Efficient Attributes for Anonymous Credentials

JAN CAMENISCH and THOMAS GROSS
IBM Research, Zurich Research Laboratory

We extend the Camenisch-Lysyanskaya anonymous credential system such that selective disclosure of attributes becomes highly efficient. The resulting system significantly improves upon existing approaches, which suffer from a linear complexity in the total number of attributes. This limitation makes them unfit for many practical applications, such as electronic identity cards. Our system can incorporate an large number of binary and finite-set attributes without significant performance impact. Our approach compresses all such attributes into a single attribute base and, thus, boosts the efficiency of *all* proofs of possession. The core idea is to encode discrete binary and finite-set values as prime numbers. We use the divisibility property for efficient proofs of their presence or absence. We contribute efficient methods for conjunctions and disjunctions, in addition. The system builds on the Strong-RSA assumption.

We demonstrate the aptness of our method in realistic application scenarios, such as electronic identity cards and complex/structured credentials. Our method has crucial advantages in devices with restricted computational capabilities, such as smartcards and cell phones.

Categories and Subject Descriptors: E.3 [**Data**]: Data Encryption—*Public key cryptosystems*

General Terms: Algorithms, Performance, Security

Additional Key Words and Phrases: Anonymous credential systems, Camenisch-Lysyanskaya, cryptographic protocols, efficient attribute encoding, Identity Mixer, privacy, private certificate systems

Authors' address: J. Camenisch and T. Groß, IBM Research, Zurich Research Laboratory, CH-8803 Rüschlikon, Switzerland, {jca,tgr}@zurich.ibm.com

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/ 2007-2013) under grant agreement n° 216483.

1. INTRODUCTION

The rise of user-centric identity management amplifies the need for a combination of strong security and privacy protection. *Anonymous credential systems* are one of the most promising answers to this need. They allow users to selectively prove statements about their identity attributes while keeping the corresponding data hidden.

Industry does not only aim at employing anonymous credential systems on desktop PCs but also on small devices with very limited computational power. Examples include cell phones and corporate or government-issued electronic identity cards. In these environments, one fundamental complexity restriction amounts to a limiting factor: The complexity of a proof of possession is linear in the total number of attributes in the credential [Brands 1999; Camenisch and Lysyanskaya 2001].

European electronic identity cards, for instance, often contain several attributes: Nationality, sex, civil status, hair and eye color, and applicable minority status[1]. These attributes are either binary or discrete values from pre-defined finite sets. They constitute the lion's share of the encoded attributes. These attributes are partially highly privacy-sensitive and require a selective disclosure of one attribute while hiding others completely. The traditional encoding renders anonymous credential systems impractical for implementation on small devices, e.g., electronic identity cards. We therefore focus on new and efficient means to encode binary and finite-set attributes.

There exist two prior approaches for encoding binary or finite-set attributes in anonymous credential systems: First, encoding each binary attribute in one attribute base (i.e., as one exponent in a discrete logarithm representation). We call this method *traditional encoding*. This method is directly impacted by the mentioned complexity restriction: proofs of possession are linear in the total number of attribute bases. Therefore, each binary attribute burdens *all* credential transactions. This traditional approach denies applications with small devices and a significant number of binary/finite-set attributes.

A second prior approach encodes binary attributes as bit vector in one attribute base. Clearly, this approach limits the number of attribute bases required. It therefore circumvents the linear computational complexity in the total attribute number. However, as soon as a user reveals some of the attributes in the bit vector the complexity is linear again. It is either impacted by the total number of (binary) attributes concerned or by the length of the bit vector, depending on the particular implementation. Hence this approach is also unacceptable for small devices.

We extend the Camenisch-Lysyanskaya credential system [Camenisch and Lysyanskaya 2001; 2003] with a finite-set encoding. It enables the efficient selective disclosure of binary and discrete-values attributes. This method overcomes the severe limitations of existing schemes. We require a solution with two key properties: (1) It only uses *at most one* attribute base for all binary and finite-set attributes. (2) It only impacts the proof complexity by the *number of used attributes* instead of the total number. Our extension provides a highly efficient toolkit of attribute proofs as well as AND, OR, and NOT proofs over binary or finite-set attributes. Our approach has a constant complexity in the number of encodable attributes. It is only restricted by space considerations for the attribute exponent length in the credential and the size of the issuer's public key.

This is the core idea of our paper: We encode binary attributes as well as discrete values

[1] such as blind, partially sighted, spectacles wearer, or hearing impaired

of finite sets as product of prime numbers in a single attribute base. We use the *coprime* property to efficiently prove the attributes' presence and absence. We also employ this property to facilitate conjunction and disjunction proofs. Where earlier schemes prove equality of exponents, we prove divisibility. The efficiency of our scheme surpasses any existing encoding of binary and finite-set attributes.

We note that other cryptographic primitives with privacy protection can also benefit from our approach, particularly group signatures, blind signatures, and electronic voting schemes.

We structure the remainder of this paper as follows: Section 2 covers related literature for anonymous credential systems as well as existing methods for encoding binary attributes. Section 3 contains preliminary definitions including the Camenisch-Lysyanskaya credential system. We define our prime encoding extensions for binary and finite-set attributes in Section 4. Section 4.1 contains the attribute representation in CL signatures, followed by setup and encoding paradigm. We treat proofs with AND, OR, and NOT statements in Section 4.4. We analyze the complexity of our scheme compared to existing approaches in Section 5. Section 6 governs possible application scenarios such as electronic identity cards and role-based access control. We conclude the paper in Section 7. Our Appendix digresses in three areas: We provide a formal protocol specification for setup and issuing in Appendix Section A. Appendix Section B elaborates on the impact of the bitlength of the prime encoding. Appendix Section C considers professional taxonomies as additional application example.

2. RELATED WORKS

Chaum pioneered privacy-enhancing cryptographic protocols that minimize the amount of personal data disclosed. His work put forth the principles of anonymous credentials [Chaum 1981; 1985; Chaum and Evertse 1987], group signatures [Chaum and van Heyst 1991], and electronic cash [Chaum 1983]. These concepts have in common that some party issue a digital signatures where the message signed includes information about the user (i.e., attributes). Subsequently, more efficient implementation of these concepts were proposed and a number of related concepts where introduced, including, group signatures [Ateniese et al. 2000; Boneh et al. 2004; Kiayias and Yung 2006], e-cash [Brands 1993; Camenisch et al. 2005; Frankel et al. 1998], anonymous credentials [Brands 1995a; 1995b; 1999; Camenisch and Lysyanskaya 2001; 2004], traceable signatures [Kiayias et al. 2004], anonymous auctions [Naor et al. 1999], and electronic voting based on blind-signatures [Fujioka et al. 1992]. Many of these schemes use as building blocks signed attributes and protocols that selectively reveal these attributes or prove properties about them. The given implementations typically encode attributes as a discrete logarithm or, more generally, as an element (exponent) of a representation of a group element, resulting in protocols where the number of group elements transmitted and the commutations performed are linear in the number of encoded attributes.

There are also some works [Cramer et al. 1994; Brands 1997; Camenisch and Michels 1999; Boudot 2000; Fujisaki and Okamoto 1997] where the respective authors propose zero-knowledge proof techniques for proving AND, OR and NOT statement about attributes encoded as discrete logarithms, e.g., "a user has attribute a OR b," basically by showing that some committed value equals a given value OR some other given value.

We significantly improve on the existing works: our proof-protocols for showing that some given attribute value is encoded in a credential is much more efficient. Also, the proof protocols for proving AND, OR, and NOT statements are far more efficient than the known ones.

3. PRELIMINARIES

3.1 Assumptions

Strong RSA Assumption [Rivest et al. 1978; Fujisaki and Okamoto 1997]: Given an RSA modulus n and a random element $g \in \mathbb{Z}_n^*$, it is hard to compute $h \in \mathbb{Z}_n^*$ and integer $e > 1$ such that $h^e \equiv g \bmod n$. The modulus n is of a special form pq, where $p = 2p' + 1$ and $q = 2q' + 1$ are safe primes. Other primitives, such as the Fiat-Shamir heuristic to generate signatures from zero-knowledge proofs of knowledge, may require additional assumptions. This is orthogonal to the credential system itself.

3.2 Integer Commitments

Recall the Pedersen commitment scheme [1992], in which the public parameters are a group G of prime order q, and generators $(g_0, \ldots, g_m)$. In order to commit to the values $(v_1, \ldots, v_m) \in \mathbb{Z}_q^m$, pick a random $r \in \mathbb{Z}_q$ and set $C = \mathsf{Com}(v_1, \ldots, v_m; r) = g_0^r \prod_{i=1}^m g_i^{v_i}$.

Damgård and Fujisaki [2001] show that if the group G is an RSA group and the committer is not privy of the factorization of the modulus, then in fact the Pedersen commitment scheme can be used to commit to *integers* of arbitrary size.

3.3 Known Discrete-Logarithm-Based, Zero-Knowledge Proofs

In the common parameters model, we use several previously known results for proving statements about discrete logarithms, such as (1) proof of knowledge of a discrete logarithm modulo a prime [Schnorr 1991] or a composite [Damgård and Fujisaki 2001; Fujisaki and Okamoto 1997], (2) proof of knowledge of equality of representation modulo two (possibly different) prime [Chaum and Pedersen 1993] or composite [Camenisch and Michels 1999] moduli, (3) proof that a commitment opens to the product of two other committed values [Brands 1997; Camenisch and Michels 1999; Camenisch 1998], (4) proof that a committed value lies in a given integer interval [Boudot 2000; Camenisch and Michels 1999; Chan et al. 1998], and also (5) proof of the disjunction or conjunction of any two of the previous [Cramer et al. 1994]. These protocols modulo a composite are secure under the strong RSA assumption and modulo a prime under the discrete logarithm assumption.

When referring to the proofs above, we will follow the notation introduced by Camenisch and Stadler [1997] for various proofs of knowledge of discrete logarithms and proofs of the validity of statements about discrete logarithms. For instance,

$$PK\{(\alpha, \beta, \delta) : y = g^\alpha h^\beta \wedge \tilde{y} = \tilde{g}^\alpha \tilde{h}^\delta \wedge (u \le \alpha \le v)\}$$

denotes a "*zero-knowledge Proof of Knowledge of integers α, β, and δ such that $y = g^\alpha h^\beta$ and $\tilde{y} = \tilde{g}^\alpha \tilde{h}^\delta$ holds, where $u \le \alpha \le v$,*1" where $y, g, h, \tilde{y}, \tilde{g}$, and $\tilde{h}$ are elements of some groups $G = \langle g \rangle = \langle h \rangle$ and $\tilde{G} = \langle \tilde{g} \rangle = \langle \tilde{h} \rangle$. The convention is that Greek letters denote quantities of which knowledge is being proven, while all other values are known to the verifier. We apply the Fiat-Shamir heuristic [Fiat and Shamir 1987] to turn such proofs of knowledge into signatures on some message m; denoted as, e.g., $SPK\{(\alpha) : y = g^\alpha\}(m)$.

Given a protocol in this notation, it is straightforward to derive actual protocol implementing the proof. Indeed, the computational complexities of the proof protocol can be easily derived from this notation: basically for each term $y = g^{\alpha} h^{\beta}$, the prover and the verifier have to perform an equivalent computation, and to transmit one group element and one response value for each exponent. With statement such as $(u \leq \alpha \leq v)$ we denote interval checks which are basically free [Camenisch and Michels 1999; Chan et al. 1998] but are not tight (however, good enough if the non-tightness can be accounted for as in our application). We note that this exclude the interval proof protocol as the one by [Boudot 2000] that are tight but computationally costly, i.e., they require the prover to provide a number of so-called integer commitments and to prove relations among them.

3.4 Camenisch-Lysyanskaya Signatures

Let us recall Camenisch-Lysyanskaya signatures [2003] (we present a slight and straightforward variant which allows messages to be negative integers as well). Let ℓ_m, ℓ_e, ℓ_n, ℓ_r and L be system parameters. ℓ_r is a security parameter, the meanings of the others will become apparent soon.

