Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 3295148, 12 pages
https://doi.org/10.1155/2018/3295148

Research Article

Multidevice Authentication with Strong Privacy Protection

Jan Hajny, Petr Dzurenda, and Lukas Malina

Brno University of Technology, Czech Republic

Correspondence should be addressed to Jan Hajny; hajny@feec.vutbr.cz

Received 12 April 2018; Revised 20 June 2018; Accepted 3 July 2018; Published 29 July 2018

Academic Editor: Kim-Kwang Raymond Choo

Card-based physical access control systems are used by most people on a daily basis, for example, at work, in public transportation, or at hotels. Yet these systems have often very poor cryptographic protection. User identifiers and keys can be easily eavesdropped on and counterfeited. The privacy-preserving features are almost missing in these systems. To improve this state, we propose a novel cryptographic scheme based on efficient zero-knowledge proofs and Boneh-Boyen signatures. The proposed scheme is provably secure and provides the full set of privacy-enhancing features, that is, the anonymity, untraceability, and unlinkability of users. Furthermore, our scheme supports distributed multidevice authentication with multiple RFID (Radio-Frequency IDentification) user devices. This feature is particularly important in applications for controlling access to dangerous sites where the presence of protective equipment is checked during each access control session. Besides the full cryptographic specification, we also show the results of our implementation on devices commonly used in access control applications, particularly the smart cards and embedded verification terminals. By avoiding costly operations on user devices, such as bilinear pairings, we were able to achieve times comparable to existing systems (around 500 ms), while providing significantly higher security, privacy protection, and features for RFID multidevice authentication.

1. Introduction

Privacy-enhancing technologies constitute a significant part of contemporary cryptography. Modern cryptographic protocols allow privacy-enhanced storing of sensitive data and its processing by cloud services, private information retrieval, or, for example, authentication based on personal attributes, instead of user identifiers. The increasing intensity of research into privacy is supported by national programs and strategies, in particular in USA [1] and EU [2]. While most of the novel schemes are aimed at electronic services, the domain of physical access control is rather neglected. We still use traditional locks, tourniquets, and classical card-based access control mechanisms to manage physical access to our premises. But with the increasing computational power of the programmable smart cards, massive expansion of various personal electronic devices, and the capabilities in RFID communication of our smart phones, we can expect penetration of privacy-enhancing technologies also to the area of physical access control. In particular, in mass applications like public transportation, e-ticketing, e-passports, and eIDs, the benefits of controlling physical access using electronic devices with advanced cryptographic protocols are very appealing.

In this paper, we propose and experimentally evaluate a novel cryptographic scheme that particularly addresses two phenomenons of contemporary cyberspace: lack of user privacy and ubiquitous presence of many personal devices (phones, smart cards, RFID tags, bluetooth dongles, smart watch, etc.) that can be leveraged for stronger authentication and more reliable access control.

In particular, we focus on safety applications in which the users wear multiple safety equipment, such as helmets, harnesses, boots, and protective suits, each with attached programmable RFID tag capable of wireless communication. A user is granted access to (potentially dangerous) premises only if all his equipment is present. In existing systems, the presence of the protective equipment is checked simply by scanning the identifiers using RFID readers. Such an approach is neither secure (identifiers can be counterfeited), nor privacy friendly (identifiers can be traced, behavioral profiles can be created, etc.).

We propose a novel cryptographic scheme for multidevice authentication that is tailored for physical access control systems where the user must prove not only his own identifier, but also *many other auxiliary identifiers* stored on separate devices. In addition, the authentication sessions must support all the key privacy-enhancing features; i.e., the access control process must be *anonymous* (i.e., a user must prove that he belongs to a group of authorized users, but without releasing his concrete identity), *unlinkable* (all the sessions of a single user cannot be linkable to a profile), and *untraceable* (system administrators must be unable to trace honest users in the system). On the other side, the scheme must provide efficient means for revocation and identification of malicious users. In our cryptographic system, we provide all the required features that are often contradictory and completely unavailable in existing schemes (in particular, the presence of many identifiers versus anonymity; the untraceability and strong cryptographic security versus efficiency on RFID tags and stickers).

In our scheme, users can be granted access to premises upon proving the presence of particular devices in their proximity (e.g., the safety equipment) or personal attributes (age, membership, citizenship, etc.). The access control process may (the extent of privacy-enhancing features can be initially set by the administrator; if required, identification or user tracing may be enforced by the access control system) proceed in a fully private manner, without disclosing user identity or being traceable in the system.

2. Related Work

Most of the existing practical physical access control systems are based on the following technologies: NXP's Mifare and DESfire; HID's Prox and iClass; and Legic Prime and Advant. NXP's Mifare Classic, introduced in 1994, is a very popular technology used in physical access control systems. Although very old and insecure, the technology is still used in many applications, even those security sensitive. The authentication protocol is based on a unique 4B card identifier UID. In some implementations, the card just reveals UID to the terminal without any authentication protocol. In that case, UID can be easily eavesdropped on and used by an attacker for impersonation. In other implementations, a simple authentication protocol is used but is considered insecure due to many existing practical attacks [3–5] on the encryption algorithm CRYPTO1. The insufficient security of the CRYPTO1 algorithm used in the Mifare Classic made NXP improve the cryptographic protection and release Mifare DESFire. The old encryption algorithm was replaced by 3DES algorithm. The authentication protocol was further improved in Mifare DESFire EV1 which supports the AES encryption algorithm [6]. The protocol itself remained without any major changes. However, even Mifare DESFire was successfully attacked, although the attacks [7, 8] were aimed on the implementation, not cryptographic weaknesses. The HID Prox technology contains no cryptographic protection. HID iClass employs an authentication protocol based on the 3DES algorithm, but attacks on this protocol are available [9]. Legic Prime has weak proprietary cryptographic protection [10]. Legic Advant is protected by symmetric block algorithms (DES [11], 3DES, and AES). None of the major commercial technologies provide any protection of privacy.

With the introduction of the first attribute-based credential schemes, such as the Idemix [12], U-Prove [13], and HM12 [14], the variants for physical access control systems also started to appear. The U-Prove scheme was implemented on Multos smart cards [15]. The user is able to prove his attribute in less than 1 s using this implementation. However, the unlinkability property cannot be provided by the cryptographic design of the protocol. The Idemix was also implemented on the Multos smart card platform, with cca 1 s needed to generate the attribute proof. The pilot implementation of the HM12 scheme using Multos ML3 smart cards [16] required around 2.4 s in total to generate and verify the proof, including the communication overhead. No testing was done on multiple devices because the distributed proof is not supported by these schemes.

