Fast Keyed-Verification Anonymous Credentials on Standard Smart Cards

Jan Camenisch[1], Manu Drijvers[1], Petr Dzurenda[2], and Jan Hajny[2]

[1] Dfinity, Zurich, CH, {jan, manu}@dfinity.org
[2] Brno University of Technology, Brno, CZ, {dzurenda, hajny}@feec.vutbr.cz

Keywords: Privacy · anonymous credentials · authentication · smart cards.

Abstract. Cryptographic anonymous credential schemes allow users to prove their personal attributes, such as age, nationality, or the validity of a ticket or a pre-paid pass, while preserving their privacy, as such proofs are unlinkable and attributes can be selectively disclosed. Recently, Chase et al. (CCS 2014) observe that in such systems, a typical setup is that the credential issuer also serves as the verifier. They introduce keyed-verification credentials that are tailored to this setting. In this paper, we present a novel keyed-verification credential system designed for lightweight devices (primarily smart cards) and prove its security. By using a novel algebraic MAC based on Boneh-Boyen signatures, we achieve the most efficient proving protocol compared to existing schemes. To demonstrate the practicality of our scheme in real applications, including large-scale services such as public transportation or e-government, we present an implementation on a standard, off-the-shelf, Multos smart card. While using significantly higher security parameters than most existing implementations, we achieve performance that is more than 44 % better than the current state-of-the-art implementation.

1 Introduction

Using cryptographic credentials, users can anonymously prove the ownership of their personal attributes, such as age, nationality, sex or ticket validity. In the recent two decades, many proposals for anonymous credential schemes have been published. Starting with the fundamental works of Chaum [24], Brands [12], Camenish and Lysyanskaya [18], until recent schemes [23, 1, 35, 28, 4], researchers try to find a scheme that fulfills all requirements on privacy, is provably secure and is so efficient that it can be implemented on constrained devices. While there are schemes that fulfill all the requirements and can be implemented on PC and smartphone platforms, existing schemes deployed on smart cards are still not sufficiently fast for many applications, such as e-ticketing and eIDs. Yet, smart cards are the most appropriate platform for storing and proving personal attributes in everyday life, due to their size, security and reliability.

The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-030-22312-0

There are two major reasons why we lack practical implementations of anonymous credentials on smart cards. First, the complexity of asymmetric cryptographic algorithms used in anonymous credentials is quite high even for modern smart cards. Second, modern cryptographic schemes, including anonymous credentials, are mostly based on operations over an elliptic curve, while most available smart cards do not provide API for these operations. Particularly, the very popular operation of bilinear maps is still unsupported on this platform and simple operations, such as EC point scalar multiplication and addition, are significantly restricted.

In this paper, we address both these concerns: First, we propose a novel keyed-verification anonymous credential scheme that is designed to allow for smart card implementations. Our scheme has the most efficient proving algorithm to date and requires only operations that are available on existing off-the-shelf smart cards. Second, we present the implementation of our anonymous credential scheme that is 44 % - 72 % faster than the current state-of-the-art implementation, while even providing a higher security level.

1.1 Related Work

Cryptographic anonymous credential schemes were first defined by the seminal works of Chaum [24], Brands [12] and Camenisch and Lysyanskaya [18]. The schemes were gradually improved by adding revocation protocols [19, 16], using more efficient algebraic structures [35, 20] and developing security models and formal proofs [17]. Idemix [22] and U-Prove [33] are the examples of the most evolved schemes aiming for a practical use. Recently, a new approach to obtain more efficient anonymous credential schemes was proposed. Chase et al. [23] argue that in many scenarios where anonymous credentials could be deployed, the issuer of the credential will also serve as the verifier. This means that the verifier possesses the issuer key, which can be leveraged to obtain more efficient anonymous credential schemes tailored to setting. They formally define these so-called *Keyed-Verification Anonymous Credentials (KVAC)* and propose two instantiations. Barki et al. [3] propose a new KVAC scheme which is currently the most efficient: Proving posession of a credential with u hidden attributes costs $u + 12$ exponentiations. Couteau and Reichle [25] construct a KVAC scheme with presentation cost $2u + 3$ exponentiations in a 2048-bit group, which is less efficient, but works in the standard model. Some of the new constructions were already implemented on the PC platform with promising results [29, 35]. Yet, the implementations on the smart card platform are available only for the former schemes that are based on traditional, rather inefficient modular structures [31, 39, 34, 8]. Furthermore, most implementations use only 1024-bit RSA groups that are considered insufficient by today's standards [38]. Implementations with higher security parameters [4, 1, 3, 5] either need distribution of computation to another device (usually a mobile phone) or use a non-standard proprietary API for EC operations and rely on pre-computations (which is impossible in crucial applications like e-ticketing and eID where the card is inactive and starts only for the attribute presentation). Regarding speed, the best-performing implementation

of Idemix by the IRMA project [39] is able to compute the unlinkable attribute proof in at least 0,9 seconds, which is not convenient for time-critical applications where the proof should be presented in less than 500 ms. Currently, there is no cryptographic proposal and its implementation that would realize unlinkable anonymous credentials on the smart card platform with performance and security parameters necessary for a practical deployment.

1.2 Our Contribution

We propose a novel cryptographic scheme for anonymous attribute-based credentials that is designed primarily for smart cards. It provides all necessary privacy-protection features, i.e., the anonymity, unlinkability, untraceability and selective disclosure of attributes. The scheme is based on our original algebraic MAC that makes its proving protocol very efficient. The computational complexity of our proving protocol is the lowest from related schemes (only $u + 2$ scalar multiplications to present an attribute ownership proof) and we need only basic arithmetic operations that are already provided by existing smart cards' APIs. We present the results of the full implementation of our proving protocol that is faster by at least 44 % than the state-of-the-art implementation. By reaching the time of 366 ms including overhead, which is required for proving personal attributes on a 192-bit EC security level, we argue that the anonymous credentials are finally secure and practical even for time-critical and large-scale applications like eIDs, e-ticketing and mass transportation.

2 Preliminaries

2.1 Notation

We describe (signature) proof of knowledge protocols (SPK) using the efficient notation introduced by Camenisch and Stadler [21]. The protocol for proving the knowledge of discrete logarithm of c with respect to g is denoted as $\mathrm{SPK}\{\alpha : c = g^{\alpha}\}$. The symbol ":" means "such that" and $|x|$ is the bitlength of x. The symbol $\mathcal{H}$ denotes a secure hash function. We write $a \xleftarrow{\$} A$ when a is sampled uniformly at random from A. Let $\mathsf{GroupSetup}(1^{\kappa})$ be an efficient algorithm that generates a group $\mathbb{G} = \langle g \rangle$ of prime order q, such that $|q| = \kappa$. Let $\mathbf{e}$ denote a bilinear map.

2.2 Weak Boneh-Boyen Signature

We recall the weak Boneh-Boyen signature scheme [11], which is existentially unforgeable against a weak (non-adaptive) chosen message attack under the q-SDH assumption.

