COMPUTING
SCIENCE

Title : Signatures and Efficient Proofs on Committed Graphs and NP-Statements
Authors: Thomas Gross
Keywords: graph, digital signature, zero-knowledge proof of knowledge, NP
Pages: 20

Abstract:

Digital signature schemes are a foundational building block enabling integrity and non-repudiation. We propose a graph signature scheme and corresponding proofs that allow a prover (1) to obtain a signature on a committed graph and (2) to subsequently prove to a verifier knowledge of such a graph signature. The graph signature scheme and proofs are a building block for certification systems that need to establish graph properties in zero-knowledge, as encountered in cloud security assurance or provenance. We extend the Camenisch-Lysyanskaya (CL) signature scheme to graphs and enable efficient zero-knowledge proofs of knowledge on graph signatures, notably supporting complex statements on graph elements. Our method is based on honest-verifier proofs and the strong RSA assumption. In addition, we explore the capabilities of graph signatures by establishing a proof system on graph 3-colorability (G3C). As G3C is NP-complete, we conclude that there exist Camenisch-Lysyanskaya proof systems for statements of NP languages.

Author Info: Thomas Gross is a Lecturer in the School of Computing Science, Newcastle University. His areas of research interest include: security and privacy, applied cryptography and human dimensions of security decision making.

TECHNICAL REPORT SERIES

No. CS-TR-1417 May 2014

TECHNICAL REPORT SERIES

No. CS-TR-1417 November, 2015

Title: Signatures and Efficient Proofs on Committed Graphs and NP-Statements
Authors: Thomas Gross

Abstract

Digital signature schemes are a foundational building block enabling integrity and non-repudiation. We propose a graph signature scheme and corresponding proofs that allow a prover (1) to obtain a signature on a committed graph and (2) to subsequently prove to a verifier knowledge of such a graph signature. The graph signature scheme and proofs are a building block for certification systems that need to establish graph properties in zero-knowledge, as encountered in cloud security assurance or provenance. We extend the Camenisch-Lysyanskaya (CL) signature scheme to graphs and enable efficient zero-knowledge proofs of knowledge on graph signatures, notably supporting complex statements on graph elements. Our method is based on honest-verifier proofs and the strong RSA assumption. In addition, we explore the capabilities of graph signatures by establishing a proof system on graph 3-colorability (G3C). As G3C is NP-complete, we conclude that there exist Camenisch-Lysyanskaya proof systems for statements of NP languages.

Printed and published by Newcastle University,
Computing Science, Claremont Tower, Claremont Road,
Newcastle upon Tyne, NE1 7RU, England.

Bibliographical details

Title : Signatures and Efficient Proofs on Committed Graphs and NP-Statements
Authors: Thomas Gross
Newcastle upon Tyne: Newcastle University: Computing Science, 2015.

(Newcastle University, Computing Science, Technical Report Series, No. CS-TR-1417)

Added entries

NEWCASTLE UNIVERSITY
Computing Science. Technical Report Series. CS-TR-1417

Abstract

Digital signature schemes are a foundational building block enabling integrity and non-repudiation. We propose a graph signature scheme and corresponding proofs that allow a prover (1) to obtain a signature on a committed graph and (2) to subsequently prove to a verifier knowledge of such a graph signature. The graph signature scheme and proofs are a building block for certification systems that need to establish graph properties in zero-knowledge, as encountered in cloud security assurance or provenance. We extend the Camenisch-Lysyanskaya (CL) signature scheme to graphs and enable efficient zero-knowledge proofs of knowledge on graph signatures, notably supporting complex statements on graph elements. Our method is based on honest-verifier proofs and the strong RSA assumption. In addition, we explore the capabilities of graph signatures by establishing a proof system on graph 3-colorability (G3C). As G3C is NP-complete, we conclude that there exist Camenisch-Lysyanskaya proof systems for statements of NP languages.

About the authors

Thomas Gross is a Lecturer in the School of Computing Science, Newcastle University. His areas of research interest include: security and privacy, applied cryptography and human dimensions of security decision making.

Suggested keywords

graph, digital signature, zero-knowledge proof of knowledge, NP Pages: 20

Signatures and Efficient Proofs on Committed Graphs and NP-Statements

Thomas Groß

School of Computing Science, Newcastle University, UK

Abstract. Digital signature schemes are a foundational building block enabling integrity and non-repudiation. We propose a graph signature scheme and corresponding proofs that allow a prover (1) to obtain a signature on a committed graph and (2) to subsequently prove to a verifier knowledge of such a graph signature. The graph signature scheme and proofs are a building block for certification systems that need to establish graph properties in zero-knowledge, as encountered in cloud security assurance or provenance. We extend the Camenisch-Lysyanskaya (CL) signature scheme to graphs and enable efficient zero-knowledge proofs of knowledge on graph signatures, notably supporting complex statements on graph elements. Our method is based on honest-verifier proofs and the strong RSA assumption. In addition, we explore the capabilities of graph signatures by establishing a proof system on graph 3-colorability (G3C). As G3C is NP-complete, we conclude that there exist Camenisch-Lysyanskaya proof systems for statements of NP languages.

1 Introduction

Digital signature schemes are foundational cryptographic primitives; they are useful in themselves to ensure integrity and non-repudiation and as building block of other systems. From their first construction by Rivest, Shamir and Adleman [27], digital signatures have been on bit-strings or group elements, on a committed sequence of bit-strings [11] or structure-preserved group elements [1]. In this work, we establish a signature scheme and corresponding proof system for committed graphs.

The basis for this work is the Camenisch-Lysyanskaya proof system: a collection of distributed algorithms that allow an issuer, a prover and a verifier to prove knowledge of committed values, issue a Camenisch-Lysyanskaya (CL) signature [10,11] on committed values, and prove knowledge of such a signature in zero-knowledge, while selectively disclosing values or proving statements about them. It uses honest-verifier Σ-proofs (Schnorr proofs [28]) and has the advantage that it keeps all attributes in the exponent. It thereby allows us to access attributes with known discrete-logarithm-based zero-knowledge proofs of knowledge [28,17,19,12,4,14]. The attributes that could be signed are, however, limited by the message space of the CL-signature scheme: a sequence of small bit-strings.

We study how to extend the Camenisch-Lysyanskaya proof system to establish signatures on committed graphs and, by extension, on committed statements from NP languages. Zero-knowledge proofs of certified or committed graphs with the capability of selective disclosure of graph elements or complex statements over graph attributes

have many significant applications beyond classical graph proof techniques [22,3] or the more recent proposal of transitive signatures [24]. The key difference to earlier work is that the graph encoding is universal, enables direct access to graph elements, and allows a prover to be flexible in the statements proven after the graph is certified. Such graph proofs are instrumental in foundational techniques, such as the zero-knowledge proof of knowledge of certified petri nets as well as in various application scenarios, such as for the certification of audited cloud topologies [2], for which propose a dedicated framework for topology proofs, including coverage, disjointness and partitions as well as connectivity and isolation, in a separate report [23].

First, we establish a new encoding of undirected graphs into the message space of CL-Signatures. The encoding allows for unlabeled, vertex- or edge-labeled graphs. The graph encoding is universal and operational in the sense that it supports efficient proofs over graph elements (vertices, edges, labels) and their relations.

Second, we extend the Camenisch-Lysyanskaya proof system to graphs by integrating the graph encoding into integer commitments and the CL-Signature bootstrapping process. This allows prover and issuer to sign committed graphs with sub-graphs contributed by both parties and to prove knowledge of graph signatures in honest-verifier Σ-proofs. The obtained graph proof system in itself allows for efficient zero-knowledge proofs of interesting graph properties, such as partitions, connectivity and isolation [23], which we demonstrate in a cloud application scenario. Graph proofs with a level of indirection between the authority on the graph (the issuer) and the verifier, established by a graph signature and with access to a wide range of graph properties, have not been covered by existing zero-knowledge graph proofs, such as [22,3,21], or transitive signatures [24]. While the former graph proofs are powerful constructions allowing for NP statements, e.g., graph 3-colorability or directed Hamiltonian cycle, their encoding does not cater for proving relations over graph elements in zero-knowledge. The latter is focused on the transitive closure along graph edges.

Third, we establish a proof system for graph 3-colorability (G3C) that allows us to obtain CL-Signatures on committed instances of 3-colorable graphs and to prove knowledge thereof to a verifier in zero-knowledge. Given that graph 3-colorability is NP-complete, we can lift the Camenisch-Lysyanskaya proof system to NP statements. Based on the 3-colorability proof system in a special RSA group and under the Strong RSA assumption, we show that there exists a Camenisch-Lysyanskaya proof system for any NP language, that is, the proof is capable of issuing CL-Signatures on committed statements from the NP language and to prove knowledge of such signatures in honest-verifier Σ-proofs. Whereas the G3C-reduction does not offer particularly efficient constructions for graph proofs, it shows the theoretical expressiveness of the graph credential system.

