Scalable Anonymous Authentication

6.857 Computer and Network Security, Spring 2015

Asya Bergal, Benjamin Tidor, Eeway Hsu and Catherine Zuo
https://github.com/anon-auth

Abstract

Anonymous authentication schemes allow members of a dynamic group to identify themselves to a central authority. Users can prove that they are a member of the group without revealing anything more about their identity. With traditional schemes[12][10], message sizes scale linearly with the number of users in the group, making these schemes infeasible to deploy in large institutions. We present an anonymous authentication protocol and proof-of-concept implementation that limit message size at the expense of adding some restrictions to which groups may be authenticated. We show that this scheme is flexible and scalable enough to be deployed on a college campus, and we argue that it also preserves anonymity and security.

1 Introduction

Authentication of membership to a group is an integral task in maintaining the security of an access-restricted system. Most systems used in practice require an individual to uniquely identify themselves during the authentication process. The individual's ID is thus received by the system and often saved, creating an access log that raises questions about privacy. The concept of anonymous authentication challenges the assumption that a unique identifier is necessary. These schemes offer the same capabilities as traditional schemes while hiding the identity of the authenticating user.

We present Scalable Anonymous Authentication, an anonymous authentication scheme designed to work seamlessly within the Massachusetts Institute of Technology community. MIT consists of approximately 20,000 individuals, comprising students, staff, and faculty. Each individual is given a set of access permissions and issued an ID card used for authentication. In our scheme, a card is an agent executing the protocol on behalf of an individual. There are approximately 1,000 limited-access doors on campus, although some share an access list. Each door is connected to the central card network and is managed by the administrator. A door executes the protocol of an authenticator in the system.

We believe that a compelling scheme must provide anonymity and revocability in an offline environment. Given our target use case, the Scalable Anonymous Authentication scheme must additionally offer the properties of extensibility and scalability.

1.1 Outline

In Section 2, we provide motivation for our project. We consider a few desirable traits of an authentication scheme and how current systems fail to meet these needs. In Section 3, we formally define the requirements of our scheme. In Section 4, we describe our adversarial model. Then, in Section 5, we present the protocol used in our Scalable Anonymous Authentication scheme. Section 6 includes an in-depth analysis of the scheme and discusses its strengths and weaknesses. Included in our work is a proof-of-concept built using two Android devices communicating over Near Field Communication (NFC). Section 7 provides an overview of this proof-of-concept implementation and describes practical limitations of the system. In Section 8.2, we respond to shortcomings of our system and develop extensions and possible alternatives. In Section 9, we review related work.

2 Motivation

The motivation for Scalable Anonymous Authentication comes from four common traits of an authentication scheme — anonymity, scalability, flexibility and security.

2.1 Anonymity

Anonymity and the privacy it allows are important to many people. From a security perspective, lack of anonymity allows an individual to be targeted and tracked by malicious actors.

The current MIT access system is based on Indala FlexISO Proximity Cards containing passive Radio-Frequency Identification (RFID) tags. To authenticate these cards, card readers are installed at all limited-access doors. Each reader is connected to a panel, which stores the IDs of all individuals with membership to the door and connects to the central MIT card database [1]. Because the door verifies membership in a group with these individual IDs, the system can track every successful and unsuccessful access attempt. As of 2002, timestamped, individually-identifiable access reports are stored for two weeks[3].

2.2 Scalability

Scalability is an important factor when performing anonymous authentication in large groups such as college communities. To be realistic and usable, authentication time and even card storage should not linearly increase with group size.

Many existing anonymous authentication schemes are largely unscalable because the individual must store information about each member of the group. For example, schemes based off ring signatures require each user to have the public key of all group members. As the number of members increases, the size of this information increases. Additionally, it is difficult in such schemes to change group membership and add or remove public keys from an individual's card without leaking information. In schemes where individuals are given unique keys, the door must broadcast many encrypted group IDs, each decryptable by a different member of the group. This scheme supports easy revocation, but authentication time and card storage requirements increase linearly with each additional individual in the group [12].

