form of presentation of a digital signature and a current certificate from an authoritative issuer. In the latter case, care must be taken to confirm that the offered proof is fresh, etc. We do not attempt to represent the specifics here.

In Message 2, the registrars send the customer a nonce. This will be signed by the customer to prove that he possesses the private key $K^{-1}_{GUP(C)}$ to prevent him from trying to register someone else's public key for himself (which would allow him to get credit for the other principal's activities). Note that $K(GUP(C))$ appears in the protocol prior to the issuance of the GUP. This is simply for notational convenience. The GUP is randomly generated by the issuers to ensure uniqueness.

In Message 3, the customer proves his public name to the registrars. We assume that this proof is bound to the entire request in the message. The customer may physically show up at each of the registrars and provide physical proof of identity, or the customer may prove his identity by means of a digital signature if this is a generally available means of proving identity. The customer also provides public signing key associated with the GUP and a signed nonce in response to the registrar challenge. These are encrypted for the issuers.

In Message 4, after verifying the public name of C, each registrar forwards the request to the issuer group. This message contains the time of the registrar request, the nonce used to challenge the client, and the public name of the customer threshold encrypted under the escrow key. In the same message, the registrar forward the encrypted component supplied by C.

Upon receipt of Message 4, the issuer group uses the encrypted string of the public name to verify that the public name has not already been issued a pseudonym. It also checks that the signature on the nonce corresponds to the public key provided by the customer, that signed nonce is the same as that provided by the registrar, and that the time stamp of the message is recent. The issuer group stores the following:

$$\lfloor time_{\mathcal{I}}, GUP(C), \{public\_name(C)\}_{K(\mathcal{E})}, K(GUP(C)), expire\_time_{\mathcal{I}} \rfloor_{K^{-1}_{\mathcal{I}}}$$

Thus, in order to look up whether a given individual has registered and what GUP and key he has on file, it is necessary for a threshold number of issuers to cooperate. (Note that despite the appearance of implicit notational overload, there is no mathematical relation between the public encryption key of the issuer group $K(\mathcal{I})$ and the private signature key of the issuer group $K^{-1}_{\mathcal{I}}$. Note also that the threshold necessary to decrypt these stored data can be different from that necessary to form the signature.) The public name of C is stored public-key encrypted for an escrow authority. Should it be necessary to determine the public name of the individual associated with a particular GUP this can be done with the cooperation of the escrow authority. The issuers also initialize the validator database for the issued GUP by sending the following:

$$\mathcal{I} \rightarrow \mathcal{V}: \qquad \{GUP(C), time\_of\_last\_update\}_{K(\mathcal{V})}$$

This message is threshold encrypted for the validator group. That means that a threshold number of issuers is necessary to encrypt it. Validation will be

explained shortly.

In Message 5, the issuer group creates a certificate containing the time of the certificate issuance, the globally unique pseudonym, the pseudonym public key, and the expiration time of the certificate. This certificate is returned to the client, via an anonymous channel.

This protocol issues only a signature-key certificate associated with a GUP. If needed, a separate (or combined) certificate for a public encryption key could easily be included in the protocol.

5.2 GUP Validation

GUP certificates can be validated using traditional approaches. In essence, one needs to obtain a timestamped assertion [24] indicating that the referenced certificate is adequately fresh. As with the issuers and registrars of the registration and issuance protocol, we assume a threshold group of validators if there is concern about compromised validators. Even if needed, the validator group would probably be quite small since the potential cost of improper validation is presumably less than that of improper GUP issuance.

M6. $C \Rightarrow_C \mathcal{V}: \quad \lfloor time_I, GUP(C), K(C), expire\_time_I \rfloor_{K_{\mathcal{I}}^{-1}}$

M7. $\mathcal{V} \Rightarrow_{\mathcal{C}} \mathcal{C}: \quad \lfloor checktime, \lfloor time_I, GUP(C), K(C), expire\_time_I \rfloor_{K_{\mathcal{I}}^{-1}} \rfloor_{K_{\mathcal{V}}^{-1}}$

In Message 6, some entity such as the customer (or merchant) requests validation of a referenced pseudonym certificate. The validator group must be aware of any updates to certificates. In particular, for any updates to original certificates, it securely stores the time of the last update:

$$GUP(C), time\_of\_last\_update$$

If the certificate hasn't expired and the validator doesn't have an update time past the time of issue in the certificate, then (in Message 7) the validator asserts that the certificate is still valid at the time of the check.

