Anonymous Authentication With Subset Queries

(Extended Abstract)

Dan Boneh*
dabo@cs.stanford.edu

Matt Franklin
franklin@parc.xerox.com

Abstract

We develop new schemes for anonymous authentication that support identity escrow. Our protocols also allow a prover to demonstrate membership in an arbitrary subset of users; key revocation is an important special case of this feature. Using the Fiat-Shamir heuristic, our interactive authentication protocols yield new constructions for non-interactive group signature schemes. We use the higher-residuosity assumption, which leads to greater efficiency and more natural security proofs than previous constructions. It also leads to an increased vulnerability to collusion attacks, although countermeasures are available.

Keywords: Anonymous authentication. Group signature. Identity escrow.

1 Introduction

Consider an office building where each employee is given a smartcard for opening the front door to the building. Employees are often concerned that their movements in and out of the building are being recorded. Consequently it is desirable that the authentication protocol carried out between the smartcard (prover) and the door lock (verifier) does not identify the employee. This is the basic problem of *Anonymous Authentication*: a user wishes to prove that she is a member of an authorized group (e.g. employees that are allowed to enter the building), but does not want to reveal her identity.

The simplest solution to anonymous authentication is to give all employees a copy of the same secret key. This way the door lock has no information as to which employee it is authenticating. But then if a crime is committed inside the building there is no authorized *identity escrow agent* that can undo the anonymity and determine who was present in the building at the time. More seriously, there is no easy way to *revoke* a user's key without reissuing keys to all participants.

In this work, we develop new schemes for anonymous authentication that support identity escrow and key revocation. The identification transcript by itself reveals nothing further about the prover's identity. The transcript together with an "escrow" key reveals the prover's identity completely. The security of our schemes rests on the higher-residuosity assumption, first considered by Cohen and Fisher [14, 3], as well as on the difficulty of computing modular roots. The use of higher-residuosity leads to increased efficiency and more natural security proofs than previous constructions. It also increases vulnerability to an attack by colluding provers to mask their identities from the escrow agent, although countermeasures are available.

We also extend our schemes to allow a prover to demonstrate membership in an arbitrary subset of users, anonymously and with identity escrow. In our office building scenario, each door might have a different list of authorized employees. To gain access, an employee proves to a door that is on its list. Revocation of a user's key is an important special case of this feature — the subset query is simply revised to exclude everyone on a revocation hotlist.

Using the Fiat-Shamir heuristic [18], our interactive authentication protocols yield non-interactive group signature schemes of great efficiency. A group signature on a certain message can be verified by anyone, but the signature reveals no information as to which member of the group generated it.

1.1 Related work

Group signatures [11, 12, 8, 24, 2] were first described by Chaum and van Heyst. Recently, Camenisch and Stadler [10] present a solution that is much more efficient than previous solutions, although it relies on somewhat unusual security assumptions. In particular, given an RSA key e, N and an element $a \in \mathbb{Z}_N^*$ of large multiplicative order, it must be infeasible to produce a triple (x, y, v) such that $a^x = y \pmod N$ and $(y+1)^{1/e} = v \pmod N$. The heuristic nature of these security assumptions is underscored by recent vulnerabilities found by Ateniese and Tsudik [2]. Kilian and Petrank [22] present anonymous authentication schemes with identity escrow, based on similar security assumptions.

*Supported by NSF.

Anonymous authentication with subset queries, but without identity escrow, follows as an application of proofs of partial knowledge [15, 17, 16]. There are a number of proposals for payment schemes with revocable anonymity, beginning with Brickell, Gemmell and Kravitz [7].

1.2 Terminology and Definitions

We give some terminology and definitions. For a more formal treatment of an identity escrow model, we refer the reader to Kilian and Petrank [22]. In our authentication schemes, there are "users" who authenticate themselves to "verifiers". There is an "issuer" who gives a secret key to each new user when the user is added to the system. There is an "escrow agent" who examines the transcript of an authentication protocol to determine the identity of the user. A "subset query" enables the user to prove membership in some subset of the user population.

We say that an authentication protocol is "sound" if the verifier rejects with overwhelming probability when the prover is not a legitimate user. A subset query protocol is sound if the verifier rejects with overwhelming probability when the prover is not in the designated subset of users. We say that a subset query protocol is k-resilient against an "outsider attack" if no coalition of k users outside the designated subset can fool the verifier.