2.3 Flexibility

Flexibility allows the administrator to add and remove members to a group at will. In the case of college communities, each year a quarter of the students graduate and must lose access. More students arrive and make their way onto various access lists. Year to year, living situations change and dormitory access must be modified. Month to month, faculty, staff and students are added and removed from lab groups and office spaces. Day to day, loss and theft occurs; IDs must be immediately revoked and quickly replaced. College communities are dynamic groups; an effective system must flexible.

A number of current anonymous authentication schemes create a group identifier, sharing it (usually encrypted, or secured through other means) with individuals as they join. In the situation where a member is revoked, a new identifier must be created and remaining members must be in effect transferred to a new group. Such schemes make supporting large dynamic groups slow and unsatisfactory.

2.4 Security

An authentication system should be secure against outside adversaries seeking to gain access. Both the administrator and individuals have this goal in mind.

MIT's proximity card system utilizes passive RFID tags. With the passive RFID tags, each card broadcasts the same data with every use. An attacker can easily extract data and duplicate a card [4]. Unless we can ensure that attackers don't exist in proximity to campus, the MIT card system is clearly insecure.

There are however, various measures that can be taken. One method to combat risk of duplication is the use of active RFID cards. Another possibility is to send a challenge and response to ensure that information is only shared with trusted authenticators. We consider only anonymous authentication schemes because we believe the privacy of an individual from the system is a necessary goal. Using anonymous authentication, information useful for tracking and monitoring an individual is never created. However, our anonymous scheme must also be secure and will incidentally offer an improvement over the current system in this regard.

3 Protocol Requirements

Scalable Anonymous Authentication improves upon the shortfalls discussed in Section 2. In context of our target use case, we define the requirements of anonymity, extensibility, revocability, offline operation, and scalability.

Anonymity means that the administrator gains no information about an authenticating card besides the result of the authentication attempt. Formally, in one exchange the administrator may select a subset of individuals (possibly subject to some restrictions). The only information the administrator learns about the card is whether or not the card is a member of that set.

Extensibility means that new individuals may be provisioned even after the initial set-up phase. This has the effect of adding the new individual to A, the access set. Our system supports unlimited extensibility: an arbitrary number of individuals may be added even after the set-up phase.

Revocability means that individuals who have access to a door may later lose that access. An ideal system allows the administrator to select an arbitrary access set A for each authentication attempt. In our weaker model, once a individual is removed from A, they may not be re-added. Our system supports parametrized revocability: up to r individuals may be removed from A before the system ceases to function.

Offline operation means that cards do not have a back-channel form of communication like a WiFi connection. All information available to a card was either pre-programmed at the time of issuance or transmitted over the door-card interface (via NFC, for example).

Scalability means that the amount of data sent over the door-card interface during each exchange is limited. Many traditional schemes send $O(n)$ messages, where n is the number of individuals in the system[12]. Our scheme provides parametrized scalability, sending only $O(r)$ messages for a parameter r. By adjusting r, the system administrator may trade speed for revocability.

4 Adversarial Model

In our model of anonymous authentication, the administrator is responsible for maintaining the system and protecting its security (note that we treat the doors as an extension of the administrator). However, the administrator is not responsible for preserving anonymity. An anonymous authentication system should protect individuals' identities in the face of a malicious administrator who might deviate from the protocol, even in ways that reduce security.

Both the administrator and the user are responsible for protecting the security of the system. If an administrator's secrets are compromised, an attacker may be able to falsely authenticate to a door. If a user's secrets are compromised, an attacker may be able to impersonate that user until that user is revoked.

5 Scalable Anonymous Authentication

5.1 Protocol Overview

The Scalable Anonymous Authentication scheme achieves anonymity by using a verifiably common password instead of a per-individual identifier to authenticate individuals. It achieves revocability by changing the password when a individual is revoked. And it achieves offline operation by pre-generating all passwords and distributing them to individuals in advance, *in hidden form.* Passwords are hidden using Shamir's Secret Sharing scheme in such a way that the administrator may choose who to reveal the password to at a later time.

