PRACTICAL TRACEABLE ANONYMOUS IDENTIFICATION

Daniel Slamanig[1], Peter Schartner[2] and Christian Stingl[1]
[1]*Department of Medical Information Technology, Healthcare IT & Information Security Group*
Carinthia University of Applied Sciences, 9020 Klagenfurt, Austria
[2]*Institute of Applied Informatics, System Security Group, Klagenfurt University, 9020 Klagenfurt, Austria*

Keywords: Anonymity, Anonymous identification, Authentication, Privacy protection, Public key cryptography, Smart cards.

Abstract: Internet privacy is of increasing interest, since online services are getting more and more ubiquitous and cover many aspects of one's daily life. Hence users leave information tracks and disclose information during usage of services which can be compiled by third parties to infer users behavior, preferences etc. and thus may violate user's privacy. In this paper we propose a practical method for traceable anonymous identification which can be used for online services in order to protect user's privacy. It enables users to authenticate themselves to a service provider, whereas the service provider is not able to identify authenticating users. However, the service provider can be sure that only authorized users are able to authenticate. Since absolute anonymity may open the door for dishonest behavior, our protocol incorporates traceability, which enables a service provider to identify authenticating users in cooperation with an offline trusted third party. The proposed method is fully compatible with real world scenarios, i.e. public key infrastructures based on X.509 certificates, and can be easily deployed using state of the art smart cards. Furthermore, the proposed method is very efficient and we give a performance analysis as well as a security analysis of the introduced protocols.

1 INTRODUCTION

Internet based services are increasing in popularity and cover many aspects of one's daily life, e.g. banking, shopping, online subscriptions, social networking, e-government and increasingly also health related services. It is indisputable, that these services provide a convenient way for everyday's activities, however, they also disclose a lot of information about user's preferences, behavior, etc. and thus may violate their privacy. In this context we are faced with a phenomenon denoted as privacy myopia (Froomkin, 2000), which means that people often are not aware of dangers related to privacy and sell or give away their data without reflecting on potential negative consequences. There is a vast body of research on anonymous communication techniques (Danezis and Diaz, 2008) which aims at providing anonymity for Internet users by means of "hiding" their network addresses, i.e. IP-addresses. However, many services require user-identification at higher layers, i.e. the service level. In addition to communication anonymity it may also be desired to provide anonymity in context of authentication, since adversaries which are often less considered are insiders at providers which host the aforementioned services, are able to access service level information and build dossiers of service users. However, in context of authentication, anonymous communication as the only measure to provide anonymity is necessary, but not sufficient. If users authenticate themselves to services, this allows insiders to link all actions conducted within a service usage to this user. Thereby, it is desirable to achieve a unique identification of a user by means of authentication, since the provider of a service wants to limit access to authorized users, access rights may be given individually to users and resources may be related to specific users. But the unique identification of users also eases to track user's behavior and consequently may violate their privacy. Hence, to protect user's privacy it is necessary to give user's the ability to anonymously authenticate to a service and at the same time give the service provider the ability to restrict access to authorized users.

1.1 Contribution of this Paper

In this paper we will introduce a practical scheme for anonymous identification, denoted as traceable ring authentication, which enables authorized users to au-

Slamanig D., Stingl C. and Schartner P. (2009).
PRACTICAL TRACEABLE ANONYMOUS IDENTIFICATION.
In *Proceedings of the International Conference on Security and Cryptography*, pages 225-232
DOI: 10.5220/0002217502250232

thenticate at a service provider, whereas this service provider is not able to identify the user. However, he can be sure that solely authorized users will pass an authentication. Our approach is comparable to, but more efficient than, deniable ring authentication (Naor, 2002) and verifiably common secret encoding (Schechter et al., 1999), which can be seamlessly integrated into existing public key infrastructures (PKIs). Furthermore, it can be seen as an improved version of (Lindell, 2007) with reduced and optimal round complexity. Additionally, it provides traceability using tamper resistant devices like smart cards, which enables a service provider to identify authenticating users in case of misuse or fraud.

One particular application that we have in mind for the introduced protocol are personal health records (PHRs), e.g. Google Health or Microsoft Health Vault, which provide health institutions the possibility to integrate user's health information, e.g. medical documents, and user's the convenient possibility to manage and access their health information online. Especially in context of highly sensitive health data, user behavior, e.g. the frequency of interaction with the service, may reveal information that can affect the user's future life negatively. Think of a user who applies for a job and the recruiter knows that the frequency of interactions of the user with his say Google Health account is far above the average within the last year. This clearly does not indicate a "perfect" state of health.