An authentication protocol is "anonymous" if no information about the prover's identity is revealed, other than that the prover is a legitimate user (or for a subset query protocol, that the prover is in a designated subset). This includes inferences that might be drawn from multiple executions of the protocol ("unlinkability"). Note that anonymity implies that no information about the user's secret is revealed to the verifier.

An authentication protocol has "identity recovery" if the escrow agent can determine the identity of the prover from a transcript of the protocol together with some trapdoor information. We say that a protocol is k-resilient against a "masking attack" if no coalition of k users can cause the verifier to accept while the escrow agent is unable to determine any of their identities.

1.3 Summary of Results

In Section 2, we present our basic scheme for anonymous authentication. It is sound if it is hard to extract roots modulo a composite of unknown factorization. It is anonymous if it is hard to distinguish high-order residues from non-residues. It is only 1-resilient against a masking attack. The scheme can be made k-resilient as discussed below. In Section 2.3, an instantiation is given in which the entire communication complexity is only five RSA-sized values sent over three rounds.

In Section 3, we present a scheme for anonymous authentication with arbitrary subset queries. Soundness and anonymity follow from the same hardness assumption as the basic scheme. The communication complexity is unchanged, although the work performed by prover and verifier is proportional to the size of the query. It is only 1-resilient against either an masking attack or an outsider attack. It can be made k-resilient against an outsider attack with an increase of a factor of $O(m^k)$ in the work of the verifier and prover, where m is the total number of users. Resilience against an outsider attack of *any* size is possible at a work increase of $O(\alpha)$, by limiting queries to a predetermined collection of α subsets.

In Section 4, we show how to achieve k-resilience against a masking attack for all of our schemes by applying ideas from collusion-secure fingerprinting codes [6]. This countermeasure is not particularly efficient. It is an open problem to design more efficient countermeasures that are provably secure under natural cryptographic assumptions. In Section 5, we give a summary and open problems.

2 The basic scheme

We present an efficient scheme that provides anonymous authentication with identity escrow. The scheme is 1-resilient against a masking attack. Let m be the total number of users to be authenticated. Let $\ell > m$ be the smallest prime larger than m. Our scheme is built on top of any proof of knowledge for the the ℓ'th root of a number modulo $N = pq$. For example, the scheme can be built on top of Guillou-Quisquater authentication [20].

Initialization To initialize the system the issuer performs the following steps:

1. It generates an n-bit RSA modulus $N = pq$ such that ℓ divides both $p-1$ and $q-1$ but ℓ^2 divides neither. The factors p and q are kept secret.
2. It picks a random $t \in \mathbb{Z}_N$ and sets $T = t^\ell \bmod N$. In addition the issuer sets $\mu \in \mathbb{Z}_N$ to be some ℓ'th root of unity such that $\mu \neq 1 \bmod p$ and $\mu \neq 1 \bmod q$.

The values N, T, ℓ are made public. The values t, μ are kept secret.

Issuing a key To issue a key to user number i (recall $i < \ell$) the issuer gives user i the secret key $\beta_i = t \cdot \mu^i \bmod N$. Note that β_i is some ℓ'th root of T.

Proving identity When user i wishes to authenticate itself to a verifier (e.g. a door lock) it executes the protocol below. The protocol simultaneously checks two things: (1) the user knows an ℓ'th root of T, and (2) the blinding factor r^ℓ used during the protocol is an ℓ'th residue modulo N.

Step 1: The user picks random $r \in \mathbb{Z}_N^*$. It computes $u = r^{\ell^2} \bmod N$ and $y = \beta_i \cdot r^\ell \bmod N$. It sends (u, y) to the the verifier.

Step 2: The verifier checks that $y^\ell = T \cdot u \bmod N$ and rejects if not.

Step 3: The user proves in zero-knowledge that u is an ℓ^2'th residue modulo N. He does so by proving knowledge of an ℓ^2'th root of u. Any of a number of protocols can be used for this purpose [21, 20].

The communication complexity of this protocol is logarithmic in the number of users. This is optimal from an information theoretic point of view since otherwise the transcript does not contain enough information for the escrow agent to recover the user's identity. In Section 2.3 we show an instantiation of this protocol with just three rounds of communication and seven modular exponentiations.

2.1 Proof of Security

We show that the identification protocol is sound, reveals no information about the user's secret, and reveals no information about the user's identity.