In effect, this work extends the reach of the Camenisch-Lysyanskaya proof system to signatures and proofs on structures of entire systems. To our knowledge, it is the first work to enable signatures on committed graphs. Notably, the graph elements are present in the exponents and, thereby, accessible to known discrete-logarithm-based zero-knowledge proofs on a wide range graph properties in honest-verifier proofs.

1.1 Outline

In §2, we discuss the preliminaries of our graph proof construction: Camenisch-Lysyaskaya signatures and Camenisch-Groß encoding. Based on the Camenisch-Groß encoding, we establish a canonical encoding for vertex- and edge-labeled graphs in §3. §4 establishes how integer commitments and CL-Signature are extended with the graph encoding. In §5, we show how this proof system is used in proofs on cloud topology signatures as a practical application scenario. We continue the main argument of the discussion in §6 to show how graph 3-colorability can be expressed in the graph proof system as proof of the encoding's theoretical reach. §8 considers earlier work on zero-knowledge proofs and signatures on graphs, while §9 draws conclusions of this work's properties.

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 [27,19]:* 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. *Quadratic Residues* The set QR_N is the set of Quadratic Residues of a special RSA group with modulus N.

2.2 Integer Commitments

Damgård and Fujisaki [17] showed for the Pedersen commitment scheme [25] that if it operates in a special RSA group and the committer is not privy to the factorization of the 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 ℓ. The public parameters are a group G with special RSA modulus N, and generators $(g_0, \ldots, g_m)$ of the cyclic subgroup 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 [28] or a composite [17,19], (2) proof of knowledge of equality of representation modulo two (possibly different) composite [12] moduli, (3) proof that a commitment opens to the product of two other committed values [5,12], (4) proof that a committed value lies in a given integer interval [4,12], and also (5) proof of the disjunction or conjunction of any two of the previous [16]. 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 [13]. For instance,

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

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 $u \leq \alpha \leq v$,*" 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 [18] 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.

2.4 Camenisch-Lysyanskaya Signatures

Let us introduce Camenisch-Lysyanskaya (CL) signatures in a Strong RSA setting [11].

Let $\ell_{\mathcal{M}}$, ℓ_e, ℓ_N, ℓ_r and L be system parameters; ℓ_r is a security parameter, $\ell_{\mathcal{M}}$ the message length, ℓ_e the length of the Strong RSA problem instance prime exponent, ℓ_N the size of the special RSA modulus. The scheme operates with a ℓ_N-bit special RSA modulus. Choose, uniformly at random, $R_0, \ldots, R_{L-1}, S, Z \in \mathrm{QR}_N$. The public key $\mathsf{pk}(\mathsf{I})$ is $(N, R_0, \ldots, R_{L-1}, S, Z)$, the private key $\mathsf{sk}(\mathsf{I})$ the factorization of the special RSA modulus. The *message space* is the set $\{(m_0, \ldots, m_{L-1}) : m_i \in \pm\{0,1\}^{\ell_{\mathcal{M}}}\}$.

Signing hidden messages. On input $m_0, \ldots, m_{L-1}$, choose a random prime number e of length $\ell_e > \ell_{\mathcal{M}} + 2$, and a random number v of length $\ell_v = \ell_N + \ell_{\mathcal{M}} + \ell_r$. To sign hidden messages, user U commits to values V in an integer commitment C and proves knowledge of the representation of the commitment. The issuer I verifies the structure of C and signs the commitment:

$$A = \left(\frac{Z}{C R_l^{m_l} \ldots R_{L-1}^{m_{L-1}} S^{v'}} \right)^{1/e} \bmod N.$$

The user completes the signature as follows: $\sigma = (e, A, v) = (e, A, (v' + R))$.

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: $Z \equiv A^e R_0^{m_0} \ldots R_{L-1}^{m_{L-1}} S^v \pmod N$, $m_i \in \pm\{0,1\}^{\ell_{\mathcal{M}}}$, and $2^{\ell_e} > e > 2^{\ell_e - 1}$ holds.

Theorem 1. *[11] 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 $(A' := AS^{-r} \bmod N, e, v' := v + er)$ 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

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

2.5 Set Membership from CL-Signatures

Set membership proofs can be constructed from CL-Signatures following a method proposed by Camenisch, Chaabouni and shelat [8]. For a set $\mathcal{S} = \{m_0, \ldots, m_i, \ldots, m_l\}$, the issuer signs all set members m_i in CL-Signatures $\sigma_i = (A, e, v)$ and publishes the set of message-signature pairs $\{(m_i, \sigma_i)\}$ integerly. To prove set membership of a value committed in C, the prover shows knowledge of the blinded signature σ_i' corresponding to the message m_i and equality of exponents with C. We explain this technique in detail in the extended version of this paper and denote a set membership proof $\mu[C] \in \mathcal{S}$, which reads μ encoded in commitment C is member of set $\mathcal{S}$.

2.6 Camenisch-Groß Encoding

The Camenisch-Groß (CG) Encoding [9] establishes structure on the CL message space by encoding multiple binary and finite-set values into a single message, and we will use a similar paradigm to encode graphs efficiently. We explain the key principles briefly and give more details in the extended version of this paper.

The core principle of the CG-Encoding is to represent binary and finite-set attribute values as prime numbers. It uses divisibility and coprimality to show whether an attribute value is present in or absent from a credential. The attribute values certified in a credential, say e_i, e_j, and e_l, are represented in a single message of the CL-Signature, by signing the product of their prime representative $E = e_i \cdot e_j \cdot e_l$ in an Integer attribute. The association between the value and the prime number of the encoding is certified by the credential issuer.

Divisibility/AND-Proof. To prove that a disclosed prime representative e_i is present in E, we prove that e_i divides the committed product E, we show that we know a secret μ' that completes the product:

$$PK\{(\mu', \rho) : \qquad D \equiv \pm (g^{e_i})^{\mu'} h^{\rho} \pmod{N}\}.$$

Coprimality/NOT-Proof. We show that one or multiple prime representatives are not present in a credential, we show coprimality. To prove that two values E and F are coprime, i.e., $\gcd(E, F) = 1$, we prove there exist integers a and b such that Bézout's Identity equals 1, where a and b for this equation do not exist, if $\gcd(E, F) > 1$.

$$PK\{(\mu, \rho, \alpha, \beta, \rho') : \qquad D \equiv \pm g^{\mu} h^{\rho} \pmod{N} \ \wedge \ g \equiv \pm D^{\alpha} (g^F)^{\beta} h^{\rho'} \pmod{N}\}.$$

OR-Proof To show that a credential contains an attribute e that is contained in an OR-list, we show there exists an integer a such that $ae = \prod_i^{\ell} e_i$; if e is not in the list, then there is no such integer a as e does not divide the product. We use the notation $\alpha \subseteq \Xi$ for an OR-proof that α contains one or more values of Ξ.

3 Graph Encoding

We consider graphs over finite vertex sets, with undirected edges or directed arcs, and finite sets of vertex and edge labels. Vertices and edges may be associated with multiple labels. We leave the encoding of directed arcs to the extended version of this paper.

$\mathcal{V}$	Finite set of vertices
$\mathcal{E} \subseteq (\mathcal{V} \times \mathcal{V})$	Finite set of edges
$\mathcal{G} = (\mathcal{V}, \mathcal{E}, t_{\mathcal{V}}, t_{\mathcal{E}})$	Graph
$\mathcal{L}_{\mathcal{V}}, \mathcal{L}_{\mathcal{E}}$	Finite sets of vertex and edge labels
$f_{\mathcal{V}} : \mathcal{V} \to \mathcal{P}(\mathcal{L}_{\mathcal{V}})$	Labels of a given vertex
$f_{\mathcal{E}} : \mathcal{E} \to \mathcal{P}(\mathcal{L}_{\mathcal{E}})$	Labels of a given edge
$n = \lvert\mathcal{V}\rvert, m = \lvert\mathcal{E}\rvert$	Number of vertices and edges

For each vertex i in $\mathcal{V}$, we introduce a vertex identifier, a prime e_i, which represents this vertex in credential and proofs. The symbol $\perp$, associated with identifier $e_\perp$ represents that a vertex is not present. All vertex identifiers are pair-wise different. We call the set of all vertex identifiers $\Xi_{\mathcal{V}}$, their product $\chi_{\mathcal{V}} = \Pi\Xi_{\mathcal{V}}$. For each label k in the label sets $\mathcal{L}_{\mathcal{V}}$ and in $\mathcal{L}_{\mathcal{E}}$, we introduce a prime representative e_k. All label representatives are pair-wise different. We call the set of all label representatives $\Xi_{\mathcal{L}}$, their product $\chi_{\mathcal{L}} = \Pi\Xi_{\mathcal{L}}$. Vertex identifiers and label representatives are disjoint:

$$\Xi_{\mathcal{V}} \cap \Xi_{\mathcal{L}} = \emptyset \quad \Leftrightarrow \quad \mathsf{gcd}(\chi_{\mathcal{V}}, \chi_{\mathcal{L}}) = 1.$$

Random Base Association We encode vertices and edges into the exponents of integer commitments and CL-Signatures and make them therefore accessible to proofs of linear equations over exponents. We randomize the base association to vertices and edges: For a vertex index set $\mathcal{V}$= 0,...,i,n-1 with vertex identifiers e_i, we choose a uniformly random permutation $\pi_{\mathcal{V}}$ of set $\mathcal{V}$ to determine the base $R_{\pi(i)}$ to encode vertex i. Edge bases $R_{\pi(i,j)}$ are chosen analogously with a random permutation $\pi_{\mathcal{E}}$.

