Anonymous Authentication with Optional Shared Anonymity Revocation and Linkability

Martin Schaffer, Peter Schartner

University of Klagenfurt, Austria
Computer Science · System Security
{m.schaffer, p.schartner}@syssec.at

Abstract. In this paper we propose three smartcard-based variants of anonymous authentication using unique one-time pseudonyms. The first variant can be used to authenticate a user. However, his identity cannot be revealed and linked to other pseudonyms unless solving the computational Diffie-Hellman problem. In the second variant a set R of revocation centers is able to revoke the anonymity in collaboration with a trust center T but they are not able to link the revealed identity to other pseudonyms of the same user. Using the third variant additionally provides linkability if R and T cooperate. Some selected applications for the proposed protocols include physical access control, secure auctions, eCoins and online gambling.

1 Introduction

Nowadays smartcards appear to be a building block in several applications. Once mainly used for physical access control, their usage has been extended to more general applications related to different areas like eCommerce in the recent years. When using a smartcard, a user normally authenticates to the smartcard by entering a personal identification number. Then the smartcard itself authenticates to an instance (e.g. device (un)locking a door or service provider). Several standard methods exist, how to perform a unilateral authentication process, most of which do not really provide the anonymity of the user. So a lot of research has taken place to provide anonymous authentication based on zero-knowledge proofs. Such protocols have two advantages. First, the anonymity can be provided and second, collected communication data of several protocol runs of the same smartcard – depending on the particular solution – are not linkable by an eavesdropper. However, several standard proofs of identity require the same public input on the verifier's side during every authentication process (e.g. proof of knowledge of a private key, where the verifier must have access to the public key). Thus, the usage of the smartcard is traceable.

Providing authentication processes with anonymity and unlinkability protects the user's privacy. However, the verifier of the authentication process has to be protected as well, namely against malicious behaviour of the smartcard-holder in the protocols run thereafter. So we also need a mechanism to revoke

the user's anonymity and – if required – the ability to make user's activities traceable by disclosing linking information.

Over the last years several solutions have been proposed in this area. Many of them are based on group signatures, which allow users to prove the membership of a group without revealing their identity [1, 2, 5, 7]. Others are based on threshold privacy where a user remains anonymous when accessing a service up to a limited number of times [18, 26]. Revocation of anonymity and (un)linkability are a main requirement in anonymous credential systems [4, 19] or electronic money [13, 15, 17]. A solution optimized for power-limited devices has been proposed in [14]. Our scheme is neither based on group signatures nor on threshold privacy (as described in [23]). Compared to more general solutions such as traceable signatures [16] our approach is more specific – namely – optimized for smartcards. We designed the protocols in a simple way based on already known techniques providing anonymous authentication and mechanisms to revoke the anonymity and linkability of a user. A second reason for using smartcards is the fact that we use a particular technique to generate globally unique pseudorandom numbers which requires the use of smartcards [21, 22].

When considering authentication schemes based on – but not limited to – smartcards, we come to the following requirements:

- *Unforgeability.* The user must not be able to forge the authentication process.
- *Anonymity.* The anonymity of the user (identifier) has to be provided during every protocol run.
- *Unlinkability.* Any two authentication processes (protocol transcripts) must not be linkable.
- *Optional Anonymity Revocation and Linkability.* Given the protocol transcript, the anonymity of the user should be revocable by some additional information (trapdoor). Moreover, disclosure of linking information should lead to the identification of all corresponding authentication processes.

In the upcoming sections we present several variants of a smartcard-based anonymous authentication, based on unique one-time pseudonyms (OTPs). Depending on the used variant of the protocol, the anonymity is revocable by a set R of revocation centers so that its owner can be identified by the trust center T. If required, the protocol can be extended, so that the revealed identity can be linked to all its corresponding OTPs. The paper provides three variants of the anonymous authentication protocol AnonAuth:

1. AnonAuth$_1$: No Anonymity Revocation / No Linkability.
2. AnonAuth$_2$: Optional Anonymity Revocation / No Linkability.
3. AnonAuth$_3$: Optional Anonymity Revocation / Optional Linkability.

The proposed authentication protocols are based on OTPs containing a user-generated globally unique identifier id, blinded by a pseudorandomly chosen value b. These one-time pseudonyms are generated and signed by T in a tamper resistant device (TRD) so that there exists no linking information to the user

data accessible by T. The output of the TRD is an encrypted batch of authentication data containing the used blinding values and the signatures proving that the OTPs have been generated by T. The batch can only be decrypted by the owner of the corresponding unique identifier.[1]

Knowing only a OTP and the corresponding signature, does not reveal any information about the holder of the pseudonym. Additionally, OTPs of the same holder are mutually unlinkable. Hence, only the owner of a pseudonym is able to prove its ownership using a zero-knowledge proof which does not reveal private information.

1.1 Three Authentication Protocols

The proposed protocols are generally done in three steps:

1. The user imports authentication data to his smartcard and decrypts it.
2. Then he sends the one-time pseudonym and the corresponding signature to the verifier, who verifies the validity.
3. The user proves in zero-knowledge that he knows the pre-image(s) of the one-time pseudonym: the unique identifier id and/or the blinding value b.

