Lightweight Anonymous Authentication with TLS and DAA for Embedded Mobile Devices

Full Version

Liqun Chen[1], Kurt Dietrich[2], Hans Löhr[3], Ahmad-Reza Sadeghi[3],
Christian Wachsmann[3], and Johannes Winter[2]

[1] Hewlett Packard Labs, Bristol, UK
liqun.chen@hp.com

[2] Institute for Applied Information Processing and Communication,
Graz University of Technology, Austria
{kurt.dietrich,johannes.winter}@iaik.tugraz.at

[3] Horst Görtz Institute for IT-Security (HGI), Ruhr-University Bochum, Germany
{hans.loehr,ahmad.sadeghi,christian.wachsmann}@trust.rub.de

Abstract. Although anonymous authentication has been extensively studied, so far no scheme has been widely adopted in practice. A particular issue with fully anonymous authentication schemes is that users cannot easily be prevented from copying and sharing credentials.

In this paper, we propose an anonymous authentication scheme for mobile devices that prevents copying and sharing of credentials based on hardware security features. Our system is an optimized adaptation of an existing direct anonymous attestation (DAA) scheme, specifically designed for resource-constrained mobile devices. Our solution provides (i) anonymity and untraceability of mobile embedded devices against service providers, (ii) secure device authentication even against collusions of malicious service providers, and (iii) allows for revocation of authentication credentials. We present a new cryptographic scheme with a proof of security, as well as an implementation on ARM TrustZone. Moreover, we evaluate the efficiency of our approach and demonstrate its suitability for mobile devices.

Key words: Mobile Phones, Privacy, Anonymity, ARM TrustZone.

1 Introduction

Modern mobile phones are powerful computing devices that are connected to various online services operated by different providers. Most currently available smartphones are equipped with reasonably sized displays and innovative user interfaces, which turns them into mobile internet access points. Moreover, integrated GPS receivers enable location-based services like turn-by-turn navigation (e.g., [22,25,16]) or online recommendation services (e.g., Loopt [19] recommends restaurants based on the user's location, CitySense [24] shows the most crowded places of cities, and Google Latitude [17] locates a user's friends). However, besides the benefits of these applications, they pose a serious threat to user privacy since they disclose both the identity and the location of a user not only to the mobile network operator, but also to content service providers. This allows service providers to create detailed user profiles that, for instance, can be misused to discriminate against specific classes of users. Hence, users are interested in protecting any information that allows them to be tracked. In particular, the service provider should not be able to identify or trace the transactions of a specific user. However, intuitively, this seems to be in conflict with the requirement that only authorized (i.e., paying) users may access the services.

In this context, anonymous credential systems (see, e.g., [13,10,9]) enable the authentication of users without disclosing their identity. In an anonymous credential system, users authenticate themselves by proving the possession of credentials from a credential issuer. However, without additional countermeasures, users could copy and share their credentials, such that unauthorized users can access services. Moreover, a service provider cannot detect if a credential has been copied, and credentials cannot be revoked. Therefore, security measures are necessary to protect authentication secrets. Existing approaches include pseudonyms (which might enable profiling), all-or-nothing sharing [10] (which assumes that all users possess a valuable secret that can be embedded in the credential), and hardware security features (see, e.g., [18]). Although the applicability of hardware-based solutions is limited to devices that support the required security features, we consider this approach to be most viable for practice. In particular, current and upcoming generations of mobile devices are usually equipped with hardware security mechanisms, e.g., Texas Instruments M-Shield [4], ARM TrustZone [2], or Mobile Trusted Modules [27]. In this paper we use existing hardware security features of common mobile devices to protect the authentication secrets and to prevent credential sharing.

Direct anonymous attestation (DAA) [9] is an anonymous credential scheme that has been specified by the Trusted Computing Group (TCG) [28] for platforms with a dedicated security chip, the Trusted Platform Module (TPM) [26]. The computation of the DAA protocols is split between the resource-constrained TPM chip and the software running on the platform, which typically is a PC. For mobile devices, the TCG specifies a Mobile Trusted Module (MTM) [27] with optional support for DAA. However, DAA is complex and computationally intensive, and thus not suitable for the hardware protection mechanisms of mobile embedded devices such as smartcards, SIM-cards, or special-purpose processor extensions with very limited computational power and memory capacity.

Contribution. In this paper, we present a lightweight anonymous authentication scheme for mobile embedded devices and its implementation on a common mobile hardware platform. Our scheme is tailored to the resource constraints and widely available hardware security architectures of mobile platforms (e.g., Texas Instruments M-Shield [4], ARM TrustZone [2], Mobile Trusted Modules [27]). Our solution is a new anonymous authentication scheme that optimizes and adapts the DAA scheme of [14] for the anonymous authentication of mobile platforms. Our protocol has several appealing features that are important for practical applications: (i) it enables devices to authenticate to verifiers (e.g., service providers) without revealing any information that allows identifying or tracking a device. Hence, even malicious verifiers cannot link the transactions of a device. (ii) our protocol ensures that even adversaries that can corrupt verifiers cannot impersonate legitimate devices to honest verifiers. (iii) our scheme supports revocation of authentication credentials via revocation lists.

We give a formal security model for anonymous authentication of mobile devices and prove that our scheme is secure in the random oracle model under the decisional Diffie-Hellman (DDH), the discrete logarithm (DL), and the bilinear LRSW assumption, which is a discrete-logarithm-based assumption introduced in [20] that is typical for pairing-based anonymous credential systems.

Further, we provide a prototype implementation of our anonymous authentication scheme and evaluate its performance. Our prototype is based on ARM TrustZone and performs all security critical computations in a shielded environment that is protected from unauthorized access by arbitrary user applications. To the best of our knowledge, this constitutes the first implementation of a pairing-based variant of DAA on an ARM TrustZone-based platform.

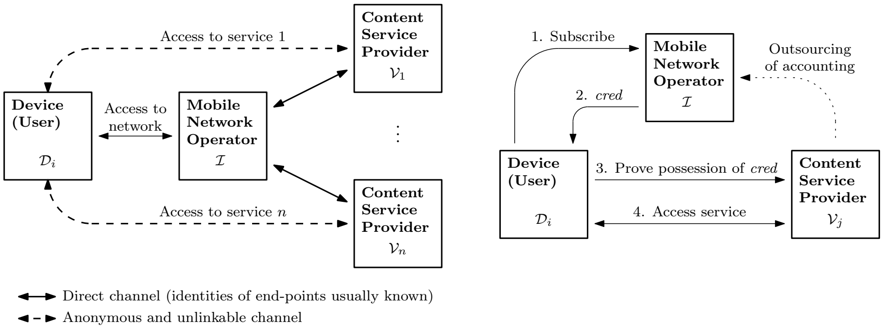
We integrate our anonymous authentication protocol into a standard Internet protocol: Transport Layer Security (TLS). For this, we use RFC-compliant TLS hello extensions and supplemental data messages, similar to the proposal in [12].

Outline. First, we introduce anonymous authentication of mobile devices in Section 2. Then we present our scheme in Section 3, together with a proof of security in Section 4. In Section 5, we describe our implementation and evaluate its performance. Finally, we discuss related work in Section 6 and conclude in Section 7.

2 Anonymous Authentication with Mobile Devices

A typical mobile network consists of a mobile network operator $\mathcal{I}$, different independent content service providers $\mathcal{V}_j$, and a set of mobile devices $\mathcal{D}_i$ owned by the users of the mobile network (see Figure 1a). Each mobile device $\mathcal{D}_i$ is connected to the network infrastructure of $\mathcal{I}$. In addition to the basic services (e.g., telephony) provided by $\mathcal{I}$, users can access (location-based) services offered by the service providers $\mathcal{V}_j$. Since the services provided by $\mathcal{V}_j$ usually are subject to charge, users (i.e., their devices $\mathcal{D}_i$) must authenticate to $\mathcal{V}_j$ before they are given access to services. However, location-based services allow the service provider to learn both the identity and the location of a user's device $\mathcal{D}_i$ at a given time, which enables the creation of user profiles. Hence, users are interested in hiding their identities from $\mathcal{V}_j$, resulting in seemingly contradicting requirements.

