COMPUTING
SCIENCE

Issuing C-L signatures on speed: signing with a constant number of exponentiations

Thomas Gross
Issuing CL-Signatures on Speed: 
Signing with a Constant Number of Exponentiations 

Thomas Groß 
School of Computing Science, Newcastle University, UK 

Abstract. 

In SCN 2002, Jan Camenisch and Anna Lysyanskaya have proposed the Strong RSA version of their Camenisch-Lysyanskaya (CL) signature scheme [8], a fundamental cryptographic building block to compute a digital signature on hidden committed messages and allow zero-knowledge proofs of knowledge on them. Ever since, the CL signature scheme has been adopted for different applications, such as anonymous credential systems, Direct Anonymous Attestation, and different prototypes for smart cards. Unfortunately, CL signatures place a significant workload on the issuer, as the signature generation requires a number of modular exponentiations linear in the number of message blocks signed, which, in turn, constitutes a significant obstacle for the broad adoption of the scheme. In this work, we propose a variant of the Strong RSA CL-signature scheme, which computes the signature with a constant number of modular exponentiations, that is, independent of the number of message blocks involved. In fact, we show that issuer can compute a commitment on an arbitrary number of message blocks with one modular exponentiation and complete the signature generation with five modular exponentiations. All the issuer needs to do is store \( n \) group elements readily available from the standard key generation with its private key and use this knowledge in the signature generation. The output of the optimized CL-issuing is fully wire-format compatible to the standard CL-issuing. We provide a comprehensive performance analysis of the optimized issuing approach, which shows that signatures with strong security parameters and even with tens of thousands of message blocks can be computed in the order of one hundred milliseconds.
Issuing CL-Signatures on Speed: 
Signing with a Constant Number of Exponentiations

Thomas Groß  
School of Computing Science, Newcastle University, UK

Abstract

In SCN 2002, Jan Camenisch and Anna Lysyanskaya have proposed the Strong RSA version of their Camenisch-Lysyanskaya (CL) signature scheme [8], a fundamental cryptographic building block to compute a digital signature on hidden committed messages and allow zero-knowledge proofs of knowledge on them. Ever since, the CL signature scheme has been adopted for different applications, such as anonymous credential systems, Direct Anonymous Attestation, and different prototypes for smart cards. Unfortunately, CL signatures place a significant workload on the issuer, as the signature generation requires a number of modular exponentiations linear in the number of message blocks signed, which, in turn, constitutes a significant obstacle for the broad adoption of the scheme. In this work, we propose a variant of the Strong RSA CL-signature scheme, which computes the signature with a constant number of modular exponentiations, that is, independent of the number of message blocks involved. In fact, we show that issuer can compute a commitment on an arbitrary number of message blocks with one modular exponentiation and complete the signature generation with five modular exponentiations. All the issuer needs to do is store $n$ group elements readily available from the standard key generation with its private key and use this knowledge in the signature generation. The output of the optimized CL-issuing is fully wire-format compatible to the standard CL-issuing. We provide a comprehensive performance analysis of the optimized issuing approach, which shows that signatures with strong security parameters and even with tens of thousands of message blocks can be computed in the order of one hundred milliseconds.

About the authors

Dr Thomas Gross is currently a tenured lecturer (assistant professor) in security, privacy and trust at the School of Computing Science at Newcastle University. He is the director of the Centre for Cybercrime and Computer Security (CCCS), a UK Academic Centre of Excellence in Cyber Security Research (ACE-CSR). His research interests are in security and privacy as well as applied cryptography and formal methods. He was a tenured research scientist in the Security and Cryptography group of IBM Research - Zurich before that and IBM's Research Relationship Manager for privacy research. Thomas received his M.Sc. (Dipl. Inf.) in Computer Science at the Saarland University, Germany, in 2004. He received his Ph.D. from the Ruhr-University Bochum, Germany, in 2009. His thesis was on the security analysis of standardized identity federation. Thomas is a member of the GI, ACM, IEEE, IACR and EATA, as well as Alumnus of the German National Academic Foundation.

Suggested keywords

graph, digital signature, zero-knowledge proof of knowledge, NP
Abstract. In SCN 2002, Jan Camenisch and Anna Lysyanskaya have proposed the Strong RSA version of their Camenisch-Lysyanskaya (CL) signature scheme [8], a fundamental cryptographic building block to compute a digital signature on hidden committed messages and allow zero-knowledge proofs of knowledge on them. Ever since, the CL signature scheme has been adopted for different applications, such as anonymous credential systems, Direct Anonymous Attestation, and different prototypes for smart cards. Unfortunately, CL signatures place a significant workload on the issuer, as the signature generation requires a number of modular exponentiations linear in the number of message blocks signed, which, in turn, constitutes a significant obstacle for the broad adoption of the scheme. In this work, we propose a variant of the Strong RSA CL-signature scheme, which computes the signature with a constant number of modular exponentiations, that is, independent of the number of message blocks involved. In fact, we show that the issuer can compute a commitment on an arbitrary number of message blocks with one modular exponentiation and complete the signature generation with five modular exponentiations. All the issuer needs to do is store $n$ group elements readily available from the standard key generation with its private key and use this knowledge in the signature generation. The output of the optimized CL-issuing is fully wire-format compatible to the standard CL-issuing. We provide a comprehensive performance analysis of the optimized issuing approach, which shows that signatures with strong security parameters and even with tens of thousands of message blocks can be computed in the order of one hundred milliseconds.

