Fair Anonymous Authentication for Location Based Services

Panayiotis Kotzanikolaou[1], Emmanouil Magkos[2], Nikolaos Petrakos[1], Christos Douligeris[1], and Vassilis Chrissikopoulos[2]

[1] University of Piraeus, Department of Informatics, 80 Karaoli & Dimitriou, 18534 Piraeus, Greece
{pkotzani,npetrako,cdoulig}@unipi.gr
[2] Ionian University, Department of Informatics, Plateia Tsirigoti 7, 49100, Kerkyra, Greece
{emagos,vchris}@ionio.gr

Abstract. We propose an efficient anonymous authentication scheme that provides untraceability and unlinkability of mobile devices, while accessing Location-Based Services. Following other recent approaches for mobile anonymity, in our scheme the network operator acts as an anonymous credential issuer for its users. However, our scheme supports credential non-transferability, without requiring embedded hardware security features. In addition it supports *fairness* characteristics. On one hand, it reduces the trust assumptions for the issuer by supporting *non-frameability*: the issuer, even in collaboration with the LBS provider, cannot simulate a transaction that opens back to an honest user. On the other hand, it supports *anonymity revocation* for illegally used credentials. Our scheme uses standard primitives such as zero-knowledge proofs, MACs and challenge/responses. We provide formal security proofs based on the intractability of the Divisible Diffie-Hellman assumption.

Keywords: Anonymity, fairness, non-frameability, non-transferability.

1 Introduction

Ongoing research seeks for efficient and *fair* solutions that correctly balance access control and privacy requirements in anonymous authentication [6,9,3]. A basic goal from the point of view of a user, is authenticating to a service, without revealing the user identity (*user untraceability*) and without allowing the linkage of different accesses (*transaction unlinkability*). Another requirement for the user is *non-frameability*, *i.e.*, it should not be possible for anyone, even a collaboration of entities, to successfully simulate an anonymous access that opens back to an innocent user. From the point of view of the service provider, a goal is preventing users from transferring or sharing their credentials with others (*non-transferability*), from using a one-show credential more than once (*non-reusability*), but also establishing accountability when users behave dishonestly. In such cases it may be required to trace a transaction (*anonymity revocation*) and/or revoke all the anonymous credentials of a user (*credential revocation*).

R. Di Pietro et al. (Eds.): DPM 2012 and SETOP 2012, LNCS 7731, pp. 1–14, 2013.

In this paper we focus on anonymous authentication of mobile users accessing Location Based Services (LBS), such as point-of-interest services where a user sporadically queries an LBS provider to receive a nearby point of interest (*e.g.*, [12,15]) or people-locator services, where a watcher asks the LBS provider for the location of a target (*e.g.*, [14]). Typically, a user is requested to provide privacy-sensitive information to LBS service providers, such as location and itinerary, along with identifying information. The provision of both location and identification information to the network operator is generally considered acceptable: In a cell network, the operator already knows location and identity information of each subscriber in the network layer (using cell information and the IMEI/IMSI numbers). Otherwise, network connectivity is not possible. This however is considered acceptable, since the user is contracted with the network operator, who is subject to legal and regulatory constraints concerning users' privacy. From the service provider side, the collaboration with a mobile operator can provide a major resource of clients. Also the LBS provider can outsource the billing and accounting to the operator and in this way they can have a mutual economic interest. Evidently however, the provision of privacy-sensitive information to LBS providers raises privacy concerns, since they may be able to create and misuse user profiles. Moreover, LBS providers can be located anywhere world-wide, making it hard to impose regulatory and audit controls.

An efficient and "fair" anonymous authentication protocol could be trivially constructed if we considered the network operator as a trusted credential issuer. An LBS provider would provide access only if the anonymous credential was validated by the issuer. In case of dishonest behavior the issuer would be able to revoke the anonymity of a user. This solution however, has two major drawbacks: First, it is easy for users to transfer their credentials to others (thus violating non-transferability). Second, it is also easy for the operator together with the LBS provider, to fabricate transaction data that open to a user (thus violating non-frameability). In a fair system no entity should be fully trusted.

Our contribution. We propose an efficient anonymous authentication scheme for LBS services. In our scheme, each anonymous credential is cryptographically linked to a long-term certified public key of the user and is authenticated by the network operator. During the user access phase the use of the corresponding private key will be required, thus preventing a user from transferring credentials to others. Our scheme achieves fairness without reverting to strong (full-trust) assumptions: while the issuer in cooperation with an LBS provider are able to establish accountability and revoke credentials of a misbehaving user, they are not able to frame legitimate users. The scheme also fulfills other fundamental requirements of anonymous authentication such as unlinkability and untraceability of mobile users from LBS providers. We provide formal security proofs based on the intractability of the Divisible Computational Diffie Hellman assumption [1]. Our scheme is efficient for mobile devices, since it requires from the user 5 exponentiations for each credential issuing and 7 for each anonymous access. In Section 2 we describe the system setup and our threat model. In Section 3 we present our scheme, while in Section 4 we provide a proof of security. In Section

5 we review related work in comparison with our scheme and we conclude in Section 6.

2 Setup and Threat Model

We consider a typical mobile network (such as GSM or UMTS infrastructure), which includes the mobile operator $\mathcal{I}$ and a number of users subscribed with $\mathcal{I}$. The users may also access LBS services of independent service providers. For simplicity, we consider a user $\mathcal{U}$ and a provider $\mathcal{SP}$, although it is easy to extend the setup for multiple users and providers. Each user is connected to the network using a mobile device, identified uniquely by the mobile operator at the network layer (IMEI/IMSI). In order to prevent the $\mathcal{SP}$ from linking the location of $\mathcal{U}$ with its real identity, the operator will act as an issuer of anonymous credentials. Each user will be able to obtain multiple one-show credentials validated by $\mathcal{I}$ for a particular provider $\mathcal{SP}$, to anonymously authenticate to $\mathcal{SP}$.

Threat model. We consider both external and internal adversaries. An external adversary may attempt to eavesdrop and intercept the communication between the system entities in order to trace users, retrieve valid user credentials, or to obtain information on whether a credential was accepted. To deal with external adversaries, we assume that the communication channels between $\mathcal{U}$ and $\mathcal{I}$ and between $\mathcal{I}$ and $\mathcal{SP}$ are encrypted and two-way authenticated[1]. The communication between $\mathcal{U}$ and $\mathcal{SP}$ is encrypted and authenticated from $\mathcal{SP}$ to $\mathcal{U}$.