Throughout the paper, we denote by $\{0,1\}^{\ell_m}$ the set of integer $\{-(2^{\ell_m}-1), ..., 2^{\ell_m}-1)\}$. Element of this set can thus be encoded as binary strings of length ℓ_m plus an additional bit carrying the sign, i.e, $\ell_m + 1$ bits in total.

Key generation. On input ℓ_n, choose an ℓ_n-bit RSA modulus n such that $n = pq$, $p = 2p' + 1$, $q = 2q' + 1$, where p, q, p', and q' are primes. Choose, uniformly at random, $R_0, \ldots, R_{L-1}, S, Z \in \mathrm{QR}_n$. Output the public key $(n, R_0, \ldots, R_{L-1}, S, Z)$ and the secret key p.

Message space is the set $\{(m_0, \ldots, m_{L-1}) : m_i \in \pm\{0,1\}^{\ell_m}\}$.

Signing algorithm. On input $m_0, \ldots, m_{L-1}$, choose a random prime number e of length $\ell_e > \ell_m + 2$, and a random number v of length $\ell_v = \ell_n + \ell_m + \ell_r$, where ℓ_r is a security parameter. Compute $A = (\frac{Z}{R_0^{m_0} \ldots R_{L-1}^{m_{L-1}} S^v})^{1/e} \bmod n$. The signature consists of (e, A, v).

Verification algorithm. To verify that the tuple (e, A, v) is a signature on message $(m_0, \ldots, m_{L-1})$, check that $Z \equiv A^e R_0^{m_0} \ldots R_{L-1}^{m_{L-1}} S^v \pmod{n}$, $m_i \in \pm\{0,1\}^{\ell_m}$, and $2^{\ell_e} > e > 2^{\ell_e - 1}$ holds.

THEOREM 3.1. *[Camenisch and Lysyanskaya 2003] The signature scheme is secure against adaptive chosen message attacks [Goldwasser et al. 1988] under the strong RSA assumption.*

Proving Knowledge of a Signature. Let us further recall how a prover can prove that she possesses a CL signature without revealing any other information about the signature.

Of course we want to use the protocols described in §3.3. Now, if A was a public value, we could do so by proving knowledge representation of Z w.r.t. R_0, …, R_{L-1}, S, and A. Obviously making A public would destroy privacy as that would make all transaction linkable. Luckily, one can randomize A: Given a signature (A, e, v), the tuple $(A' := AS^{-r} \bmod n, e, v' := v + er)$ is also a valid signature as well. Now, provided that $A \in \langle S \rangle$ and that r is chosen uniformly at random from $\{0,1\}^{\ell_n+\ell_\varnothing}$, the value A' is distributed statistically close to uniform over $\mathbb{Z}_n^*$. Thus, the user could compute a fresh A' each time, reveal it, and then run the protocol

$$\begin{aligned} &PK\{(\varepsilon, \nu', \mu_0, \ldots, \mu_{L-1}) : \\ &\quad Z \equiv \pm R_0^{\mu_0} \cdots R_{L-1}^{\mu_{L-1}} A'^{\varepsilon} S^{\nu'} \pmod{n} \land \\ &\quad \mu_i \in \pm\{0,1\}^{\ell_m} \ \land \ \varepsilon \in [2^{\ell_e - 1}+1, 2^{\ell_e}-1]\} \end{aligned}$$

Now, there is a technical consequence from this proof protocol regarding the statements $\mu_i \in \pm\{0,1\}^{\ell_m} \ \land \ \varepsilon \in [2^{\ell_e-1}+1, 2^{\ell_e}-1]$. While these can be implemented virtually for free, they requires that the actually secret lie in a smaller interval, i.e., the signer needs to choose e from $[2^{\ell_e-1} - 2^{\ell'_e}+1, 2^{\ell_e-1}+2^{\ell'_e}-1]$ with $\ell'_e < \ell_e - \ell_\varnothing - \ell_\mathcal{H} - 3$, where $\ell_\varnothing$ and $\ell\mathcal{H}$ are security parameters (the first controlling statistical zero-knowledge and the second one being the size of the challenge message in the *PK* protocol). Similarly, we require $m_i \in \pm\{0,1\}^{\ell_m - \ell_\varnothing - \ell_\mathcal{H} - 2}$ when input to the signature scheme (cf. [Camenisch and Lysyanskaya 2003]). As the proofs can only guarantee that the absolute value of the messages are smaller then 2^{ℓ_m} we also include negative message in the message space for consistency. Finally, we note that in $Z \equiv \pm R_0^{\mu_0} \cdots R_{L-1}^{\mu_{L-1}} A'^{\varepsilon} S^{\nu'} (\bmod\ n)$ there appears a ±. This is a technicality in the used proofs of knowledge in RSA. While this is not a problem for the application at hand, we refer to the standard literature for details.

3.5 Traditional Encoding of Attributes and Proofs About Them

We discuss how attributes are typically encoded in privacy-enhancing primitives found in the literature. While we focus on CL signatures, the principles are the same in other settings.

As we have seen in the previous subsection, CL signatures allow one to sign several messages such that later on one can prove knowledge of a signature without revealing the signature nor the signed messages. Now, if each attribute is encoded as a single message, revealing these attributes selectively can be done using the zero-knowledge protocol discussed above as follows. Assume for example that messages m_1 and m_3 encode the attributes that one wants to reveal. Then the protocol will be as follows:

$$\begin{aligned} &PK\{(\varepsilon, \nu', \mu_0, \mu_2, \mu_4, \ldots, \mu_{L-1}) : \\ &\quad Z R_1^{-m_1} R_3^{-m_3} \equiv \pm R_0^{\mu_0} R_2^{\mu_2} R_4^{\mu_4} \cdots R_{L-1}^{\mu_{L-1}} A'^{\varepsilon} S^{\nu'} \pmod{n} \land \\ &\quad \mu_i \in \pm\{0,1\}^{\ell_m} \ \land \ \varepsilon \in [2^{\ell_e-1}+1, 2^{\ell_e}-1]\} \end{aligned}$$

If one wanted to encode binary attributes (or attributes that take on only a few fixed values), this approach of selectively revealing attributes requires one to encode each attribute value (a single or only a few bits) as a full ℓ_m-bit message. Thus, if a large number of such attributes shall be encoded, this approach would not be very efficient: The computation and the number of group elements to be exchanged in the proof protocol would be linear in the number of attributes.

As an alternative, one could encode all the binary attributes as a bit vector and then include that vector in a single message (or several ones, depending on the number of attributes). The resulting proof for revealing attributes selectively seems in general to require to commit to each individual attribute, proving that each commitment indeed only commits to a binary value, prove that these committed values correspond to the certified message, and then to reveal the attribute by opening the commitments. The communication and computational complexities of the resulting protocol would be even worse than of the one discussed priorly.

Now, if only a single attribute should be revealed, one could obtain a more efficient proof protocol as follows. That is, if the vector looked as $(a||b||c)$, where b is the representation of the attribute that shall be revealed and a and c are the representations of the remaining attributes, one would provide a commitment to a and c and then prove (1) that these have the right binary length and (2) that $(a||b||c)$ indeed is signed by the issuer (without revealing a or c of course). A proof that a commitment contains a ℓ-bit number can be done relatively efficient with the range proof proposed by Boudot [2000]. While this solution provides a proof protocol that does not depend on the number of attributes encoded, each range proofs require about 12 commitments plus some additional values, and one needs to do $l + 1$ of them if l attributes are revealed. Even if one wanted to apply this for revealing a single attribute, this would in practice not be considered efficient anymore. This holds in particular if one considers using anonymous credentials for electronic identity cards. Indeed, for the scenarios we consider in the application section, it would even be more efficient to use the other methods we discussed.

4. EFFICIENT ATTRIBUTES FOR CL

In this section we, provide the means to efficiently encode a number of attributes into an anonymous credential. We focus on the computational cost when issuing and using a credential (note that the communication cost when using a credential is directly related to the computational cost—hence we only consider the latter). That is, we show how to encode attribute values as (small) prime values such that a number of attributes can be encoded into a single message m_j. We also show how one can reveal these attribute values selectively and how one can prove simple statements about them (OR, AND, and NOT connectives). Our basic idea is in fact very simple: we set m_j equal to the product of the primes corresponding the values of the different attributes. Now that allows us to show that an attribute is set to a given value encoded by, say prime e_j by proofing that e_j divides the message contained in the credential. We can also show that it is not set to the given value by showing that e_j does not divide the message. Realizing OR statements, i.e., that the credential encoded either e_j or e_l can be done by proving that there exists a value that divides both the product of e_j and e_l as well as the message contained in the credential. As we will see, this idea gives us very efficient proof statements and leads to, e.g., an efficient implementation of an electronic identity card.

While we present the method for encoding attributes in context of the RSA-based CL credential system [Camenisch and Lysyanskaya 2001; 2003], it can be applied as well to other anonymity related schemes such as group signatures, e-cash systems, or voting schemes.

4.1 The CL Credential System and Attributes

We provide an explanation of how the Camenisch-Lysyanskaya (CL) credential system [Camenisch and Lysyanskaya 2001; 2003] works and how attributes can be encoded into credentials. This will make it clear how to use the results presented in the remainder of this section to build a fully fledge credential systems with all other features with which the basic system has been extended over the years and as described in the literature (e.g., revocation [Camenisch and Lysyanskaya 2002], k-spendability [Camenisch et al. 2005], clone protection [Camenisch et al. 2006]).

In the CL credential system each user has a secret identity, i.e., a *single* secret key s_U. In contrast to how credentials are issued to the user in a traditional PKI (e.g, X.509), an issuing party now uses the CL signature scheme to sign the user's secret key as well as all attributes the issuer wants to assert about the user. This signing is of course done in a "blind" way such the issuer does not learn the user's secret key (cf. [Camenisch and Lysyanskaya 2001; 2003]). Thus, the user will have obtained a signature (A, e, v) such that $Z \equiv \pm R_0^{s_u} R_1^{a_1} \cdots R_{L-1}^{a_{L-1}} A'^{\varepsilon} S^{\nu'} (\bmod\ n)$ holds, where $a_1, \ldots, a_{L-1}$ are the attested attributes and $(Z, R_0, \ldots, R_{L-1}, S, n)$ are the issuer's public key. How a user can show that she obtained a credential from some issuer and selectively reveal some of the attributes (or prove statement about them, e.g., my attested birth date lies further in the past than 21 years) using the proof of knowledge of a signature that we recalled in the previous section.

As discussed, our approach to achieve efficient encoding and proving of attributes for the CL credential system, we are going to encode products of primes into a user's credential, e.g., we set a_1 as the product of the relevant primes e_j. Thus, it remains to show how the user can selectively reveal attribute values encoded like this, that one out of a list of attribute values is encoded, or that an attribute value is not encoded into her credential.

4.2 Set Up

The issuer performs the following setup. On input ℓ_n, choose an ℓ_n-bit RSA modulus n such that $n = pq$, $p = 2p' + 1$, $q = 2q' + 1$, where p, q, p', and q' are primes. Choose, uniformly at random, $R_0, \ldots, R_{L-1}, S, Z \in \mathrm{QR}_n$. In addition, we require bases g and h for an integer commitment. For this, we can use the signer's RSA modulus as well, thus, let h and g be element of QR_n. We establish a further group for our new OR-proofs. It has prime order $\mathfrak{q}$. We choose two generators $\mathfrak{g}$ and $\mathfrak{h}$ of that group such that $\log_{\mathfrak{h}} \mathfrak{g}$ is unknown.

The public key becomes $(n, R_0, \ldots, R_{L-1}, S, Z, g, h, \mathfrak{g}, \mathfrak{h})$.