1 Introduction

Digital signature schemes are a foundational cryptographic building blocks that offer integrity and non-repudiation. The Camenisch-Lysyanskaya (CL) signature scheme [8] is an interesting primitive in this family, as it enables signatures on hidden committed messages and subsequent access on message blocks of the signature with known discrete-log based zero-knowledge proofs of knowledge [21,14,16,10,3,12]. There have been various proposals to adopt CL-signatures in different application domains, such as in Anonymous Credential Systems [7] or prototypes for smart card [2]. It is has found widespread adoption in computer systems from laptops to virtualization servers with Trusted Platform Modules (TPM) in the guise of Direct Anonymous Attestation (DAA) [5]. But there have also been recent proposals to issue CL-signatures on many message blocks, such as in large-scale signatures on cloud topologies [17].

Independent of whether we look at small devices or large servers, efficiency of the signature scheme was often named by skeptics as caveat. One of the greatest benefits
of the scheme, that all message blocks are available in the exponent, also impedes its performance. The performance for issuing as well as proving knowledge of a signature is dependent on the number of message blocks encoded: The number of modular exponentiations needed is always linear in the number of the message blocks. Let us compare that with signing an X.509 certificate with a standard RSA signature: An issuer writes all message blocks, such as certificate Version, serial number, validity, subject in a message and computes just one RSA signature on it, that is, computes one modular exponentiation. For CL-signatures that is different: As the messages blocks are encoded in the signature itself, one needs to have a separate base and needs to compute one modular exponentiation for each. In our X.509 example the issuer would need to compute more than 10 exponentiations to complete a CL-signature. Camenisch and Gross [6] have approached this problem with special encoding, the Camenisch-Gross(CG) encoding, to fold many binary and finite-set attributes into a single message block, which reduces the number of exponentiations significantly. Still, the fundamental problem remains.

A long story short: Issuing Camenisch-Lysyanskaya signatures is expensive. And one key reason for that is that multiple operations in the signature generation need modular exponentiations linear to the number of message blocks. Perhaps issuers will think twice whether they adopt a scheme that uses many exponentiations, i.e., valuable cycles on highly secure infrastructure, for which the computation time is even dependent on the size of the message, when they could compute an RSA signature for a certificate with just one exponentiation. The higher computation time means a lower throughput, which may indeed create an obstacle for high-frequency issuing such as for identity systems with millions of users.

In this research, we tackle a key problem identified: that issuing standard CL-signatures costs modular exponentiations linear in the number of message blocks. We present for the first time a variant of the CL-issuing that uses only a constant number of modular exponentiations wrt. the number of message blocks. In fact, the issuer can compute a commitment in \textit{one modular exponentiation} and the core signature generation in \textit{five modular exponentiations}, with an exponent bitlength in the size of the group order. This is in the same order as computing a standard RSA signature.

How is this improvement possible? We propose to make full use of the information the issuer has readily available. In particular, according to the standard issuing of CL signatures [18], the issuer is privy to the discrete logarithms between the message bases and the group generator. We modify the signature scheme to store this knowledge as part of the issuer’s secret key, that is, to store one group element for each message base. This additional knowledge allows the issuer to lift the computations necessary for the issuing to the exponent group by a discrete-logarithm reduction. To complete the computation, the issuer can then perform a single modular exponentiation in the normal group. The discrete-logarithm reduction we employ here, that is, lifting computations to the exponent and using discrete logarithms to bridge between bases, is already a known concept in cryptography. We are the first, however, to recognize that the SRSA Camenisch-Lysysanskaya key generation has all information at its disposal to employ the reduction throughout the issuing process, which yields a strong efficiency advantage for an optimized CL-issuing.
The issuing scheme we present is highly performant, even for CL-signatures with tens of thousands message blocks. In our comprehensive performance analysis, we come to the conclusion that even such signatures can be computed in the order of 100–300 ms. This performance is strictly better than the standard CL-issuing, and by many orders of magnitudes better. It enables issuing of signatures with many message blocks, which was previously thought unviable. It thereby enables new application areas such as signatures large data structures or network topology graphs.

**Our contributions** are, first, in the conceptual evaluation of the discrete-logarithm reduction for an efficient issuing of Camenisch-Lysyanskaya signatures. While the idea of using discrete logarithms and a lifting to the exponent group is already known, this is the first time that this technique has been systematically employed for the CL-issuing. It applies all issuing steps from commitment computation and verification, to signing and credential updating. Thus, we see the merits of this work mostly in lessons learned for efficient implementation. Second, we offer a detailed specification on an optimized version of the CL-issuing that makes good use of the technique. The optimized issuing only requires a small constant number of modular exponentiations: Whereas the standard CL-issuing takes on the order of number of message blocks plus five exponentiations, the optimized issuing only takes five. Third, we offer a detailed asymptotic and experimental performance evaluation based on the function of the Identity Mixer library and strong security parameters, which shows clearly that the optimized CL-issuing offers advantages with signatures with few message blocks as well as signatures with message blocks previously thought unfeasible. The presented optimization makes the Camenisch-Lysyanskaya signature scheme more viable for broad adoption. In fact, we believe that the optimized CL-issuing opens the door for completely new applications of the signature scheme.

2 Preliminaries

2.1 Assumptions

**Special RSA Modulus** A special RSA modulus has the form \( N = pq \), where \( p = 2p' + 1 \) and \( q = 2q' + 1 \) are safe primes, the corresponding group is called special RSA group.