We provide the following three authentication protocols:

AnonAuth$_1$ – *No Anonymity Revocation / No Linkability.* Here the user proves in zero-knowledge that he knows id and b without revealing information. However, no one is able to revoke the anonymity or link OTPs to the user except himself by publishing private information.

AnonAuth$_2$ – *Optional Anonymity Revocation / No Linkability.* Here the user attaches the blinding value b encrypted by the public key of R to the second step of the authentication protocol. In the third step he proves in zero-knowledge that the correct b is contained in the ciphertext. Thus, the user's anonymity can only be revoked by a set of revocation centers by using threshold decryption which acts as a partial trapdoor to the OTP-generation process. However, the revoked information can only be used to identify the owner of a specific pseudonym but cannot be used to find other pseudonyms of this user.

AnonAuth$_3$ – *Optional Anonymity Revocation / Optional Linkability.* Here the user additionally attaches encrypted linking information. In the third step he proves in zero-knowledge that the correct linking information is contained in the ciphertext. Shared decryption of b and the linking information acts as full trapdoor to the OTP-generation process. The disclosure of the pre-images of the used one-time pseudonym enables the trust center to identify all one-time pseudonyms that belong to the revealed unique identifier.

[1] The TRD might be replaced by a solution based on multi-party computation [12].

1.2 Core Components

User U_i. The user owns a smartcard containing the unique Integrated Chip-Card Serial Number ($ICCSN_i$). During the setup and registration phase his smartcard is provided with a unique user identifier and several keys. Encrypted authentication data is stored on U_i's local machine and can only be decrypted by U_i's smartcard. The user's part of the authentication process is done exclusively on his smartcard.

Trust Center T. The trust center owns the commitment of the user's identifier linked to the user's passport data, user's public key and signature. Moreover, the trust center owns a TRD which has two tasks:

1. Signing the user's data during the registration process.
2. Generating user's encrypted authentication data.

Bulletin Board BB. Encrypted authentication data is posted here, so that the user is able to download it if required.

Revocation Centers $R_1, \ldots, R_n$. In the second variant of the authentication protocol, a set of revocation centers is able to decrypt the blinding information, which leads to the anonymity revocation at T. In the third variant, they are able to decrypt linking information as well so that all OTPs of the revealed unique identifier can be found.

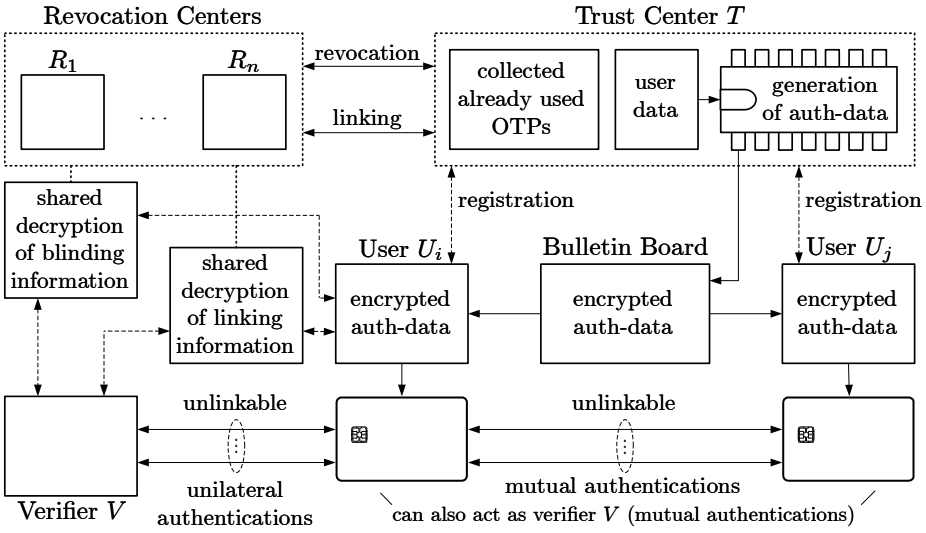

Fig. 1. System Architecture.

1.3 Selected Applications

eCoins. The proposed system can be used for "double spending detection" of eCoins. Therefore, the authentication data may contain information about the value of the eCoin and the user would have to pay for each authentication data according to its value. If he uses an eCoin he simply runs the proposed authentication process. Later on, the receiver of the eCoin sends the corresponding pseudonym to a double spending detection server which logs the used eCoins. If an eCoin has been sent twice, it is obvious that it has been used twice. In this case R and T can identify the cheating party.

Secure Auctions. Here, the participants can remain anonymous until one wins the auction. In this case the winner may have an interest to reveal his identity. If he refuses to pay, the auction chair can reveal his identity with the help of R and T.

Patent Search. The proposed scheme can be used for research activities in patent databases. Thus, a business rival is not able to link e.g. queries and hence is not able to associate them to a common identifier.

Physical Access Control. The standard application according to smartcards is physical access control. Using our scheme, the holder of a smartcard is not traceable anymore within buildings. If he (physically) misbehaves, his anonymity can be revoked. Moreover, his path through a building can be traced then as well.

Authenticity of Casino-Chips. Assume that every chip is provided with a contactless smart device. For instance, when a player places a chip in a roulette session, it automatically authenticates to the gambling-table. This makes the usage of forged chips detectable. Depending on the used authentication mechanism, the chip can be made traceable or not.[2]

Traceability of Gamblers. Assume that every gambler is provided with a Personal Digital Assistant that is used for online gambling in a casino. When playing (e.g. roulette), a person authenticates himself using the proposed protocols. If he loses a game he has to pay or his identity will be revealed for this particular game. Additionally all the games in which he participated can be linked to him if required. The advantage is that the behaviour of the player is untraceable as long as the linking information has not been decrypted by R.