Internal adversaries are constructed by malicious ($\mathcal{U}$ and $\mathcal{SP}$) and semi-trusted ($\mathcal{I}$) internal entities. We distinguish the following cases. An authentication adversary $\mathcal{A}^{auth}$ models a malicious user (or collusion of users) and has access to all the users' credentials. The goal of $\mathcal{A}^{auth}$ is to transfer a usable credential to another user or to generate a new valid credential, and is only limited not to reveal the long-term private key of the user. A tracing adversary $\mathcal{A}^{trace}$ models a malicious service provider (or collusion of them), having access to all the secret information of $\mathcal{SP}$ and the history of all the user access instances. The goal of $\mathcal{A}^{trace}$ is to trace and/or link users by combining all available information. We emphasize that the issuer is not allowed to participate in $\mathcal{A}^{trace}$. Also, we assume that $\mathcal{A}^{trace}$ cannot link/trace users at lower layers (data-link or IP) or by using application content or context to trace/link users (we leave out of scope query content attacks). A framing adversary $\mathcal{A}^{frame}$ models a collusion of a malicious service provider and a semi-trusted issuer. The goal of $\mathcal{A}^{frame}$ is to frame a legitimate user by creating a transaction that opens to the user. $\mathcal{I}$ is semi-trusted since it is allowed to collude with $\mathcal{SP}$ in $\mathcal{A}^{frame}$, but is trusted not to collude with $\mathcal{SP}$ in $\mathcal{A}^{trace}$. Finally, we assume that all the adversaries are polynomially bounded and do not have the ability to break the computational assumptions of the underlying cryptographic assumptions. We also assume that a Public Key Infrastructure (PKI) for certificate management is already in place.

[1] This can be achieved by combining certified signature keys with the TLS protocol.

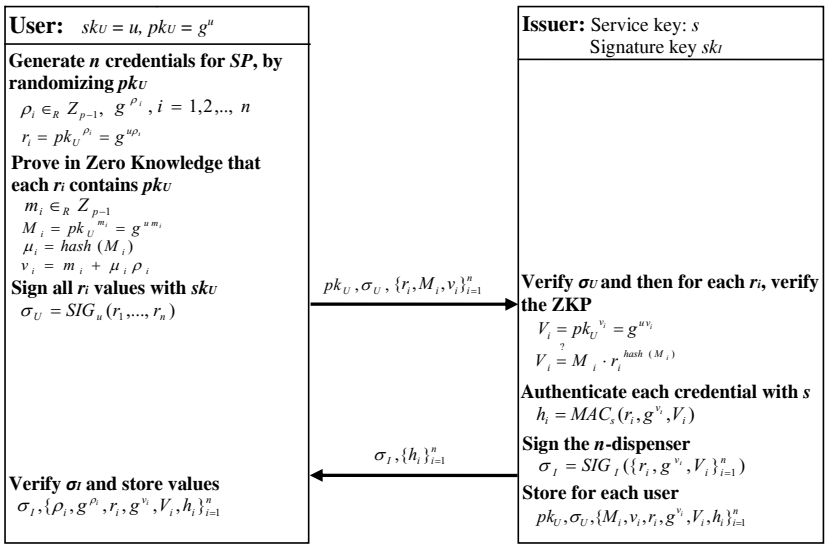

Fig. 1. The credential issuing protocol

3 A Fair, Anonymous Authentication Scheme for LBS

Let $\mathcal{I}$ denote the network operator responsible to issue, update and revoke anonymous user credentials, used to anonymously access an LBS provider $\mathcal{SP}$. We will use a discrete logarithm setting. Let p, q be two sufficiently large primes such that $q|p-1$ and let g be a generator of a multiplicative group G of order p (unless stated otherwise, all operations are done modulo p). Let $sk_U = u$, $pk_U = g^u$ be a certified signature key pair of $\mathcal{U}$ in a discrete log setting (*e.g.*, an ElGamal key pair) with the same generator g. Let sk_I, sk_{SP} be the certified signing keys of $\mathcal{I}$ and $\mathcal{SP}$ respectively, using any signature scheme. Finally, let s be a secret service key, shared between $\mathcal{I}$ and $\mathcal{SP}$.

3.1 Credential Issuing

The credential issuing protocol is executed between $\mathcal{U}$ and $\mathcal{I}$ (Fig. 1). Initially $\mathcal{U}$ chooses n random values $\rho_1, \rho_2, ..., \rho_n \in_R Z^*_{p-1}$ and computes n randomizations of his/her public key as: $r_i = (pk_U)^{\rho_i} \equiv g^{u\rho_i}$, $i = 1, ..., n$. Then, $\mathcal{U}$ will compute n zero-knowledge proofs of knowledge that the values r_i are correctly formed, *i.e.*, that they contain $\mathcal{U}$'s public key g^u. An efficient *non-interactive* proof can be constructed by using the approach in [11]. Specifically, $\mathcal{U}$ chooses a value $m_i \in_R Z^*_{p-1}$, computes $M_i = (pk_U)^{m_i} \equiv g^{um_i}$, then $\mu_i = hash(M_i)$, where $hash$ is a cryptographic hash function, and $\mathrm{v}_i = m_i + \mu_i\rho_i$. For accountability purposes, $\mathcal{U}$ also computes a signature σ_U over the concatenation of $r_1, ..., r_n$. Finally, $\mathcal{U}$ sends to $\mathcal{I}$ the values pk_U, σ_U and $\{r_i, M_i, \mathrm{v}_i\}_{i=1}^n$.

On receiving these, $\mathcal{I}$ will first verify the signature σ_U and then will verify the zero-knowledge proofs M_i, v_i for each r_i. To do this, $\mathcal{I}$ computes $V_i = (pk_U)^{\mathrm{v}_i}$ and