4.3 Encoding

The number of bits we can encode into a message field of a CL signature is ℓ_m as described in the previous section. Now assume we wants to encode t attributes into a single message filed. Thus we can only use primes of length up to ℓ_m/t. Now, if we each attributes takes at most k different values, then we need choose our ℓ_m such that there exist tk primes smaller than $2^{\ell_m/t}$ (or, alternatively, choose t and encode the attributes into two or more messages). Let $\ell_t < \ell_m/t$ be the length of the primes that we will be using. See also Appendix Section B

Assume we want to encode the attribute vector $(a_1, ..., a_t)$ with $a_i \in \{1, \ldots, k\}$ and that we have enumerated all the primes $2 < e_i < 2^{\ell_m/t}$. Now we encode $(a_1, ..., a_t)$ by including the value $E = \prod_{j=1}^{t} e_{((j-1)k+a_j)}$ in the credential. This means that the product E will be one of the messages that the issuer signs. (Here we assumed that each attribute takes k different values — adapting the construction to cases where some attributes take fewer values is straightforward.)

4.4 Proofs About Attributes

We now assume that the user (prover) has obtained a CL credential containing E, i.e., signature (A, e, v) on messages m_0 and m_1 with $m_1 = E$. (The attribute m_0 typically encodes the user's secret key [Camenisch and Lysyanskaya 2003]).

Efficiently proving that a credential contains an attribute with a given value. Let us first discuss has the user can convince the verifier that E encodes a given attribute, e.g, how she can prove that her identity card states that her hair color is blond. Assume that the attribute *hair color blond* is encoded by the prime e_j. Thus to convince the verifier that she got issues a credential with this attribute, i.e., that e_j divides the E included in her credential, the user engages with the following proof with the verifier:

$$\begin{aligned}&PK\{(\varepsilon, \nu', \mu_0, \mu_1') : \\ &\quad Z \equiv \pm R_0^{\mu_0} (R_1^{e_j})^{\mu_1'} A'^{\varepsilon} S^{\nu'} \pmod{n} \wedge \\ &\quad \mu_0 \in \pm\{0,1\}^{\ell_m} \ \wedge \ \mu_1' \in \pm\{0,1\}^{\ell_m - \ell_t} \wedge \\ &\quad \varepsilon \in [2^{\ell_e - 1} + 1, 2^{\ell_e} - 1]\}\end{aligned}$$

THEOREM 4.1. *If a prover is successful in the above protocol, he was issued a credential encoding the attribute corresponding to e_j.*

PROOF. It is standard to show that there exists a knowledge extractor who can extract from a convincing prover values ε, ν', μ_0, μ_1' such that $Z \equiv tR_0^{\mu_0}(R_1^{e_j})^{\mu_1'} A'^{\varepsilon} S^{\nu'} \pmod{n}$ holds for some t (see, e.g., [Damgård and Fujisaki 2001]). Moreover, as we have chosen n as the product of two safe primes, t must be ± 1. Now, as CL signatures are unforgeable we can conclude that there must exist some E such that $Z \equiv \pm R_0^{\mu_0} R_1^E A'^{\varepsilon} S^{\nu'} \pmod{n}$. Thus, we have $R_1^E = R_1^{e_j \mu_1'} \pmod{n}$ from which we can conclude that $E \equiv e_1 \mu_1' \pmod{p'q'}$. This implies that $E = e_j \mu_1'$ must hold over the integers as we could factor n otherwise. Therefore e_j is indeed a factor of E as claimed. □

It is not hard to see that one can extend this proof to show that several attributes are encoded, e.g., that e_i, e_j, and e_l are contained in E all at once:

$$\begin{aligned}&PK\{(\varepsilon, \nu', \mu_0, \mu_1') : \\ &\quad Z \equiv \pm R_0^{\mu_0} (R_1^{e_i e_j e_l})^{\mu_1'} A'^{\varepsilon} S^{\nu'} \pmod{n} \ \wedge \\ &\quad \mu_0 \in \pm\{0,1\}^{\ell_m} \ \wedge \ \mu_1' \in \pm\{0,1\}^{\ell_m - 3\ell_t} \ \wedge \\ &\quad \varepsilon \in [2^{\ell_e - 1} + 1, 2^{\ell_e} - 1]\}\end{aligned}$$

In other words, we have just shown how to very efficiently implement an AND statement over the attributes.

Showing that an attribute is not contained in E, i.e., how to prove a NOT-relation. Now, proving that a given e_j is not contained in her credential amounts to show that $e_j \nmid E$ is the case] The user can do so by showing that there exist two integers a and b such that $aE + be_j = 1$. Note that a and b do not exist if $e_j \mid E$. Also note that a and b can be computed efficiently with the extended Euclidian algorithm.

The protocol that achieves this is as follows:

After having computed a and b, the user chooses a sufficiently large random r (about 80 bits larger n) and computes a commitment $D = g^E h^r \bmod n$. She sends D to the verifier and runs the following protocol with him (where a and b are the secret denoted by α and

β, respectively). Finally, the user engages with the verifier in the proof:

$$\begin{aligned}&PK\{(\varepsilon, \nu', \mu_0, \mu_1, \alpha, \beta, \rho, \rho') :\\&\quad Z \equiv \pm R_0^{\mu_0} R_1^{\mu_1} A'^{\varepsilon} S^{\nu'} \pmod{n} \ \wedge\\&\quad D \equiv \pm g^{\mu_1} h^{\rho} \bmod n \ \wedge \ g \equiv \pm D^{\alpha} (g^{e_j})^{\beta} h^{\rho'} \bmod n \ \wedge\\&\quad \mu_0, \mu_1 \in \pm\{0,1\}^{\ell_m} \ \wedge \ \varepsilon \in [2^{\ell_e - 1} + 1, 2^{\ell_e} - 1]\}\end{aligned}$$

THEOREM 4.2. *If a prover is successful in the above protocol, then she was issued a credential that does not contain the attribute encoded by e_j.*

PROOF. As before, it is standard to show that there exists a knowledge extractor who can extract from a convincing prover values ε, ν', μ_0, μ_1 ρ, α, β, ρ' such that the equations given in the proof specification hold. Let us see what one can derive from these equations. First consider $D \equiv \pm g^{\mu_1} h^{\rho} \bmod n$ and $g \equiv \pm D^{\alpha} (g^{e_j})^{\beta} h^{\rho'} \bmod n$ from which we can derive that $g \equiv \pm g^{\mu_1 \alpha} h^{\rho\alpha} (g^{e_j})^{\beta} h^{\rho'} \bmod n$ holds. Unless the prover can compute $\log_g h$ or a multiple of the order of g it must hold that $1 = \mu_1 \alpha + e_j \beta$. Now, it is well known that α and β can only exist if e_j and μ_1 are co-prime and thus $e_j \mid \mu_1$. Because of the first equation of the proof specification, we know that μ_1 is contained in the CL signature and thus we can conclude that the e_j is not one of the attributes encoded in μ_1, i.e., in the credential the user proves knowledge of. □

Obviously the protocol can be extended to show several attribute values are not contained in a credential in just one proof by replacing e_j by the product of the respective primes.

Showing that one of of a list if attributes is contained in a credential (OR-relation). Let us now show how we can implement a proof of a statement such as *I'm either a student, a retiree, or unemployed* as might be the case if one would be eligible for a reduce entrance fee to a museum. More generally, we assume that we are given a list of encodings $\{e_1, ..., e_\ell\}$ of attribute values (possibly ranging over different attributes), for some ℓ. The idea we use here is that if a credential contains an attribute e that is contained in this list, then there exists an integer a such that $ae = \prod_i^{\ell} e_i$; if e is not in the list, then no such integer a as e does not divide the product. Let us first assume that the issuer imposes that only one attribute gets encoded into a signed message. We will later see how we can extend this to several attributes.

To prove that her credential contains one of the attributes values $\{e_1, ..., e_\ell\}$, a user can employ the following protocol. First, the user computes a commitment D to the attribute contained in her credential (in the same way as for the other protocols), sends it to the verifier, and then runs with the verifier the following proof protocol:

$$\begin{aligned}&PK\{(\varepsilon, \nu', \mu_0, \mu_1, \alpha, \rho, \rho') :\\&\quad Z \equiv \pm R_0^{\mu_0} R_1^{\mu_1} A'^{\varepsilon} S^{\nu'} \pmod{n} \ \wedge\\&\quad D \equiv \pm g^{\mu_1} h^{\rho} \bmod n \ \wedge \ g^{\prod_i^{\ell} e_i} \equiv \pm D^{\alpha} h^{\rho'} \bmod n \ \wedge\\&\quad \mu_0, \mu_1 \in \pm\{0,1\}^{\ell_m} \ \wedge \ \varepsilon \in [2^{\ell_e - 1} + 1, 2^{\ell_e} - 1]\}\end{aligned}$$

We leave the proof for this protocol to the reader and extend it to work also in case more than one attribute is encoded into a signed message. So now, the goal is to show that one of the attribute values encoded in the credential is contained in the list $\{e_1, ..., e_\ell\}$. The idea here is that the user commits to that attribute value and then shows that it divides the protocol of the attribute values on the list as well as the message encoded in the credential. However, we must take some special care as this statement also holds for ± 1 and so we must make sure that the commitment does not contain ± 1. To this end we need to employ of further group, i.e., one of prime order $\mathfrak{q}$ and two generators $\mathfrak{g}$ and $\mathfrak{h}$ of that group such that $\log_{\mathfrak{h}} \mathfrak{g}$ is unknown. Now, except the commitment D to the attribute value in question, say e_j, as before, the user further computes the commitment $\mathfrak{D} = \mathfrak{g}^{e_j}\mathfrak{h}^r$, where r is a random element from $\mathbb{Z}_{\mathfrak{q}}$. Finally, the following proof protocol will achieve our goal:

$$\begin{aligned}
&PK\{(\varepsilon, \nu', \mu_0, \mu_1, \alpha, \beta, \delta, \rho, \rho', \varphi, \gamma, \psi, \xi, \sigma) :\\
&\quad Z \equiv \pm R_0^{\mu_0} R_1^{\mu_1} A'^{\varepsilon} S^{\nu'} \pmod{n} \;\wedge\; D \equiv \pm g^{\alpha} h^{\rho} \bmod n \;\wedge\\
&\quad g^{\prod_i^{\ell} e_i} \equiv \pm D^{\delta} h^{\rho'} \bmod n \;\wedge\; 1 \equiv \pm D^{\beta} g^{\mu_1} h^{\rho'} \bmod n \;\wedge\\
&\quad \mathfrak{D} = \mathfrak{g}^{\alpha}\mathfrak{h}^{\varphi} \;\wedge\; \mathfrak{g} = (\frac{\mathfrak{D}}{\mathfrak{g}})^{\gamma}\mathfrak{h}^{\psi} \;\wedge\; \mathfrak{g} = (\mathfrak{g}\mathfrak{D})^{\sigma}\mathfrak{h}^{\xi} \;\wedge\\
&\quad \mu_0, \mu_1 \in \pm\{0,1\}^{\ell_m} \;\wedge\; \varepsilon \in [2^{\ell_e - 1} + 1, 2^{\ell_e} - 1]\}
\end{aligned}$$

THEOREM 4.3. *A user who can successfully run the protocol above must have been issued a credential that encodes at least one of the attribute values* $\{e_1, ..., e_\ell\}$.