2 Preliminaries & Notation

2.1 The Discrete Logarithm Problem Family

The unlinkability and security of our system relies on the security of the discrete logarithm problem (DLP), the computational Diffie-Hellman problem (CDP) as

[2] Note: Unlike the system proposed in [6] our scheme aims at *physical* casino-chips containing cryptographic hardware.

well as the decisional Diffie-Hellman problem (DDP). Let g be the generator of a cyclic group $\mathbb{Z}_q^*$, then it is hard to compute x by only knowing g^x (DLP). Moreover, it is hard to compute $g^{x \cdot y}$ by only knowing g^x and g^y (CDP). Given the values g^x, g^y and Z it his hard to decide whether $Z = g^{x \cdot y}$ or Z has been chosen at random (DDP). A triple $(g^x, g^y, g^{x \cdot y})$ is called Diffie-Hellman triple. Several variations of the Diffie-Hellman problem can be found in [3].

2.2 ElGamal's Cryptosystem and Signature Scheme

Let h be the generator of a cyclic group $\mathbb{Z}_q^*$. Then the ElGamal key generation outputs the encryption key $e = h^d$ and the decryption key d. The encryption/decryption is done as follows [10]:

$$E(m, a, e) = (C_1, C_2), \quad C_1 = h^a, \quad C_2 = m \cdot e^a, \quad a \in_R \mathbb{Z}_q^*$$
$$D((C_1, C_2), d) = m, \quad m = C_2 \cdot (C_1^d)^{-1}$$

We abstract the encryption of larger plaintext by $E'(m, e) = C$. The signature generation/verification is performed over sign key s/verification key v:

$$S(m, s) = \sigma, \quad V(m, \sigma, v) \in \{\text{true}, \text{false}\}$$

Note, that we defined S and V as blackbox-functions because they can be replaced by any other signature scheme.

2.3 ElGamal Threshold Decryption

If we consider a single party not to be trustworthy enough to perform a decryption only on request, then there is a need to share the decryption function over a set of instances. In [9] Desmedt and Frankel proposed a shared variation of ElGamal's decryption function. Therefore, the private key d has to be generated in a distributed way by using e.g. the protocol in [11] providing each decryptor P_i with a share d_i. In the following we consider the shared decryption protocol as a blackbox-function:

$$\widetilde{D}((C_1, C_2), (d_1, \ldots, d_n)) = m$$

2.4 Locally Generated Globally Unique Pseudorandom Numbers

In [22] a method to locally generate globally unique pseudorandom numbers has been proposed. Therefore, a smartcard, a unique identifier and a symmetric cryptosystem are needed. In the current paper we use this method to generate the unique user identifier and the blinding values. A globally unique pseudorandom number UN can be generated in the user's smartcard as follows [22]:

$$UN = E_{DES}(ICCSN||Pad, \tau_k)||\tau_k$$

where Pad is a random padding up to the input-size. Here, τ_k is a randomly chosen DES-key and E_{DES} is the DES encryption function. Due to the fact that UN is never accessible by unauthorized instances (we only use its discrete logarithm (DL) commitment g^{UN}), it is computationally hard to reveal it. Thus, the security of DES does not play a role, because the ciphertext is never available to an attacker. A similar approach which is based on the RSA cryptosystem [20] can be found in [21]. There UN can be uniquely generated as follows:

$$UN = E_{RSA}(ICCSN||Pad, \tau_e)||\tau_e||\tau_n$$

where (τ_e, τ_n) is a randomly chosen RSA public key. We use the RSA-version for the generation of unique ElGamal keys (UKG). For a proof of uniqueness we refer to [21] and [22] respectively.

2.5 Unique One-time Pseudonyms

In this paper we use OTPs of the form $\eta_j = (g^{b_j}, g^{b_j \cdot id_i})$. We require each pseudorandom value b_j to be uniquely generated in the TRD. Moreover, we require the unique user identifier to be locally generated by the user himself (in his smartcard). For both values we use the unique pseudorandom number generation (URNG) based on symmetric encryption as described in section 2.4. To avoid local doublets when generating b_j, the TRD has to include a counter to the generation process. Due to the fact, that b_j is unique g^{b_j} is unique as well. The second part of η_j commits id_i to the pseudonym, so that all pseudonyms of the same holder can be linked to his unique identifier if required.

2.6 Used Zero-knowledge Proofs

We use a very efficient abstract notation for proofs of knowledge (PK) introduced in [5]. For detailed information on the following proofs we refer to [24] and [25].