Many types of personal and wearable devices forming the so-called Internet of Things (IoT) have appeared recently. Authentication issues have been solved by different techniques on these devises. Xu and Weitao propose biometric authentication using wearables with face recognition using smart-glass and gait recognition using smart watch. Riva et al. [17] combine multiple sources of authentication data, which is close to our approach. However, all these schemes are using mainly biometric authentication factors. Cha et al. [18] present a simple model for two-device authentication for micropayment systems using mobile and wearable devices. Nevertheless, their proposal lacks details and concrete cryptographic functions. Butun et al. [19] address multilevel authentication issue in cloud computing. Gonzalez-Manzano et al. [20] present an access control mechanism for cloud-based storage service access by using a set of devices. However, their scheme is based on symmetric cryptography and thus does not provide nonrepudiation. Hajny et al. [21] use many wearable and IoT devices to do the authentication process. However, the scheme misses privacy-enhancing properties, because each user is uniquely represented by his/her public key.

In summary, there are several authentication solutions that involve IoT devices. However, there are only very few papers focusing on multidevice authentication. Currently, none of the proposals is provably secure and supports the privacy-enhancing features. Furthermore, most of the schemes remain only theoretic.

2.1. Our Contribution. The cryptographic scheme presented in this paper takes a novel approach for the access control based on rather the presence of multiple devices in user's proximity than the direct verification of user identifiers. The novel approach has two key benefits: it significantly improves the privacy protection of users and allows the authentication based on the presence of many low-performance devices. Our scheme is the first practical proposal with implementation results that combines strong security, all standard privacy-enhancing features, and efficiency.

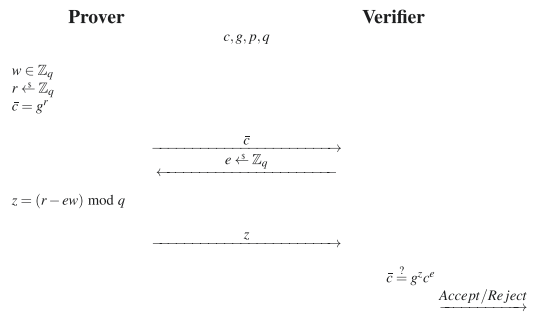

FIGURE 1: Schnorr's proof of knowledge of discrete logarithm PK $\{w : c = g^w\}$ in $\mathbb{Z}_p^*$.

(i) *Provable security*: all algorithms are provably secure, based on primitives with rigorous formal proofs.

(ii) *Multidevice authentication*: the scheme allows user authentication based on the presence of many personal devices.

(iii) *Anonymity*: the scheme allows authentication based on anonymous proofs of knowledge of private user and/or device identifiers.

(iv) *Unlinkability*: the scheme prevents creating user behavior profiles based on the authentication sessions linking.

(v) *Untraceability*: the scheme prevents any entity from tracing users (or their devices).

(vi) *Efficiency*: the authentication protocol is fast on constrained user devices (i.e., smart cards) and embedded verification terminals.

(vii) *Revocation and identification*: the proposed scheme is compatible with major revocation and identification schemes [22–24] for attribute-based credentials.

We not only provide the cryptographic description and security proofs of our scheme, but also provide practical implementation results based on benchmarks on RFID devices and an embedded hardware terminal. These results prove that the scheme can be practically implemented on existing off-the-shelf devices.

3. Preliminaries

3.1. Notation. We describe proof of knowledge protocols (PK) using the efficient notation introduced by Camenisch and Stadler [25]. The protocol for proving the knowledge of discrete logarithm of c with respect to g is denoted as PK$\{\alpha : c = g^\alpha\}$. The symbol ":" means "such that" and "$|x|$" is the bit length of x. We write $a \xleftarrow{\$} A$ when a is sampled uniformly at random from A. A secure hash function is denoted as $\mathscr{H}$.

3.2. Proofs of Knowledge. The statements about discrete logarithms in prime order groups can be easily proven using the Σ-protocols [26].

A simple yet very useful protocol for proving the discrete logarithm knowledge is based on the Schnorr signature scheme [27]. Using this protocol, the prover proves his knowledge of a discrete logarithm with respect to public parameters c, g, p, q; i.e., he proves the knowledge of $w : c = g^w \bmod p$, where p is prime modulus, q is group order, and g is $\mathbb{Z}_p^*$ generator. The protocol is depicted in Figure 1.

The proof of discrete logarithm knowledge is a simple 3-way protocol where the prover commits to a random number r in the first step, receives a challenge e in the second step, and responds by z to the challenge in the third step. The protocol is Honest Verifier Zero-Knowledge (HVZK). Note that the verifier does not have to know the private input w of the prover to be able to verify its knowledge. We recall the properties of the protocol below.

Proof. Completeness: prover who knows w is always accepted: $\bar{c} = g^z c^e = g^{r-ew} g^{ew} = g^r g^{-ew} g^{ew} = g^r = \bar{c}$. □

Proof. Soundness: let us assume a cheating prover is ready to answer at least 2 random challenges e, e' after committing to r without knowing w. Then, his responses z, z' must be accepted in verifier's checks:

$$\bar{c} = g^z c^e, \tag{1}$$

$$\bar{c} = g^{z'} c^{e'}, \tag{2}$$

we divide (1) and (2) and get

$$\frac{\bar{c}}{\bar{c}} = \frac{g^z c^e}{g^{z'} c^{e'}} \iff 1 = g^{z-z'} c^{e-e'}, \tag{3}$$

after multiplying both sides of (3) by $g^{-(z-z')}$ and raising to the power of $(e - e')^{-1}$, we get

$$g^{(z'-z)(e-e')^{-1}} = c \tag{4}$$

and we get the discrete logarithm $w = (z' - z)(e - e')^{-1}$ that is easy to efficiently compute for the dishonest prover; thus we reached the contradiction because the cheating prover unaware of w was assumed. □

Proof. Honest Verifier Zero-Knowledge: the ZK property is proven by proving the existence of the following ZK simulator M_V^*:

(1) the simulator randomly chooses the response $z' \overset{\$}{\leftarrow} \mathbb{Z}_q$.

(2) the simulator randomly chooses $e' \overset{\$}{\leftarrow} \mathbb{Z}_q$.

(3) the simulator computes $\overline{c'} = g^{z'} c^{e'}$.

□

The M_V^*'s output $\overline{c}', e', z'$ is computationally indistinguishable from the real protocol output c, e, z.

The protocol for proving the knowledge of a discrete logarithm described above can be extended to the discrete logarithm representation proof and discrete logarithm equivalence proof [25]. The principles and security proofs remain the same.

