A q-SDH-based Graph Signature Scheme on Full-Domain Messages with Efficient Protocols

Syh-Yuan Tan and Thomas Groß

Newcastle University, Newcastle upon Tyne, United Kingdom
{syh-yuan.tan, thomas.gross}@newcastle.ac.uk

Abstract. A graph signature scheme is a digital signature scheme that allows a recipient to obtain a signature on a graph and subsequently prove properties thereof in zero-knowledge proofs of knowledge. In this paper, we present an efficient and provably secure graph signature scheme in the standard model with tight reduction. Based on the MoniPoly ABC, the MoniPoly graph signature scheme can provide show proofs on logical statements such as the existence of vertices, graph connectivity or isolation.

1 Introduction

A graph signature scheme is digital signature scheme that operates on a message space of graphs $\mathcal{G}$ and offers efficient proof protocols to assert graph properties, such as connectivity or isolation, in Zero-Knowledge Proofs of Knowledge (ZKPoK) while maintaining confidentiality of further information on the the structure or labeling of the graph itself. Given the versatility of graphs as a data structure, graph signatures are an interesting public-key cryptography primitive for a range of applications, including confidentiality-preserving security assurance, that is, the certification and attestation of topologies of systems of computer systems, first realized as part of cloud security and privacy assurance efforts [13]. Further applications on, for example, social network graphs, state machines, or provenance graphs have been proposed.

While ZKPoK on graphs have been instrumental in the first zero-knowledge proof constructions [3, 12] and transitive signature schemes operating on graph edges have been proposed earlier [19, 2], the first graph signature scheme with a framework of efficient proof protocols was proposed by Groß [15, 14]. This first construction was founded on the SRSA-based Camenisch-Lysyanskaya (CL) signature scheme [8], provably secure in the standard model. The CL signature can be extended to sign multiple messages without increasing the signature size and has been the core engine for attribute-based anonymous credential (ABC) systems, such as IBM's Identity Mixer [16]. Camenisch and Groß (CG) established a prime encoding [6, 7] to enhance the efficiency of the CL-RSA signature scheme by rendering the messages as prime exponents. Exploiting the coprimality and non-divisibility of the prime numbers, they showed that the encoded signature can be used to construct a highly efficient ABC system to provide efficient zero-knowledge proofs for AND, OR, and NOT/NAND logical statements.

Groß [14] CG-ABC system into a graph signature scheme which can sign the vertices and edges of a graph in the prime encoded form, while rendering the graph's algebraic structure accessible to subsequent ZKPoK. Apart from providing zero-knowledge proofs on a grammar of graph predicates [15], the graph signature also supports joint graph issuing protocol that allows the issuer to combine a graph of its choice with recipient's hidden graph during the signing protocol. This allows a recipient to source different graphs from multiple signers, a feature enabling privacy-preserving bootstrapping of signatures on complex graphs. Furthermore, Groß [14] proved the graph signature scheme's capability to encode and efficiently prove graph 3-colorability implying expressivity in terms of arbitrary statements from NP-languages. Due to these useful features, the graph signature was adopted in a real-world application [13]. While these are useful features, Groß' original construction of the graph signature scheme retains shortcomings that have yet to be addressed: (i) the message space is restricted to prime numbers, (ii) the encoding needs to be pre-certified, entailing to large public keys, (iii) the scheme lacks an overall rigorous security model, relying on properties inherited from the underlying CL- and CG- schemes [8, 7]. In terms of research gap, the existing SRSA graph signature construction suffers from complex show proofs, where the underlying quadratic-residue special RSA group setup leads to considerable computation and message complexity.

Our Contribution. We propose the first graph signature scheme with efficient protocols based on an elliptic-curve group setup. To that end, we establish a new variant of MoniPoly set commitment scheme [23, 24] that achieves the same efficiency and security as that of the original scheme, while being conducive to encoding graph data structures and proving of their well-formedness. Based on MoniPoly ABC system [23, 24], we then construct a new graph signature scheme which is provably secure in the standard model with a tight security reduction under the q-SDH assumption. In contrast to Groß' SRSA-based construction [15, 14], our novel graph signature scheme (i) enables graph encoding on arbitrary strings, not just prime numbers, (ii) supports the same expressive range of predicates with shorter proofs, (iii) features rigorous security analysis with respect to general security requirements.

Organization. The paper is organized as follows. Section 2 introduces the related work to our solution. Section 3 lists the preliminaries for our proposed graph signature scheme, followed by Section 4 which explains the MoniPoly encoding, the SDH counterpart for RSA prime encoding. We present the proposed graph signature scheme in Section 6.4 and the show the arguments for the graph properties in Section 7.

2 Related Works

Recently, Tan and Groß proposed an efficient ABC system with expressive show proofs, called MoniPoly [23, 24]. By design, MoniPoly ABC bears a number of similarities to the Camenisch-Groß-ABC [6, 7]. For instance, MoniPoly ABC is built on CL-SDH signature scheme [9, 1, 22], the pairing counterpart of CL-RSA signature scheme [8]. Moreover, MoniPoly ABC features an encoding function that converts an attributes into non-dividable values in $\mathbb{Z}_p^*$.

By extending the MoniPoly ABC system's attribute space to an appropriate graph encoding, it can yield a graph signature scheme, in principle. The main technical difficulty in achieving this is to establish the graph data structure in the credential. Groß' graph signature scheme [14] achieves this ability by creating an unambiguous encoding for a graph data set. The *graph prime encoding* then yields the capability to produce a graph well-formedness proof to assure the signed hidden graph is correctly encoded. Unfortunately, the original MoniPoly ABC cannot achieve the graph well-formedness in the same way, because the opening value and the graph data set share the same domain. For instance, a dishonest user can cheat by encoding two vertices i, j and their edge (i, j) as:

$$C = a_i^{(x'+o_i)(x'+i)} a_j^{(x'+o_j)(x'+i)} a_{(i,j)}^{(x'+o_{(i,j)})(x'+i)(x'+j)} \mod p$$

in a MoniPoly commitment where the opening values $o_i, o_j, o_{(i,j)}$ are chosen from the vertex universe. A naive solution might ask the user to compute a NOT proof with respect to the vertex identifier and labels universes for each opening value during every show proofs. This naive approach, however, would create a forbidding overhead and yield an impractical graph signature scheme.

Another related research area is the authenticated data structure [20, 10, 25] (ADS) which allows a data owner to outsource computations to a server, requiring the operations to be verifiable. Since ADS can be viewed as a database, it has been adopted to realize verifiable computation of database queries, particularly in the graph and relational databases [26, 18, 27]. For instance, the graph database constructed by Mandal et al. [18] supports query for meta-data, similar to proving the relationships among vertices in a graph signature scheme. However, the application scenario of databases from ADS require the data owners to disclose their data to the database server. Thereby, they enable the database server to answer the clients' queries on behalf of the data owners.

The scenario and security requirements for a graph signature scheme, however, are different: It requires information about the graph beyond the predicates proven to stay confidential. The owner of a graph signature, hence, responds to a verifier's queries on his own, using the signature of his committed graph as secret input. [T1]

3 Preliminaries

Definition 1. *Discrete Logarithm Assumption (DLOG). An algorithm $\mathcal{C}$ is said to $(t_{\mathsf{dlog}}, \varepsilon_{\mathsf{dlog}})$-break the DLOG assumption if $\mathcal{C}$ runs in time at most t_{dlog} and furthermore:*

$$\Pr[x \in \mathbb{Z}_p : \mathcal{C}(g, g^x) = x] \geq \varepsilon_{\mathsf{dlog}}$$

[T1] **TGr:** More research on authenticated data structures cf. ERC grant *for a negligible probability* $\varepsilon_{\mathsf{dlog}}$*. We say that the DLOG assumption is* $(t_{\mathsf{dlog}}, \varepsilon_{\mathsf{dlog}})$*-secure if no algorithm* $(t_{\mathsf{dlog}}, \varepsilon_{\mathsf{dlog}})$*-solves the DLOG problem.*

Definition 2. *co-Discrete Logarithm Assumption (co-DLOG) [11]. An algorithm* $\mathcal{C}$ *is said to* $(t_{\mathsf{codlog}}, \varepsilon_{\mathsf{codlog}})$*-break the co-DLOG assumption if* $\mathcal{C}$ *runs in time at most* t_{codlog} *and furthermore:*

$$\Pr[x \in \mathbb{Z}_p : \mathcal{C}(g_1, g_1^x \in \mathbb{G}_1, g_2, g_2^x \in \mathbb{G}_2) = x] \geq \varepsilon_{\mathsf{dlog}}$$

for a negligible probability $\varepsilon_{\mathsf{codlog}}$*. We say that the co-DLOG assumption is* $(t_{\mathsf{codlog}}, \varepsilon_{\mathsf{codlog}})$*-secure if no algorithm* $(t_{\mathsf{codlog}}, \varepsilon_{\mathsf{codlog}})$*-solves the co-DLOG problem.*

Definition 3. $q-$*Strong Diffie-Hellman Assumption (SDH) [22]. An algorithm* $\mathcal{C}$ *is said to* $(t_{\mathsf{sdh}}, \varepsilon_{\mathsf{sdh}})$*-break the SDH assumption if* $\mathcal{C}$ *runs in time at most* t_{sdh} *and furthermore:*

$$\Pr[x \in \mathbb{Z}_p, c \in \mathbb{Z}_p \setminus \{-x\}] : \mathcal{C}(g_1, g_1^x, \ldots, g_1^{x^q}, g_2, g_2^x) = (g_1^{\frac{1}{x+c}}, c)] \geq \varepsilon_{\mathsf{sdh}}$$

for a negligible probability $\varepsilon_{\mathsf{sdh}}$*. We say that the SDH assumption is* $(t_{\mathsf{sdh}}, \varepsilon_{\mathsf{sdh}})$*-secure if no algorithm* $(t_{\mathsf{sdh}}, \varepsilon_{\mathsf{sdh}})$*-solves the SDH problem.*

Definition 4. $q-$*co-Strong Diffie-Hellman Assumption (co-SDH) [11]. An algorithm* $\mathcal{C}$ *is said to* $(t_{\mathsf{cosdh}}, \varepsilon_{\mathsf{cosdh}})$*-break the co-SDH assumption if* $\mathcal{C}$ *runs in time at most* t_{cosdh} *and furthermore:*

$$\Pr[x \in \mathbb{Z}_p, c \in \mathbb{Z}_p \setminus \{-x\}] : \mathcal{C}(g_1, g_1^x, \ldots, g_1^{x^q}, g_2, g_2^x, \ldots, g_2^{x^q}) = (g^{\frac{1}{x+c}}, c)] \geq \varepsilon_{\mathsf{cosdh}}$$

for a negligible probability $\varepsilon_{\mathsf{cosdh}}$*. We say that the co-SDH assumption is* $(t_{\mathsf{cosdh}}, \varepsilon_{\mathsf{cosdh}})$*-secure if no algorithm* $(t_{\mathsf{cosdh}}, \varepsilon_{\mathsf{cosdh}})$*-solves the co-SDH problem.*

3.1 Pedersen Commitment

Pedersen commitment scheme [21] is perfectly hiding and computationally binding under the discrete logarithm assumption. The public parameters $pk_{PC} = (a, b \in \mathbb{G})$ are based on a group $\mathbb{G}$ of order p in which the discrete logarithm assumption holds. In order to commit to a message m, one computes

$$C = \mathsf{Commit}(pk_{PC}, m, r) = a^m b^r$$

where $r \in \mathbb{Z}_p^*$ is the randomly selected opening value.

3.2 MoniPoly Set Commitment

MoniPoly set commitment scheme [23, 24] is perfectly hiding and computationally binding under the co-SDH assumption. The public parameter is $pk_{MP} = (\{a_k = a^{x'^k}\}_{k=1}^n \in \mathbb{G}_1, \{X_k = g_2^{x'^k}\}_{k=1}^n \in \mathbb{G}_2, \mathsf{MPEncode})$ where $\mathsf{MPEncode} : \mathbb{Z}_p^n \to \mathbb{Z}_p^{n+1}$ converts a set of n messages into coefficients for a monic polynomial of degree $n+1$. In order to commit to a set of messages $A = \{m_1, \ldots, m_{n-1}\}$, one computes:

$$C = \mathsf{Commit}(pk_{MP}, A, o) = a_0^{(x'+o)\prod_{k=1}^{n-1}(x'+m_k)} = \prod_{k=0}^{n} a_k^{\mathsf{m}_k}$$

where $o \in \mathbb{Z}_p^*$ is the randomly selected opening value and $\{\mathsf{m}_j\} = \mathsf{MPEncode}(A \cup \{o\})$. MoniPoly commitment scheme supports zero-knowledge proofs on set operations:

1. intersection: proves the knowledge of an intersection set $I = A' \cap A$,
2. difference: proves the knowledge of an difference set $D = A' - A$,

which give rise to the show proofs in the MoniPoly ABC system [24].

3.3 Camenisch-Lysyanskaya SDH Signature Scheme

The CL-SDH signature scheme [9, 1, 22] is closely related to the BBS signature scheme [5]. Since the MoniPoly ABC system [23, 24] is the foundation of our proposed graph signature scheme, we recall the CL-SDH signature variant described by Tan and Groß where the signed messages are the input to an encoding function MPEncode as described above:

$\mathsf{KeyGen}(1^k, 1^n)$. Construct three cyclic groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ of order p based on an elliptic curve whose bilinear pairing is $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. Select random generators $a, b, c \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$ and two secret values $x, x' \in \mathbb{Z}_p^*$. Compute the values $X = g_2^x, \{a_i = a^{x'^i}, X_i = g_2^{x'^i}\}_{0 \le i \le n}$ to output the public key $pk = (\mathsf{e}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, b, c, \{a_i, X_i\}_{0 \le i \le n}, X)$ and the secret key $sk = (x, x')$.

$\mathsf{Sign}(pk, sk, \{m_1, \ldots, m_n\})$. On input $m_1, \ldots, m_n$, choose the random values $s, t \in \mathbb{Z}_p^*$ to compute:

$$v = \left(a_0^{\prod_{i=1}^{n}(x'+m_i)} b^s c\right)^{\frac{1}{x+t}}$$

and output the signature as $\sigma = (t, s, v)$.

$\mathsf{Verify}(pk, \sigma, \{m_1, \ldots, m_n\})$. Given $\sigma = (t, s, v)$, compute $\{\mathsf{m}_i\}_{0 \le i \le n} = \mathsf{MPEncode}(\{m_i\}_{1 \le i \le n})$. Output 1 if the following holds:

$$\mathsf{e}(v, X) = \mathsf{e}\left(\prod_{i=0}^{n} a_i^{\mathsf{m}_i} b^s c v^{-t}, g_2\right)$$

and output 0 otherwise.