Schnorr's Proof of Knowledge. This proof is required by the first authentication protocol, where a one-time pseudonym can neither be opened nor linked without the cooperation of the user. Let $X = g^x$ be a public value in $\mathbb{Z}_q^*$ with secret pre-image x. Then the prover can convince the verifier in zero-knowledge that he knows x using Schnorr's proof of knowledge [24]. Using the abstract notation Schnorr's PK looks as follows:

$$PK\{(\alpha) : X = g^{\alpha}\}$$

Mapping: $\alpha = x$

Stadler's Proof of Knowledge. Let $X = g^x$ and $(C_1, C_2) = (h^a, x^{-1} \cdot e^a)$ an ElGamal ciphertext. In [25] Stadler proposed a PK where one can prove, that (C_1, C_2) is a correct ElGamal ciphertext and contains the inverse of x. This can

only be done by the prover iff he knows a and x. In our scheme this proof can be used to prove that the pre-images of a OTP are contained in an attached ElGamal ciphertext. Using the abstract notation Stadler's PK looks as follows:

$$PK\{(\alpha, \beta, \gamma) : X = g^{\alpha} \ \wedge \ (C_1, C_2) = E(\gamma, \beta, e)\}.$$

Mapping: $\alpha = x \quad \beta = a \quad \gamma = x^{-1}$

Concurrent Executions. By using the techniques described by Damgard in [8] the above protocols can be made concurrent zero-knowledge. This means, that even if they are executed in parallel, they remain zero-knowledge. Such a modification is of extreme importance for our scheme, because we use smartcards on the user's side. Hence, we have to keep the number of sent messages as minimal as possible.

3 On the Linkability of the used One-time Pseudonyms

In the following we consider several variations of how to identify the holder of a pseudonym. Moreover, we discuss the ability of T to link pseudonyms to a user U_i. For our consideration we assume that all generated pseudonyms are available to T without linkage to the corresponding unique identifier.

	amount of open information		
unique identifier	nothing	b_j	b_j and id_i
id_i	1. linkable by T	2. linkable by T	3. linkable by T
g^{id_i}	4. unlinkable (CDP)	5. unlinkable (CDP)	6. linkable by T
	anonymity not revocable	anonymity revocable	

Table 1. Linkability of User U_i to his OTPs

Table 1 shows the possible unique identifier with its linking-property based on the amount of open information resulting in the following 6 variations:

1. For every $\eta_j = (\eta_{j1}, \eta_{j2})$ T verifies if $\eta_{j1}^{id_i} = \eta_{j2}$ holds. Each successful verification links the pseudonym to U_i.
2. Opening $\eta_j = (g^{b_j}, g^{b_j \cdot id_i})$ results in g^{id_i}. For each id_i' T has to verify if $g^{id_i'} = g^{id_i}$ holds. If one holds the owner of the pseudonym has been found. The linkability does not depend on the anonymity revocation.
3. Opening $\eta_j = (g^{b_j}, g^{b_j \cdot id_i})$ results in id_i which speeds up the identification of a user because T does not have to perform the verifications described in 2. Again, the linkability does not depend on the revocation.
4. For each $\eta_j = (\eta_{j1}, \eta_{j2})$ T would have to verify if $\eta_{j1}^{id_i} = \eta_{j2}$ holds. To perform such verifications T has to solve the CDP because he only knows g^{id_i}.

5. Opening $\eta_j = (g^{b_j}, g^{b_j \cdot id_i})$ results in g^{id_i}. Thus, the owner U_i can be identified by T but no open information of his other pseudonyms is revealed.
6. Opening $\eta_j = (g^{b_j}, g^{b_j \cdot id_i})$ results in id_i. For each $g^{id'_i}$ T has to verify if $g^{id'_i} = g^{id_i}$ holds. If one holds the owner of the pseudonym has been found. Moreover, all pseudonyms of U_i can be revealed as described in 1.

4 The Authentication Scheme

4.1 Setup

First the system parameters have to be generated in a secure environment. A suitable cyclic group $\mathbb{Z}_q^*$, $q \in \mathbb{P}$ and the according generators h (for ElGamal) and g (for OTPs) have to be chosen. The value n denotes the number of revocation centers and t the threshold of tolerated dishonest revocation centers. The parameter l specifies the number of OTPs included in a batch of authentication data generated in the TRD. The security parameter k specifies the number of necessary rounds of the used zero-knowledge proof. We now assume that each instance of the system is provided with all necessary system parameters.

The user U_i generates a globally unique identifier id_i and an ElGamal key-pair (e_i, d_i) where d_i is the private key:

$$\begin{aligned} id_i &= E_{DES}(ICCSN_i||Pad, \tau_{k_i})||\tau_{k_i} \\ d_i &= E_{RSA}(id_i||Pad, \tau_{e_i})||\tau_{e_i}||\tau_{n_i}, \quad e_i = h^{d_i} \end{aligned}$$

such that $id_i, d_i \in \mathbb{Z}_q^*$. The TRD generates a globally unique sign key s_t:

$$s_t = E_{RSA}(TRDID||Pad, \tau_{e_t})||\tau_{e_t}||\tau_{n_t}, \quad v_t = h^{s_t}$$

such that $s_t \in \mathbb{Z}_q^*$. The verification key v_t is exported to T. The set of revocation centers generate a decryption key d_r in a shared way (e.g. with the solutions in [11]) without reconstructing it, resulting in the private-key-shares $d_{r_1}, \ldots, d_{r_n}$ and the corresponding (reconstructed) public key e_r.

4.2 User Registration

First of all U_i computes g^{id_i} and sends the pair (g^{id_i}, e_i) to T – more precise to the TRD – during a face-to-face authentication. The TRD signs (g^{id_i}, e_i) with the sign key s_t resulting in the signature $\sigma(id_i)$. Then T stores the data of unique identification $UI_i = (\text{passport data}, g^{id_i}, e_i, \sigma(id_i))$ of U_i to the database and returns (v_t, e_r) to U_i's smartcard.