**Strong RSA Assumption** [20,1,16]: 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 \( h^e \equiv g \mod N \). The modulus \( N \) is of a special form \( pq \), where \( p = 2p' + 1 \) and \( q = 2q' + 1 \) are safe primes.

**Quadratic Residues** The set \( \text{QR}_N \) is the cyclic subgroup of Quadratic Residues of a special RSA group with modulus \( N \). The order \( \text{ord}(\text{QR}_N) = p'q' \) is the cardinality of the set. We write \( \langle S \rangle = \text{QR}_N \) to state that this set is generated by a generator \( S \). The order \( \text{ord}(S) \) is the period of \( S \) and defined as the smallest number, such that \( S^{\text{ord}(S)} \equiv 1 \mod N \), where \( \text{ord}(S) = p'q' \).

2.2 Integer Commitments

Damgård and Fujisaki [14] showed for the Pedersen commitment scheme [19] that if it operates in a special RSA group and the committer is not privy to the factorization of the
Commitment \( \tilde{y} = g^{\tilde{a}} h^{\tilde{b}} \), where \( \tilde{a} \) and \( \tilde{b} \) are uniformly chosen random numbers.

Challenge \( c = H(\text{context} || y || \tilde{y} || m) \)

Responses \( \hat{a} = \tilde{a} + c \alpha \) and \( \hat{b} = \tilde{b} + c \beta \)

The prover sends \((y, c, \hat{a}, \hat{b})\) to the verifier.

Verification The verifier computes \( \hat{y} = y^{-c} g^{\hat{a} h^{\hat{b}}} \) and \( \hat{c} = H(\text{context} || \hat{y} || m) \).

The verifier accepts if \( \hat{c} = c \), otherwise the verifier aborts.

Fig. 1. Compilation of a signature on \( m \) with \( y = g^{\alpha} h^{\beta} \) to a non-interactive \( \Sigma \)-protocol.

modulus, then the commitment scheme can be used to commit to integers of arbitrary size. The commitment scheme is information-theoretically hiding and computationally binding. The security parameter is \( \ell \). The public parameters are a group \( G \) with special RSA modulus \( N \), and generators \((g_0, \ldots, g_m)\) of the cyclic subgroup \( \text{QR}_N \). In order to commit to the values \((V_1, \ldots, V_l) \in (\mathbb{Z}_n^*)^l \), pick a random \( R \in \{0, 1\}^\ell \) and set \( C = g_0^R \prod_{i=1}^l g_i^{V_i} \).

## 2.3 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 [21] or a composite [14,16], (2) proof of knowledge of equality of representation modulo two (possibly different) composite [10] moduli, (3) proof that a commitment opens to the product of two other committed values [4,10], (4) proof that a committed value lies in a given integer interval [3,10], and also (5) proof of the disjunction or conjunction of any two of the previous [13]. These protocols modulo a composite are secure under the strong RSA assumption and modulo a prime under the discrete logarithm assumption.

Proofs as described above can be expressed in the notation introduced by Camenisch and Stadler [11]. For instance,

\[
PK \{(\alpha, \beta, \delta) : y = g^\alpha h^\beta \land \tilde{y} = g^{\tilde{a}} h^{\tilde{b}} \land (u \leq \alpha \leq v)\}
\]

denotes a “zero-knowledge Proof of Knowledge of integers \( \alpha, \beta, \) and \( \delta \) such that \( y = g^\alpha h^\beta \) and \( \tilde{y} = g^{\tilde{a}} h^{\tilde{b}} \) holds, where \( u \leq \alpha \leq v \),” 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 [15] 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 an actual protocol implementing the proof, where Fig. 1 outlines the compilation of the SPK above in the Schnorr-proof variant that the Identity Mixer library [18]. While most this computation is standard, highlight the prover’s dominant computation is on the commitment \( \tilde{y} \) with modular exponentiations in the number of secrets proven \( n \). For the verifier, we have \( n + 1 \) modular exponentiations in the computation of his version of the commitment \( \hat{y} \).
2.4 Camenisch-Lysyanskaya Signatures

Let us introduce Camenisch-Lysyanskaya (CL) signatures in a Strong RSA setting [9].

Let $\ell_M$, $\ell_e$, $\ell_N$, $\ell_r$ and $L$ be system parameters; $\ell_r$ is a security parameter, $\ell_M$ the message length, $\ell_e$ the length of the Strong RSA problem instance prime exponent, $\ell_N$ the size of the special RSA modulus. The scheme operates with a $\ell_N$-bit special RSA modulus. Choose, uniformly at random, $R_0, \ldots, R_{L-1}, S, Z \in \mathbb{QR}_N$.

The public key $pk(I)$ is $(N, R_0, \ldots, R_{L-1}, S, Z)$, the private key $sk(I)$ the factorization of the special RSA modulus. The message space is the set $\{(m_0, \ldots, m_{L-1}) : m_i \in \pm \{0,1\}^{\ell_M}\}$.

Signing hidden messages. 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 $\ell_v = \ell_N + \ell_M + \ell_r$. Compute

$$A = \left( \frac{Z}{R_0^{m_0} \cdots R_{L-1}^{m_{L-1}} S^v} \right)^{1/e} \mod N.$$ 

The signature consists of $(e, A, v)$.