Theorem 1. *[22, 24] The signature above is strongly existential unforgeable against chosen message attack in the standard model if the SDH problem is intractable.*

4 MoniPoly Graph Encoding

In this section, we offer a brief overview of the *graph prime encoding* proposed by Groß [14] before presenting our new encoding, namely, the *MoniPoly graph encoding*. It is conceptually similar to the former in nature and idea, yet supports graph encoding over $\mathbb{Z}_p^*$ instead of prime numbers.

4.1 Prime Graph Encoding

The *prime graph encoding* views every vertex and edge as a prime exponent. Camenisch and Groß [7] showed that a prime encoding in general can significantly speed up the show proofs by exploiting the co-primality and divisibility among messages. The main parameters for prime graph encoding are as follows:

- $\mathcal{V}$: Vertex universe
- $\mathcal{E} \subseteq (\mathcal{V} \times \mathcal{V})$: Edge universe
- $\mathcal{G} = (\mathcal{V}, \mathcal{E})$: Graph
- $\Xi_{\mathcal{V}}$: Vertex identifier universe
- $\Xi_{\mathcal{L}}$: Labels universe
- $f_{\mathcal{V}} : \mathcal{V} \to \mathcal{P}(\Xi_{\mathcal{L}})$: Labels of a given vertex
- $f_{\mathcal{E}} : \mathcal{E} \to \mathcal{P}(\Xi_{\mathcal{L}})$: Labels of a given edge
- $\chi_{\mathcal{V}}$: product of all vertex identifiers $\prod_{i \in \Xi_{\mathcal{V}}} i$
- $\chi_{\mathcal{L}}$: product of all labels $\prod_{i \in \Xi_{\mathcal{L}}} i$

where vertex identifiers and label are disjoint:

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

following the fundamental theorem of arithmetic.

In the following explanation, we focus on an unlabeled graph for clarity without loss of generality. In order to encode an unlabeled graph using the prime graph encoding, every vertex $i \in \mathcal{V}$ is mapped to a predefined prime number e_i. Next, let R_i be the base for the i-th vertex and $R_{(i,j)}$ be the base for the (i,j)-th edge, while $m_i = e_i \prod_{k \in f_{\mathcal{V}}(i)} e_k$ and $m_{(i,j)} = e_i e_j \prod_{k \in f_{\mathcal{E}}(i,j)} e_k$ denote the full encoding of vertices and edges, respectively. Assuming the signer knows the discrete logarithms of every base with respect to the public key element S, a graph can be represented by its vertex exponents $\bar{e}_i = \mathsf{dlog}_S(R_i) m_i$ and its edge exponents $\bar{e}_{(i,j)} = \mathsf{dlog}_S(R_{(i,j)}) m_{(i,j)}$.

4.2 Encoding Graphs Into the MoniPoly Set Commitment

In the MoniPoly set commitment scheme [23, 24], the messages are converted into a set of monic polynomial coefficients using a conversion function $\mathsf{MPEncode} : \mathbb{Z}_p^n \to \mathbb{Z}_p^{n+1}$ before being committed. Let us view the exponents in MoniPoly as encoded elements such that:

$$(x' + i) \mod p \Leftrightarrow e_i \mod \phi(N).$$

Then, $\mathsf{MPEncode}$ can be used as an encoding for graphs similar to the prime graph encoding.

Recalling from Section 3.2, let the public parameters be $\{a_{0_k} = a_{0_0}^{x'^k}, X_{0_k} = X_{0_0}^{x'^k}\}_{k=0}^n$. To encode a graph $\mathcal{G}$ of maximum size L, the additional public parameters

$$\{\{a_{i_k}, X_{i_k}\}_{i=1}^L\}_{k=0}^n$$

are generated. Subsequently, let $\{\mathsf{m}_{i_k}\} = \mathsf{MPEncode}(i, f_{\mathcal{V}}(i))$ and $\{\mathsf{m}_{(i,j)_j}\} = \mathsf{MPEncode}(i, j, f_{\mathcal{E}}(i,j))$, we have

$$\prod_{k=1}^{n_i} a_{i_k}^{\mathsf{m}_{i_k}} \Leftrightarrow R_i^{m_i} \text{ and } \prod_{k=1}^{n_{(i,j)}} a_{(i,j)_k}^{\mathsf{m}_{(i,j)_k}} \Leftrightarrow R_{(i,j)}^{m_{(i,j)}}$$

to represent a vertex and an edge, respectively. The parameters for prime graph encoding can then be adjusted accordingly to suit the MoniPoly encoding. Moreover, the MoniPoly encoding fulfills the requirement that vertex identifiers and labels be disjoint under the fundamental theorem of algebra. Building upon Groß' graph signature scheme [14], we define graph well-formedness in the context of the MoniPoly graph encoding as follows:

Definition 5. *We call a graph encoding well-formed if and only if:*

1. *The encoding only contains MoniPoly encoding representatives* $(x' + i) \in \Xi_{\mathcal{V}} \cup \Xi_{\mathcal{L}}$ *in the exponents of the base* a_i*.*
2. *A base* a_i *contains either exactly one vertex identifier* $(x'+i) \in \Xi_{\mathcal{V}}$*, pair-wise different from other vertex identifiers and zero or more label representatives* $(x' + k) \in \Xi_{\mathcal{L}}$*, or;*
3. *contains exactly two vertex identifiers* $(x' + i), (x' + j) \in \Xi_{\mathcal{V}}$ *and zero or more label representatives* $(x' + k) \in \Xi_{\mathcal{L}}$*.*

Considering the concepts we have introduced so far, we are not yet in the position the MoniPoly encoding to construct a SDH-based graph signature scheme in a way similar to Groß' SRSA-based graph signature scheme [14, 15]. This is because when an encoded graph is committed as a MoniPoly commitment naively, it loses its graph well-formedness property due to the shared message and opening space. This is unlike the combination of prime encoding and Pedersen commitment proposed by Groß. In the following section, we show how to overcome this problem.

5 Graph Well-formedness from a MoniPoly Commitment

We propose a variant of MoniPoly set commitment scheme featuring an externalized random blinding which is as secure and as efficient as the original scheme [23, 24]. We show that a further extension of this commitment scheme can commit MoniPoly encoded graphs securely. We continue to prove that the extension supports graph well-formedness if it has binding property.

5.1 An Externally-Blinded MoniPoly Set Commitment Scheme

To illustrate why a naive use of the original MoniPoly set commitment scheme fails desired security properties, we first consider the case in which a graph $\mathcal{G}$ has only one vertex $V = \{i, f_{\mathcal{V}}(i)\}$. The element a_{i_0} then holds the MoniPoly encoded graph as exponents. It can be viewed as an insecure MoniPoly commitment $C' = \prod_{k=0}^{n} a_{i_k}^{\mathsf{m}_k}$. Specifically, the opening value for C' is one of the committed graph elements: there exist finitely many of those and they are not random. Therefore, C' is not perfect hiding, though the computational binding property remains intact under the SDH assumption. Adding the opening value o breaks the graph well-formedness as $(x' + o)$ is not part of the graph encoding.

This problem can be resolved by selecting a random blinding $o \in \mathbb{Z}_p^*$ to compute the commitment as $C = C'^o$. One can thereby consider $C = C'^{x'+o'}$ as a MoniPoly commitment where $o = (x' + o') \mod p$ contains unknown opening value $o' \in \mathbb{Z}_p^*$. Therefore, computing $C = \mathsf{Commit}(pk, \mathcal{G}, o)$ is equivalent to computing $C = \mathsf{Commit}(pk, \mathcal{G}, o - x')$. Since finding x' yields an intractable DLOG problem and finding two different opening values that produce the same C breaks the co-SDH assumption, the externally-blinded MoniPoly variant is as secure as the original scheme.

We describe our proposed MoniPoly set commitment variant as follows:

$\mathsf{Setup}(1^k)$. Same as that in Section 3.2.

$\mathsf{Commit}(pk, A, o)$. Taking as input a message set $A = \{m_1, \ldots, m_n\} \in \mathbb{Z}_p^*$ and the random opening value $o \in \mathbb{Z}_p^*$, output the commitment as

$$C = \left(a_0^{\prod_{k=1}^{n}(x'+m_k)} \right)^o = \left(\prod_{k=0}^{n} a_k^{\mathsf{m}_k} \right)^o$$

where $\{\mathsf{m}_k\} = \mathsf{MPEncode}(A)$.

$\mathsf{Open}(pk, C, A, o)$. Return 1 if $C = \prod_{k=0}^{n} \left(a_j^{\mathsf{m}_k}\right)^o$ holds where $\{\mathsf{m}_k\} = \mathsf{MPEncode}(A)$ and return 0 otherwise.

$\mathsf{OpenIntersection}(pk, C, A, o, (A', l))$. If $|A' \cap A| \geq l$ holds, return an intersection set $I = A' \cap A$ of length l and a witness such that:

$$W = \left(a_0^{\prod_{m_k \in (A-I)}(x'+m_k)} \right)^o$$
$$= \left(\prod_{k=0}^{n} a_j^{\mathsf{w}_j} \right)^o$$

where $\{\mathsf{w}_k\} = \mathsf{MPEncode}(A - I)$. Otherwise, return a null value $\perp$. The correctness can be verified as follows:

$$
\begin{aligned}
C &= W^{\prod_{m_k \in I}(x'+m_k)} \\
&= \left(a_0^{o \prod_{m_k \in (A-I)}(x'+m_k)} \right)^{\prod_{m_k \in I}(x'+m_k)} \\
&= \left(a_0^{\prod_{m_k \in A}(x'+m_k)} \right)^{o}.
\end{aligned}
$$

$\mathsf{VerifyIntersection}(pk, C, I, W, (A', l))$. Return 1 if

$$
\mathsf{e}\left(C \prod_{k=0}^{|A'|} a_k^{\mathsf{m}_{1,k}}, X_0 \right) = \mathsf{e}\left(W \prod_{k=0}^{|A'|-l} a_k^{\mathsf{m}_{2,k}}, \prod_{k=0}^{l} X_k^{\mathsf{i}_k} \right)
$$

holds and return 0 otherwise, where $\{\mathsf{i}_k\} = \mathsf{MPEncode}(I)$, $\{\mathsf{m}_{1,k}\} = \mathsf{MPEncode}(A')$ and $\{\mathsf{m}_{2,k}\} = \mathsf{MPEncode}(A' - I)$. The correctness is shown as follows:

$$
\begin{aligned}
&\mathsf{e}\left(C \prod_{k=0}^{|A'|} a_j^{\mathsf{m}_{1,k}}, X_0 \right) \\
&= \mathsf{e}\left(C, X_0\right) \mathsf{e}\left(\prod_{k=0}^{|A'|} a_k^{\mathsf{m}_{1,k}}, X_0 \right) \\
&= \mathsf{e}\left(a_0^{o \prod_{m_k \in A}(x'+m_k)}, X_0 \right) \mathsf{e}\left(a_0^{\prod_{m_k \in A'}(x'+m_k)}, X_0 \right) \\
&= \mathsf{e}\left(a_0^{o \prod_{m_k \in (A-I)}(x'+m_k)}, X_0^{\prod_{m_k \in I}(x'+m_k)} \right) \mathsf{e}\left(a_0^{\prod_{m_k \in (A'-I)}(x'+m_k)}, X_0^{\prod_{m_k \in I}(x'+m_k)} \right) \\
&= \mathsf{e}\left(W, \prod_{k=0}^{l} X_k^{\mathsf{i}_k} \right) \mathsf{e}\left(\prod_{k=0}^{|A'|-l} a_k^{\mathsf{m}_{2,k}}, \prod_{k=0}^{l} X_k^{\mathsf{i}_k} \right) \\
&= \mathsf{e}\left(W \prod_{k=0}^{|A'|-l} a_k^{\mathsf{m}_{2,j}}, \prod_{k=0}^{l} X_k^{\mathsf{i}_j} \right)
\end{aligned}
$$

$\mathsf{OpenDifference}(pk, C, A, o, (A', \bar{l}))$. If $|A' \cap A| \geq \bar{l}$ holds, return a difference set $D = A' - A$ of length $\bar{l}$ and the witness $\left(W = \prod_{k=0}^{n-\bar{l}} a_k^{\mathsf{w}_k}, \{\mathsf{r}_k\}_{k=0}^{\bar{l}-1} \right)$. The values $(\{\mathsf{w}_k\}, \{\mathsf{r}_k\}) = \mathsf{MPEncode}(A)/\mathsf{MPEncode}(D)$ are computed using expanded synthetic division such that $\{\mathsf{w}_k\}$ are the coefficients of quotient $q(x')$ and $\{\mathsf{r}_k\}$

are the coefficients of remainder $r(x')$. Specifically, let the polynomial divisor be $d(x') = \sum_k^{\bar{l}} \mathsf{d}_k x'^k$ where $\{\mathsf{d}_k\} = \mathsf{MPEncode}(D)$, the monic polynomial $f(x')$ in the commitment $C = a_0^{f(x')}$ can be rewritten as $f(x') = d(x')q(x') + r(x')$. Note that $\prod_{k=0}^{\bar{l}-1} a_k^{\mathsf{r}_k} \neq 1_{\mathbb{G}_1}$ whenever $d(x')$ cannot divide $f(x')$, i.e., the sets A and D are disjoint. The correctness can be verified from the following:

$$\begin{aligned} C &= \left(a_0^{\prod_{m_k \in A}(x'+m_k)}\right)^o \\ &= \left(a_0^{q(x')d(x')+r(x')}\right)^o \\ &= \left(a_0^{o\sum_{k=0}^{n-\bar{l}} \mathsf{w}_k x'^k}\right)^{d(x')} \left(\prod_{k=0}^{\bar{l}-1} a_k^{\mathsf{r}_k}\right)^o \\ &= W^{d(x')} \left(\prod_{k=0}^{\bar{l}-1} a_k^{\mathsf{r}_k}\right)^o . \end{aligned}$$

$\mathsf{VerifyDifference}(pk, C, D, (W, \{\mathsf{r}_k\}_{k=0}^{\bar{l}-1}), (A', \bar{l}))$. Return 1, if the following holds:

$$\mathsf{e}\left(C\left(\prod_{k=0}^{\bar{l}-1} a_k^{-\mathsf{r}_k}\right)^o \prod_{k=0}^{|A'|} a_k^{\mathsf{m}_{1,k}}, X_0\right) = \mathsf{e}\left(W \prod_{k=0}^{|A'|-\bar{l}} a_k^{\mathsf{m}_{2,k}}, \prod_{k=0}^{\bar{l}} X_k^{\mathsf{d}_k}\right), \prod_{k=0}^{\bar{l}-1} a_k^{\mathsf{r}_k} \neq 1_{\mathbb{G}_1}$$

and return 0 otherwise, where $\{\mathsf{d}_k\} = \mathsf{MPEncode}(D)$, $\{\mathsf{m}_{1,k}\} = \mathsf{MPEncode}(A')$ and $\{\mathsf{m}_{2,k}\} = \mathsf{MPEncode}(A' - D)$. The correctness is as follows:

$$\begin{aligned}
&\mathsf{e}\left(C\left(\prod_{k=0}^{\bar{l}-1} a_k^{-\mathsf{r}_k}\right)^o \prod_{k=0}^{|A'|} a_k^{\mathsf{m}_{1,k}}, X_0\right) \\
&= \mathsf{e}\left(C\left(\prod_{k=0}^{\bar{l}-1} a_k^{-\mathsf{r}_k}\right)^o, X_0\right) \mathsf{e}\left(\prod_{k=0}^{|A'|} a_k^{\mathsf{m}_{1,k}}, X_0\right) \\
&= \mathsf{e}\left(a_0^{od(x')q(x')+or(x')} a_0^{-or(x')}, X_0\right) \mathsf{e}\left(a_0^{\prod_{m_k \in A'} (x'+m_k)}, X_0\right) \\
&= \mathsf{e}\left(a_0^{od(x')q(x')}, X_0\right) \mathsf{e}\left(a_0^{\prod_{m_k \in (A'-D)} (x'+m_k)}, X_0^{\prod_{m_k \in D} (x'+m_k)}\right) \\
&= \mathsf{e}\left(a_0^{o\sum_{k=0}^{n-\bar{l}} \mathsf{w}_k x'^k}, X_0^{d(x')}\right) \mathsf{e}\left(\prod_{k=0}^{|A'|-\bar{l}} a_k^{\mathsf{m}_{2,k}}, X_0^{d(x')}\right) \\
&= \mathsf{e}\left(W \prod_{k=0}^{|A'|-\bar{l}} a_k^{\mathsf{m}_{2,k}}, \prod_{k=0}^{\bar{l}} X_k^{\mathsf{d}_k}\right).
\end{aligned}$$

For the completeness of security analysis, we support the security for the proposed scheme with Theorem 2.