5.2 Basic Protocol

We begin by describing a simplified version of the protocol, then discuss the necessary augmentations to improve anonymity and security.

The protocol is parametrized by r, the maximum number of revocations. We additionally define a series of $r + 1$ polynomials $f_0(x), f_1(x), \ldots, f_r(x)$, where the k^{th} polynomial has degree k. All polynomial arithmetic is performed in $GF(p)$, where p is a large prime. With these polynomials, the k^{th} password is encoded as $f_k(0)$. For each polynomial $f_k(x)$, there is a set of public points P_k sampled from that polynomial. There is also a list of revoked users.

In this protocol, we describe the administrator performing *centralized* actions such as generating polynomials and taking samples (choosing samples for P_k, in particular, should only be done once per polynomial). Doors, on the other hand, are deployed devices that authenticate users. We assume that the administrator shares all information with the doors.

5.2.1 Set-up

1. Administrator selects a value for r.

2. Administrator generates $f_0(x), f_1(x), \ldots, f_r(x)$ with random coefficients.

3. Administrator initializes polynomial index $k \leftarrow 0$.

4. Administrator defines $P_0 = \varnothing$. The remaining P_k are initially undefined.

5.2.2 Provisioning a Card

The administrator can add individuals to the system by sampling r *private points* $f_0(u), f_1(u), \ldots, f_r(u)$ using a unique individual identifier u. These private points are installing onto the card at the time of issuance. In order to support revocation, the administrator must record which individual identifier corresponds to which individual.

5.2.3 Authentication

1. Door broadcasts k and P_k.
2. Card combines P_k with $f_k(u)$ to obtain $k+1$ points sampled from f_k.
3. Card interpolates these points to recover the password, $f_k(0)$.
4. Card sends $f_k(0)$ to the door.
5. Door verifies the card's response and outputs Success iff it is correct.

5.2.4 Revocation

1. Administrator increments $k \leftarrow k+1$.
2. Administrator adds the individual to the list of revoked users, bringing its size to k.
3. Administrator sets $P_k = \{f_k(a), f_k(b), f_k(c), \ldots\}$, where $a, b, c, \ldots$ are the identifiers of all revoked users.

Once $k = r$, no more revocations may be performed.

When the revoked individuals attempt to authenticate, they will be asked to produce the password $f_k(0)$. Revoked individuals, however, have a private point that is repeated in the set of public points. They will therefore see only k unique points instead of $k+1$ and will be unable to recover the password. Because k increases after revocation, no revoked individual will have yet created this password either. Authorized individuals will be able to combine the new public points P_k with their private point to recover the password.

5.3 Preventing De-anonymization

The Basic Protocol described above leaves individuals vulnerable to de-anonymization by a malicious administrator. By having the door broadcast a fake set of points not sampled from the original polynomial, the administrator can cause each individual to calculate a different password. Because the administrator knows which individual was issued which private point, she can examine the password in the response to determine the individual who calculated it.

In order to remedy this vulnerability, we must ensure that every individual responds the same way to a given challenge. To do this, we modify the protocol so that the door broadcasts $hash(f_k(0))$ in addition to k and P_k. This commits the door to a password and allows the card to verify that the hash matches before sending the password in its response.

5.4 Replay Protection

In order to prevent eavesdroppers from gaining access to a door by replaying the password, we use a challenge-response scheme to prove that the card knows the password without sending it over the wire. The door broadcasts a challenge c, and instead of replying with $f_k(0)$, the card replies with an HMAC using $f_k(0)$ as the key and c as the data. The door can verify this MAC by calculating it independently. This scheme protects against replay attacks so long as implementors take care not to re-use challenges.

6 Analysis

6.1 Complexity

In an exchange, the door broadcasts k points, a fixed-length hash value and a fixed-length challenge. The card responds with a fixed-length HMAC. In the worst case, k is as large as r. As the system scales, the size of the messages transmitted over the door-card interface grows as $O(r)$. The door stores one private point from each of $r + 1$ polynomials. The storage space required on the card then grows as $\Theta(r)$.