6 Global Pseudonymous Attributes

In this section we describe how to obtain, validate, and use attribute certificates in conjunction with a GUP.

6.1 Issuing GUP-Attribute Certificates

We now show how to issue attribute certificates related to a GUP. The messages between C and $\mathcal{R}_A$, the attribute registrar, may be due to the customer and registrar being co-present. Thus we assume the messages between these entities

have the obvious authenticity, integrity, and confidentiality properties. If this is done remotely, cryptographic protections may need to be added. As in GUP registration, the attribute registrar may be a (threshold) group of entities to which the individual presents himself.

M8. $C \rightarrow \mathcal{R}_{\mathcal{A}}$:
$attribute\_type, public\_name(C)$, Proof of $attribute\_value$ and $public\_name(C)$

M9. $\mathcal{R}_{\mathcal{A}} \rightarrow \mathcal{C}$: $nonce_{\mathcal{R}_{\mathcal{A}}}$

M10. $C \rightarrow \mathcal{R}_{\mathcal{A}}$: $\{salt, \lfloor nonce_{\mathcal{R}_{\mathcal{A}}}, attribute\_type, attribute\_value \rfloor_{K^{-1}_{GUP(C)}}\}_{K(\mathcal{I}_{\mathcal{A}})}$

M11. $\mathcal{R}_{\mathcal{A}} \rightarrow \mathcal{I}_{\mathcal{A}}$:
$\{\lfloor time_R, nonce_{\mathcal{R}_{\mathcal{A}}}, attribute\_type, attribute\_value, h(public\_name(C)) \rfloor_{K^{-1}_{\mathcal{R}_{\mathcal{A}}}}\}_{K(\mathcal{I}_{\mathcal{A}})}$
$\{salt, \lfloor nonce_{\mathcal{R}_{\mathcal{A}}}, attribute\_type, attribute\_value \rfloor_{K^{-1}_{GUP(C)}}\}_{K(\mathcal{I}_{\mathcal{A}})}$

M12. $\mathcal{I}_{\mathcal{A}} \Rightarrow_{\mathcal{C}} \mathcal{C}$: $\lfloor time_{\mathcal{I}_{\mathcal{A}}}, attribute\_type, attribute\_value, GUP(C), K(GUP(C)), expire\_time_{\mathcal{I}_{\mathcal{A}}} \rfloor_{K^{-1}_{\mathcal{I}_{\mathcal{A}}}}$

This protocol is fairly similar to that for issuance of the GUP itself. The main difference is that the checks, what is stored, and what is sent to the validators now associates/checks attributes against a GUP and public ID rather than associating/checking a GUP and GUP key against a public IDs.

Alternatively attributes might be issued without enabling registrars to profile who has registered, i.e., without public names. For example, if compiling attacks[3] are not at issue, or if there are methods to counter them, e.g., appropriately configured smart cards, then simple bearer authentications of, e.g., some local activity or locally verifiable property can be put in certificates (not bound to a GUP unless you use a smart card and count that as the GUP).

6.2 Validating GUP-Attribute Certificates

Validation of GUP-attribute certificates is virtually the same as the validation of GUP certificates themselves. The only difference is that the attribute validators (i.e., $\mathcal{V}_{\mathcal{A}}$) store and compare

$h(GUP(C)), attribute\_type, time\_of\_last\_update(h(GUP(C)), attribute\_type)$

[3] Compiling attacks are characterized by creating a profile compiled from multiple attributes obtained illegitimately.

6.3 Server Specific Pseudonyms

We now describe a protocol for issuing server specific pseudonyms. We can enforce the property that the client is unable to get more than one identity for use with a server. A collusion between the merchant and the issuer is unable to reveal which client is accessing the service. Also, the protocol has escrow abilities whereby, given a client identifier, one can get assistance from an escrow authority to revoke access by the client. Alternatively, the escrow authority can recover the identity of a malicious client given misbehavior using an access key.