4.3 Establishing a Batch of Authentication Data

Prior to generating authentication data, the TRD has to verify if (g^{id_i}, e_i) has been signed with s_t during the registration process. Therefore, it verifies if $\sigma(id_i)$ is the corresponding signature. Iff the verification succeeds, TRD's task is to

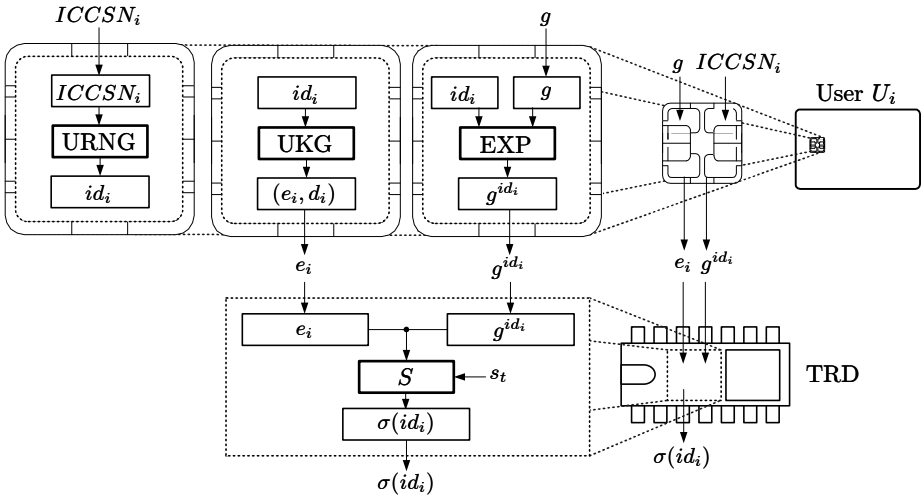

Fig. 2. User Registration – Computations on the Smartcard and the TRD respectively.

perform the function GAD (Generate Authentication Data) for g^{id_i} without revealing information about the internally chosen pseudorandom blinding values $b_1, \ldots, b_l$ and the corresponding signatures $\sigma_1, \ldots, \sigma_l$:

$$GAD(g^{id_i}, e_i, s_t, g, l) = (\lambda_1, \ldots, \lambda_l) := \Lambda(id_i)$$
$$\forall_{1 \le j \le l}: \quad b_j = E_{DES}(TRDID||Cnt, \tau_{k_t})||\tau_{k_t}, \quad \eta_j = (g^{b_j}, (g^{id_i})^{b_j}),$$
$$\sigma_j = S(\eta_j, s_t), \quad \lambda_j = E'(b_j||\sigma_j, e_i)$$

where $TRDID$ is the unique identifier of the TRD and Cnt a counter to gain uniqueness. The batch $\Lambda(id_i)$ is posted on BB indexed by g^{id_i} or e_i. Now user U_i is able to access $\Lambda(id_i)$ and store it to his local machine. Depending on the capacity of the smartcard, U_i can import the whole batch or only a subset.

4.4 Unilateral Anonymous Authentication

Assume that the user U_i wants to authenticate himself to a verifier V using λ_j which corresponds to $\eta_j = (g^{b_j}, g^{b_j \cdot id_i})$.

No Anonymity Revocation / No Linkability

Protocol 1 (AnonAuth$_1$). User U_i holds id_i and v_t in his smartcard and λ_j on his local machine. The verifier V holds v_t.

1. U_i imports λ_j to his smartcard and decrypts it resulting in $b_j||\sigma_j$.
2. (a) U_i computes $\eta_j = (g^{b_j}, g^{b_j \cdot id_i})$ and verifies its correspondence to σ_j.
   (b) The tuple (η_j, σ_j) is sent to V.

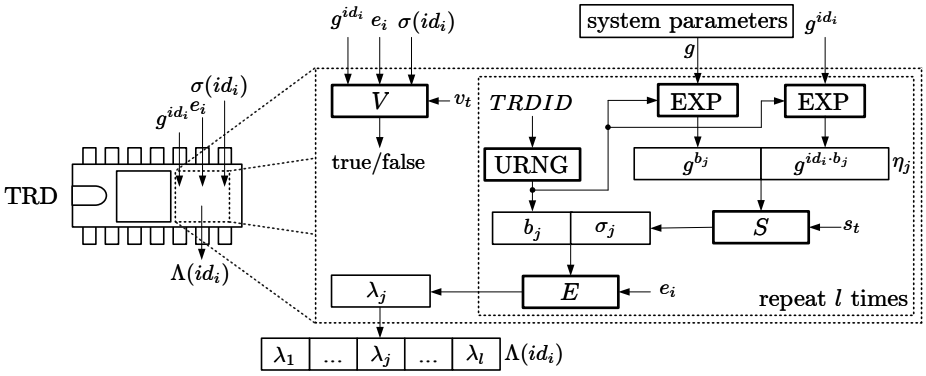

Fig. 3. Establishment of Authentication Data – Computations in the TRD.

(c) V verifies if σ_j is the signature to η_j.