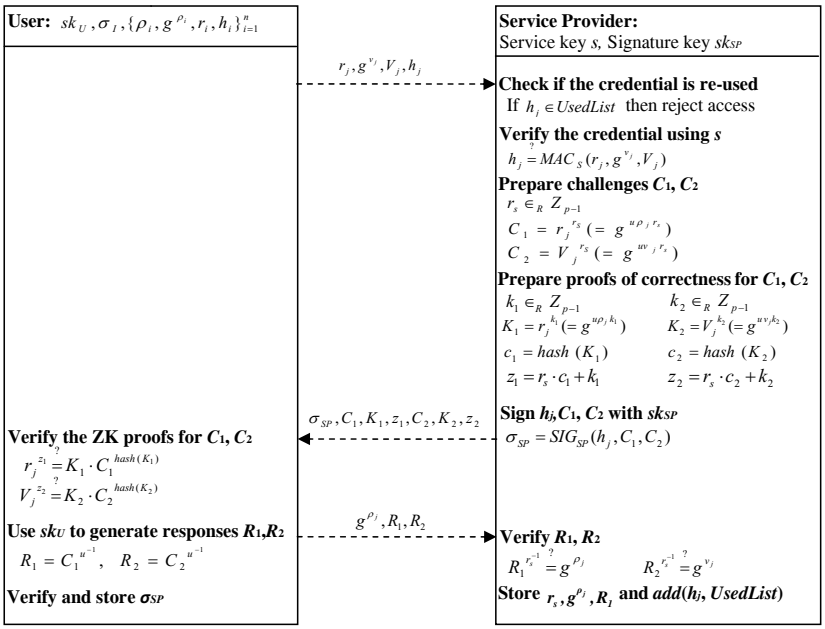

Fig. 2. The user access protocol

checks if $V_i \equiv M_i \cdot r_i^{hash(M_i)}$. If the verifications are successful, $\mathcal{I}$ authenticates each triplet $\{r_i, g^{v_i}, V_i\}$ by computing n Message Authentication Codes (MAC) with the service secret key s (shared only with $\mathcal{SP}$) as: $h_i = MAC_s(r_i, g^{v_i}, V_i)$, $i = 1, ..., n$. Then, $\mathcal{I}$ signs with its private signature key sk_I the concatenation of all the values $\{r_i, g^{v_i}, V_i\}_{i=1}^n$, and returns to $\mathcal{U}$ the credential authenticators h_i. At the same time, $\mathcal{I}$ stores, the public key pk_U, the signature σ_U and the n tuples $\{M_i, v_i, r_i, g^{v_i}, V_i, h_i\}_{i=1}^n$, to be used for possible accountability and/or revocation purposes. $\mathcal{U}$ verifies the signature σ_I and stores it along with the credential values $\{r_i, g^{v_i}, V_i, h_i\}$, and the secret random values ρ_i, and g^{ρ_i}, $i = 1, 2, .., n$. Note that the values g^{ρ_i} are also secret from the issuer and will be later used for non-frameability purposes.

3.2 User Access

The registered user will anonymously access the service provider $\mathcal{SP}$, for n unlinkable transactions, by using his private key $sk_U = u$, a different anonymous credential $\{r_i, g^{v_i}, V_i, h_i\}_{i=1}^n$ and the corresponding value g^{ρ_i} (Fig. 2).

For each access, $\mathcal{U}$ sends a different credential r_j, g^{v_j}, V_j, h_j to $\mathcal{SP}$. The $\mathcal{SP}$ will first verify if the credential has been already used, by checking whether it belongs to the list *UsedList*. If $h_j \in$ *UsedList* the access is rejected. Else the anonymous authentication continues by verifying (with the secret key s) that h_j is a valid MAC. If the credential is valid, the $\mathcal{SP}$ wants to be assured that it is

only used by the legitimate user, who knows the secret key sk_U that corresponds to the public key used during credential issuing. To do this, the $\mathcal{SP}$ prepares two challenges C_1 and C_2, which the anonymous user can respond properly only if he uses the correct key $sk_U = u$. The $\mathcal{SP}$ chooses a random number r_s and computes $C_1 = {r_j}^{r_s} \equiv g^{u\rho_j r_s}$ and $C_2 = {V_j}^{r_s} \equiv g^{uv_j r_s}$. The challenge for $\mathcal{U}$ is to remove from C_1, C_2 the correct user's private key u, *i.e.*, the one that corresponds to the public key used to construct r_j, V_j during the registration phase.

Note that a misbehaving $\mathcal{SP}$ could cheat $\mathcal{U}$ by sending a challenge C_1 (resp. C_2) that does not include r_j (resp. V_j). In that case, the $\mathcal{SP}$ would be able to remove all the exponents from the user's response, get $g^{u^{-1}}$ and link all the transactions of a user. In order to prevent this, the $\mathcal{SP}$ must also prepare zero-knowledge proofs that the challenges C_1, C_2 indeed contain the values r_j and V_j respectively. The $\mathcal{SP}$ chooses random numbers $k_1, k_2 \in_R Z_{p-1}$ and prepares $K_1 = {r_j}^{k_1} \equiv g^{u\rho_j k_1}$ and $K_2 = {V_j}^{k_2} \equiv g^{uv_j k_2}$. Then, the $\mathcal{SP}$ computes the hash values $c_1 = hash(K_1)$, $c_2 = hash(K_2)$. Finally, the $\mathcal{SP}$ computes the values $z_1 = r_s \cdot c_1 + k_1$, and $z_2 = r_s \cdot c_2 + k_2$. The $\mathcal{SP}$ also signs the challenges C_1, C_2 along with the used credential h_j using the key sk_{SP} as $\sigma_{SP} = SIG_{SP}(h_j, C_1, C_2)$. The server sends to $\mathcal{U}$ the values $\sigma_{SP}, C_1, K_1, z_1, C_2, K_2, z_2$.

On receiving these, $\mathcal{U}$ first verifies σ_{SP}, and then verifies that C_1, C_2 contain r_j and V_j, by checking that $r_j^{z_1} \equiv K_1 \cdot C_1^{hash(K_1)}$ and $V_j^{z_2} \equiv K_2 \cdot C_2^{hash(K_2)}$. If the verifications are successful, $\mathcal{U}$ can safely use his/her private key u and generate the responses: $R_1 = {C_1}^{u^{-1}}$ and $R_2 = {C_2}^{u^{-1}}$, without leaking additional information. The user $\mathcal{U}$ stores σ_{SP} (along with h_j, C_1, C_2) in its local store and sends to $\mathcal{SP}$ the responses g^{ρ_j}, R_1 and R_2.

Finally, the $\mathcal{SP}$ will verify that $(R_1)^{r_s^{-1}} \equiv g^{\rho_j}$ and $(R_2)^{r_s^{-1}} \equiv g^{v_j}$, where g^{v_j} is contained in the authenticated credential h_j. We note that showing g^{ρ_j} only during the user access for the verification of C_1 provides the non-frameability property: if g^{ρ_j} was known to $\mathcal{I}$ from the registration phase, it would be possible for $\mathcal{I}$ and $\mathcal{SP}$ to simulate a user access and frame a user (see Section 4.3 for a security proof). The second challenge provides non-transferability (see Section 4.4). If the responses are valid, $\mathcal{U}$ is allowed to anonymously access the requested service. The $\mathcal{SP}$ will store for future reference the values r_s, g^{ρ_j} associated to h_j and will add the credential to the $UsedList$.