We use over-lining to indicate blinding: e.g., '$\overline{X}$' refers to the result of blinding X, for use with the appropriate signature key. $\mathcal{E}$ represents the entity trusted to uncover the keys associated with the new pseudonyms.

M13. $C \Rightarrow_C \mathcal{I}_\mathcal{A}$: $\{GUP\_attribute\_cert_1, \ldots, GUP\_attribute\_cert_m,$
$\lfloor \underline{Request\_Merchant\_Pseudonym : M;}\ K,$
$(\overline{h(K(C,M)_1, expire\_time)}, \{GUP(C), K(C,M)_1\}_{K(\mathcal{E})}), \ldots,$
$(\overline{h(K(C,M)_n, expire\_time)}, \{GUP(C), K(C,M)_n\}_{K(\mathcal{E})})\rfloor_{K^{-1}_{GUP_{(C)}}}\}_{K(\mathcal{I}_\mathcal{A})}$

M14. $\mathcal{I}_\mathcal{A} \Rightarrow_\mathcal{C} \mathcal{C}$: $\{challenge : e_{i_n}\}_K$

M15. $C \Rightarrow_C \mathcal{I}_\mathcal{A}$: $\{expire\_time, K(C,M)_{e_{i_1}}, \ldots, K(C,M)_{e_{i_{n-1}}}\}_K$

M16. $\mathcal{I}_\mathcal{A} \Rightarrow_\mathcal{C} \mathcal{C}$: $\{[\overline{h(K(C,M)_{e_{i_n}}, expire\_time)}]_{\mathcal{I}_\mathcal{A}}\}_K$

In message M13, the client requests a merchant pseudonym for merchant, M, from some gatekeeper, $\mathcal{I}_\mathcal{A}$ who insures that merchant access policy, e.g., one-per-customer, or authorization to access only one merchant of a given group of merchants, is satisfied. The client, thus, also includes any attribute certificates necessary to obtain a pseudonym for the specific merchant. Also included in the message is a session key, and tuples of a) blinded hashes of proposed certificates containing public keys and expiration times, and b) escrow elements. The escrow elements consist of a binding between the proposed public key and the GUP of the requesting entity. We assume that expiration times are chosen with course enough granularity to preclude associating them with any run of the server specific pseudonym issuance protocol. Next, in message M14, $\mathcal{I}_\mathcal{A}$ challenges C to reveal all but the one certificate. In message M15, C responds with all the proposed keys except for the one. The issuing authority verifies the correct construction of the proposed certificates and escrows. If all is in order, in message M16, it signs the remaining blinded certificate and returns it to the requesting entity.

To use this certificate, the client unblinds it and authenticates knowledge of the corresponding key when talking to the server.

7 Global Anonymous Attributes

We now give a brief overview of our second main development. Our basic approach consists of the steps of:

- Issuing Multi-group Attribute Certificates. Issuing attribute certificates by attribute issuing authorities using the Ateniese-Tsudik scheme where each joined group uses the same private key. (This private key serves as a responsibility secret for the entity.)
- Issuing Tickets. Issuing short-lived serial number tickets by attribute issuing authorities (in a manner that escrows the relationship between the GUP and the serial number ticket),
- Validating Tickets and Knowledge of Group Keys. Checking the validity of short-lived tickets by merchants and validating knowledge of group membership keys.
- Revising Group Keys. Updating group keys periodically to flush out entities having invalid group keys.
- Revoking Tickets. As an option, tickets can be revoked, cancelling even fairly recent authorizations.

Because our focus is to allow a single individual to prove multiple varying distinct attributes our discussion is in terms of multi-group signatures. However, the approach is largely independent of the specifics of the group signature scheme. It should thus be generally applicable, for example, to those mentioned in the introduction. Note that, unlike other schemes for anonymous group membership, we can restrict continued operation of a particular group member by not issuing additional tickets.