3.3. Bilinear Pairing. Let $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ be groups of prime order q. A bilinear map $\mathbf{e} : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$ is a map which satisfies bilinearity, i.e., $\mathbf{e}(g_1^x, g_2^y) = \mathbf{e}(g_1, g_2)^{xy}$ for all $x, y \in \mathbb{Z}_q$; nondegeneracy, i.e., for all generators $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$, $\mathbf{e}(g_1, g_2)$ generates $\mathbb{G}_T$; and efficiency, i.e., there exists an efficient algorithm $\mathcal{G}(1^k)$ that outputs the bilinear group $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathbf{e}, g_1, g_2)$. The pairing is a bilinear map and it is symmetric if $\mathbb{G}_1 = \mathbb{G}_2$. There are many types of pairings; however only Weil, Tate, Ate, and Eta parings are mainly used in cryptography due to their efficient computation. Our scheme makes use of the Tate pairing [28], since it is the fastest among them. Most of the known parings uses Millers algorithm [29] to do computations on elliptic curves.

3.4. Weak Boneh-Boyen Signature. The weak Boneh-Boyen (wBB) signature scheme [30] can be used to efficiently sign (blocks of) messages. Furthermore, the signature scheme can be easily integrated with the zero-knowledge proofs so that the knowledge of signed messages (and signatures themselves) can be proven anonymously, unlinkably, and untraceably. We recall the signing and verification algorithms below; the efficient proofs of knowledge are described, e.g., in [24].

`Setup`: On input security parameter k, generate a bilinear group $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathbf{e}, g_1, g_2) \longleftarrow \mathcal{G}(1^k)$. Take $sk \overset{\$}{\leftarrow} \mathbb{Z}_q$, compute $pk = g_2^{sk}$, and output sk as private key and $pk = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, \mathbf{e}, pk)$ as public key.

`Sign`: On input message $m \in \mathbb{Z}_q$ and secret key sk, output $\sigma = g_1^{1/(sk+m)}$.

`Verify`: On input the signature σ, message m, and public key pk, output 1 iff $\mathbf{e}(\sigma, pk) \cdot \mathbf{e}(\sigma^m, g_2) = \mathbf{e}(g_1, g_2)$ holds.

Showing the constant signature σ multiple times would make the authentication protocol linkable. All user sessions would be linkable to a single profile, which would make the resulting scheme very privacy unfriendly. To avoid linkability of signatures, users can only prove the knowledge of a valid signature by using the proof defined in [24]. In this proof, the user chooses a random value $r \overset{\$}{\leftarrow} \mathbb{Z}_q$ and computes randomized auxiliary values $\sigma' = \sigma^r$ and $\overline{\sigma} = \sigma'^{-m} g_1^r$. Then, the knowledge of a signature is proven by constructing the zero-knowledge proof $\pi = PK\{(m, r) : \overline{\sigma} = \sigma'^{-m} g_1^r\}$ and verifying $\mathbf{e}(\overline{\sigma}, g_2) = \mathbf{e}(\sigma', pk)$. The verifier is convinced that the user indeed knows a valid signature on a known message, although the proof does not release any of these values. That construction is perfect for our scheme, because the users want to convince verifiers that they know (device) identifiers signed by registrars, in an anonymous, untraceable, and unlinkable manner.

The wBB signatures were proven existentially unforgeable against a weak (nonadaptive) chosen message attack under the q-SDH assumption [30].

4. Multidevice Authentication with Privacy Protection

First, we define the formal requirements on the authentication scheme. Next, we define the algorithms and entities in the scheme. Finally, we present the concrete instantiation of the privacy-enhanced multidevice authentication scheme based on the wBB signatures described in the previous section.

4.1. Requirements. We require the scheme to be secure, private, and efficient.

Security Requirements

Completeness: registered users must be accepted by the `Authenticate` protocol.

Soundness: unregistered users must be rejected by the `Authenticate` protocol.

Zero-Knowledge: the `Authenticate` protocol transcript must be simulatable without the knowledge of identifiers, thus provably releasing no sensitive information.

Privacy Requirements

Anonymity: users must be able to prove the knowledge of their identifiers anonymously, without disclosing them.

Untraceability: user authentication sessions must be untraceable by all system entities, including registrars.

Unlinkability: all single user's authentication sessions must be mutually unlinkable.

Efficiency Requirements

Readiness for RFID devices: the scheme must be fast on constrained devices, in particular smart cards. No operations, that are unavailable on RFID devices (such as bilinear pairings), can be used in user's algorithms.

4.2. Definition of Algorithms. We define the algorithms and protocols of our scheme in this section. The communication pattern is depicted in Figure 2 and employs the registrar (i.e., a central server that manages users and their equipment), users (i.e., user devices such as smart cards or smart phones), terminals (i.e., embedded devices with RFID readers typically attached next to doors), and tags (i.e., devices that need to be present during authentication and access control, typically

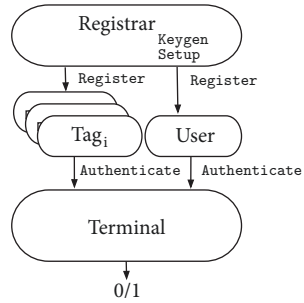

Figure 2: Architecture of multidevice authentication with privacy protection.

safety equipment with programmable RFID sticks, such as the helmet, respirator, or harness).

$(par) \longleftarrow \texttt{Setup}(1^k, n)$: the algorithm is run by the registrar. It inputs the security parameter k and the maximum number of tag classes (i.e., helmets, harnesses, boots, etc.). The algorithm outputs the public system parameters par.

$(sk_r, pk_r) \longleftarrow \texttt{Keygen}(par)$: the algorithm is run by the registrar. On the input of public system parameters par, it generates its private key sk_r and public key pk_r. The registrar distributes the public key to all other entities.

$(\langle ID_i, \sigma_i\rangle_{i=1}^n, ID_u, \sigma_u) \longleftarrow \texttt{Register}(par, sk_r, pk_r)$: the algorithm is run by the registrar. On the input of system parameters and its keypair, the registrar generates the tags' identifiers ID_i with corresponding signatures σ_i and user's identifier ID_u with a corresponding signature σ_u. The tag identifiers and signatures are securely delivered to tags and the user identifier and signature are delivered securely to the user device.

$(0/1) \longleftarrow \texttt{Authenticate}(par, \langle ID_i, \sigma_i\rangle_{i=1}^n, ID_u, \sigma_u, pk_r)$: the cryptographic protocol is run jointly by the user device, tags, and the terminal. It inputs system parameters, registrar's public key, private identifiers, and corresponding signatures and returns 1 iff signatures and IDs are valid, or 0 otherwise.