1.2 Public Key Encryption Scheme

A public key encryption scheme is a triple of polynomial time algorithms (G,E,D), whereas $G(1^k)$ is a key generation algorithm which, given a security parameter k in unary, outputs a secret decryption key SK and a corresponding public encryption key PK. In order to encrypt a message m, the encryption algorithm E is given m, the public encryption key PK and some auxiliary random input ω. The algorithm outputs a ciphertext c and the encryption is denoted as $c = E_{PK}(m,\omega)$. The random input ω indicates that the encryption scheme is probabilistic and we assume that, unless stated otherwise, it provides semantic security, i.e. indistinguishability under chosen plaintext attacks (IND-CPA). The decryption algorithm is given the ciphertext c and the secret decryption key and outputs the message m which is denoted as $m = D_{SK}(c)$.

2 BASIC IDEA

We will now briefly sketch the idea of the proposed approach. As mentioned in section 1.1 the main goal is to provide users anonymous access to services, whereas the access must be limited to authorized users. One approach that is diametric to ours is private information retrieval (PIR) (Chor et al., 1995). In a PIR scheme a user queries data from a server, whereas the server does not learn anything about the queried data. Our approach targets at querying data from a server, whereas the server learns which data was queried, however has no clue who actually has queried the data. Therefore we assume that the data which is queried provides no identifying information on the owner or authorized users respectively, whereas we will not discuss the issue on how to realize this. For simplicity, in context of a PHR we may assume that user-centric encryption is used, whereas every document is encrypted by a party prior to providing this data to the service.

Anonymous identification means that a user proves to a service provider (SP) that he is a member of the set of authorized users without revealing his identity. Thus, from the point of view of SP every user is equally likely to be the one who is actually authenticating to the service. A trivial solution to this problem would be to give every authorized user the same secret key k, which could be used in conjunction with a standard challenge-response authentication protocol. However, this approach suffers from some serious drawbacks, i.e. a compromised key requires the reissuing of a new secret key k' and so does the revocation of a single user.

Our approach can be described as follows: The service provider encrypts a random challenge using the public keys of all authorized users and sends the resulting vector to the anonymous user. The user decrypts the respective element of the vector and checks whether the same challenge was encrypted for every authorized user. If this check holds, the anonymous user provides the challenge to the service provider. If both challenges match, the user must be an authorized user. This protocol also provides unlinkability, i.e. different executions of the protocol of the same user cannot be linked together. In order to be able to identify users in case of misuse or fraud, we employ a tamper resistant security token, e.g. a smart card, which encrypts the user's identity for an traceability authority (TA) and appends it to the responded challenge. Consequently, the SP can give a transcript to the TA, which is able to identify the corresponding user, whereas the TA does not need to be online all the time.

3 RELATED WORK

Anonymous credential systems enable user's to anonymously obtain credentials for a pseudonym from an identity provider, e.g. a signed token of the age of the user, which can be anonymously shown to other parties. These credentials can either be one-show (Brands et al., 2007), which are essentially based on blind signatures, or multi-show (Camenisch and Lysyanskaya, 2001), which are based on group signatures. The latter means that multiple showings of the same credential cannot be linked. Clearly, anonymous identification can be implemented by means of anonymous credential systems. However, we do not require the variety of features of anonymous credential systems.

Thus, one may use the underlying concept of group signatures (Ateniese et al., 2000; Chaum and van Heyst, 1991) instead. Group signatures enable users to anonymously sign messages on behalf of a group and there exists a designated party, the so called group manager, which is able to identify signers in case of misuse or fraud. However, in contrast to group signatures our approach is fully compatible with real-world scenarios, i.e. public key infrastructures based on X.509 certificates, and adding as well as removing users can be easily achieved at a constant cost. Nevertheless, (Canard and Girault, 2002) have proposed a practical and efficient group signature approach based on smart cards. But, their approach suffers form a main drawback, i.e. compromising the smart card of a single user requires a reinitialization of the entire system. Another approach similar to group signatures are ring signatures (Rivest et al., 2001). Ring signatures enable users to anonymously sign messages on behalf of a group, however, they provide fully ad-hoc groups, there is no group manager involved and their anonymity is unconditional. Consequently, there is no possibility to revoke the anonymity of malicious users. Ring signatures have also been used to realize anonymous identification (Persiano and Visconti, 2003) and there are also approaches to realize ring signatures which provide anonymity revocation (Xu and Yung, 2004). Nevertheless, in ring signature schemes the user needs to perform a number of public key operations that is linear in the size of the ad-hoc group. In contrast to ring signatures, our approach reduces the computational cost by means of probabilistic anonymity and thus provides more efficiency and higher anonymity compared to ring signatures.