Theorem 2. *The externally-blinded MoniPoly set commitment scheme is perfectly hiding and computational binding under the q-SDH assumption.*

Proof. (Sketch.) The externally-blinded commitment scheme described here is similar to the original MoniPoly commitment scheme [24] such that one can prove its security by adapting the latter's security proofs. Specifically, we can view the externally-blinded variant as a MoniPoly commitment with a randomized base in the proof such that:

$$C = \left(\prod_{k=0}^{n} a_k^{\mathsf{m}_k}\right)^o = \prod_{k=0}^{n} (a_k^o)^{\mathsf{m}_k}.$$

As one can easily see, the new scheme inherits the perfectly hiding property from original MoniPoly scheme. On the other hand, if an adversary outputs $(A, o') \neq (A^*, o'')$ such that

1. $\mathsf{Open}(pk, \mathsf{Commit}(pk, A, o'), A, o') =$
   $\mathsf{Open}(pk, , \mathsf{Commit}(pk, A^*, o''), A^*, o'')$,
2. $\mathsf{OpenIntersection}(pk, \mathsf{Commit}(pk, A, o'), A, o', (A', l)) =$
   $\mathsf{OpenIntersection}(pk, \mathsf{Commit}(pk, A^*, o''), A^*, o'', (A', l))$,

3. $\mathsf{OpenDifference}(pk, \mathsf{Commit}(pk, A, o'), A, o', (A', \bar{l})) =$
   $\mathsf{OpenDifference}(pk, \mathsf{Commit}(pk, A^*, o''), A^*, o'', (A', \bar{l}))$,

a q-SDH solution can be extracted from the fact that two different sets $(A, o'), (A^*, o'')$ yield the same commitment C. The extraction is the same as in the original MoniPoly scheme but uses the randomized bases $a_k^{o'}, a_k^{o''}$. Note that we can always derandomize the bases to gain their original form as in the given SDH instance because o' and o'' are known. □

5.2 An Extended Externally-Blinded MoniPoly Set Commitment

The proposed variant of MoniPoly set commitment can be used to commit a Monipoly encoded graph, if we add parameters $\{\{a_{i_k}, X_{i_k}\}_{k=0}^n\}_{i=1}^L$ specific for the MoniPoly encoding to the commitment parameters. While the commitment opening algorithms can be amended accordingly, the security proof requires a considerate modification. In particular, while the perfectly hiding property follows trivially, the computational binding property is now based on the hardness of the co-DLOG problem.

Theorem 3. *The extended externally-blinded MoniPoly set commitment is binding if the co-DLOG problem is hard.*

Proof. Given a co-DLOG instance $(g_1, h_1 = g_1^x \in \mathbb{G}_1, g_2, h_2 = g_2^x \in \mathbb{G}_2)$, we construct a challenger $\mathcal{C}$ that runs the adversary $\mathcal{A}$ of extended MoniPoly set commitment scheme as a sub-routine to find the solution x. $\mathcal{C}$ sets $\{a_{0_k} = g_1^{x'^k}, X_{0_k} = g_2^{x'^k}\}_{k=0}^n$ and $\{\{a_{i_k} = h_1^{b_i x'^k}, X_{i_k} = h_2^{b_i x'^k}\}_{k=0}^n\}_{i=1}^L$ for randomly chosen $b_i, x' \in \mathbb{Z}_p^*$. $\mathcal{C}$ publishes $\{\{a_{i_k}, X_{i_k}\}_{k=0}^n\}_{i=0}^L$ as the public parameters.

Without loss of generality, we assume a graph is MoniPoly encoded using the bases a_{i_k} in the sequence $i = 1, 2, \ldots, L$. If an adversary can output an extended MoniPoly set commitment C for two different graph data sets $(\mathcal{G}, \mathcal{G}^*)$ such that $|\mathcal{G} \cap \mathcal{G}^*| \geq 2$:

$$\mathcal{G} = \{V \cup E\} = \{(i, f_{\mathcal{V}}(i), o_i') \in V, (i, j, f_{\mathcal{E}}(i,j), o_{(i,j)}') \in E\},$$
$$\mathcal{G}^* = \{V^* \cup E^*\} = \{(i^*, f_{\mathcal{V}}(i^*), o_i'') \in V^*, (i^*, j^*, f_{\mathcal{E}}(i^*, j^*), o_{(i,j)}'') \in E^*\},$$

where $\{o_i', o_{(i,j)}', o_i'', o_{(i,j)}''\}$ are the opening values for vertices and edges, respectively, a co-DLOG solution can be extracted. This follows from:

$$C = \prod_{i \in V} \left(a_{i_0}^{(x'+i) \prod_{w \in f_{\mathcal{V}}(i)} (x'+w)} \right)^{o_i'} \prod_{i \in E} \left(a_{(i,j)_0}^{(x'+i)(x'+j) \prod_{w \in f_{\mathcal{E}}(i,j)} (x'+w)} \right)^{o_{(i,j)'}}$$
$$= \prod_{i^* \in V^*} \left(a_{i_0}^{(x'+i^*) \prod_{w^* \in f_{\mathcal{V}}(i^*)} (x'+w^*)} \right)^{o_i''} \prod_{i^* \in E} \left(a_{(i,j)_0}^{(x'+i^*)(x'+j^*) \prod_{w^* \in f_{\mathcal{V}}(i^*, j^*)} (x'+w^*)} \right)^{o_{(i^*, j^*)}},$$

giving the following equations:

$$g_1^{(x'+i)\prod_{w\in f_{\mathcal{V}}(i)}(x'+w)} h^{\sum_2^{|V|}(x'+i)\prod_{w\in f_{\mathcal{V}}(i)}(x'+w)+\sum_1^{|E|}(x'+i)(x'+j)\prod_{w\in f_{\mathcal{E}}(i,j)}(x'+w)}$$
$$= g_1^u h_1^v$$
$$= g_1^{(x'+i^*)\prod_{w^*\in f_{\mathcal{V}}(i^*)}(x'+w^*)} h^{\sum_2^{|V^*|}(x'+i^*)\prod_{w^*\in f_{\mathcal{V}}(i^*)}(x'+w^*)+\sum_1^{|E^*|}(x'+i^*)(x'+j^*)\prod_{w^*\in f_{\mathcal{E}}(i^*,j^*)}(x'+w^*)}$$
$$= g_1^{u^*} h_1^{v^*}$$

Therein, the Challenger $\mathcal{C}$ can compute:

$$x = \frac{u - u^*}{v^* - v} \mod p$$

to solve the co-DLOG problem. The argument on the OpenIntersection and OpenDifference algorithms is similar to that in the proof of Theorem 2. □

Thereafter, whenever we mention a MoniPoly commitment, we mean the extended externally-blinded MoniPoly set commitment scheme.

5.3 Graph Composition

The MoniPoly commitment $C = \prod_{i\in V} C_i \prod_{(i,j)\in E} C_{(i,j)}$ on a graph $\mathcal{G}$ is the product of vertex commitments C_i and edge commitments $C_{(i,j)}$, where:

$$C_i = \left(\prod_{k=0}^{n_i} a_{i_k}^{\mathsf{m}_{i_k}}\right)^{o_i}, C_{(i,j)} = \left(\prod_{k=0}^{n_{(i,j)}} a_{(i,j)_k}^{\mathsf{m}_{(i,j)_k}}\right)^{o_{(i,j)}}$$

To prove the composition of the graph commitment C, one can run proof of knowledge protocols for the MoniPoly Open algorithms on every $C_i \in V$ and $C_{(i,j)} \in E$. These proof of knowledge protocols can be combined into a graph composition statement $\mathsf{graph}(C)$:

$$PK\Bigg\{(\forall i \in V : \varepsilon_{i_0}, \varepsilon_{i_1}, \forall (i,j) \in E : \varepsilon_{(i,j)_0}, \varepsilon_{(i,j)_1}) :$$
$$\mathsf{e}(C, X_{0_0}) = \prod_{i\in V} \mathsf{e}\left(W_i', \prod_{k=0}^{1} X_{0_k}^{\varepsilon_{i_k}}\right) \prod_{(i,j)\in E} \mathsf{e}\left(W_{(i,j)}', \prod_{k=0}^{1} X_{0_k}^{\varepsilon_{(i,j)_k}}\right)\Bigg\}.$$

The correctness can be verified from the equation below:

$$\mathsf{e}(C, X_{0_0}) = \prod_{i\in V'} \mathsf{e}\left(a_{i_0}^{r_i^{-1} o_i \prod_{w\in f_{\mathcal{V}}(i)}(x'+w)}, X_{0_0}^{r_i(x'+i)}\right) \cdot$$
$$\prod_{(i,j)\in E'} \mathsf{e}\left(a_{(i,j)_0}^{r_{(i,j)}^{-1} o_{(i,j)}(x'+j)\prod_{w\in f_{\mathcal{E}}(i,j)}(x'+w)}, X_{0_0}^{r_{(i,j)}(x'+i)}\right)$$

We can replace the exponent i for X_{0_0} with a random attribute w without affecting the randomness in the proof as claimed in Lemma 1. The graph composition proof also appears in the $\mathsf{possession}(\sigma, \mathcal{G})$ statement to prove the validity of a graph signature.

Lemma 1. *The randomization of C in* graph *is perfectly hiding.*

Proof. The MoniPoly opening values $o_i, o_{(i,j)}$ turns the $\mathbb{G}_T$ elements:

$$\prod_{i \in V'} \mathsf{e}\left(a_{i_0}^{r_i^{-1} \prod_{w \in f_{\mathcal{V}}(i)}(x'+w)}, X_{0_0}^{r_i(x'+i)}\right)^{o_i} \prod_{(i,j) \in E'} \mathsf{e}\left(a_{(i,j)_0}^{r_{(i,j)}^{-1}(x'+j) \prod_{w \in f_{\mathcal{E}}(i,j)}(x'+w)}, X_{0_0}^{r_{(i,j)}(x'+i)}\right)^{o_{(i,j)}}$$

into a Pedersen set commitment (Lemma 8) which is perfectly hiding. When C consists of one vertex or one edge only, the single $\mathbb{G}_T$ element is a MoniPoly commitment and it is perfectly hiding. □

5.4 Bootstrapping of MoniPoly-Encoded Graphs

The graph statement proves that C can be decomposed into MoniPoly commitments $C_i, C_{(i,j)}$ only. In order to prove that C is a commitment for a correctly encoded graph, we need to further show that the n vertex commitments $C_i \in V$ and m edge commitments $C_{(i,j)} \in E$ is correctly encoded. Firstly, instead of selecting a random exponent for the X_{0_k} elements in graph, we set all encoded labels as exponents for the witness $W'_i, W'_{(i,j)}$ and the encoded identifiers as the exponents for the X_{0_k} elements. Secondly, assuming the universe for identifiers ($\Xi_{\mathcal{V}}$) and labels ($\Xi_{\mathcal{L}}$) are publicly known, we need to prove the existence of every commitment's identifier and labels in $\Xi_{\mathcal{V}}$ and $\Xi_{\mathcal{L}}$, respectively. Computing the $n + m$ MoniPoly commitment for $\Xi_{\mathcal{V}}$ and $\Xi_{\mathcal{L}}$ is required because every commitment C_i (resp. $C_{(i,j)}$) is computed on a different base a_{i_k} (resp. $a_{(i,j)_k}$).