Encoding Vertices To encode a vertex and its associated labels into a graph commitment or CL-Signature, we encode the product of the vertex identifier $e_i \in \Xi_{\mathcal{V}}$ and the prime representatives $e_k \in \Xi_{\mathcal{L}}$ for $k \in f_{\mathcal{V}}(i)$ of the labels into a single of the signature message. The product of prime representatives is encoded as exponent of dedicated vertex bases $R \in G_{\mathcal{V}}$.

Encoding Edges To get a compact encoding and efficient proofs thereon, the encoding needs to maintain the graph structure and to allow us to access it to proof higher-level properties, such as connectivity and isolation. The proposal we make in this paper after evaluating multiple approaches is to use divisibility and coprimality similar to the CG-Encoding to afford us these efficient operations over the graph structure, while offering a compact encoding of edges.

Recall that each vertex is certified with an vertex identifier from $\Xi_{\mathcal{V}}$, e.g., e_i or e_j. For each edge $(i, j) \in \mathcal{E}$, we include an edge attribute as exponent of a random edge

Table 1. Interface of the graph signature scheme.

$\mathsf{Commit}(\mathcal{G}; R)$	A PPT algorithm computing an Integer commitment on a graph.
$\mathsf{Keygen}(1^\ell, params)$	A PPT algorithm computing the key setup.
$\mathsf{HiddenSign}(C, \mathcal{V}_\mathsf{U}, \mathcal{V}_\mathsf{I}, pk_\mathsf{I})$	An interactive PPT algorithm signing a committed graph.
Private inputs:	User U: $\mathcal{G}_\mathsf{U}$, commitment randomness R; Issuer I: $\mathcal{G}_\mathsf{I}$, sk_I.
$\mathsf{Verify}(pk_\mathsf{I}, C, R', \sigma)$	A verification algorithm on graph commitment C and signature σ.

base $R_{\pi(i,j)} \in G_\mathcal{E}$, containing the product of the vertex identifiers and the associated label representatives $e_k \in \Xi_\mathcal{L}$ for $k \in f_\mathcal{E}(i,j)$ of the edge:

$$E_{(i,j)} := e_i \cdot e_j \cdot \Pi_{k \in f_\mathcal{E}(i,j)} e_k.$$

Whereas we usually consider simple graphs, specialities such as multigraphs, loops (i,i) encoded as e_i^2 or half-edges encoded as $(e_j, e_\perp)$ can be included.

Well-formed Graphs

Definition 1 (Well-formed graph). *We call a graph encoding* well-formed *iff 1. the encoding only contains prime representatives $e \in \Xi_\mathcal{V} \cup \Xi_\mathcal{L}$ in the exponents of designated vertex and edge bases $R \in G_\mathcal{V} \cup G_\mathcal{E}$, 2. each vertex base $R \in G_\mathcal{V}$ contains exactly one vertex identifier $e_i \in \Xi_\mathcal{V}$, pair-wise different from other vertex identifiers and zero or more label representatives $e_k \in \Xi_\mathcal{L}$, and 3. each edge base $R \in G_\mathcal{E}$ contains exactly two vertex identifiers $e_i, e_i \in \Xi_\mathcal{V}$ and zero or more label representatives $e_k \in \Xi_\mathcal{L}$.*

Theorem 2 (Unambiguous encoding and decoding). *A well-formed graph encoding on the integers is unambiguous modulo the base association. [Proof A.1]*

4 Signatures on Committed Graphs

CL-signatures are signatures on committed messages, where messages can be contributed by issuer and user. This translates to a user committing to a hidden partial graph $\mathcal{G}_\mathsf{U}$, which is then completed by the issuer $\mathcal{G}_\mathsf{I}$, as outline in the interface in Table 1. We establish the setup for the construction first, explain the proof of representation second, and the issuing third. We discuss notions of secrecy and imperfections of this construction in §4.1.

As a point of reference, we give the structure of the graph signatures first. We have bases $R_{\pi(i)} \in G_\mathcal{V}$, which store attributes encoding vertices, and bases $R_{\pi(i,j)} \in G_\mathcal{E}$, which store attributes encoding edges. Observe that which base stores which vertex or edge is randomized by permutations $\pi_\mathcal{V}$ and $\pi_\mathcal{E}$.

$$Z = \underbrace{\cdots R_{\pi(i)}^{e_i \Pi_{k \in f_\mathcal{V}(i)} e_k} \cdots\cdots}_{\forall \text{ vertices } i} \underbrace{R_{\pi(i,j)}^{e_i e_j \Pi_{k \in f_\mathcal{E}(i,j)} e_k} \cdots}_{\forall \text{ edges } (i,j)} A^e S^v \bmod N$$

4.1 Secrecy Notion

In a *known-graph* proof, the structure of the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is an auxiliary input to the verifier. Such a proof occurs if the prover needs to prove knowledge of a (NP-hard) property of the entire graph, e.g., a proper coloring in graph 3-colorability (cf. §6.1).

A *hidden-graph* proof keeps the structure of the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ secret. For instance, there are graph proofs in which a local property is proven and the graph structure itself kept secret, e.g., when proving that disclosed vertices of the graph are connected by a hidden path (cf. § 5).

The number of bases from $\mathcal{G}_\mathcal{V}$ and $\mathcal{G}_\mathcal{E}$ in a CL-Signature reveals an upper-bound on the number of vertices n and edges m of the signed graph. A suitable padding can be introduced by encoding nil-vertices $e_\perp$ and nil-edges $(e_\perp, e_\perp)$.

Proving properties over multiple attributes reveals which bases were involved in the proof. Characteristic patterns over said bases may interfere with the CL-Signature's multi-use unlinkability. For instance, if the prover shows that vertices i and j are connected by an edge (i, j) along with properties on the vertices themselves, the verifier will learn that the bases for the vertex identifiers e_i and e_j are related to the base for the encoding of edge (i, j). To overcome this linking, the prover can obtain a collection of CL-Signatures on the same graph, each with a randomized association between bases and vertices/edges, that is, using different random permutations $\pi_\mathcal{V}$ and $\pi_\mathcal{E}$. When proving a property over the graph the prover chooses a CL-Signature from the collection uniformly at random and proves possession over that instance.

4.2 Proof of Representation

For a full proof of representation, we need to establish that the encoded graph in a graph commitment or CL-Signature is indeed well-formed (Def. 1). Given a graph commitment C the prover and verifier engage in the following proof of representation (the proof for a CL credential work analoguously). We show that vertex bases contain a bi-partition of one and only one vertex identifier $e_i \in \Xi_\mathcal{V}$ and a set of labels $e_l \in \Xi_\mathcal{L}$. Edge bases contain a bi-partition of a product of exactly two vertex identifiers $(e_i \cdot e_j)$ and a set of labels $e_l \in \Xi_\mathcal{L}$. To prove that the representation contains exactly one vertex identifier for a vertex base and two vertex identifiers for an edge base, we establish a set membership proof.

1. Commitments The prover computes Integer commitments on the exponents of all vertex and edge bases. For each vertex i and for each edge (i, j), the prover computes commitments on vertex attribute and identifier (all $\mathrm{mod} N$)::

$$C_i = R^{e_i \Pi_{k \in f_\mathcal{V}(i)} e_k} S^r \quad \text{and} \quad \breve{C}_i = R^{e_i} S^{\breve{r}};$$

$$C_{(i,j)} = R^{e_i e_j \Pi_{k \in f_\mathcal{E}(i,j)} e_k} S^r, \quad \breve{C}_{(i,j)} = R^{e_i e_j} S^{\breve{r}} \quad \text{and} \quad \dot{C}_i = R^{e_i} S^{\dot{r}}.$$

2. Proof of knowledge. We build up the proof of possession and well-formedness step by step, where it is understood the proofs will be done in one compound proof of knowledge with *referential integrity between the secret exponents*. Let us consider a proof fragment for vertices i, j and an edge (i, j) committed in a graph commitment C (the same proof structure is used for CL-Signatures).