4 TRACEABLE RING AUTHENTICATION

In context of traceable ring authentication (TRA) we speak of the service provider (SP) who represents the verifier and a group $\mathcal{U}$ of authorized users, the so called ring, whereas every user $u \in \mathcal{U}$ may play the role of a prover. The task for a prover is to identify himself to the verifier, by proving membership in the group $\mathcal{U}$, such that the verifier solely learns the membership, but not the exact identity of the prover. Traceable ring authentication additionally provides the possibility to identify a user who has conducted the anonymous identification by means of a third party, the so called traceability authority (TA).

Definition 1. A traceable ring authentication (TRA) protocol is said to be secure if it satisfies the following properties:

Anonymity. A TRA scheme is said to be anonymous, if a SP is not able to determine the identity of an authenticating user with probability higher than $1/|\mathcal{U}|$.

Correctness. A TRA scheme is said to be correct, if the verifier always accepts a proof when he performs the protocol with an honest prover in $\mathcal{U}$.

Unforgeability. A TRA scheme is said to be unforgeable, if every non-member $u \notin \mathcal{U}$ is unable to run a protocol successfully with respect to any $\mathcal{U}' \subseteq \mathcal{U}$.

Unlinkability. A TRA scheme is said to provide unlinkability, if different transcripts of the protocol produced by the same prover can not be linked.

Traceability. A TRA scheme is said to be traceable, if SP, given the protocol transcript, in cooperation with the TA is able to identify the user who has conducted the anonymous identification.

No-missatribution. A TRA scheme is said to provide no-missatribution, if the SP is not able to manipulate the identity escrow information in such a way, that the TA would be able to attribute an anonymous identification to a user who has not conducted the anonymous identification. This property also needs to hold for all users too.

4.1 Ring Authentication

Anonymous identification realized by ring authentication can be described by means of the following protocols.

- REGISTER. An interactive protocol between a user u_i and the SP. User u_i provides identifying information ID_{u_i} together with a certified public key PK_{u_i} suitable for encryption, to SP, who adds the tuple (ID_{u_i}, PK_{u_i}) to a public directory $\mathcal{D}$.

- PROVE. An interactive protocol between a user u_i and the SP. SP sends an encrypted challenge vector to u_i who extracts the challenge and sends it back to SP. If both challenges match, SP accepts, otherwise he rejects the ring authentication.

Protocol 1. PROVE

1. $\mathbf{SP} \rightarrow \mathbf{u_i}$: Choose random $r \in_R \{0,1\}^k$. Generate and send $\langle C_1 = E_{PK_{u_1}}(r, \omega_1), \ldots, C_n = E_{PK_{u_n}}(r, \omega_n)\rangle$, where $\omega_i = f_R(r, ID_{u_i})$.
2. $\mathbf{u_i} \rightarrow \mathbf{SP}$: Decrypt C_i to obtain r'. Check for all $j \neq i$, $1 \leq j \leq n$, whether $C_j = E_{PK_{u_j}}(r', \omega_j)$ holds, where $\omega_j = f_R(r', ID_{u_j})$. If this is true send r' otherwise terminate the protocol.
3. $\mathbf{SP}$: Check whether $r' = r$ holds.

Subsequently we will provide a detailed description of the PROVE protocol (see protocol 1). For simplicity, let us assume that user u_i proves membership in the entire group $\mathcal{U}$ of n users, i.e. all users listed in $\mathcal{D}$. The idea behind PROVE is that SP encrypts a random challenge r for every user $u_i \in \mathcal{U}$ and the auxiliary random coins ω_i for the probabilistic public key encryption scheme, which are also called randomizers, are not chosen uniformly at random, but computed by means of a pseudorandom function f_R which is parametrized by the challenge r and the identity of the respective user ID_{u_i}. Note, that the output distribution of a pseudorandom function is indistinguishable from uniformly distributed strings of equal length for every computationally bound distinguisher. The user decrypts the challenge r and, checks by means of the pseudorandom function f_R, whether SP behaves honestly, i.e. has encrypted the same challenge for every user. If this holds, the user returns the challenge to the service provider, who on his part checks whether the challenges match. It should be noted that for efficiency purposes the checking on the user's side may also be probabilistic, i.e. the user only checks whether $\kappa < n$ randomly chosen elements of the vector were encrypted properly. However, this provides only probabilistic anonymity, whereas the chances for a cheating verifier heavily depend on the parameters κ and n (see also section 4.4).

One efficient implementation of protocol 1 can be achieved by using OAEP (Bellare and Rogaway, 1993) with low exponent RSA, which is also reasonable for a practical implementation, since RSA is among the most widespread cryptosystems in use today. If we treat cryptographic hash functions, e.g. SHA-1, as random oracles, then we can instantiate our pseudorandom function f_R by means of a collision resistant cryptographic hash function H, i.e. $\omega_i = H(r||ID_{u_i})$.