Our solution is an anonymous authentication scheme that works as follows (see Figure 1b): the service provider $\mathcal{V}_j$ outsources the accounting and billing to the network operator $\mathcal{I}$, e.g., subscription fees of $\mathcal{V}_j$ are accounted with the user's telephone bill issued by $\mathcal{I}$. To access some service, $\mathcal{D}_i$ first subscribes for


Access to service 1
Content Service Provider $\mathcal{V}_1$
Device (User) $\mathcal{D}_i$
Access to network
Mobile Network Operator $\mathcal{I}$
Access to service n
Content Service Provider $\mathcal{V}_n$
Direct channel (identities of end-points usually known)
Anonymous and unlinkable channel
1. Subscribe
Mobile Network Operator $\mathcal{I}$
Outsourcing of accounting
2. *cred*
Device (User) $\mathcal{D}_i$
3. Prove possession of *cred*
4. Access service
Content Service Provider $\mathcal{V}_j$

(a) Communication model

(b) Use case

Fig. 1. Anonymous authentication in mobile networks

the service at $\mathcal{I}$ (step 1), which issues an anonymous credential *cred* for $\mathcal{D}_i$ on behalf of $\mathcal{V}_j$ (step 2). From now on, $\mathcal{D}_i$ can use *cred* to anonymously authenticate to $\mathcal{V}_j$ (step 3) to get access to the services provided by $\mathcal{V}_j$ (step 4).

Hereby, $\mathcal{I}$ must be trusted by both $\mathcal{V}_j$ and $\mathcal{D}_i$. Note that $\mathcal{D}_i$ must trust $\mathcal{I}$ in any case since $\mathcal{I}$ usually can identify and trace devices due to the technology of todays mobile network infrastructures.[4] Note that the trust relation among $\mathcal{I}$ and $\mathcal{V}_j$ can be established by legal means. Moreover, $\mathcal{I}$ is interested in providing access to the services of many different service providers to its users. Hence, in practice $\mathcal{I}$ will do nothing that violates the security objectives of $\mathcal{V}_j$, since this could damage $\mathcal{I}$'s reputation with a large number of service providers, which would then refuse to cooperate with $\mathcal{I}$, thus damaging $\mathcal{I}$'s business model.

The resulting requirements to an anonymous authentication scheme for mobile devices can be summarized as follows:

- *Correctness:* Users with valid credentials must be able to (anonymously) authenticate to the service provider.
- *Unforgeability:* Users must not be able to forge an authentication, i.e., they must not be able to authenticate without having obtained a valid credential.
- *Unclonability:* Valid credentials cannot be copied (cloned).
- *Unlinkability:* Sessions must be unlinkable (also called full anonymity).
- *Revokability:* It must be possible to revoke users.
- *Practicability:* All protocols should be efficient and based on well-established standards. Moreover, the implementation should be fast and based on widely used soft- and hardware.

[4] Subscribers in GSM- and UMTS-based cellular networks are uniquely identifiable by their International Mobile Subscriber Identity (IMSI) and can be located based on the radio cell they are currently connected to.

3 Our Lightweight Anonymous Authentication Scheme

Before presenting the details of our protocol, we first give an informal description of our protocol, the underlying trust relations and assumptions, and introduce our notation. We formalize the relevant security aspects in Section 4.

3.1 Protocol Overview

The players in our scheme are (at least) a credential issuer $\mathcal{I}$ (e.g., the mobile network operator), a set of verifiers $\mathcal{V}_j$ (e.g., service providers), and a set of mobile devices $\mathcal{D}_i$. Our anonymous authentication scheme is a three party protocol that is executed between a verifier $\mathcal{V}_j$ and a device $\mathcal{D}_i$ that is composed of a (semi-trusted) host $\mathcal{H}_i$ (e.g., the operating system of $\mathcal{D}_i$), and a secure component $\mathcal{S}_i$ (e.g., MTM [27], Texas Instruments M-Shield [4], ARM TrustZone [2]) as depicted in Figure 2. The goal of our protocol is to authenticate $\mathcal{D}_i$ to $\mathcal{V}_j$ such that $\mathcal{V}_j$ only learns whether $\mathcal{D}_i$ is legitimate without obtaining any information that allows $\mathcal{V}_j$ to identify or trace $\mathcal{D}_i$. $\mathcal{D}_i$ is called *legitimate* if it has been initialized by $\mathcal{I}$. The main idea of the protocol is to split the computations to be performed by $\mathcal{D}_i$ in the authentication protocol between a secure component $\mathcal{S}_i$, where all security critical operations are performed, and the (semi-trusted) host $\mathcal{H}_i$ of $\mathcal{D}_i$ that performs all privacy-related computations.[5]

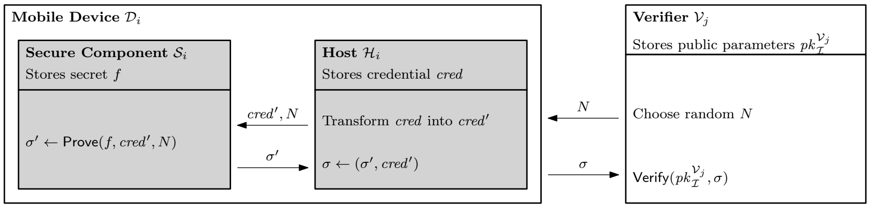

Mobile Device $\mathcal{D}_i$
Secure Component $\mathcal{S}_i$
Stores secret f
$\sigma' \leftarrow \mathsf{Prove}(f, cred', N)$
$cred', N$
σ'
Host $\mathcal{H}_i$
Stores credential $cred$
Transform $cred$ into $cred'$
$\sigma \leftarrow (\sigma', cred')$
N
σ
Verifier $\mathcal{V}_j$
Stores public parameters $pk_{\mathcal{I}}^{\mathcal{V}_j}$
Choose random N
$\mathsf{Verify}(pk_{\mathcal{I}}^{\mathcal{V}_j}, \sigma)$

Fig. 2. Protocol overview

Each device $\mathcal{D}_i$ is initialized by $\mathcal{I}$ with a specific signing key f and a corresponding anonymous credential *cred*.[6] In the anonymous authentication protocol, $\mathcal{V}_j$ challenges $\mathcal{D}_i$ to sign a random message N. $\mathcal{D}_i$ returns a signature σ on N, which can be verified w.r.t. *cred* using the public key $pk_{\mathcal{I}}^{\mathcal{V}_j}$ associated with verifier $\mathcal{V}_j$.[7] If the verification succeeds, $\mathcal{V}_j$ has assurance that σ has been created by a device that has been initialized by $\mathcal{I}$. Therefore, the structure of σ ensures that (i) only $\mathcal{I}$ can create a valid *cred* for any secret f on behalf of $\mathcal{V}_j$, (ii) only a device that has been initialized by $\mathcal{I}$ can create a valid σ that can

[5] In case the secure component has sufficient computing power and memory capabilities, then all computations can also be performed by the secure component.

[6] A device may have a set of credentials to authenticate to different service providers.

[7] Note that $\mathcal{I}$ must use a different $(sk_{\mathcal{I}}^{\mathcal{V}_j}, pk_{\mathcal{I}}^{\mathcal{V}_j})$ for each service provider $\mathcal{V}_j$ to prevent interleaving attacks against the anonymous authentication protocol.

be verified w.r.t. $pk_{\mathcal{I}}^{\mathcal{V}_j}$, and (iii) $\mathcal{V}_j$ does not learn any information that allows $\mathcal{V}_j$ to deduce the identity of $\mathcal{D}_i$. Since *cred* must be included in each signature σ issued by $\mathcal{D}_i$, *cred* could be used as an identifier of $\mathcal{D}_i$. This would allow for linking all signatures σ created by $\mathcal{D}_i$ and thus tracking $\mathcal{D}_i$. Hence, to provide untraceability, it is crucial that each signature σ issued by $\mathcal{D}_i$ contains a different *cred*. Therefore, the construction of *cred* allows to transform (*re-randomize*) *cred* into different anonymous credentials $cred_1, cred_2, \ldots$ for the same secret f without knowing the secret key of $\mathcal{I}$. Since the secure component in most currently available mobile platforms often has only limited computational and memory capabilities, our solution allows for outsourcing the privacy-related computations to a (semi-trusted) host, e.g., the operating system of the mobile device. Note that $\mathcal{H}_i$ usually controls the communication of $\mathcal{D}_i$ and hence must be trusted not to disclose any information that allows tracking of $\mathcal{D}_i$.