4.3. Instantiation Using wBB Signatures. In this section, we present the concrete instantiations of cryptographic algorithms defined in Section 4.2. We use the wBB signature scheme to certify the identifiers of tags and users in the `Register` algorithm and interactive proofs of knowledge to prove the knowledge of respective signatures and identifiers in the `Authenticate` protocol. We use the Camenisch-Stadler notation [25] to describe the proof of knowledge protocols.

`Setup`. The algorithm inputs the security parameter k and the maximum number of tag classes n. It generates the bilinear group with parameters $par = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \mathbf{e}, g_1, \ldots, g_n, g_u \in \mathbb{G}_1, g_2 \in \mathbb{G}_2)$ satisfying $|q| = k$.

`Keygen`. The algorithm inputs the public parameters par, selects random registrar's private keys $sk_r = (sk_0, sk_1, \ldots, sk_n, sk_u) \overset{\$}{\leftarrow} \mathbb{Z}_q^*$, and computes the public keys $pk_r = (pk_0 = g_2^{sk_0}, pk_1 = g_2^{sk_1}, \ldots, pk_n = g_2^{sk_n}, pk_u = g_2^{sk_u})$. It outputs the private keys as registrar's private output and the public key as the public output.

`Register`. The algorithm inputs the registrar's keys and public parameters, randomly selects tag and user identifiers $(ID_1, \ldots, ID_n, ID_u) \overset{\$}{\leftarrow} \mathbb{Z}_q$, and computes the wBB signatures $(\sigma_1, \ldots, \sigma_n)$ on tag identifiers $(ID_1, \ldots, ID_n)$ and the aggregated user signature σ_u and auxiliary values $\langle \sigma_{u_i}, \sigma_{u_i}^{-ID_i}\rangle_{i=1}^n$, $\sigma_{u_u}, \sigma_{u_u}^{-ID_u}$ that allow the construction of efficient proofs of knowledge in the `Authenticate` protocol. The algorithm outputs the tag identifiers and corresponding signatures as a private output to tags. The user identifier, the aggregated signature, and auxiliary values are outputted to the user as a private output. Both tags and the user receive the initial *seed* required for the synchronization of the zero-knowledge proofs as a private input. The algorithm is depicted in Figure 3.

`Authenticate`. `Authenticate` is an algorithm distributed among the user, terminal, and tags that inputs the identifiers and respective signatures and outputs 1 iff (1) all signatures are valid and created by the registrar and (2) all identifiers of the user are present and signed. Otherwise it outputs 0. The protocol is a distributed proof of knowledge of wBB signatures where the tags prove that they know their identifiers and corresponding signatures (without actually revealing them) and, at the same time, the user proves that he has an aggregated signature on all his tag identifiers, plus his own identifier. As the user does not know the tag identifiers, all tags must be present and participate on the proof construction. As a result, the user is able to anonymously, untraceably, and unlinkably prove his valid registration by the registrar and the presence of all his tags, i.e., the safety equipment. The protocol is depicted in abstract CS notation in Figure 4. We also provide the full description in Figure 5 in Section 6 focused on implementation.

5. Security Analysis

The registrar issues the wBB signatures to tags and users in the `Register` algorithm. Then, the user and tags prove

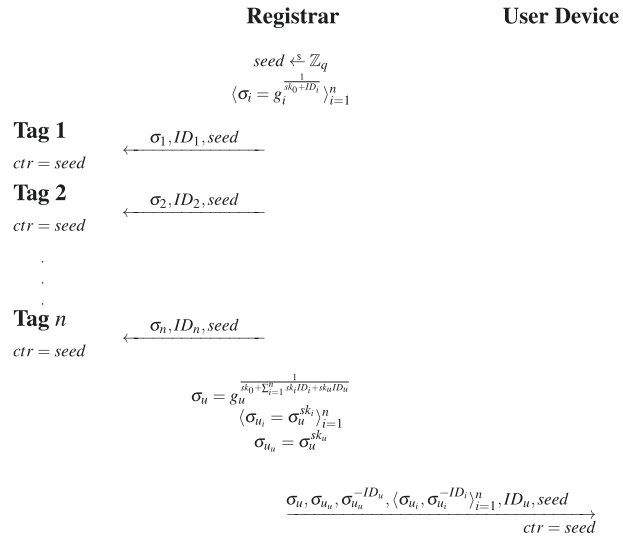

FIGURE 3: `Register` protocol.

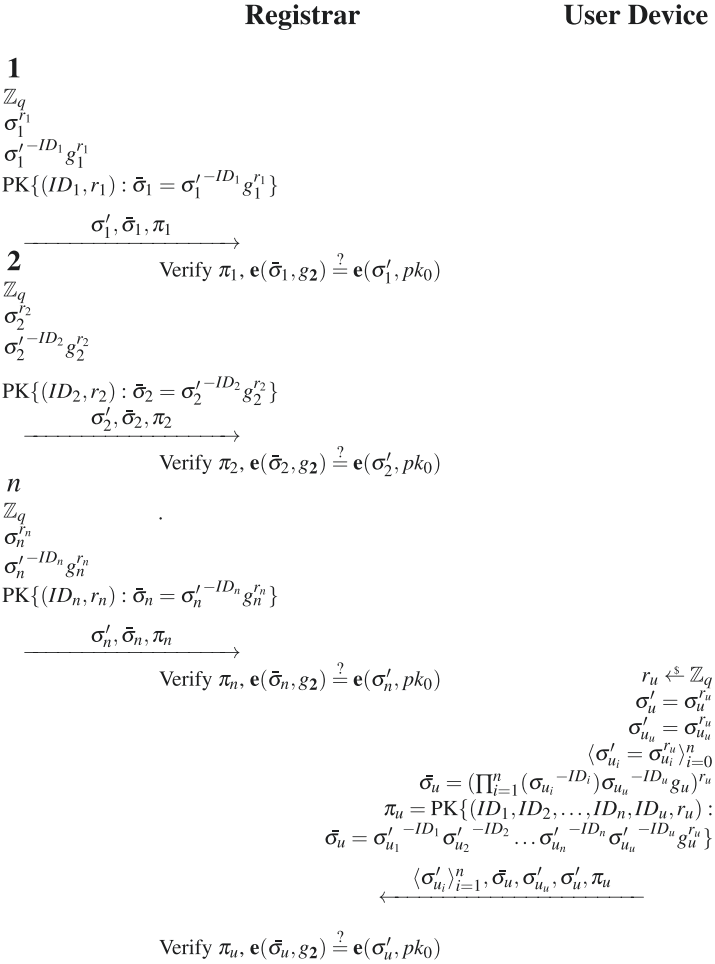

FIGURE 4: `Authenticate` protocol in CS notation.