7.1 Issuing the Multi-Group User Certificate

We now describe an approach for issuing multi-group user certificates. It can be built on top of the basic GUP and attribute issuance protocols. By doing so, one can obtain many of the benefits due to the basic GUP protocol, e.g., linkage to responsibility of the GUP while hiding the true identity of the relevant principal, and restricting multiple pseudonyms for the same identity. Alternatively, it might be built on top of traditional certificate based protocols. It is thus independent but complementary of the previously presented GUP protocols. To capture this independence we use ID_c to represent, e.g., either C or $GUP(C)$. If the true identity of C is to be hidden from $\mathcal{I}$, then the communication in the following protocol should be anonymized (wrt C).

M17. $C \rightarrow \mathcal{I}$: $\{\lfloor time_{ID_c}, nonce_{ID_c}, \textit{Request for User Certificate} \rfloor_{K^{-1}_{ID_c}}, \lfloor time_I, ID_c, K(ID_c), \textit{expire\_time}_I \rfloor_{K^{-1}_{CA}}, K\}_{K(\mathcal{I})}$

M18. $\mathcal{I} \rightarrow \mathcal{C}$: $\{\lfloor nonce_{\mathcal{I}}, nonce_{ID_c}, ID_c, (g,n)\rfloor_{K_{\mathcal{I}}^{-1}}\}_K$

M19. $C \rightarrow \mathcal{I}$: $\{\lfloor nonce_{\mathcal{I}}, y, KP(x)\rfloor_{K_{ID_c}^{-1}}\}_K$

M20. $\mathcal{I} \rightarrow \mathcal{C}$: $\{\lfloor nonce_{ID_c}, (y+c)^d)\rfloor_{K_{\mathcal{I}}^{-1}}$,
$\lfloor time_{\mathcal{I}}, expire\_time_{\mathcal{I}}, ID_c, (g,n)\rfloor_{K_{\mathcal{I}}^{-1}}\}_K$

In message M17, the customer requests service. The request contains a nonce, and an indication of the type of request. It is signed using the appropriate key and includes the corresponding certificate. If this protocol is based on the previous GUP protocol, then that key is used as the signing key. Alternatively the ID_c key may be due to some traditional certificate authority (CA). At this point $\mathcal{I}$ validates that the request is recent and not a replay using the timestamp, and validates $K(ID_c)$. In message M18, $\mathcal{I}$ responds with the public parameters for a user multi-group key where ID_c is to be the only group member. Upon receipt of message M19, ID_c observes that the message is in response to his request by checking the nonce, and, according to [1], responds with $y = a^x (mod\ n)$ and $KP(x)$, proof of knowledge of x. In message M20, $\mathcal{I}$ provides membership information to ID_c and issues an explict membership certificate binding ID_c to the public key components of the multi-group key.

7.2 Issuing Multi-Group Attribute Memberships

We now describe how to enroll members in attribute groups based on [1].

M21. $C \Rightarrow_C \mathcal{I}_{\mathcal{A}}$: *Request for Attribute Membership*

M22. $\mathcal{I}_{\mathcal{A}} \Rightarrow_{\mathcal{C}} \mathcal{C}$:
$nonce_{\mathcal{I}_{\mathcal{A}}}, \lfloor time_{\mathcal{I}_{\mathcal{A}}}, expire\_time_{\mathcal{I}_{\mathcal{A}}}, attribute\_type, attribute\_value, (g',n')\rfloor_{K_{\mathcal{I}_{\mathcal{A}}}^{-1}}$

M23. $C \Rightarrow_C \mathcal{I}_{\mathcal{A}}$: $\{\lfloor nonce_{\mathcal{I}_{\mathcal{A}}}, y', KP_{[(g,n),(g',n')]}(x)\rfloor_{K_{ID_c}^{-1}}\}_{K(\mathcal{I}_{\mathcal{A}})}$,
$\lfloor time_I, ID_c, K(ID_c), expire\_time_I\rfloor_{K_{CA}^{-1}}, \lfloor time_{\mathcal{I}}, expire\_time_{\mathcal{I}}, ID_c, (g,n)\rfloor_{K_{\mathcal{I}}^{-1}}$

M24. $\mathcal{I}_{\mathcal{A}} \Rightarrow_{\mathcal{C}} \mathcal{C}$: $\{\lfloor nonce_{\mathcal{I}_{\mathcal{A}}}, (y'+c')^{d'}\rfloor_{K_{\mathcal{I}_{\mathcal{A}}}^{-1}}\}_K$