In our scheme, we adapt the anonymous credential system proposed in [14], which is very promising w.r.t. upcoming embedded mobile platforms equipped with a secure component since it (i) allows for splitting the signature creation process and (ii) it is based on elliptic curve cryptography, which allows for short parameter sizes. We adapted the scheme of [14] for our purposes by removing support for user-controlled anonymity, which leads to a simpler and more efficient construction. This means that our protocol always ensures the unlinkability of all signatures issued by a user's device, whereas the scheme in [14] allows the user to decide to what extend signatures can be linked. Moreover, we consider the case where the mobile network operator acts as credential issuer, thus disposing the need of providing anonymity against the issuer since the network operator cannot be prevented from tracing the user. Instead, we focus on unlinkability of users against service providers.

The main security objective of our protocol is anonymous authentication. More precisely, $\mathcal{V}_j$ should *only* accept legitimate devices *without* being able to link their transactions (anonymity and unlinkability of devices against verifiers).

3.2 Trust Model and Assumptions

We assume the adversary $\mathcal{A}$ to control the communication between devices $\mathcal{D}_i$ and verifiers $\mathcal{V}_j$. This means that $\mathcal{A}$ can eavesdrop, manipulate, delete and reroute all protocol messages sent by $\mathcal{D}_i$ and $\mathcal{V}_j$. Moreover, $\mathcal{A}$ can obtain useful information on whether $\mathcal{V}_j$ accepted $\mathcal{D}_i$ as a legitimate device, e.g., by observing whether the connection between $\mathcal{D}_i$ and $\mathcal{V}_j$ aborts or not. The issuer $\mathcal{I}$ and $\mathcal{V}_j$ are assumed to be trusted. Moreover, we assume that $\mathcal{I}$ initializes all $\mathcal{D}_i$ and $\mathcal{V}_j$ in a secure environment. Each $\mathcal{D}_i$ is equipped with a secure component $\mathcal{S}_i$ that cannot be compromised by $\mathcal{A}$. This means that $\mathcal{A}$ cannot eavesdrop or tamper any data stored and any computation performed by $\mathcal{S}_i$. Besides $\mathcal{S}_i$, each $\mathcal{D}_i$ is equipped with a host $\mathcal{H}_i$ that can be compromised by $\mathcal{A}$, e.g., by viruses or Trojans. Hence, $\mathcal{A}$ can access any information stored by $\mathcal{H}_i$ and controls any computation performed by a compromised $\mathcal{H}_i$.

3.3 Notation and Preliminaries

For a finite set S, $|S|$ denotes the size of set S whereas for an integer or bitstring n the term $|n|$ means the bit-length of n. The term $s \in_R S$ means the assignment of a uniformly chosen element of S to variable s. Let A be a probabilistic algorithm. Then $y \leftarrow \mathsf{A}(x)$ means that on input x, algorithm A assigns its output to variable y. The term $[\mathsf{A}(x)]$ denotes the set of all possible outputs of A on input x. $\mathsf{A}_K(x)$ means that the output of A depends on x and some additional parameter K (e.g., a secret key). Let E be some event (e.g., the result of a security experiment), then $\Pr[E]$ denotes the probability that E occurs. Probability $\epsilon(l)$ is called *negligible* if for all polynomials f it holds that $\epsilon(l) \leq 1/f(l)$ for all sufficiently large l. Probability $1 - \epsilon(l)$ is called *overwhelming* if $\epsilon(l)$ is negligible.

Definition 1 (Admissible Pairing). *Let $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ be three groups of large prime exponent $q \approx 2^{l_q}$ for security parameter $l_q \in \mathbb{N}$. The groups $\mathbb{G}_1, \mathbb{G}_2$ are written additively with identity element 0 and the group $\mathbb{G}_T$ multiplicatively with identity element 1. A pairing is a mapping $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ that is*

1. *bilinear: for all $P, P' \in \mathbb{G}_1$ and all $Q, Q' \in \mathbb{G}_2$ it holds that*

$$e(P + P', Q + Q') = e(P, Q) \cdot e(P, Q') \cdot e(P', Q) \cdot e(P', Q') \,.$$

2. *non-degenerate: for all $P \in \mathbb{G}_1^*$ there is a $Q \in \mathbb{G}_2^*$ (and for all $Q \in \mathbb{G}_2^*$ there is a $P \in \mathbb{G}_1^*$, respectively) such that $e(P, Q) \neq 1$.*
3. *efficiently computable: there is a probabilistic polynomial time (p.p.t.) algorithm that computes $e(P, Q)$ for all $(P, Q) \in \mathbb{G}_1 \times \mathbb{G}_2$.*

Let P_1 be a generator of $\mathbb{G}_1$ and P_2 be a generator of $\mathbb{G}_2$. A pairing e is called admissible if $e(P_1, P_2)$ is a generator of $\mathbb{G}_T$.

We denote with $\mathsf{GenPair}(1^{l_q}) \rightarrow (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, P_1, P_2, e)$ an algorithm that on input a security parameter $l_q \in \mathbb{N}$ generates three groups $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ of large prime exponent $q \approx 2^{l_q}$, two generators $\langle P_1 \rangle = \mathbb{G}_1$ and $\langle P_2 \rangle = \mathbb{G}_2$, and an admissible pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$.

3.4 Protocol Specification

System initialization: Given a security parameter $l = (l_q, l_h, l_e, l_n) \in \mathbb{N}^4$, the issuer $\mathcal{I}$ generates the secret key $sk_\mathcal{I}$ and the corresponding public parameters $pk_\mathcal{I}^\mathcal{V}$ associated with verifier $\mathcal{V}$ and the revocation list $\mathtt{RL}$. $\mathcal{I}$ generates $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, P_1, P_2, e) \leftarrow \mathsf{GenPair}(1^{l_q})$, chooses two secrets $x, y \in_R \mathbb{Z}_q$, and computes $X \leftarrow xP_2$ and $Y \leftarrow yP_2$ in $\mathbb{G}_2$. Then, $\mathcal{I}$ chooses a collision-resistant one-way hash function $\mathsf{Hash} : \{0,1\}^* \rightarrow \{0,1\}^{l_h}$ and initializes the revocation list $\mathtt{RL} \leftarrow \emptyset$. The secret key of $\mathcal{I}$ is $sk_\mathcal{I} \leftarrow (x, y)$ while the public parameters are $pk_\mathcal{I}^\mathcal{V} \leftarrow (l, q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, P_1, P_2, e, X, Y, \mathsf{Hash})$ and $\mathtt{RL}$.