3. U_i proves in zero-knowledge (concurrent executions of Schnorr's PK) that he knows the pre-images of η_j:

$$PK\{(\alpha, \beta) : \ \eta_{j1} = g^{\alpha} \ \wedge \ \eta_{j2} = g^{\beta}\}.$$

Mapping: $\alpha = b_j \quad \beta = b_j \cdot id_i$

Optional Anonymity Revocation / No Linkability

Protocol 2 (AnonAuth$_2$). User U_i holds id_i, e_r and v_t in his smartcard and λ_j on his local machine. The verifier V holds e_r and v_t respectively.

1. U_i imports λ_j to his smartcard and decrypts it resulting in $b_j||\sigma_j$.
2. (a) U_i computes $\eta_j = (g^{b_j}, g^{b_j \cdot id_i})$ and verifies its correspondence to σ_j.
   (b) U_i computes $r_j = E(b_j^{-1}, a, e_r)$.
   (c) The triple (η_j, σ_j, r_j) is sent to V.
   (d) V verifies if σ_j is the signature to η_j.
3. U_i proves in zero-knowledge (concurrent executions of Stadler's PK) that r_j contains the inverse of the pre-image of η_{j1}:

$$PK\{(\alpha, \beta, \gamma) : \ \eta_{j1} = g^{\alpha} \ \wedge \ r_j = E(\gamma, \beta, e_r)\}.$$

Mapping: $\alpha = b_j \quad \beta = a \quad \gamma = b_j^{-1}$

Optional Anonymity Revocation / Optional Linkability

Protocol 3 (AnonAuth$_3$). User U_i holds id_i, e_r and v_t in his smartcard and λ_j on his local machine. The verifier V holds e_r and v_t respectively.

1. U_i imports λ_j to his smartcard and decrypts it resulting in $b_j||\sigma_j$.

2. (a) U_i computes $\eta_j = (g^{b_j}, g^{b_j \cdot id_i})$ and verifies its correspondence to σ_j.
   (b) U_i computes $r_j = E(b_j^{-1}, a, e_r)$ and $l_j = E((b_j \cdot id_i)^{-1}, a', e_r)$.
   (c) The tuple $(\eta_j, \sigma_j, r_j, l_j)$ is sent to S.
   (d) V verifies if σ_j is the signature to η_j.
3. U_i proves in zero-knowledge (concurrent executions of Stadler's PK) that r_j contains the inverse of the pre-image of η_{j1} and l_j contains the inverse of the pre-image of η_{j2}:

$$PK\{(\alpha, \beta, \gamma, \delta, \varepsilon, \zeta) : \ \eta_{j1} = g^{\alpha} \ \wedge \ r_j = E(\gamma, \beta, e_r) \ \wedge \ \eta_{j2} = g^{\delta} \ \wedge \ l_j = E(\zeta, \varepsilon, e_r)\}.$$

Mapping: $\alpha = b_j$ $\quad \beta = a$ $\quad \gamma = b_j^{-1}$
$\delta = b_j \cdot id_i$ $\quad \varepsilon = a'$ $\quad \zeta = (b_j \cdot id_i)^{-1}$

The proposed protocols can also be used for mutual authentication as well. Therefore, the steps of the interactive proofs have to be teethed.

4.5 Shared Revocation

If the anonymity of an authentication process has to be revoked the verifier has to convince at least $t+1$ revocation centers and T to agree with the revocation process. If only the user identifier g^{id_i} has to be revealed the revocation centers need the used OTP η_j and the encrypted open information r_j. U_i's anonymity can be revoked as follows:

$$\eta_{j2}^{\widetilde{D}(r_j, (d_{r_1}, \ldots, d_{r_n}))} = \eta_{j2}^{b_j^{-1}} = g^{b_j \cdot id_i \cdot b_j^{-1}} = g^{id_i}$$

If it is additionally required that all pseudonyms belonging to U_i need to be found, the revocation centers need l_j as well. Then they are able to compute id_i as follows:

$$\widetilde{D}(l_j, (d_{r_1}, \ldots, d_{r_n}))^{-1} \cdot \widetilde{D}(r_j, (d_{r_1}, \ldots, d_{r_n})) = b_j \cdot id_i \cdot b_j^{-1} = id_i$$

Once the anonymity has been revoked including linkability information, each used pseudonym of the user can be linked to g^{id_i}. If we do not require the user's future-used OTPs to be linkable, he has to locally generate a new user-id and re-register at T.

5 Efficiency and Pre-computation

For efficiency reasons the used zero-knowledge proofs have to be run with the modifications described in [8]. Thus, we achieve concurrent executions without loosing the zero-knowledge property. Protocol AnonAuth$_1$ uses Schnorr's PK which can be run in one round only computing one first-message. This is possible because the challenge space is $\mathbb{Z}_q^*$ in the concurrent model. However, the other two protocols use Stadler's PK whose challenge space is {0,1}. Thus, concurrent

executions require the computation of k first messages. This means in our case, that a smartcard has to perform $O(k)$ exponentiations in $\mathbb{Z}_q^*$ which – depending on the bit-length – can be time-consuming. If this appears to be a problem (which depends on the used type of smartcard) the proposed scheme can be extended so that the TRD pre-computes the k first-messages for each OTP which will then be contained in the encrypted authentication data.[3] Hence, the smartcard only has to compute k third-messages which can be done by negligible $O(k)$ multiplications in $\mathbb{Z}_q^*$. In any case we suggest using ElGamal based on the elliptic curve discrete logarithm problem to speed up all protocols.