In message M21, ID_c requests to become a member of an attribute group for which $\mathcal{I}_{\mathcal{A}}$ is an authority. In message M22, $\mathcal{I}_{\mathcal{A}}$ responds with a nonce and public key information concerning the attribute group. In message M23, ID_c proves it should be a member of the attribute group by showing its attribute certificate (from our earlier GUP protocol or something similar from some more traditional certificate authority). Also, ID_c provides information concerning its private key for joining the group. Finally, following [1], ID_c must prove that they use the same secret for both the user certificate and attribute membership

by proving equality of two double discrete logarithms. This is represented by $KP_{[(g,n),(g',n')]}(x)$ where (g,n) are the parameters for the user certificate and (g',n') are the parameters of the attribute group. In message M24, $\mathcal{I}_\mathcal{A}$ issues the information needed by ID_c to join the group.

7.3 Issuing Tickets

The process of issuing tickets is similar to that of issuing server specific pseudonyms in Section 6.3

M25. $C \Rightarrow_C \mathcal{I}_\mathcal{A}$:
$\{\lfloor Ticket\_Request,\ K, (\overline{h(S_1, expire\_time)}, \{ID_c, S_1\}_{K(\mathcal{E})})^1, \ldots,$
$(\overline{h(S_n, expire\_time)}, \{ID_c, S_n\}_{K(\mathcal{E})})^n \rfloor_{K^{-1}_{ID_c}} \}_{K(\mathcal{I}_\mathcal{A})}$

M26. $\mathcal{I}_\mathcal{A} \Rightarrow_\mathcal{C} \mathcal{C}$: $\{challenge, (e_{i_1}, \ldots, e_{i_j})\}_K$

M27. $C \Rightarrow_C \mathcal{I}_\mathcal{A}$: $\{expire\_time, S_{i_1}, \ldots, S_{i_j}\}_K$

M28. $\mathcal{I}_\mathcal{A} \Rightarrow_\mathcal{C} \mathcal{C}$: $\{[\overline{h(S_{i_{n-j}}, expire\_time)}]_{\mathcal{I}_\mathcal{A}}, \ldots, [\overline{h(S_{i_n}, expire\_time)}]_{\mathcal{I}_\mathcal{A}}\}_K$

In message M25, C submits a ticket request containing n blinded witnesses to a (sufficiently large) random number and expiration time. (The ticket issuer can require "fresh" entropy of her choosing as input to the selection of the random serial number. However, the resulting number must still be sufficiently random from the issuer's perspective.) This message also contains the proposed serial number and user identifier encrypted under the public key of the escrow authority, $\mathcal{E}$. The serial number is chosen at random from a sufficiently large space that it is computationally infeasible for one to obtain the serial number by re-encrypting guesses under $K(\mathcal{E})$. In message M26, $\mathcal{I}_\mathcal{A}$ challenges C to reveal all but $n-j$ of the blinded commitments for the issuance of $n-j$ tickets. In message M27, C reveals the serial numbers and blinding factors for a subset (i.e., j) of the candidates. Due to n being adequately large with respect to j, $\mathcal{I}_\mathcal{A}$ verifies that with high probability only tickets with correct serial numbers and identifier have been submitted. This is done by verifying the blinded hash, and encrypting the serial numbers and identifiers under the key of $\mathcal{E}$. In message M28, $\mathcal{I}_\mathcal{A}$ signs the remaining blinded tickets and returns them to the C.

Should ID_c be revoked from the system, the serial number of tickets issued to ID_c can be revealed by $\mathcal{E}$ decrypting the escrowed tuples (e.g., $\{ID_c, S\}_{K(\mathcal{E})}$). We have included revocation for full generality; however, because tickets have a short lifetime, it may be considered unnecessary. If so, the protocol can be simplified and escrow eliminated.

7.4 Redemption

We give an example session of how a customer might prove he is a valid attribute member to some merchant.