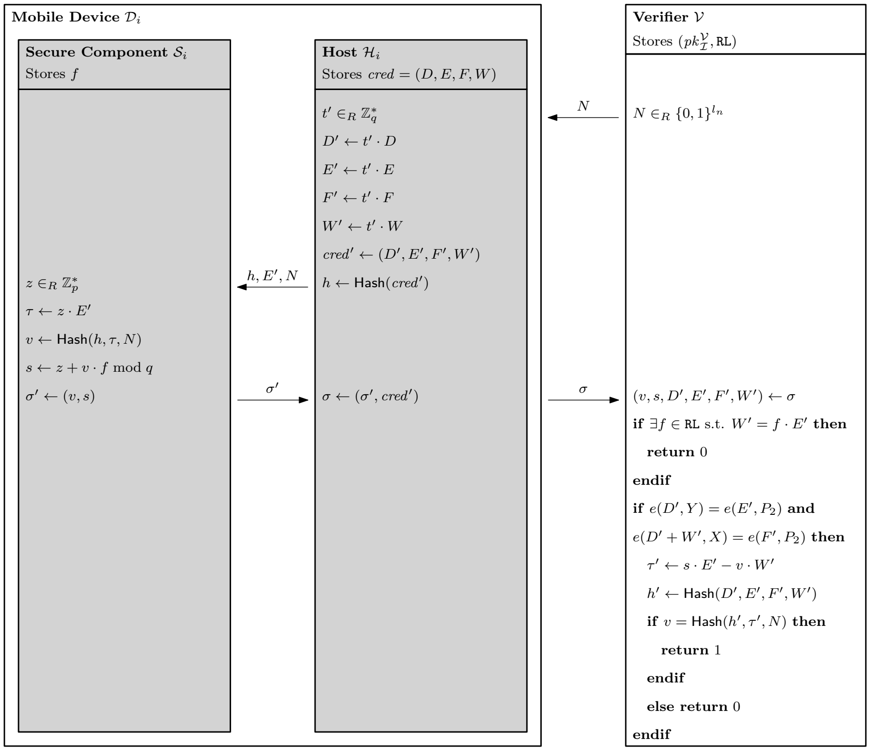

Mobile Device $\mathcal{D}_i$
Secure Component $\mathcal{S}_i$
Stores f
Host $\mathcal{H}_i$
Stores $cred = (D, E, F, W)$
Verifier $\mathcal{V}$
Stores $(pk_{\mathcal{I}}^{\mathcal{V}}, \mathtt{RL})$
N
$N \in_R \{0,1\}^{l_n}$
$t' \in_R \mathbb{Z}_q^*$
$D' \leftarrow t' \cdot D$
$E' \leftarrow t' \cdot E$
$F' \leftarrow t' \cdot F$
$W' \leftarrow t' \cdot W$
$cred' \leftarrow (D', E', F', W')$
$h \leftarrow \mathsf{Hash}(cred')$
h, E', N
$z \in_R \mathbb{Z}_p^*$
$\tau \leftarrow z \cdot E'$
$v \leftarrow \mathsf{Hash}(h, \tau, N)$
$s \leftarrow z + v \cdot f \bmod q$
$\sigma' \leftarrow (v, s)$
σ'
$\sigma \leftarrow (\sigma', cred')$
σ
$(v, s, D', E', F', W') \leftarrow \sigma$
if $\exists f \in \mathtt{RL}$ s.t. $W' = f \cdot E'$ **then**
return 0
endif
if $e(D', Y) = e(E', P_2)$ **and**
$e(D' + W', X) = e(F', P_2)$ **then**
$\tau' \leftarrow s \cdot E' - v \cdot W'$
$h' \leftarrow \mathsf{Hash}(D', E', F', W')$
if $v = \mathsf{Hash}(h', \tau', N)$ **then**
return 1
endif
else return 0
endif

Fig. 3. Anonymous authentication protocol

Device initialization: $\mathcal{I}$ generates a secret signing key f and a corresponding anonymous credential $cred = (D, E, F, W)$ on behalf of $\mathcal{V}$. Therefore, $\mathcal{I}$ chooses $f, r \in_R \mathbb{Z}_q$ and computes $D \leftarrow rP_1$, $E \leftarrow yD$, $F \leftarrow (x + xyf)D$, and $W \leftarrow fE$. Finally, the mobile device $\mathcal{D}_i$ is initialized with f and $cred$. Hereby, f is securely stored in and must never leave the secure component $\mathcal{S}_i$, whereas the credential $cred$ can be stored in and used by the (semi-trusted) host $\mathcal{H}_i$ of $\mathcal{D}_i$.

Anonymous authentication: In the anonymous authentication protocol, a mobile device $\mathcal{D}_i$ anonymously authenticates to a verifier $\mathcal{V}$ as shown in Figure 3 Hereby, $\mathcal{V}$ challenges $\mathcal{D}_i$ to sign a random challenge N. Upon receipt of N, $\mathcal{H}_i$ randomizes the credential $cred$ to $cred'$. Next, $\mathcal{H}_i$ passes to the secure component $\mathcal{S}_i$ the hash digest h of $cred'$, the value E' of $cred'$, and N. $\mathcal{S}_i$ then computes a signature of knowledge σ' in a similar way as in [14] and returns σ' to $\mathcal{H}_i$. Next, $\mathcal{H}_i$ composes the final anonymous signature $\sigma \leftarrow (\sigma', cred')$ and sends it to $\mathcal{V}$. Upon receipt of σ, $\mathcal{V}$ verifies that (i) $cred'$ has not been revoked, (ii) $cred'$ is a valid (randomized) credential w.r.t. $pk_{\mathcal{I}}^{\mathcal{V}}$, and (iii) σ' is a valid signature of knowledge on N w.r.t. $cred'$ and $pk_{\mathcal{I}}^{\mathcal{V}}$. If the verification is successful, then $\mathcal{V}$ accepts $\mathcal{D}_i$ as a legitimate device and returns 1. Otherwise $\mathcal{V}$ rejects $\mathcal{D}_i$ and returns 0.

Device revocation: To revoke a device $\mathcal{D}_i$, $\mathcal{I}$ adds the authentication secret f of $\mathcal{D}_i$ to the revocation list $\texttt{RL}$, and sends the updated revocation list $\texttt{RL}$ to $\mathcal{V}$ using an authentic channel.

4 Security Analysis

For our security proofs, we need the following intractability assumptions:

Definition 2 (Bilinear LRSW Assumption [14]). *Let $l \in \mathbb{N}$ be a security parameter, $pk_e \leftarrow \mathsf{GenPair}(1^l)$, where $pk_e = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, P_1, P_2, e)$, $x, y \in_R \mathbb{Z}_q$, $X \leftarrow xP_2$, and $Y \leftarrow yP_2$. Moreover, let $\mathcal{O}_{x,y}$ be an oracle that on input $f \in \mathbb{Z}_q$ outputs a triple $\big(D, yD, (x + fxy)D\big)$ where $D \in_R \mathbb{G}_1$. Let Q be the set of oracle queries made to $\mathcal{O}_{x,y}$. The bilinear LRSW assumption is that for every p.p.t. adversary $\mathcal{A}$ and every $(f, D, E, F) \in \big[\mathcal{A}^{\mathcal{O}_{x,y}}(pk_e, X, Y)\big]$ it holds that*

$$\Pr\big[f \in \mathbb{Z}_q^* \wedge f \notin Q \wedge D \in \mathbb{G}_1 \wedge E = yD \wedge F = (x + fxy)D\big]$$

is negligible in l.

Definition 3 (Gap-DL Assumption [14]). *Let $l \in \mathbb{N}$ be a security parameter, $pk_e \leftarrow \mathsf{GenPair}(1^l)$, where $pk_e = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, P_1, P_2, e)$, $x \in_R \mathbb{Z}_q$, and $X \leftarrow xP_1$. Moreover, let $\mathcal{O}_x$ be an oracle that on input $Y \in \mathbb{G}_1$ outputs xY. The Gap-DL assumption is that for every p.p.t. adversary $\mathcal{A}$ and every $x' \in \big[\mathcal{A}^{\mathcal{O}_x}(pk_e, X)\big]$ it holds that $\Pr\big[x' = x\big]$ is negligible in l.*

Definition 4 (DDH Assumption [14]). *Let $l \in \mathbb{N}$ be a security parameter, $pk_e \leftarrow \mathsf{GenPair}(1^l)$, where $pk_e = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, P_1, P_2, e)$, $x, y \in_R \mathbb{Z}_q$, $X \leftarrow xP_1$, $Y \leftarrow yP_1$, and $Z \in_R \mathbb{G}_1$. The decisional Diffie-Hellman assumption in $\mathbb{G}_1$ is that every p.p.t. adversary $\mathcal{A}$ has negligible (in l) advantage*

$$\mathbf{Adv}_{\mathcal{A}}^{\mathrm{DDH}} = \big|\Pr\big[1 \leftarrow \mathcal{A}(pk_e, X, Y, xyP_1)\big] - \Pr\big[1 \leftarrow \mathcal{A}(pk_e, X, Y, Z)\big]\big| .$$

Now we formally define and prove device authentication and unlinkability.