2.1 Proof of representation. We prove that commitment C can be decomposed into commitments C_i, C_j, one for each vertex i, j and one commitment $C_{(i,j)}$ for each edge (i, j):

$$
\begin{aligned}
&PK\{(\mu_i, \mu_j, \mu_{(i,j)}, \rho, \rho_i, \rho_j, \rho_{(i,j)}) : \\
&\quad C \equiv \pm \cdots R_{\pi(i)}^{\mu_i} \cdots R_{\pi(j)}^{\mu_j} \cdots R_{\pi(i,j)}^{\mu_{(i,j)}} \cdots S^{\rho} \pmod N \ \wedge && (1)\\
&\quad C_i \equiv \pm R^{\mu_i} S^{\rho_i} \pmod N \ \wedge \ C_j \equiv \pm R^{\mu_j} S^{\rho_j} \pmod N \ \wedge && (2)\\
&\quad C_{(i,j)} \equiv \pm R^{\mu_{(i,j)}} S^{\rho_{(i,j)}} \pmod N\}. && (3)
\end{aligned}
$$

2.2 Vertex composition. Second, we need to show properties of the vertex composition, that the encoding for each vertex i contains exactly one vertex identifier $e_i \in \Xi_{\mathcal{V}}$ and zero or multiple label representatives $e_k \in \Xi_{\mathcal{L}}$. We show this structure with help of the commitments $\breve{C}_i$ and set membership and prime-encoding OR proofs. This proof is executed for all vertices.

$$
\begin{aligned}
&PK\{(\varepsilon_i, \breve{\rho}_i, \gamma_i, \rho_i') : \\
&\quad \breve{C}_i \equiv \pm R^{\varepsilon_i} S^{\breve{\rho}_i} \pmod N \ \wedge \ C_i \equiv \pm \breve{C}^{\gamma_i} S^{\rho_i'} \pmod N \ \wedge && (4)\\
&\quad \gamma_i[C_i] \subseteq \Xi_{\mathcal{L}} \ \wedge \ \varepsilon_i[\breve{C}_i] \in \Xi_{\mathcal{V}}\}. && (5)
\end{aligned}
$$

2.3 Edge composition. Third, we prove the structure of each edge (i, j) over the commitments $C_{(i,j)}$, showing that each commitment contains exactly two vertex identifiers $e_i, e_j \in \Xi_{\mathcal{V}}$ as well as zero or more label representative $e_k \in \Xi_{\mathcal{L}}$:

$$
\begin{aligned}
&PK\{(\varepsilon_j, \rho_{(i,j)}, \gamma_{(i,j)}, \rho_{(i,j)}') : \\
&\quad \breve{C}_{(i,j)} \equiv \pm \dot{C}_i^{\varepsilon_j} S^{\rho_{(i,j)}} \pmod N \ \wedge && (6)\\
&\quad C_{(i,j)} \equiv \pm \breve{C}_{(i,j)}^{\gamma_{(i,j)}} S^{\rho_{(i,j)}'} \pmod N \ \wedge \ \gamma_{i,j} \subseteq \Xi_{\mathcal{L}}\}. && (7)
\end{aligned}
$$

2.4 Pair-wise difference. Finally, we prove pair-wise difference of vertices by showing that the vertex representatives are pair-wise co-prime over the commitments $\breve{C}_i$ and $\breve{C}_j$.

$$
PK\{(\forall i, j : \alpha_{i,j}, \beta_{i,j}, \rho_{i,j}) : \qquad R \equiv \pm \breve{C}_i^{\alpha_{i,j}} \breve{C}_j^{\beta_{i,j}} S^{\rho_{i,j}} \pmod N\}. \qquad (8)
$$

4.3 Joint Graph Issuing

To jointly issue a graph CL-signature, a user commits to a hidden partial graph and the issuer adds further elements to the graph (cf. §2.4)

In the setup, the issuer establishes a user vertex space and issuer vertex space, i.e., a bi-partition on vertex and edge bases, $G_{\mathcal{V}}$ and $G_{\mathcal{E}}$ and on vertex identifiers $\Xi_{\mathcal{V}}$. Thus, user and issuer can encode partial graphs without interfering with each other.

In the joint graph issuing, user and issuer designate and disclose connection points (vertex identifiers) that allow the user and the issuer to connect their sub-graphs deliberately. The user constructs a graph representation by choosing two uniformly random

permutation $\pi_{\mathcal{V}}$ and $\pi_{\mathcal{E}}$ for the base association on the user bases and commits to his sub-graph in a graph commitment. The user interacts with the issuer in a proof of representation of his committed sub-graph. The issuer verifies this proof, chooses uniformly random permutations for his graph elements and encodes them into his base range. The issuer creates the pre-signature of the CL-Signature scheme on the entire graph, proving that the added sub-graph is well-formed. The user completes the CL-Signature with his own randomness.

Theorem 3 (Security of graph signatures). *The graph signature scheme maintains confidentiality and integrity of the encoded graphs and offers existential unforgability against adaptive chosen message attacks under the strong RSA assumption.[Proof A.1]*

5 Application Scenario: Cloud Topology Proofs

In cloud security assurance, recent research aims to prove the security properties of a topology, where the infrastructure configuration is modelled as a graph and security properties are verified with either specialized analysis tools, model checkers or graph rewriting tools, such as in Bleikertz et al. [2]. How can a cloud provider convince a customer that the infrastructure fulfills the customer's security requirements without disclosing the topology itself?

So far, zero-knowledge proofs on clouds have been focused on Direct Anonymous Attestation (DAA) of hosts and virtual machines [6] with an optional binding of user credentials to them [7]. The graph proof acts as connective to lift DAA to attestations of entire infrastructure clouds (including their topology): We can embed DAA proofs into the context of the topology of the machine in question.

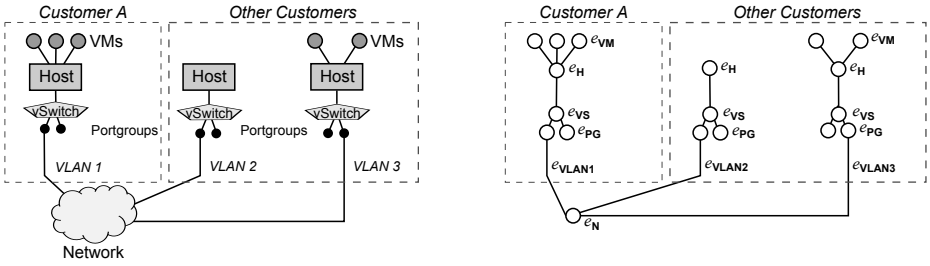

Fig. 1. Model of a cloud topology (left), encoded graph representation (right).

Let us consider the example depicted in Fig. 1 that customer A owns an DAA-verified host as well as the VLAN with ID VLAN1 and requires the provider to prove that *no other customer has access to* VLAN1, which implies isolation of network communication. As the customer has no reason to trust the cloud provider, the topology status needs to be certified by an issuer, an independent auditor. As the clouds ask for rapid provisioning, the auditor must be able to be offline at the time of the proof.

Therefore, the certification of the topology graph and the proof of properties over it needs to be decoupled: We need the capability that the auditor can create a digital

signature on the topology over which the cloud provider can dynamically create zero-knowledge proofs of knowledge of security properties relevant to the customer and integrate it with DAA-proofs of the hosts and virtual machines [7]. Hence, we have a protocol model with an independent auditor as issuer of graph signatures, the cloud provider as prover of machine and graph properties and the customer as verifier of such proofs.

We model the system in Fig. 1 by encoding all components of the topology as L graph vertices, labelled with the type of the component: $\mathcal{L}_\mathcal{V} = \{\mathsf{VM}, \mathsf{H}, \mathsf{VS}, \mathsf{PG}, \mathsf{N}\}$. Edges are encoded as indicated by the topology connections, where the edge labels represent the VLAN IDs: $\mathcal{L}_\mathcal{E} = \{\mathsf{VLAN1}, \mathsf{VLAN2}, \mathsf{VLAN3}\}$. The issuer associates prime representatives to all vertices and labels. During the issuing phase, the issuer certifies his independent analysis of the infrastructure.

We discuss a comprehensive framework for cloud topology proofs in a separate report [23] and only include an proof of isolation as an example here. To convince the verifier in zero-knowledge that *no part of the hidden graph* outside of customer A's infrastructure has access to the same VLAN VLAN1, the prover computes a graph bi-partition between the sub-graph belonging to customer A and remainder of the cloud (wlog we reorder the vertex and edge bases of the remainder to cover $0, \ldots, \ell$).