PROOF. Again, one can extract from a successful prover values $(\varepsilon, \nu', \mu_0, \mu_1, \rho, \alpha, \beta, \rho, \rho', \varphi, \gamma, \psi, \xi, \sigma)$ such that all the equations given in the proof protocol specification hold. Let us consider what we can derive from these equations. First, consider the equations $\mathfrak{D} = \mathfrak{g}^{\alpha}\mathfrak{h}^{\varphi}$ and $\mathfrak{g} = (\frac{\mathfrak{D}}{\mathfrak{g}})^{\gamma}\mathfrak{h}^{\psi}$. Assuming the hardness of computing $\log_{\mathfrak{g}} \mathfrak{h}$, we have $1 \equiv \gamma(\alpha - 1) \pmod{\mathfrak{q}}$ from which we can derive that $\alpha \not\equiv 1 \pmod{\mathfrak{q}}$. A similar argument can be made with $\mathfrak{g} = (\mathfrak{g}\mathfrak{D})^{\sigma}\mathfrak{h}^{\xi}$ regarding the statement $\alpha \not\equiv -1 \pmod{\mathfrak{q}}$ and hence $\alpha \neq \pm 1$ will also hold over the integers. Now consider $D \equiv \pm g^{\alpha} h^{\rho} \bmod n$ and $g^{\prod_i^{\ell} e_i} = D^{\delta} h^{\rho'} \bmod n$. Assuming the hardness of factoring, we can conclude from these that $\alpha \mid \prod_i^{\ell} e_i$ and thus that α equals one of the e_i's or a product of them (as we know that $\alpha \neq \pm 1$). Now, from the equation $1 = D^{\beta} g^{\mu_1} h^{\rho'} \bmod n$ we can derive that $\beta\alpha = \mu_1$ holds over the integers provided factoring is hard. As we thus have $Z \equiv \pm R_0^{\mu_0} R_1^{\beta\alpha} A'^{\varepsilon} S^{\nu'} \pmod{n}$ it follows that α is encoded in the credential and there for that at least one of the attribute value encodings $\{e_1, ..., e_\ell\}$ is contained in the credential issued to the prover. □

4.5 Prime Encodings For Other Schemes

We have presented how one can encode attributes efficiently for the RSA-based CL credential system. Our method is intrinsically based on integer factorization and to be able to proves multiplicative relations among committed values over the *integers*. The only known efficient commitment scheme that works for the latter is the one by Damgård, Fujisaki, Okamoto, which relies on the Strong RSA assumption. Thus, our method works naturally with anonymity-providing schemes that themselves are based on the Strong RSA assumption (e.g., [Ateniese et al. 2000; Kiayias et al. 2004]) and indeed we made use of the fact that these scheme basically have the Damgård-Fujisaki-Okamoto built into them.

Nevertheless, our method can also be applied to other anonymity-providing schemes but requires one to add the Strong RSA assumption to the list of assumption the scheme is based upon, as we rely on the integer commitment scheme for proving the relations. Let us have sketch how this would be done. First, the scheme needs to be capable to encode attributes as exponents (most efficient scheme allow for that, e.g., [Boneh et al. 2004; Brands 1995a; Camenisch and Lysyanskaya 2004]). However, such encoding is usually done in the exponent of a group where the prover knows the order, say q, and therefore all relations that can be proven about values contained in the credential hold modulo q only. Therefore, the prover needs to provide an integer commitment to the attribute values contained in the credential (i.e., the product of the prime encoded attribute values), so that we can do our proofs for the prime encodings over the integers. Of course, the prover needs to prove that the very same value contained in the commitment is also encoded in the credential. Here one must apply some care to avoid that the prover cannot add a multiple of q to the committed value, i.e., the prover needs to prove that the commitment contains a value between 0 and q which we can do using the range proof by Boudot [2000] at the cost of about 5 extra commitments and 5 proof-terms.

5. EFFICIENCY

Our improved credential system encodes a large set of binary and finite set attributes without significant performance impacts. The computational complexity of a traditional CL proof of possession is linear in the total number of attributes, whereas our system's complexity only depends on the number of string/integer attributes. Binary and finite-set attributes are essentially for free. Even though the number of binary/finite-set attributes has a influence on the performance, we can consider it as constant for all practical purposes[2]. Both schemes have identical complexity if credentials only contain string or integer attributes, as soon as binary or finite-set attributes are involved the prime encoding scheme achieves superior efficiency.

5.1 Measurement Method

Our key goal is to improve the efficiency of the CL signature scheme on small devices, particularly on smartcards. We encountered the following three properties during the evaluation of smartcard capabilities: (1) Most smartcards do not provide a hardware primitive for multi-base exponentiation. One either needs to resort to a software implementation or the hardware's modular exponentiation. (2) Partially, the cards do not provide sufficient access to the square and multiply primitives, which hinders an efficient software implementation of multi-base exponentiations. (3) Partially, the cards have severe transient memory restrictions, which hamper a multi-base exponentiation in one go. A software implementation can therefore experience a negative performance impact. We conclude, that we cannot restrict our efficiency measurement to multi-base exponentiations, but need to examine modular exponentiations.

How to compute the number of exponentiations from a Camenisch-Stadler [1997] term of a CL Signature(A, e, v)? First, the prover computes a blinded CL signature (A', e, v'), which amounts to one exponentiation:

$$A' := AS^{v*} \text{ for } v* \in_R \{0,1\}^{\ell_v}$$

[2] Our system uses prime exponents with a very short bit-length and treats them in a single exponentiation. This single exponentiation becomes costly for a large prime product $E = \prod_{1 \leq i \leq \ell} e_j$, as it requires $O(\log_2 E)$ multiplications. If one restricts the bitlength of E to the standard message length parameter of Identity Mixer ($\ell_m = 256$) one can handle 37 binary attribute realizations with the same complexity as one single exponentiation for a string/integer attribute. See Appendix Section B.

Second, the prover facilitates the following proof of knowledge:

$$\begin{aligned}&PK\{(\varepsilon, \nu', \mu_0, \mu_1, \mu_2) :\\ &\quad Z \equiv \pm R_0^{\mu_0} R_1^{\mu_1} R_2^{\mu_2} A'^{\varepsilon} S^{\nu'} \pmod{n}\\ &\quad \mu_0 \in \pm\{0,1\}^{\ell_m} \;\wedge\; \mu_i \in \pm\{0,1\}^{\ell_m - 3\ell_t} \;\wedge\\ &\quad \varepsilon \in [2^{\ell_e - 1} + 1, 2^{\ell_e} - 1]\}\end{aligned}$$

Camenisch and Stadler [1997] define how to transform such a statement in a Schnorr proof. The prover choses random values r_ε, $r_{\nu'}$, r_{μ_0}, r_{μ_1}, r_{μ_2} from $\{0,1\}^{\ell_\cdot}$ for each proven value. $\ell_\cdot$ represents length parameter of the system specification corresponding to this randomness. The prover computes a commitment

$$T = A'^{r_\varepsilon} R_0^{r_{\mu_0}} R_1^{r_{\mu_1}} R_2^{r_{\mu_2}} S^{r_{\nu'}} \pmod{n}.$$

This involves either one multi-base exponentiation or modular exponentiations in the number of attributes plus two. For a given challenge c, the prover computes response values s_ε, $s_{\nu'}$, s_{μ_0}, s_{μ_1}, and s_{μ_2} as follows:

$$s_\varepsilon = r_\varepsilon + c\varepsilon; \quad s_{\nu'} = r_{\nu'} + c\nu'; \quad s_{\mu_1} = r_{\mu_1} + c\mu_1; \quad s_{\mu_2} = r_{\mu_2} + c\mu_2$$

Thus, in total we have number of attributes plus three exponentiations. We observe that the number of exponents in the Camenisch-Stadler notation determines the number of exponentiations in the corresponding zero-knowledge proofs.

How to treat partially disclosed attributes? In our prime encoding scheme, we create proofs over known prime exponents, say e_i and e_j, proving knowledge of the remainder μ_1' as follows:

$$\begin{aligned}&PK\{(\varepsilon, \nu', \mu_0, \mu_1') :\\ &\quad Z \equiv \pm R_0^{\mu_0} (R_1^{e_i e_j})^{\mu_1'} A'^{\varepsilon} S^{\nu'} \pmod{n} \;\wedge\; [\ldots]\}\end{aligned}$$

We will count a term $(R_1^{e_i e_j})^{\mu_1'}$ as one exponentiation because of the following rationale: The prover can choose a random value $r_{\mu_1'}$ from $\{0,1\}^{\ell_m}$ and then compute the product $r_{\mu_1} = e_i e_j r_{\mu_1'}$ as randomness. The prover includes this randomness with only one exponentiation into the commitment T.

5.2 Qualitative Analysis

Let us first consider the differences of traditional encoding in credential systems and the prime encoding. We do so by comparing different proof statements for a credential with only two finite-set attributes. We focus on the computational workload of the prover, as small devices usually act in this role. In principle, all proofs with the Camenisch-Lysyanskaya credential system are structured as follows: the user provides a proof of possession of the credential first, potentially disclosing certain attribute values. The prover may commit to some attributes to facilitate further proofs (e.g., range). The proof of possession requires one term and exponentiations linear of the number of attribute bases[3]:

$$\begin{aligned}
&PK\{(\varepsilon, \nu', \mu_0, \mu_1, \mu_2) : \\
&\quad Z \equiv \pm R_0^{\mu_0} R_1^{\mu_1} R_2^{\mu_2} A'^{\varepsilon} S^{\nu'} \pmod{n} \wedge \\
&\quad \mu_0 \in \pm\{0,1\}^{\ell_m} \;\wedge\; \mu_i \in \pm\{0,1\}^{\ell_m - 3\ell_t} \;\wedge \\
&\quad \varepsilon \in [2^{\ell_e - 1} + 1, 2^{\ell_e} - 1]\}
\end{aligned}$$

This sets the baseline of complexity for all subsequent proofs with the credential system. Subsequently, we omit the ranges of the attribute messages and exponents for readability (denoted by [...]).

5.2.1 *AND-Proof.* Let us consider an example where a user wants to prove her expertise according to the ACM Computing Classification Scheme [Association for Computing Machinery (ACM) 2007]: E.Data ∧ D.SW. We refer to Appendix Section C for such an application example. An AND-proof with the traditional encoding in the CL-signature uses one exponentiation for each attribute base. The prover facilitates the following proof of knowledge with a selective disclosure of the attribute values E.Data and D.SW:

$$\begin{aligned}
&PK\{(\varepsilon, \nu', \mu_0) : \\
&\quad Z \equiv \pm R_0^{\mu_0} R_1^{\mathsf{E.Data}} R_2^{\mathsf{D.SW}} A'^{\varepsilon} S^{\nu'} \pmod{n} \;\wedge\; [\ldots]\}
\end{aligned}$$

Proving knowledge of several prime-encoded attribute does not produce any overhead. The required attributes are encoded in one attribute base and their conjunctive selective disclosure can be implemented with one modular exponentiation. This realizes a AND-proof in a constant number of exponentiations.

$$\begin{aligned}
&PK\{(\varepsilon, \nu', \mu_0, \mu_1') : \\
&\quad Z \equiv \pm R_0^{\mu_0} (R_1^{e_i e_j})^{\mu_1'} A'^{\varepsilon} S^{\nu'} \pmod{n} \;\wedge\; [\ldots]\}
\end{aligned}$$

5.2.2 *NOT-Proofs.* The NOT-proof methods of the traditional approach and the prime-encoding are very similar. Both methods require a commitment to the relevant attribute and a linear relationship proof. For the traditional approach the proof is constructed as follows:

$$\begin{aligned}
&PK\{(\varepsilon, \nu', \mu_0, \mu_1, \mu_2, \alpha, \beta, \rho) : \\
&\quad Z \equiv \pm R_0^{\mu_0} R_1^{\mu_1} R_2^{\mu_2} A'^{\varepsilon} S^{\nu'} \pmod{n} \;\wedge \\
&\quad D \equiv \pm g^{\mu_1} h^{\rho} \bmod n \;\wedge\; g \equiv \pm (D/g^{\rho})^{\alpha} h^{\beta} \;\wedge\; [\ldots]\}
\end{aligned}$$

The NOT-proof of the new system needs to take the structure of the dedicated prime attribute into account, however, does not differ conceptually from the traditional approach. Given that we count $(g^{e_j})^{\beta}$ as one exponentiation, both methods have the same complexity:

$$\begin{aligned}
&PK\{(\varepsilon, \nu', \mu_0, \mu_1, \alpha, \beta, \rho, \rho') : \\
&\quad Z \equiv \pm R_0^{\mu_0} R_1^{\mu_1} A'^{\varepsilon} S^{\nu'} \pmod{n} \;\wedge \\
&\quad D \equiv \pm g^{\mu_1} h^{\rho} \bmod n \;\wedge \\
&\quad g \equiv \pm D^{\alpha} (g^{e_j})^{\beta} h^{\rho'} \bmod n \;\wedge\; [\ldots]\}
\end{aligned}$$