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

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

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

2.3 Algebraic MACs

Compared to traditional Message Authentication Codes (MACs), algebraic MACs can be efficiently combined with zero knowledge proofs. In terms of security, algebraic MACs [23] are no different from traditional MACs. A MAC scheme consists of algorithms $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{MAC}, \mathsf{Verify})$. Setup sets up the system parameters par that are given as implicit input to the other algorithms. KeyGen creates a new secret key, $\mathsf{MAC}(sk, m)$ computes a MAC on message m, and Verify is used to verify MACs. We recall the security definitions due to Dodis et al. [26] and slightly strengthened by Chase et al. [23], and require completeness and unforgeability under chosen message and verification attack (uf-cmva).

Definition 1. *A MAC scheme* $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{MAC}, \mathsf{Verify})$ *is complete if the following probability is negligible in* κ *for all messages* m*:*

$$\Pr\Big[\mathsf{Verify}(sk, m, \sigma) = 0 \mid par \xleftarrow{\$} \mathsf{Setup}(1^\kappa), \\ (ipar, sk) \xleftarrow{\$} \mathsf{KeyGen}(par), \sigma \xleftarrow{\$} \mathsf{MAC}(sk, m)\Big].$$

Definition 2. *A MAC scheme* $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{MAC}, \mathsf{Verify})$ *is* $(t, \epsilon, q_{\mathsf{MAC}}, q_{\mathsf{Verify}})$*-unforgeable under chosen message and verification attack if there exists no adversary* $\mathcal{A}$ *running in time* t *making at most* q_{MAC} MAC *queries and at most* q_{Verify} Verify *queries, for which the following probability is at least* ϵ*:*

$$\Pr\Big[\mathsf{Verify}(sk, m^*, \sigma^*) = 1 \wedge m^* \notin Q \mid par \xleftarrow{\$} \mathsf{Setup}(1^\kappa), \\ (ipar, sk) \xleftarrow{\$} \mathsf{KeyGen}(par), (\sigma^*, m^*) \xleftarrow{\$} \mathcal{A}^{\mathcal{O}^{\mathsf{MAC}(\mathtt{sk},\cdot)}, \mathcal{O}^{\mathsf{Verify}(\mathtt{sk},\cdot,\cdot)}}(par, ipar)\Big].$$

3 Our Algebraic MAC

This section describes our novel algebraic MAC scheme $\mathsf{MAC}_{\mathsf{wBB}}$, which is based on the weak Boneh-Boyen signature. It works in a prime order group and can MAC vectors of n messages $\vec{m} = (m_1, \ldots, m_n)$, with $m_i \in \mathbb{Z}_q^*$, using the technique due to Camenisch et al. [14] to extend the Boneh-Boyen signature to multiple messages. The scheme is composed of the following algorithms.

$\mathsf{Setup}(1^\kappa)$: Output $par = (\mathbb{G}, g, q) \leftarrow \mathsf{GroupSetup}(1^\kappa)$.

$\mathsf{KeyGen}(par)$: Choose $x_i \xleftarrow{\$} \mathbb{Z}_q^*$ for $i = (0, \ldots, n)$. Output secret key $sk = (x_0, \ldots, x_n)$ and public issuer parameters $ipar \leftarrow (X_0, \ldots, X_n)$ with $X_i = g^{x_i}$.

$\mathsf{MAC}(sk, \vec{m})$: Let $sk = (x_0, \ldots, x_n)$ and $\vec{m} = (m_1, \ldots, m_n)$. Compute $\sigma = g^{\frac{1}{x_0 + \sum_{i=1}^{n} m_i x_i}}$ and auxiliary information $\sigma_{x_i} \leftarrow \sigma^{x_i}$ for $i = (1, \ldots, n)$.[4] Output the authentication code $(\sigma, \sigma_{x_1}, \ldots, \sigma_{x_n})$.

$\mathsf{Verify}(sk, \vec{m}, \sigma)$: Let $sk = (x_0, \ldots, x_n)$ and $\vec{m} = (m_1, \ldots, m_n)$. Output 1 iff $g = \sigma^{x_0 + \sum_{i=1}^{n} m_i x_i}$.

Unforgeability of our MAC scheme holds under the SCDHI assumption, which is a variation of the SDDHI assumption [15].

Theorem 1. *Our MAC scheme is unforgeable, as defined in Definition 2, under the SCDHI assumption. More precisely, if n-SCDHI is (t, ϵ)-hard, then our MAC scheme is (t, ϵ)-unforgeable.*

We formally prove Theorem 1 in Appendix B. We introduce the SCDHI problem in Appendix A and prove its hardness in generic groups in Theorem 3.

4 Keyed-Verification Anonymous Credential Scheme

We construct our keyed-verification anonymous credential (KVAC) scheme using the algebraic MAC scheme presented in Section 3 above. Unlike traditional anonymous attribute-based credential schemes (ABCs), the verifier needs to know the secret keys to be able to verify user's attributes in keyed-verification anonymous credential schemes. This feature is particularly convenient for scenarios where attribute issuers are the same entities as attribute verifiers. The mass transportation settings is an example of such a scenario because the transportation authority both issues and checks the tickets and passes. The KVAC scheme supports all the standard privacy-enhancing features of ABC schemes, such as anonymity, unlinkability, untraceability, and selective disclosure of attributes, and is compatible with major credential schemes [22, 33] and standard revocation schemes [18, 13].

4.1 Definition of Keyed-Verification Anonymous Credential Schemes

A KVAC scheme consists of algorithms (Setup, $\mathsf{CredKeygen}$, Issue, Obtain, Show, $\mathsf{ShowVerify}$)[5] that are executed by users and an issuer who also serves as a verifier.

$\mathsf{Setup}(1^k)$ takes as input the security parameter and outputs the system parameters *par*. We assume that *par* is given as implicit input to all algorithms.

[4] Note that the the auxiliary information σ_{x_i} can be omitted as they are not required for verification. However, in our keyed verification credentials, it will turn out that adding these values will make credential presentation more efficient.

[5] Note that Chase et al. [23] define $\mathsf{BlindIssue}$ and $\mathsf{BlindObtain}$, but as we do not show efficient algorithms for blind issuance, we omit them from the definition here. Instead, we define Obtain, which lets a user check that a credential is indeed valid using only the public issuer parameters.

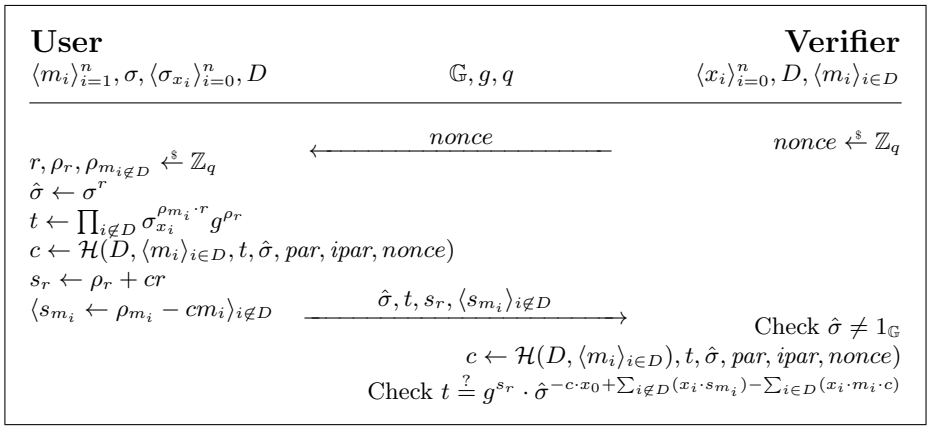


Fig. 1: Definition of the Show and ShowVerify algorithms of our KVAC scheme.