This complexity overhead can be avoided by an additional step of bootstrapping [14] that switches the bases $\{a_i\}_{i \in V}, \{a_{(i,j)}\}_{(i,j) \in E}$ to a_0 such that:

$$C_i = \left(\prod_{k=0}^{n_i} a_{0_k}^{\mathsf{m}_{i_k}}\right)^{o_i}, C_{(i,j)} = \left(\prod_{k=0}^{n_{(i,j)}} a_{0_k}^{\mathsf{m}_{(i,j)_k}}\right)^{o_{(i,j)}}.$$

The two MoniPoly commitments for $\Xi_{\mathcal{V}}$ and $\Xi_{\mathcal{L}}$ are now computed only once on the base a_{0_k} and referenced by all set membership proofs. Moreover, since $\{C_i, C_{(i,j)}\}$ are computed on the same base, we can prove the encoding correctness for all the vertices and edges by using only two set membership proofs plus two cumulative product proofs, respectively. We first explain the simpler

$\mathsf{bootstrap}$ statement as follows:

$$PK\Bigg\{(\forall i \in V : \varepsilon_{i_0}, \varepsilon_{i_1}, \forall (i,j) \in E : \varepsilon_{(i,j)_0}, \varepsilon_{(i,j)_1}, \varepsilon_{(i,j)_2}) :$$
$$\mathsf{e}\left(C, X_{0_0}\right) = \prod_{i \in V} \mathsf{e}\left(W'_i, \prod_{k=0}^{1} X_{0_k}^{\varepsilon_{i_k}}\right) \prod_{(i,j)\in E} \mathsf{e}\left(W'_{(i,j)}, \prod_{k=0}^{2} X_{0_k}^{\varepsilon_{(i,j)_k}}\right) \wedge$$
$$\mathsf{e}\left(\prod_{i\in V} C_i \prod_{(i,j)\in E} C_{(i,j)}, X_{0_0}\right) =$$
$$\prod_{i\in V} \mathsf{e}\left(W_i, \prod_{k=0}^{1} X_{0_k}^{\varepsilon_{i_k}}\right) \prod_{(i,j)\in E} \mathsf{e}\left(W_{(i,j)}, \prod_{k=0}^{2} X_{0_k}^{\varepsilon_{(i,j)_k}}\right) \wedge$$
$$\mathsf{e}\left(\prod_{i\in V} W'_i \prod_{(i,j)\in E} W'_{(i,j)}, X_{0_0}\right) = \prod_{i\in V} \mathsf{e}\left(W_i, X_{i_0}\right) \prod_{(i,j)\in E} \mathsf{e}\left(W_{(i,j)}, X_{(i,j)_0}\right)\Bigg\}$$

where $\varepsilon_{i,1} = r_i, \varepsilon_{(i,j),2} = r_{(i,j)}, W'_i = a_{i_0}^{o_i r_i^{-1} \prod_{w \in f_{\mathcal{V}}(i_l)} (x'+w)}$ and $W'_{(i,j)} = a_{(i,j)_0}^{o_{(i,j)} r_{(i,j)}^{-1} \prod_{w \in f_{\mathcal{E}}((i,j)_l)} (x'+w)}$ for randomly selected $o_i, r_i, o_{(i,j)}, r_{(i,j)} \in \mathbb{Z}_p^*$ and commitments $\{C, C_i, C_{(i,j)}, W'_i, W'_{(i,j)}, W_i, W_{(i,j)}\}$ are public inputs.

The first statement is the $\mathsf{graph}(C)$ statement while the second and third statements are the bootstrapping statement $\mathsf{bootstrap}(C)$. The MoniPoly commitments $\{C_i, C_{(i,j)}, W_i, W_{(i,j)}\}$ that appear in the bootstrapping will be used in $\mathsf{vertices}$ and edges which are described in the next sections. Note that the bootstrapping can be compressed to result in a more efficient proof of representation as follows:

$$PK\Bigg\{(\forall i \in V : \varepsilon_{i_0}, \varepsilon_{i_1}, \forall (i,j) \in E : \varepsilon_{(i,j)_0}, \varepsilon_{(i,j)_1}, \varepsilon_{(i,j)_2}) :$$
$$\mathsf{e}\left(C, X_{0_0}\right) = \prod_{i \in V} \mathsf{e}\left(W'_i, \prod_{k=0}^{1} X_{0_k}^{\varepsilon_{i_k}}\right) \prod_{(i,j)\in E} \mathsf{e}\left(W'_{(i,j)}, \prod_{k=0}^{2} X_{0_k}^{\varepsilon_{(i,j)_k}}\right) \wedge$$
$$\mathsf{e}\left(\prod_{i\in V} W'_i C_i \prod_{(i,j)\in E} W'_{(i,j)} C_{(i,j)}, X_{0_0}\right) =$$
$$\prod_{i\in V} \mathsf{e}\left(W_i, X_{i_0} \prod_{k=0}^{1} X_{0_k}^{\varepsilon_{i_k}}\right) \prod_{(i,j)\in E} \mathsf{e}\left(W_{(i,j)}, X_{(i,j)_0} \prod_{k=0}^{2} X_{0_k}^{\varepsilon_{(i,j)_k}}\right)\Bigg\}.$$

Lemma 2. *The randomization of C in* $\mathsf{bootstrap}$ *is perfectly hiding.*

Proof. The proof is the same as Lemma 1. □

5.5 Vertex Composition

The core idea in the $\mathsf{vertices}$ statement is to prove the computation correctness for the cumulative products of all vertex identifiers i and labels $f_{\mathcal{V}}(i)$, respectively, in a graph $\mathcal{G}$. Also, $\mathsf{vertices}$ proves the cumulative products are a subset of vertex identifier universe (e.g. $i \in \Xi_{\mathcal{V}}$) and label universe (e.g. $f_{\mathcal{V}}(i) \in \Xi_{\mathcal{L}}$), respectively. We note that the proof semantics of this proof are different from the original graph signature scheme [14] as the latter's vertex composition statement only decomposed the graph signature into singleton commitments on encodings of individual vertices.

Let $V = \bigcup\{V_1, \ldots, V_\ell\}$ be the vertex set in a graph $\mathcal{G}$, the $\mathsf{vertices}$ statement is described as the following protocol:

$$PK\{(\forall i \in V : \varepsilon_{i_0}, \varepsilon_{i_1}) : \mathsf{e}\left(\prod_{i\in V} C_i, X_{0_0}\right) = \prod_{l=1}^{\ell} \mathsf{e}\left(W_l, \prod_{k=0}^{1} X_{0_k}^{\varepsilon_{l_k}}\right) \wedge$$

$$\mathsf{e}\left(W_{\mathcal{V}_1}, X_{0_0}\right) = \mathsf{e}\left(a_{0_0}, \prod_{k=0}^{1} X_{0_k}^{\varepsilon_{1_k}}\right) \wedge \mathsf{e}\left(a_{0_0}, W_{\mathcal{L}_1}\right) = \mathsf{e}\left(W_1, X_{0_0}\right) \wedge$$

$$\mathsf{e}\left(W_{\mathcal{V}_2}, X_{0_0}\right) = \mathsf{e}\left(W_{\mathcal{V}_1}, \prod_{k=0}^{1} X_{0_k}^{\varepsilon_{2_k}}\right) \wedge \mathsf{e}\left(W_{\mathcal{V}_3}, X_{0_0}\right) = \mathsf{e}\left(W_{\mathcal{V}_2}, \prod_{k=0}^{1} X_{0_k}^{\varepsilon_{3_k}}\right) \wedge$$

$$\cdots \wedge \mathsf{e}\left(a_{0_0}, W_{\mathcal{V}_\ell}\right) = \mathsf{e}\left(W_{\mathcal{V}_{\ell-1}}, \prod_{k=0}^{1} X_{0_k}^{\varepsilon_{\ell_k}}\right) \wedge \mathsf{e}\left(\prod_{k=0}^{|\Xi_{\mathcal{V}}|} a_{0_k}^{\mathsf{m}_{\mathcal{V}_k}}, X_{0_0}\right) = \mathsf{e}\left(W_{\Xi_{\mathcal{V}}\setminus V}, W_{\mathcal{V}_\ell}\right) \wedge$$

$$\mathsf{e}\left(a_{0_0}, W_{\mathcal{L}_2}\right) = \mathsf{e}\left(W_2, W_{\mathcal{L}_1}\right) \wedge \mathsf{e}\left(a_{0_0}, W_{\mathcal{L}_3}\right) = \mathsf{e}\left(W_3, W_{\mathcal{L}_2}\right) \wedge$$

$$\cdots \wedge \mathsf{e}\left(a_{0_0}, W_{\mathcal{L}_\ell}\right) = \mathsf{e}\left(W_\ell, W_{\mathcal{L}_{\ell-1}}\right) \wedge \mathsf{e}\left(\prod_{k=0}^{|\Xi_{\mathcal{L}}|} a_{0_k}^{\mathsf{m}_{\mathcal{L}_k}}, X_{0_0}\right) = \mathsf{e}\left(W_{\Xi_{\mathcal{L}}\setminus f_{\mathcal{V}}(V)}, W_{\mathcal{L}_\ell}\right)$$

$$\}$$

where $\{\mathsf{m}_{\mathcal{V}k}\} = \mathsf{MPEncode}(\Xi_{\mathcal{V}}), \{\mathsf{m}_{\mathcal{L}_{\mathcal{V}k}}\} = \mathsf{MPEncode}(\Xi_{\mathcal{L}_{\mathcal{V}}}), W_l = \prod_{k=0}^{|f_{\mathcal{V}}(i_l)|} a_{0_k}^{\mathsf{m}_{l_k}}$, $\{\varepsilon_{l_0}, \varepsilon_{l_1}\} = r_l \times \mathsf{MPEncode}(\{i_l\})$ and $\{\mathsf{m}_{l_j}\} = o_l \cdot r_l^{-1} \times \mathsf{MPEncode}(f_{\mathcal{V}}(i_l))$ for randomly selected $o_l, r_l \in \mathbb{Z}_p^*$. The public inputs $(W_1, \ldots, W_\ell)$ are witnesses for the vertex labels, $(W_{\mathcal{V}_1}, \ldots, W_{\mathcal{V}_\ell}, W_{\Xi_{\mathcal{V}}\setminus V})$ are witnesses for the cumulative product of vertex identifiers while $(W_{\mathcal{L}_1}, \ldots, W_{\mathcal{L}_\ell}, W_{\Xi_{\mathcal{L}}\setminus f_{\mathcal{V}}(V)})$ are witnesses for the cumulative product of vertex labels. The first statement is the $\mathsf{bootstrap}$ statement while the correctness for the cumulative products can be verified from the following equationss:

$$\begin{aligned}
\mathsf{e}\left(W_{\mathcal{V}_l}, X_{0_0}\right) &= \mathsf{e}\left(a_{0_0}^{r_l(x'+i_l)\prod_{k=1}^{l-1} r_k(x'+i_k)}, X_{0_0}\right) \\
&= \mathsf{e}\left(a_{0_0}^{\prod_{k=1}^{l-1} r_k(x'+l_k)}, X_{0_0}^{r_l(x'+i_l)}\right) \\
&= \mathsf{e}\left(W_{\mathcal{V}_{l-1}}, \prod_{k=0}^{1} X_{0_k}^{\varepsilon_{l_k}}\right)
\end{aligned}$$

and,

$$\begin{aligned}
\mathsf{e}\left(a_{0_0}, W_{\mathcal{L}_l}\right) &= \mathsf{e}\left(a_{0_0}, X_{0_0}^{o_l \cdot r_l^{-1}\prod_{w\in f_{\mathcal{V}}(i_l)}(x'+w)\prod_{k=1}^{l-1} o_k\cdot r_k^{-1}\prod_{w\in f_{\mathcal{V}}(i_k)}(x'+w)}\right) \\
&= \mathsf{e}\left(a_{0_0}^{o_l\cdot r_l^{-1}\prod_{w\in f_{\mathcal{V}}(i_l)}(x'+w)}, X_{0_0}^{\prod_{k=1}^{l-1} o_k\cdot r_k^{-1}\prod_{w\in f_{\mathcal{V}}(i_k)}(x'+w)}\right) \\
&= \mathsf{e}\left(W_l, W_{\mathcal{L}_{l-1}}\right).
\end{aligned}$$

The proofs of cumulative product for the vertex identifiers and labels implicitly prove the pair-wise differences for every vertex. Let $W_{\mathcal{V}_0} = a_{0_0}, W_{\mathcal{L}_0} = X_{0_0}$, simplifying the pairing notations in the proof above gives:

$$\begin{aligned}
PK\Bigg\{ &(\forall i \in V : \varepsilon_{i_0}, \varepsilon_{i_1}) : \mathsf{e}\left(\prod_{i\in V} C_i, X_{0_0}\right) = \prod_{l=1}^{\ell} \mathsf{e}\left(W_l, \prod_{k=0}^{1} X_{0_k}^{\varepsilon_{l_k}}\right) \wedge \\
&\mathsf{e}\left(\prod_{k=0}^{|\Xi_{\mathcal{V}}|} a_{0_k}^{\mathsf{m}_{\mathcal{V}_k}} \prod_{l=1}^{\ell-1} W_{\mathcal{V}_l}, X_{0_0}\right) = \\
&\qquad\qquad \prod_{l=1}^{\ell} \mathsf{e}\left(W_{\mathcal{V}_{l-1}}, \prod_{k=0}^{1} X_{k}^{\varepsilon_{l_k}}\right) \mathsf{e}\left(a_{0_0}^{-1} W_{\Xi_{\mathcal{V}}\setminus V}, W_{\mathcal{V}_\ell}\right) \wedge \\
&\mathsf{e}\left(\prod_{k=0}^{|\Xi_{\mathcal{L}}|} a_{0_k}^{\mathsf{m}_{\mathcal{L}_k}}, X_{0_0}\right) \mathsf{e}\left(a_{0_0}, \prod_{l=1}^{\ell} W_{\mathcal{L}_l}\right) = \\
&\qquad\qquad \prod_{l=1}^{\ell} \mathsf{e}\left(W_l, W_{\mathcal{L}_{l-1}}\right) \mathsf{e}\left(W_{\Xi_{\mathcal{L}}\setminus f_{\mathcal{V}}(V)}, W_{\mathcal{L}_\ell}\right) \Bigg\}
\end{aligned}$$

to establish the $\mathsf{vertices}(V)$ statement[1] where C_i links to the graph signature. In the subsequent sections, we use $\varepsilon_i[C_i] \in \Xi_{\mathcal{V}}$ and $W_i[C_i] \subseteq \Xi_{\mathcal{L}}$ as the short form for the last two statements. They are read as ε_i encoded in commitment C_i is a member of $\Xi_{\mathcal{V}}$, and correspondingly, W_i encoded in commitment C_i is a witness for a member of $\Xi_{\mathcal{L}}$.

[1] The first statement can be combined with the second and third statements to save another $\ell + 2$ pairing operations but we do not present the proof as such for clarify purposes.