[3] To be precise: $L + 2$ exponentiations for the attributes including the secret key plus one exponentiation for the blinding

5.2.3 *OR-Proofs.* We use an example where a user proves that either the attribute $\mathit{social\_benefit} = \mathsf{social\_benefit}$ or the attribute $\mathit{profession} = \mathsf{student}$. We elaborate on such a case in the environment of electronic identity cards in Section 6.2. The traditional approach needs to produce an overhead proportional to the number of relevant attributes as well as to the number of comparison alternatives. The system first commits to the relevant attribute values, which means computing the commitment as well as proving knowledge of it:

$$\begin{aligned}
&PK\{(\varepsilon, \nu', \mu_0, \mu_1, \mu_2, \rho, \rho') : \\
&\quad Z \equiv \pm R_0^{\mu_0} R_1^{\mu_1} R_2^{\mu_2} A'^{\varepsilon} S^{\nu'} \pmod{n} \ \wedge \\
&\quad D_1 \equiv \pm g^{\mu_1} h^{\rho} \pmod{n} \wedge \\
&\quad D_2 \equiv \pm g^{\mu_2} h^{\rho'} \pmod{n} \ \wedge [\ldots]\}
\end{aligned}$$

It then facilitates proofs of knowledge over the committed attribute values, in this case a disjunction of equality proofs. If the user intends to prove that her attribute is one out of ten variants, she needs to provide ten equality proofs similar to those:

$$\begin{aligned}
&PK\{(\rho, \rho') : \\
&\quad D_1 / g^{\mathsf{social\_benefit}} \equiv \pm h^{\rho} \pmod{n} \vee \\
&\quad D_2 / g^{\mathsf{student}} \equiv \pm h^{\rho'} \pmod{n}\}
\end{aligned}$$

Our new method also facilitates the OR-proof in a constant number of exponentiations. It involves (1) committing to the dedicated prime attribute $R_1^{\mu_1}$ in the first line of the proof statement, (2) showing that the user's attribute value is contained in the list of options and that it divides the credential message (second line), and (3) proving that the commitment is free from ± 1 (two last lines):

$$\begin{aligned}
&PK\{(\varepsilon, \nu', \mu_0, \mu_1, \alpha, \beta, \delta, \rho, \rho', \varphi, \gamma, \psi, \xi, \sigma) : \\
&\quad Z \equiv \pm R_0^{\mu_0} R_1^{\mu_1} A'^{\varepsilon} S^{\nu'} \pmod{n} \wedge \\
&\quad D \equiv \pm g^{\alpha} h^{\rho} \bmod n \wedge \\
&\quad g^{\prod_i^{\ell} e_i} \equiv \pm D^{\delta} h^{\rho'} \bmod n \ \wedge \ 1 \equiv \pm D^{\beta} g^{\mu_1} h^{\rho'} \bmod n \ \wedge \\
&\quad \mathfrak{D} = \mathfrak{g}^{\alpha} \mathfrak{h}^{\varphi} \wedge \\
&\quad \mathfrak{g} = (\frac{\mathfrak{D}}{\mathfrak{g}})^{\gamma} \mathfrak{h}^{\psi} \ \wedge \ \mathfrak{g} = (\mathfrak{g}\mathfrak{D})^{\sigma} \mathfrak{h}^{\xi} \ \wedge [\ldots]\}
\end{aligned}$$

This construction requires two commitments (computing the commitments and proving their knowledge) and four linear relationship proofs in total. The number of terms and exponentiations is essentially independent from the number of OR-Terms. We note that prime exponents are publicly known and very small, thus, $g^{\prod_i^{\ell} e_i}$ counts as one exponentiation for all practical purposes. Thus, we account for a constant overhead of 23 exponentiations over a normal proof of possession.

5.3 Quantitative Analysis

We compare the computational complexity between the Camenisch-Lysyanskaya system, a bit-vector encoding, and our method in Table I. We count the number of multi-base exponentiations and the number of modular exponentiations.

Table I. Computational Complexity. Gray color denotes best result.
Notation: M. are multi-base exponentiations, E. modular exponentiations. L is the total number of attribute bases without secret key. l is the number of string/integer attributes, whereas k: number of prime-encodable attributes/value set for multi-variate finite-set attributes. i is the number of attributes referenced in a proof.

Parameter	**Base Encoding**		**Bit-Vector Encoding**		**Prime Encoding**	
	absolute	*asymp.*	*absolute*	*asymp.*	*absolute*	*asymp.*
Number of bases	$l+k$	$O(L)$	$l+1$	$O(l)$	$l+1$	$O(l)$
Proof of possession	$l+k+4$ E.	$O(L)$	$l+5$ E.	$O(l)$	$l+5$ E.	$O(l)$
Knowledge of 1 binary attr.	1 M. $l+k+4$ E.	$O(1)$ $O(L)$	$1+2k+2$ M. $l+4k+7$ E.	$O(k)$ $O(L)$	1 M. $l+5$ E.	$O(1)$ $O(l)$
AND of i binary attr.	1 M. $l+k+4$ E.	$O(1)$ $O(L)$	$2k+i+2$ M. $l+4k+i+7$ E.	$O(2i)$ $O(L+i)$	1 M. $l+5$ E.	$O(1)$ $O(l)$
NOT of 1 binary attr.	4 M. $l+k+11$ E.	$O(1)$ $O(L)$	$2k+4$ M. $l+4k+10$ E.	$O(k)$ $O(L)$	4 M. $l+13$ E.	$O(1)$ $O(l)$
OR of i binary attr.	$3i+1$ M. $l+k+6i+4$ E.	$O(i)$ $O(L+i)$	$2k+3i+1$ M. $l+3k+i+7$ E.	$O(k+i)$ $O(L+i)$	9 M. $l+23$ E.	$O(1)$ $O(l)$

The comparison is based on the following parameters:

—L: total number of attribute bases without secret key,

—l: number of string/integer attributes,

—k: number of prime-encodable attributes/value set for multi-variate finite-set attributes,

—i: number of attributes referenced in a proof.

We notice that in general the proofs of possessions of CL credentials are impacted by the total number of attribute bases L, whereas the bit-vector and prime encoding only depend on the number of string/integer attributes l. For simple attribute proofs, CL and bit-vector encoding require $O(L)$ exponentiations whereas our system only depends on the number of string/integer attributes $O(l)$. If one considers a credential with only binary or finite-set attribute, CL and bit-vector encoding have a complexity of $O(L)$, whereas our system runs in constant time $O(1)$. The AND proofs are impacted by the total number of attributes and require $O(L)$ exponentiations. Once the proof of possession is complemented by an OR-statement, CL encoding requires $O(i)$ terms and $O(L+i)$ exponentiations[4]. A traditional bit-vector encoding as discussed in Section 4.1 involves bit-commitments to all encoded attributes (two exp. for computing, two for proving), bitwise OR-proofs for all attributes (two exp.), and one equality proof over their product (two exp.). This amounts to $O(k+i)$ terms and $O(L+i)$ exponentiations. Our system allows for proofs with a constant term number. The total number of prime-encodable attributes k does not impact the performance at all. This comes at a cost of a constant overhead of 18 exponentiations. We discuss the structures of the AND/OR proof statements in Sections 5.2.1 and 5.2.3.

To stress our point, we make an experiment with the number of prime-encodable attributes being large against the number of string/integer attributes: $k \gg l$. Say we only encode a huge number of binary or finite-set attributes ($L = k \gg i$). In this case the results are as follows: then proof statements with CL and bit-vector encoding will converge to $O(L)$ exponentiations. Our system, however, converges to a constant number of terms and exponentiations $O(1)$. Note that the bitlength of the prime product exponent E impacts

[4] Per i we have two exp. to compute the commitment, two to prove its knowledge, and two to prove equality.

the complexity of this single exponentiation, as it requires $O(\log_2 E)$ multiplications. For all practical applications it is negligible[5].

6. APPLICATIONS

6.1 Requirements

An application of our extension to the CL credential system needs to fulfill two requirements: (1) a sufficient supply of prime-encodings and (2) a certified binding between prime-encoding and discrete values. First, we observe that the number of primes below a certain number x is estimated by the *prime number theorem* as outlined in [Weisstein 2008] and converges to $\pi(x) = x/\mathsf{ln}(x)$. There exist, for instance, roughly 75.638 prime numbers smaller than 20 bits. This is a plentiful supply for most application scenarios. Second, the issuer needs to sign the binding between primes and discrete values in its public key. Thus, the binary/discrete values used by the credential system are static. This excludes highly dynamic applications with ad-hoc issuing of credentials with new attribute types, however, does not impact any of the standard application scenarios for credential systems. Typical organizations issuing credentials are governments, banks, mobile operators, etc. Their vocabulary for binary/finite-set attributes is standardized well in advance.

We observe that space constraints impact the number of finite set attributes the system can govern. The size of the attribute exponent in the user's credential limits the number of prime flags set in a credential. We elaborate on this fact in Appendix Section B. The size of the issuer's public key limits the total number of attribute realizations certified for the system. Thus, even if many sets in real world are inherently finite, a system needs to balance between efficiency gain and and space consumption.

The proposed credential system is particularly suited for multiple classes of attributes:

Binary: The attribute can either be present or not, true or false, e.g., being a civil servant.

Finite Set: A finite set of discrete attribute values, where a user may realize at most one potential value. E.g., hair color.

Multi-Variate Finite Set: A finite set of discrete attribute values, where a user may realize any subset of values. A user may hold multiple values for an attribute such as profession.

Finite Data Structures: Complex data structures of discrete values from a finite set, where trees are most useful: a user may realize a sub-tree or path of a super-tree predetermined by the issuer. Examples for such attributes are expertise or health taxonomies as well as role hierarchies.

These attribute types impact a large variance of application scenarios. We choose electronic identity cards as primary example and complement that with complex expertise as well as medical credentials in the Appendix C.

[5] We show in Appendix Section B, that one can encode 37 attribute realizations in a standard-size message exponent of Camenisch-Lysyanskaya.

Table II. Potential attributes on electronic identity cards.

String/Integer	Binary/Finite Set	Example Values
1) name	6) sex	{male, female}
2) first name	7) nationality	193 recognized states
3) date of birth	8) place of birth	6400 villages and cities (e.g., Germany)
4) identification number	9) type of card	{EID, kids_card}
5) date of issuance	10) place of issuance	429 districts (e.g., Germany)
	11) validity time	{2_year, 5_year, 10_year}
	12-13) eye and hair color	6 hair colors, 8 eye colors
	14-16) minority status	{blind, vis_impaired, deaf, hear_impaired, phys_impaired}
	17-18) social benefit status	{none, unemployed, social_benefit}
	19-21) profession	{student, teacher, civil_servant, doctor, ... }
	22-23) academic degree	{B.S., M.S., Ph.D., M.D., ... }

6.2 Examples for Electronic Identity Cards

Currently, different European countries are issuing different variants for electronic identity cards (EID). The computational restrictions of such smartcards are obvious. The desire for protection of citizen rights by privacy-enabling technologies is also a recurring topic. Particularly, in the area of secondary use—that is, when a third party is accessing the user's data—privacy concerns surface quickly. In early proposals, arbitrary third parties could access the full data set about the user.

We surveyed different data sets for EID and driver's license cards and use the Belgium EID card as example [SPF Intérieur 2005]. Table II outlines a superset of example attributes, where the left column contains string and integer attributes, whereas the right column contains attributes encodable by our prime representation. We explicitly mention minority status as, for instance, the Belgium EID card specification [SPF Intérieur 2005, pp.12] explicitly covers this option[6].