CredKeygen(par) outputs a issuer secret key sk and issuer parameters $ipar$.

Issue($sk, (m_1, \ldots, m_n)$) takes as input the issuer secret key and attribute values $(m_1, \ldots, m_n)$ and outputs a credential $cred$. The issuance of attributes must be done over a secure channel (as the attributes and private AMAC are sent between the user and issuer) and the credential should be stored on a tamper-proof device (we use a smart-card).

Obtain($ipar, cred, (m_1, \ldots, m_n)$) lets a user verify a credential by giving as input the public issuer parameters, the credential and the attribute values.

Show($ipar, cred, (m_1, \ldots, m_n), \phi$) $\leftrightarrow$ ShowVerify(sk, ϕ) is an interactive algorithm. The user runs Show on input the public issuer parameters, the credential, the attribute values and attribute predicate, and the verifier runs ShowVerify on input the issuer secret key and the attribute predicate, which will output 1 iff it accepts the credential presentation.

4.2 Our KVAC Scheme Based on $\mathsf{MAC}_{\mathsf{wBB}}$

In this section, we present our novel KVAC scheme that uses $\mathsf{MAC}_{\mathsf{wBB}}$ as introduced in Section 3. Our scheme certifies attributes in $\mathbb{Z}_q^*$ and is parametrized by n, the amount of attributes in a credential. We describe our scheme using *selective disclosure* as attribute predicates, i.e., a predicate ϕ can be seen as a set $D \subseteq \{1, \ldots, n\}$ containing the indices of the disclosed attributes and the attribute values of the disclosed attributes $\langle m_i \rangle_{i \in D}$. On a high level, we follow the approach from Chase et al [23] and build our KVAC scheme from our algebraic MAC presented in Section 3 and zero knowledge proofs. One novel trick allows us to strongly improve the efficiency of our scheme. Instead of computing a standard noninteractive Schnorr-type proof of knowledge, we use the fact that the verifier knows the secret key. This allows us to omit elements that the verifier can compute by itself and saves the prover a lot of work.

We note that our Issue algorithm does not support the efficient issuance of committed attributes, i.e., the blind issuance. This feature is useful in applications where a user needs to transfer his attributes among credentials or needs to get issued attributes that are only private to him. However, we consider these scenarios rare in targeted applications such as e-ticketing, mass transportation and loyalty cards. Furthermore, if the issuance of committed attributes is necessary, it can be done by employing Paillier encryption [32], as is shown in [6].

$\mathsf{Setup}(1^k)$: Output $par = (\mathbb{G}, g, q) \leftarrow \mathsf{GroupSetup}(1^\kappa)$.

$\mathsf{CredKeygen}(par)$: Run $(sk, ipar) \leftarrow \mathsf{MAC}_{\mathsf{wBB}}.\mathsf{KeyGen}(par)$ and output sk and $ipar$.

$\mathsf{Issue}(sk, (m_1, \ldots, m_n))$: Run $(\sigma, \langle \sigma_{x_i} \rangle_{i=0}^n) \leftarrow \mathsf{MAC}_{\mathsf{wBB}}.\mathsf{MAC}(sk, (m_1, \ldots, m_n))$. Next, provide a proof that allows a user to verify the validity of the credential: $\pi \leftarrow SPK\{(x_0, \ldots, x_n) : \bigwedge_{i=0}^n \sigma_{x_i} = \sigma^{x_i} \wedge X_i = g^{x_i}\}$. Output credential $cred \leftarrow (\sigma, \langle \sigma_{x_i} \rangle_{i=0}^n, \pi)$.

$\mathsf{Obtain}(ipar, cred, (m_1, \ldots, m_n))$: Parse $ipar$ as $(X_0, \ldots, X_n)$ and parse $cred$ as $(\sigma, \langle \sigma_{x_i} \rangle_{i=0}^n, \pi)$. Check that $\sigma_{x_0} \cdot \prod_{i=1}^n \sigma_{x_i}^{m_i} = g$ and verify π with respect to $ipar$ and σ.

$\mathsf{Show}(ipar, cred, (m_1, \ldots, m_n), (D, \langle m_i \rangle_{i \in D}))$: In credential presentation, we want to let the user prove posession of a valid credential with the desired attributes. On a high level, we want to prove knowledge of a weak Boneh-Boyen signature, so we can apply the efficient proof due to Arfaoui et al. [2] and Camenisch et al. [13], by extending it to support a vector of messages: Take a random $r \xleftarrow{\$} \mathbb{Z}_q^*$ and let $\hat{\sigma} \leftarrow \sigma^r$ and $\hat{\sigma}_{x_i} \leftarrow \sigma_{x_i}^r$ for $i = 0, \ldots, n$, and prove

$$SPK\{(\langle m_i \rangle_{i \notin D}, r) : \hat{\sigma}_{x_0} \prod_{i \in D} \hat{\sigma}_{x_i}^{m_i} = g^r \prod_{i \notin D} \hat{\sigma}_{x_i}^{-m_i}\}.$$

The verifier simply checks that the $\hat{\sigma}_{x_i}$ values are correctly formed and verifies the proof.

While this approach is secure and conceptually simple, it is not very efficient. We now present how we can construct a similar proof in a much more efficient manner. The key observation is that the user does not have to compute anything that the verifier, who is in possession of the issuer secret key sk, can compute. This means we can omit the computation of the $\hat{\sigma}_{x_i}$ values and define Show as follows. Randomize the credential by taking a random $r \leftarrow \mathbb{Z}_q^*$ and setting $\hat{\sigma} \leftarrow \sigma^r$. Take $\rho_r, \rho_{m_{i \notin D}} \xleftarrow{\$} \mathbb{Z}_q$ and compute

$$t = \prod_{i \notin D} \sigma_{x_i}^{\rho_{m_i} \cdot r} g^{\rho_r}, \quad c \leftarrow \mathcal{H}(D, \langle m_i \rangle_{i \in D}), t, \hat{\sigma}, par, ipar, nonce),$$

and let $s_r = \rho_r + cr, \langle s_{m_i} = \rho_{m_i} - cm_i \rangle_{i \notin D}$. Send $(\hat{\sigma}, t, s_r, \langle s_{m_i} \rangle_{i \notin D})$ to the verifier.

$\mathsf{ShowVerify}(sk, (D, \langle m_i \rangle_{i \in D}))$: The verifier running $\mathsf{ShowVerify}$ will receive $(\hat{\sigma}, t, s_r, \langle s_{m_i} \rangle_{i \notin D})$ from the user. It recomputes

$$c \leftarrow \mathcal{H}((m_1, \ldots, m_n), (D, \langle m_i \rangle_{i \in D}), t, \hat{\sigma}, par, ipar, nonce)$$

and checks

$$t \stackrel{?}{=} g^{s_r} \cdot \hat{\sigma}^{-c \cdot x_0 + \sum_{i \notin D}(x_i \cdot s_{m_i}) - \sum_{i \in D}(x_i \cdot m_i \cdot c)}.$$

Output 1 if valid and 0 otherwise. The Show and $\mathsf{ShowVerify}$ algorithms are depicted in Fig. 1.