M29. $C \Rightarrow_C M$: $\{K, Service\_Request\}_{K(M)}$

M30. $M \Rightarrow_C C$: $\{Required\_Attributes : A\}_K$

M31. $C \Rightarrow_C M$: $\{S_m, expire\_time, [h(S_m, expire\_time)]_{\mathcal{I}_\mathcal{A}}, KP(x)\}_K$

M32. $M \Rightarrow_C C$: $\{Service\_Granted\}_K$

In message M29, C requests service from M and establishes a session key, K. In message M30, M indicates the required membership attributes for the service request. In message M31, C provides the serial number, expiration time and the corresponding unblinded ticket (signed by the issuing authority for the required attribute). Also, C proves knowledge of x corresponding to membership of the attribute group. Upon receiving this message, M checks the signature on the ticket and checks that the ticket has not been revoked. This check can be performed by many of the traditional methods for checking revocation of certificates. Also, it verifies C's knowledge of x proving membership to the required attribute group. Assuming all checks pass, the merchant grants the service in message M32.

8 Security Properties, Security Goals, and Trust Assumptions

In this section we summarize trust assumption for our protocols and define security properties relating to profiling. We go on to discuss which of these properties are goals of the various protocols.

Summary of Trust Assumptions A summary of the basic trust assumptions of the protocols are as follows.

- The clients trust *each* registrar to protect the confidentiality of the client identity. An untrustworthy registrar can collude with the issuer to reveal the association of the user identity with the pseudonym.
- The issuers trust a threshold of registrars to validate the identity of the clients. An untrustworthy threshold of registrars could manufacture bogus identities.
- Merchants trust the system of registrars, issuers, and validators with enforcing the basic system goals of a) one globally unique pseudonym per entity, and b) accurate GUP and multi-group signature attributes.

Definition of Security Properties Profiling properties are as follows:

Attribute Profile: One or more attributes associated with a (possibly pseudonymous) principal.

Transactional Profile: One or more actions associated with a (possibly pseudonymous) principal.

Locational Profile: One or more servers associated with contact by a (possibly pseudonymous) principal.

Local Profile: Any of the above profiles, singly or in combination, in connection with a single server (merchant).

Distributed Profile: Two or more local profiles linked to the same (possibly unknown) principal.

For the following discussion we assume that the cardinality and use of attribute groups is such that principals cannot be uniquely identified (even pseudonymously) by intersecting attribute groups in any way. Discussion of degrees of anonymity that can be specified by such considerations can be found in [27]. The relationship between the above profiling properties and the security definitions in [27] is the topic for ongoing work.

We now give security goals of our various protocols using the properties defined above. We also briefly summarize the trust assumptions of the protocols. We leave precise arguments that they are satisfied for an expanded paper.

Goals of GUP issuance and GUP attribute issuance. The following properties are goals for both GUP issuance and GUP attribute issuance.

- One and only one GUP per individual.
- One and only one GUP key at a time per individual.
- No attribute profiling by fewer than threshold many attribute issuers.
- Only a threshold number of GUP issuers or GUP attribute issuers can associate a GUP and/or GUP key with a principal's public name.

In our protocols, no registrar sees the GUP. And any server (e.g., merchant) that sees a GUP cannot associate it with a public name. The salt in the GUP issuance and GUP-attribute issuance protocols prevents registrars individually or collectively from making dictionary attacks on this association.

Goals of multi-group attribute issuance. The following properties are guaranteed by the use of multi-group signatures.

- No attribute profiling by fewer than threshold many attribute issuers.
- Only one attribute value for any attribute type at a time per individual.
- Particular server specific pseudonym issuance policies can be enforced, for example, one-to-a-customer.

Goals of Server Specific Pseudonyms, and/or Multi-group Attribute Proving. The following properties are provided if clients use server specific pseudonyms, prove multi-group attributes at servers, or a combination of both.

- No distributed transactional profile by anyone: Neither colluding merchants nor colluding merchants and attribute issuers are able to form a distributed transactional profile. However, local transactional profiling may occur.
- No distributed locational profile by anyone: Neither colluding merchants nor colluding merchants and attribute issuers are able to form a distributed profile of which sites a principal visits.
- No distributed attribute profile by anyone: Neither colluding merchants nor colluding merchants and attribute issuers are able construct a distributed attribute profile.

Also for Multi-group Attribute proving we have:

- No local transactional profile by anyone.
- No local locational profile by anyone.