Lemma 2.1 *Let $\mathcal{P}$ be a user for which the authentication protocol succeeds with probability at least ϵ. Then there exists a polynomial time (in n and $1/\epsilon$) extractor that extracts an ℓth root of T from $\mathcal{P}$. Consequently, any prover for which the authentication protocol succeeds knows an ℓ'th root of T.*

Proof In Step 3 we use a proof of knowledge of an ℓ^2th root of u. Therefore, there exists a polynomial time (in n and $1/\epsilon$) extractor that when interacting with P extracts an ℓ^2th root of u. Let r be the extracted root. Since the verifier accepts the interaction with P we know that $y^\ell = T \cdot u \bmod N$ is satisfied. Therefore $y^\ell = T \cdot r^{\ell^2} \bmod N$ and hence $y/r^\ell \bmod N$ is an ℓth root of T. The extractor outputs $y/r^\ell \bmod N$. □

To prove the protocol is zero knowledge we show a simulator. Let

$$\mathbb{Z}_{(2)}(n) = \{N = pq \ : \ \ell | \gcd(p-1, q-1) \ , N \text{ is } n \text{ bits long}\}$$

The correctness of the simulation relies on a standard assumption that for $N \in \mathbb{Z}_{(2)}(n)$ the set of ℓ'th residues in $\mathbb{Z}_N^*$ is indistinguishable from all of $\mathbb{Z}_N^*$.

Lemma 2.2 *Assuming indistinguishability of ℓ'th residues of random $N \in \mathbb{Z}_{(2)}(n)$, the identification protocol can be simulated by a polynomial time simulator $\mathcal{S}$.*

Proof Let V be some verifier. We show how to simulate the interaction of the prover with V. First the simulator $\mathcal{S}$ picks a random $y \in \mathbb{Z}_N^*$. It computes $u = y^\ell / T \bmod N$, and outputs (u, y) as the first part of the simulation. Since in Step 3 we use a zero-knowledge proof of ℓ^2'th residuosity there exists a simulator $\mathcal{S}'$ that takes u, N and V and generates a transcript of Step 3 indistinguishable from a real transcript. We show that by concatenating (u, y) and the output of $\mathcal{S}'$ we get a transcript indistinguishable from a real transcript.

Suppose a distinguisher $\mathcal{D}$ exists. We construct a distinguisher $\mathcal{D}'$ that distinguishes between a random $r^\ell \in \mathbb{Z}_N^*$ and a random $y \in \mathbb{Z}_N^*$. This will contradict the assumption. On input $x \in \mathbb{Z}_N^*$ algorithm $\mathcal{D}'$ works as follows: First it picks a random $t \in \mathbb{Z}_N^*$ and sets $T = t^\ell$. It then computes $u = x^\ell$ and $y = t \cdot x$. It runs $\mathcal{S}'$ on u, N and V. Finally it runs $\mathcal{D}$ on the concatenation of (u, y) and the output of $\mathcal{S}'$ and outputs the same answer as $\mathcal{D}$. When x is an ℓ'th residue modulo N we give $\mathcal{D}$ the same distribution as in the real interaction. When x is random in $\mathbb{Z}_N^*$ we give $\mathcal{D}$ the simulated distribution. Hence, by definition of $\mathcal{D}$ algorithm $\mathcal{D}'$ will have the required properties. □

The simulation shows that no information about the users identity is exposed during the protocol. The simulation also proves *unlinkability*: the verifier cannot determine whether the same user interacts with the verifier multiple times.

We note that our hardness assumption is slightly different from the one introduced by Cohen and Fisher [14]. In their setting, $q-1$ is not a multiple of ℓ (while $p-1$ is still a multiple of ℓ but not ℓ^2). This would be quite dangerous for our scheme, because two colluding users could compute $\gcd(\beta_i/\beta_j - 1, N) = q$. In our settings ℓ divides both $p-1$ and $q-1$.

2.2 Identity Recovery by Escrow Agent

We now show how an escrow agent can recover the user's identity given the transcript of our identification protocol. In the simplest version of the scheme the escrow agent is given p and $\mu \bmod p$ as its secrets. It recovers the user's identity from the value y sent by the user during the identification protocol.

For an honest user i we know that $y = \beta_i r^\ell$ for some unknown r. Since $\beta_i = t\mu^i$ we have $y = t\mu^i r^\ell$. Let $\lambda = (p-1)/\ell$. Then it follows that

$$y^\lambda = t^\lambda \mu^{\lambda i} \pmod p \qquad (1)$$

Indeed, $r^{\ell\lambda} = 1 \pmod p$. Since ℓ does not divide λ it follows that μ^λ has order ℓ. Therefore, there exists a unique $i \in [1, \ldots, \ell]$ satisfying equation (1). To recover the user's identity the escrow agent tries all $i = 1, \ldots, \ell$ until an i is found satisfying condition 1. By using a "baby-step, giant-step" trick, the work of the escrow agent can be reduced to $O(\sqrt{\ell} \log \ell)$ with $O(\sqrt{\ell} \log \ell)$ precomputation and $O(\sqrt{\ell})$ storage.