4.2 Achieving Traceability

In order to achieve traceability we employ a tamper resistant device, e.g. a smart card, for every user. This device performs, among others, identity escrow on behalf of the user in such a way, that a cheating user is not able to manipulate the escrowed identity information. Therefore we additionally introduce a new entity called the traceability authority (TA) which is in possession of a key pair (SK_{TA}, PK_{TA}) of a public key encryption scheme that provides non-malleability under chosen plaintext attacks (NM-CPA). The public key PK_{TA} will be integrated into the user's smart card. It must be mentioned, that this party will not be involved online in the protocols, but may be contacted by the SP in case of misuse or fraud. Furthermore, in order to firstly achieve improved reliability and secondly to reduce the required trust, the secret decryption key corresponding to PK_{TA} may also be shared among n TAs, e.g. by means of a (t,n)-threshold scheme. Subsequently, we will describe the protocols, whereas the REGISTER protocol stays unchanged and will not be explicitly treated here.

- REGISTER_ESCROW. An interactive protocol between the user u_i, his smart card SC_{u_i} and the traceability authority (TA). The user chooses a second identifier, i.e. a pseudonym, γ_{u_i} at random and sends it together with ID_{u_i} to the TA. The TA stores the tuple (γ_{u_i}, ID_{u_i}) and keeps γ_{u_i} secret, such that it is only known to u_i and TA, and gives γ_{u_i} and PK_{TA} to SC_{u_i}.
- PROVE_T. An interactive protocol between a user u_i, his smart card SC_{u_i} and SP. u_i chooses $\mathcal{D}' \subset \mathcal{D}$ and sends a suitable encoding of the identities in $\mathcal{D}'$ to SP. SP sends an encrypted challenge vector to u_i, who gives the challenge vector, a vector of all public keys in $\mathcal{D}'$ and a security parameter κ to SC_{u_i}. SC_{u_i} decrypts the challenge and checks for κ public keys whether the challenge was encrypted properly. If the check fails SC_{u_i} returns $\perp$, otherwise it encrypts the challenge together with the identity of u_i for TA, the resulting ciphertext and the challenge for SP and returns the result to u_i. Subsequently, u_i sends the result back to SP. If the decrypted challenge and the send challenge match, SP accepts, otherwise he rejects the anonymous identification.
- IDENTIFY_TRA. An interactive protocol between SP and TA, whereas SP sends a transcript of the traceable ring authentication protocol to TA and TA returns the identity ID_{u_i} of the corresponding user.

Protocol 2. PROVE_T

1. $\mathbf{u_i} \rightarrow \mathbf{SP}$: u_i randomly chooses $\mathcal{D}' \subset \mathcal{D}$, whereas $u_i \in \mathcal{D}'$, and sends $\textsc{enc}(\mathcal{D}')$ to SP.
2. $\mathbf{SP} \rightarrow \mathbf{u_i}$: Choose random $r \in_R \{0,1\}^k$. Parse $\textsc{enc}(\mathcal{D}')$, generate and send $\langle C_1 = E_{PK_{u_1}}(r, \omega_1), \ldots, C_n = E_{PK_{u_n}}(r, \omega_n)\rangle$, where $\omega_i = f_R(r, ID_{u_i})$.
3. $\mathbf{u_i} \leftrightarrow \mathbf{SC_{u_i}}$: Send $\langle\langle C_1, \ldots, C_n\rangle, \langle PK_{u_1}, \ldots PK_{u_n}\rangle, \kappa\rangle$ to SC_{u_i}. SC_{u_i} decrypts C_i to obtain r'. For $1, \ldots, \kappa$ it chooses $j \in_R \{1, \ldots, i-1, i+1, \ldots, n\}$ without duplicates and checks whether $C_j = E_{PK_{u_j}}(r', \omega_j)$ holds, where $\omega_j = f_R(r', ID_{u_j})$. If this holds for all κ checks, it chooses ρ_1, ρ_2 at random, computes $c_1 = E_{PK_{TA}}(r'||\gamma_{u_i}, \rho_1)$ and $c_2 = E_{PK_{SP}}(r'||c_1, \rho_2)$ and returns c_2 to u_i. Otherwise it returns $\perp$.
4. $\mathbf{u_i} \rightarrow \mathbf{SP}$: If $c_2 \neq \perp$ send c_2 to SP.
5. $\mathbf{SP}$: Compute $r'||c_1 = D_{SK_{SP}}(c_2)$ and check whether $\textsc{msb}_k(r'||c_1) = r$. Store the tuple $(r, c_1, \textsc{time})$.[a]

[a] $\textsc{msb}_i(s)$ denotes the most significant i bits of the bitstring s and TIME represents a timestamp.