Commitments. The prover computes integer commitments $\tilde{C}_i$ on the cumulative products of vertex and edge attributes of the remainder, each with uniformly-chosen randomness r_i:

$$\tilde{C}_0 = R^{m_0} S^{r_0} \bmod N, \quad \tilde{C}_1 = R^{m_0 m_1} S^{r_1}, \ldots, \tilde{C}_\ell = R^{\Pi_0^\ell m_i} S^{r_\ell} \bmod N$$

Proof of knowledge. The prover sends the randomized graph signature A' and the commitments $\tilde{C}_i$ to the verifier and interacts with it in the following proof of knowledge:

$$\begin{aligned} PK\{&(\mu_0, \ldots, \mu_{L-1}, \varepsilon, \nu', \rho_0, \ldots, \rho_\ell, \alpha, \beta, \rho) : \\ &Z \equiv \pm R_0^{\mu_0} \cdots R_\ell^{\mu_\ell} R_{\ell+1}^{\mu_{\ell+1}} \cdots R_{L-1}^{\mu_{L-1}} A'^{\varepsilon} S^{\nu'} \pmod N \ \wedge \\ &\tilde{C}_0 \equiv \pm R^{\mu_0} S^{\rho_0} \pmod N \ \wedge \ldots \wedge \ \tilde{C}_\ell \equiv \pm \tilde{C}_{\ell-1}^{\mu_\ell} S^{\rho_\ell} \pmod N \ \wedge \\ &R \equiv \pm (R^{e_{\mathsf{VLAN1}}})^\alpha \tilde{C}_\ell^\beta S^\rho \pmod N \ \wedge \\ &\mu_i \in \pm\{0,1\}^{\ell_\mathcal{M}} \ \wedge \ \varepsilon \in [2^{\ell_e - 1} + 1, 2^{\ell_e} - 1]\}. \end{aligned}$$

Thereby, the prover convinces the verifier that the label representative e_{VLAN1} is coprime to the cumulative product of the topology sub-graph not belonging to customer A:

$$\mathsf{gcd}(e_{\mathsf{VLAN1}}, \Pi_0^\ell m_i) = 1 \quad \Leftrightarrow \quad 1 = a e_{\mathsf{VLAN1}} + b \Pi_0^\ell m_i$$

Hence, customer A can be confident that his part of the infrastructure is isolated from the remainder of the cloud by VLAN separation. The proof is efficient as it has computational and communication complexity of $O(m)$ as expected of a proof that inadvertently needs to touch all edges. Yet, are graph proofs are practical for larger infrastructures? We approach this question similarly to topology verification [2]: lifting the static treatment of the entire graph to a differential treatment on graph diffs. Thus, differential graph signatures is part of our future work.

6 Graph 3-Colorability and NP Statements

6.1 Graph 3-Colorability

We adapt the following definition from Goldreich, Michali and Wigderson [22].

Definition 2 (Graph 3-Colorability). *A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is said to be* 3-colorable *if there exists a vertex label mapping $f_{\mathcal{V}} : \mathcal{V} \to \{1, 2, 3\}$ called proper coloring such that every two adjacent vertices are assigned different color labels. This means that for each edge $(i, j) \in \mathcal{E}$ $f_{\mathcal{V}}(i) \neq f_{\mathcal{V}}(j)$. The language graph 3-colorability, denoted G3C, consists of the set of undirected graphs that are 3-colorable. Graph 3-Colorability is known to be NP-complete. [20]*

We adapt the graph 3-colorability problem to show in honest-verifier zero-knowledge that the prover knows an CL signature on an instance of a proper coloring of a given graph $\mathcal{G}$.

Without loss of generality, we assume that graph $\mathcal{G}$ is simple and connected. The three color labels $\mathcal{L} = \{1, 2, 3\}$ are encoded with three primes $\Xi_{\mathcal{L}} = \{e_1, e_2, e_3\}$. The graph is encoded with vertex identifiers $\Xi_{\mathcal{V}}$ and these vertex labels. In addition to the conditions for a well-formed graph (Def. 1), we require that each vertex base contains exactly one label representative from $\Xi_{\mathcal{L}}$, which we show with a set membership proof on the secret vertex label.

The prover shows knowledge of a proper graph coloring by showing that the product of vertex identifiers and label representatives for each pair of adjacent vertices (i, j) are coprime.

Common inputs: Graph $\mathcal{G}$, public-key of the CL-issuer.

Prover input: CL-Signature on proper coloring for G3C.

1. Credential randomization and commitments. The prover computes randomizations for the graph signature as well as for all occurrences of set membership proofs. The prover computes Integer commitments on the exponents of all vertex and edge bases. For each vertex i, the prover computes two commitments on the vertex attribute and the vertex identifier:

$$C_i = R^{e_i e_{f_{\mathcal{V}}(i)}} S^r \bmod N \quad \text{and} \quad \breve{C}_i = R^{e_i} S^r \bmod N.$$

For each edge (i, j), the prover computes the commitment:

$$\breve{C}_{i,j} = R^{e_i e_j} S^r \bmod N.$$

2. Proof of knowledge. The prover sends the commitments to the verifier. Then, prover and verifier engage in the following proof of possession over the graph signature and vertices i and j and all edges (i, j). We build upon the proof of representation and well-formedness presented in §4.2 with the following differences: Instead of proving that a vertex contains zero or multiple labels, we prove that the vertex contains *exactly one label*. Further, the proof is simplified because the edges do not contain labels. Again, we explain the proofs step by step, while it is understood that the proofs are executed as compound proof of knowledge *with referential integrity between the secret exponents*.

2.1 Possession of CL-Signature. First, we prove of possession of the graph signature and representation of the commitments. Clause 1 proves possession of the CL-Signature on the graph. The clauses 2 and 3 prove the representation on the integer commitments on signed attributes for vertices j, j and edges (i, j), and, thereby, make the attributes accessible for the analysis of the exponents.

$$
\begin{aligned}
&PK\{(\mu_i, \mu_j, \mu_{(i,j)}, \varepsilon, \nu', \rho_i, \rho_j, \rho_{(i,j)}) : \\
&\quad Z \equiv \pm \cdots R_{\pi(i)}^{\mu_i} \cdots R_{\pi(j)}^{\mu_j} \cdots R_{\pi(i,j)}^{\mu_{(i,j)}} \cdots (A')^{\varepsilon} S^{\nu'} \pmod{N} \ \wedge && (1)\\
&\quad C_i \equiv \pm R^{\mu_i} S^{\rho_i} \pmod{N} \ \wedge \ C_j \equiv \pm R^{\mu_j} S^{\rho_j} \pmod{N} \ \wedge && (2)\\
&\quad C_{(i,j)} \equiv \pm R^{\mu_{(i,j)}} S^{\rho_{(i,j)}} \pmod{N} \ \wedge && (3)\\
&\quad \mu_i, \mu_j, \mu_{((i,j))} \in \pm\{0,1\}^{\ell_{\mathcal{M}}} \ \wedge \ \varepsilon \in [2^{\ell_e - 1} + 1, 2^{\ell_e} - 1]\}
\end{aligned}
$$

2.2 Well-formedness. Second, we establish that the vertex attributes are well-formed: Clause 4 establishes the relation between C_i and $\breve{C}_i$ and, thereby, shows that a vertex attribute is bi-partitioned onto a vertex identifier and a label representative part. Clause 5 establishes that they contain exactly one vertex identifier and label representative of the certified sets $\Xi_{\mathcal{V}}$ and $\Xi_{\mathcal{L}}$.

$$
\begin{aligned}
&PK\{(\varepsilon_i, \rho_i, \gamma_i, \breve{\rho}_i) : \\
&\quad \breve{C}_i \equiv \pm R^{\varepsilon_i} S^{\rho_i} \pmod{N} \ \wedge \ C_i \equiv \pm \breve{C}^{\gamma_i} S^{\breve{\rho}_i} \pmod{N} \ \wedge && (4)\\
&\quad \gamma_i[C_i] \in \Xi_{\mathcal{L}} \ \wedge \ \varepsilon_i[\breve{C}_i] \in \Xi_{\mathcal{V}}\}. && (5)
\end{aligned}
$$

Clause 5 is different from a proof of well-formedness as introduced in §4.2, as it enforces that that vertex i contains exactly one label.