6.2 Anonymity

The modifications described in Section 5.3 make the Scalable Anonymous Authentication scheme fairly resistant to de-anonymization by a malicious administrator. If the exchange proceeds properly, the contents of the card's reply (i.e. the HMAC) are identical no matter which user is authenticating. If the exchange does not proceed properly, the card does not reply. Depending on the underlying technology, the door may not even notice that a card attempted to authenticate. Because a card can only either send a correct reply or no reply at all, this scheme meets the definition of anonymity presented in Section 3. However, a successful authentication must leak some information about the authenticating user (namely, the fact that the user is on the access list). Because the administrator has near-complete control over the access list, this enables a class of "one-guess attacks" against anonymity.

6.2.1 One-guess Attacks

In a one-guess attack, the administrator must guess which user is attempting to authenticate during an exchange. A simple form of this attack is alluded to in Section 5.3: if the door broadcasts a fake set of points *and* the administrator guesses which user is attempting to authenticate, calculates which hash will result and broadcasts that, the administrator can determine whether or not their guess was correct. If the administrator can link successive authentication attempts, she can identify an unknown user with n exchanges.

Note that some technologies may prevent the administrator from receiving any feedback during an unsuccessful authentication attempt. In this analysis, we assume the worst. In the real world, an administrator may have outside sources of information that record when users attempt to authenticate.

In a more sophisticated attack, the administrator revokes half of the users and observes whether the exchange succeeds or fails. If the administrator can link successive authentication attempts, she can binary search the space of users to identify an unknown user with $\log n$ exchanges. In Section 8.1.1 we discuss a possible solution for this issue.

In the above attacks, the administrator gets one "guess" per exchange. If an individual makes repeated authentication attempts, many exchanges will occur. In Section 8.1.2, we describe mitigation for this issue.

6.3 Security

With the modification described in Section 5.4, our protocol is protected against replay attacks so long as the legitimate card's response reaches the reader and causes the challenge to change. Like many systems, wireless authentication schemes are vulnerable to *relay* attacks. If an attacker can perform an exchange with a legitimate card (by installing a fake card reader in a popular location, for instance), they can relay traffic between a target reader and the victim's card to open a door of their choice. Forestalling these sorts of attacks is an area of active research[19]. As a mitigation, we recommend that implementors expire challenges frequently.

6.4 Revocability

The limitation on revocations is r, the degree of our largest secret storing polynomial. When the cumulative number of revoked users exceeds r, the system must be reset and all cards reissued. In situations where mass revocations are planned in advance, it is possible to make optimizations, which section 8.2 explores in more detail. While Scalable Anonymous Authentication offers less-than-perfect revocability, it allows the administrator to select a value for r that trades revocability for speed according to their needs.

6.5 More Properties

Our scheme is extensible because it allows the administrator to add new cards after the set-up phase. It also is offline, as cards are not connected to a wider network and may only communicate through the door.

7 Implementation

We create an Android implementation of our protocol, consisting of two Android applications: one for the card (`anon-auth-card`) and one for the door (`anon-auth-door`)[1]. Both applications are preloaded with hard-coded polynomials. We use NFC (Near-Field Communication) to execute our protocol, enabling our card application to act like a smartcard by using Android's Host Card Emulation feature,. We base our application off of Google's card emulation and card reader examples. Communication happens over the ISO-DEP (ISO 14443-4) smart card communication protocol.

The door first sends a Select Service APDU (Application Protocol Data Unit) command to the card. The card responds with a `SELECT_OK_SW` message. The door then sends out a Broadcast APDU command containing a byte encoding of the public points, secret hash, polynomial degree, and challenge. The card interpolates the secret, verifies the hash, and sends back a byte encoding of HMAC(K = secret, M = challenge). After confirming the HMAC, the door application displays a "Door Open" message on the screen. In our testing, we have shown that the door app correctly recognizes a successful authentication exchange and correctly rejects an invalid one.