Extending the scheme for many LBS providers. This only requires that $\mathcal{I}$ shares a different secret service key s_ℓ for each provider $\mathcal{SP}_\ell$.

3.3 Anonymity Revocation and Credential Revocation

Anonymity revocation. If $\mathcal{SP}$ has convincing arguments of service abuse for a particular access, then $\mathcal{SP}$ securely transfers the particular instance of the access protocol $\{r_j, g^{v_j}, V_j, h_j\}$ to $\mathcal{I}$. From user registration, $\mathcal{I}$ has stored in its database, $pk_U, \sigma_U, \{r_i, g^{v_i}, V_i, h_i\}$, $i = 1, 2, .., n$. The issuer will first verify that $\{r_j, g^{v_j}, V_j, h_j\}$ is indeed a valid credential. Then it will search in its user database, to find in which user's $\widetilde{\mathcal{U}}$ n-dispenser the credential in question is contained. The user $\widetilde{\mathcal{U}}$ will be identified as the misbehaving user. The user cannot

deny that the misused credential belongs to him, since during the issuing phase the user has signed with σ_U all the values r_i used in each credential.

Credential revocation. In order to revoke all the credentials of a user, $\mathcal{I}$ manages a revocation list $revList$, which contains the credentials that had been issued to users that are no longer authorized. To revoke a user $\widetilde{\mathcal{U}}$, $\mathcal{I}$ simply appends to $revList$ all the n credentials that were issued to $\widetilde{\mathcal{U}}$, signs and publishes the list to $\mathcal{SP}$. The updated list is: $revList \leftarrow (revList, \{r_i, g^{\mathrm{v}_i}, V_i, h_i\}_{i=1}^{n})$. The $\mathcal{SP}$ will reject a user access request, if the credential is in $revList$. Note that in order to prevent $\mathcal{SP}$ from linking past transactions of a revoked user, the revocation list should be initialized with random values and then, for each new revoked credential added, the list should be randomly re-ordered. In this way a curious $\mathcal{SP}$ would not be able to link adjacent values within the revocation list.

3.4 Global Accountability / Non-frameability

Say that $\mathcal{SP}$ contacts $\mathcal{I}$ for a possible illegal user access and $\mathcal{I}$ runs the anonymity revocation protocol of Section 3.3 in order to reveal the identity of the user $\widetilde{\mathcal{U}}$ for a particular anonymous access. However, if $\mathcal{I}$ and $\mathcal{SP}$ are able to simulate all the messages of the user, it is possible that they have framed the user $\widetilde{\mathcal{U}}$, and thus the anonymous access scheme cannot provide global accountability. The question here is how to prove to a legal authority (*e.g.*, a Judge) that it is indeed $\widetilde{\mathcal{U}}$ who performed the transaction. For global accountability, the following protocol will be initiated by the global verifier (Judge). The Judge will ask from the parties involved, to provide the following: The $\mathcal{SP}$ gives the disputed user access instance $(r_j, g^{\mathrm{v}_j}, V_j, h_j)$, the random value r_s used to randomize the challenges and the response R_1 along with the value g^{ρ_j}, provided by the user during the user access. The issuer $\mathcal{I}$ provides $pk_{\widetilde{U}}, \sigma_{\widetilde{U}}$ for the user $\widetilde{\mathcal{U}}$ in question. Finally, $\widetilde{\mathcal{U}}$ is asked to provide σ_I, σ_{SP} and the value $\widetilde{\rho_j}$, that he/she used for the construction of his j-th credential. The Judge will execute the following protocol:

1. Check whether $\{r_j, g^{\mathrm{v}_j}, V_j\} \in \sigma_I$, (*i.e.* this is a valid credential signed by $\mathcal{I}$).
2. Check if $r_j \in \sigma_{\widetilde{U}}$, (*i.e.*, $\widetilde{\mathcal{U}}$ has committed to r_j during the issuing phase).
3. Check if $C_1 \equiv r_j^{r_s}$, *i.e.*, r_s is the correct random value that was used by the $\mathcal{SP}$ during the generation of the challenges for this user access instance. Recall that the $\mathcal{SP}$ has committed to the values h_j, C_1, C_2 with σ_{SP} and thus will provide the correct random value.
4. Check that $(pk_{\widetilde{U}})^{\widetilde{\rho_j}} \equiv r_j$, *i.e.* $\widetilde{\mathcal{U}}$ cannot lie for $\widetilde{\rho_j}$, since $r_j \in \sigma_U$.
5. Check if $R_1^{r_s^{-1}} \equiv g^{\rho_j} \equiv g^{\widetilde{\rho_j}}$, where g^{ρ_j} was given to the Judge by the $\mathcal{SP}$ and $\widetilde{\rho_j}$ was given by $\widetilde{\mathcal{U}}$.

If the last check is true, then $\widetilde{\mathcal{U}}$ has performed the user access, since $\mathcal{I}$ and $\mathcal{SP}$ could not present the correct value g^{ρ_j} otherwise (see Section 4.3). If the check is false, this is a framing attempt. Recall that during the issuing phase the user only provides the proof of correctness that r_i contains his public key and not the values ρ_i or g^{ρ_i} used in $r_i = g^{u\rho_i}$. Finally, note that the signature σ_{SP} prevents the

$\mathcal{SP}$ from reusing g^{ρ_j} in order to frame a user. Since the user verifies σ_{SP} before responding to C_1, C_2, if the $\mathcal{SP}$ reused the opened value g^{ρ_j}, the user would provide two signatures $\sigma_{SP} = SIG_{SP}(h_j, C_1, C_2)$ and $\sigma'_{SP} = SIG_{SP}(h'_j, C'_1, C'_2)$, both linked to $g^{\widetilde{\rho_j}}$ and the Judge would be convinced that $\mathcal{SP}$ is framed.

3.5 Efficiency Analysis

Computation. The computation cost for the user for the credential issuing is: $5n + 2$ exponentiations *i.e.* almost 5 exponentiations per credential. The issuer performs $2n + 2$ exponentiations. For each anonymous access, the user performs 7 exponentiations and the provider also performs 7 exponentiations. In total, the user performs 12 exponentiations for each anonymous access. The cost is feasible for mobile devices. If the user runs the issuing protocol from a typical computer and then load the mobile device with the issued credentials, then the cost for the mobile devices can be reduced only to the user access phase.

Bandwidth. Allowing for 128 bytes for public-key operations, 10 bytes for random number selection and 20 bytes for hashing operations, the communication cost of the credential issuing protocol (Fig. 1) is, $542n+384$, for issuing n credentials. Similarly, the communication cost during a user access (Fig. 2) is 1566 bytes.