Lemma 3. *The randomization of* $(W_{\mathcal{V}_1}, \ldots, W_{\mathcal{V}_\ell}, W_{\Xi_\mathcal{V} \setminus V})$ *and* $(W_{\mathcal{L}_1}, \ldots, W_{\mathcal{L}_\ell}, W_{\Xi_\mathcal{L} \setminus f_\mathcal{V}(V)})$ *in* vertices *are perfectly hiding.*

Proof. The random values r_l, o_l turn the witnesses in the cumulative products:

$$W_{\mathcal{V}_l} = \left(a_{0_0}^{\prod_{k=1}^{l}(x'+i_k)}\right)^{\prod_{k=1}^{l} r_k}, W_{\Xi_\mathcal{V} \setminus V} = \left(a_{0_0}^{\prod_{k=1}^{|\Xi_\mathcal{V} \setminus V|}(x'+i_k)}\right)^{\prod_{k=1}^{\ell} r_k^{-1}}$$

and

$$W_{\mathcal{L}_l} = \left(X_{0_0}^{\prod_{k=1}^{l} \prod_{w \in f_\mathcal{V}(i_k)}(x'+w)}\right)^{\prod_{k=1}^{l} o_k \cdot r_k^{-1}}, W_{\Xi_\mathcal{L} \setminus E} = \left(a_{0_0}^{\prod_{k=1}^{|\Xi_\mathcal{L} \setminus E|} \prod_{w \in f_\mathcal{V}(i_k)}(x'+w)}\right)^{\prod_{k=1}^{\ell} o_k^{-1} \cdot r_k}$$

into MoniPoly commitments which are perfectly hiding. □

5.6 Edge Composition

The edge composition works in the similar way as the vertex composition and is, again, conceptually different from the edge composition statement of the original graph signature scheme [14]. Here, we need to run at least three proofs of cumulative products: one for the even-indexed edge identifiers, one for the odd-indexed edge identifiers and one for the edge labels. This is because all but the first and the last vertex identifiers appear at least twice in the edges. The number of set membership proof increases with respect to the number of branches in a graph. A simple way to separate the edges is to traverse the entire graph and record the edge identifiers into vertex sets $E_1, \ldots, E_\ell$ such that every vertex identifier is unique in its own set. We first consider the scenario of an undirected acyclic graph. X2 The protocol below establishes the statement edges(E):

$$PK\Big\{(\forall (i,j) \in E : \varepsilon_{(i,j)_0}, \varepsilon_{(i,j)_1}, \varepsilon_{(i,j)_2}) :$$
$$\mathsf{e}\left(\prod_{(i,j)\in E} C_{(i,j)}, X_{0_0}\right) = \prod_{(i,j)\in E} \mathsf{e}\left(W_{(i,j)}, \prod_{j=0}^{2} X_j^{\varepsilon_{(i,j)_j}}\right) \wedge$$
$$\varepsilon_{(i,j)}[C_{(i,j)_{E_1}}] \in \Xi_\mathcal{V} \wedge \cdots \wedge \varepsilon_{(i,j)}[C_{(i,j)_{E_\ell}}] \in \Xi_\mathcal{V} \wedge$$
$$W_{(i,j)}[C_{(i,j)_E}] \in \Xi_\mathcal{L} \Big\}$$

where the first statement is the bootstrap statement.

Lemma 4. *The randomization of* $\{W_{\mathcal{V}_1}, \ldots, W_{\mathcal{V}_{|E_k|}}, W_{\Xi_\mathcal{V} \setminus E_k}\}_{k=1}^{\ell}$ *and* $(W_{\mathcal{L}_1}, \ldots, W_{\mathcal{L}_{|E|}}, W_{\Xi_\mathcal{L} \setminus f_\mathcal{E}(V)})$ *in* edges *are perfectly hiding.*

X2 **XXX TODO:** This description is opaque to me and lacks intuition. Why is the cumulative product computed? Why on odd vs. even-indexed edges. This will need fixing before submission to PKC

Proof. The proof is the similar to that of Lemma 3. □

In addition, the edge identifier in a directed acyclic graph is represented by three vertex identifiers. For instance, (i,j,j) represents a directed edge from i to j. We can construct the edge composition proof for directed acyclic graph with a minor modification to its undirected counterpart. To be precise, the prover runs the proof of undirected version using (i,j) extracted from (i,j,j), in addition to proving j is a valid vertex identifier from $\Xi_{\mathcal{V}}$. We describe the proof as follows:

$$
\begin{aligned}
PK\Bigg\{ & (\forall (i,j,j) \in E : \varepsilon_{(i,j,j)_0}, \varepsilon_{(i,j,j)_1}, \varepsilon_{(i,j,j)_2}, \varepsilon_{(i,j,j)_3}, \varepsilon_{j_0}, \varepsilon_{j_1}) : \\
& \mathsf{e}\left(\prod_{(i,j,j)\in E} C_{(i,j,j)}, X_{0_0}\right) = \prod_{(i,j,j)\in E} \mathsf{e}\left(W_{(i,j,j)}, \prod_{k=0}^{3} X_{0_k}^{\varepsilon_{(i,j,j)_j}}\right) \wedge \\
& \mathsf{e}\left(a_{0_0}, \prod_{(i,j,j)\in E} \prod_{k=0}^{3} X_{0_k}^{\varepsilon_{(i,j,j)_j}}\right) = \prod_{(i,j,j)\in E} \mathsf{e}\left(W_{(i,j)}, a_{0_1}^{\varepsilon_{j_1}} X_{0_0}^{\varepsilon_{j_0}}\right) \wedge \\
& W_{(i,j)}[C_{(i,j,j)_{E_1}}] \in \Xi_{\mathcal{V}} \wedge \cdots \wedge W_{(i,j)}[C_{(i,j,j)_{E_\ell}}] \in \Xi_{\mathcal{V}} \wedge \varepsilon_j[C_{(i,j,j)}] \in \Xi_{\mathcal{V}} \\
& W_{(i,j,j)}[C_{(i,j,j)_E}] \in \Xi_{\mathcal{L}} \Bigg\}
\end{aligned}
$$

where an additional proof of cumulative product is needed for every extra vertices $j \in E$. At this point, it is clear that whether a graph is acyclic or cyclic does not has an impact on our edges statement. In the subsequent sections, we assume the graph is always an undirected graph in order to ease the explanation.

5.7 Graph Well-formedness

Combining the statements above, we can construct a proof of well-formedness for an encoded graph as below:

$$
\begin{aligned}
PK\{(\forall i \in V : \varepsilon_{i_0}, \varepsilon_{i_1}, \forall (i,j) \in E : \varepsilon_{(i,j)_0}, \varepsilon_{(i,j)_1}, \varepsilon_{(i,j)_2}) : \\
\mathsf{graph}(C) \wedge \mathsf{bootstrap}(C) \wedge \mathsf{vertices}(V) \wedge \mathsf{edges}(E)\}
\end{aligned}
$$

Theorem 4. *The compound proof of knowledge of* graph, bootstrap, vertices *and* edges *establishes the well-formedness of an encoded graph.*

Proof. Lemma 1 to Lemma 4 reduces the hardness of breaking the perfect hiding property in the proofs to breaking that of MoniPoly set commitment and Pedersen set commitment. This implicitly reduces the hardness of breaking the graph well-formedness to the hardness of breaking the binding property of the two commitment schemes which assume the hardness of the co-DLOG problem and DLOG problem, respectively. Therefore, the graph well-formedness holds if co-DLOG and DLOG problems are hard. □

6 A New Graph Signature Scheme

6.1 Interface

We adapt the scheme interface from the MoniPoly anonymous attribute-based credential (ABC) system [23, 24] for our new graph signature scheme[2] In the graph signature context, the access control policy ϕ_{stmt} is termed as the predicate an we view a graph signature as six algorithms $\mathsf{GS} = \{\mathsf{KeyGen_S}, \mathsf{KeyGen_U}, (\mathsf{Obtain}, \mathsf{Sign}), (\mathsf{Prove}, \mathsf{Verify})\}$.

To fully exploit the features of a graph signature scheme, we further divide the signing process (Obtain, Sign) into three parts: 1) begins with an initial signing (InitObtain, InitSign), 2) optionally continues to at least one intermediate signing for graph accumulation (IntermObtain, IntermSign), 3) ends with a final signing (FinalObtain, FinalSign). If a user runs the initial signing followed by the final signing, it resembles Groß' single signing [14]. If there is at least an intermediate signing in between the initial and final signings, it realizes Groß' joint graph signing [14]. This design makes the support of multi-signer applications a trivial effort, while still conforming to our rigorous security model. The scheme interface is described as follows:

1. $\mathsf{KeyGen_S}(1^k, 1^L, 1^n) \to (pk_\mathsf{S}, sk_\mathsf{S})$: This algorithm is executed by the signer (S). On the input of the security parameter k, maximum supported graph size L and the graph attributes upper bound n, it generates a key pair $(pk_\mathsf{S}, sk_\mathsf{S})$.
2. $\mathsf{KeyGen_U}(pk_\mathsf{S}) \to (sk_\mathsf{U})$: This algorithm is executed by the user to generate a user (U) secret key pair sk_U.
3. $(\mathsf{Obtain}(pk_\mathsf{S}, A_\mathsf{U}), \mathsf{Sign}(pk_\mathsf{S}, sk_\mathsf{S}, A_\mathsf{S})) \to (\sigma \text{ or } \perp)$: The signing protocol outputs a valid graph signature σ for two message sets $(A_\mathsf{U}, A_\mathsf{S})$ if it completes successfully. Otherwise, it outputs a null value $\perp$. The signing protocol starts with the initial signing and ends with the final signing:
   (a) $(\mathsf{InitObtain}(pk_\mathsf{S}, \{sk_\mathsf{U}, s'\}), \mathsf{InitSign}(pk_\mathsf{S}, sk_\mathsf{S}, \{\mathcal{G}_\mathsf{S}\})) \to (\sigma_{init} \text{ or } \perp)$: The signing process always starts with this initial signing protocol. It signs a committed user secret key sk_U and a graph data set $\mathcal{G}_\mathsf{S}$ given by the signer. User private input is a set $A_\mathsf{U} = \{sk_\mathsf{U}, s'\}$ which contains the user secret key sk_U and its opening value s'. Signer private inputs are the signer secret key sk_S and a graph data set $A_\mathsf{S} = \mathcal{G}_\mathsf{S}$. At the end of the protocol, the algorithm outputs a valid initial signature σ_{init} and a null value $\perp$ otherwise.
   (b) $(\mathsf{IntermObtain}(pk_\mathsf{S}, \{sk_\mathsf{U}, s', \mathcal{G}_\mathsf{U}, \{o_i, o_{(i,j)}\}\}), \mathsf{IntermSign}(pk_\mathsf{S}, sk_\mathsf{S}, \{\mathcal{G}_\mathsf{S}\})) \to (\sigma_{interm} \text{ or } \perp)$: This optional intermediary signing protocol signs a committed user secret key sk_U, a committed hidden graph data set $\mathcal{G}_\mathsf{U}$ and

[2] While Groß' graph signature scheme [14] was the first and only one of its kind and bears similarities to the Camenisch-Groß ABC system [7], both systems inherited their security properties from the underlying CL signature scheme and its proof system and were not proven rigorously for their overall systems. As we aim for a tight reduction proof, a properly defined interface and security notions, such as that by MoniPoly ABC system [24] appear to be a sound starting point.

a graph data set $\mathcal{G}_{\mathsf{S}}$ given by the signer. User private input is a set $A_{\mathsf{U}} = \{sk_{\mathsf{U}}, s', \mathcal{G}_{\mathsf{U}}, \{o_i, o_{(i,j)}\}\}$ which contains the user secret key sk_{U} and his hidden graph $\mathcal{G}_{\mathsf{U}}$ where s' and $\{o_i, o_{(i,j)}\}$ are their opening values, respectively. Signer private inputs are the signer secret key sk_{S} and a graph data set $A_{\mathsf{S}} = \mathcal{G}_{\mathsf{S}}$. At the end of the protocol, the algorithm outputs a valid intermediate signature σ_{interm} and a null value $\perp$ otherwise.

(c) $(\mathsf{FinalObtain}(pk_{\mathsf{S}}, \{sk_{\mathsf{U}}, s', \mathcal{G}_{\mathsf{U}}, \{o_i, o_{(i,j)}\}\}), \mathsf{FinalSign}(pk_{\mathsf{S}}, sk_{\mathsf{S}}, \perp)) \to (\sigma \text{ or } \perp)$: The signing process always ends with this final signing protocol. It signs a committed user private key sk_{U} and a committed hidden graph $\mathcal{G}_{\mathsf{U}}$. User private input is a set $A_{\mathsf{U}} = \{sk_{\mathsf{U}}, s', \mathcal{G}_{\mathsf{U}}, \{o_i, o_{(i,j)}\}\}$ which contains the user secret key sk_{U} and his hidden graph $\mathcal{G}_{\mathsf{U}}$ where s' and $\{o_i, o_{(i,j)}\}$ are their opening values, respectively. Signer private input is the signer secret key sk_{S}. At the end of the protocol, the algorithm outputs a valid graph signature σ and a null value $\perp$ otherwise.

4. $(\mathsf{Prove}(pk_{\mathsf{S}}, \sigma, \phi_{\mathsf{stmt}}), \mathsf{Verify}(pk_{\mathsf{S}}, \phi_{\mathsf{stmt}}) \to (b)$: This interactive showing protocol establishes a show proof for the predicate ϕ_{stmt} requested by verifier such that $\phi_{\mathsf{stmt}}(A) = 1$. If $\phi_{\mathsf{stmt}}(A) = 0$, the graph signature holder aborts and Verify outputs $b = 0$. At the end of the protocol, Verify outputs $b = 1$ if it accepts prover's proof and outputs $b = 0$ otherwise.

In the subsequent sections, we will use Obtain (resp. Sign) to represent all three sub-modules.

6.2 Proof of Knowledge Predicates

Our proposed new graph signature scheme can support all predicate statements offered by Groß's graph signature [15]. For instance, Groß's partition statement is implicitly realized by our vertices statement. The disjoint statement in Groß' scheme proves the pair-wise differences of vertices and edges in a graph but this has been indirectly done by our vertices and edges through their proof of cumulative products. Our disjoint statement is based on the NAND proof of the MoniPoly ABC [23, 24] which proves that a queried graph $\mathcal{G}'$ does not appear in the graph signature. We elaborate the proof details for the statements above in Section 6.4 and Section 7. Table 1 displays the supported statements in a predicate.

6.3 Security Requirements

We apply the *impersonation resilience* and *unlinkability* security notions from the MoniPoly ABC system to graph signature schemes due to their similar security requirements.

Impersonation Resilience. We define our security model as the security against impersonation under active and concurrent attacks (imp-aca) in the game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ as follows.

Table 1. Proof of knowledge predicate for graph signature.

Statement	Description
$\mathsf{set}(\mathcal{G}')$	Representation of a set $\mathcal{G}' \subseteq \mathcal{G}$, based on $\mathsf{AND}(\mathcal{G}')$.
$\mathsf{disjoint}(\mathcal{G}')$	Graph disjointness $(\mathcal{G}' \cap \mathcal{G}) = \emptyset$, based on $\mathsf{NAND}(\mathcal{G}')$.
$\mathsf{cover}(\ell, \mathcal{G}')$	Graph coverage $\lvert\mathcal{G}' \cap \mathcal{G}\rvert \geq \ell$, based on $\mathsf{ANY}(\ell, \mathcal{G}')$.
vertices	Proof of composition of graph vertices, based on set.
edges	Proof of composition of graph edges, based on set.
possession	Proof of possession of a graph signature σ on the graph $\mathcal{G} = (V \cup E)$.
$\mathsf{edge}(i, j)$	Adjacency of (i, j) is a set statement on an edge identifier.
$\mathsf{connected}(i, j, \ell)$	Existence of the ℓ-path between vertices i and j, based on edges and set statements.
$\mathsf{isolated}(i, j)$	Isolation of vertices i and j, based on set and disjoint.