A detailed description of PROVE_T is given in protocol 2. Note, that user u_i does not authenticate against the entire directory $\mathcal{D}$, but a subset $\mathcal{D}'$ of appropriate size, such that the protocol can be used efficiently but provides enough anonymity, e.g. $|\mathcal{D}'| = 100$. Furthermore we assume that ENC provides a compact encoding of the indices of all ID_{u_i} in $\mathcal{D}$. For simplicity, we assume in protocol 2, that $\mathcal{D}'$ is of cardinality n and the authenticating user u_i holds position i in $\mathcal{D}$. Furthermore, for simplicity we assume that there is a single traceability authority (TA). The idea behind PROVE_T is, that protocol 1 is extended by means of a tamper resistant smart card, which performs all cryptographic operations on behalf of the user in a way such that the user is not able to manipulate the escrowed identity information. Therefore, the decrypted challenge r' must not be visible to the user in plain at any time. Clearly, if the smart card would provide (r', c_1) to the user, the user may easily substitute c_1 with any bit string, without the SP being able to detect it. Hence, in case of misuse or fraud the TA would not be able to recover the identity of the user. Therefore the smart card additionally encrypts (r', c_1) for the SP and provides c_2 to the user.

Protocol 3. IDENTIFY_TRA

1. $\mathbf{SP} \rightarrow \mathbf{TA}$: Send $(r, c_1, \textsc{time})$ to TA.
2. $\mathbf{TA} \rightarrow \mathbf{SP}$: Compute $r'||\gamma_{u_i} = D_{SK_{TA}}(c_1)$ and verify whether $\textsc{msb}_k(r'||\gamma_{u_i}) = r$. If this holds find ID_{u_i} corresponding to γ_{u_i} and send ID_{u_i} to SP.

Since we require the public key encryption scheme for the latter operation to provide NM-CPA security and the escrow identities γ for all other users are not known to the user and furthermore are chosen at random, the user will not be able to misattribute the protocol to another user. The same holds for the SP, which is also not in possession of the escrow identities γ of all users.

In the IDENTIFY_TRA protocol (see protocol 3) the SP, who wants to identify a user who conducted an anonymous identification at time TIME for some reason, provides the stored tuple $(r, c_1, \textsc{time})$ to the TA, which decrypts c_1 by means of its secret decryption key SK_{TA} and verifies whether the provided and the encrypted challenge matches. If this holds it looks up the identity ID_{u_i} corresponding to γ_{u_i} and sends ID_{u_i} to SP. The SP may subsequently remove the entry (ID_{u_i}, PK_{u_i}) from $\mathcal{D}$ such that user u_i will not be able to anonymously identify himself to SP in the future anymore.

4.3 Separability & IBE Setting

As we have mentioned earlier, one efficient realization of (traceable) ring authentication can be achieved by means of RSA-OAEP. However, we are not limited to a specific public key encryption scheme. Moreover, users may register to a service provider using public keys of different schemes. But it should be noted, that firstly the pseudorandom function f_R needs to be chosen according to the respective scheme and secondly the user's smart card must be capable of computing all cryptographic operations for these schemes. Alternatively, an elegant way of realizing (traceable) ring authentication is the use of identity-based encryption (IBE) schemes, e.g. the FULLIDENT scheme of (Boneh and Franklin, 2001). In contrast to traditional public key cryptography, in IBE the public key of a user can be computed by means of an identity string. Obviously, this reduces the size of entries in $\mathcal{D}$. Traditional public keys integrated in X.509 certificates consume about 1 KByte of storage space and consequently transmission bandwidth, whereas the representation of the public key of a user is reduced to a few bytes, e.g. an email address, in case of IBE.

4.4 Efficiency Considerations

The proposed protocol for TRA is very efficient in terms of round complexity and in particular solely needs one round of communication. However, a large number of authorized users ($\mathcal{D}$) may represent a bottleneck for the efficiency of the scheme. As already implicitly applied in protocol 2, one, however, may choose a subset $\mathcal{D}' \subset \mathcal{D}$ of cardinality n of all authorized users for a traceable ring authentication. For instance, the choice of $|\mathcal{D}'| = 100$ would require the user's smart card to perform 99 public key operations and a single private key operation considering the challenge vector. This can be realized at the additional cost of one message (send $\mathcal{D}'$ to SP), i.e. three messages overall.