4.1 Device Authentication

Device authentication means that adversary $\mathcal{A}$ should not be able to make an honest verifier $\mathcal{V}$ to accept $\mathcal{A}$ as some legitimate device $\mathcal{D}_i$. We formalize device authentication by a security experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathrm{aut}} = \texttt{out}_{\mathcal{V}}^{\pi}$, where a p.p.t. adversary $\mathcal{A}$ must make an honest $\mathcal{V}$ to authenticate $\mathcal{A}$ as a legitimate $\mathcal{D}_i$ by returning $\texttt{out}_{\mathcal{V}}^{\pi} = 1$ in some instance π of the anonymous authentication protocol. Hereby, $\mathcal{A}$ can arbitrarily interact with $\mathcal{V}$, $\mathcal{I}$, and all $\mathcal{D}_i$. However, since in general it is not possible to prevent simple relay attacks, $\mathcal{A}$ is not allowed to just forward all messages from $\mathcal{D}_i$ to $\mathcal{V}$ in instance π. Hence, at least some of the protocol messages that made $\mathcal{V}$ accept must have been (partly) computed by $\mathcal{A}$ without knowing $\mathcal{D}_i$'s secrets.

Definition 5 (Device Authentication). *An anonymous authentication scheme achieves device authentication if for every p.p.t. adversary* $\mathcal{A}$ $\Pr[\mathbf{Exp}_{\mathcal{A}}^{\mathrm{aut}} = 1]$ *is negligible in* l*.*

Theorem 1. *The anonymous authentication scheme described in Section 3.4 achieves device authentication (Definition 5) in the random oracle model under the bilinear LRSW (Definition 2) and the Gap-DL assumption (Definition 3).*

The detailed proof of Theorem 1 can be found in Appendix A.

Proof (Theorem 1, Sketch). Assume by contradiction that $\mathcal{A}$ is an adversary s.t. $\Pr[\mathbf{Exp}_{\mathcal{A}}^{\mathrm{aut}} = 1]$ is non-negligible. We show that if such an $\mathcal{A}$ exists, then $\mathcal{A}$ either violates the bilinear LRSW assumption (Definition 2), the Gap-DL assumption (Definition 3), or the collision-resistance of the underlying hash function.

Note that $\mathbf{Exp}_{\mathcal{A}}^{\mathrm{aut}} = 1$ implies that, for a given verifier challenge N, $\mathcal{A}$ successfully computed a signature $\sigma = (v, s, D, E, F, W)$ such that $e(D, Y) = e(E, P_2)$, $e(D + W, X) = e(F, P_2)$ and $v = \mathsf{Hash}(h, \tau, N)$, where $h = \mathsf{Hash}(D, E, F, W)$ and $\tau = sE - vW$. Hereby, $\mathcal{A}$ has two possibilities: (i) reuse a credential $\mathit{cred}' = (D', E', F', W')$ from a previous device authentication protocol-run, or (ii) create a new credential cred''. We show that if $\mathcal{A}$ is successful in the first case, then $\mathcal{A}$ can either be used (a) to find a collision of Hash for v, which contradicts the assumption that Hash is a random oracle, or (b) to slove the Gap-DL problem since computing a valid signature of knowledge s for a new N implies that $\mathcal{A}$ knows f, which violates the Gap-DL assumption (Definition 3). Moreover, if $\mathcal{A}$ is successful in the second case, then $\mathcal{A}$ violates the bilinear LRSW assumption (Definition 2) by computing a valid cred''. Hence, the random oracle property of Hash, the Gap-DL assumption, and the bilinear LRSW assumption ensure that $\Pr[\mathbf{Exp}_{\mathcal{A}}^{\mathrm{aut}} = 1]$ is negligible. □

4.2 Unlinkability of Devices

Unlinkability means that an adversary $\mathcal{A}$ cannot distinguish devices $\mathcal{D}_i$ based on their communication.[8] This means that the protocol messages generated by $\mathcal{D}_i$ should not leak any information to $\mathcal{A}$ that allows $\mathcal{A}$ to identify or trace $\mathcal{D}_i$. We formalize device authentication by a security experiment $\mathbf{Exp}_{\mathcal{A}}^{\text{prv-b}} = b'$ for $b \in_R \{0, 1\}$, where a p.p.t. adversary $\mathcal{A}$ interacts with an oracle $\mathcal{O}_b$ that either represents two identical ($b = 0$) or two different ($b = 1$) legitimate devices $\mathcal{D}_0$ and $\mathcal{D}_1$. Hereby, $\mathcal{A}$ can arbitrarily interact with $\mathcal{V}$, $\mathcal{I}$, all $\mathcal{D}_i$ for $i \notin \{0, 1\}$, and $\mathcal{O}_b$. Finally $\mathcal{A}$ returns a bit b' to indicate interaction with $\mathcal{O}_{b'}$.

Definition 6. *An anonymous authentication scheme achieves unlinkability if every probabilistic polynomial time (p.p.t.) adversary* $\mathcal{A}$ *has negligible (in* l*) advantage* $\mathbf{Adv}_{\mathcal{A}}^{\mathrm{prv}} = \left|\Pr\left[\mathbf{Exp}_{\mathcal{A}}^{\text{prv-0}} = 1\right] - \Pr\left[\mathbf{Exp}_{\mathcal{A}}^{\text{prv-1}} = 1\right]\right|$.

[8] Note that unlinkability implies anonymity since an adversary who can identify devices can also trace them.

Theorem 2. *The anonymous authentication scheme described in Section 3.4 achieves unlikability (Definition 6) in the random oracle model under the decisional Diffie-Hellman assumption in $\mathbb{G}_1$ (Definition 4).*

The detailed proof of Theorem 2 can be found in Appendix B.

Proof (Theorem 2, Sketch). We show that if $\mathcal{A}$ can distinguish whether $\mathcal{O}_b$ represents two identical or two different devices, i.e., if $\mathcal{A}$ has non-negligible advantage $\mathbf{Adv}_{\mathcal{A}}^{\mathrm{prv}}$, then $\mathcal{A}$ can be used to break the DDH assumption in $\mathbb{G}_1$ (Definition 4).

With $\sigma\big[f, \mathit{cred}(f)\big]$ we denote a signature σ that has been generated by $\mathcal{O}_b$ using the secret signing key f and the credential $\mathit{cred}(f)$ on f. Let f_0 be the signing key of $\mathcal{D}_0$ and f_1 be the signing key of $\mathcal{D}_1$ and let (D_i, E_i, F_i, W_i) be the credential used to compute signature σ_i. Note that both $\mathcal{D}_0$ and $\mathcal{D}_1$ are simulated by $\mathcal{O}_b$. We show that the distributions $\Delta = \big\langle \sigma_0\big[f_0, \mathit{cred}(f_0)\big], \sigma_1\big[f_0, \mathit{cred}(f_0)\big]\big\rangle$ and $\Delta' = \big\langle \sigma_2\big[f_0, \mathit{cred}(f_0)\big], \sigma_3\big[f_1, \mathit{cred}(f_1)\big]\big\rangle$ are computationally indistinguishable. More precisely, we show that for all signatures in Δ it holds that $(F_0, D_1, F_1) = (\alpha D_0, \gamma D_0, \alpha\gamma D_0)$ is a DDH-tuple for $\alpha = x + xyf_0$ and some $\gamma \in \mathbb{Z}$, while this is *not* true for the signatures in Δ'. Hence, if $\mathcal{A}$ can distinguish between Δ and Δ' with non-negligible advantage $\mathbf{Adv}_{\mathcal{A}}^{\mathrm{prv}}$, then $\mathcal{A}$ can be used to construct an algorithm $\mathcal{A}_{\mathrm{DDH}}$ that violates the DDH assumption in $\mathbb{G}_1$ (Definition 4). □

5 Architecture and Implementation

Our anonymous authentication scheme requires a hardware-protected environment that ensures confidentiality of the secret key. This can be achieved by different means, depending on the hardware used.