To sign hidden messages, user $U$ commits to $l$ values $V$ in an integer commitment $C$ and proves knowledge of the representation of the commitment. The issuer $I$ verifies the structure of $C$, which as we have seen in Fig. 1 will involve number of message blocks $l$ plus one modular exponentiations. Then, $I$ signs the commitment:

$$A = \left( \frac{Z}{CR_0^{m_0} \cdots R_{L-1}^{m_{L-1}} S^v} \right)^{1/e} \mod N,$$

which costs $L - l + 1$ modular exponentiations. The user completes the signature as follows: $\sigma = (e, A, v)$.

To verify that the tuple $(e, A, v)$ is a signature on message $(m_0, \ldots, m_{L-1})$, check that the following statements hold: $Z \equiv A^v R_0^{m_0} \cdots R_{L-1}^{m_{L-1}} S^v \mod N$, $m_i \in \pm \{0,1\}^{\ell_M}$, and $2^{\ell_e} > e > 2^{\ell_e-1}$ holds.

**Theorem 1.** [9] The signature scheme is secure against adaptive chosen message attacks under the strong RSA assumption.

**Proving Knowledge of a Signature.** The prover randomizes $A$: Given a signature $(A, e, v)$, the tuple $(A' := AS^{-r} \mod N, e, v' := v + er)$ is also a valid signature as well. Now, provided that $A \in \langle S \rangle$ and that $r$ is chosen uniformly at random from $\{0,1\}^{\ell_N+\ell_e}$, the value $A'$ is distributed statistically close to uniform over $Z_N^*$. Thus, the user could compute a fresh $A'$ each time, reveal it, and then run the protocol

$$PK\{(\varepsilon, v', \mu_0, \ldots, \mu_{L-1}) :$$
$$Z \equiv \pm R_0^{m_0} \cdots R_{L-1}^{m_{L-1}} A^{v'} S^{v'} \mod N \land$$
$$\mu_i \in \pm \{0,1\}^{\ell_M} \land \varepsilon \in [2^{2e-1} + 1, 2^{2e-1} - 1]\}$$
3 Optimized Issuing of Camenisch-Lysyanskaya Signatures

The key idea of our work is that the issuer is privy to the discrete logarithms of the message bases and can use this knowledge together with his knowledge of the group order to lift multi-base exponentiations to the exponent group $\mathbb{Z}_{\text{ord}(G)}$. We call this step the discrete-logarithm reduction:

**Definition 1 (Discrete-Logarithm Reduction).** For a multiplicative cyclic group $G$ with known group order $\text{ord}(G)$ and generator $g$, such that $G = \langle g \rangle$, we define the discrete logarithm reduction as follows: On input of bases $h_0, \ldots, h_\ell$, known discrete logarithms $x_i = \log_S h_i$ for $0 \leq i \leq \ell$, and a description of a multi-base exponentiation in $G$ $[C = h_0^{m_0} \cdots h_\ell^{m_\ell} g^r]$, one computes

$$x \equiv \sum_{i=0}^{\ell} (x_i m_i) + r \pmod{\text{ord}(G)},$$

and subsequently outputs $C \equiv g^x = h_0^{m_0} \cdots h_\ell^{m_\ell} g^r$ in $G$.

In the following we re-define the CL-signature issuing and updating with the discrete-logarithm reduction in a nutshell and will see that the reduction allows us render the number of modular exponentiations independent from the number of message blocks.

3.1 Key Generation

We outline the interface of the key generation in Fig. 2. The parameters are chosen as in standard CL-Signatures [18]. The scheme operates with a $\ell_N$-bit special RSA modulus in the cyclic subgroup of the Quadratic Residues $\mathbb{QR}_N$. We implement the random choice of the bases $R_0, \ldots, R_{L-1}, S, Z \in \mathbb{QR}_N$:

- Compute a generator $S$ of $\mathbb{QR}_N$ with order $p'q'$.
- Choose uniformly random numbers $x_{R_0}, \ldots, x_{R_{L-1}}, x_Z \in R [2, p'q' - 1]$.
- Compute $R_i := S^{x_{R_i}} \pmod{N}$ for $0 \leq i \leq L - 1$ and $Z := S^{x_Z} \pmod{N}$.

By modular exponentiation with such a uniformly random exponent, $R_0, \ldots, R_{L-1}, Z$ will be uniformly distributed in $\mathbb{QR}_N$, which is how the Identity Mixer Library [18] implements the base generation. The public key $\text{pk}(l)$ is $(N, R_0, \ldots, R_{L-1}, S, Z)$ as usual. Our scheme differs from the standard implementation by setting the private key

$$\text{sk}(l) := (p', q', x_{R_0}, \ldots, x_{R_{L-1}}, x_Z).$$

Thus, we store the discrete logarithm between the message bases and generator $S$, an overhead of $L + 1$ group elements from $\mathbb{Z}_{p'q'}$, and thereby establish the precondition of the discrete-logarithm reduction from Def. 1.

3.2 Signing a Hidden Message

The interface of the revised issuing in Fig. 3 is fully compatible with standard Camenisch-Lysyanskaya signature generation and follows the issuing specification of IBM Research closely [18].

---

1 Note that the specification [18] calls $M_c, M_h$ and $M_k$: $A_c, A_h$ and $A_k$. We have consistently renamed those index sets to avoid confusion with the signature element $A$. 

**Inputs:** The following system parameters are given:
- \( \ell_N \): bitlength of the special RSA modulus,
- \( \ell_M \): bitlength of the message space,
- \( \ell_e \): bitlength Strong RSA problem instance prime exponent,
- \( \ell_v \): security parameter,
- \( L \): total number of message bases.