These properties are not provided if clients use basic GUPs at servers instead of multi-group certificates. Note that ordinary Chaumian pseudonyms provide protection against distributed transactional profiles, *except* the transactions of registration itself, assuming this must be authenticated. Likewise locational profiling and attribute profiling are not protected by Chaumian pseudonyms.

All of these properties are part of the GUP protocol. However, there is also only one embedded secret per individual enforced by the multi-group user certificate issuance. And, the attribute value and server specific policies can also be enforced for the multi-group case. Although we did not set out a server specific multi-group issuance protocol, it is a fairly straightforward use of multi-group attribute memberships.

- Principals cannot generate pseudonyms or multi-group memberships.
- Principals cannot get credit for attributes they do not hold.

- Principals cannot get credit for another's attributes or behavior.
- Principals can get credit for their own attributes and behavior.

The only way to obtain a new GUP is through the GUP issuance protocol, which requires unique proof of identity to a threshold group of registrars. The multi-group issuance protocol makes use of either a GUP or another form of unique ID to initiate. Also, the scheme in [1] makes some modifications to the basic Camenisch-Stadler approach to preclude the construction of new group members by even collaborating valid group members. Even if this were not adequate, the inability of the group members to obtain new tickets (unless they contain the escrowed identity (or GUP) of the valid member principal who signed the ticket request) would make the multi-group membership unusable. For similar reasons, principals cannot obtain attribute certificates for attributes that they do not possess. It is impossible to get credit for any attributes or behavior without possessing either a GUP signature key or a "responsibility secret" that proves unique multi-group membership. Thus, one can only get credit for another's activity with the other's direct cooperation. One cannot get credit for behavior done in the multi-group scheme (because there is no associated pseudonym). One can obtain pseudonymous credit for any attribute or behavior authenticated by one's GUP key. For local behavior, one can get credit for activity conducted under a server specific key. For more global credit, one can reveal the escrowed GUP in any given server specific pseudonym.

Many of the above properties were largely possible due to our validation and revocation techniques - particularly that of using tickets. However, such techniques are not completely secure. As with any revocation, there is a window of failure based on any non-zero freshness policy. A window of vulnerability occurs from the point where an entity is no longer authorized to be a member of a group and ending when the group key is updated. Herein, the entity may be able to use another entity's ticket. However, there is some vulnerability to the loaning entity since her identity is embedded in the escrow of the ticket. Thus she may not be completely at ease loaning out the ticket.

9 Conclusion

We have presented mechanisms for clients to maintain fine-grained anonymity control (including profile freedom) over various styles of private profile information while enabling merchants to authenticate the accuracy of information provided. In so doing, we have also introduced a mechanism to permit an individual to prove that it has been recently authorized to use a given group signature while still not revealing its identity.

References

[1] Giuseppe Ateniese and Gene Tsudik. "Some Open Issues and New Directions in Group Signatures" in *Preproceedings of Financial Cryptography: FC'99*.

[2] Dan Boneh and Matthew Franklin. "An Efficient Public Key Traitor Tracing Scheme", in *Advances in Cryptology – CRYPTO '99*, M. Wiener (ed.), Springer-Verlag, LNCS vol. 1666, pp. 338–353, 1999.

[3] Dan Boneh and Matthew Franklin. "Anonymous Authentication with Subset Queries", in *CCS'99 - 6^{th} ACM Conference on Computer and Communications Security*, ACM Press, November 1999.

[4] Jan Camenisch and Markus Stadler. "Efficient Group Signature Schemes for Large Groups", in *Advances in Cryptology – CRYPTO '97*.

[5] David Chaum "Security without Identification: Transaction Systems to Make Big Brother Obsolete", *CACM* (28,10), October 1985, pp. 1030–1044.

[6] David Chaum and Eugène van Heyst. in *Advances in Cryptology – EUROCRYPT '91*.

[7] CyberGold. `www.cybergold.com`

[8] Yvo Desmedt and Yair Frankel. "Threshold Cryptosystems" in *Advances in Cryptology – CRYPTO '89*, Springer-Verlag, 1990, pp. 307–315.