6 Security Analysis

In this section we analyze the security of the proposed scheme. Therefore, we consider the security according to the requirements stated in section 1. First of all, we analyse the possible dishonest behaviour of the verifier and external adversaries to gain any information about the user's identity or the linking (prover's point of view). Then, we analyse how an external attacker would try to impersonate a registered user (verifier's point of view).

6.1 Prover's Point of View

Anonymity

User Registration. The user U_i generates his unique identifier id_i locally without interaction. He only sends g^{id_i} to T. So T is not able to extract id_i due to the discrete logarithm problem. The uniqueness of id_i has been proven in [22].

Establishment of Authentication Data. The batches of authentication data are generated by the TRD. The input of *GAD* has to be authentic – otherwise the TRD could be faked. Therefore, the user data (g^{id_i}, e_i) must have been signed by the TRD during the user registration. The output of the TRD is encrypted with e_i. An adversary would have to break the ElGamal cryptosystem to get information about the blinding values which would reveal g^{id_i}. The security mainly relies on the tamper resistant property of the used device and the CDP.

Protocol 1 (AnonAuth_1). In every protocol run a OTP and the corresponding signature is sent to the verifier. The verifier neither gains information about id_i out of the OTP (due to the CDP) nor out of Schnorr's PK (which is proven to be zero-knowledge if used correctly).

Protocol 2 (AnonAuth_2). Here the verifier additionally receives an ElGamal ciphertext containing the blinding value of the used OTP. To extract id_i, the verifier would have to break Stadler's PK (which is proven to be zero-knowledge). To receive g^{id_i}, he would have to break the ElGamal cryptosystem or compromise at least $t + 1$ revocation centers.

[3] Note: For efficiency reasons the authentication data should be encrypted using hybrid encryption (e.g. AES + ElGamal).

Protocol 3 (AnonAuth$_3$). Here the verifier additionally receives an ElGamal ciphertext containing the linking information of the used OTP. To extract id_i the verifier would either have to break the ElGamal cryptosystem or Stadler's PK or compromise $t+1$ revocation centers.

External Adversary. An external adversary would have to compromise U_i's smartcard, compromise trust center T and solve the DLP, break the ElGamal cryptosystem to retrieve g^{id_i} (or id_i) or compromise at least $t+1$ revocation centers.

Unlinkability

Adversary knows g^{id_i}. If the adversary knows g^{id_i} and has access to all OTPs of the system, he would have to solve the DDP that is for any OTP $\eta_j = (\eta_{j1}, \eta_{j2})$ to decide whether $(g^{id_i}, \eta_{j1}, \eta_{j2})$ forms a Diffie-Hellman triple or not. Due to the fact that the used proofs of knowledge are zero-knowledge, the adversary does not gain any information about id_i.

Adversary knows id_i. If the adversary knows id_i and has access to all OTPs, then he is able to find all pseudonyms linked to U_i (see section 3).

Optional Anonymity Revocation and Linkability

Protocol 1 (AnonAuth$_1$). In this protocol the user does not give encrypted open information to the verifier. Even if he behaves dishonest after a successful authentication process the verifier is never able to reveal the user's identity except he compromises him or solves the CDP.

Protocol 2 (AnonAuth$_2$). Here the user additionally sends encrypted open information to the verifier. In case of malicious behaviour the verifier has to convince at least $t+1$ revocation centers to decrypt the blinding value and reveal g^{id_i}. Knowing g^{id_i}, the trust center is able to identify the user via the linked passport data.

Protocol 3 (AnonAuth$_3$). Here the user sends the encrypted open- and linking information to the verifier. If required at least $t+1$ revocation centers are able to decrypt both. Knowing the resulting plaintext the revocation centers can compute id_i. So T can identify the user via the linked passport data. If all used OTPs of the system are available, U_i's pseudonyms can be found as well.

6.2 Verifier's Point of View

Forging OTPs. If an adversary knows g^{id_i} he would be easily able to forge OTPs of U_i, but then he would have to be able to forge the corresponding signature as well. Therefore, he would have to compromise the TRD or the used signature scheme.

Replay Attacks. If the communication process is not encrypted, an eavesdropper can make a copy of the used OTP and the corresponding signature. If he tries to use the stolen OTP in a different authentication process he would have to fake the used zero-knowledge proof.

For security considerations of the used PK we refer to [24] and [25].

7 Conclusion & Future Research

In this paper we proposed three protocols providing anonymous authentication. The first protocol allows a user to prove that he is registered. However, there is no chance to revoke the user's anonymity. Moreover, the authentication processes are mutually unlinkable. This protocol is very useful if the user himself has a strong interest in revealing his identity himself if required (e.g. secure auctions). The second protocol gives the verifier the possibility to revoke the user's identity together with a set of revocation centers and the trust center. Such a protocol can be used if the verifier has a strong interest in the user behaving honest in the protocols performed *after* the authentication process. The third protocol enables the verifier in collaboration with the revocation centers and the trust center to make a user traceable if he behaves dishonest.