**Outputs:** The issuer outputs a CL-keypair with an enhanced private key:
- \( \text{pk}(l) = (N, R_0, \ldots, R_{L-1}, S, Z) \) and \( \text{sk}(l) = (p', q', x_{R_0}, \ldots, x_{R_{L-1}}, x_Z) \).

Signature proof of knowledge on the correct base generation (all congruences \( \bmod N \)):
- \( \text{SPK}\{ (\alpha_0, \ldots, \alpha_{L-1}, \alpha_Z) : R_0 \equiv \pm S^\alpha \bmod R_{L-1} \equiv \pm S^\alpha \bmod Z \equiv \pm S^\alpha \} \)

**Complexity:** Probabilistic generation of two safe primes and a generator \( S \) of QR\(_N\). \( 2L + 2 = O(L) \) modular exponentiations.

**Public key size:** \( L + 3 \) group elements. **Private key size:** \( L + 3 \) group elements.

**Fig. 2. Interface of the Key Generation.**

**Inputs:** Commitment of the user \( U = S^{\nu'} \prod R_j^{m_j} \bmod N \), for \( j \in (M_e \cup M_h) \),
- \( P_1 = \text{SPK}\{ (\nu', \mu_j) | j \in (M_e \cup M_h) \} : U \equiv \pm S^{\nu'} R_j^{\mu_j} \bmod N \} \).
- \( m_i \) for \( i \in M_k \). CL-key pair: \( \text{pk}(l) = (N, R_0, \ldots, R_{L-1}, S, Z) \) and \( \text{sk}(l) = (p', q', x_{R_0}, \ldots, x_{R_{L-1}}, x_Z) \).

**Outputs:** \( (A, e, v'') \), \( P_2 = \text{SPK}\{ (e^{-1}) : A \equiv \pm Q^{-1} \bmod N \} \) and \( (m_i)_{i \in M_k} \).

**Complexity:** Probabilistic generation of prime \( e \), EEA to find \( e^{-1} \bmod (p'q') \), overall five modular exponentiations \( O(1) \) wrt. number of message blocks \( L \).

**Fig. 3. Interface of the Signature Generation (Round 2).**

**Verification of SPK \( P_1 \).** The issuer first needs to recompute the view of the user’s commitment:

\[
\hat{U} = U^{-e} S^{\nu'} \prod_{j \in (M_e \cup M_h)} R_j^{m_j} \bmod N \quad \text{[18]}
\]

We compute this value with the discrete-logarithm reduction:

\[
x_{\hat{U}} := \hat{v}' + \sum_{j \in (M_e \cup M_h)} (x_{R_j} \hat{m}_j) \bmod p'q' \quad \text{and} \quad \hat{U} := U^{-e} S^{\nu'} \bmod N. \quad (1)
\]

The remainder of the SPK-verification is as specified in the preliminaries, §2. The verification thereby takes two modular exponentiations, instead of \(|(M_e \cup M_h)| + 2 = O(L) \).

**Generation of the Signature.** The issuer chooses a random prime \( e \).

**Remark 1 (Prime Generation as Black Box).** We treat the computation of a random prime as a black-box algorithm and do not count its internal exponentiations. The expected runtime of a Rabin-Miller-based prime generation of \( e \) is \( O(\ell_e^4 + \ell_p \ell_e^3) \) [22].

The issuer chooses a random integer \( \hat{v}' \in \{0, 1\}^{\ell_e - 1} \) computes \( v'' := 2^{\ell_e - 1} + \hat{v} \). The remaining specification requires to establish \( Q \):

\[
Q = US^{v''} \prod_{i \in M_k} R_i^{x_i} \bmod N \quad \text{[18]}
\]
We employ the discrete-logarithm reduction again for the remaining computations of the signature generation:

\[ x_Q := x_Z - v'' - \sum_{i \in M_k} (x_i m_i) \mod p'q' \quad \text{and} \quad Q := \frac{S^{x_Q}}{U} \tag{2} \]

The issuer finishes the signature generation with

\[ A := Q^{-1} \mod p'q' \mod N \quad \text{and} \quad P_2 := \text{SPK}\{(e^{-1}) : A \equiv \pm Q^{-1} \mod N\} \]

and sends \((A, e, v'')\), \(P_2\) and \((m_i)_{i \in M_k}\) to the user.

Instead of \(|(M_k)| + 3 = O(L)\), the signature generation takes three modular exponentiations (counting the SPK). Altogether, instead of \(|(M_c \cup M_h)| + |(M_k)| + 5 = O(L)\), we have five modular exponentiations for the entire process independent from the number of message blocks \(L\).

### 3.3 Credential Update

The Identity Mixer library [18] allows issuers to update a CL-signature as shown in Fig. 4, operating on message block deltas: \(\Delta m_i := \bar{m}_i - m_i\) for \(i \in M_k\). We mention the discrete-logarithm reduction for the credential update for completeness.

\[
\bar{Q} = \prod_{i \in M_k} \frac{Q}{R_i^{xR_i \Delta m_i}} \mod N \tag{18}
\]

\[
x_{\bar{Q}} := v'' + \sum_{i \in M_k} (xR_i \Delta m_i) \mod p'q' \quad \text{and} \quad \bar{Q} := \frac{Q}{S^{x_{\bar{Q}}}} \mod N \tag{3}
\]

\(\text{SPK}\{(e^{-1}) : A \equiv \pm Q^{-1} \mod N\}\) is computed as above. Instead of \(|(M_k)| + 3 = O(L)\), the issuer takes three modular exponentiations.