2.3 Proper coloring. Third, clauses 6 and 7 complete the statement by establishing that there is a proper coloring for the adjacent vertices i and j: Clause 6 shows that commitment $C_{(i,j)}$ is on an edge (i, j). Finally, Clause 7 establishes that the attributes for vertex i and j are coprime, by proving that Bézout's Identity equals 1. It follows that the labels of both vertices must be different.

$$
\begin{aligned}
&PK\{(\varepsilon_i, \rho'_{(i,j)}, \alpha_{(i,j)}, \beta_{(i,j)}, \rho_{(i,j)''}) : \\
&\quad \breve{C}_{(i,j)} \equiv \pm \breve{C}_j^{\varepsilon_i} S^{\rho'_{(i,j)}} \pmod{N} \ \wedge && (6)\\
&\quad R \equiv \pm C_i^{\alpha_{(i,j)}} C_j^{\beta_{(i,j)}} S^{\rho''_{(i,j)}} \pmod{N}\}. && (7)
\end{aligned}
$$

3. Verification. The verifier outputs *accept* if the proof of knowledge checks out; *reject* otherwise.

Lemma 1 (Knowledge of a CL-Signature of G3C). *The prover convinces the verifier in zero-knowledge that the prover knows a proper graph 3-coloring for known graph $\mathcal{G}$. [Proof A.2]*

Lemma 2. *The proof has an asymptotic computation complexity of $O(n+m)$ exponentiations and a communication complexity of $O(n + m)$ group elements and is thereby a polynomial time proof. [Proof A.2]*

6.2 Proofs Systems for Languages in NP

Having established a proof for certified graph 3-colorability, we can use the fact that G3C is NP-complete to establish that such Camenisch-Lysyanskaya proof systems exist for statements from other NP languages.

Definition 3. *We call a* Camenisch-Lysyanskaya proof system *a set of PPT machines Prover* P*, Verifier* V *and Issuer* I *that engage in the following protocols:*

Proof of representation $\mathsf{P} \rightarrow \mathsf{I}$: *Proof of representation on committed values* V.
Issuing $\mathsf{I} \rightarrow \mathsf{P}$: *Issuing of CL-Signature* σ *on hidden committed values* V.
Proof of possession $\mathsf{P} \rightarrow \mathsf{V}$: *Proof of possession of CL-Signature* σ.

The issuer I *can act in the role of the verifier* V *and thereby allow the bootstrapping of further CL-Signatures from the hidden v alues of existing CL-Signatures.*

Compared to a zero-knowledge proof system for an NP language, this construction offers a level of indirection: The issuer acts as auditor with authority to decide whether the statement of an NP language is fulfilled in a certain environment, and its signature binds this statement to that environment. The instance of the NP language can either be provided by the issuer or provided by the prover and verified by the issuer.

The proof follows the same strategy as one of the initial results that all languages in NP have zero-knowledge proof systems, by Goldreich, Micali and Widgerson [22]: Given a CL proof system for G3C, we use the existing poly-time NP reductions to transform any NP language statement into an instance of G3C. This instance is then encoded as a graph in a CL-Signature and knowledge of the signature proven to a verifier. Lemma 1 shows that this is a zero-knowledge proof of knowledge of a proper coloring.

Theorem 4. *Statements of languages in NP can efficiently be proven in a Camenisch-Lysyanskaya proof system based in honest-verifier zero-knowledge. [Proof A.3]*

7 Efficiency Analysis

We display the efficiency analysis for the proof predicates in Table 2, where vertex and edge composition proofs show the overhead over the basic proof of possession (cf. topology proofs [23]). We measure computational complexity in multi-base exponentiations. The communication complexity is dominated by the transmitted group elements from $\mathbb{Z}_N^*$, which is equal to the number of multi-base exponentiations (one for each Integer and Schnorr proof commitment). The most expensive proof is the complete graph representation established in the issuing, where the set membership proofs (4 MExps) and the OR-based subset proofs (6 MExps) constitute significant overhead. The square-complexity is introduced by the final disjointness proof to establish that the graph is indeed well-formed. In the down-stream proofs, the verifier trusts the issuer to only certify well-formed graphs, which allows us to reduce complexity by only the computing the proof of possession and the statement proven.

The modular exponentiations for message bases R_i are with small exponents of size of $\ell_{\mathcal{M}} \ll \ell_N$, where the parameter $\ell_{\mathcal{M}}$ can be chosen similarly small as in Direct Anonymous Attestation (DAA) [6].

Table 2. Efficiency of proofs of predicates in multi-base exponentiations (MultiExps) dependent on the number of vertices n and of edges m. For a simple graph holds $m \leq \frac{n(n-1)}{2}$. $\ell < m$ is the number of edges the unknown remainder of a graph bi-partition in §5.

Predicate	**Basis**	**Commitments**	**MultiExps**	
		#	#	O
Possession		$n + m$	$2n + 2m + 1$	$O(n + m)$
Vertex Composition	Possession	n	$3n$	$O(n)$
Edge Composition	Possession	$2m$	$4m$	$O(m)$
Total Well-formed Graph		$2n + 3m$	$n^2 + 8n + 8m + 1$	$O(n^2)$
Cloud Topology Isolation (§5)		$\ell < m$	$2\ell + 2 < 2m + 2$	$O(m)$
Graph-3 Colorability (§6)		$n + m$	$6n + 4m + 1$	$O(n + m)$

In addition, the Σ-proofs employed in this work benefit from batch-proof techniques, such as [26]. The graph proofs are likely to be transformed to signature proofs of knowledge with the Fiat-Shamir heuristic [18] and can thereby be computed offline.

8 Related Work

Establishing zero-knowledge proofs on graphs and their properties is a classic area of research. Such proofs were instrumental in showing that there exist zero-knowledge proof systems for all NP languages. We discuss their graph modelling: Goldreich, Micali and Wigderson [22] offered such a construction with $O(m^2)$ rounds and $O(n)$ messages each. Based on the existence of a nonuniformly secure encryption function, they explored graph isomorphism and non-isomorphism as well as graph 3-colorability (G3C). Blum's proof [3] shows directed Hamiltonian cycles (DHC) in graphs. Both proofs use a metaphor of locked boxes to formulate the proof. Goldreich et al.'s G3C proof encodes the colors of adjacent vertices in boxes. Blum's proof of Hamiltonian cycles encodes the graph's adjacency matrix randomly in $n + \binom{n}{2}$ such boxes, giving the verifier the choice to either verify the correct graph representation or the knowledge of the Hamiltonian cycle. Blum offers an alternative construction for G3C with a similar methodology, encoding the graph representation and the coloring of each vertex in separate yet related boxes and operating on an adjacency matrix lifted to the labeling. Goldreich and Kahan [21] offered a constant-round construction based on the existence of collections of claw-free functions, also using G3C as NP-problem. We observe that these constructions are specific to the statement to be proven and do not cater for a level of indirection through a signature scheme.

A related notion to full graph signatures is transitive signature schemes, e.g., as proposed by Michali and Rivest [24]. They are concerned with the transitive closure of signatures on graph elements, where vertices and edges are signed individually; however, they do not offer zero-knowledge proofs of knowledge on graph properties.

9 Conclusion

We have introduced a practical construction of signatures on committed graphs and zero-knowledge proofs over their structure. The scheme is special in that it enables proofs over the entire graph structure, including statements such as isolation (two vertices are not connected by any sequence of edges). The construction derives its security from the properties of the Camenisch-Lysyanskaya (CL) signature scheme under the Strong RSA assumption. The interactive proofs are honest-verifier zero-knowledge if executed with multiple rounds with small challenges. While we have established a framework for graph topology proofs separately [23], this work focuses on the foundations of graph encoding in CL-signatures itself. We show its theoretical expressiveness by proving that the scheme is capable of signing committed NP statements and proving properties thereof, via reduction to graph 3-colorability. The presented scheme is efficient and practical because once the issuer has established graph well-formedness in $O(n^2)$, the prover can resort to proofs over the graph structure in linear time. The used Σ-proofs can be handled efficiently with batch processing techniques [26]. As future work, we aim at establishing a differential graph signature scheme, which can be employed for large-scale graph topologies as found in virtualized infrastructures.