We tailored our scheme to attributes that have a finite set of values. The user may realize a multi-valued subset. Minority status, profession, or academic degree are such attributes. A citizen may, for instance, be a doctor as well as a civil servant. Traditional CL signatures encodes each attribute in a separate attribute base, for multi-variate attributes from finite set it even needs to encode *each potential realization* in a base. For the attributes in Table II, this results in 23 attribute bases.[7]

With our prime encoding we can fold all binary and finite-set attributes into one attribute base. We choose a prime e_i for all binary attributes and finite set attribute realizations in Table II: this involves 193 possible realizations of nationality, 429 realizations of place of issuance districts, 6400 districts for place of birth as well as several hundred professions, and 14 color variations for hair and eyes. We dedicate the first attribute base, for the product of the corresponding prime numbers e_j that the user realizes. We are left with five normal attribute bases and one attribute base R_1 for the prime encoding. Thus, the number

[6] Application of such an attribute varies much from country to country. For instance, Belgium encodes a status for blind and for the visually impaired citizens. The German driver's license also encodes the requirement to wear glasses. Further attributes for deaf or hearing-impaired citizens are thinkable. Though countries also envision attributes such as profession (e.g., doctor) or role as civil servant, their storage on the EID card itself is currently subject to much dispute.

[7] Nationality and place of issuance will be encoded by a index number. We assume the minority and social benefit status as multi-variate attributes with seven realizations spread over five attributes. For the potentially multi-variate attributes profession and academic degree we reserve five attributes in total. The number of attribute bases is therefore 23 and $L = 24$.

Table III. Complexity in EID Scenario: $\mathcal{P}_{\mathsf{EID,CS}}$ – CL requires nine commitments and eleven equality proofs; prime encoding contents itself two commitments and four relationship proofs. $\mathcal{P}_{\mathsf{EID,OP}}$ – CL requires and AND proof over all L attributes, whereas the prime encoding is only impacted by the l string/integer attributes.

Parameter	**CL**	**Prime**
Number of attribute bases	23	6
E. proof of possession	27	10
M. in AND-proof for $\mathcal{P}_{\mathsf{EID,OP}}$	1	1
E. in AND-proof for $\mathcal{P}_{\mathsf{EID,OP}}$	27	10
M. in OR-proof for $\mathcal{P}_{\mathsf{EID,CS}}$	28	9
E. in OR-proof for $\mathcal{P}_{\mathsf{EID,CS}}$	67	28

of bases is already one fourth and *all* proofs of possessions speed up by factor four.

Opinion Polls. An often discussed example is online opinion polls. In this scenario, a user proves that she belongs to a certain statistical class while retaining a suitable anonymity set. Opinion polls usually gather demographic data, but may also collect educational and professional parameters. We leave the range proof for the date of birth aside, as it is equally costly for both methods.[8] The remaining proof may be constructed according to a conjunctive selective disclosure as specified in policy $\mathcal{P}_{\mathsf{EID,OP}}$:

$$\begin{gathered} sex = \mathsf{female} \ \wedge \ nationality = \mathsf{French} \ \wedge \\ place\_of\_birth = \mathsf{Paris} \ \wedge \ social\_benefit = \mathsf{none} \ \wedge \\ profession_1 = \mathsf{doctor} \ \wedge \ profession_2 = \mathsf{civil\_servant} \ \wedge \\ ac\_degree_1 = \mathsf{M.D.} \ \wedge \ ac\_degree_2 = \mathsf{Ph.D.} \end{gathered}$$

We discussed this general proof structure in Section 5.2.1. A traditional approach requires a proof of possession over L attribute bases[9]. We outline in Table III that our new system facilitates the proof without any overhead to the proof of possession. It is therefore only impacted by the l string/integer attributes. It is three times as efficient, even though we only safe the k attribute bases for the prime-encoded attributes.

Cultural Subsidies. Virtually all countries grant subsidies for access to cultural institutions to particular population groups: children, students, seniors as well as handicapped persons and persons eligible for social benefits. Partially, the corresponding groups show hesitation to disclose their special status because of privacy concerns. Policy $\mathcal{P}_{\mathsf{EID,CS}}$ is depicts a disjunction proof over attributes from Table II:

$$\begin{gathered} \bigvee\nolimits_{minority}(\mathsf{blind}, \mathsf{vis\_impaired}, \mathsf{deaf}, \ldots) \ \vee \\ \bigvee\nolimits_{social\_benefit}(\mathsf{unemployed}, \mathsf{social\_benefit}) \ \vee \\ \bigvee\nolimits_{profession}(\mathsf{student}, \mathsf{teacher}, \mathsf{civil\_servant}) \ \vee \\ (type = \mathsf{kids\_card}) \end{gathered}$$

[8] Of course, there is also a very efficient method for coarse-grained range proofs leveraging the prime encoding. We do not use it for this comparison.

[9] We leverage the system's capability to have multiple values for the attribute $profession$ and academic degree ac_degree.

As demonstrated in Section 5.2.3, this amounts to a proof of possession, attribute commitment to all relevant attributes (nine terms)[10], and a second step disjunction of equality proofs for the possible attribute values (eleven terms). Our new scheme reduces the effort to a single multi-element OR-proof in the prime encoding. As shown in Section 4.4, the user needs to provide a proof of possession with four terms, a commitment to the prime attribute base and proofs of their knowledge and division (three terms) as well as proofs that the commitment does not contain ± 1 in a second group (three terms). We compare both methods in Table III and observe that our new method is three times as efficient for all proofs of possessions as well as the OR-proof for policy $\mathcal{P}_{\mathsf{EID,CS}}$.

6.3 Discussion

Our method is a enabler for credential systems on small devices. Until now, application designers for this area restricted themselves to simple scenarios: credentials must only govern a minimal number of attributes, proof statements must be as simple as possible. The linear complexity in the total number of attributes for the proofs of possession puts credential-enabled EID systems at peril. The vastly growing number of terms and commitments, and thus computational and communication costs, for complex proof statements acted as second bottleneck. These tremendous limitations rendered sensible applications on small devices virtually impossible.

We have shown that our prime-encoding idea makes complex proofs in various application scenarios possible. Be it benefit access with a large anonymity set or collecting demographic data in a private manner in the example of an electronic identity card; be it complex expertise taxonomies of a corporate card; or be it structured diagnostic statements in a healthcare card—our system achieves tremendous performance boosts. This does not only hold for the AND/OR example policies, but also for the overall reduction of attribute bases. The latter parameter impacts every single proof. All these improvements bring applications barely running in feasible time with traditional encodings well in reach of small devices.

In addition to occupying a high ground in the quest for performance, the system comes with two subtle advantages: (1) discrete and structured attributes, (2) significant policy independence. First, we focus on discrete values from finite sets. These may be as simple as binary flags or complex data structures. In contrast to an unstructured integer/string encoding, discrete values can be manipulated by equality and relationship proofs. Their semantic is accessible to the credential system itself. Second, we observe that our method requires a constant number of terms and commitments for pure equality, conjunction and disjunction proofs with binary/finite-set attributes. Independent from the number of AND/OR clauses in the policy, the proof only uses a fixed low number of exponentiations. This makes a transaction and their expected response time predictable to device producers.[11]

[10] Note that the attributes $minority$, $social\_benefit$, and $profession$ are spread over multiple attribute bases to account of multiple realizations by a user. We assumed three for the $minority$, two for $social\_benefit$, and three for $profession$, thus, nine terms in total.

[11] In traditional encodings this is not the case. If a traditional CL system receives a policy from a service provider that requires 25 finite-set attributes and 100 OR-clauses, the system will facilitate 25 commitments and 100 equality proofs executing roughly 150 exponentiations. Our system would finish after 23 exponentiations independently from the policy. We stress that this holds for pure conjunctions (only containing AND-clauses) and disjunctions (only containing OR-clauses). This situation is more diverse for nested logical statements.

7. CONCLUSION

We presented an extension to the Camenisch-Lysyanskaya credential system. It features efficient encoding and proofs of binary and finite-set attributes. The idea to use coprime and divisibility properties in proofs gains us strong performance improvements. We pay the price of certifying prime/attribute value relationships in the issuer's public key and win the ability to facilitate proofs of possession, equality, AND, NOT, and OR proofs very efficiently. Our method overcomes the fundamental limitation of all existing credential systems that their complexity is linear in the total number of attributes. It allows us to fold many finite set attributes in a single attribute base and therefore boosts the performance of all proofs of possession. Our new proof primitives on the prime-encoding facilitate AND, NOT, and OR statements with constant complexity and minimal overhead to a standard proof of possession. Our method does not require additional cryptographic assumptions apart from Strong-RSA.

Our method targets the major attribute classes of credential systems. In fact, we perceive that only a minority of attributes requires a generic string or integer attribute (such as name and birthday), whereas most attributes are either binary or taken from a finite set of discrete values. Those are the attributes which applications need for logical statements. Emerging efforts to standardize vocabulary for identity federation protocols in different application areas support our hypothesis. We demonstrated that our method impacts real applications such as electronic identity cards or complex forms of professional and medical credentials.

APPENDIX

A. PROTOCOL SPECIFICATION

A.1 Parameters and Notation

We shortly name the main parameters of the Camenisch-Lysyanskaya signature scheme and refer to [IBM 2009] for a complete list.

—ℓ_n: Bitlength of the SRSA modulus n, for instance 2048 bits.

—ℓ_e: Bitlength of the e values of certificates, by default 596 bits.

—ℓ'_e: Size of the interval the e values are taken from, by default: 120 bits.

—ℓ_m: Bitlength of attributes, by default 256 bits.

—ℓ_v: Size of the v values of the certificates, by default 2723 bits.

—$\ell_\o$: Security parameter that governs the statistical zero-knowledge property, by default 80 bits.

We distinguish multiple index sets for attribute values:

—A_{KN}: issuer-chosen or known attributes. Common input to both Issuer and Recipient.

—A_{E}: known attributes encoded as prime numbers in the prime product exponent E. Common input to both Issuer and Recipient.

—A_{UC}: user-chosen attributes, only known to the Recipient.

Please note the we omit committed attributes for brevity, even though they compose easily with the prime encoding. The Identity Mixer Specification [IBM 2009] defines their use.

A.2 Issuer Setup

Common input:	Basic system parameters See Identity Mixer Specification [IBM 2009].
Issuer **output:**	$sk_I = (p, q)$
Public **output:**	$pk_I = (n, S, Z, R_1, \ldots, R_l, P, (a_j, e_j)_{1 \le j \le \Lambda}, g, h, \mathfrak{g}, \mathfrak{h})$

Attribute Space. Issuer chooses a number of attribute bases ℓ.

Generation of a Safe RSA Key Pair. Issuer generates the safe primes p and q , $p = 2p' + 1$ and $q = 2q' + 1$, then computes $n = pq$. Consult the Idemix Specification [IBM 2009] for suitable bitlength for n (ℓ_n bits), p and q ($\ell_n/2$ bits).

Generation of CL Parameters. Issuer chooses

$$S \in_R QR_n, \text{ and}$$
$$Z, R_1, \ldots, R_l \in_R \langle S \rangle$$

(where QR_n is the group of quadratic residues mod n and $\langle S \rangle$ is the subgroup generated by S). S must also have order $\#QR_n = p'q'$. The second step is done by choosing $x_Z, x_{R_1}, \ldots, x_{R_l} \in_R [2, p'q' - 1]$ and computing $Z = S^{x_Z}$, $R_i = S^{x_{R_i}}$ for $1 \le i \le l$.

Commitment Groups. Issuer generates bases g and h for integer commitments. It uses its RSA modulus n and chooses g and h as elements of QR_n. Issuer generates a prime-order group with order $\mathfrak{q}$ and two generators $\mathfrak{g}$ and $\mathfrak{h}$ that $\log_{\mathfrak{h}} \mathfrak{g}$ is unknown.

Finite-Set Attributes. Issuer standardizes the set of binary and finite-set attributes: It enumerates all Λ possible attribute values a_j. It determines the the maximally possible attribute realization E_{MAX}. Issuer chooses an odd prime number e_j for each attribute value a_j.