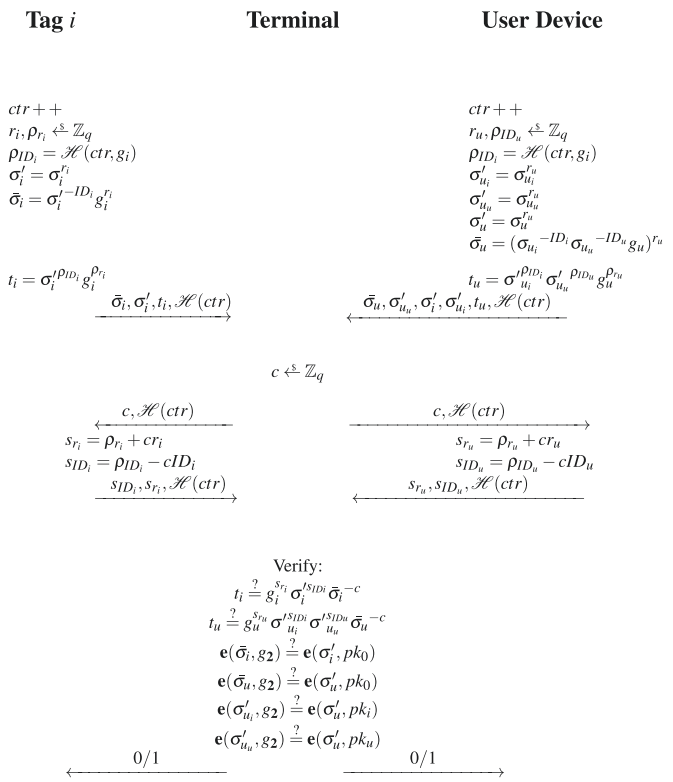

FIGURE 5: `Authenticate` protocol in full notation for i^{th} tag.

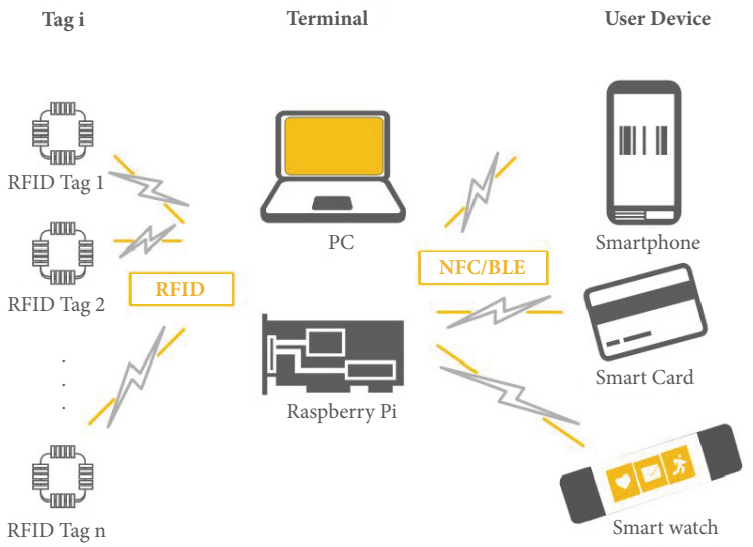

FIGURE 6: Tested scenario.

the knowledge of such signatures to the terminal using the distributed zero-knowledge proofs in the `Authenticate` protocol.

Lemma 1. *The weak Boneh-Boyen signatures are unforgeable against a weak chosen message attack under the q-Static Diffie-Hellman assumption [30].*

Lemma 1 is proven in [30].

Lemma 2. *The protocol presented in Figure 5 is complete, sound, and zero-knowledge.*

We construct the proof for a tag i using the standard proving technique for zero-knowledge protocols. For other

devices and the user, the proof is constructed analogically.

Proof. Completeness: honest users pass the terminal's check.

$$t_i = g_i^{s_{r_i}} \sigma_i'^{s_{IDi}} \overline{\sigma_i}^{-c} \tag{5}$$

$$= g_i^{\rho_{r_i}} g_i^{cr_i} \sigma_i^{r_i s_{ID_i}} \sigma_i'^{ID_i c} g_i^{-cr_i} \tag{6}$$

$$= g_i^{\rho_{r_i}} g_i^{cr_i} \sigma_i^{r_i \rho_{ID_i}} \sigma_i^{-r_i c ID_i} \sigma_i^{r_i ID_i c} g_i^{-cr_i} \tag{7}$$

$$= g_i^{\rho_{r_i}} \sigma_i'^{\rho_{ID_i}} \tag{8}$$

$$\mathbf{e}\left(\overline{\sigma_i}, g_2\right) = \mathbf{e}\left(\sigma_i', pk_0\right) \tag{9}$$

$$\mathbf{e}\left(\sigma_i^{-ID_i r_i} g_i^{r_i}, g_2\right) = \mathbf{e}\left(\sigma_i', g_2^{sk_0}\right) \tag{10}$$

$$\mathbf{e}\left(g_i^{(-ID_i r_i + r_i(sk_o + ID_i))/(sk_0 + ID_i)}, g_2\right) = \mathbf{e}\left(\sigma_i', g_2^{sk_0}\right) \tag{11}$$

$$\mathbf{e}\left(g_i^{r_i sk_o/(sk_0 + ID_i)}, g_2\right) = \mathbf{e}\left(\sigma_i', g_2^{sk_0}\right) \tag{12}$$

$$\mathbf{e}\left(\sigma_i'^{sk_0}, g_2\right) = \mathbf{e}\left(\sigma_i', g_2^{sk_0}\right) \tag{13}$$

□

Error Probability. If implemented correctly, the user will be always accepted.