The heart of our scheme's proof-of-concept is in our cryptography library, `libanonauth`. We port a public Python implementation of Shamir's Secret Sharing System [18] to use in our library. Our `ProtocolCard` and `ProtocolDoor` classes contain all of the methods necessary for the card and door, respectively, to carry out the protocol. The `SecretBox` object represents an instance of a Shamir secret; it holds a specific polynomial whose y-intercept is the secret. Secret shares are represented by `Point` objects; their x-coordinate and y-coordinates are of size 2 and 16 bytes, respectively. `SecretBox` also has the `secretHash` method to generate a 32-byte hash of its secret, and the `hmac` method which generates a 32-byte HMAC using its secret and challenge. The `libanonauth` library includes a set of unit and integration tests that demonstrate its functionality. These tests include simulated card-door exchanges and a simulated revocation.

A `ProtocolDoor`'s challenge is a randomly generated, positive 128-bit (16 byte) number. Currently, we have our door generate a new challenge with every successful authentication. The door resets the challenge to a new random value with every successful authentication. Ideally, it should also automatically generate a new challenge around once every 10 seconds to prevent replay attacks.

[1] The source code is published at `https://github.com/anon-auth`.

For example, if Eve captures the door's broadcast and then uses a fake door to transmit this broadcast and capture a Alice's card response, she can successfully replay this snooped response to the real door because the challenge has stayed the same. We left off this feature because of time constraints, but a future extension that implements this would significantly improve the Android apps' security.

Traditional group authentication schemes transmit $O(n)$ bits to specify group membership. Our system is parametrized by r, the maximum number of revocations (in the cryptography library, the maximum polynomial degree). Transmission and card storage are then $O(r)$. Assuming an ideal NFC speed of 53 KB/sec, a 1-second authorization corresponds to $r \approx 1500$.

r	Data transmitted (KB)
100	3.3
1,000	33.2
10,000	332

Figure 1. Transmission size given r

Through Figure 1, we see that the amount of data transmitted is linear with growth in r. To minimize the need for large r while retaining support for dynamic groups, we can use the pre-planned revocations extension described in Section 8.2. As data transmission rates in on-market hardware increase, we can also see the efficiency of this protocol improve.

Our proof of concept uses NFC, which establishes a connection between two devices before sending data. In practice, a stronger implementation would use radio or some other one-way broadcasting system to broadcast the points, so that a reader has no knowledge that someone is at the door. This denies the reader feedback when an authentication attempt fails. Traditional static NFC tags also have a UID which the reader can access. Our phone application does not have such a static UID, but in practice a smartcard-based application should take care to randomize this UID on every activation to prevent de-anonymization.

8 Further Work

8.1 Mitigating One-guess Attacks

8.1.1 Trusted Third Party

One possible solution to the one-guess anonymity attack is to have a trusted third party verify all revoked points broadcast by the door. This would require de-anonymizing all revoked users, publicly revealing all of their points. The trusted third party would independently verify that points correspond to legitimate users that have been revoked, rather than temporary revocations used by the door in a malicious one-guess attack. The door would then additionally broadcast a list of all revoked points, signed by the trusted third party. Cards would verify this signature and confirm that broadcast points were valid before attempting to access the door.

8.1.2 Card Response Rate Limiting

In the situation where a one-guess attack is deemed permissible, there should be a check in place to limit the number of attempts a malicious administrator can guess. We create an extension where an additional step is added to the card protocol. When a card has confirmed a hashed secret from the door but been denied access, the card is locked from sending a response to the door for a short amount of time. This rate limiting is used to extend the average time a malicious administrator must take to de-anonymize an individual. The additional lock time can also notify the individual of possible attack and deter sending excessive responses to a malicious door.