As already noted in section 4.1, the user may only perform $\kappa < n$ public key operation which results in probabilistic anonymity, i.e. SP may cheat without the user being able to detect it. Clearly, SP may encrypt distinct challenges $r_1, \ldots, r_n$ in order to uniquely identify the authenticating user. But this will only work if the user chooses $\kappa = 0$. Instead, SP may only encrypt some distinct r in order to reduce the anonymity of users. However, if the user chooses κ appropriately, i.e. $n-1 \gg \kappa \geq 10$, the probability that a cheating SP succeeds will be $2^{-\kappa}$. Hence, the chances to cheat unnoticeable decrease exponentially in κ as κ increases. Thus, the choice of κ mentioned before seems reasonable for practical purposes. Consequently, the number of public key operations which need to be performed can be reduced to a small value of κ.

It must be noted, that in our scheme SP solely requires to manipulate $\mathcal{D}$ to add new or remove users. Hence, users need to update their local copy of $\mathcal{D}$ from time to time in order to use the actual set of authorized users. Nevertheless, sate of the art security tokens provide enough storage to manage $\mathcal{D}$. Moreover, users may only update and maintain *ID*'s of authorized users and load corresponding public keys from time to time.

In order to obtain an understanding of the performance of the proposed protocol, we will provide an estimation of the user's computation cost based on state of the art cryptographic hardware for security tokens (see table 2) subsequently. Due to the fact, that RSA is actually the most common public key cryptosystem used for encryption in context of security tokens we will base our analysis on the RSA scheme. More precisely, we will use RSA-OAEP with a modulus size of $m_1 = 1024$ and $m_2 = 2048$ bit for encryption keys of users respectively. The security of 1024 and 2048 bit RSA is assumed to be sufficient till 2010 and 2030 respectively, assuming that there will be no breakthrough in quantum computation. Furthermore, it must be mentioned that values encrypted under user's public keys only have a very short life time. More care should be taken with the choice of the escrow key of the *TA*. For the time being, however, we assume that 2048 bit will be sufficient. In our performance estimation hash function evaluations and other operations will be neglected, and we will only consider public and private key operations as well as the protocol versions providing probabilistic anonymity.

Table 1: Cryptographic performance of a state of the art cryptographic controller for security tokens (SLE 88CFX4002P from Infineon) for private key (SK) and public key (PK) operations.

Scheme	**[ms]**
RSA 1024 bit (PK)	0.5
RSA 1024 bit (SK)	4
RSA 2048 bit (PK)	35
RSA 2048 bit (SK)	11

Table 2: Performance evaluation for $|\mathcal{D}'| = 100$ and $\kappa = 10$ and modul size of $m_1 = 1024$ and $m_2 = 2048$. The table provides computation cost for the user (C-U), estimated duration of the computation for the user (T-U), computation cost for the service provider (C-SP) and communication costs (Comm).

	C-U	**T-U [ms]**	**C-SP**	**Comm [bit]**
RA$_{1024}$	$10PK+SK$	9	$100PK$	$100m_1+k$
TRA$_{1024}$	$12PK+SK$	11	$101PK$	$100m_1+m_2$
RA$_{2048}$	$10PK+SK$	361	$100PK$	$100m_2+k$
TRA$_{2048}$	$12PK+SK$	466	$101PK$	$101m_2$

5 SECURITY ANALYSIS

5.1 Some Aspects

One problem that is inherent to the anonymity of the two protocols is the following: If a user chooses a strict subset $\mathcal{D}'$ of users in $\mathcal{D}$ for efficiency purposes, say of cardinality 100, the SP may have inserted fake identities and fake certified public keys into the directory $\mathcal{D}$. Assume, that a user u_i, who conducts an anonymous identification using some $\mathcal{D}'$ of cardinality 100, may unluckily chose 50 fake certificates. Consequently, the anonymity will be reduced to $1/50$, since SP is able to sort out the faked certificates. However, it must be mentioned that if public keys are certified by some commonly trusted certification authority, which also checks the identity of the respective user before issuing certificates, this threat does no longer exist. It is desirable that the communication channel between the user and the SP provides confidentiality and integrity. Clearly, all message from the user to SP can be encrypted by means of the public encryption key PK_{SP} of SP. However, securing the communication from SP to the user cannot be realized by means of public keys since this would contradict the anonymity. However, a user can randomly choose a secret key of a block cipher, e.g. AES, for every anonymous identification and send this key encrypted under SP's public key to SP. The communication from the SP to the user can consequently be encrypted using a mode of operation that provides authenticated encryption, e.g. the Galois/Counter Mode (GCM) (Dworkin, 2007) using this single secret key.

5.2 Traceable Ring Authentication

Theorem 1. *The traceable ring authentication presented in section 4.2 is secure with respect to definition 1.*

Subsequently, we sketch the proof of theorem 1 by inspecting all properties.