Theorem 2. *Our keyed-verification credential scheme is secure following the definition by Chase et al. [23] (ommitting the blind issuance), under the n-SCDHI assumption (as defined in Definition 3) in the random oracle model.*

We formally prove Theorem 2 in Appendix C.

4.3 Efficiency

Our Show and $\mathsf{ShowVerify}$ algorithms were designed to be efficient enough to run on smart cards. We avoided computing bilinear pairings due to their computational cost and the lack of support on existing smart cards. The use of the second most expensive operation, the exponentiation (or scalar multiplication of EC points respectively), is reduced to a minimum. Our proving algorithm, the part of the protocol we envision being executed on a smart card, only requires $u + 2$ exponentiations, where u is the number of undisclosed attributes.

Table 1 compares the efficiency of our Show protocol to existing KVAC schemes [23, 4], well-known anonymous credential schemes U-Prove [33] and Identity Mixer [22], and a recent scheme by Ringers et al. [35]. Idemix takes place in the RSA group, meaning that the exponentiations are much more expensive than exponentiations in a prime order group. U-Prove lacks the unlinkability property. Compared to $\mathsf{MAC}_{\mathsf{BB}}$, our scheme requires only 2 exponentiations without hidden attributes, whereas $\mathsf{MAC}_{\mathsf{BB}}$ requires 12, showing that especially for a small number of undisclosed attributes, our scheme is significantly faster than $\mathsf{MAC}_{\mathsf{BB}}$.

Table 1: Comparison of presentation protocols of credential schemes.

	Exp. prime	**Exp. RSA**	**Unlink.**	**MAC**	**Security**
U-Prove [33]	$u + 1$	0	✗	✗	-
Idemix [22]	0	$u + 3$	✓	✗	sRSA [36]
Ringers et al. [35]	$n + u + 9$	0	✓	✗	whLRSW [40]
$\mathsf{MAC}_{\mathsf{DDH}}$ [23]	$6u + 12$	0	✓	✓	DDH [9]
$\mathsf{MAC}_{\mathsf{GGM}}$ [23]	$5u + 4$	0	✓	✓	GGM [37]
$\mathsf{MAC}_{\mathsf{BB}}$ [4]	$u + 12$	0	✓	✓	q-sDH [10]
NIKVAC [25]	$2u + 3$	0	✓	✓	GGM+IND-CPA
This work	$u + 2$	0	✓	✓	n-SCDHI (Def. 3)

5 Implementation Results

There are many cryptographic schemes for anonymous attribute-based credentials available. Nevertheless, the smart card implementations are only very few [31, 39, 27] and not practically usable as they use only small insecure security parameters to be able to achieve reasonable speed. Particularly, only 1024-bit RSA or DSA groups are used. That is considered insecure for any practical deployment today.

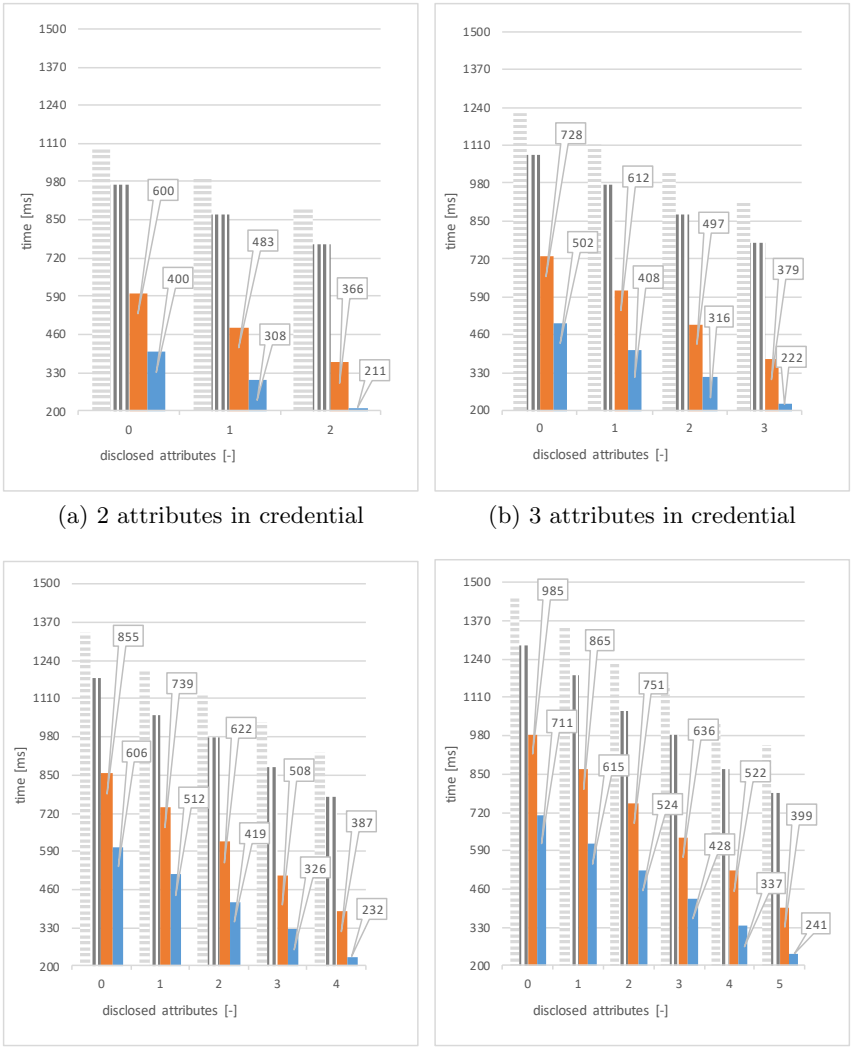
The `Show` and `ShowVerify` algorithms of our scheme were implemented using a standard NIST P-192 curve [30] on the Multos ML3 smart card. Only standard Multos API and free public development environment (Eclipse IDE for C/C++ Developers, SmartDeck 3.0.1, MUtil 2.8) were used. For terminal application, Java BigInteger class and BouncyCastle API were used. We compare our results (blue and orange) with the state-of-the-art results of Vullers and Alpár (VA) [39] (black and white) for different numbers of attributes stored and disclosed in Fig. 2. We note that our implementation uses significantly higher security parameters (1024-bit used by Vullers and Alpár vs. 1776-bit DSA group equivalent according to [38] used by us). The algorithm time (blue) tells the time necessary to compute all algorithms on the card. The overhead time (orange) adds time necessary to do all the supporting actions, mainly establishing the communication with a reader connected to PC and transferring APDUs. All results are arithmetic means of 10 measurements in milliseconds. Compared to VA's implementation of Idemix, our implementation of all proving protocol algorithms on the card is at least 44% faster in all cases, see Fig. 2 for details.

In the case of only 2 attributes stored on the card, our scheme is by 72 % faster than VA's implementation. The card needs only 211 ms to compute the ownership proof for disclosed attributes. The total time of around 360 ms necessary for the whole proof generation on the card including communication with and computations on a terminal (standard PC, Core i7 2.4 GHz, 8 GB RAM) makes the implementation suitable also for time-critical applications like public transportation and ticketing. We also evaluated our scheme using an embedded device (Raspberry Pi 3) instead of the PC as a terminal. Even in that case the total time including overhead was below 450 ms. Based on our benchmarks, we expect that increasing security parameters to the 256-bit EC level would cost acceptable 15 % - 20 % in performance.