5.1 Security Extensions for Embedded CPUs

Two different security extensions for mobile embedded systems exist. The most interesting one for smartphone CPUs is ARM TrustZone [2], which allows the partitioning of the memory and the CPU of a device into two virtual domains: the so-called *secure-world* and the *normal-world*. While untrusted applications (e.g., user applications) are executed in the normal-world, security critical code is executed in the secure-world. The information flow between both the secure and the normal-world is controlled by a *secure monitor*, which is controlled by the secure-world operating system. Another security extension is Texas Instruments M-Shield [4] which is similar and binary compatible with ARM TrustZone. Both security extensions provide a secure execution environment (SEE) that can only be accessed by trusted applications.

5.2 Integration of our Scheme into Transport Layer Security (TLS)

The TLS protocol [8] defines a mechanism to authenticate clients to servers and works as follows: the server sends a list with accepted certificate authorities

(CAs) to the client. The client then transmits its certificate and the information of which CA has issued this certificate to the server. Now the server can validate the authenticity of the client using the client certificate. However, the standard TLS client authentication mechanism allows the server to identify the client.

From the technical point of view, there are two solutions to achieve anonymous client authentication with TLS: (i) instead of using a conventional (e.g., RSA or ECDSA) signature within the client authentication process, anonymous signature can be used. (ii) the signature that actually authenticates the client to the server is generated with an ephemeral (e.g., RSA or ECDSA) signing key that is certified (i.e., signed) with an anonymous signature. In our proof-of-concept implementation, we follow the first approach and directly use the anonymous signature for authentication. Moreover, we modified the TLS messages (e.g., `ClientHello`, `ServerHello`, `CertificateRequest`) as described in [12] to transport the anonymous credential and the client's signature to the server. Furthermore, the CA certificate selected by the client contains the issuer's public-key that indicates the group affiliation of the client.

5.3 Overview of the Implementation

In our proof-of-concept implementation, we split the computations between a *secure component* and the *host* (see Figure 2). As shown in Figure 4, the secure component is located in the ARM TrustZone secure-world environment, while the host is represented by the (semi-trusted) operating system of the device located in the normal-world. Hereby, the secure monitor is used to exchange information between the secure-world and the normal-world. The software for computing the point multiplications and modular reductions is available in both the secure- and the normal-world.

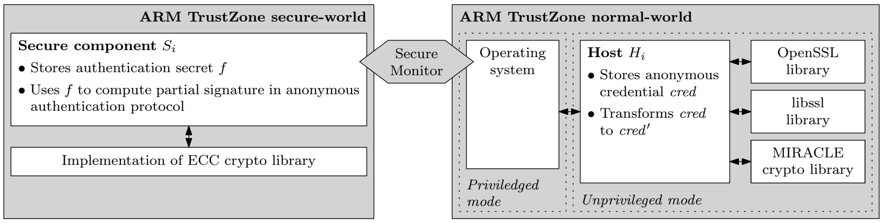

ARM TrustZone secure-world
Secure component S_i
• Stores authentication secret f
• Uses f to compute partial signature in anonymous authentication protocol
Implementation of ECC crypto library
Secure Monitor
ARM TrustZone normal-world
Operating system
Priviledged mode
Host H_i
• Stores anonymous credential *cred*
• Transforms *cred* to *cred'*
Unprivileged mode
OpenSSL library
libssl library
MIRACLE crypto library

Fig. 4. Splitting the anonymous authentication on ARM TrustZone

5.4 Performance Evaluation

We measured the performance of the implementation of our anonymous authentication scheme on a development platform equipped with a Texas Instruments OMAP 3530 ARM11 CPU, which runs at a clock speed of 600 MHz. Table 1 shows the average performance taken over 100 test-runs on the development platform. The column *host* refers to the computations performed in the normal-world and the column *secure component* denotes the computations performed by the TrustZone secure-world. The second row shows the performance of our implementation when the compiler is instructed to produce highly optimized binaries for ARM CPUs (ARM-Thumb code [3]).

Table 1. Performance of the DAA Sign Protocol.

DAA sign	Host	Secure component	Total
ARM	94.98 ms	23.75 ms	118.75 ms
ARM Thumb	92.57 ms	23.16 ms	115.73 ms

To the best of our knowledge, the only other implementation of a DAA-based protocol on mobile phones has been presented by Dietrich [15]. However, a direct comparison of our solution to this approach is difficult for two reasons: first, Dietrich implemented the original RSA-based DAA scheme [9] on an ARM11 device, while our scheme uses elliptic curves and pairings. Second, Dietrich's implementation is using Java that in general is much slower then C, which has been used for our implementation.

In order to support a large number of embedded devices and applications, we based our implementation on the widespread OpenSSL13 v1.0 libcrypto and libssl libraries. Moreover, we strongly use the architecture discussed in [12] to integrate our implementation into the OpenSSL security framework. For the cryptographic operations, we use the MIRACLE [9] crypto library.

6 Related Work

Anonymous authentication has been extensively studied in scientific literature (see, e.g., [13,23,21,18]). Since their introduction by Chaum [13], various anonymous credential systems have been proposed. For this paper, the Camenisch-Lysyanskaya (CL) credential system [10] is of particular importance since it is the basis for most DAA schemes. Several variants of CL credentials exist, which are either based on the strong RSA assumption [10] or pairings over elliptic curves [11]. Recently, a credential system called Idemix that is based on CL credentials has been implemented within the PRIME project [6,1]. Compared to Idemix, we employ hardware security features to prevent credential sharing. Further, our implementation uses more efficient pairing-based protocols than the Idemix implementation, which is based on the strong RSA assumption. Moreover, Idemix requires its protocols to be executed over a secure channel, e.g., a TLS connection, whereas our solution explicitly combines TLS with anonymous authentication. On the other hand, the objectives of PRIME and Idemix are set in a much wider scope than plain anonymous authentication, which is the topic

[9] Multiprecision Integer and Rational Arithmetic C/C++ Library (`www.shamus.ie`).

of this paper. Cesena et al. [12] combine TLS with DAA and present an implementation on a standard PC. In contrast, we propose a protocol specifically suited to the capabilities and the hardware security features of mobile embedded devices and provide a formal security analysis. Bichsel et al. [7] present an implementation of CL credentials that uses a JavaCard as hardware security module, providing portable credentials and multi-application support. This solution prevents credential sharing, provided the JavaCard is secure. However, users need additional hardware (i.e., a JavaCard and a card reader), whereas our solution uses ARM TrustZone that is available on many recent smartphones. Batina et al. [5] implement a scheme for blinded attribute certificates based on pairings and ECDSA on a JavaCard. However, the security properties of this scheme are unclear since the authors do not provide a formal security analysis. Moreover, this scheme has no built-in support for revocation.

7 Conclusion

In this paper, we proposed an anonymous authentication scheme for mobile devices that prevents copying of credentials based on the hardware security features of modern mobile platforms. Our scheme relies on pairings and features revocation of credentials. Moreover, we introduced a formal security model for anonymous authentication of mobile devices and proved the security of our scheme within this model. We presented the first implementation of a pairing-based variant of direct anonymous attestation (DAA) on a mobile phone equipped with ARM TrustZone. Furthermore, we integrate our scheme into TLS, using RFC-compliant TLS extensions and supplemental data messages. Our performance evaluations show that our protocol is efficient on current mobile hardware.

Acknowledgements. We thank Frederik Armknecht for several useful discussions and the anonymous reviewers for their helpful comments. This work has been supported in part by the European Commission through the FP7 programme under contract 216646 ECRYPT II, 238811 UNIQUE, and 257433 SEPIA.