Adjustment of System Parameters. Issuer determines the bitlength of E_{MAX} and adjusts the required length for messages ℓ_m and e-value of the CL-certificates ℓ_e.

Key Certification. Issuer computes a non-interactive zero-knowledge proof of knowledge P, to prove that the public key was generated correctly.[12] Issuer signs the map of attribute value-prime pairs $(a_j, e_j)_{1 \le j \le \Lambda}$ with the Fiat-Shamir heuristic. It also signs the bases for the commitment groups. The proof P will convince a verifier that $Z, R_i \in \langle S \rangle$ for $1 \le i \le l$ and establish a binding between attribute values and prime encoding.

Key Output. Issuer's public key is $pk_I = (n, S, Z, R_1, \ldots, R_l, P, (a_j, e_j)_{1 \le j \le \Lambda}, g, h, \mathfrak{g}, \mathfrak{h})$ and the private key is $sk_I = (p, q)$.

A.3 Issuing

Common input:	pk_I, $\mathcal{S}$ (incl. $\ell_n, \ell_e, \ell_m, \ell_\emptyset$)), $\{m_i\}_{i \in A_{\mathrm{KN}}}$, $\{(a_j, e_j)\}_{j \in A_{\mathrm{E}}}$ where $\mathcal{S}$is the issue certificate specification
Recipient **private input:**	m_1, $\{m_k\}_{k \in A_{\mathrm{UC}}}$, r
Issuer **private input:**	sk_I
Recipient **output:**	$(A, e, v), m_1, \ldots, m_{l'}$, $\{(a_j, e_j)\}_{j \in A_{\mathrm{E}}}$, $l' \le l$ or $\perp$

[12] We leave the detailed specification of this proof to the Identity Mixer Specification [IBM 2009].

Round 0: Nonce. Issuer chooses a random nonce $n_1 \in_R \{0,1\}^{\ell_\emptyset}$ and sends it to the Recipient.

Round 1: User Commitment

1.1. Recipient chooses a random integer $v' \in_R \pm\{0,1\}^{\ell_n+\ell_\emptyset}$.

1.2. Recipient computes

$$U := S^{v'} \cdot X \bmod n$$

where X contains the user-chosen attributes:

$$X := \prod_{k \in A_{\mathrm{UC}}} R_k^{m_k} \bmod n$$

Next, Recipient computes a non-interactive proof P_1 that this was done correctly.

$$\begin{aligned} SPK\{((m_k)_{k \in A_{\mathrm{UC}}}, v') :& \\ & U \equiv \pm S^{v'} \prod_{k \in A_{\mathrm{UC}}} R_k^{m_k} \pmod{n} \wedge \\ & m_k \in \pm\{0,1\}^{\ell_m+\ell_\emptyset+\ell_H+2} \;\; \forall\, k \in A_{\mathrm{UC}}\} \end{aligned} \tag{1}$$

1.4. Recipient → Issuer: U, P_1, $n_2 \in_R \{0,1\}^{\ell_\emptyset}$.

Round 2: Signature Generation

2.0. Issuer verifies P_1.

2.1. Issuer generates a CL signature on the attributes.

2.1.1. Choose a random prime

$$e \in_R \left[2^{\ell_e-1}, 2^{\ell_e-1} + 2^{\ell'_e-1}\right].$$

2.1.2. Choose a random integer $\tilde{v} \in_R \{0,1\}^{\ell_v-1}$, and compute $v'' = 2^{\ell_v-1} + \tilde{v}$.

2.1.3. Look-up finite-set and binary attribute values $\{(a_j, e_j)\}_{j \in A_{\mathrm{E}}}$. Compute

$$E := \prod_{j \in A_{\mathrm{E}}} e_j .$$

2.1.4. Compute

$$Q := \frac{Z}{U(\prod_{i \in A_{\mathrm{KN}}} R_i^{m_i})(R_{\mathrm{E}}^{E=\prod_{j \in A_{\mathrm{E}}} e_j})S^{v''}} \bmod n \quad \text{and} \quad A := Q^{e^{-1} \bmod p'q'} \bmod n\ .$$

2.1.5. Send the pre-signature (A, e, v'') to the Recipient.

Round 3: Signature Completion

3.0. Compute $v := v'' + v'$.

3.1. Recipient verifies (A, e, v), using the CL-sig verification algorithm:

3.1.1. Check that e is prime, and $e \in \left[2^{\ell_e-1}, 2^{\ell_e-1} + 2^{\ell'_e-1}\right]$.

3.1.2. Check that the values $\{e_j\}_{j \in A_{\mathrm{E}}}$ from the Issuer public key are indeed prime.

3.1.3. Compute $\hat{Z} := A^e S^v (\prod_{i \in \mathcal{S}} R_i^{m_i})(R_{\mathrm{E}}^{E=\prod_{j \in A_{\mathrm{E}}} e_j}) \bmod n$.

3.1.4. If $\hat{Z} \not\equiv Z \pmod{n}$, abort IssueCertificateProtocol.

3.2. Recipient verifies P_2.

3.2.1. Compute $\hat{A} := A^{c'+s_e e} S^{v' s_e} \bmod n$.

3.2.2. Compute $\hat{c} = H(\mathtt{context} \| Q \| A \| \hat{A} \| n_2)$.

3.2.3. If $\hat{c} \neq c'$, abort IssueCertificateProtocol.

3.3. (*output*) If steps 3.1 and 3.2 succeed, store the certificate

$$(m_i)_{i \in \mathcal{S}}, (a_j, e_j)_{j \in A_{\mathrm{E}}}, (A, e, v)), l' \leq l.$$

B. BITLENGTH CONSTRAINTS OF THE PRIME ENCODING

Let us consider how many finite-set attributes we can compress in a single attribute base. We start with a theoretical consideration and exemplify these thoughts with a practical example. Let us consider (1) which attribute values are taken into account, (2) which bitlength their product consumes, and (3) what impact that bitlength has.

We first draw a distinction between an attribute and attribute realization. An *attribute* is a named set of one or more values. Consider for instance the attribute $profession$ with a value set of $\{$ none, doctor, civil_servant, teacher, $\ldots\}$. The value set is the set for the whole user population. It is important to observe that any single user may only realize a small subset of these values. We call this subset *attribute realization*. Even though there may exist hundreds of attribute values, a user usually only realizes a number and the issuer can enforce an upper bound on it. Therefore, we will discuss the number of permissible attribute realizations, not the number of attributes itself.

Second, we consider the bitlength consumed by attribute realizations. The issuer determines the maximal bitlength of a possible attribute realization E_{MAX} in the system. This is the bitlength of the corresponding prime product. We note that the issuer can safely choose very small primes for the encoding, in fact, it can start with prime 5. Let us for a moment analyze how the bitlength of such prime products grows. We consider binary flags, i.e., attributes with two associated prime values, where the user can only realize one of them. Figure 1 depicts the growth of the bitlength of the products of n primes. We observe that the bitlength of prime products shows an approximately linear growth for the first few hundred numbers.

Third, the bitlength of attribute realizations has a two-fold impact. First, as the prime product exponent E grows, the computation time for the corresponding exponentiation grows significantly. Second and more importantly, the possible bitlength of an attribute exponent in Camenisch-Lysyanskaya is limited by the bitlength of the SRSA exponent e. This exponent needs to be resized as well to accommodate for attribute realizations.

Let us consider the standard case according to the Identity Mixer specification: the bitlength of e is specified as 596 bits and the bitlength of an attribute limited to 256 bits. This allows for 37 binary attributes to be stored without any change of system parameters. The user's credential can realize one value for all 37 attributes at the same time.

C. PROFESSIONAL TAXONOMIES

Virtually all professional organizations have elaborate taxonomies of expertise and attributes of clients and objects. Most taxonomies are hierarchically organized and benefit from statements on all their granularity levels. The user may realize any sub-tree or path to a terminal leave from the full taxonomy tree.

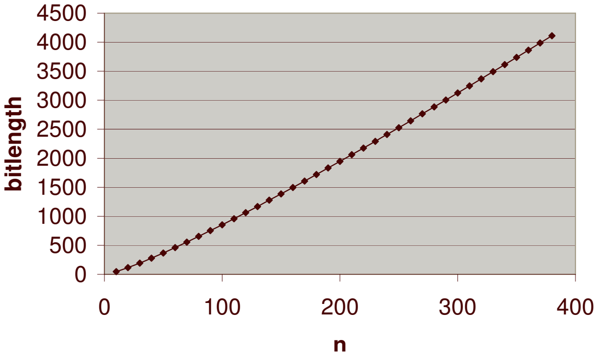

Fig. 1. Bitlength of product of the first n primes. The graph starts with prime 5, we only count every second prime.

Expertise

Let us assume that the ACM decided to issue credentials to their members. These credentials shall contain an expertise classification according to the ACM Computing Classification Scheme [Association for Computing Machinery (ACM) 2007]. This well-known taxonomy is a tree with depth four with eleven areas and roughly 1400 disciplines, subdisciplines, and topics. To encode one path to a terminal leaf of the taxonomy (e.g., "*E. Data – 3. Data encryption – Public key cryptosystems – PKI*"), a traditional credential system would require four attribute bases. That is, four bases per expertise area the user can realize at the same time. We assume that a ACM member may choose three expertise areas. In addition, the ACM allows a choice from sixteen general terms, which are in fact a multi-variate finite set. For this, a traditional credential system reserves additional bases. The total number of required bases is proportional to the depth of the taxonomy times the potential attribute realizations with an offset for the multi-variate finite set. The prime-encoding can represent arbitrarily many attribute realizations in just one attribute base.

Let us assume that a user wants to prove the following policy $\mathcal{P}_{\mathsf{CCS}}$:

$$\begin{gathered} \mathit{expertise} \supset \{\mathsf{E.Data}, \mathsf{3.Encryption}, \mathsf{E.3.PKI}\} \quad \wedge \\ \mathit{expertise} \supset \{\mathsf{D.SW}, \mathsf{4.OS}, \mathsf{4.6.Security}, \mathsf{4.6.Auth}\} \quad \wedge \\ \mathit{general} \supset \{\mathsf{performance}, \mathsf{security}\} \end{gathered}$$

This policy asks for an conjunction proof over all these attributes. We analyzed in Section 5.2.1 which steps different encodings require. Our system encodes the proof as a single multi-element AND-proof integrated in the proof of possession. We compare the complexity in Table IV.

Medical Records

Our new credential system impacts healthcare and medical record credentials tremendously. Healthcare practitioners classify all diseases according to the International Statistical Classification of Diseases and Related Health Problems (ICD) [World Health Organization (WHO) 2005; 2007]. This is a taxonomy tree with depth five: chapters, sub-chapters,

Table IV. Complexity in expertise scenario: CL requires one commitment and equality proof per relevant attribute; the prime encoding's proof of possession covers the AND-proof cost-free.

Parameter	CL	Prime
Number of attribute bases	28	1
E. proof of possession	32	5
M. in AND-proof for $\mathcal{P}_{\mathsf{CCS}}$	1	1
E. in AND-proof for $\mathcal{P}_{\mathsf{CCS}}$	32	5

Table V. Complexity in medical scenario: CL requires one commitment to the disease attribute and 25 equality proofs; the prime encoding manages on two commitments and four relationship proofs.

Parameter	CL	Prime
Number of attribute bases	25	1
E. proof of possession	29	5
M. in OR-proof for $\mathcal{P}_{\mathsf{MED}}$	28	9
E. in OR-proof for $\mathcal{P}_{\mathsf{MED}}$	83	23

section, class, and sub-class.[13] Likewise, the Diagnostic and Statistical Manual of Mental Disorders (DSM-VI-TR) [American Psychiatric Association 2000; Wikipedia 2008] is a taxonomy tree of depth five. Psychiatrists classify mental disorders according to five axes, 16 categories, subcategories, disorder classes and sub-classes.[14] To encode a single path in such a taxonomy in a traditional credential system, one would require five attribute bases per realized terminal leaf. And clearly, there is a need for specifying multiple symptoms or potential diagnoses.