References

1. ABE, M., FUCHSBAUER, G., GROTH, J., HARALAMBIEV, K., AND OHKUBO, M. Structure-preserving signatures and commitments to group elements. In *Advances in Cryptology–CRYPTO 2010*. Springer, 2010, pp. 209–236.
2. BLEIKERTZ, S., GROSS, T., AND MÖDERSHEIM, S. Automated Verification of Virtualized Infrastructures. In *ACM Cloud Computing Security Workshop (CCSW'11)* (Oct 2011), ACM.
3. BLUM, M. How to prove a theorem so no one else can claim it. In *Proceedings of the International Congress of Mathematicians* (1986), vol. 1, p. 2.
4. BOUDOT, F. Efficient proofs that a committed number lies in an interval. In *Advances in Cryptology — EUROCRYPT 2000* (2000), B. Preneel, Ed., vol. 1807 of *Lecture Notes in Computer Science*, Springer Verlag, pp. 431–444.
5. BRANDS, S. Rapid demonstration of linear relations connected by boolean operators. In *Advances in Cryptology — EUROCRYPT '97* (1997), W. Fumy, Ed., vol. 1233 of *Lecture Notes in Computer Science*, Springer Verlag, pp. 318–333.
6. BRICKELL, E., CAMENISCH, J., AND CHEN, L. Direct anonymous attestation. In *Proc. 11th ACM Conference on Computer and Communications Security* (2004), acm press, pp. 225–234.
7. CAMENISCH, J. Protecting (anonymous) credentials with the Trusted Computing Group's TPM v1.2. In *SEC* (2006), vol. 201 of *IFIP*, Springer, pp. 135–147.
8. CAMENISCH, J., CHAABOUNI, R., AND SHELAT, A. Efficient protocols for set membership and range proofs. In *Advances in Cryptology-ASIACRYPT 2008* (2008), Springer, pp. 234–252.
9. CAMENISCH, J., AND GROSS, T. Efficient attributes for anonymous credentials. *ACM Transactions on Information and System Security (TISSEC) 15*, 1 (2012), 4.
10. CAMENISCH, J., AND LYSYANSKAYA, A. Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation. In *Advances in Cryptology — EUROCRYPT 2001* (2001), B. Pfitzmann, Ed., vol. 2045 of *LNCS*, Springer Verlag, pp. 93–118.

11. CAMENISCH, J., AND LYSYANSKAYA, A. A signature scheme with efficient protocols. In *Security in Communication Networks SCN 2002* (2003), vol. 2576 of *LNCS*, Springer Verlag, pp. 268–289.
12. CAMENISCH, J., AND MICHELS, M. Proving in zero-knowledge that a number n is the product of two safe primes. In *Advances in Cryptology — EUROCRYPT '99* (1999), J. Stern, Ed., vol. 1592 of *Lecture Notes in Computer Science*, Springer Verlag, pp. 107–122.
13. CAMENISCH, J., AND STADLER, M. Efficient group signature schemes for large groups. In *Advances in Cryptology — CRYPTO '97* (1997), B. Kaliski, Ed., vol. 1296 of *Lecture Notes in Computer Science*, Springer Verlag, pp. 410–424.
14. CHAN, A., FRANKEL, Y., AND TSIOUNIS, Y. Easy come – easy go divisible cash. In *Advances in Cryptology — EUROCRYPT '98* (1998), K. Nyberg, Ed., vol. 1403 of *Lecture Notes in Computer Science*, Springer Verlag, pp. 561–575.
15. COOK, S. A. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing* (1971), ACM, pp. 151–158.
16. CRAMER, R., DAMGÅRD, I., AND SCHOENMAKERS, B. Proofs of partial knowledge and simplified design of witness hiding protocols. In *Advances in Cryptology — CRYPTO '94* (1994), Y. G. Desmedt, Ed., vol. 839 of *LNCS*, Springer Verlag, pp. 174–187.
17. DAMGÅRD, I., AND FUJISAKI, E. An integer commitment scheme based on groups with hidden order. `http://eprint.iacr.org/2001`, 2001.
18. FIAT, A., AND SHAMIR, A. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology — CRYPTO '86* (1987), A. M. Odlyzko, Ed., vol. 263 of *Lecture Notes in Computer Science*, Springer Verlag, pp. 186–194.
19. FUJISAKI, E., AND OKAMOTO, T. Statistical zero knowledge protocols to prove modular polynomial relations. In *Advances in Cryptology — CRYPTO '97* (1997), B. Kaliski, Ed., vol. 1294 of *Lecture Notes in Computer Science*, Springer Verlag, pp. 16–30.
20. GAREY, M. R., JOHNSON, D. S., AND STOCKMEYER, L. Some simplified np-complete problems. In *Proceedings of the sixth annual ACM symposium on Theory of computing* (1974), ACM, pp. 47–63.
21. GOLDREICH, O., AND KAHAN, A. How to construct constant-round zero-knowledge proof systems for NP. *Journal of Cryptology 9*, 3 (1996), 167–190.
22. GOLDREICH, O., MICALI, S., AND WIGDERSON, A. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM 38*, 3 (1991), 690–728.
23. GROSS, T. Certification and efficient proofs of committed topology graphs. Cryptology ePrint Archive Report 2014/255, IACR, 2014. `http://eprint.iacr.org/`.
24. MICALI, S., AND RIVEST, R. L. Transitive signature schemes. In *Topics in Cryptology-CT–RSA 2002*. Springer, 2002, pp. 236–243.
25. PEDERSEN, T. P. Non-interactive and information-theoretic secure verifiable secret sharing. In *Advances in Cryptology – CRYPTO '91* (1992), J. Feigenbaum, Ed., vol. 576 of *Lecture Notes in Computer Science*, Springer Verlag, pp. 129–140.
26. PENG, K., BOYD, C., AND DAWSON, E. Batch zero-knowledge proof and verification and its applications. *ACM Transactions on Information and System Security (TISSEC) 10*, 2 (2007), 6.
27. RIVEST, R. L., SHAMIR, A., AND ADLEMAN, L. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM 21*, 2 (Feb. 1978), 120–126.
28. SCHNORR, C. P. Efficient signature generation for smart cards. *Journal of Cryptology 4*, 3 (1991), 239–252.

A Proofs

A.1 Wellformed Encoding and Security

Proof (Unambiguous encoding and decoding: Theorem 2). We show that there is a bijection between encoding and graph.
Graph $\rightarrow$ encoding: For each graph there exits a unique encoding modulo base association. For all vertices $i \in \mathcal{V}$ choose the vertex identifier $e_i \in \Xi_{\mathcal{V}}$, for the labels $k \in f_{\mathcal{V}}(i)$ choose the prime representative $e_k \in \Xi_{\mathcal{L}}$ and compute their product. As said factors are prime, it follows from the fundamental theorem of arithmetic that the $e_i \Pi_{k \in f_{\mathcal{V}}(i)} e_k$ represents a unique integer. Given that the user is not privy to the discrete logarithm between one base and another (guaranteed by the CL-Signature setup), the bases unabiguously separate the exponents. Thus, apart from the random permutation of the base association, the encoding is unambiguous.
Encoding $\rightarrow$ graph: With knowledge of the elements of $\Xi_{\mathcal{V}}$ and $\Xi_{\mathcal{L}}$, an encoded product can be decoded efficiently and unabiguously into the elements of the graph. That the parties are not privy to the discrete logarithm between base and another guarantees attribute separation. The base designates unambiguously whether a vertex or an edge is encoded. Given that all representatives of the encoding are prime, the product can be decomposed into a unique factorization by the fundamental theorem of arithmetic. Each representative unambiguously represents either a vertex identifier in $\Xi_{\mathcal{V}}$ or a label in $\Xi_{\mathcal{L}}$, as both sets are disjoint. □

Proof (Security of graph signatures: Theorem 3). The security of the scheme is directly derived from the unambiguous embedding into Integer commitments and Camenisch-Lysyanskaya signatures and their security properties. Theorem 2 establishes that the graph encoding encodes graphs unamiguously into the CL-message space. The graph structure is encoded in the exponents of the Integer commitment and CL-signature schemes. Confidentiality is derived from the information-theoretical hiding property of the Integer commitment scheme and the hiding properties of CL-signatures on committed messages. Under the condition that the adversary is not privy to the group-order of the commitment and the CL signature scheme, we obtain that integrity for both schemes holds over the integers and thereby the graph encoding (cf. [17]). We obtain existential unforgability against chosen message attacks directly from the CL-signature scheme in Theorem 1 [11].

A.2 Graph 3-Colorability (G3C)

Proof (Graph 3-Colorability: Lemma 1). **1. Proof of Knowledge.** It is standard to show that there exists a knowledge extractor for all exponents of the proof such that the equality of exponents equations are fulfilled.

We obtain from Clause 1 that the prover knows the representation of a CL-Signature of the given structure. From the existential unforgeability of CL-Signatures, we see that the issuer must have signed the secret attributes μ_i, μ_j and $\mu_{(i,j)}$. Proving equality of exponents with corresponding integer commitments is standard, by which the arguments over the commitments, such as C_i, $\breve{C}_i$ and $C_{(i,j)}$ transfer to the structure of the signed messages.