The proof that a single malicious user cannot hide its identity from the escrow agent depends on a standard assumption known as the *ℓ'th root of unity assumption*. Namely, for a prime $\ell > 2$, no polynomial time algorithm can find (with non negligible probability) a non trivial ℓ'th root of unity in $\mathbb{Z}_N^*$ for a random $N \in \mathbb{Z}_{(2)}^n$.

Lemma 2.3 *Suppose user i can employ a polynomial time adversarial prover P so that (1) the verifier accepts the interaction with probability at least ϵ, and (2) the escrow agent fails to recover the user's identity from the transcript with probability at least ϵ. Then the ℓ'th root of unity assumption is false.*

Proof Pick a random $N \in \mathbb{Z}_{(2)}$. We show how P can be used to find an ℓ'th root of unity modulo N with probability at least ϵ. Pick a random $t \in \mathbb{Z}_N^*$ and compute $T = t^\ell \bmod N$. Set user i's secret key β_i to be $\beta_i = t$. Then given N, T and β_i prover P will succeed in masking i's identity with probability ϵ. Run the extractor of Lemma 2.1 on prover P. Let t' be the result. We know t' is an ℓ'th root of T. One can show that with probability at least ϵ we have

$t \neq t'$. Hence, $\mu = t/t'$ is a non-trivial ℓ'th root of unity. $\square$

A coalition of two or more users can create a new key that cannot be traced by the escrow agent. Indeed, if β_i and β_j are keys belonging to two users then $\beta = \beta_i^a \beta_j^b \bmod N$ is a valid key whenever $a + b = 1$. Furthermore, β does not reveal the identity of either user. This attack will frame an innocent user if the index $ia + jb \bmod \ell$ was assigned. If the indices are issued secretly from a sparse subset of size $m << \ell$, then the probability that a masking attack will frame an innocent user can be greatly reduced. We discuss how to make our scheme k-resilient against a masking attack in Section 4.

2.3 An instantiation using Guillou-Quisquater

We describe an efficient instantiation of our basic authentication protocol.

Step 1: The user picks random $r, s \in \mathbb{Z}_N^*$, and sends $(u, v) = (r^{\ell^2} \bmod N, s^{\ell^2} \bmod N)$ to the verifier.

Step 2: The verifier sends back a random $c \in Z_N^*$.

Step 3: The user responds with $(y, z) = (\beta_i r^\ell \bmod N, r^c s \bmod N)$.

Step 4: The verifier accepts the authentication if $y^\ell \equiv Tu \bmod N$ and $z^{\ell^2} \equiv u^c v \bmod N$.

The protocol takes only three rounds of communication and is efficient in computation. The proof of security follows from the proof of the Guillou-Quisquater protocol and Lemmas 2.1 and 2.2. Recall that the GQ protocol is only provably secure against a passive verifier (i.e. a verifier that properly follows the protocol and then tries to gain some information). Consequently our instantiated protocol is provably secure in the same model. As always, this is the price one has to pay for efficiency [25].

2.4 Group Signature

Using the Fiat-Shamir heuristic [18], our scheme can give an anonymous signature algorithm with identity escrow (i.e., a "group signature" as introduced by Chaum and van Heyst [11]). For example, consider the Guillou-Quisquater instantiation from the previous subsection. Let H be a cryptographically strong hash function. A group signature of message m is $(u, v, y, z) = (r^{\ell^2} \bmod N, s^{\ell^2} \bmod N, \beta_i r^\ell \bmod N, r^c s \bmod N)$ where $r, s \in \mathbb{Z}_N^*$ and $c = H(m, u, v)$. This technique can be applied whenever the proof of knowledge from step 3 of the authentication protocol is a public-coin protocol.

2.5 Security Enhancements

Here are some additional security enhancements that can be applied to our basic scheme. These enhancements apply to later schemes as well.

Untrusted escrow It is possible in all of our schemes to prevent the escrow agent from being able to masquerade as one of the users. The idea is that the escrow agent need not know the full factorization of N. For example, in the basic scheme we may use a modulus N that is a product of *three* prime $N = pqr$. The escrow agent need only be given p to recover the user's identity. Thus, without knowing the complete factorization of N it cannot masquerade as one of the users.