Even though healthcare cards are still in their infancy, there are debates on storing certified medical data on such cards. This option is very privacy-sensitive. Our proposal allows for selective-disclosure of medical information according to standardized taxonomies with variable granularity. Implementing this on smartcards with traditional credential systems is virtually impossible as the growing number of attribute bases would render any proof of possession inefficient. Our system allows to encode many realizations of deep taxonomies within a single attribute base with strong performance improvements.

For instance, let us assume that user holds a certified diagnosis credential that may reserve five possible path in the taxonomy. The policy $\mathcal{P}_{\mathsf{MED}}$ demands to prove that one of the diagnoses matches either one of a set of, say 25, bacterial disease classes that are eligible for acquiring broad-spectrum antibiotics. Clearly, it is highly desirable to hide the actual diagnosis in certain applications. Therefore, the proof must be done with an OR-proof without disclosing the actual disease. This policy is similar to earlier OR-proofs, such as exercised in Section 5.2.3, only restricted to a single relevant attribute. Our credential system does the same proof a third of the term number and a fourth of the required exponentiations (Table V).

[13] A classification could for instance be "*I. Determined infectious and parasitic diseases – Infectious diseases (A) – Infectious abdominal diseases (A00–A09) – A01.- Typhus-alike – A01.2 Paratyphus B*"

[14] For instance sleepwalking would be "*Axis I – 13. Sleep disorders – Parasomnias – Sleepwalking*".

ACKNOWLEDGMENTS

The authors are grateful to the anonymous reviewers of this paper at ACM CCS. They provided valuable and thorough feedback. We thank Saner Celebi for his careful review and comments.

REFERENCES

AMERICAN PSYCHIATRIC ASSOCIATION. 2000. *Diagnostic and Statistical Manual of Mental Disorders (DSM-IV-TR)*, Fourth Edition, Text Revision ed. American Psychiatric Publishing Inc., 1000 Wilson Boulevard, Suite 1825, Arlington, VA 22209. ISBN 9780890420249.

ASSOCIATION FOR COMPUTING MACHINERY (ACM). 2007. ACM computing classification system (CCS). `http://oldwww.acm.org/class/1998/ccs98.html`.

ATENIESE, G., CAMENISCH, J., JOYE, M., AND TSUDIK, G. 2000. A practical and provably secure coalition-resistant group signature scheme. In *Advances in Cryptology — CRYPTO 2000*, M. Bellare, Ed. Lecture Notes in Computer Science, vol. 1880. Springer Verlag, 255–270.

BONEH, D., BOYEN, X., AND SHACHAM, H. 2004. Short group signatures. In *Advances in Cryptology — CRYPTO 2004*, M. K. Franklin, Ed. Lecture Notes in Computer Science, vol. 3152. Springer Verlag, 41–55.

BOUDOT, F. 2000. Efficient proofs that a committed number lies in an interval. In *Advances in Cryptology — EUROCRYPT 2000*, B. Preneel, Ed. Lecture Notes in Computer Science, vol. 1807. Springer Verlag, 431–444.

BRANDS, S. 1993. An efficient off-line electronic cash system based on the representation problem. Tech. Rep. CS-R9323, CWI. Apr.

BRANDS, S. 1995a. Restrictive blinding of secret-key certificates. Tech. Rep. CS-R9509, CWI. Sept.

BRANDS, S. 1995b. Secret-key certificates. Tech. Rep. CS-R9510, CWI. Sept.

BRANDS, S. 1997. Rapid demonstration of linear relations connected by boolean operators. In *Advances in Cryptology — EUROCRYPT '97*, W. Fumy, Ed. Lecture Notes in Computer Science, vol. 1233. Springer Verlag, 318–333.

BRANDS, S. 1999. Rethinking public key infrastructure and digital certificates— building in privacy. Ph.D. thesis, Eindhoven Institute of Technology, Eindhoven, The Netherlands.

CAMENISCH, J., HOHENBERGER, S., KOHLWEISS, M., LYSYANSKAYA, A., AND MEYEROVICH, M. 2006. How to win the clonewars: efficient periodic n-times anonymous authentication. In *ACM Conference on Computer and Communications Security*, A. Juels, R. N. Wright, and S. D. C. di Vimercati, Eds. ACM, 201–210.

CAMENISCH, J., HOHENBERGER, S., AND LYSYANSKAYA, A. 2005. Compact E-cash. In *Advances in Cryptology — Eurocrypt 2005*, R. Cramer, Ed. Lecture Notes in Computer Science, vol. 3494. Springer, 302–321.

CAMENISCH, J. AND LYSYANSKAYA, A. 2001. Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation. In *Advances in Cryptology — EUROCRYPT 2001*, B. Pfitzmann, Ed. Lecture Notes in Computer Science, vol. 2045. Springer Verlag, 93–118.

CAMENISCH, J. AND LYSYANSKAYA, A. 2002. Dynamic accumulators and application to efficient revocation of anonymous credentials. In *Advances in Cryptology — CRYPTO 2002*, M. Yung, Ed. Lecture Notes in Computer Science, vol. 2442. Springer Verlag, 61–76.

CAMENISCH, J. AND LYSYANSKAYA, A. 2003. A signature scheme with efficient protocols. In *Security in Communication Networks, Third International Conference, SCN 2002*, S. Cimato, C. Galdi, and G. Persiano, Eds. Lecture Notes in Computer Science, vol. 2576. Springer Verlag, 268–289.

CAMENISCH, J. AND LYSYANSKAYA, A. 2004. Signature schemes and anonymous credentials from bilinear maps. In *Advances in Cryptology — CRYPTO 2004*, M. K. Franklin, Ed. Lecture Notes in Computer Science, vol. 3152. Springer Verlag, 56–72.

CAMENISCH, J. AND MICHELS, M. 1999. Proving in zero-knowledge that a number n is the product of two safe primes. In *Advances in Cryptology — EUROCRYPT '99*, J. Stern, Ed. Lecture Notes in Computer Science, vol. 1592. Springer Verlag, 107–122.

CAMENISCH, J. AND STADLER, M. 1997. Efficient group signature schemes for large groups. In *Advances in Cryptology — CRYPTO '97*, B. Kaliski, Ed. Lecture Notes in Computer Science, vol. 1296. Springer Verlag, 410–424.

CAMENISCH, J. L. 1998. Group signature schemes and payment systems based on the discrete logarithm problem. Ph.D. thesis, ETH Zürich. Diss. ETH No. 12520, Hartung Gorre Verlag, Konstanz.

CHAN, A., FRANKEL, Y., AND TSIOUNIS, Y. 1998. Easy come – easy go divisible cash. In *Advances in Cryptology — EUROCRYPT '98*, K. Nyberg, Ed. Lecture Notes in Computer Science, vol. 1403. Springer Verlag, 561–575.

CHAUM, D. 1981. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM 24,* 2 (Feb.), 84–88.

CHAUM, D. 1983. Blind signatures for untraceable payments. In *Advances in Cryptology — Proceedings of CRYPTO '82*, D. Chaum, R. L. Rivest, and A. T. Sherman, Eds. Plenum Press, 199–203.

CHAUM, D. 1985. Security without identification: Transaction systems to make big brother obsolete. *Communications of the ACM 28,* 10 (Oct.), 1030–1044.

CHAUM, D. AND EVERTSE, J.-H. 1987. A secure and privacy-protecting protocol for transmitting personal information between organizations. In *Advances in Cryptology — CRYPTO '86*, M. Odlyzko, Ed. Lecture Notes in Computer Science, vol. 263. Springer-Verlag, 118–167.

CHAUM, D. AND PEDERSEN, T. P. 1993. Wallet databases with observers. In *Advances in Cryptology — CRYPTO '92*, E. F. Brickell, Ed. Lecture Notes in Computer Science, vol. 740. Springer-Verlag, 89–105.

CHAUM, D. AND VAN HEYST, E. 1991. Group signatures. In *Advances in Cryptology — EUROCRYPT '91*, D. W. Davies, Ed. Lecture Notes in Computer Science, vol. 547. Springer-Verlag, 257–265.

CRAMER, R., DAMGÅRD, I., AND SCHOENMAKERS, B. 1994. Proofs of partial knowledge and simplified design of witness hiding protocols. In *Advances in Cryptology — CRYPTO '94*, Y. G. Desmedt, Ed. Lecture Notes in Computer Science, vol. 839. Springer Verlag, 174–187.

DAMGÅRD, I. AND FUJISAKI, E. 2001. An integer commitment scheme based on groups with hidden order. `http://eprint.iacr.org/2001`.

FIAT, A. AND SHAMIR, A. 1987. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology — CRYPTO '86*, A. M. Odlyzko, Ed. Lecture Notes in Computer Science, vol. 263. Springer Verlag, 186–194.

FRANKEL, Y., TSIOUNIS, Y., AND YUNG, M. 1998. Fair off-line e-cash made easy. In *Advances in Cryptology — ASIACRYPT '98*, K. Kim and T. Matsumoto, Eds. Lecture Notes in Computer Science, vol. 1514. Springer Verlag, 257–270.

FUJIOKA, A., OKAMOTO, T., AND OHTA, K. 1992. A practical secret voting scheme for large scale elections. In *ASIACRYPT*, J. Seberry and Y. Zheng, Eds. Lecture Notes in Computer Science, vol. 718. Springer, 244–251.

FUJISAKI, E. AND OKAMOTO, T. 1997. Statistical zero knowledge protocols to prove modular polynomial relations. In *Advances in Cryptology — CRYPTO '97*, B. Kaliski, Ed. Lecture Notes in Computer Science, vol. 1294. Springer Verlag, 16–30.

GOLDWASSER, S., MICALI, S., AND RIVEST, R. 1988. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal on Computing 17,* 2 (Apr.), 281–308.

IBM. 2009. Cryptographic protocols of the Identity Mixer library, v. 1.0. IBM Research Report RZ3730, IBM Research. `http://domino.research.ibm.com/library/cyberdig.nsf/index.html`.

KIAYIAS, A. AND YUNG, M. 2006. Secure scalable group signature with dynamic joins and separable authorities. *IJSN 1,* 1/2, 24–45.

KIAYIAS, A., YUNG, M., AND TSIOUNIS, Y. 2004. Traceable signatures. In *Advances in Cryptology — EUROCRYPT 2004*, C. Cachin and J. Camenisch, Eds. Lecture Notes in Computer Science, vol. 3027. Springer, 571–589.

NAOR, M., PINKAS, B., AND SUMNER, R. 1999. Privacy preserving auctions and mechanism design. In *Proc. 1st ACM Conference on Electronic Commerce*.

PEDERSEN, T. P. 1992. Non-interactive and information-theoretic secure verifiable secret sharing. In *Advances in Cryptology – CRYPTO '91*, J. Feigenbaum, Ed. Lecture Notes in Computer Science, vol. 576. Springer Verlag, 129–140.

RIVEST, R. L., SHAMIR, A., AND ADLEMAN, L. 1978. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM 21,* 2 (Feb.), 120–126.

SCHNORR, C. P. 1991. Efficient signature generation for smart cards. *Journal of Cryptology 4,* 3, 239–252.

SPF INTÉRIEUR. 2005. *Instructions generales relatives à la carte d'indentité électronique*. SPF Intérieur, Service Registres de la Population et Cartes d'identité, Parc Atrium, rue des Colonies, 11, 1000 Bruxelles. `http://www.registrenational.fgov.be`.

WEISSTEIN, E. W. 2008. Prime number theorem. From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/PrimeNumberTheorem.html.

WIKIPEDIA. 2008. DSM-IV codes. http://en.wikipedia.org/wiki/DSM-IV_Codes.

WORLD HEALTH ORGANIZATION (WHO). 2005. *International Statistical Classification of Diseases and Health Related Problems (ICD-10)*, 2nd edition, 10th revision ed. World Health Organization, Geneva.

WORLD HEALTH ORGANIZATION (WHO). 2007. ICD-10 codes. http://www.who.int/classifications/apps/icd/icd10online.