The Clause 4 shows that a message μ_i consists of two factors known to the prover: $\mu_i = \varepsilon_i \gamma_i$. The following Clause 5 employs a set membership proof to show that $\varepsilon_i \in \Xi_{\mathcal{V}}$ and that $\gamma_i \in \Xi_{\mathcal{L}}$. We use that the set membership from §2.5 guarantees that ε_i and γ_i are exactly one member of the set to conclude that a message μ_i contains exactly one vertex identifier and one label identifier. Thus, μ_i is well-formed. Similarly, Clause 6 establishes the structure $\mu_{(i,j)} = \varepsilon_i \varepsilon$ for the edge (i, j), showing it to be well-formed. Because the prover is not privy to the group order, these statements hold over the integers, by the results of Damgård and Fujisaki [17]. Therefore, with

the proof of representation including pair-wise difference, we conclude that the signed graph is well-formed.

Clause 7 shows that the labeling $f_{\mathcal{V}}$ of the signed graph is a proper coloring. Again, we employ Damgård and Fujisaki's [17] result that equations hold over the integers. We have that for each edge (i, j), the corresponding signed messages have the following structure:

$$\mu_i = \varepsilon_i \gamma_i \quad \text{and} \quad \mu_j = \varepsilon_j \gamma_j.$$

We show that the secret labels γ_i and γ_j are different by showing that μ_i and μ_j are coprime, where we use Bézout's Identity:

$$\mathsf{gcd}(\mu_i, \mu_j) = 1 \quad \Leftrightarrow \quad 1 = \alpha_{(i,j)}\mu_i + \beta_{(i,j)}\mu_j.$$

The equality of exponent proof of Clause 7 achieves this as follows

$$\begin{aligned}
R &\equiv \pm C_i^{\alpha_{(i,j)}} C_j^{\beta_{(i,j)}} S^{\rho_{(i,j)}} \pmod N \\
R^1 &\equiv \pm (R_i^{\mu} S^{\rho_i})^{\alpha_{(i,j)}} (R_j^{\mu} S^{\rho_j})^{\beta_{(i,j)}} S^{\rho_{(i,j)}} \pmod N \\
R^1 &\equiv \pm R^{\alpha_{(i,j)}\mu_i} S^{\alpha_{(i,j)}\rho_i} R^{\beta_{(i,j)}\mu_j} S^{\beta_{(i,j)}\rho_j} S^{\rho_{(i,j)}} \pmod N \\
R^1 &\equiv \pm R^{\alpha_{(i,j)}\mu_i + \beta_{(i,j)}\mu_j} S^{\alpha_{(i,j)}\rho_i + \beta_{(i,j)}\rho_j + \rho_{(i,j)}} \pmod N
\end{aligned}$$

From this equation we can conclude that $\mathsf{gcd}(\mu_i, \mu_j) = 1$ and that, therefore, $\gamma_i \neq \gamma_j$, which implies that $f_{\mathcal{V}}(i) \neq f_{\mathcal{V}}(j)$ and that the CL signature indeed contains a proper coloring. □

2. Zero-Knowledge. We claim that proof does not disclose anything else than the statement made that the prover knows a CL-Signature of a proper coloring on known graph $\mathcal{G}$.

The Σ-proofs here are zero-knowledge in an honest verifier setting if performed with multiple rounds and small challenges. It is standard to construct a simulator for all Σ-proofs of representation for the CL-Signature and the commitments as well as for their conjunction [13,16], showing that the verifier does not learn anything else than the relations on exponents shown.

It remains to be shown what the relations disclose. We will argue on the statements made on the secret messages γ_i, which contain the color. Clause 4 establishes that γ_i is part of commitment C_i, but does not disclose further information than the equality of exponents.

Clause 5 proves that γ_i is a member of the set $\Xi_{\mathcal{L}} = \{1, 2, 3\}$. This statement itself is part of the known problem definition of G3C. The set membership proof is a proof of representation for an anonymized CL-Signature and a standard proof of equality of exponents, and thereby, does not disclose further information.

Finally, Clause 7 references $\mu_i = \epsilon_i \gamma_i$ to prove that γ_i and γ_j of an adjacent edge are coprime. As the vertex identifiers are pair-wise different by definition and as all representatives are primes, this only establishes that $\gamma_i \neq \gamma_j$ as required by the G3C problem, but nothing else. □

Proof (Polynomial Proof of G3C: Lemma 2). **Precomputation:** The prover computes $2n + 1$ signature randomizations with one exponentiation each and $2n + m$ integer commitments with 2 exponentiations each. The pre-computation phase uses $6n + 2m + 1$ exponentiations, transmits $4n + m + 1$ group elements, and thereby has a computation complexity of $O(n + m)$ and a communication complexity of $O(n + m)$.

Proof of Knowledge: The Schnorr proofs in the proof of knowledge are zero-knowledge if executed with small challenges over multiple rounds and can be connected with techniques from Cramer et al. [16]. The round complexity of the overall protocol is dependent on the proof mode (cf. Brands [5]).

Clause 1 is executed once yielding a Schnorr proof with $n + m + 2$ exponentiations for the prover.

The clauses 2 are executed once for each vertex, such as i and j, Therefore we have n Schnorr proofs with 2 exponentiations each for the prover.

The clauses 3 are executed once for each edge (i, j), making m Schnorr proofs with 2 exponentiations each for the prover.

The clauses 4 are executed once for each vertex, such as i or j. We have $2n$ Schnorr proofs with 2 exponentiations each for the prover.

The set membership proofs of Clauses 5 are executed once for each vertex and its label. Each set membership proof is a proof of representation of a designated CL-Signature for the set member, amounting to 3 exponentiations for the prover. In total, we have $2n$ such proofs of possessions, all done with a single Schnorr proof proving equality of exponents with the corresponding commitment.

Clause 6 proves the edge structure and is executed once per edge, yielding m Schnorr proofs with 2 exponentiations each for the prover. Finally, the proper graph coloring in Clause 7 is shows once for each edge (i, j) amounting to m Schnorr proofs with 3 exponentiations for the prover.

The proof of knowledge of graph coloring thereby requires $5n + 3m + 1 = O(n + m)$ Schnorr proofs with a computational complexity for the prover of $13n + 8m + 2 = O(n + m)$ exponentiations.

The total computational complexity is therefore $O(n + m)$, the communication complexity is $O(n + m)$ group elements. The G3C proof is done in polynomial time. The round complexity depends on the proof mode, where variants with multiple rounds (number of rounds depending on the error probability), with four rounds and initial commitments of the verifier on challenges, and three rounds in a Σ-proof (not zero-knowledge) are possible. □

A.3 CL Proof Systems for NP-Statements

Proof (Sketch NP-Statements: Theorem 4). Let a NP language $\mathfrak{L}$ be given. Let τ be a polynomial-time computable and invertible reduction from $\mathfrak{L}$ to Graph 3-Colorability (G3C): τ can be constructed by composing a polynomial-time reduction of $\mathfrak{L}$ to 3SAT by Cook's proof [15] and a polynomial-time reduction from 3SAT to G3C. We have that $x \in \mathfrak{L}$ iff $\tau(x)$ is 3-colorable.

On common input x, both prover and verifier compute graph $G \leftarrow \tau(x)$. In Goldreich, Micali and Widgerson's work, the proof proceeds to use any interactive zero-knowledge proof system to prove that G is 3-colorable and thereby show that $x \in \mathfrak{L}$. Our proof continues from this point to show that there exists a Camenisch-Lysyanskaya proof system.

On obtaining $\mathcal{G} = \tau(x)$, the prover constructs a graph commitment C on $\mathcal{G}$ as defined in §3, including a labeling $f_{\mathcal{V}}$ of a proper coloring of $\mathcal{G}$. The known-graph proof transmits $\mathcal{G}$ itself, yet keeps the proper coloring confidential as default.

Proof of representation $\mathsf{P} \rightarrow \mathsf{I}$: The prover interacts with an CL-Signature issuer, proving representation and wellformedness of the commitment C in a known-graph proof, disclosing information to satisfy the verification requirements of the issuer. As $\tau(x)$ is invertible, this proof of representation of G and the proper coloring serves as proof of representation for x and $x \in \mathfrak{L}$.

Issuing $\mathsf{I} \rightarrow \mathsf{P}$: Upon acceptance of the proof, the issuer signs the committed graph $\mathcal{G}$ in a CL-Signature σ. Given the invertibility of τ, this signature holds for x as well. $sigma$ is a CL-Signature on $\tau(x)$ and the proper coloring of $\tau(x)$ iff $x \in \mathfrak{L}$.

Proof of possession $\mathsf{P} \rightarrow \mathsf{V}$: The prover interacts with the verifier to proof knowledge of the CL-Signature σ on a proper coloring on $\mathcal{G}$ and thereby shows graph 3-colorability of $\tau(x)$, which holds iff $x \in \mathfrak{L}$. Thereby, the proof of possession of σ translates to a proof of possession of the statement $x \in \mathfrak{L}$. The proof is zero-knowledge if executed with small challenges over multiple rounds. □