References

1. Ardagna, C., Camenisch, J., Kohlweiss, M., Leenes, R., Neven, G., Priem, B., Samarati, P., Sommer, D., Verdicchio, M.: Exploiting cryptography for privacy-enhanced access control: A result of the PRIME project. Journal of Computer Security 18, 123–160 (2010)
2. ARM: TrustZone website. `http://www.arm.com/products/security/trustzone/` (September 2009)
3. ARM, Ltd.: Instruction set architectures. ARM White Paper (Feb 2008), `http://www.arm.com/products/processors/technologies/instruction-set-architectures.php`
4. Azema, J., Fayad, G.: M-Shield$^{\text{TM}}$ mobile security technology: Making wireless secure. Texas Instruments White Paper (February 2008), `http://focus.ti.com/pdfs/wtbu/ti_mshield_whitepaper.pdf`

5. Batina, L., Hoepman, J.H., Jacobs, B., Mostowski, W., Vullers, P.: Developing efficient blinded attribute certificates on smart cards via pairings. In: CARDIS 2010. LNCS, vol. 6035, pp. 209–222. Springer Verlag (2010)
6. Bichsel, P., Binding, C., Camenisch, J., Groß, T., Heydt-Benjamin, T., Sommer, D., Zaverucha, G.: Cryptographic protocols of the identity mixer library. Tech. Rep. RZ 3730 (#99740), IBM Research (2009)
7. Bichsel, P., Camenisch, J., Groß, T., Shoup, V.: Anonymous credentials on a standard Java Card. In: Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS'09). ACM Press (2009)
8. Blake-Wilson, S., Nystrom, M., Hopwood, D., Mikkelsen, J., Wright, T.: Transport layer security (TLS) extensions (2003)
9. Brickell, E., Camenisch, J., Chen, L.: Direct Anonymous Attestation. In: Proceedings of the 11th ACM Conference on Computer and Communications Security (CCS'04). pp. 132–145. ACM Press (2004)
10. Camenisch, J., Lysyanskaya, A.: An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In: Advances in Cryptology – EUROCRYPT 2001. LNCS, vol. 2045, pp. 93–118. Springer (2001)
11. Camenisch, J., Lysyanskaya, A.: Signature schemes and anonymous credentials from bilinear maps. In: Advances in Cryptology – CRYPTO 2004. LNCS, vol. 3152, pp. 56–72. Springer (2004)
12. Cesena, E., Löhr, H., Ramunno, G., Sadeghi, A.R., Vernizzi, D.: Anonymous authentication with TLS and DAA. In: Proceedings of the Third International Conference on Trust and Trustworthy Computing (TRUST'10). LNCS, vol. 6101, pp. 47–62. Springer Verlag (2010)
13. Chaum, D.: Security without identification: Transaction systems to make big brother obsolete. Communications of the ACM 28(10), 1030–1044 (1985)
14. Chen, L., Page, D., Smart, N.P.: On the design and implementation of an efficient DAA scheme. In: 9th IFIP WG 8.8/11.2 International Conference on Smart Card Research and Advanced Application (CARDIS'10), Proceedings. LNCS, vol. 6035, pp. 223–237. Springer Verlag (2010)
15. Dietrich, K.: Anonymous credentials for Java enabled platforms. In: Chen, L., Yung, M. (eds.) INTRUST 2009. pp. p. 101 – 116 (2009)
16. Google, Inc.: Google Maps Navigation. http://www.google.com/mobile/navigation/
17. Google, Inc.: Google Latitude. http://www.google.com/latitude (June 2010)
18. Lindell, A.Y.: Anonymous authentication. Aladdin Knowledge Systems Inc., http://www.aladdin.com/blog/pdf/AnonymousAuthentication.pdf (2006)
19. Loopt: Loopt website. http://www.loopt.com/ (June 2010)
20. Lysyanskaya, A., Rivest, R.L., Sahai, A., Wolf, S.: Pseudonym systems. In: 6th Annual International Workshop, SAC'99, Kingston, Ontario, Canada, August 9–10, 1999, Proceedings. LNCS, vol. 1758, pp. 184–199. Springer Verlag (1999)
21. Nguyen, L., Safavi-Naini, R.: Dynamic k-times anonymous authentication. In: 3rd International Conference on Applied Cryptography and Network Security (ACNS'05). LNCS, vol. 3531, pp. 318–333. Springer (2005)
22. Nokia: OviMaps website. http://maps.ovi.com/ (June 2010)
23. Schechter, S., Parnell, T., Hartemink, A.: Anonymous authentication of membership in dynamic groups. In: Third International Conference on Financial Cryptography (FC'99). LNCS, vol. 1648, pp. 184–195. Springer (1999)
24. Sense Networks, Inc.: CitySense. http://www.citysense.com/ (June 2010)
25. TomTom: TomTom website. http://www.tomtom.com/ (June 2010)

26. Trusted Computing Group: TCG TPM Specification, Version 1.2, Revision 103. `http://www.trustedcomputinggroup.org/` (July 2007)
27. Trusted Computing Group: TCG MTM Specification, Version 1.0, Revision 6. `http://www.trustedcomputinggroup.org/` (June 2008)
28. Trusted Computing Group: TCG website. `https://www.trustedcomputinggroup.org` (June 2010)

A Proof of Theorem 1

In the following, we prove that the anonymous authentication scheme described in Section 3.4 achieves device authentication (Definition 5) in the random oracle model under the Bilinear LRSW (Definition 2) and the Gap-DL Assumption (Definition 3).

Proof. Assume by contradiction that $\mathcal{A}$ is an adversary s.t. $\Pr[\mathbf{Exp}_{\mathcal{A}}^{\mathrm{aut}} = 1]$ is non-negligible. We show that if such an $\mathcal{A}$ exists, then $\mathcal{A}$ either violates the Bilinear LRSW Assumption (Definition 2), the Gap-DL Assumption (Definition 3), or the collision-resistance of the underlying hash function.

Note that $\mathbf{Exp}_{\mathcal{A}}^{\mathrm{aut}} = 1$ implies that, for a given verifier challenge N, $\mathcal{A}$ computed a signature $\sigma = (v, s, D, E, F, W)$ such that $e(D, Y) = e(E, P_2)$, $e(D+W, X) = e(F, P_2)$ and $v = \mathsf{Hash}(h, \tau, N)$, where $h = \mathsf{Hash}(D, E, F, W)$ and $\tau = sE - vW$. Hereby, $\mathcal{A}$ has two possibilities: (i) reuse a credential (D, E, F, W) from a previous device authentication protocol-run, or (ii) create a new (forged) credential (D, E, F, W). We show that if $\mathcal{A}$ is successful in the first case, then $\mathcal{A}$ can either be used (i) to find a collision of Hash, which contradicts the assumption that Hash is a random oracle, or (ii) to slove the Gap-DL problem, which violates the Gap-DL Assumption (Definition 3). Moreover, if $\mathcal{A}$ is successful in the second case, then $\mathcal{A}$ violates the Bilinear LRSW Assumption (Definition 2). Hence, the random oracle property of Hash, the Gap-DL Assumption, and the Bilinear LRSW Assumption ensure that $\Pr[\mathbf{Exp}_{\mathcal{A}}^{\mathrm{aut}} = 1]$ is negligible.

Case 1: *$\mathcal{A}$ reuses old credential.* Assume by contradiction that $\mathcal{A}$ uses a randomized version of a credential $\mathit{cred}' = (D', E', F', W')$ from a previous transcript $\big(N', (v', s', \mathit{cred}')\big)$ of the device authentication protocol to forge a signature (v, s) on a new verifier challenge N. Note that $\Pr[N = N']$ is negligible since N is uniformly chosen at random in each execution of the device authentication protocol. Hence, if $\mathcal{V}$ accepts an old signature (v', s') for a new challenge N, then with overwhelming probability $v' = \mathsf{Hash}(h', \tau', N') = \mathsf{Hash}(h', \tau', N)$ such that $N \neq N'$. This means that $\mathcal{A}$ found a collision of Hash. However, since Hash is assumed to be collision-resistant, this can only happen with negligible probability. Therefore, $\mathcal{A}$ must have computed a new signature of knowledge (v, s) such that $v = \mathsf{Hash}(h', \tau, N)$ and $s = z + v \cdot f \bmod q$, where $\tau = zE$. Note that (v, s) includes a proof of knowledge of a value f such that $e(D' + f \cdot E', X) = e(F', P_2)$, which is a standard Σ-protocol for proving knowledge of a discrete logarithm. It follows from the proof-of-knowledge property that, if $\mathcal{A}$ can compute a valid (v, s), then there is a p.p.t. algorithm (knowledge extractor) that can extract

f from $\mathcal{A}$. This implies that $\mathcal{A}$ knows f. Since the Gap-DL Assumption (Definition 3) ensures that $\mathcal{A}$ can compute f from $W' = fE'$ only with negligible probability, the probability that $\mathcal{A}$ knows f is negligible. Hence, the proof-of-knowledge property and the Gap-DL Assumption ensure that $\mathcal{A}$ can forge a signature (v, s) on a given message N for a given credential cred' only with negligible probability.