Proof. Soundness: only registered users pass terminal's check. □

Assume a user who is not registered (i.e., does not know the identifier ID_i) and passes the terminal's check for two different challenges c and c' with two different responses s and s':

$$t_i = g_i^{s_{r_i}} \sigma_i'^{s_{IDi}} \overline{\sigma_i}^{-c} \tag{14}$$

$$t_i = g_i^{s'_{r_i}} \sigma_i'^{s'_{IDi}} \overline{\sigma_i}^{-c'} \tag{15}$$

and we get

$$\overline{\sigma_i}^{c-c'} = g_i^{s_{r_i} - s'_{r_i}} \sigma_i'^{s_{IDi} - s'_{IDi}} \tag{16}$$

and therefore

$$\overline{\sigma_i} = g_i^{(s_{r_i} - s'_{r_i})/(c-c')} \sigma_i'^{(s_{IDi} - s'_{IDi})/(c-c')} \tag{17}$$

Thus the user can efficiently compute both the randomizer $r_i = (s_{r_i} - s'_{r_i})/(c - c')$ and the identifier $ID_i = (s_{IDi} - s'_{IDi})/(c - c')$ and we reached the contradiction to our original assumption.

Error Probability. The attacker will pass the verification check if he can predict the challenge c. The probability of soundness error is thus $P = 2^{-|c|} = 2^{-q} = 2^{-224}$, which is negligible. With an expected rate of 100 ms per challenge, the expected time of breach is 4×10^{58} years.

Proof. Zero-Knowledge: the protocol releases no private information, i.e., there exists a zero-knowledge simulator M_V^*. Using the public parameters and the public key $(\overline{g}, \overline{g}^x)$ (we follow the proof presented in [24] that allows the simulator to input an auxiliary public key $(\overline{g}, \overline{g}^x)$: $\overline{g} \overset{\$}{\leftarrow} \mathbb{Z}_q$ from the registrar), the simulator chooses randomly and uniformly $(s_{r_i}, s_{ID_i}, r, c) \overset{\$}{\leftarrow} \mathbb{Z}_q$, computes $\sigma_i' = \overline{g}^r$, $\overline{\sigma_i} = (\overline{g}^x)^r$, $t_i = g_i^{s_{r_i}} \sigma_i'^{s_{IDi}} \overline{\sigma_i}^{-c}$, and outputs the proof $\pi = (\overline{\sigma_i}, \sigma_i', t_i, \mathcal{H}(r), c, (s_{r_i}, s_{ID_i}))$. The simulated transcript is computationally indistinguishable from the real run of the protocol. □

Error Probability. The attacker can try to guess the randomizers $r_i, \rho_{r_i}, r_u, \rho_{r_u}$ and break the discrete logarithm assumption. The probability is $P = 2^{-q} = 2^{-224}$ for each device, which is negligible. With an expected rate of 10 ms per computing the guess (the exponentiation), the expected time of breach is 4×10^{57} years.

As a result of the zero-knowledge property and randomization of all signatures, the protocol is also *anonymous*, *untraceable,* and *unlinkable*.

6. Implementation Aspects

The `Authenticate` protocol has been implemented as a standard 3-way interactive zero-knowledge proof of knowledge protocol described in Section 3.2. We use a parallel composition with one challenge and one response for all tags of a user to construct an AND proof for both tag and user signatures. The `Authenticate` protocol for i^{th} tag is fully specified in Figure 5.

To keep user devices synchronized, we use a counter that is initialized by a seed generated by the registrar. In the beginning of each session, the counter increments. To avoid losing synchronization, the hashed counter is broadcasted by the terminal so that the devices can compare it with their actual counter value (and with, e.g., 10 next precomputed values) and sync in case their counter is behind. The hashed counter also serves as the session identifier and thus is present in all three steps of the protocol.

In the first step of the protocol, the tag generates randomizers $r_i, \rho_{r_i}, \rho_{ID_i}$, computes randomized signatures $\sigma_i', \overline{\sigma_i}$, and computes the commitment to randomizers t_i. The randomized signatures, commitment to randomizers, and hashed randomizers are sent to the terminal.

In the second step, the terminal randomly selects its challenge c and sends it to all tags and devices, together with the obtained hash.

In the third step, the tag computes their answers s_{r_i}, s_{ID_i} of the zero-knowledge protocol.

After receiving the answers, the terminal is able to verify that the tag knows a valid signature and a corresponding tag identifier with respect to registrar's public key pk, without actually learning any user- or tag-identifying values.

The proof construction for the user is the same with the exception that the answers containing tag IDs are omitted, because the terminal makes use of the values received by

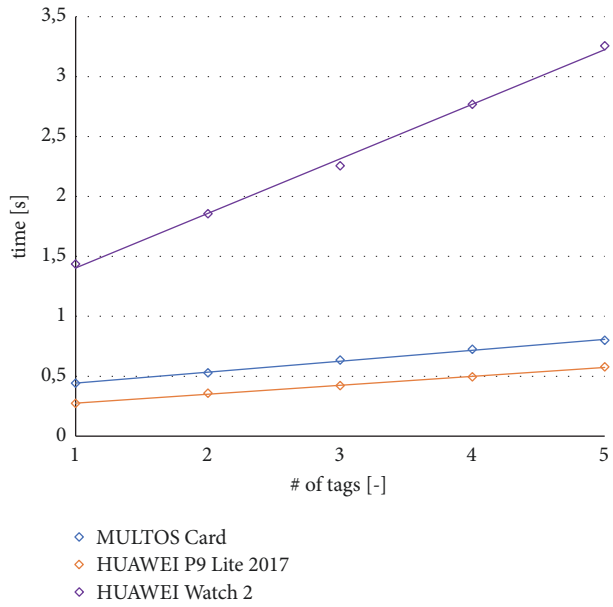

Figure 7: Dependence of the proof construction time on the number of user device.

the devices. Instead of proving tag IDs, the user proves the knowledge of his own user ID.

6.1. Performance Analysis. The scheme was designed to be practical and fast on constrained RFID devices, such as smart cards and programmable RFID tags. Therefore, the bilinear pairings, which are the most computationally complex operations in our algorithm, are only computed in the terminal which normally has more resources than user device. The second most complex operation is the exponentiation (implemented as scalar multiplication of an elliptic-curve point) and it is reduced to a minimum. The user device needs $(5 + 2d)$ exponentiations to construct a "user proof" with d personal tags. Each tag must compute 5 exponentiations to generate a "tag proof". However, our implementation uses only 4 exponentiations, since the value $(\sigma_i^{-ID_i} g_i)$ is precomputed within a `Register` protocol and is used for the randomized signature $\overline{\sigma_i} = \sigma_i'^{-ID_i} g_i'^{r_i} = (\sigma_i^{-ID_i} g_i)^{r_i}$ construction. The complexity of the other operations (random number generation, addition, and multiplication) are only minor, compared to pairings and exponentiations. In order to verify the proof, the terminal must compute $(4 + 4d)$ bilinear pairings and $(4 + 3d)$ exponentiations.