**Inputs:** \(Q\) from a previous signature generation, old and updated message blocks \((m_i)_{M_k}\) and \((\bar{m}_i)_{M_k}\).

**CL-key pair:** \(\text{pk}(l) = (N, R_0, \ldots, R_{L-1}, S, Z)\) and \(\text{sk}(l) = (p', q', xR_0, \ldots, xR_{L-1}, x_Z)\).

**Outputs:** \((A, e, v''), P_2 = \text{SPK}\{(e^{-1}) : A \equiv \pm Q^{-1} \mod N\}\) and \((m_i)_{i \in M_k}\).

**Complexity:** Probabilistic generation of prime \(e\), EEA to find \(e^{-1} \mod p'q'\), and three modular exponentiations.

Fig. 4. Interface of the Credential Update.

### 4 Analysis and Correctness Proof

We started from the observation that the issuer of Camenisch-Lysyanskaya Signatures knows the discrete logarithms between the generator \(S\) of the Quadratic Residue group \(\text{QR}_n\), and the remaining bases of the issuer’s public key.
Recall from §2 that $S$ is a generator of the Quadratic Residues $\text{QR}_n$ with order $\text{ord}(S) = \text{ord}(\langle S \rangle) = p'q'$. In the key generation, these bases are computed by modular exponentiations with uniformly chosen randomness $x_Z, x_{R_0}, \ldots, x_{R_{L-1}} \in R \ [2, p'q' - 1]$, as follows:

$$Z = S^{x_Z} \mod N \quad \text{and} \quad R_i = S^{x_{R_i}} \quad \text{for} \ 0 \leq i \leq L - 1.$$  

Given this key generation, the issuer can store the discrete logarithms between $S$ and the bases $x_Z, x_{R_0}, \ldots, x_{R_{L-1}}$ as part of its secret key as proposed in §3.1. This is secure as a malicious issuer could store the discrete logarithms from its key generation anyway.

In the following sections, we evaluate whether the discrete-logarithm reduction yields the same results as the computations prescribed in the specification [18]. We establish this for the archetypical computations for commitments and the signature generation and derive correctness of the concrete computations of the issuing protocol as a consequence.

### 4.1 Commitment Structure Computation

The issuer needs to compute multi-base exponentiations of the form of an integer commitment() for bases re-indexed to $0, \ldots, \ell$ wlog.:

$$C = R_0^{m_0} \ldots R_\ell^{m_\ell} S^v \mod N.$$  

We claim that the issuer can compute a single modular exponentiation instead by reducing the equation with the known discrete logarithms, Def. 1:

$$x_C := x_{R_0}^{m_0} + \ldots + x_{R_\ell}^{m_\ell} + v \mod \text{ord}(S) \quad \text{and} \quad C := S^{x_C} \mod N \quad (4)$$  

**Theorem 2 (Commitment computation with knowledge of base discrete-logarithms).**

*Given the knowledge of base discrete logarithms $x_Z, x_{R_0}, \ldots, x_{R_\ell} \in R \ [2, p'q' - 1]$ and the knowledge of the order of the Quadratic Residue group $\text{ord}(\text{QR}_N) = p'q'$ an issuer can compute a multi-base exponentiation $C = R_0^{m_0} \ldots R_\ell^{m_\ell} S^v \mod N$ in a single modular exponentiation in a discrete-logarithm reduction:*

$$C := S^{(x_{R_0}^{m_0} + \ldots + x_{R_\ell}^{m_\ell} + v \mod p'q')} \mod N.$$  

**Proof (Correctness commitment computation).** With the knowledge of the discrete logarithms between the bases and generator $S$, we transform the given multi-base exponentiation into a modular exponentiation by substituting the bases as expressions of $S$ and the discrete logarithms $x_Z, x_{R_0}, \ldots, x_{R_\ell}$:

$$C \equiv (R_0)^{m_0} \ldots (R_\ell)^{m_\ell} S^v \mod N \quad (\text{mod} \ N)$$

$$\equiv (S^{x_{R_0}})^{m_0} \ldots (S^{x_{R_\ell}})^{m_\ell} S^v \mod N \quad (\text{mod} \ N) \quad | \ \text{Knowledge DLs} \ x$$

$$\equiv S^{(x_{R_0}^{m_0} + \ldots + x_{R_\ell}^{m_\ell} + v \mod p'q')} \mod N \quad (\text{mod} \ N) \quad | \ \text{ord}(S) = 1 \ (\text{mod} \ N) \quad (5)$$

$$\equiv S^{(x_{R_0}^{m_0} + \ldots + x_{R_\ell}^{m_\ell} + v \mod p'q')} \mod N \quad (\text{mod} \ N) \quad | \ \text{ord}(S) = p'q' \quad (6)$$
In Equation 5, we perform a standard exponent reduction as usually performed with Euler’s Theorem. Given that 

\[ S^{\text{ord}(S)} \equiv 1 \pmod{N} \]

we have 

\[ S^m \equiv S^{\text{ord}(S)k+m'} \equiv S^m \pmod{\text{ord}(S)} \pmod{N}. \]