4 Security Analysis

4.1 Preliminaries and Notations

The size of a finite set $\mathbf{S}$ is denoted as $|\mathbf{S}|$. The term $s \in_R \mathbf{S}$ denotes the assignment of a uniformly chosen element of $\mathbf{S}$ to a variable s. Let A be a p.p.t algorithm. Then $\mathrm{A}(x_1, ..., x_n) = y$ means that on input $x_1, ..., x_n$, the algorithm outputs a value that is assigned to variable y. Let E be some event, such as the result of a security experiment, then $Pr[\mathrm{E}]$ denotes the probability that E occurs. The probability $\epsilon(l)$ is called negligible (in l), if for all polynomials f it holds that $\epsilon(l) \leq 1/f(l)$, for all sufficiently large l. In that case, the probability $1 - \epsilon(l)$ is called overwhelming.

Definition 1 (Divisible Computation Diffie-Hellman assumption.). *Let $l_p \in \mathbb{N}$ be a security parameter, $\mathbb{G}$ be a group of large prime exponent $p \approx 2^{l_p}$. Let g be an element of $\mathbb{G}$ of prime order p. Let $x, y \in_R Z_p^*$ and $X = g^x \mod p$, $Y = g^y \mod p$ and $Z \in_R Z_p$. The DCDH assumption is that every p.p.t adversary $\mathcal{A}^{DCDH}$ has negligible advantage (in l_p):*

$$Adv_{\mathcal{A}}{}^{DCDH} = |Pr[\mathcal{A}(p, g, X, Y) = g^{x/y}]|.$$

Definition 2 (Decisional Divisible Computation Diffie-Hellman assumption.). *Let $l_p \in \mathbb{N}$ be a security parameter, $\mathbb{G}$ be a group of large prime exponent $p \approx 2^{l_p}$. Let g be an element of $\mathbb{G}$ of prime order p. Let $x, y \in_R Z_p^*$ and $X = g^x \mod p$, $Y = g^y \mod p$ and $Z \in_R Z_p$. The DDCDH assumption is that every p.p.t adversary $\mathcal{A}^{DCDH}$ has negligible advantage (in l_p):*

$$Adv_{\mathcal{A}}{}^{DDCDH} = |Pr[\mathcal{A}(p, g, X, Y, g^{x/y}) = 1] - Pr[\mathcal{A}(p, g, X, Y, Z) = 1]|.$$

4.2 Proof of Untraceability and Unlinkability

The tracing adversary $\mathcal{A}^{trace}$ (described in Section 2) should not be able to trace the identity of a user running the anonymous access protocol of Section 3.2. This implies that the protocol messages generated by a user $\mathcal{U}$, should not leak information that will allow $\mathcal{A}^{trace}$ to trace $\mathcal{U}$. The only value that could trace the identity of $\mathcal{U}$, is the long-term public key g^u, used during the credential issuing. Let $\mathbf{U}$ be the set of all the users' public keys. Let $\pi_j^{g^u}$ denote an instance of the anonymous access protocol, in which the j-th credential of a user $\mathcal{U}$ was used. Since every key $g^{\widetilde{u}} \in \mathbf{U}$ is publicly known, the goal of $\mathcal{A}^{trace}$ is to decide if, an anonymous access instance $\pi_j^{g^u}$ is linked or not, with each public key $g^{\widetilde{u}} \in \mathbf{U}$.

We formalize anonymous access by a security experiment $\text{Exp}_{\mathcal{A}}^{tr}$, where $\mathcal{A}^{trace}$ interacts with an oracle $\mathcal{O}^{trace}$ that takes as input, the public parameters p, g, the secret key $sk_{\mathcal{SP}}$ an instance $\pi_j^{g^u}$ of the anonymous access protocol of Section 3.2 and a test public key $g^{\widetilde{u}} \in \mathbf{U}$ and outputs: $b = 1$ if $g^u = g^{\widetilde{u}}$ or $b = 0$ if $g^u \neq g^{\widetilde{u}}$.

Now we must define the information learned from each instance $\pi_j^{g^u}$ of the access protocol (Section 3.2). This information will be given as input to the adversary. Each user credential used by $\mathcal{U}$ contains $r_j = g^{u\rho_j}$, g^{v_j}, $V_j = g^{u\mathrm{v}_j}$ and h_j. The value h_j is a hash of the above and thus does not provide additional information. During the response, the user also sends $g^{u\rho_j}$ and R_1, R_2. We examine what information is leaked during this step. The response to the challenges $C_1 = g^{u\rho_j r_s}$ and $C_2 = g^{u\mathrm{v}_j r_s}$ does not provide additional information to the adversary, other than g^{ρ_j} and g^{v_j}. This is assured by the proofs of correctness K_1, z_1 and K_2, z_2 respectively. By verifying these proofs, $\mathcal{U}$ is assured that the challenge C_1 (resp. C_2) is a randomization of $g^{u\rho_j}$ (resp. $g^{u\mathrm{v}_j}$). Thus the information learned in each anonymous user access protocol instance of $\mathcal{U}$ is: $\pi_j^{g^u} = (g^{u\rho_j}, g^{\mathrm{v}_j}, g^{u\mathrm{v}_j}, g^{\rho_j})$.

Definition 3 (Untraceability of our scheme.). *The anonymous authentication protocol described in Section 3.2 achieves untraceability, if every p.p.t adversary $\mathcal{A}^{trace}$ has negligible (in l_p) advantage:*
$Adv_{\mathcal{A}}^{tr} = |Pr[\text{Exp}_{\mathcal{A}}^{tr}(p, g, sk_{\mathcal{SP}}, \pi_j^{g^u}, g^u) = 1] - Pr[\text{Exp}_{\mathcal{A}}^{tr}(p, g, sk_{\mathcal{SP}}, \pi_j^{g^u}, g^{\widetilde{u}}) = 1]$.

Theorem 1. *The anonymous authentication scheme achieves unlinkability under the Decisional Divisible Computation Diffie-Hellman assumption.*

Proof. Assume by contradiction that the advantage of $\mathcal{A}^{trace}$ is non-negligible, *i.e.* the adversary can distinguish whether a public key $g^{\widetilde{u}}$ and the access protocol instance $\pi_j^{g^u}$ given as input to the oracle $\mathcal{O}^{trace}$ are linked. Give to the oracle: the public parameters g, p, the provider's secret key $sk_{\mathcal{SP}}$, the access protocol instance of a user $\pi_j^{g^u} = (g^{\mathrm{v}_j}, g^{u\mathrm{v}_j}, g^{\rho_j}, g^{u\rho_j})$, and a public key $g^{\widetilde{u}}$. The oracle will output $b = 1$, if the input public key $g^{\widetilde{u}} = g^u$ (i.e. it is the public key used in $\pi_j^{g^u}$), and $b = 0$ otherwise.