8.2 Pre-planned Revocations Extension

To take advantage of large groups of pre-planned revocations, we create an extension which changes changes two parameters of the set-up and card provision protocol. First, instead of having polynomials of degree 0 to r, we will have polynomials of degree 1 to $r+1$. Then, instead of receiving only one individual share of the secret, an individual is provisioned 2 points per polynomial. The first point, an individual share of the secret, remains the same. The second point is the date point, shared amongst all individuals with the same revocation date. When a pre-scheduled revocation date arrives, only one point per polynomial, the shared date point, is publicized and now part of the door broadcast. The intuition used in Shamir's Secret Sharing, the idea that it takes $t+1$ points to define a t degree polynomial means individuals part of a group revoke will now be short one point. With one less point, no information can be gained.

This date revocation method decreases the necessary size of r while maintaining the ability to support dynamic groups. It slows the growth rate of k, the current polynomial degree. The overall polynomial degrees is shifted up by one, increasing the computation time a negligible amount. However, as two points, instead of one, are being stored on card, the data stored doubles in size.

8.3 Benevolent System Incentive Scheme

An interesting field we hope to further research involves anonymous access schemes which incentivive an administrator to conform to protocol. We consider a system where deviating from protocol diminishes the security of the system.

9 Related Work

9.1 Group Signatures

Kilian and Petrank proposed an escrowed identity scheme using group signatures. Identity escrow provides anonymous authentication. However, while individuals can be easily added to a group, no method for revocation is provided. Additionally, this scheme allows trusted third parties to de-anonymize revoked individuals [20].

9.2 Anonymous Authentication in Dynamic Groups

Schechter, Parnell and Hartemink proposed a series of anonymous authentication protocols for dynamic groups [12], which allow a user to authenticate themselves as a member of a group without giving away their actual identity. A user Alice holds a single identity key-pair, and an authentication key pair per dynamic group she is a member of. To prove herself to the authenticator Bob, Alice engages in a challenge-response procedure similar to our own protocol, where viewing the exchanges between Alice and Bob does not give away Alice's identity, and Alice may prove to herself that her authentication will be anonymous and unlinkable.

The time and space needed for this protocol is $O(n)$, where n is the number of group members (intuitively, since there are 2^n possible subsets that could be left after some number of revocations, Bob must send an $O(n)$-bit challenge). In comparison our protocol is $O(r)$, where r is the maximum number of revocations the system can have before it must be reset. The parameter r is chosen to be optimal for each individual case, leaving our system more easily scalable. Schechter, Parnell and Hartemink propose authenticating users in subsets of the group to increase scalability. However, this leaves individuals anonymous and unlinkable only among the members of their subset. While our system has a bound for the number of revocations before reset, the extension described in Section 8.2 allows us to extend the time before such reset is necessary.

9.3 Ring Signatures

Most current ring signature schemes for anonymous authentication prove impractical. When the number of members n increases, the length of the ring signatures increases linearly, leading to low efficiency. The issue of how to support dynamic groups is difficult to address. Liu and Tian propose a scheme which fixes the ring length with a one-way accumulator [6]. However, this scheme does not address how the revocation and addition of individuals to a group would be handled. While an administrator is not necessary in ring signatures due to the fact that we are using accessible public keys, an individual must know which public keys to sign with for anonymous authentication. In a on-line system where every member is updated post addition or revocation, $O(n)$ messages would be required.

10 Conclusion

We introduce Scalable Anonymous Authentication, a novel scheme that achieves a compromise between the speed of authenticating with little network traffic and the flexibility of authenticating arbitrary groups while preserving near-perfect anonymity. Furthermore, this trade off is parametrized by the revocation number r, which may be set by the system administrator at set-up time. Larger values of r allow for more revocations but also increase the time it takes to authenticate a user.

In today's world, there is a growing realization that accounting or "meta-data" information such as access logs pose a significant and widespread threat to privacy. Whether these problems will be solved by technology, policy, or some other combination of factors, we believe that exploring the limits of technology is essential to society's ability to discuss and address these issues. Scalable Anonymous Authentication demonstrates that even in a community as large as MIT's, anonymous authentication is a practical possibility.