Initialization There is no need for the issuer to generate N and μ. Instead one can use the techniques of [5] so that N and μ are generated among k parties and none of them know the factorization of N.

3 Subset Queries

In this section we discuss how to add subset queries to our basic scheme. We show that the basic scheme can be extended to enable a user to anonymously prove membership in a given subset of users. Key revocation is a special case of this feature: to revoke user i's key the verifier (e.g. the doorlock) asks the prover (e.g. a user's smartcard) to prove membership in the subset P containing all users except user i. All users except for user i will be successful in their interaction with the verifier. User i will be unable to complete the proof.

We begin with a version that is simple to explain, allows every subset query, and is very efficient. Unfortunately, it is only 1-resilient to both a masking attack *and* an outsider attack. Then we give variations that increase the resistance to an outsider attack. The communication complexity of all of the protocols in this section is the same as the basic scheme from Section 2, although there is an increase in the computation performed by the prover and verifier. The ideas on which these constructions are based have appeared and reappeared in the literature; our work was most influenced by versions described in Chick-Tavares [13] and Fiat-Naor [19].

3.1 Simple Subset Query Protocol

Here is a simple scheme that allows every subset query. It is only 1-resilient against an outsider attack. The work performed by the prover and verifier increases over the basic scheme, although the communication complexity remains the same. Let m be the number of users.

Initialization The system issuer generates N, ℓ, T, μ as in the basic scheme. The issuer then computes $T^* = T^{p_1 \cdots p_m} \bmod N$ where $p_1, \ldots, p_m$ are distinct small primes different from ℓ. The values $N, T^*, \ell, p_1, \ldots, p_m$ are made public (but not T).

Issuing a key To issue a key to user number i (recall $i < \ell$) the issuer gives user i the secret key $\gamma_i = (t \cdot \mu^i)^{p_i} \bmod N$. Note that $\gamma_i = \beta_i^{p_i} \bmod N$ where β_i is as in the basic scheme.

Proving identity User i proves membership in an arbitrary set of users P as follows:

Step 1: The user picks random $r \in \mathbb{Z}_N^*$. It computes $u = r^{\ell^2} \bmod N$ and $y = \gamma_i^{w_P/p_i} \cdot r^\ell \bmod N$, where $w_P = \Pi_{j \in P} p_j$. It sends (u, y) to the verifier.

Step 2: The verifier checks that
$y^{\ell p_1..p_m/w_P} = T^* \cdot u^{p_1..p_m/w_P} \bmod N$
and rejects if not.

Step 3: The user proves in zero-knowledge that u is an ℓ^2'th residue $\bmod N$ as in the basic scheme.

Instead of proving possession of an ℓth root of T mod N as in the basic scheme, here the user is proving possession of an ℓth root of $T^{w_P} \bmod N$. In fact, the user does this by proving possession of an $\ell p_1..p_m/w_P$th root of $T^* \bmod N$. It is important that T^* is used here while T is secret, because a single user could mount a gcd-based outsider attack from $\gamma_i = \beta_i^{p_i} \bmod N$ and $T = \beta_i^\ell \bmod N$.

Zero-knowledge follows as for the basic scheme. For identity recovery, the escrow agent finds the unique solution to $y^{\lambda w_P} = t^\lambda (\mu^\lambda)^i \bmod p$, where $\lambda = (p-1)/\ell$. For soundness, we can adapt our proof of Lemma 2.1, incorporating ideas from a proof of Akl and Taylor [1]:

Lemma 3.1 *If a user $i \notin P$ can prove membership in P with probability at least ϵ, then there exists a polynomial time (in n and $1/\epsilon$) algorithm for taking arbitrary p_ith roots modulo N.*

Proof We describe a root-finding algorithm that takes as input an arbitrary $z \in Z_N^*$, and outputs $z^{1/p_i} \bmod N$. The algorithm first computes $\hat{T}^* = z^{\ell p_1..p_m/p_i} \bmod N$. It then plays the role of the prover in the authentication protocol with public values $N, \hat{T}^*, \ell, p_1, .., p_m$ and secret key $\hat{\gamma}_i = z$. In time polynomial in n and $1/\epsilon$, this produces a transcript $(\hat{u}, \hat{y})$ that the verifier accepts.