Now the adversary $\mathcal{A}^{trace}$ can be used as a subroutine to break the DDCDH assumption of Definition 2. The adversary $\mathcal{A}^{DDCDH}$ will give to the oracle $\mathcal{O}^{trace}$

of the adversary $\mathcal{A}^{trace}$: g^{v_j} (as $X = g^x$), $g^{u\mathrm{v}_j}$ (as $Y = g^y$) and $g^{\widetilde{u}}$ (as Z). Now $\mathcal{A}^{DDCDH}$ will check the output b of the oracle to decide the Decisional Divisible Computational Diffie-Hellman problem with non-negligible probability. If the oracle $\mathcal{O}^{trace}$ outputs $b = 0$, then ${Adv_{\mathcal{A}}}^{DDCDH}$ can decide that $g^{\widetilde{u}} = g^{x/y}$. Else it decides that $g^{\widetilde{u}} \neq g^{x/y}$. $\mathcal{A}^{DDCDH}$ can also use g^{ρ_j} as X and $g^{u\rho_j}$ as Y with the same advantage.

It is easy to prove that the protocol also provides unlinkability under the DDCDH assumption. In that case, the adversary would take as input two different protocol instances and a public key. The output b of the oracle would be 1 if the *two* protocol instances are linked with the public key and 0 otherwise.

4.3 Proof of Non-frameability

The adversary $\mathcal{A}^{frame}$ (described in Section 2) should not be able to simulate a user access protocol instance $\pi_j^{g^u}$, that will open to user's public key g^u, if the global accountability protocol of Section 3.4 is run. We formalize a framing attempt by a security experiment $\mathrm{Exp}_{\mathcal{A}}^{frame}$, where $\mathcal{A}^{frame}$ interacts with an oracle $\mathcal{O}^{frame}$ that takes as input: the public parameters $p,\ g$, the j-th instance of the credential issuing protocol (Section 3.1) of the user $\mathcal{U}$, denoted as $\varpi_j^{g^u}$, the combined secret information of $\mathcal{I}$ and $\mathcal{SP}$ (*i.e.* the private keys sk_{SP}, sk_I and s and the randomness r_s used for the challenges C_1, C_2) and outputs a valid user response of the anonymous access protocol $\pi_j^{g^u}$.

The input information given to $\mathcal{O}^{frame}$ from each instance $\varpi_j^{g^u}$ contains g^u, $r_j = g^{u\rho_j}$, M_j, v_j. Since M_j is only used for the zero knowledge proof of correctness of r_j, it does not provide additional information to the adversary. Thus, for the experiment $\mathrm{Exp}_{\mathcal{A}}^{frame}$, the information given to the adversary oracle for each credential issuing protocol instance is: $\varpi_j^{g^u} = (g^u, g^{u\rho_j}, \mathrm{v}_j)$. In order to simulate a user, $\mathcal{O}^{frame}$ must output all the information generated by the target user $\mathcal{U}$, during the user access protocol of Section 3.2. Thus the output of the oracle is: $\pi_j^{g^u} = (r_j, g^{\mathrm{v}_j}, V_j, h_j,\ g^{\rho_j}, R_1, R_2)$.

Definition 4 (Non-frameability of our scheme.). *The accountability protocol of Section 3.4 will always trace a simulated framing anonymous access instance of the protocol of Section 3.2, if every p.p.t adversary $\mathcal{A}^{frame}$ has negligible (in l_p) advantage:*

$$Adv_{\mathcal{A}}^{frame} = |Pr[\mathrm{Exp}_{\mathcal{A}}^{frame}(p, g, \varpi_j^{g^u}, sk_{SP}, sk_I, s, r_s) = \pi_j^{g^u}].$$

Theorem 2. *The anonymous authentication scheme achieves non-frameability under the Divisible Computation Diffie-Hellman assumption.*

Proof. Assume by contradiction that the advantage of $\mathcal{A}^{frame}$ is non-negligible. $\mathcal{A}^{frame}$ gives as input to the oracle, the public parameters $g,\ p$, an instance of the credential issuing protocol of Section 3.1 for a target user $\mathcal{U}$, $\varpi_j^{g^u} = (g^u, g^{u\rho_j}, \mathrm{v}_j)$, the private keys sk_{SP}, sk_I and s and the randomness r_s used for the challenges $C_1 = g^{u\rho_j r_s}$, $C_2 = g^{u\mathrm{v}_j r_s}$. The oracle outputs a simulated

instance of the anonymous access protocol $\pi_j^{g^u} = (r_j, g^{\mathrm{v}_j}, V_j, h_j, g^{\rho_j}, R_1, R_2) = (g^{u\rho_j}, g^{\mathrm{v}_j}, g^{u\mathrm{v}_j}, h_j, g^{\rho_j}, g^{\rho_j r_s}, g^{\mathrm{v}_j r_s})$, which is indistinguishable from a protocol instance run by the real user $\mathcal{U}$.

Now the adversary $\mathcal{A}^{frame}$ can be used as a subroutine to break the DCDH assumption of Definition 1. The adversary $\mathcal{A}^{DCDH}$ will give to the oracle $\mathcal{O}^{frame}$ as input: $g^{u\rho_j}$ (as $X = g^x$) and g^u (as $Y = g^y$) and v_j. The oracle will output $\pi_j^{g^u}$ as follows: $r_j = g^{u\rho_j}$ (already given in the input), $g^{\mathrm{v}_j}, V_j = g^{u\mathrm{v}_j}$ (using the input values v_j and g^u), h_j (using the key s), g^{ρ_j}, and $R_1 = g^{\rho_j r_s}, R_2 = g^{\mathrm{v}_j r_s}$, (using g^{ρ_j}, g^{v_j} and r_s). However, the output value $g^{\rho_j} \equiv g^{x/y}$, which contradicts the DCDH assumption. Note that possible replay attacks based on an previously received g^{ρ_j} are not possible, due to the use of the signature σ_{SP}, as described in Section 3.4. It is easy to see that the scheme also provides exculpability.