Note: AND,NAND,ANY are the show proofs from TG-ABC systems.

Game 1 $(\mathsf{imp} - \mathsf{aca}(\mathcal{A}, \mathcal{C}))$

1. ***Setup:*** *$\mathcal{C}$ runs $\mathsf{KeyGen}(1^k, 1^n)$ and sends pk_{S} to $\mathcal{A}$.*
2. ***Phase 1:*** *$\mathcal{A}$ is able to issue concurrent queries to the Obtain, Prove and Verify oracles where he plays the role of user, prover and verifier, respectively. $\mathcal{A}$ can query a set A_{U_i} (resp., ϕ_i) of his choice to Obtain (resp., Prove and Verify) in the i-th query. $\mathcal{A}$ can also issue queries to the SignTranscript oracle which takes in A_{U_i} and returns its signing protocol transcript.*
3. ***Challenge:*** *$\mathcal{A}$ outputs the challenge data set A^*_{U} and its corresponding predicate ϕ^* such that $\phi^*(A_{\mathsf{U}_i}) = 0$ and $\phi^*(A^*_{\mathsf{U}}) = 1$ for every A_{U_i} queried to the Obtain oracle during Phase 1.*
4. ***Phase 2:*** *$\mathcal{A}$ can continue to query the oracles as in Phase 1 with the restriction that it cannot query a data set A_{U_i} to Obtain such that $\phi^*(A_{\mathsf{U}_i}) = 1$.*
5. ***Impersonate:*** *$\mathcal{A}$ completes a showing protocol as the prover with $\mathcal{C}$ as the verifier for the predicate $\phi^*(A^*_{\mathsf{U}}) = 1$. $\mathcal{A}$ wins the game if $\mathcal{C}$ outputs 1.*

Definition 6. *An adversary $\mathcal{A}$ is said to $(t_{\mathsf{imp}}, \varepsilon_{\mathsf{imp}})$-break the security against impersonation under active and concurrent attacks (imp-aca) of a graph signature scheme if $\mathcal{A}$ runs in time at most t_{imp} and wins in Game 1 such that:*

$$\Pr[(\mathcal{A}, \mathsf{Verify}(pk, \phi^*)) = 1] \geq \varepsilon_{\mathsf{imp}}$$

for a negligible probability $\varepsilon_{\mathsf{imp}}$. We say that a graph signature scheme is imp-aca-secure if no adversary $(t_{\mathsf{imp}}, \varepsilon_{\mathsf{imp}})$-wins Game 1.

Unlinkability. The security model for graph unlinkability under active and concurrent attacks (gunl-aca) is defined as a game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. The gunl-aca security does not consider the collusion in between $\mathcal{A}$

and signer. We argue that such security is sufficient for a graph signature scheme because the signer is always a trusted party.

Game 2 ($\mathsf{gunl} - \mathsf{aca}(\mathcal{A}, \mathcal{C})$)

1. ***Setup:*** *$\mathcal{C}$ runs* KeyGen *and sends $pk_\mathsf{S}, sk_\mathsf{S}$ to $\mathcal{A}$.*
2. ***Phase 1:*** *$\mathcal{A}$ is able to issue concurrent queries to the* Commit*,* Obtain*,* Prove *and* Verify *oracles where he plays the role of committer, user, signer, prover and verifier, respectively, on any hidden set A_{U_i} of his choice in the i-th query. $\mathcal{A}$ can also issue queries to an additional oracle, namely,* Corrupt *that when queried with a protocol session identifier, returns the entire internal state of a user in a signing protocol, or the entire internal state of a prover in the showing protocol.*
3. ***Challenge:*** *$\mathcal{A}$ decides the two non-empty hidden set $A_{\mathsf{U}_0} = \{sk_{\mathsf{U}_0}, s'_0, \mathcal{G}_{\mathsf{U}_0} = \{V_0 \cup E_0\}, \{o_{0_i}, o_{0_{(i,j)}}\}\}, A_{\mathsf{U}_0} = \{sk_{\mathsf{U}_1}, s'_1, \mathcal{G}_{\mathsf{U}_1} = \{V_1 \cup E_1\}, \{o_{1_i}, o_{1_{(i,j)}}\}\}$ with $|V_0| = |V_1|, |E_0| = |E_1|$ and the predicate ϕ^* which he wishes to challenge such that $\phi^*(A_{\mathsf{U}_0}) = \phi^*(A_{\mathsf{U}_0}) = 1$. $\mathcal{A}$ is allowed to select $A_{\mathsf{U}_0}, A_{\mathsf{U}_1}$ from the existing queries to* Obtain *in Phase 1. $\mathcal{C}$ responds by randomly choosing a challenge bit $b \in \{0, 1\}$ and interacts as the user with $\mathcal{A}$ as the signer to complete the protocols in the order:*

$$(\mathsf{InitObtain}(pk_\mathsf{S}, \{sk_\mathsf{U}, s'\}), \mathsf{InitSign}(pk_\mathsf{S}, sk_\mathsf{S}, \{\mathcal{G}_{\mathsf{U}_b}\})) \to \sigma_{init,b},$$
$$(\mathsf{FinalObtain}(pk_\mathsf{S}, A_{\mathsf{U}_b}), \mathsf{FinalSign}(pk_\mathsf{S}, sk_\mathsf{S}, \perp)) \to \sigma_b,$$
$$(\mathsf{InitObtain}(pk_\mathsf{S}, \{sk_\mathsf{U}, s'\}), \mathsf{InitSign}(pk_\mathsf{S}, sk_\mathsf{S}, \{\mathcal{G}_{\mathsf{U}_{1-b}}\})) \to \sigma_{init,1-b},$$
$$(\mathsf{FinalObtain}(pk_\mathsf{S}, A_{\mathsf{U}_{1-b}}), \mathsf{FinalSign}(pk_\mathsf{S}, sk_\mathsf{S}, \perp)) \to \sigma_{1-b}$$

   where sk_U, s' are randomly chosen by $\mathcal{C}$. If there are n times intermediary signing in between, the challenge graph is formatted as $\mathcal{G}_{\mathsf{U}_b} = \bigcup\{\mathcal{G}_{b,0}, \ldots, \mathcal{G}_{b,n}\}$ such that $\mathcal{G}_{b,i} = \{V_{b,i} \cup E_{b,i}\}, |V_{b,i}| = |V_{1-b,i}|, |E_{b,i}| = |E_{1-b,i}|$ for $0 \leq i \leq n$. The initial signer signs the graphs $\mathcal{G}_{b,0}, \mathcal{G}_{1-b,0}$ and the i-th intermediary signer signs on the graphs $\mathcal{G}_{b,i}, \mathcal{G}_{1-b,i}$. Subsequently, $\mathcal{C}$ interacts as the prover with $\mathcal{A}$ as the verifier for polynomially many times as requested by $\mathcal{A}$ to complete the protocols in the same order:

$$(\mathsf{Prove}(pk_\mathsf{S}, \sigma_b, \phi^*), \mathsf{Verify}(pk_\mathsf{S}, \phi^*)) \to 1,$$
$$(\mathsf{Prove}(pk_\mathsf{S}, \sigma_{1-b}, \phi^*), \mathsf{Verify}(pk_\mathsf{S}, \phi^*)) \to 1.$$

4. ***Phase 2:*** *$\mathcal{A}$ can continue to query the oracles as in Phase 1 except querying the transcripts of the challenged signing and presentation protocols to* Corrupt*.*
5. ***Guess:*** *$\mathcal{A}$ outputs a guess b' and wins the game if $b' = b$.*

Definition 7. *An adversary $\mathcal{A}$ is said to $(t_\mathsf{gunl}, \varepsilon_\mathsf{gunl})$-break the security of graph unlinkability under active and concurrent attacks (*gunl-aca*) of a graph signature scheme if $\mathcal{A}$ runs in time at most t_gunl and wins in Game 2 such that:*

$$|\Pr[b = b'] - \frac{1}{2}| \geq \varepsilon_\mathsf{gunl}$$

for a negligible probability $\varepsilon_{\mathsf{gunl}}$. We say that a graph signature is gunl-aca*-secure if no adversary $(t_{\mathsf{gunl}}, \varepsilon_{\mathsf{gunl}})$-wins Game 2.*

The protocol unlinkability under active and concurrent attack (punl-aca) is defined as a game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ as follows:

Game 3 ($\mathsf{punl} - \mathsf{aca}(\mathcal{A}, \mathcal{C})$)

1. ***Setup:*** *Same to that of graph unlinkability game.*
2. ***Phase 1:*** *Same to that of graph unlinkability game.*
3. ***Challenge:*** *Same as that of graph unlinkability game except that $\mathcal{C}$ responds by randomly choosing two challenge bits $b_1, b_2 \in \{0,1\}$ and interacts as the user with $\mathcal{A}$ as the signer to complete the protocols in the order:*

$$(\mathsf{InitObtain}(pk_{\mathsf{S}}, \{sk_{\mathsf{U}}, s'\}), \mathsf{InitSign}(pk_{\mathsf{S}}, sk_{\mathsf{S}}, \{\mathcal{G}_{\mathsf{U}_{b_1}}\})) \to \sigma_{init,b_1},$$
$$(\mathsf{FinalObtain}(pk_{\mathsf{S}}, A_{\mathsf{U}_{b_1}}), \mathsf{FinalSign}(pk_{\mathsf{S}}, sk_{\mathsf{S}}, \perp)) \to \sigma_{b_1},$$
$$(\mathsf{InitObtain}(pk_{\mathsf{S}}, \{sk_{\mathsf{U}}, s'\}), \mathsf{InitSign}(pk_{\mathsf{S}}, sk_{\mathsf{S}}, \{\mathcal{G}_{\mathsf{U}_{b_2}}\})) \to \sigma_{init,b_2},$$
$$(\mathsf{FinalObtain}(pk_{\mathsf{S}}, A_{\mathsf{U}_{b_2}}), \mathsf{FinalSign}(pk_{\mathsf{S}}, sk_{\mathsf{S}}, \perp)) \to \sigma_{b_2}.$$

   If there are n intermediary signing protocols, they also follow the order above. Subsequently, $\mathcal{C}$ interacts as the prover with $\mathcal{A}$ as the verifier for polynomially many times as requested by $\mathcal{A}$ to complete the protocols in the same order:

$$(\mathsf{Prove}(pk_{\mathsf{S}}, \sigma_{b_2}, \phi^*), \mathsf{Verify}(pk_{\mathsf{S}}, \phi^*)) \to 1,$$
$$(\mathsf{Prove}(pk_{\mathsf{S}}, \sigma_{1-b_2}, \phi^*), \mathsf{Verify}(pk_{\mathsf{S}}, \phi^*)) \to 1.$$

4. ***Phase 2:*** *Same to that of graph unlinkability game.*
5. ***Guess:*** *$\mathcal{A}$ outputs a guessed pair of signing protocol transcript $\pi_{(O,S)}$ and show proof transcript $\pi_{(P,V)}$ and wins the game if the pair is under the same credential such that $\sigma_{\pi_{(O,S)}} = \sigma_{\pi_{(P,V)}}$.*

Definition 8. *An adversary $\mathcal{A}$ is said to $(t_{\mathsf{punl}}, \varepsilon_{\mathsf{punl}})$-break the security of the protocol unlinkability under active and concurrent attacks (punl-aca) of a graph signature scheme if $\mathcal{A}$ runs in time at most t_{punl} and wins in Game 3 such that:*

$$|\Pr[\sigma_{\pi_{(O,S)}} = \sigma_{\pi_{(P,V)}}] - \frac{1}{2}| \geq \varepsilon_{\mathsf{punl}}$$

for a negligible probability $\varepsilon_{\mathsf{punl}}$. We say that an ABC system is punl-aca*-secure if no adversary $(t_{\mathsf{punl}}, \varepsilon_{\mathsf{punl}})$-wins Game 3.*

6.4 A New Graph Signature Scheme

In this section, we first describe the KeyGen of the graph signature, followed by the graph signing protocols and its showing protocols. At a quick glance, we show the structure of the graph signature, which is a SDH-CL signature on the MoniPoly encoded graph:

$$v^{x+t} = h^{sk_{\mathsf{U}}} \underbrace{\cdots a_{i,0}^{(x'+i)\prod_{k \in f_{\mathcal{V}}}(x'+k)} \cdots}_{\forall \text{ vertices } i} \underbrace{\cdots a_{(i,j),0}^{(x'+i)(x'+j)\prod_{k \in f_{\mathcal{E}}}(x'+k)} \cdots}_{\forall \text{ edges } (i,j)} b^s c$$

6.5 Key Generation (S)

Construct three cyclic groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ of order p based on an elliptic curve whose bilinear pairing is $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. Choose two random secret values $x, x' \in \mathbb{Z}_p^*$ and select random generators $b, c, d, h \in \mathbb{G}_1, \{a_{i_0} \in \mathbb{G}_1, X_{i_0} \in \mathbb{G}_2\}_{i=0}^{L}$ to compute the values $\{\{a_{i_k} = a_{i_0}^{x'^k}, X_{i_k} = X_{i_0}^{x'^k}\}_{i=0}^{L}\}_{k=1}^{n}$. Define the function $\mathsf{MPEncode} : \mathbb{Z}_p^n \rightarrow \mathbb{Z}_p^{n+1}$ that converts a set of n attributes into coefficients for a monic polynomial of degree $n+1$. The public key is $pk = (\mathsf{e}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, b, c, d, h, \{\{a_{i_k}, X_{i_k}\}_{i=0}^{L}\}_{k=0}^{n}, X, \Xi_{\mathcal{V}}, \Xi_{\mathcal{L}})$ and the secret key is $sk = (x, x')$ where $\Xi_{\mathcal{V}}$ is the list of vertex identifiers and $\Xi_{\mathcal{L}}$ is the list of vertex and edge labels. The parameter L is the maximum vertices and edges allowed in a graph, while n indicates that each vertex can have no more than $n-1$ labels and each edge can have no more than $n-2$ labels.

6.6 Key Generation (U)

The user generates the user secret key as $sk_{\mathsf{U}} \in \mathbb{Z}_p^*$.

6.7 Graph Signing Protocols

As these signing protocols require the signers to transfer the graphs to the user, we assume a secure channel is always in place.

6.7.1 Initial Signing The signing protocol begins with a user providing a proof of representation for the Pedersen commitment C of his secret key sk_{U}. If the proof is verified, using the CL-SDH signature, signer signs on C together with an assigned graph $\mathcal{G}_{\mathsf{S}}$ decided by the signer. If the returned graph signature σ' is a valid signature on C and $\mathcal{G}_{\mathsf{S}}$, the user finalizes the graph signature as σ_{init}.