By the proof of knowledge in Step 3, there is an extractor that can interact with the prover and output an ℓ^2th root of $\hat{u}$. Let $\hat{r}$ be the extracted root. By the verification in Step 2, we know that $\hat{y}^{\ell p_1...p_m/w_P} \equiv \hat{T}^* \cdot \hat{u}^{p_1...p_m/w_P} \bmod N$. Then $(\hat{y}/\hat{r}^\ell)^{p_1...p_m/w_P} \equiv (\hat{T}^*)^{1/\ell} \bmod N$, which implies that $\hat{y}/\hat{r}^\ell \equiv z^{w_P/p_i} \bmod N$. Since $i \notin P$, we know that $\gcd(p_i, w_P) = 1$, and so there exist integers a, b such that $ap_i + bw_P = 1$. Then the root-finding algorithm can compute $(\hat{y}/\hat{r}^\ell)^b z^a \equiv z^{bw_P/p_i} z^{ap_i/p_i} \equiv z^{1/p_i} \bmod N$. $\square$

The scheme is useful in environments where there is no fear of users colluding. Unfortunately, if two users outside of P get together they can combine their secret keys γ_i and γ_j to create a new key γ that will let them prove membership in P. Indeed, from γ_i, γ_j it is easy to compute some ℓth root of T, and thus some ℓth root of $T^{w_P} \bmod N$ for any P. We refer to such an attack as an *outsider attack*. More generally, we say that a scheme is k-resilient against an outsider attack if no k users outside of P can combine their secret keys to create a new prover that will fool the verifier into thinking it is a user in P. The above discussion shows that, as is, the scheme is only 1-resilient against an outsider attack. In the next subsection, we show how to increase the resilience against an outsider attack.

As in the previous section, the above scheme is only 1-resilient against a masking attack. We discuss how to make the scheme k-resilient against a masking attack in Section 4.

3.2 Increased Resilience against Outsider Attack

We describe a generalization that achieves k-resilience against an outsider attack. It also allows every subset query. The work performed by the prover and verifier increases over the basic scheme, while the communication complexity remains the same. Despite increased resilience against an outsider attack, note that we still have the same vulnerability to a masking attack as the basic scheme.

Let $S_1, \ldots, S_n \subseteq [1 \ldots m]$ be all subsets of size at least $m - k$. Let $p_1, \ldots, p_n$ be distinct primes. Let $w_i = \Pi_{j \in [1...n]: i \notin S_j} p_j$. User i is issued the secret key $\gamma_i = \beta_i^{w_i} \bmod N$. The value $T^* = T^{p_1 \cdots p_n} \bmod N$ is made public. For any subset P of users, let $w_P = \Pi_{P \not\subseteq S_j} p_j$, and let $\bar{w}_P = (p_1 \ldots p_n)/w_P$. To prove membership in an arbitrary subset P, the user proves possession of an ℓth root of $T^{w_P} \bmod N$. It is easy for any $i \in P$ to compute such a root, since w_P is a multiple of w_i whenever $i \in P$. It is easy to prove possession of such a root to a verifier who knows T^*, by proving possession of an $\ell \bar{w}_P$th root of $T^* \bmod N$ as in the proof of identity in Section 3.1.

Zero-knowledge and identity escrow follow as for the previous scheme. For soundness versus any coalition F of size at most k, note that $[1 \ldots m] - F = S_j$ for some j. Then p_j divides $\gcd\{w_i : i \in F\}$. But if $F \cap P = \emptyset$ then p_j does not divide w_P. That is a contradiction unless the extractor can compute p_jth roots of arbitrary values modulo N [1].

3.3 Eliminating Outsider Attacks for Limited Queries

The subset query schemes that we have discussed so far support every possible subset query. Sometimes there is a smaller fixed collection of subsets $P_1, \ldots, P_a$ from which the queries will come. This is true for the office building scenario from Section 1 when there is no revocation, i.e., each door has a fixed set of authorized entrants. In this case, we can proceed as follows to get resistance against an outsider attack of *any* size, with a work increase of $O(a)$ for the prover and verifier. Choose primes $p_1, \ldots, p_a$ corresponding to the possible subset queries. Each user i is issued $\gamma_i = \beta_i^{u_i} \bmod N$, where $u_i = \Pi_{i \notin P_j} p_j$. The value $T^* = T^{p_1 \cdots p_a} \bmod N$ is made public. To prove membership in P_j, a user demonstrates possession of an ℓp_jth root of $T^* \bmod N$. The security analysis is similar to earlier cases.