Case 2: *$\mathcal{A}$ creates new credential.* Assume that $\mathcal{A}$ can construct a new signature $\sigma = (v, s, \mathit{cred})$ where $\mathit{cred} = (\tilde{D}, \tilde{E}, \tilde{F}, \tilde{W})$ is *not* a randomized version of a credential from a previous device authentication protocol. In the following, we show that $\mathcal{A}$ can be used to construct an adversary $\mathcal{A}_{\text{bLRSW}}$ against the bilinear LRSW assumption (Definition 2). Given access to oracle $\mathcal{O}_{x,y}$ and the public parameters $\mathit{pk}_{\text{bLRSW}} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, P_1, P_2, e, X, Y)$, $\mathcal{A}_{\text{bLRSW}}$ simulates the initialization algorithm Init of the anonymous authentication system to $\mathcal{A}$ as specified in Section 3.4 but uses $\mathit{pk}_{\text{bLRSW}}$ to construct $\mathit{pk}^{\mathcal{V}}_{\mathcal{I}}$. Note that $\mathcal{A}_{\text{bLRSW}}$ does *not* know the secret parameters (x, y) of the simulation of the anonymous authentication scheme, which are required for the simulation of the device initialization algorithm. However, $\mathcal{A}_{\text{bLRSW}}$ can simulate the device initialization algorithm with the help of $\mathcal{O}_{x,y}$: instead of using (x, y) to compute the credential (D, E, F, W) for the device to be initialized, $\mathcal{A}_{\text{bLRSW}}$ chooses $f \in_R \mathbb{Z}_q$ and queries $\mathcal{O}_{x,y}(f)$, which responds with a tuple $\big(D, yD, (x + fxy)D\big)$. Note that by definition of $\mathcal{O}_{x,y}$ it holds that $D \in \mathbb{G}_1$, which means that D can be expressed as $D = rP_1$ for an unknown $r \in \mathbb{Z}_q$. Further, $\mathcal{A}_{\text{bLRSW}}$ computes $W \leftarrow f \cdot (yD)$. Therefore, $\mathcal{O}_{x,y}$ can be used to construct a valid credential (D, E, F, W) and hence, the simulation of the device initialization algorithm by $\mathcal{A}_{\text{bLRSW}}$ is perfect. Moreover, $\mathcal{A}_{\text{bLRSW}}$ can perfectly simulate all other algorithms and protocols of the anonymous authentication system since they do not require knowledge of (x, y) or r. Thus, after a polynomial number of queries to $\mathcal{A}_{\text{bLRSW}}$, $\mathcal{A}$ returns a new signature $\sigma = (v, s, \tilde{D}, \tilde{E}, \tilde{F}, \tilde{W})$ for a given N that makes $\mathcal{V}$ to accept $\mathcal{A}_{\text{bLRSW}}$ as a legitimate device. Since (v, s) includes a proof of knowledge of a value $\tilde{f}$ such that $e(\tilde{D} + \tilde{f} \cdot \tilde{E}, X) = e(\tilde{F}, P_2)$, $\mathcal{A}_{\text{bLRSW}}$ can use the corresponding knowledge extractor to extract $\tilde{f}$ from $\mathcal{A}$. Finally, $\mathcal{A}_{\text{bLRSW}}$ returns a tuple $(\tilde{f}, \tilde{D}, \tilde{E}, \tilde{F})$. Since $\mathit{cred} = (\tilde{D}, \tilde{E}, \tilde{F}, \tilde{W})$ is not a randomized version of a credential from a previous device authentication protocol, it holds that $\mathcal{O}_{x,y}$ has never been queried for the corresponding secret $\tilde{f}$. Hence, $(\tilde{f}, \tilde{D}, \tilde{E}, \tilde{F})$ represents a valid solution to the bilinear LRSW problem, which is a contradiction to the bilinear LRSW assumption (Definition 2). In turn this means that the bilinear LRSW assumption ensures that $\mathcal{A}$ has negligible advantage in generating a valid signature $\sigma = (v, s, \mathit{cred})$ for a given message N that is not based on an existing credential. □

B Proof of Theorem 2

In the following, we prove that the anonymous authentication scheme described in Section 3.4 achieves unlikability (Definition 6) in the random oracle model under the Decisional Diffie-Hellman Assumption in $\mathbb{G}_1$ (Definition 4).

Proof (Theorem 2, Sketch). Recall that unlinkability (Definition 6) requires that $\mathcal{A}$ cannot distinguish whether $\mathcal{O}_b$ represents two identical or two different devices. We show that if $\mathcal{A}$ has non-negligible advantage $\mathbf{Adv}_{\mathcal{A}}^{\mathrm{prv}}$, then $\mathcal{A}$ can be used to break the DDH-Assumption in $\mathbb{G}_1$ (Definition 4).

With $\sigma\big[f, \mathit{cred}(f)\big]$ we denote a signature σ that has been generated by $\mathcal{O}_b$ using the secret signing key f and the credential $\mathit{cred}(f)$ on signing key f. Let f_0 be the signing key of $\mathcal{D}_0$ and f_1 be the signing key of $\mathcal{D}_1$. Note that both $\mathcal{D}_0$ and $\mathcal{D}_1$ are simulated by $\mathcal{O}_b$. In the following, we show that the distributions $\mathit{\Delta} = \big\langle \sigma_0\big[f_0, \mathit{cred}(f_0)\big], \sigma_1\big[f_0, \mathit{cred}(f_0)\big]\big\rangle$ and $\mathit{\Delta}' = \big\langle \sigma_2\big[f_0, \mathit{cred}(f_0)\big], \sigma_3\big[f_1, \mathit{cred}(f_1)\big]\big\rangle$ are computationally indistinguishable. More precisely, we show that if $\mathcal{A}$ can distinguish between $\mathit{\Delta}$ and $\mathit{\Delta}'$ with non-negligible advantage $\mathbf{Adv}_{\mathcal{A}}^{\mathrm{prv}}$, then $\mathcal{A}$ can be used to construct an algorithm $\mathcal{A}_{\mathrm{DDH}}$ that violates the DDH-Assumption in $\mathbb{G}_1$ (Definition 4).

Let (D_i, E_i, F_i, W_i) be the credential used to compute signature σ_i. Note that all credentials (D_i, E_i, F_i, W_i) for $i \in \{0, 1, 2\}$ are randomizations of the credential $\mathit{cred}(f_0)$. Hence, $F_i = \alpha D_i$ for $i \in \{0, 1\}$ and $\alpha = x + xyf_0$. Moreover, for all signatures in $\mathit{\Delta}$ there is a $\gamma \in \mathbb{Z}$ such that $D_1 = \gamma D_0$. Similarly, all credentials (D_3, E_3, F_3, W_3) are randomized versions of $\mathit{cred}(f_1)$ and $F_3 = \alpha' D_3$ for $\alpha' = x + xyf_1$. Further, for all signatures in $\mathit{\Delta}'$ there is a $\gamma' \in \mathbb{Z}$ such that $D_3 = \gamma' D_2$. Note that for all signatures in $\mathit{\Delta}$ it holds that $(F_0, D_1, F_1) = (\alpha D_0, \gamma D_0, \alpha\gamma D_0)$ is a DDH-tuple, while this is *not* true for the signatures in $\mathit{\Delta}'$, i.e., in general $(F_2, D_3, F_3) \neq (\alpha D_2, \gamma' D_2, \alpha'\gamma' D_2)$. However, the DDH-Assumption in $\mathbb{G}_1$ (Definition 4) ensures that both distributions $\mathit{\Delta}$ and $\mathit{\Delta}'$ are computationally indistinguishable. Hence, $\mathcal{A}$ cannot link devices based on their communication (i.e., their signatures) of the anonymous authentication protocol. □