We provide performance measurement of crucial operations on common devices, which are widely used in the access control applications, i.e., a smart card, smart phone, smart watch (as user devices), and a custom-built RFID terminal with ARM or Intel CPU and programmable RFID tags (as RFID tags attached to safety equipment). The hardware and software specification of all the devices is presented in Table 1.

The testing scenario is depicted in Figure 6. The user needs to hold a wearable device, such as a smart phone (HUAWEI P9 Lite 2017), a smart card (Multos Card), or smart watch (HUAWEI Watch 2) and some safety equipment, such as helmets, harnesses, boots, and protective suits, each of them with a programmable RFID tag attached. The tag is equipped with a programmable chip SC23Z018 with Multos 4.2 operation system. The proofs are collected and verified by a terminal. We use Raspberry Pi 3 to represent the terminal. In another scenario, PC (Intel i7-7700 CPU, 16 GB RAM) acts as a central authentication server representing the case of a centralized access control system. The system uses RFID communication between tags and a terminal, and NFC or BLE communication between a terminal and a user device.

The performance of critical operations and the estimation of the running time of the `Authenticate` protocol with one RFID tag and one user device are presented in Table 2. In addition, we provide measurement of the selected devices where we consider different elliptic curves types, in particular type A and D. Both curves satisfy the NIST key recommendation for 80-bit security strength [31]. The performance is measured in milliseconds (the measurement of clock cycles is unavailable on the smart card platform) and the values are an average of 10 measurements, excluding communication overhead. For the implementation of EC operations, the PBC library was used [32] on the terminal and jPBC [33] library on Android devices. Native assembler code was used to perform operations on the Multos smart card.

The proposed authentication scheme can be used in many types of access control scenarios and for different types of devices. Therefore, we provide the results of each protocol using one RFID tag. Furthermore, we present the crucial EC operations' benchmarks on a wide range of devices in Table 3. The time is measured in milliseconds and the values are an average of 10 measurements, as in the previous case. All measurements were performed by using the elliptic curve d159 from the PBC library. We did not consider Android devices as a terminal device, since the pairing operation requires too much time and therefore it is not usable in practice.

Figure 7 depicts the time required for a proof construction on different devices (Multos smart card, Android smart

Table 1: Specification of tested devices.

	Type	CPU/MCU	OS	RAM
Tag	SC	SC23Z018	Multos 4.2	2 KB
User	SC	SC23Z018	Multos 4.2	2 KB
User	Phone	Kirin 655	Android 7.0	3 GB
User	Watch	ARM Cortex-A7	Android 7.0	768 MB
Terminal	Pi 3	ARM Cortex-A53	Raspbian 9.3	1 GB
Terminal	PC	Intel i7-7700	Debian 8.6	16 GB

Tag: programmable RFID stick, User: user device, Terminal: terminal, SC: smart card, Phone: HUAWEI P9 Lite 2017, Pi 3: Raspberry Pi 3 Model B, and Watch: HUAWEI Watch 2.

Table 2: Benchmark results based on EC type.

	Terminal [ms]	User Device [ms]	Tag [ms]
Elliptic Curve Type A			
Exponentiation	10	67	81
Pairing	15	125	-
Verification	192	-	-
Tag Proof Generation	-	-	444
User Proof Generation	-	448	-
Elliptic Curve Type D			
Exponentiation	4	38	40
Pairing	31	1050	-
Verification	271	-	-
Tag Proof Generation	-	-	277
User Proof Generation	-	273	-

Table 3: Benchmark results of all tested devices.

	SC [ms]	Phone [ms]	Watch [ms]	Pi 3 [ms]	PC [ms]
Exponentiation	40	38	207	3.3	0.4
Pairing	-	1050	6571	31	2.4
Tag Proof Generation	277	154	900	18	4
User Proof Generation	441	273	1502	24	5
Verification	-	-	-	271	21

phone, and smart watch for various number of tags). These devices act as a user device.

6.2. Revocation and Identification. Besides strong privacy-enhancing features, there must be also mechanisms to revoke and/or identify malicious users. All users are theoretically identifiable and traceable by their user IDs. However, these IDs are "hidden" in the signatures as the exponents. Due to the discrete logarithm problem assumption, one cannot easily get the identifiers and do the revocation and identification. However, our scheme is compatible with the major revocation schemes that are already available for cryptographic anonymous credential schemes [22–24]. In these revocation schemes, the hidden exponent (the user ID) is used as a revocation handle and can be disclosed only by designated authorities. Additionally, valid users remain anonymous while malicious users are identifiable and traceable by a designated authority, such as a court. Such schemes are provably secure, efficient, and compatible without any modification; thus we refer to their specification (e.g., the scheme designed directly for smart cards [24]) in case revocation is needed.

7. Conclusions

We presented a cryptographic scheme that allows a novel approach for controlling physical access. Instead of the verification of fixed user or device identifiers, the terminals can check only the knowledge of such identifiers in a private manner, without explicitly exposing any personal information or the identifiers themselves. Furthermore, the presence of other RFID devices, possibly the safety equipment, can be enforced. Our protocols are based on proven cryptographic

algorithms and are very practical—the proofs can be generated in under 500 ms on constrained devices, such as smart cards. We provided the full cryptographic description of all algorithms, the security and efficiency analysis, and the implementation results on constrained devices. We find the scheme especially useful in applications where the physical access to dangerous environment is granted upon proving the presence of required safety equipment and where the strong privacy-protection regulation is enforced by law.

As for the future work, we will focus on the optimization of the verification algorithm, since the current verification time grows linearly with the number of tags involved in the authentication protocol. In particular, we would like to reduce the number of bilinear pairings which is the most time-consuming operation in the protocol.

Data Availability

All necessary information is provided in the paper and in cited literature.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

Research described in this paper was financed by the National Sustainability Program under Grant LO1401. For the research, infrastructure of the SIX Center was used.

References

[1] W. Xu, "Mobile applications based on smart wearable devices," in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems, SenSys 2015*, pp. 505-506, Republic of Korea, November 2015.

[2] NIST. Federal information processing standards publication (FIPS 197). Advanced Encryption Standard (AES), 2001.

[3] N. Courtois, K. Nohl, and S. ONeil, *Algebraic attacks on the crypto-1 stream cipher in mifare classic and oyster cards*, IACR Cryptology ePrint Archive, 2008.

[4] N. T. Courtois, "The dark side of security by obscurity: And cloning MiFare classic rail and building passes, anywhere, anytime," in *Proceedings of the SECRYPT 2009 - International Conference on Security and Cryptography*, pp. 331–338, Italy, October 2009.