In Equation 6, we use that the issuer knows the factorization of special RSA modulus \( N \) and the group order of \( \text{QR}_N \). We conclude that a commitment structure computed with the discrete-logarithm reduction is congruent with the a commitment structure computed in the standard version. In consequence, the issuer can compute exponent \( x_C \) as a polynomial in \( \mathbb{Z}_{p'q'} \) and perform just one modular exponentiation with it:

\[ x_C := x_{R_0}m_0 + \ldots + x_{R_L}m_L + v \pmod{p'q'}, \text{by which} \]

\[ S^x \equiv R_0^{m_0} \cdot R_L^{m_L} S^v \pmod{N} \]

**Lemma 1.** The signature verification of the SPK in Equation 1 of §3 with the discrete-logarithm reduction is correct.

### 4.2 Signature Generation and Credential Update

Let us consider the final computation of a Camenisch-Lysyanskaya (CL) signature:

\[ A = \left( \frac{Z}{CR_0^{m_0} \cdots R_{L-1}^{m_{L-1}} S^v} \right)^{1/e} \pmod{N}, \]

in which \( C \) is an integer commitment provided by the user. All other quantities are expressible with the generator \( S \) and the known discrete logarithms \( x_{Z}, x_{R_0}, \ldots, x_{R_{L-1}} \).

**Theorem 3 (Signature computation with knowledge of base discrete logarithms).** Given the knowledge of base discrete logarithms \( x_{Z}, x_{R_0}, \ldots, x_{R_{L-1}} \in \mathbb{R} \) \( [2, p'q' - 1] \) and the knowledge of the order of the Quadratic Residue group \( \text{ord}(\text{QR}_N) = p'q' \) an issuer can compute a multi-base exponentiation

\[ A = \left( \frac{Z}{CR_0^{m_0} \cdots R_{L-1}^{m_{L-1}} S^v} \right)^{1/e} \pmod{N} \]

in a product of two modular exponentiation of the form

\[ A = (C^{-1})^{1/e} \left( S^{1/e(x_{Z} - x_{R_0}m_0 - \ldots - x_{R_{L-1}}m_{L-1} - v')} \pmod{p'q'} \right) \pmod{N}. \]
Proof (Correctness of signature computation). Following the method described above, we transform the signature issuing to a single modular exponentiation:

\[
A \equiv \left( \frac{Z}{R_1^{m_1} \cdots R_L^{m_L-1} S^{v'}} \right)^{1/e} \pmod{N}
\]

\[
A \equiv \left( \frac{S^z}{C(S^z R_1)_{m_1} \cdots (S^z R_L)_{m_L-1} S^{v'}} \right)^{1/e} \pmod{N}
\]

\[
A \equiv \left( \frac{1}{C^{s^z x}(S^z R_1)_{m_1} \cdots (S^z R_L)_{m_L-1} S^{v'}} \right)^{1/e} \pmod{N}
\]

\[
A \equiv \left( \frac{1}{S^{x z}}(x R_1 m_1 + \ldots + x R_{L-1} m_{L-1} + v')^{1/e} \pmod{p' q'} \right) \pmod{N}
\]

\[
A \equiv \left( S^{1/e(x Z - x R_1 m_1 - \ldots - x R_{L-1} m_{L-1} - v')} \pmod{p' q'} \right) \pmod{N}
\]

The signature computed with discrete-logarithm reduction is therefore congruent to the one computed in the standard version. The issuer can compute

\[
x := \frac{1}{e}(x Z - x R_1 m_1 - \ldots - x R_{L-1} m_{L-1} - v') \pmod{p' q'}
\]

and use that for a modular exponentiation to compute the signature efficiently:

\[
A := C^{-1/e} S^x \pmod{N}
\]

Lemma 2 (Correctness of Signature Generation and Credential Update). The signature generation in Equation 2 of §3.2 and in Equation 3 of §3.3 with the discrete-logarithm reduction is correct.

5 Performance Evaluation

5.1 Asymptotic Complexity

The issuer it needs to store number of message bases \( L \) group elements from \( \mathbb{Z}_{p' q'} \) in its private key. We measure performance in the modular exponentiations, where we treat probabilistic generation of primes and generator \( S \) as black box.

Definition 2 (Complexity Measure). We use modular exponentiation as principal unit of computation for this analysis. Computational complexity is expressed as number of modular exponentiations. We call the complexity constant, \( O(1) \), in the number of message blocks \( L \), if the number of modular exponentiations is independent from the number of message blocks.

We evaluate various computations, where we find a representative structure which will be further considered for the experimental evaluation:

\[
C = R_1^{m_1} \cdots R_L^{m_L} S^u \pmod{N},
\]
thus involve number of message based plus one \( n + 1 \) modular exponentiations. Measured in modular exponentiations as unit of computation, we gain a reduction from \( O(n) \) to \( O(1) \) as shown in Table 1. We see in the experimental analysis of §5.2 that this reduction is particularly visible if the CL signature scheme is employed for signatures with a large number of message bases, for instance, in the area of graph signatures [17].