4.4 Proof of Non-transferability

Although the response g^{ρ_j}, R_1 of the challenge C_1 provides non-frameability, it does not prevent a user from transferring one or more credentials to a non-registered user. For example, $\mathcal{U}$ could transfer his j-th credential by giving to another user $\mathcal{U}'$ the values $\rho_j, (\alpha \cdot u^{-1}), g^{\rho_j \alpha}$, for some α. Then $\mathcal{U}'$ would be able to respond C_1, by sending to $\mathcal{SP}$ the response $R_1' = C_1^{(\alpha \cdot u^{-1})}$ and $g^{\rho_j \alpha}$. The verification would work, the long term private key u would not be revealed and the credential transfer would be revealed only if the accountability protocol of Section 3.4 was run. This however is executed only in case of disputes, while non-transferability should be verified for each anonymous access. To avoid, this, the second challenge C_2 is used. The response R_2 is verified against g^{v_j}, which is authenticated with the MAC h_j and thus cannot be manipulated.

We formalize a credential transferring attempt by a security experiment $\mathrm{Exp}_{\mathcal{A}}^{auth}$, where the adversary $\mathcal{A}^{auth}$ (described in Section 2) interacts with an oracle $\mathcal{O}^{auth}$ that takes as input: the public parameters p, g, all the private information related with the j-th user credential $cred_j$ and possible one-way transformations of u, but *not* the long-term private key u of the user, and outputs a valid user response of the anonymous access protocol $\pi_j^{g^u} = (g^{u\rho_j}, g^{\mathrm{v}_j}, g^{u\mathrm{v}_j}, h_j, g^{\rho_j}, R_1, R_2)$. The input information given to $\mathcal{O}^{auth}$ with $cred_j$ includes the public key g^u the credential $g^{u\rho_j}, g^{\mathrm{v}_j}, g^{u\mathrm{v}_j}$, the authenticator h_j, and the values v_j and ρ_j. The oracle also gets as input any one-way transformation of the user's private key u denoted as $f_\alpha(u)$, for every $\alpha \neq r_s$, as well as an one-way transformation of α (this will allow $\mathcal{O}^{auth}$ to simulate the response to the first challenge, without revealing u to $\mathcal{O}^{auth}$). We also give to the oracle, the one-way transformation $f_\alpha(u) = \alpha \cdot u^{-1}$ and g^α. The value $\alpha \in_R Z_p$ is kept secret from $\mathcal{O}^{auth}$, since otherwise the oracle could compute u. Thus for the experiment $\mathrm{Exp}_{\mathcal{A}}^{auth}$, the input information $cred_j$ contains: $cred_j = (g^{u\rho_j}, g^{\mathrm{v}_j}, g^{u\mathrm{v}_j}, h_j, \rho_j, \mathrm{v}_j, f_\alpha(u) = \frac{\alpha}{u}, g^\alpha)$.

Definition 5 (Non-transferability of our scheme.). *The anonymous authentication scheme of Section 3.2 achieves non-transferability, if every p.p.t adversary $\mathcal{A}^{auth}$ has negligible (in l_p) advantage:*

$Adv_{\mathcal{A}}^{auth} = |Pr[\mathrm{Exp}_{\mathcal{A}}^{auth}(p, g, g^u, g^{u\rho_j r_s}, g^{u\mathrm{v}_j r_s}, cred_j) = \pi_j^{g^u}]$.

Theorem 3. *The anonymous authentication scheme achieves non-transferability under the Divisible Computation Diffie-Hellman assumption.*

Proof. Assume by contradiction that the advantage of $\mathcal{A}^{auth}$ is non-negligible. The oracle $\mathcal{O}^{auth}$ outputs a simulated instance of the anonymous access protocol $\pi_j^{g^u} = (g^{u\rho_j}, g^{v_j}, g^{uv_j}, h_j, g^{\rho'_j} = g^{\alpha\rho_j}, R'_1 = g^{\rho'_j r_s}, R_2 = g^{v_j r_s})$, which is indistinguishable from a protocol instance run by the real user $\mathcal{U}$. Recall that during the user access protocol the verifier receives both parts of response that verifies the first challenge C_1. It is easy for the oracle to provide $g^{u\rho_j}, g^{v_j}, g^{uv_j}, h_j$ and $g^{\rho'_j}$, using its input values. This however is not possible for the second response, which the verifier will accept only if it matches with the value g^{v_j}, which is authenticated through h_j. Since the advantage of $\mathcal{A}^{auth}$ is non-negligible, we assume that the oracle's output contains the correct value $R_2 = g^{v_j r_s}$. Now $\mathcal{A}^{DCDH}$ can use $\mathcal{A}^{auth}$ as a subroutine to break DCDH assumption. The adversary $\mathcal{A}^{DCDH}$ gives to the oracle $\mathcal{O}^{auth}$ the values $C_2 = g^{uv_j r_s}$ (as $X = g^x$), g^u (as $Y = g^y$) and the oracle outputs $R_2 = g^{v_j r_s} \equiv g^{x/y}$. This however contradicts the DCDH assumption. It is easy to see that non-transferability also implies unforgeability and user-coalition resistance.

5 Related Work

Anonymous authentication is an extensively studied field and can be categorized in two main frameworks. A first line of works is based on Brands [4] and Chaum's blind signatures [10], and has been implemented in Microsoft's U-Prove technology [20]. Credentials of this category are inherently one-show (*i.e.*, linkable when used more than once), however they are suitable in cases where a single credential needs to be traced or when a credential can only be used once. The second line of works is based on the framework of Camenish and Lysyanskaya (the CL framework) [7,8], proposed in [2] in the standard model and extended in [19,22], while a variation of the technology is implemented as the Idemix system [16]. Credentials of this category are multi-show with built-in unlinkability. Schemes of this category are inherently less efficient than the Brands' framework.

Both of these categories of work are focusing on anonymous authentication systems with a much wider scope than our scheme, which focuses on anonymity of mobile users from LBS services. The first scheme for mobile anonymity that makes use of the network operator as the credential issuer is the lightweight scheme of [23]. This scheme is efficient, since it transforms the RSA-based direct anonymous attestation scheme [5] to an elliptic curve scheme and pairings. The scheme of [23] only requires 5 scalar multiplications, which is the computation-intensive operation in their setting. Our scheme is more expensive but is feasible for mid-range modern mobile devices, since it requires 12 exponentiations for each anonymous access (including both credential issuing and user access). However, in the scheme of [23] credential non-transferability is based on the existence of embedded hardware, which is not required in our scheme. Moreover, our scheme improves system fairness, by providing user non-frameability.