Our implementation is artificially limited to 10 attributes per a user, but the smart card's available memory resources (approx. 1.75 KB RAM and 7.5 KB usable EEPROM) would allow storing upto 50 attributes on a single card.

6 Conclusion

Practical anonymous credential schemes are only very few, with implementations on smart cards either too slow or providing insufficient security levels. Our approach to address this problem was twofold: 1) to propose a novel cryptographic scheme that is more efficient than all comparable schemes and formally prove its



(a) 2 attributes in credential

(b) 3 attributes in credential

(c) 4 attributes in credential

(d) 5 attributes in credential

Fig. 2: Speed of our proving protocol compared to Vullers and Alpár (VA) implementation [39]. Blue - our algorithm time, orange - our total time with overhead, verticals - VA algorithm time and horizontals - VA total time with overhead.

security; and 2) to develop a software implementation that is significantly faster than existing implementations, although they use lower security parameters. By achieving these results, we hope that we get privacy-enhanced authentication closer to practical applications.

Our future steps, besides further optimization, are the integration with a suitable revocation scheme (e.g., [13]) and implementation and benchmarks on

higher security levels, hopefully on a wider range of smart cards, if they become available on the market.

7 Acknowledgment

This paper is supported by European Union's Horizon 2020 research and innovation programme under grant agreement No 830892, project SPARTA, the Ministry of Industry and Trade grant # FV20354 and the National Sustainability Program under grant LO1401. For the research, infrastructure of the SIX Center was used.

References

1. Arfaoui, G., Lalande, J.F., Traoré, J., Desmoulins, N., Berthomé, P., Gharout, S.: A practical set-membership proof for privacy-preserving NFC mobile ticketing. PETS'15 Proceedings pp. 25–45 (2015)
2. Arfaoui, G., Lalande, J., Traoré, J., Desmoulins, N., Berthomé, P., Gharout, S.: A practical set-membership proof for privacy-preserving NFC mobile ticketing. PoPETs pp. 25–45 (2015)
3. Barki, A., Brunet, S., Desmoulins, N., Gambs, S., Gharout, S., Traoré, J.: Private eCash in Practice (Short Paper), pp. 99–109 (2017)
4. Barki, A., Brunet, S., Desmoulins, N., Traoré, J.: Improved algebraic MACs and practical keyed-verification anonymous credentials. In: SAC'16 Proceedings (2016)
5. Barki, A., Desmoulins, N., Gharout, S., Traoré, J.: Anonymous attestations made practical. In: ACM WiSec'17 Proceedings. pp. 87–98 (2017)
6. Belenkiy, M., Camenisch, J., Chase, M., Kohlweiss, M., Lysyanskaya, A., Shacham, H.: Randomizable Proofs and Delegatable Anonymous Credentials, pp. 108–125 (2009)
7. Bellare, M., Neven, G.: Multi-signatures in the plain public-key model and a general forking lemma. In: Proceedings of the 13th ACM CCS'06. pp. 390–399 (2006)
8. Bichsel, P., Camenisch, J., Groß, T., Shoup, V.: Anonymous credentials on a standard java card. In: ACM CCS'09 Proceedings. pp. 600–610 (2009)
9. Boneh, D.: The decision diffie-hellman problem. In: Algorithmic Number Theory. pp. 48–63. Springer Berlin Heidelberg (1998)
10. Boneh, D., Boyen, X.: Short signatures without random oracles. In: EUROCRYPT'04. pp. 56–73. Springer Berlin Heidelberg (2004)
11. Boneh, D., Boyen, X.: Short signatures without random oracles and the SDH assumption in bilinear groups. Journal of Cryptology pp. 149–177 (2008)
12. Brands, S.A.: Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy (2000)
13. Camenisch, J., Drijvers, M., Hajny, J.: Scalable revocation scheme for anonymous credentials based on n-times unlinkable proofs. In: ACM CCS WPES'16 Proceedings. pp. 123–133 (2016)
14. Camenisch, J., Dubovitskaya, M., Neven, G.: Oblivious transfer with access control. In: ACM CCS'09 Proceedings. pp. 131–140 (2009)
15. Camenisch, J., Hohenberger, S., Kohlweiss, M., Lysyanskaya, A., Meyerovich, M.: How to win the clonewars: efficient periodic n-times anonymous authentication. In: ACM CCS'06 Proceedings. pp. 201–210 (2006)

16. Camenisch, J., Kohlweiss, M., Soriente, C.: Solving Revocation with Efficient Update of Anonymous Credentials, pp. 454–471 (2010)
17. Camenisch, J., Krenn, S., Lehmann, A., Mikkelsen, G.L., Neven, G., Pedersen, M.Ø.: Scientific comparison of ABC protocols (2014)
18. Camenisch, J., Lysyanskaya, A.: An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation, pp. 93–118 (2001)
19. Camenisch, J., Lysyanskaya, A.: Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials, pp. 61–76 (2002)
20. Camenisch, J., Neven, G., Rückert, M.: Fully Anonymous Attribute Tokens from Lattices, pp. 57–75 (2012)
21. Camenisch, J., Stadler, M.: Efficient group signature schemes for large groups, pp. 410–424 (1997)
22. Camenisch, J., Van Herreweghen, E.: Design and implementation of the idemix anonymous credential system. In: ACM CCS'02 Proceedings. pp. 21–30 (2002)
23. Chase, M., Meiklejohn, S., Zaverucha, G.: Algebraic MACs and keyed-verification anonymous credentials. In: ACM SIGSAC'14 Proceedings. pp. 1205–1216 (2014)
24. Chaum, D.: Security without identification: Transaction systems to make big brother obsolete. Commun. ACM pp. 1030–1044 (1985)
25. Couteau, G., Reichle, M.: Non-interactive keyed-verification anonymous credentials. Cryptology ePrint Archive, Report 2019/117 (2019), `https://eprint.iacr.org/2019/117`
26. Dodis, Y., Kiltz, E., Pietrzak, K., Wichs, D.: Message authentication, revisited. In: EUROCRYPT'12 Proceedings. pp. 355–374 (2012)
27. Hajny, J., Malina, L.: Unlinkable Attribute-Based Credentials with Practical Revocation on Smart-Cards, pp. 62–76 (2013)
28. Hinterwälder, G., Riek, F., Paar, C.: Efficient e-cash with attributes on Multos smartcards. In: RFIDSec'15 Proceedings (2015)
29. Isaakidis, M., Halpin, H., Danezis, G.: UnlimitID: Privacy-Preserving Federated Identity Management Using Algebraic MACs. In: ACM CCS WPES'16 Proceedings. pp. 139–142 (2016)
30. Kerry, C.F., Secretary, A., Director, C.R.: FIPS PUB 186-4 Federal Information Processing Standards Publication: Digital Signature Standard (DSS) (2013)
31. Mostowski, W., Vullers, P.: Efficient U-Prove implementation for anonymous credentials on smart cards. In: International Conference on Security and Privacy in Communication Systems. pp. 243–260 (2011)
32. Paillier, P.: Public-Key Cryptosystems Based on Composite Degree Residuosity Classes, pp. 223–238 (1999)
33. Paquin, C.: U-Prove cryptographic specification v1.1. Tech. rep., Microsoft Corporation (2011)
34. de la Piedra, A., Hoepman, J.H., Vullers, P.: Towards a Full-Featured Implementation of Attribute Based Credentials on Smart Cards (2014)
35. Ringers, S., Verheul, E.R., Hoepman, J.: An efficient self-blindable attribute-based credential scheme. In: FC'17 Proceedings. pp. 3–20 (2017)
36. Rivest, R.L., Kaliski, B.: RSA Problem, pp. 532–536. Springer US (2005)
37. Shoup, V.: Lower Bounds for Discrete Logarithms and Related Problems, pp. 256–266 (1997)
38. Smart, N.: Yearly report on algorithms and keysizes. Tech. rep., Katholieke Universiteit Leuven (2012)
39. Vullers, P., Alpár, G.: Efficient selective disclosure on smart cards using idemix. In: IFIP Working Conference on Policies and Research in Identity Management. pp. 53–67 (2013)