We assume the signer knows the discrete logarithms of bases $d, a_{0_0}, a_{1_0}, \ldots, a_{L_0}$ with respect to b. Without loss of generality, we also assume the signer always utilize the bases in an incremental sequence such that $a_{1_0}, \ldots, a_{L_0}$. For $\mathcal{G}_{\mathsf{S}}$, let the vertex exponents be $\bar{e}_i = \mathsf{dlog}_b(a_{i_0})(x'+i)\prod_{w \in f_{\mathcal{V}}(i)}(x'+w)$ while the edge exponents be $\bar{e}_{(i,j)} = \mathsf{dlog}_b(a_{(i,j)_0})(x'+i)(x'+j)\prod_{w \in f_{\mathcal{E}}(i,j)}(x'+w)$. The initial graph signing protocol is as follows:

1. User randomly selects $s' \in \mathbb{Z}_p^*$ and interacts with signer to prove the well-formedness of his hidden graph:
$$PK\{(\zeta, \rho) : C = h^{\zeta} b^{\rho}\}$$
where $\zeta = sk_{\mathsf{U}}$ and $\rho = s'$.
2. If the proof is verified, signer signs on C and an assigned graph $\mathcal{G}_{\mathsf{S}}$ as:
$$v = \left(Cb^{s''+\ldots+\bar{e}_i+\ldots+\bar{e}_{(i,j)}+\ldots+\mathsf{dlog}_b(d)}\right)^{(x+t)^{-1}}$$
for randomly selected $s'', t \in \mathbb{Z}_p^*$. Signer returns $(\sigma' = (t, s'', v), \mathcal{G}_{\mathsf{S}})$ to the user.

3. If σ' is a valid SDH-CL signature on C and $\mathcal{G}_{\mathsf{S}}$, user finalizes his graph signature as $\sigma_{init} = (t, s, v, \mathcal{G}_{\mathsf{U}})$ where $s = s' + s'' \mod p$ and hidden graph $\mathcal{G}_{\mathsf{U}} = \mathcal{G}_{\mathsf{S}}$.

Remark 1. The signing protocol above dispense with a secure channel by employing an additional encryption step. The Pedersen commitment of the user secret key $C = h^{sk_{\mathsf{U}}} b^{s'}$ can be treated as an ElGamal public key. The signer can perform a hybrid ElGamal encryption on the signature and graphs to hides all the information transferred to the user. This approach is also applicable to the intermediary and final signing protocols.

6.7.2 Intermediary Signing The intermediary signing protocol is useful for the applications which require the user to approach different signers to gather the needed graphs. This protocol is the same as above except it begins with a user providing a proof of possession for a blinded initial signature $\sigma'_{init} = (t, s, v)$ such that:

$$v' = v^{r_1 y^{-1}}, s' = sr_1 \mod p, t' = ty \mod p$$

where $r_1, y \in \mathbb{Z}_p^*$ are the random blinding factors. The correctness for the blinded signature σ'_{init} can be verified as follows:

$$\begin{aligned}
&\mathsf{e}\left(h^{sk_{\mathsf{U}} \cdot r_1} \prod_{i \in V} C_i^{r_1} \prod_{(i,j) \in E} C_{(i,j)}^{r_1} b^{s'} d^{r_1} v'^{-t'}, X_{0_0}\right) \\
&= \mathsf{e}\left(h^{sk_{\mathsf{U}}} \prod_{i \in V} C_i \prod_{(i,j) \in E} C_{(i,j)} b^s d (v^{y^{-1}})^{-ty}, X_{0_0}\right)^{r_1} \\
&= \mathsf{e}(v^{(x+t)} v^{-t}, X_{0_0})^{r_1} \\
&= \mathsf{e}(v'^{y}, X)
\end{aligned}$$

where $\mathcal{G}_{\mathsf{U}} = (V \cup E)$. The user also explicitly proves the correctness for the Pedersen commitment of his secret key sk_{U}:

$$C_1 = h^{sk_{\mathsf{U}} r_1} b^{sr_1}$$

and the MoniPoly commitment of his hidden graph $\mathcal{G}_{\mathsf{U}}$:

$$\begin{aligned}
&C_i = a_{i_0}^{o_{i_1}(x'+i) \prod_{w \in f_{\mathcal{V}}(i)} (x'+w)}, C_{(i,j)} = a_{(i,j)_0}^{o_{(i,j)_1}(x'+i)(x'+j) \prod_{w \in f_{\mathcal{E}}(i,j)} (x'+w)}, \\
&C_2 = \prod_{i \in V_{\mathsf{U}}} C_i^{r_1 o_{i_1}^{-1}} \prod_{(i,j) \in E_{\mathsf{U}}} C_{(i,j)}^{r_1 o_{(i,j)_1}^{-1}}
\end{aligned}$$

where $r_1, o_{i_1}, o_{(i,j)_1} \in \mathbb{Z}_p^*$ are the random blinding factors. We describe the intermediary signing protocol as follows:

1. User randomly selects $r_1, y, o_{i_1}, o_{(i,j)_1}, r_i, r_{(i,j)} \in \mathbb{Z}_p^*$ and interacts with the signer to prove the possession of σ_{init} and the representation of his hidden graph $\mathcal{G}_{\mathsf{U}}$:

$$PK\Bigg\{((\forall i \in V_{\mathsf{U}} : \varepsilon_{i_0}, \varepsilon_{i_1}, \varepsilon_i), (\forall (i,j) \in E_{\mathsf{U}} : \varepsilon_{(i,j)_0}, \varepsilon_{(i,j)_1}, \varepsilon_{(i,j)}), \zeta, \rho, \omega, \tau, \gamma) :$$

$$\mathsf{e}\left(C_1 C_2 d^{\omega} v'^{-\tau}, X_{0_0}\right) = \mathsf{e}(v'^{\gamma}, X) \ \wedge C_1 C_2 = h^{\zeta} b^{\rho} \prod_{i \in V_{\mathsf{U}}} C_i^{\varepsilon_i} \prod_{(i,j) \in E_{\mathsf{U}}} C_{(i,j)}^{\varepsilon_{(i,j)}} \wedge$$

$$\mathsf{e}\left(\prod_{i \in V_{\mathsf{U}}} C_i \prod_{(i,j) \in E_{\mathsf{U}}} C_{(i,j)}, X_{0_0} \right) =$$

$$\prod_{i \in V_{\mathsf{U}}} \mathsf{e}\left(W_i', \prod_{k=0}^{1} X_{0_k}^{\varepsilon_{i_k}} \right) \prod_{(i,j) \in E_{\mathsf{U}}} \mathsf{e}\left(W_{(i,j)}', \prod_{k=0}^{1} X_{0_k}^{\varepsilon_{(i,j)_k}} \right) \Bigg\}$$

where $\zeta = sk_{\mathsf{U}} r_1, \rho = sr_1, \omega = r_1, \tau = ty, \gamma, = y, \varepsilon_{i_1} = r_i, \varepsilon_{i_0} = r_i i, \varepsilon_i = r_1 o_{i_1}^{-1}, \varepsilon_{(i,j)_1} = r_{(i,j)}, \varepsilon_{(i,j)_0} = r_{(i,j)} i, \varepsilon_{(i,j)} = r_1 o_{(i,j)_1}^{-1}$ while the witnesses are

$$W_i' = a_{i_0}^{o_{i_1} r_i^{-1} \prod_{w \in f_{\mathcal{V}}(i)} (x'+w)}$$

and

$$W_{(i,j)}' = a_{(i,j)_0}^{o_{(i,j)_1} r_{(i,j)}^{-1} (x'+j) \prod_{w \in f_{\mathcal{E}}(i,j)} (x'+w)}.$$

Note that the exponent i for vertices and edges can always be swapped with a randomly selected attribute w.
2. If the proof is verified, signer signs on $C_1, \{C_i, C_{(i,j)}\}$ and an assigned graph $\mathcal{G}_{\mathsf{S}}$ as:

$$v = \left(C_1 \prod_{i \in V_{\mathsf{U}}} C_i \prod_{(i,j) \in E_{\mathsf{U}}} C_{(i,j)} b^{s'' + \ldots + \bar{e}_i + \ldots + \bar{e}_{(i,j)} + \ldots + \mathsf{dlog}_b(d)} \right)^{(x+t)^{-1}}$$

for randomly selected $s'', t \in \mathbb{Z}_p^*$. Signer returns $(\sigma' = (t, s'', v), \mathcal{G}_{\mathsf{S}})$ to the user.
3. If σ' is a valid SDH-CL signature on $C_1, \{C_i, C_{(i,j)}\}$ and $\mathcal{G}_{\mathsf{S}}$, user updates his secret key as $sk_{\mathsf{U}} r_1$ and graph signature as $\sigma_{interm} = (t, sr_1 + s'', v, \mathcal{G}, \{o_{i_1}, o_{(i,j)_1}\})$ where $\mathcal{G} = \mathcal{G}_{\mathsf{U}} + \mathcal{G}_{\mathsf{S}}$.

Remark 2. A user can interact with more than one intermediate signer to accumulate the required graphs.

6.7.3 Final Signing

The final signing protocol is the same as the intermediary signing protocol except the signer does not assign a new graph. Similarly, the user also prove the correctness for the Pedersen commitment of his secret key sk_{U}:

$$C_1 = h^{sk_{\mathsf{U}} r_2} b^{sr_2}$$

and the MoniPoly commitment of his hidden graph $\mathcal{G} = \mathcal{G}_{\mathsf{U}} \cup \mathcal{G}_{\mathsf{S}}$:

$$\begin{aligned}
\forall i \in V_{\mathsf{U}} : C_i &= a_{i_0}^{o_{i_2} o_{i_1} (x'+i) \prod_{w \in f_{\mathcal{V}}(i)} (x'+w)}, \\
\forall i \in V_{\mathsf{S}} : C_i &= a_{i_0}^{o_{i_2} (x'+i) \prod_{w \in f_{\mathcal{V}}(i)} (x'+w)}, \\
\forall i \in E_{\mathsf{U}} : C_{(i,j)} &= a_{(i,j)_0}^{o_{(i,j)_2} o_{(i,j)_1} (x'+i)(x'+j) \prod_{w \in f_{\mathcal{E}}(i,j)} (x'+w)}, \\
\forall i \in E_{\mathsf{S}} : C_{(i,j)} &= a_{(i,j)_0}^{o_{(i,j)_2} (x'+i)(x'+j) \prod_{w \in f_{\mathcal{E}}(i,j)} (x'+w)}, \\
C_2 &= \prod_{i \in V} C_i^{r_2 o_{i_2}^{-1}} \prod_{(i,j) \in E} C_{(i,j)}^{r_2 o_{(i,j)_2}^{-1}}
\end{aligned}$$

where $r_2, o_{i_2}, o_{(i,j)_2} \in \mathbb{Z}_p^*$ are the random blinding factor. We describe the final signing protocol as follows:

1. User randomly selects $r_2, y, o_{i_2}, o_{(i,j)_2}, r_i, r_{(i,j)} \in \mathbb{Z}_p^*$ and interacts with the signer to prove the possession of $\sigma_{interm} = (t, s, v, \{o_{i_1}, o_{(i,j)_1}\})$ and the representation of his hidden graph:

$$PK\Bigg\{ ((\forall i \in V : \varepsilon_{i_0}, \varepsilon_{i_1}, \varepsilon_i), (\forall (i,j) \in E : \varepsilon_{(i,j)_0}, \varepsilon_{(i,j)_1}, \varepsilon_{(i,j)}), \zeta, \rho, \omega, \tau, \gamma) :$$

$$\mathsf{e}\left(C_1 C_2 d^{\omega} v'^{-\tau}, X_{0_0}\right) = \mathsf{e}(v'^{\gamma}, X) \ \wedge C_1 C_2 = h^{\zeta} b^{\rho} \prod_{i \in V} C_i^{\varepsilon_i} \prod_{(i,j) \in E} C_{(i,j)}^{\varepsilon_{(i,j)}} \wedge$$

$$\mathsf{e}\left(\prod_{i \in V} C_i \prod_{(i,j) \in E} C_{(i,j)}, X_{0_0} \right) =$$

$$\prod_{i \in V} \mathsf{e}\left(W_i', \prod_{k=0}^{1} X_{0_k}^{\varepsilon_{i_k}} \right) \prod_{(i,j) \in E} \mathsf{e}\left(W_{(i,j)}', \prod_{k=0}^{1} X_{0_k}^{\varepsilon_{(i,j)_k}} \right) \Bigg\}$$

   where $\varepsilon_{i_1} = r_i, \varepsilon_{i_0} = r_i i, \varepsilon_{(i,j)_1} = r_{(i,j)}, \varepsilon_{(i,j)_1} = r_{(i,j)} i$. We also have the witnesses as follows:

$$\begin{aligned}
\forall i \in V_{\mathsf{U}} : W_i &= a_{i_0}^{o_{i_2} o_{i_1} r_i^{-1} \prod_{w \in f_{\mathcal{V}}(i)} (x'+w)}, \\
\forall i \in V_{\mathsf{S}} : W_i &= a_{i_0}^{o_{i_2} r_i^{-1} \prod_{w \in f_{\mathcal{V}}(i)} (x'+w)}, \\
\forall i \in E_{\mathsf{U}} : W_{(i,j)} &= a_{(i,j)_0}^{o_{(i,j)_2} o_{(i,j)_1} r_{(i,j)}^{-1} (x'+j) \prod_{w \in f_{\mathcal{E}}(i,j)} (x'+w)}, \\
\forall i \in E_{\mathsf{S}} : W_{(i,j)} &= a_{(i,j)_0}^{o_{(i,j)_2} r_{(i,j)}^{-1} (x'+j) \prod_{w \in f_{\mathcal{E}}(i,j)} (x'+w)}
\end{aligned}$$

   such that the exponent i for vertices and edges can be swapped with a randomly selected attribute w.
2. If the proof is verified, signer signs on C_1 and $\{C_i, C_{(i,j)}\}$ as:

$$v = \left(C_1 \prod_{i \in V} C_i \prod_{(i,j) \in E} C_{(i,j)} b^{s'' + \mathsf{dlog}_b(c)} \right)^{(x+t)^{-1}}$$

for randomly selected $s'', t \in \mathbb{Z}_p^*$. Signer returns $(\sigma' = (t, s'', v))$ to the user.

3. If σ' is a valid SDH-CL signature on C_1 and $\{C_i, C_{(i,j)}\}$, user finalizes his secret key as $sk_{\mathsf{U}} r_2$ and graph signature as

$$\sigma = (t, sr_2 + s'', v, \mathcal{G}, \{o_{i_1} o_{i_2}, o_{(i,j)_1} o_{(i,j)_2}\}, \{o_{i_2}, o_{(i,j)_2}\})$$

where $\mathcal{G} = \mathcal{G}_{\mathsf{U}} + \mathcal{G}_{\mathsf{S}}$.