Anonymity. Firstly, we will look at a honest but curious service provider, represented as adversary $\mathcal{A}$. Let $c_1,\ldots,c_n$ be the challenge vector which is sent by $\mathcal{A}$ to some user. Hence it must hold that there exist $\omega_i,\ldots,\omega_n$ such that $c_i = E_{PK_{u_i}}(r,\omega_i)$ holds for all i. By correctness of the used encryption scheme this implies that $r = D_{SK_{u_i}}(c_i)$ for all i. Consequently, $\mathcal{A}$s view of this attack is identical to the view for any j chosen in experiment $\mathsf{Expt}^{\mathsf{anon}}_{TRA,\mathcal{A}_{SP},n}(k)$ and the probability of $j = i$ is at most $1/n$.Secondly, we need to investigate the aspect of pseudorandomly chosen randomizers. Since the pseudorandom function f_R is treated as a random oracle, i.e. the cryptographic hash function H, and furthermore the random challenge r is of appropriate size and fully unknown to to any party except the service provider, the semantic security of the encryption scheme holds.

Correctness. The correctness of the TRA protocol holds by construction.

Unforgeability. If we assume there exists an adversary $\mathcal{A}$ which is able to win the unforgeability experiment $\mathsf{Expt}^{\mathsf{unf}}_{TRA,\mathcal{A}_{NA},n(k)}$ with non-negligible probability, then adversary $\mathcal{A}$ could be used by an adversary $\mathcal{A}_E$ that attacks the used encryption scheme, i.e. the IND-CPA security. Therefore $\mathcal{A}_E$ is given public keys $PK_1,\ldots,PK_n$ and chooses two messages m_0 and m_1. A bit b is chosen at random (unknown to $\mathcal{A}_E$) and $c_1,\ldots,c_n$ is given to $\mathcal{A}_E$, where the c_i's encrypt m_b. $\mathcal{A}_E$ gives $c_1,\ldots,c_n$ as challenge vector to $\mathcal{A}$. Consequently $\mathcal{A}_E$ receives back m from the user part of $\mathcal{A}$. $\mathcal{A}_E$ checks whether $m = m_0$ or $m = m_1$ holds and outputs b'. Thus, $\mathcal{A}_E$ is able to win the IND-CPA experiment with non-negligible probability and this contradicts the assumption that the encryption scheme provides IND-CPA security.

Unlinkability. The unlinkability property of the TRA follows from the anonymity property. What we need to look at is the identity escrow information c_1 of every TRA protocol. Since the used encryption scheme is NM-CPA secure, all possible plaintext are equally probable to result in the escrowed identity information c_1. Since SP will not have access to a decryption oracle (TA solely provides a result for valid escrowed identities and otherwise will accuse SP to be dishonest) he will not be able to link transcripts of the TRA protocol by means of escrowed identity information.

Traceability. Since the smart card is tamper resistant and trusted, we can be sure that γ_{u_i} is escrowed if SC_{u_i} runs a TRA protocol with user u_i. Consequently, TA will be able to extract γ_{u_i} from any tuple $(r,c1,\text{TIME})$ and will be able to provide ID_{u_i} to SP.

No-missatribution. By construction of the TRA protocol, user u_i registers a pseudonym γ_{u_i} with the traceability authority (TA). Hence, γ_{u_i} is not known to the service provider (SP) for all users $1 \leq i \leq n$. Recall, SP stores the tuple $(r,c1,\text{TIME})$ for every instance of the TRA protocol, whereas $c_1 = E_{PK_{TA}}(r'||\gamma_{u_i},\rho_1)$. The tuple is sent to TA in case of anonymity revocation. Hence, in order to miss attribute an instance of the TRA to some user, SP would need to construct c'_1 which decrypts to r' and some valid γ_{u_j} for some user $1 \leq j \leq n,\ j \neq i$. Clearly, SP knows r', but none of the pseudonyms γ_{u_i}. Since we require the public key encryption scheme to provide NM-CPA security, which implies IND-CPA security and SP has no access to a decryption oracle, SP can only guess γ_{u_j}. Since we assume that the bit length of γ is chosen appropriately, the success probability of SP is negligible. □

6 CONCLUSIONS

In this paper we have proposed a practical protocol for traceable anonymous identification which can easily deployed using state of the art smart cards. Moreover, the protocol is highly efficient since it has optimal round complexity and furthermore it is fully compatible with real world scenarios, i.e. public key infrastructures based on X.509 certificates.

REFERENCES

Ateniese, G., Camenisch, J., Joye, M., and Tsudik, G. (2000). A Practical and Provably Secure Coalition-

Resistant Group Signature Scheme. In *CRYPTO '00*, volume 1880 of *LNCS*, pages 255–270. Springer.

Bellare, M. and Rogaway, P. (1993). Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In *CCS '93*, pages 62–73, New York, NY, USA. ACM.