40. Wei, V.K., Yuen, T.H.: More short signatures without random oracles (2005), https://eprint.iacr.org/2005/463

A The Strong Computational DH Inversion Problem

We define the Computational Diffie-Hellman Inversion Problem, which is a computational variant of the SDDHI problem [15].

Definition 3 (n-Strong Computational Diffie-Hellman Inversion Problem (SCDHI)). *Let $\mathcal{O}^{\mathsf{bb}}(\cdot)$ on input $(m_1, \ldots, m_n) \in \mathbb{Z}_q^{*n}$ add $(m_1, \ldots, m_n)$ to Q and output $g^{1/(x_0+\sum_{i=1}^{n} x_i m_i)}$. Let $\mathcal{O}^{\mathsf{dh_i}}(\cdot)$ on input h output h^{x_i}. Define the advantage of $\mathcal{A}$ as follows.*

$$\mathsf{Adv}_{n-SCDHI}(\mathcal{A}) = \mathsf{Pr}\Big[(\mathbb{G}, g, q) \leftarrow \mathsf{GroupSetup}(1^\kappa), (x_0, \ldots, x_n) \xleftarrow{s} \mathbb{Z}_q^{*n+1},$$
$$(y, m_1^*, \ldots, m_n^*) \leftarrow \mathcal{A}^{\mathcal{O}^{\mathsf{bb}}(\cdot), \mathcal{O}^{\mathsf{dh_0}}(\cdot), \ldots, \mathcal{O}^{\mathsf{dh_n}}(\cdot)}(g) : y = g^{\frac{1}{x_0+\sum_{i=1}^{n} x_i m_i^*}} \wedge (m_1^*, \ldots, m_n^* \notin Q\Big].$$

SCDHI is (t, ϵ)-hard if no t-time adversary has advantage at least ϵ.

Theorem 3. *The n-SCDHI problem is hard in the generic group model. More precisely, an adversary working in a generic group of order q with advantage ϵ requires time $\Omega(\sqrt[3]{\epsilon \cdot q})$.*

Proof. This proof is inspired by a proof due to Camenisch et al. [15], who prove similar but slightly different statements in the generic group model. Consider an algorithm $\mathcal{B}$ that interacts with $\mathcal{A}$ as follows. $\mathcal{B}$ maintains a list of pairs $L = \{(F_i, \xi_i) : i = 0, \ldots, \tau - 1\}$, representing the mapping between representations of group elements and their corresponding exponents as rational functions, i.e., fractions where both the numerator and denominator are a polynomial over variables representing the secret signing key $(x_0, \ldots, x_n)$.

It initializes the lists with $L = \{(F_0 = x_0, \xi_0)\}$ and initializes $\tau \leftarrow 1$. The adversary $\mathcal{A}$ is initialized with (ξ_0). $\mathcal{A}$'s queries to perform the group operation, queries to oracles $\mathcal{O}^{\mathsf{bb}}(\cdot)$, and queries to $\mathcal{O}^{\mathsf{dh_i}}(\cdot)$, are handled as follows.

Group Operation: Given two elements ξ_i, ξ_j and $i, j < \tau$, and a bit selecting multiplication or division, $\mathcal{B}$ computes $F_\tau \leftarrow F_i \pm F_j$, where the operation depends on the operation selection bit. If $F_\tau = F_l$ for some $l < \tau$, set $\xi_\tau \leftarrow \xi_l$, otherwise set ξ_τ to a string in $\{0,1\}^*$ distinct from all previous ξ values. Add (F_τ, ξ_τ) to L. Return ξ_τ to $\mathcal{A}$ and increment τ by one if a new element was added.

Queries to $\mathcal{O}^{\mathsf{bb}}$: On input $(m_1, \ldots, m_n) \in \mathbb{Z}_q^{*n}$, let $F_\tau \leftarrow \frac{1}{X_0+\sum X_i m_i}$. If $F_\tau = F_l$ for some $l < \tau$, set $\xi_\tau \leftarrow \xi_l$, otherwise set ξ_τ to a string in $\{0,1\}^*$ distinct from all previous ξ values. Add $\{(F_\tau, \xi_\tau)\}$ to L. Return ξ_τ to $\mathcal{A}$ and increment τ by one if a new element was added.

Queries to $\mathcal{O}^{\mathtt{dh_i}}$: On input ξ_j, let $F_\tau \leftarrow X_i \cdot F_j$. If $F_\tau = F_l$ for some $l < \tau$, set $\xi_\tau \leftarrow \xi_l$, otherwise set ξ_τ to a string in $\{0,1\}^*$ distinct from all previous ξ values. Add $\{(F_\tau, \xi_\tau)\}$ to L. Return ξ_τ to $\mathcal{A}$ and increment τ by one if a new element was added.

After making oracle queries, $\mathcal{A}$ outputs $(\xi_i, m_1^*, \ldots, m_n^*)$, with $i < \tau$. Observe that the representations of all group elements have form

$$F = \alpha + \sum_{i=0}^{n} \sum_{j} \beta_{i,j} x_i^j + \sum_{k} \frac{\gamma_k}{x_0 + \sum_{l=1}^{n} x_l m_{k,l}} \ ,$$

where $\mathcal{A}$ controls the constants $\alpha, \beta_{i,j}, \gamma_i$. This shows that $\mathcal{A}$ cannot output a group element with $F_i = \frac{1}{x_0 + \sum_{k=1}^{n} x_i m_i^*}$.

Only now, we take $(x_0, \ldots, x_n) \xleftarrow{\$} \mathbb{Z}_q^*$ and evaluate all functions. We must show that $\mathcal{B}$ had simulated the operations and oracles correctly.

First, if $\mathcal{A}$ makes a $\mathcal{O}^{\mathtt{bb}}$ query or chooses an output with message $m_1, \ldots, m_n$ but $x_0 + \sum_{i=1}^{n} x_i m_i = 0$, the simulation simulated incorrectly. As the x_i values are chosen randomly after the m_i values were chosen, this probability is $\frac{1}{q-1}$ per query, and in total at most $\frac{t}{q-1}$ for an adversary running in time t.