Other Variations It is possible to generalize further, based on the idea of key distribution patterns (KDP's) due to Mitchell and Piper [23]. This yields a subset query scheme to prove membership in any qualified subset while resisting an outsider attack by any disqualified subset. KDP's are also a unifying idea for key predistribution schemes [4] and broadcast encryption [19]; see Stinson [26] for a helpful survey.

As with the basic scheme, the Fiat-Shamir heuristic can be applied to our subset query schemes. The result is a group signature scheme in which the signature demonstrates membership (or non-membership) in a subset of the signer's choice.

4 Defending against Masking Attacks

The schemes presented in Section 2 and Section 3 are only 1-resilient with respect to a masking attack. In other words, two users can combine their secret keys to create a new key that hides their identity from the escrow agent. This weakness is due to the fact that if $\beta_1, \ldots, \beta_k$ are ℓth roots of $T \bmod N$, then so is $\Pi_{i=1}^{k} \beta_i^{c_i} \bmod N$ whenever $\Sigma_{i=1}^{k} c_i = 1$. That gives an easy way for two or more colluders to mask their identity by producing a new key that cannot be traced back to them.

One way to protect against masking attacks is by using constructions for collusion-secure fingerprinting due to Boneh and Shaw [6]. We illustrate for the basic scheme from Section 2. Assign to each user i a distinct length-K codeword $c_i = c_{i1}, \ldots, c_{iK}$ in a collusion-secure fingerprinting code over a binary alphabet. Create K instances of our basic authentication scheme, where μ_j is a nontrivial ℓth root of unity modulo N_j, and where t_j is an ℓth root of $T_j \bmod N_j$, $1 \leq j \leq K$. The secrets that are given to user i will be $t_1 \mu_1^{c_{i1}} \bmod N_1, \ldots, t_K \mu_K^{c_{iK}} \bmod N_k$. To authenticate, the verifier and the prover execute the K instances of the basic protocol in parallel.

The important observation is that if all of the colluders' codewords agree in position j, then all of the colluders have the same secret in the jth instance of the basic scheme. Then they cannot find a new ℓth root for T_j by taking convex combinations. If the verifier accepts, the colluders must have used their common secrets in every instance where all of their codewords agree. The escrow agent will recover these codeword values, and might recover arbitrary values for all of the other codeword positions. By the properties of fingerprint codes, this suffices to recover a colluder's identity. To protect against a coalition of size k out of a population of size m with probability $1-\epsilon$, the best known fingerprinting construction requires $K = O(k^4 \log(m/\epsilon) \log(1/\epsilon))$ [6].

We can eliminate the need for multiple protocol instances if we proceed as follows. Let $\pi_1, \ldots, \pi_K$ be the first K odd primes, and let $\ell = \Pi_{j=1}^{K} \pi_j$. We construct a *single* instance of the basic authentication scheme using a modulus $N = pq$ where ℓ divides both $p-1$ and $q-1$. As in the basic scheme let μ be an ℓth root of unity, and let t be a random element of $\mathbb{Z}_N$. The secret given to user i is $\beta_i = t \cdot \mu^{a_i} \bmod N$, where $a_i \bmod \pi_j = c_{ij}$ for all j. As in the multi-instance construction, the escrow agent will be able to use the underlying fingerprint code to recover a colluder's identity from a convex combination of k or fewer secrets. Hence, we achieve k-resilience using a single instance of the protocol. The downside is that the size of the modulus N depends on the length of the fingerprinting code K.

5 Summary and open problems

In summary, the higher-residuosity assumption leads to new constructions for anonymous authentication with identity escrow and group signature schemes. Variations of the basic scheme include the ability for a user to prove membership in an arbitrary subset of users, which gives a simple approach to the key revocation problem. These constructions are efficient, and provably secure under standard security assumptions, although somewhat more awkward to protect against collusion than previous constructions.

It is an important open question to improve the defense against masking attacks while maintaining the efficiency of the basic authentication protocol. It is tempting to try to achieve this by encoding more information in each user's secret, but this could be quite dangerous. For example, let $N = p_1 \ldots p_K$. Let each μ_j be a non-trivial ℓth root of unity modulo p_j. Let each λ_j be a Chinese Remainder Theorem coefficient: $\lambda_j = 1 \bmod p_j$ and $\lambda_j = 0 \bmod p_i$ for all $i \neq j$. Let each $\beta_i = T^{1/\ell}(\Sigma_{j=1}^{K} \lambda_j \mu_j^{c_{ij}}) \bmod N$. This could directly encode an element of $\mathbb{Z}_\ell^K$ in each user's secret. However, two colluding users could factor the modulus: $\gcd(N, \beta_1^{c_{1j}}/\beta_2^{c_{2j}} - 1) = p_j$.