[9] DoubleClick. `www.doubleclick.com`

[10] Cynthia Dwork, Jeffrey Lotspiech, Moni Naor. "Digital Signets: Self-Enforcing Protection of Digital Information" in *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing* (STOC '96).

[11] Matthew K. Franklin and Dahlia Malkhi. "Auditable Metering with Lightweight Security", in *Financial Cryptography: FC '97, Proceedings*, R. Hirschfeld (ed.), Springer-Verlag, LNCS vol. 1318, pp. 151–160, 1998.

[12] Free PC. `www.free-pc.com`

[13] Eran Gabber, Phillip B. Gibbons, David M. Kristol, Yossi Matias, and Alain Mayer. "Consistent, Yet Anonymous, Web Access with LPWA", *Communications of the ACM*, vol. 42 no. 2, February 1999, pp. 42–47.

[14] David Goldschlag, Michael Reed and Paul Syverson. "Onion Routing for Anonymous and Private Internet Connection", *Communications of the ACM*, vol. 42 no. 2, February 1999, pp. 39–41. (More information and further publications at `www.onion-router.net`)

[15] Joe Kilian and Erez Petrank. "Identity Escrow", in *Advances in Cryptology—CRYPTO '98*, H. Krawczyk (ed.), Springer-Verlag, LNCS vol. 1462, pp. 169–185, 1998.

[16] Anna Lysyanskaya, Ronald L. Rivest, Amit Sahai, and Stefan Wolf. "Pseudonym Systems", in *Proceedings of the Sixth Annual Workshop on Selected Areas in Cryptography* (SAC '99) forthcoming in Springer-Verlag LNCS.

[17] Declan McCullagh. "Big Brother, Big 'Fun' at Amazon", *Wired News*, Aug. 25, 1999. `www.wired.com/news/news/business/story/21417.html`

[18] David Mazières and M. Frans Kaashoek. "The Design, Implementation and Operation of an Email Pseudonym Server", in *CCS'98 - 5^{th} ACM Conference on Computer and Communications Security*, ACM Press, pp. 27–36, November 1998.

[19] NetZero. `www.netzero.com`

[20] nym.alias.net `www.publius.net/n.a.n.html`. Homepage of a well known nym server described in [18].

[21] Passport from Microsoft. `www.passport.com`.

[22] Proxymate. `www.proxymate.com`. This is the system once known as LPWA, cf. [13].

[23] Michael G. Reed, Paul F. Syverson, and David M. Goldschlag. "Protocols using Anonymous Connections: Mobile Applications", in *Security Protocols: 5^{th} International Workshop*, B. Christianson, B. Crispo, M. Lomas, and M. Roe (eds.), Springer-Verlag, LNCS vol. 1361, pp. 13–23, 1997.

[24] Stuart Stubblebine. "Recent-Secure Authentication: Enforcing Revocation in Distributed Systems" in *Proceedings of the 1995 IEEE Symposium on Security and Privacy*, IEEE CS Press, pp. 224-234, May 1995.

[25] Stuart G. Stubblebine, Paul F. Syverson, and David M. Goldschlag. "Unlinkable Serial Transactions: Protocols and Applications", *ACM Transaction on Information and Systems Security*, Vol. 2, No 4, 1999. A preliminary version of this paper appears in [26].

[26] Paul F. Syverson, Stuart G. Stubblebine, and David M. Goldschlag. "Unlinkable Serial Transactions", in *Financial Cryptography: FC '97, Proceedings*, R. Hirschfeld (ed.), Springer-Verlag, LNCS vol. 1318, pp. 39–55, 1998.

[27] Paul Syverson and Stuart Stubblebine. "Group Principals and the Formalization of Anonymity", in *FM'99 – Formal Methods, Vol. I*, J.M. Wing, J. Woodcock, and J. Davies (eds.), Springer-Verlag, LNCS vol. 1708, pp. 814–833, 1999.

[28] Thomas Wu, Michael Malkin, and Dan Boneh. "Building Intrusion Tolerant Applications", in *Proceedings of the Eighth USENIX Security Symposium (Security '99)*, The USENIX Association, pp. 79–91, August 1999.

[29] Zero-Knowledge Systems. `www.zeroknowledge.com`