References

[1] P. Agarwal, N. Bhargava, C. Chandrasekhar, A. Dahya, J.D. Zamfirescu *The MIT Card System: Analysis and Recommendations*, 2004. `http://groups.csail.mit.edu/mac/classes/6.805/student-papers/fall04-papers/mit_id/`

[2] J. Mandel, A. Roach, K. Winstein, *MIT Proximity Card Vulnerabilities*. `https://josephhall.org/tmp/mit_prox_vulns.pdf`

[3] L. Lebon, *Card Entries to Campus Buildings, Labs Tracked*, 2003. `http://tech.mit.edu/V122/N66/66card.66n.html`

[4] M. Szczys, *Passive RFID Tag Cloning*, 2011. `http://hackaday.com/2011/09/30/passive-rfid-tag-cloning/`

[5] A. Lysyanskaya, *Anonymity: an Overview*, 2011. `http://csrc.nist.gov/groups/ST/PEC2011/presentations2011/lysyanskaya.pdf`

[6] D. Liu, H. Tian, A New Anonymous Authentication Method Based On One-way Accumulator 2010. `http://www.aicit.org/jcit/ppl/02.%20JCIT5-454016.pdf`

[7] A. Fiat, M. Naor, *Broadcast Encryption*, 1993. `http://www.wisdom.weizmann.ac.il/~naor/PAPERS/broad_abs.html`

[8] L. Nguyen, R. Safavi-Naini, *Dynamic k-Times Anonymous Authentication* 2005. `https://eprint.iacr.org/2005/168.pdf`

[9] D. Nosowitz, *Everything You Need to Know About NFC*, 2011. `http://www.popsci.com/gadgets/article/2011-02/near-field-communication-helping-your-smartphone.-replace-your-wallet-2010`

[10] R. Rivest, A. Shamir, Y. Tauman, *How to Leak a Secret* 2001. `http://people.csail.mit.edu/rivest/pubs/RST01.pdf`

[11] BasicCard Smart Card Operating System `http://www.basiccard.com/`

[12] S. Schechter, T. Parnell, A. Hartemink, *Anonymous Authentication of Membership in Dynamic Groups*, 1999. `http://www.psrg.csail.mit.edu/pubs/fc99lncs.pdf`

[13] J. Hajny, L. Malina, *Anonymous Credentials with Practical Revocation*, `http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6400081&isnumber=6400058`

[14] P. Tsang, M.H. Au, A. Kapadia, S. Smith, *PEREA: Towards Practical TTP-Free Revocation in Anonymous Authentication*, 2008.
http://dl.acm.org/citation.cfm?id=1455813

[15] J. Furukawa, N. Attrapadung, *Fully Collusion Resistant Black-Box Traitor Revocable Broadcast Encryption with Short Private Keys*, 2007.
http://link.springer.com/chapter/10.1007%2F978-3-540-73420-8_44

[16] M. Ak, S. Pehlivanoglu, A.A. Selcuk, *Anonymous Trace and Revoke* 2014.
http://www.sciencedirect.com/science/article/pii/S0377042713005633

[17] J. Kim, S. Choi, K. Kim, C. Boyd, *Anonymous Authentication Protocol for Dynamic Groups with Power-Limited Devices*, 2003.
http://caislab.kaist.ac.kr/publication/paper_files/2003/SCIS2003/6B-2%28Jeongsung%20Kim%29.pdf

[18] OneName, *A system for sharing secrets using Shamir's Secret Sharing Scheme*, 7d3b1e4f4f.
https://github.com/onenameio/secret-sharing

[19] S. Drimer, S. Murdoch, *Keep Your Enemies Close: Distance Bounding Against Smartcard Relay Attacks*, 2007.
http://www.cl.cam.ac.uk/research/security/banking/relay/bounding.pdf

[20] J. Kilian, E. Petrank. *Identity Escrow*, 1998.
http://link.springer.com/chapter/10.1007%2FBFb0055727