It seems as though a large ℓ and a secret assignment of encoded strings to users is necessary to prevent this factoring attack. Then it is possible that this approach could be developed into a defense against masking attacks, by carefully choosing the encoding information given to each user. This approach might also be useful in a setting with multiple escrow agents. By the choice of which factors go to each agent, and by the choice of each user's encoded string, the ability of different escrow agents to recover information about a user can be finely controlled.

References

[1] S. Akl and P. Taylor, "Cryptographic solution to a problem of access control in a hierarchy" *ACM Trans. Comput. Sys.* (1983) 1, 239–248.

[2] G. Ateniese and G. Tsudik, "Some open issues and new directions in group signatures", in proc. Financial Crypto '99, pp. 196–211.

[3] J. Benaloh, "Verifiable secret-ballot elections", Ph.D. Thesis, Yale University, 1987.

[4] R. Blom, "An optimal class of symmetric key generation systems", in proc. Eurocrypt '84, pp. 335–338.

[5] D. Boneh, M. Franklin, "Efficient generation of shared RSA keys", in proc. CRYPTO '97, pp. 425–439.

[6] D. Boneh and J. Shaw, "Collusion-secure fingerprinting for digital data", in proc. Crypto '95, pp. 452–465.

[7] E. Brickell, P. Gemmell, D. Kravitz, "Trustee-based tracing extensions to anonymous cash and the making of anonymous change," in proc. ACM-SIAM Symposium on Discrete Algorithms (SODA), 1995, pp. 457-466.

[8] J. Camenisch, "Efficient and generalized group signatures," in proc. Eurocrypt '97, pp. 465–479.

[9] J. Camenisch,

[10] J. Camenisch, M. Stadler, "Efficient group signature schemes for large groups", in proc. CRYPTO '97, pp. 410–424.

[11] D. Chaum and E. van Heyst, "Group signatures", in proc. Eurocrypt '91, pp. 257–265.

[12] L. Chen and T. Pedersen, "New group signature schemes", in proc. Eurocrypt '94, pp. 171–181.

[13] G. Chick, S. Tavares, "Flexible access control with master keys", in proc. CRYPTO '89, pp. 316–322.

[14] J. Cohen and M. Fisher, "A robust and verifiable cryptographically secure election scheme", in proc. IEEE Symposium on Foundations of Computer Science, 1985, 373–382.

[15] R. Cramer, I. Damgård, B. Schoenmakers, "Proofs of partial knowledge and simplified design of witness hiding protocols", in proc. Crypto '94, 174–187.

[16] A. De Santis, G. Di Crescenzo, G. Persiano, "Communication-efficient anonymous group identification", in proc. 3rd ACM Conference on Computer and Communications Security, 1998, pp. 73–82.

[17] A. De Santis, G. Di Crescenzo, G. Persiano, M. Yung, "On monotone formula closure of SZK", in proc. of IEEE Foundations of Computer Science, 1994, pp. 454–465.

[18] U. Feige, A. Fiat, A. Shamir, "Zero-knowledge proofs of identity", *J. Crypt.* vol. 1, 77–94, 1988.

[19] A. Fiat and M. Naor, "Broadcast encryption", in proc. Crypto '93, pp. 480–491.

[20] L. Guillou and J.-J. Quisquater, "A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory", in proc. Eurocrypt '88, pp. 123–128.

[21] S. Goldwasser, S. Micali, C. Rackoff, "The knowledge complexity of interactive proof systems", *SIAM J. of Computing*, vol. 18, pp. 186–208, 1989.

[22] J. Kilian, E. Petrank, "Identity escrow", in proc. CRYPTO '98, pp. 169–185.

[23] C. Mitchell and F. Piper, "Key storage in secure networks, *Discrete Applied Mathematics*, vol. 21, pp. 215–228, 1988.

[24] H. Petersen, "How to convert any digital signature scheme into a group signature scheme", in proc. Security Protocols Workshop, Paris, 1997.

[25] C. Schnorr, "Efficient signature generation by smart-cards", J. Cryptology, vol. 4, pp. 161–174, 1991.

[26] D. Stinson "On some methods for unconditionally secure key distribution and broadcast encryption", *Designs, Codes and Cryptography*, vol. 12, pp. 215–243, 1997.