Boneh, D. and Franklin, M. K. (2001). Identity-Based Encryption from the Weil Pairing. In *CRYPTO '01*, volume 2139 of *LNCS*, pages 213–229, London, UK. Springer.

Brands, S., Demuynck, L., and Decker, B. D. (2007). A Practical System for Globally Revoking the Unlinkable Pseudonyms of Unknown Users. In *ACISP 2007*, volume 4586 of *LNCS*, pages 400–415. Springer.

Camenisch, J. and Lysyanskaya, A. (2001). An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation. In *EUROCRYPT '01*, volume 2045 of *LNCS*, pages 93–118, London, UK. Springer.

Canard, S. and Girault, M. (2002). Implementing Group Signature Schemes with Smart Cards. In *CARDIS '02*, pages 1–10. USENIX.

Chaum, D. and van Heyst, E. (1991). Group Signatures. In *EUROCRYPT '91*, volume 547 of *LNCS*, pages 257–265. Springer.

Chor, B., Goldreich, O., Kushilevitz, E., and Sudan, M. (1995). Private Information Retrieval. In *FOCS '95*, pages 41–50. IEEE Computer Society.

Danezis, G. and Diaz, C. (2008). A Survey of Anonymous Communication Channels. Technical Report MSR-TR-2008-35, Microsoft Research.

Dworkin, M. (2007). Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC. In *National Institute of Standards and Technology SP 800-38D*.

Froomkin, M. (2000). The Death of Privacy? *Stanford Law Review*, 52(5):1461–1543.

Lindell, Y. (2007). Anonymous Authentication - Preserving Your Privacy Online. *Black Hat 2007*.

Naor, M. (2002). Deniable Ring Authentication. In *CRYPTO '02*, volume 2442 of *LNCS*, pages 481–498. Springer.

Persiano, P. and Visconti, I. (2003). A Secure and Private System for Subscription-Based Remote Services. *ACM Trans. Inf. Syst. Secur.*, 6(4):472–500.

Rivest, R. L., Shamir, A., and Tauman, Y. (2001). How to Leak a Secret. In *ASIACRYPT '01*, volume 2248 of *LNCS*, pages 552–565. Springer.

Schechter, S., Parnell, T., and Hartemink, A. (1999). Anonymous Authentication of Membership in Dynamic Groups. In *Proc. International Conference on Financial Cryptography 1999*, volume 1648 of *LNCS*, pages 184–195. Springer.

Xu, S. and Yung, M. (2004). Accountable Ring Signatures: A Smart Card Approach. In *CARDIS'04*, pages 271–286. Kluwer.

APPENDIX

Below we define two experiments for the anonymity and unforgeability property respectively where n represents the number of authorized users and k is a security parameter for the key generation algorithm G.

The Anonymity Experiment $\mathsf{Expt}^{\mathsf{anon}}_{TRA,\mathcal{A}_{SP},n}(k)$:

1: G generates $PK_1,\dots,PK_n$.

2: Index i is secretly chosen uniformly at random from $1,\dots,n$.

3: The malicious SP $\mathcal{A}_{SP}$ is given all public keys $PK_1,\dots,PK_n$ and user u_i is given SK_i.

4: The TRA protocol is executed between $\mathcal{A}_{SP}$ and u_i, whereas $\mathcal{A}_{SP}$ has access to an encryption oracle $\mathcal{O}^E(m,j)$, which encrypts a message m with the public key PK_j, $1 \leq j \leq n$.

5: At the end of the experiment, $\mathcal{A}_{SP}$ outputs an index i', $1 \leq i' \leq n$. $\mathcal{A}_{SP}$ has *succeeded* in the experiment, if and only if $i' = i$, which is denoted as $\mathsf{Expt}^{\mathsf{anon}}_{TRA,\mathcal{A}_{SP},n}(k) = 1$.

The Unforgeability Experiment $\mathsf{Expt}^{\mathsf{unf}}_{TRA,\mathcal{A}_{NA},n}(k)$:

1: G generates $PK_1,\dots,PK_n$.

2: The SP and the malicious non authorized user $\mathcal{A}_{NA}$ are both given all public keys $PK_1,\dots,PK_n$.

3: The TRA protocol is executed between SP and $\mathcal{A}_{NA}$, whereas $\mathcal{A}_{NA}$ has access to an encryption oracle $\mathcal{O}^E(m,j)$, which encrypts a message m with the public key PK_j, $1 \leq j \leq n$.

4: At the end of the experiment, $\mathcal{A}_{NA}$ has *succeeded* in the experiment, if and only if SP accepts the TRA protocol, which is denoted as $\mathsf{Expt}^{\mathsf{unf}}_{TRA,\mathcal{A}_{NA},n}(k) = 1$.