We are currently optimizing the protocols with the following goals:

- Multi-party solution to replace the TRD by a set of standard PCs.
- A simple way to establish OTPs where the user only receives one root-OTP and a root-signature based on which he is able to derive several globally unique OTPs and the corresponding signatures.
- An improved version of Stadler's PK, that is more efficient concerning the number of messages for concurrent executions (larger challenge space).
- Some variations of the protocols optimized for selected applications.

8 Acknowledgements

The authors would like to thank Dieter Sommer for his useful comments.

References

1. G. Ateniese, et al. A practical and provably secure coalition-resistant group signature scheme. Adv. in Crypt.: CRYPTO 2000, LNCS 1880, pp. 255–270, Springer-Verlag, 2000.
2. M. Bellare, H. Shi, C. Zhang. Foundations of Group Signatures: The Case of Dynamic Groups. Cryptology ePrint Archive: Report 2004/077.
3. F. Bao, R.H. Deng, H. Zhu. Variations of Diffie-Hellman Problem. Proc. of ICICS'03, LNCS 2836, Springer Verlag, 2003.
4. J. Camenisch, A. Lysyanskaya. An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation. Adv. in Crypt.: EUROCRYPT'01, LNCS 2045, pages 93+, Springer Verlag, 2001.

5. J. Camenisch, A. Stadler. Efficient group signature schemes for large groups. Adv. in Crypt.: CRYPTO'97, LNCS 1296, pp. 410–424, Springer Verlag, 1997.
6. J. Castella-Roca et al. Digital chips for an on-line casino. Proc. of ITCC'05, IEEE Computer Society, vol. I, pp. 494–499, 2005.
7. D. Chaum, E. van Heyst. Group signatures. Adv. in Crypt.: EUROCRYPT'91, LNCS 547, pp.257–265, Springer-Verlag, 1991.
8. I. Damgard. Efficient Concurrent Zero-Knowledge in the Auxiliary String Model. Adv. in Crypt.: EUROCRYPT'00, LNCS 1807, pp. 418–430, Springer Verlag, 2000.
9. Y. Desmedt, Y. Frankel. Threshold Cryptosystems. Adv. in Crypt.: CRYPTO'89, LNCS 435, pp. 307–315, Springer-Verlag, 1990.
10. T. ElGamal. A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. Adv. in Crypt.: CRYPTO'84, LNCS 196, pp. 10–18, Springer-Verlag, 1985.
11. R. Gennaro et al. Secure Distributed Key Generation for Discrete-Log Based Cryptosystems. Adv. in Crypt.: EUROCRYPT'99, LNCS 1592, pp. 295–310, Springer-Verlag, 1999.
12. O. Goldreich et al. How to play any mental game – a completeness theorem for protocols with honest majority. Proc. 19th ACM STOC, pp. 218–229, 1987.
13. M. Jakobsson, M. Yung. Revokable and Versatile Electronic Money. In Proc. of the 3rd CCCS, pages 76–87, ACM press, 1996.
14. J. Kim, et al. Anonymous Authentication Protocol for Dynamic Groups with Power-Limited Devices. Proc. of SCIS2003, vol 1/2, pp 405–410, 2003.
15. H. Kim, et al. Design and Implementation of Revocable Electronic Cash System based on Elliptic Curve Discrete Logarithm Problem. Proc. of WISA'02, pp. 85–102, Korea, 2000.
16. A. Kiayias, Y. Tsiounis, M. Yung. Traceable signatures. Adv. in Crypt.: EUROCRYPT'04, LNCS 3027, pp. 571–589, Springer-Verlag, 2004.
17. T. Nakanishi, M. Shiota, Y. Sugiyama. An Unlinkable Divisible Electronic Cash with User's Less Computations Using Active Trustees. In Proc. ISITA2002, pp. 547–550, Xi'an, 2002.
18. L. Nguyen, R. Safavi-Naini. Dynamic k-Times Anonymous Authentication. Proc. of ACNS'05, LNCS 3531, pp. 318–333, Springer-Verlag, 2005.
19. A. Pashalidis, C.J. Mitchell. A Security Model for Anonymous Credential Systems. IFIP Conf. Proc. 148, pp. 183–189, Kluwer Academic Publishers, Boston, 2004.
20. R. Rivest, A. Shamir, L. Adelman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM, 21 (1978), pp. 120–126.
21. P. Scharnter, M. Schaffer. Unique User-generated Digital Pseudonyms. Proc. of MMM-ACNS'05, LNCS 3685, pp. 194–206, Springer-Verlag, 2005.
22. P. Schartner. Security Tokens – Basics, Applications, Management, and Infrastructures. IT-Verlag (2001).
23. K. Sako, S. Yonezawa, I. Teranishi. Anonymous Authentication: For Privacy and Security. NEC Journal of Advanced Technology, Vol. 2, No. 1, p. 79–83, 2005.
24. C.P. Schnorr. Efficient Signature Generation for Smart Cards. Adv. in Crpyt.: EUROCRYPT'88, LNCS 330, pp. 239–252, Springer Verlag, 1990.
25. A. Stadler. Publicly Verifiable Secret Sharing. Adv. in Crypt.: Eurocrypt'96, LNCS 1070, pp. 190–199, Springer-Verlag, 1996.
26. L. Teranisi, J. Furukawa, K. Sako. k-Times Anonymous Authentication. Adv. in Crypt.: ASIACRYPT'04, LNCS 3329, pp. 308–322, Springer-Verlag, 2004.