Another view of anonymity in LBS is privacy-preserving access control for LBS services. Two frameworks can be considered in this area: (a) TTP-based schemes, which adopt a centralized model for privacy in LBSs, where online and/or offline TTPs are employed for either protecting the location information of users (*i.e.*, TTP spatial k-anonymity [13], TTP cloaking/obfuscation (*e.g.*, [15])), or for protecting the link between location information and user identity (*i.e.*, identity privacy with simple pseudonyms [14] or multiple, unlinkable pseudonyms [17,23]). (b) TTP-free solutions: Here trust assumptions are very weak or completely removed. The category contains *client-server* architectures based on the (inefficient) PIR cryptographic primitive (*e.g.*, [12]), where communication takes place between a user and an untrusted LBS provider, as well as fully-distributed or *collaborative* settings (*e.g.*, [21]), where trust is distributed among a set of system peers that form ad-hoc networks and collaborate to achieve privacy against a set of untrusted entities (*i.e.*, the LBS provider, and/or mobile peers or even the network operator). The main problems in both frameworks are the strong assumptions made by most TTP-based schemes and the high computation and communication costs of TTP-free schemes [18].

6 Conclusions

In this paper we proposed an efficient, secure and fair anonymous authentication scheme for mobile devices accessing LBS services. As future work, we plan to transform our scheme into a formal and general-use, fair anonymous authentication scheme, which will provide non-frameability and other fundamental properties in an efficient manner. We also plan to build a system prototype, and empirically measure the efficiency of our scheme.

References

1. Bao, F., Deng, R.H., Zhu, H.: Variations of Diffie-Hellman Problem. In: Qing, S., Gollmann, D., Zhou, J. (eds.) ICICS 2003. LNCS, vol. 2836, pp. 301–312. Springer, Heidelberg (2003)
2. Belenkiy, M., Chase, M., Kohlweiss, M., Lysyanskaya, A.: P-signatures and Noninteractive Anonymous Credentials. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 356–374. Springer, Heidelberg (2008)
3. Bethencourt, J.: Cryptographic Techniques for Privacy Preserving Identity. Ph.D. thesis, EECS Department, University of California, Berkeley (May 2011), http://www.eecs.berkeley.edu/Pubs/TechRpts/2011/EECS-2011-58.html
4. Brands, S.A.: Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy. MIT Press, Cambridge (2000)
5. Brickell, E.F., Camenisch, J., Chen, L.: Direct anonymous attestation. In: Atluri, V., Pfitzmann, B., McDaniel, P.D. (eds.) ACM Conference on Computer and Communications Security, pp. 132–145. ACM (2004)
6. Burmester, M., Desmedt, Y., Wright, R.N., Yasinsac, A.: Accountable Privacy. In: Christianson, B., Crispo, B., Malcolm, J.A., Roe, M. (eds.) Security Protocols 2004. LNCS, vol. 3957, pp. 83–95. Springer, Heidelberg (2006)

7. Camenisch, J., Lysyanskaya, A.: An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 93–118. Springer, Heidelberg (2001)
8. Camenisch, J., Lysyanskaya, A.: Signature Schemes and Anonymous Credentials from Bilinear Maps. In: Franklin, M.K. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 56–72. Springer, Heidelberg (2004)
9. Camenisch, J., Neven, G.: Saving On-Line Privacy. In: Bezzi, M., Duquenoy, P., Fischer-Hübner, S., Hansen, M., Zhang, G. (eds.) IFIP AICT 320. IFIP AICT, vol. 320, pp. 34–47. Springer, Heidelberg (2010)
10. Chaum, D.: Blind signatures for untraceable payments. In: Chaum, D., Rivest, R.L., Sherman, A.T. (eds.) CRYPTO 1982, pp. 199–203. Plenum Press (1982)
11. Fiat, A., Shamir, A.: How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1987)
12. Ghinita, G., Kalnis, P., Khoshgozaran, A., Shahabi, C., Tan, K.L.: Private queries in location based services: anonymizers are not necessary. In: Wang, J.T.L. (ed.) SIGMOD Conference, pp. 121–132. ACM (2008)
13. Gruteser, M., Grunwald, D.: Anonymous usage of location-based services through spatial and temporal cloaking. In: MobiSys 2003, pp. 31–42. ACM, New York (2003)
14. Hauser, C., Kabatnik, M.: Towards privacy support in a global location service. In: IFIP Workshop on IP and ATM Traffic Management, WATM/EUNICE 2001 (2001)
15. Hengartner, U.: Location privacy based on trusted computing and secure logging. In: SecureComm 2008: Proceedings of the 4th International Conference on Security and Privacy in Communication Networks, pp. 1–8. ACM (2008)
16. IBM: IDentity Mixer - Idemix, http://www.zurich.ibm.com/~pbi/identityMixer_gettingStarted/, (accessed March 18, 2012)
17. Kölsch, T., Fritsch, L., Kohlweiss, M., Kesdogan, D.: Privacy for Profitable Location Based Services. In: Hutter, D., Ullmann, M. (eds.) SPC 2005. LNCS, vol. 3450, pp. 164–178. Springer, Heidelberg (2005)
18. Magkos, E.: A survey of cryptographic approaches for privacy preservation in location-based services. International Journal of Information Technologies and the Systems Approach (IJITSA) 4(2), 48–69 (2011)
19. Nguyen, L., Safavi-Naini, R.: Dynamic k-Times Anonymous Authentication. In: Ioannidis, J., Keromytis, A.D., Yung, M. (eds.) ACNS 2005. LNCS, vol. 3531, pp. 318–333. Springer, Heidelberg (2005)
20. Paquin, C., Thompson, G.: U-prove ctp white paper. Microsoft Corporation (2010)
21. Solanas, A., Martínez-Ballesté, A.: Privacy Protection in Location-Based Services Through a Public-Key Privacy Homomorphism. In: López, J., Samarati, P., Ferrer, J.L. (eds.) EuroPKI 2007. LNCS, vol. 4582, pp. 362–368. Springer, Heidelberg (2007)
22. Teranishi, I., Sako, K.: k-Times Anonymous Authentication with a Constant Proving Cost. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T. (eds.) PKC 2006. LNCS, vol. 3958, pp. 525–542. Springer, Heidelberg (2006)
23. Wachsmann, C., Chen, L., Dietrich, K., Löhr, H., Sadeghi, A.-R., Winter, J.: Lightweight Anonymous Authentication with TLS and DAA for Embedded Mobile Devices. In: Burmester, M., Tsudik, G., Magliveras, S., Ilić, I. (eds.) ISC 2010. LNCS, vol. 6531, pp. 84–98. Springer, Heidelberg (2011)