### Table 1. Asymptotic complexity on \#modular exponentiations (ModExp) for \#message blocks \( L \).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Key Generation §3.1</td>
<td>( L + 1 )</td>
<td>( O(L) )</td>
</tr>
<tr>
<td>Signing Hidden Message §3.2</td>
<td>(</td>
<td>M_e \cup M_h</td>
</tr>
<tr>
<td>Verification of SPK ( P_1 )</td>
<td>(</td>
<td>(M_e)</td>
</tr>
<tr>
<td>Signature Generation</td>
<td>(</td>
<td>(M_k)</td>
</tr>
<tr>
<td>Credential Update §3.3</td>
<td>(</td>
<td>(M_k)</td>
</tr>
<tr>
<td>Commitment Computation §4.1</td>
<td>( \ell + 1 )</td>
<td>( O(L) )</td>
</tr>
<tr>
<td>Signing Operation §4.2</td>
<td>( L + 2 )</td>
<td>( O(L) )</td>
</tr>
</tbody>
</table>

**Remark 2 (Complexity in Input Bitlength).** When measuring the computational complexity we may also measure the asymptotic expected time wrt. to the bitlength of the inputs (particularly significant for the probabilistic prime generation). In that case, the complexity is dependent on the bitlength of the modulus \( \ell_N \) and the bitlength of the exponents, which may be \( \ell_m, \ell_n, \ell_e, \ell_v \) or length of associated random exponents. However, the complexity differences by modulus bitlength and exponent bitlength are dominated by the number of modular exponentiations in \( L \).

### 5.2 Experimental Evaluation

We assume that issuer has established the Quadratic Residues \( \text{QR}_n \) under a special RSA modulus \( n \) as specified by the Identity Mixer Library, where the system parameters (\( \ell_n \), etc.) are chosen exactly as prescribed in the library setup [18]. The performance analysis is executed on 64-bit Java JDK 1.7.13 on a Windows 7 SP 1 Thinkpad X220 Tablet, on Intel CPU i5-2520 with 2.5 GHz, 8 GB RAM, where all computations are performed on one processor core only. The performance analysis uses the math utility functions of the Identity Mixer Library for the computation of randomness and exponentiations, that is, its MultiExp facility. Thus, both the standard and the optimized variant operate under equal conditions.

Fig. 5(a) compares the following two computations (both \( \mod N \)):

\[
C := R_0^{m_0} \cdots R_\ell^{m_\ell} S^v \quad \text{and} \quad C := S^{x_{\ell_0} m_0 + \cdots + x_{\ell_\ell} m_\ell + v \pmod{p'q'}}
\]

where the exponents are uniform random bitstrings of specified bitlength (\( |m_i| = \ell_{m_i}; |v| = \ell_v \)) and where both equations operate on the same randomness. The independent variables are the number of bases in the CL-Signature (on the \( x \)-axis) and the
modulus length (different series), the dependent variable is computation time in ms (in log-scale on the $y$-axis). The analysis is performed for modulus bitlengths of 2048 and 3072 bits to evaluate the scalability over the modulus bitlength. We also provide an analysis for small numbers of message blocks/bases in Fig 5(b): The optimization is visible from the outset, where the standard CL-issuing crosses the 100ms boundary with 8 message blocks.

**Example 1 (Throughput for eID Cards).** If one estimates the throughput of the complete issuing §3.2 with a back-of-an-envelope calculation, say for 23 message blocks as quoted in earlier studies as attribute number of an identity card [6] and a 3072-bit modulus, the standard scheme has a hourly throughput of less than 10k signatures (falling with number of bases), whereas the optimized scheme of more than 65k signatures (independent of number of bases): Issuing signatures for an entire county of 80M people takes about one year computing time with the standard scheme and 50 days with the optimized scheme.

**Example 2 (Feasibility of signing large topologies and data structures).** If one takes for granted for a moment that there may be applications for creating signatures (and enable zero-knowledge statements) on committed elements of large-scale data structures [17], then using the optimization is imperative: Assuming a signing operation for 50,000 message blocks and 3072-bit modulus, the standard CL-issuing takes roughly 15 minutes for a single signature, the optimized version 765 ms. That means that the optimized version can issue signature updates with high frequency, an important feature for dynamically changing topologies.

6 Conclusion

We study the efficiency of the issuing of Camenisch-Lysyanskaya (CL) signatures, where we tackle the problem that the signing operation costs modular exponentiations linear in the number of message blocks signed. This imposes a significant obstacle for the broad adoption of the signature scheme and for applications which operate on large numbers of message blocks. We observe that an issuer making full use of its knowledge of discrete logarithms between the group generator and bases of its public key can achieve great performance advantages. We employ a very simple technique of a discrete-logarithm reduction to lift the computation of multi-base exponentiations in all issuing steps to the exponent group. While the technique itself is already known in cryptographic folklore, this is the first time that it is consistently applied to the issuing of CL-signatures. We propose a variant of CL-issuing, which only uses a small constant number of modular exponentiations and is highly performant. Signatures for even tens of thousands of message blocks and strong key sizes can be computed in the order of hundred milliseconds. With that the CL-signing time is on a par with standard signature schemes—and we dare say that this is indeed issuing CL-signatures on speed.
(a) Large base numbers: x-axis contains the number of bases $L$ and the y-axis a log-scale of computation time in ms.

(b) Small base numbers: x-axis contains the number of bases $L$ and the y-axis a linear scale of the mean computation time over 25 runs in ms. The error bars show the standard deviation.

Fig. 5. Experimental performance analysis on standard and optimized issuing computations: $C = R_{1}^{m_{1}} \cdots R_{\ell}^{m_{\ell}} S^{v} \mod N$ (standard) and $C = S^{(e_{1} R_{1}^{m_{1}} + \cdots + e_{\ell} R_{\ell}^{m_{\ell}} + v + (\mod p'q'))} \mod N$ (optimized), where the x-axis contains the number of bases $L$ and the y-axis a log-scale of computation time in ms. The computation is performed with modulus sizes $l_{n}$ of 2048 and 3072.
References