Next, the simulation considers values with distinct polynomials to be distinct group elements, but after choosing concrete $x_0, \ldots, x_n$, distinct polynomials might evaluate to the same point, meaning the simulation was incorrect: the simulator either gave distinct representations for a single group element, or incorrectly rejected the forgery.

The difference between F and F' can be written as

$$F - F' = \alpha + \sum_{i=0}^{n} \sum_{j=1}^{q_{DH}} \beta_{i,j} x_i^j + \sum_{k=1}^{q_{bb}} \frac{\gamma_k}{x_0 + \sum_{l=1}^{n} x_l m_{k,l}} \ ,$$

for an adversary making at most q_{DH} queries to the Diffie-Hellman oracles and making at most q_{bb} $\mathcal{O}^{\mathtt{bb}}$ queries. $\mathcal{B}$ simulated incorrectly if we have two distinct functions F and F' in L such that

$$F - F' = \alpha + \sum_{i=0}^{n} \sum_{j=1}^{q_{DH}} \beta_{i,j} x_i^j + \sum_{k=1}^{q_{bb}} \frac{\gamma_k}{x_0 + \sum_{l=1}^{n} x_l m_{k,l}} = 0 \ .$$

We can rewrite this to

$$\left(\alpha + \sum_{i=0}^{n} \sum_{j=1}^{q_{DH}} (\beta_{i,j} x_i^j) \right) \prod_{k=1}^{q_{bb}} (x_0 + \sum_{l=1}^{n} x_l m_{k,l}) + \sum_{k=0}^{q_{bb}} \gamma_k \prod_{k'=1, k \neq k'}^{q_{bb}} (x_0 + \sum_{l=1}^{n} x_l m_{k',l}) = 0 \ .$$

For an adversary running in time t, we have $q_{DH} + q_{bb} \leq t$, so this is a polynomial of degree at most t, so the probability it evaluates to zero is $\frac{t}{q-1}$. Summing over all pairs of functions and using $\tau \leq t$, we can bound the probability of a simulation error at $\frac{(t)^3}{q-1}$. Adding the probability of a division by zero, we have an

overall probability of a simulation error $\epsilon \leq \frac{t}{q-1} + (1 - \frac{t}{q-1})\frac{(t)^3}{q-1}$. Using $t < (q-1)$, we can simplify to $\epsilon \leq \frac{t}{q-1} + \frac{(t)^3}{q-1} = O((t)^3/q)$, from which the bounds of the lemma follow.

B Security of Our MAC Scheme

We now prove Theorem 1.

Proof. Reduction $\mathcal{B}$ assumes for contradiction a (t, ϵ)-MAC-forger $\mathcal{A}$ and uses it to break the n-SCDHI problem.

$\mathcal{B}$ receives the group description $par = (\mathbb{G}, g, q)$ from the SCDHI game, and runs $\mathcal{A}$ on input par. $\mathcal{B}$ answers $\mathcal{A}$'s $\mathcal{O}^{\mathsf{MAC}}$ queries on $(m_1, \ldots, m_n)$ by setting $\sigma \leftarrow \mathcal{O}^{\mathsf{bb}}(m_1, \ldots, m_n)$ and setting $\sigma_{x_i} \leftarrow \mathcal{O}^{\mathtt{dh}_1}(\sigma)$ for $i = 1, \ldots, n$, and outputting $(\sigma, \sigma_{x_1}, \ldots, \sigma_{x_n})$. $\mathcal{B}$ answers $\mathcal{A}$'s $\mathcal{O}^{\mathsf{Verify}}$ queries on $(m_1, \ldots, m_n, \sigma)$ by checking $g \stackrel{?}{=} \mathcal{O}^{\mathtt{dh}_0}(\sigma) \cdot \prod_{i=1}^{n} \mathcal{O}^{\mathtt{dh}_1}(\sigma^{m_i})$.

As this simulation is perfect, with probability ϵ $\mathcal{A}$ outputs MAC forgery $(\sigma^*, (m_1^*, \ldots, m_n^*))$ such that it made no MAC query on $(m_1^*, \ldots, m_n^*)$. This means that $\mathcal{B}$ can submit $(\sigma^*, (m_1^*, \ldots, m_n^*))$ to the SCDHI game to win with probability ϵ. Observe that the runtime of $\mathcal{B}$ is approximately the runtime of $\mathcal{A}$, as $\mathcal{B}$ only makes oracle queries, showing that $\mathcal{B}$ (t, ϵ)-breaks n-SCDHI, contradicting the (t, ϵ)-hardness of n-SCDHI.

C Security of Our Keyed-Verification Credentials

C.1 Formal Security Model

Here we recall the definition of security of keyed-verification credentials, as defined by Chase et al. [23]. The definition of security considers the setting without blind issuance, meaning that the issuer sees all the attribute values. They later show how to extend to include blind issuance, but in this definition of security we limit ourselves to open issuance. We consider security properties *correctness*, *unforgeability*, *anonymity*, and *key-parameter consistency*.

Definition 4. *Let Φ be the set of statements supported by a credential system, and $\mathcal{U}$ be the universe of attribute sets. Then a keyed-verification credential system* $(\mathsf{CredKeygen}, \mathsf{Issue}, \mathsf{CredVerify}, \mathsf{Show}, \mathsf{ShowVerify})$ *is* correct *for Φ, $\mathcal{U}$, if for all $(m_1, \ldots, m_n) \in \mathcal{U}$, for all κ,*

$$\Pr\Big[par \xleftarrow{\$} \mathsf{Setup}(1^\kappa), (ipar, sk) \xleftarrow{\$} \mathsf{CredKeygen}(par), \\ cred \xleftarrow{\$} \mathsf{Issue}(sk, (m_1, \ldots, m_n)) : \mathsf{CredVerify}(sk, (m_1, \ldots, m_n) = 0)\Big] = 0 \ ,$$

and for all $\phi \in \Phi$, $(m_1, \ldots, m_n) \in \mathcal{U}$ such that $\phi(m_1, \ldots, m_n) = 1$, for all κ,

$$\Pr\Big[par \xleftarrow{\$} \mathsf{Setup}(1^\kappa), (ipar, sk) \xleftarrow{\$} \mathsf{CredKeygen}(par), cred \xleftarrow{\$} \mathsf{Issue}(sk, (m_1, \ldots, m_n)), \\ \mathsf{Show}(ipar, cred, (m_1, \ldots, m_n), \phi) \leftrightarrow \mathsf{ShowVerify}(sk, \phi) \rightarrow b : b = 0\Big] = 0 \ .$$

Definition 5. *A keyed-verification credential system* (CredKeygen, Issue, CredVerify, Show, ShowVerify) *is* (t, ϵ, q_H)*-unforgeable there exists no adversary* $\mathcal{A}$*, running in time* t *and making at most* q_H *random oracle queries, for which the following probability is at least* ϵ*:*

$$\Pr\Big[par \xleftarrow{\$} \mathsf{Setup}(1^\kappa), (ipar, sk) \xleftarrow{\$} \mathsf{CredKeygen}(par),$$
$$(\mathsf{st}, \phi) \xleftarrow{\$} \mathcal{A}(par, ipar)^{\mathcal{O}^{\mathsf{Issue}(\mathsf{sk},\cdot)}, \mathcal{O}^{\mathsf{ShowVerify}(\mathsf{sk},\cdot)}}, \mathcal{A}(\mathsf{st}) \leftrightarrow \mathsf{ShowVerify}(sk, \phi) \to b : b = 1,$$
$$with\ (\forall (m_1, \ldots, m_n) \in Q, \phi(m_1, \ldots, m_n) = 0\Big] \ .$$