[5] F. D. Garcia, P. Van Rossum, R. Verdult, and R. W. Schreur, "Wirelessly pickpocketing a Mifare Classic card," in *Proceedings of the 2009 30th IEEE Symposium on Security and Privacy*, pp. 3–15, USA, May 2009.

[6] C. Paquin, "U-prove cryptographic specification v1.1," Technical report, 2011.

[7] O. Riva, C. Qin, K. Strauss, and D. Lymberopoulos, "Progressive authentication: Deciding when to authenticate on mobile phones," in *In USENIX Security Symposium*, pp. 301–316, 2012.

[8] V. S. Miller, "The Weil pairing, and its efficient calculation," *Journal of Cryptology. The Journal of the International Association for Cryptologic Research*, vol. 17, no. 4, pp. 235–261, 2004.

[9] W. Mostowski and P. Vullers, "Efficient U-prove implementation for anonymous credentials on smart cards," *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering*, vol. 96, pp. 243–260, 2012.

[10] P. Vullers and G. Alpár, "Efficient Selective Disclosure on Smart Cards Using Idemix," in *Policies and Research in Identity Management*, vol. 396 of *IFIP Advances in Information and Communication Technology*, pp. 53–67, Springer, Berlin, Heidelberg, 2013.

[11] D. Oswald and C. Paar, "Breaking Mifare DESFire MF3ICD40: Power analysis and templates in the real world," in *Cryptographic Hardware and Embedded Systems–CHES 2011*, vol. 6917, pp. 207–222, 2011.

[12] J. Camenisch and E. V. Herreweghen, "Design and implementation of the idemix anonymous credential system," in *Proceedings of the 9th ACM Conference on Computer and Communications Security CCS '02*, pp. 21–30, ACM, New York, NY, USA, November 2002.

[13] S. Micali and C.-P. Schnorr, "Efficient, perfect polynomial random number generators," *Journal of Cryptology. The Journal of the International Association for Cryptologic Research*, vol. 3, no. 3, pp. 157–172, 1991.

[14] J. Hajny and L. Malina, "Unlinkable attribute-based credentials with practical revocation on smart-cards," in *Smart Card Research and Advanced Applications - CARDIS*, S. Mangard, Ed., vol. 7771 of *Lecture Notes in Computer Science*, pp. 62–76, Springer, Berlin, Heidelberg, 2013.

[15] NIST. Federal information processing standards publication (FIPS 46-3). Data Encryption Standard (DES), 1999.

[16] J. Hajny, L. Malina, and O. Tethal, "Privacy-Friendly Access Control Based on Personal Attributes," in *Proceedings of the The 9th International Workshop on Security*, vol. 8639 of *Lecture Notes in Computer Science*, pp. 1–16, Springer International Publishing.

[17] https://srlabs.de/analyzing-legic-prime-rfids/.

[18] B.-R. Cha, S.-H. Lee, S.-B. Park, and Y.-K. Ji, "Design of micropayment to strengthen security by 2 factor authentication with mobile wearable devices," *Advanced Science and Technology Letters (ASTL)*, vol. 109, no. 7, pp. 28–32, 2015.

[19] I. Butun, M. Erol-Kantarci, B. Kantarci, and H. Song, "Cloud-centric multi-level authentication as a service for secure public safety device networks," *IEEE Communications Magazine*, vol. 54, no. 4, pp. 47–53, 2016.

[20] L. Gonzalez-Manzano, J. d. Fuentes, and A. Orfila, "Access Control for the Cloud Based on Multi-device Authentication," in *Proceedings of the 2015 IEEE Trustcom/BigDataSE/ISPA*, pp. 856–863, Helsinki, Finland, August 2015.

[21] J. Hajny, P. Dzurenda, and L. Malina, "Multi-device authentication using wearables and iot," in *In Proceedings of the 13th International Joint Conference on e-Business and Telecommunications, ICETE 2016*, pp. 483–488, SCITEPRESS - Science and Technology Publications, Lda, Portugal, 2016.

[22] E. Brickell, J. Camenisch, and L. Chen, "Direct anonymous attestation," in *Proceedings of the 11th ACM Conference on Computer and Communications Security (CCS '04)*, pp. 132–145, ACM, New York, NY, USA, 2004.

[23] P. Bichsel, J. Camenisch, T. Groß, and V. Shoup, "Anonymous credentials on a standard Java card," in *Proceedings of the 16th ACM Conference on Computer and Communications Security, CCS'09*, pp. 600–610, New York, NY, USA, November 2009.

[24] J. Camenisch, M. Drijvers, and J. Hajny, "Scalable revocation scheme for anonymous credentials based on n-times unlinkable

proofs," in *Proceedings of the 15th ACM Workshop on Privacy in the Electronic Society, WPES 2016*, pp. 123–133, New York, NY, USA, 2016.

[25] J. Camenisch and M. Stadler, "Efficient group signature schemes for large groups," in *Advances in Cryptology — CRYPTO '97*, vol. 1294 of *Lecture Notes in Computer Science*, pp. 410–424, Springer Berlin Heidelberg, Berlin, Heidelberg, 1997.

[26] R. Cramer, *Modular Design of Secure Yet Practical Cryptographic Protocols*, Universiteit van Amsterdam, 1997.

[27] The White House. National strategy for trusted identities in cyberspace, 2011. http://www.whitehouse.gov/sites/default/files/rss_viewer/NSTICstrategy_041511.pdf .

[28] S. D. Galbraith, K. Harrison, and D. Soldera, "Implementing the tate pairing," in *Algorithmic Number Theory*, C. Fieker and D. R. Kohel, Eds., pp. 324–337, Springer, Berlin, Heidelberg, 2002.

[29] I. Naumann and G. Hogben, "Enisa: Privacy features of eid cards," *Network Security Newslette*, vol. 2008, no. 8, pp. 9–13, 2008.

[30] D. Boneh and X. Boyen, "Short signatures without random oracles," in *Advances in Cryptology—EUROCRYPT 2004*, vol. 3027 of *Lecture Notes in Computer Science*, pp. 56–73, Springer, Berlin, Germany, 2004.

[31] E. Barker, "Recommendation for key management part 1: General (revision 4)," in *NIST Special Publication Part 1*, vol. 800 (57), pp. 1–147, 2016.

[32] M. Meriac, "Heart of darkness-exploring the uncharted backwaters of hid iclasstm security," *Heart*, 2010.

[33] A. De Caro and V. Iovino, "jPBC: Java pairing based cryptography," in *Proceedings of the 16th IEEE Symposium on Computers and Communications (ISCC '11)*, IEEE, pp. 850–855, Kerkyra, Corfu, Greece, July 2011.