Definition 6. *A keyed-verification credential system* (CredKeygen, Issue, CredVerify, Show, ShowVerify) *is* anonymous *if for all PPT adversaries* $\mathcal{A}$*, there exists an efficient algorithm* SimShow *such that for all* κ*, for all* $\phi \in \Phi$ *and* $(m_1, \ldots, m_n) \in \mathcal{U}$ *such that* $\phi(m_1, \ldots, m_n) = 1$*, and for all* $par \xleftarrow{\$} \mathsf{Setup}(1^\kappa)$ *and all* $(ipar, sk) \xleftarrow{\$} \mathsf{CredKeygen}(par)$*, for all cred such that* $\mathsf{CredVerify}(sk, (m_1, \ldots, m_n), cred) = 1$*:*

$$\mathsf{Show}(ipar, cred, (m_1, \ldots, m_n), \phi) \leftrightarrow \mathcal{A} \to \mathsf{st} \approx \mathsf{SimShow}(ipar, sk, \phi) \ ,$$

i.e., the adversary's view given the proof can be simulated by SimShow *given only* ϕ *and a valid secret key corresponding to ipar.*

Definition 7. *A keyed-verification credential system* (CredKeygen, Issue, CredVerify, Show, ShowVerify) *is* key-parameter consistent *if for any PPT adversary* $\mathcal{A}$*, the probability that* $\mathcal{A}$ *given* $par \xleftarrow{\$} \mathsf{Setup}(1^\kappa)$ *can produce* $(ipar, sk_1, sk_2)$ *with* $sk_1 \neq sk_2$ *such that* $(ipar, sk_1)$ *and* $(ipar, sk_2)$ *are both in the range of* $\mathsf{CredKeygen}(par)$ *is negligible in* κ *(where the probability is taken over the choice of par and the random coins of* $\mathcal{A}$*).*

C.2 Proof of Theorem 2

We now formally prove that our keyed-verfication credential scheme is secure. The scheme supports selective disclosure, meaning that $\phi \in \Phi$ describes indices of disclosed attributes D, and the attribute values m_i for $i \in D$. The attribute set $\mathcal{U}$ is $(m_1, \ldots, m_n)$, where $m_i \in \mathbb{Z}_q^*$.

Lemma 1. *Our KVAC scheme is correct, as defined in Definition 4.*

Proof. This follows directly from the completeness of our MAC scheme and the completeness of the zero knowledge proofs.

Lemma 2. *Our KVAC scheme is* (t, ϵ, q_H)*-unforgeable, as defined in Definition 5, if the n-SCDHI problem is* $(2t, \epsilon')$*-hard, where* $\epsilon' = \epsilon \cdot (\frac{\epsilon}{q_H} - \frac{1}{q})$*.*

Proof. Assume for contradiction a (t, ϵ, q_H)-forger $\mathcal{A}$. We prove this lemma by contstructing a reduction $\mathcal{B}$ that attacks the n-SCDHI problem. $\mathcal{B}$ receives paramters $par = (\mathbb{G}, g, q)$ from the SCDHI game and computes $ipar = (X_0, \ldots, X_n)$, using $\mathcal{O}^{\mathtt{dh}_i}(g)$ to compute X_i. It invokes $\mathcal{A}$ on input $(par, ipar)$. $\mathcal{B}$ answers

$\mathcal{A}$'s Issue queries on message $(m_1, \ldots, m_n)$ by setting $\sigma \leftarrow \mathcal{O}^{\mathtt{bb}}(m_1, \ldots, m_n)$, $\sigma_{x_i} \leftarrow \mathcal{O}^{\mathtt{dh}_i}(\sigma)$ for $i = 1, \ldots, n$, and simulates the proof π by programming the random oracle.

$\mathcal{B}$ answers $\mathcal{A}$'s $\mathsf{ShowVerify}$ queries by using the Diffie-Hellman oracles to compute $\hat{\sigma}_i = \hat{\sigma}^{x_i}$ for $i = 0, \ldots, n$, computes $c \leftarrow \mathcal{H}(D, \langle m_i \rangle_{i \in D}), t, \hat{\sigma}, par, ipar, nonce)$ and checks $t \stackrel{?}{=} \prod_{i \notin D} \hat{\sigma}_{x_i}^{s_{m_i}} g^{s_r} \cdot (\hat{\sigma}_{x_0} \prod_{i \in D} \hat{\sigma}_{x_i}^{m_i})^{-c}$.

As the simulation is perfect, with probability ϵ $\mathcal{A}$ wins the unforgeability game by running a $\mathsf{ShowVerify}$ sending $\hat{\sigma}, \langle \hat{\sigma}_{x_i} \rangle_{i=1}^n, t, s_r, \langle s_{m_i} \rangle_{i \notin D}$ to the verifier, proving it possessess attributes $\langle m_i \rangle_{i \in D}$ without having made a Issue query on inputs containing those attributes. By applying the forking lemma [7], we can extract the witness r and the hidden attributes m_i from the zero-knowledge proof using rewinding, such that we have $\sigma = g^{1/(x_0 + \sum_{i=1}^n m_i x_i)}$ for $\sigma \leftarrow \hat{\sigma}^{1/r}$. By the bounds of the forking lemma, this attacker forges a MAC with probability $\epsilon' = \epsilon \cdot (\frac{\epsilon}{q_H} - \frac{1}{q})$ running in time $2t$.

Lemma 3. *Our KVAC scheme is anonymous, as defined in Definition 6, in the random oracle model.*

Proof. We define $\mathsf{SimShow}(ipar, sk, (D, \langle m_i \rangle_{i \in D}))$ as follows. It takes random $\hat{\sigma} \xleftarrow{\$} \mathbb{G}$ and simulates the corresponding zero knowledge proof by taking s_r, s_{m_i} for $i \notin D$, and c all uniformly at random in $\mathbb{Z}_q$. It computes

$$t \leftarrow g^{s_r} \cdot \hat{\sigma}^{-c \cdot x_0 + \sum_{i \notin D}(x_i \cdot s_{m_i}) - \sum_{i \in D}(x_i \cdot m_i \cdot c)} \quad .$$

Then, it programs the random oracle such that $c = \mathcal{H}(D, \langle m_i \rangle_{i \in D}), t, \hat{\sigma}, par, ipar, nonce)$. Clearly, the output of $\mathsf{SimShow}$ is equally distributed to the output from the real prover.

Lemma 4. *Our KVAC scheme is key-parameter consistent, as defined in 7.*

Proof. $\mathsf{CredKeygen}$ outputs $sk = (x_0, \ldots, x_n) \in \mathbb{Z}_q^*$ and $ipar = (X_0, \ldots, X_n)$ with $X_i = g^{x_i}$. As we are working in a prime order group, every element has a unique discrete logarithm, so there is a unique sk corresponding to every $ipar$.