Remark 3. The proposed scheme can be modified to let the final signer assign a dummy graph that contains information such as the final signer's identity, signing date and a unique signature ID for the purpose of revocation.

In order to ease the explanation for the subsequent showing protocols, in the subsequent sections, we assume no intermediary signing has been performed prior to the generation of the final graph signature σ.

6.7.4 Proof of Possession The proof of possession is specified by the $\phi_{\mathsf{possession}}$ predicate. It is a compound proof of knowledge on the validity of prover's blinded final graph signature σ' and its encoded graph $\mathcal{G} = (V \cup E)$. Prover can interact with the verifier to construct the proof of possession as follows:

$$PK\Big\{(\forall i \in V : \varepsilon_{i_0}, \varepsilon_{i_1}), (\forall (i,j) \in E : \varepsilon_{(i,j)_0}, \varepsilon_{(i,j)_1}), \zeta, \rho, \omega, \tau, \gamma) :$$
$$\prod_{i \in V} \mathsf{e}\left(W'_i, X_{0_1}^{\varepsilon_{i_1}} X_{0_0}^{\varepsilon_{i_0}}\right) \prod_{(i,j) \in E} \mathsf{e}\left(W'_{(i,j)}, X_{0_1}^{\varepsilon_{(i,j)_1}} X_{0_0}^{\varepsilon_{(i,j)_0}}\right) \cdot$$
$$\mathsf{e}\left(h^{\zeta} b^{\rho} c^{\omega} v'^{-\tau}, X_{0_0}\right) = \mathsf{e}(v'^{\gamma}, X)\Big\}$$

where $v', \{W'_i, W'_{(i,j)}\}$ are the public inputs. Note that the prover can set $\{\varepsilon_j, \varepsilon_{(i,j)}\}$ as an attribute randomly chosen from the vertex and edge attributes, respectively.

Remark 4. Proving the knowledge of $\{\varepsilon_i, \varepsilon_{(i,j)}\}$ is a must in the possession statement. It should not be replaced by a witness W such that the equation

$$\mathsf{e}\left(W h^{\zeta} b^{\rho} c^{\omega} v'^{-\tau}, X_{0_0}\right) = \mathsf{e}(v'^{\gamma}, X)$$

holds. This is because an adversary knowing the quadruple $(v', W, h^{rsk_{\mathsf{U}}} b^{rs} c^r v^{rt}, v^r)$ from the transcript can easily create another valid quadruple:

$$(v'' = v'^y, W' = \left(\frac{W h^{rsk_{\mathsf{U}}} b^{rs} c^r v^{rt}}{h^{sk'_{\mathsf{U}}} b^{s'} c v^{t'}}\right)^z, h^{zsk'_{\mathsf{U}}} b^{zs'} c^z v''^{zt'}, v''^z)$$

such that:

$$\mathsf{e}\left(W' h^{r' sk'_{\mathsf{U}}} b^{r's'} c^{r'} v^{r't'}, X_{0_0}\right) = \mathsf{e}(v''^z, X).$$

We highlight that this feature is useful for a group signature scheme. Concisely, viewing the graph signing protocol as the group joining protocol, (t, s, v) is the group user secret key. Let W combine all elements at the left hand side, the validity of the group user secret key can be proven by the equation $\mathsf{e}(W, X_{0_0}) = \mathsf{e}(v'', X)$. Subsequently, the signing and verification can be constructed by using the Fiat-Shamir paradigm.

6.8 set Proof

When the verifier requests a show proof for the predicate $\phi_{\mathsf{set}(\mathcal{G}')}$, i.e., showing a sub-graph $\mathcal{G}' = (V' \cup E')$ is inside the graph $\mathcal{G}$ such that $\mathcal{G}' \subseteq \mathcal{G}$, prover can interact with the verifier to construct the set proof as follows:

$$PK\Bigg\{((\forall i \in V \setminus V' : \varepsilon_{i_0}, \varepsilon_{i_1}), (\forall (i,j) \in E \setminus E' : \varepsilon_{(i,j)_0}, \varepsilon_{(i,j)_1}), \zeta, \rho, \omega, \tau, \gamma) :$$
$$\prod_{i \in V \setminus V'} \mathsf{e}\left(W'_i, X_{0_1}^{\varepsilon_{i_1}} X_{0_0}^{\varepsilon_{i_0}}\right) \prod_{(i,j) \in E \setminus E'} \mathsf{e}\left(W'_{(i,j)}, X_{0_1}^{\varepsilon_{(i,j)_1}} X_{0_0}^{\varepsilon_{(i,j)_0}}\right) \cdot$$
$$\prod_{i \in V'} \mathsf{e}\left(W'_i, \prod_{k=0}^{|V'_i|} X_{0_k}^{\mathsf{m}_{i_k}}\right) \prod_{(i,j) \in E'} \mathsf{e}\left(W'_{(i,j)}, \prod_{k=0}^{|E'_{(i,j)}|} X_{0_k}^{\mathsf{m}_{(i,j)_k}}\right) \cdot$$
$$\mathsf{e}\left(h^{\zeta} b^{\rho} c^{\omega} v'^{-\tau}, X_{0_0}\right) = \mathsf{e}(v'^{\gamma}, X)\Bigg\}.$$

The correctness for the witnesses $\{W_i, W_{(i,j)}\}$ for the queried graph $\mathcal{G}'$ can be verified from the following:

$$\mathsf{e}\left(\prod_{i \in V'} C_i \prod_{(i,j) \in E'} C_{(i,j)}, X_{0_0}\right)$$
$$= \prod_{i \in V'} \mathsf{e}\left(a_{i_0}^{ro_i(x'+i)\prod_{w \in f_{\mathcal{V}}(i)}(x'+w)}, X_{0_0}\right) \prod_{(i,j) \in E'} \mathsf{e}\left(a_{(i,j)_0}^{ro_{(i,j)}(x'+i)(x'+j)\prod_{w \in f_{\mathcal{E}}(i,j)}(x'+w)}, X_{0_0}\right)$$
$$= \prod_{i \in V'} \mathsf{e}\left(a_{i_0}^{ro_i}, X_{0_0}^{(x'+i)\prod_{w \in f_{\mathcal{V}}(i)}(x'+w)}\right) \prod_{(i,j) \in E'} \mathsf{e}\left(a_{(i,j)_0}^{ro_{(i,j)}}, X_{0_0}^{(x'+i)(x'+j)\prod_{w \in f_{\mathcal{E}}(i,j)}(x'+w)}\right)$$
$$= \prod_{i \in V'} \mathsf{e}\left(W'_i, \prod_{k=0}^{|V'_i|} X_{0_k}^{\mathsf{m}_{i_k}}\right) \prod_{(i,j) \in E'} \mathsf{e}\left(W'_{(i,j)}, \prod_{k=0}^{|E'_{(i,j)}|} X_{0_k}^{\mathsf{m}_{(i,j)_k}}\right)$$

The set proof above gives rise to some diversification and we describe a few which we think are important for a graph signature scheme:

1. $\phi_{\mathsf{set}(V')}$: remove $\prod_{(i,j) \in E'} \mathsf{e}\left(W'_{(i,j)}, \left(\prod_{k=0}^{|E'_{(i,j)}|} X_{0_k}^{\mathsf{m}_{(i,j)_k}}\right)^{\varepsilon_{(i,j)}}\right)$.
2. $\phi_{\mathsf{set}(E')}$: remove $\prod_{i \in V'} \mathsf{e}\left(W'_i, \left(\prod_{k=0}^{|V'_i|} X_{0_k}^{\mathsf{m}_{0_k}}\right)^{\varepsilon_i}\right)$.

3. $\phi_{\mathsf{set}(A_i)}$: given labels $A_i \subset V_i'$, replace $\prod_{i \in V'} \mathsf{e}\left(W_i', \left(\prod_{k=0}^{|V_i'|} X_{0_k}^{\mathsf{m}_{0_k}}\right)^{\varepsilon_i}\right)$ with $\mathsf{e}\left(W_i', \left(\prod_{k=0}^{|A_i|} X_{0_k}^{\mathsf{m}_{i_k}}\right)^{\varepsilon_i}\right)$.
4. $\phi_{\mathsf{set}(A_{(i,j)})}$: given labels $A_{(i,j)} \subset E_{(i,j)}'$, replace $\prod_{(i,j) \in E'} \mathsf{e}\left(W_{(i,j)}', \left(\prod_{k=0}^{|E_{(i,j)}'|} X_{0_k}^{\mathsf{m}_{(i,j)_k}}\right)^{\varepsilon_{(i,j)}}\right)$ with $\mathsf{e}\left(W_{(i,j)}', \left(\prod_{k=0}^{|A_{(i,j)}|} X_{0_k}^{\mathsf{m}_{(i,j)_k}}\right)^{\varepsilon_{(i,j)}}\right)$.
5. Complex cases: given $\phi_{\mathsf{set}(i_1,\ldots,i_\ell)}$ and additionally asked to prove the connection among the vertex identifiers. This resembles the edge, connected and isolated statements which are elaborated in Section 7.

Lemma 5. *The randomization of $\mathcal{G}'$ in the* set *predicate is perfectly hiding.*

Proof. The blinding factors $o_i, o_{(i,j)}$ act as the opening values and turns the $\mathbb{G}_2$ elements into MoniPoly commitments:

$$\prod_{i \in V'} \mathsf{e}\left(a_{i_0}^r, \left(\prod_{k=0}^{|V_i'|} X_{0_k}^{\mathsf{m}_{i_k}}\right)^{o_i}\right) \prod_{(i,j) \in E'} \mathsf{e}\left(a_{(i,j)_0}^r, \left(\prod_{k=0}^{|E_{(i,j)}'|} X_{0_k}^{\mathsf{m}_{(i,j)_k}}\right)^{o_{(i,j)}}\right)$$

which are perfectly hiding. Moreover, the $\mathbb{G}_T$ elements also form a Pedersen set commitment:

$$\prod_{i \in V'} \mathsf{e}\left(a_{i_0}^r, \prod_{k=0}^{|V_i'|} X_{0_k}^{\mathsf{m}_{i_k}}\right)^{o_i} \prod_{(i,j) \in E'} \mathsf{e}\left(a_{(i,j)_0}^r, \prod_{k=0}^{|E_{(i,j)}'|} X_{0_k}^{\mathsf{m}_{(i,j)_k}}\right)^{o_{(i,j)}}$$

that is perfectly hiding according to Lemma 8. □

6.9 cover Proof

If a prover is requested to prove the knowledge of ℓ out of all vertices and edges in a set proof, it becomes a coverage proof cover. To be precise, when the verifier requests a show proof for the predicate $\phi_{\mathsf{cover}(\ell,\mathcal{G}')}$, the prover proves that part of $\mathcal{G}'$ is in $\mathcal{G}$ such that $|\mathcal{G}' \cap \mathcal{G}| \geq \ell$, without verifier knowing what $(\mathcal{G}' \cap \mathcal{G})$ is.

The protocol can be executed as follows:

$$PK\Big\{(\forall i \in V : \varepsilon_{i_0}, \varepsilon_{i_1}), (\forall (i,j) \in E : \varepsilon_{(i,j)_0}, \varepsilon_{(i,j)_1}), (\forall i \in V' - V : \varepsilon_{i_0}, \varepsilon_{i_1}),$$

$$(\forall (i,j) \in E' - E : \varepsilon_{(i,j)_0}, \varepsilon_{(i,j)_1}), \zeta, \rho, \omega, \tau, \gamma) :$$

$$\prod_{i \in (V' \cap V)} \mathsf{e}\left(W'_i, X_{0_1}^{\varepsilon_{i_1}} X_{0_0}^{\varepsilon_{i_0}}\right) \prod_{(i,j) \in (E' \cap E)} \mathsf{e}\left(W'_{(i,j)}, X_{0_1}^{\varepsilon_{(i,j)_1}} X_{0_0}^{\varepsilon_{(i,j)_0}}\right) \cdot$$

$$\prod_{i \in (V - V')} \mathsf{e}\left(W'_i, X_{0_1}^{\varepsilon_{i_1}} X_{0_0}^{\varepsilon_{i_0}}\right) \prod_{(i,j) \in (E - E')} \mathsf{e}\left(W'_{(i,j)}, X_{0_1}^{\varepsilon_{(i,j)_1}} X_{0_0}^{\varepsilon_{(i,j)_0}}\right) \cdot$$

$$\mathsf{e}\left(h^{\zeta} b^{\rho} c^{\omega} v'^{-\tau}, X_{0_0}\right) = \mathsf{e}(v'^{\gamma}, X) \wedge$$

$$\prod_{i \in V'} \mathsf{e}\left(W'_i, X_{0_1}^{\varepsilon_{i_1}} X_{0_0}^{\varepsilon_{i_0}}\right) \prod_{(i,j) \in E'} \mathsf{e}\left(W'_{(i,j)}, X_{0_1}^{\varepsilon_{(i,j)_1}} X_{0_0}^{\varepsilon_{(i,j)_0}}\right) =$$

$$\prod_{i \in V'} \mathsf{e}\left(W'_i, \prod_{k=0}^{|V'_i|} X_{0_k}^{\mathsf{m}_{i_k}}\right) \prod_{(i,j) \in E'} \mathsf{e}\left(W'_{(i,j)}, \prod_{k=0}^{|E'_{(i,j)}|} X_{0_k}^{\mathsf{m}_{(i,j)_k}}\right)\Big\}$$

where $v', \{W'_i, W'_{(i,j)}\}$ are the public inputs. The details for the exponents are $\{\mathsf{m}_{i_k}\} = \mathsf{MPEncode}(V'_i), \{\mathsf{m}_{(i,j)_k}\} = \mathsf{MPEncode}(E'_{(i,j)}), \zeta = r \times sk_{\mathsf{U}}, \rho = s', \omega = r, \varepsilon_{i_1} = r_i, \varepsilon_{(i,j)_1} = r_{(i,j)}, \tau = t', \gamma = y$ for the randomly chosen blinding factors $r, y, r_i, r_{(i,j)} \in \mathbb{Z}_p^*$. The same diversification for set proof is applicable to the cover proof.

Lemma 6. *The randomization of $\mathcal{G}'$ in the cover predicate is perfectly hiding if $\mathcal{G}' \in \mathcal{G}_{\mathsf{U}}$.*

Proof. The proof is similar to that of Lemma 5. □

6.10 disjoint Proof

When the verifier requests a show proof for the predicate $\phi_{\mathsf{disjoint}(\mathcal{G}')}$, i.e., showing a sub-graph $\mathcal{G}' = (V' \cup E')$ is not inside the graph $\mathcal{G}$, it is sufficient to show that the vertices in both graphs are disjoint such that $(V' \cap \mathcal{G}) = \emptyset$. In order to achieve this, the prover needs to separate the vertex sets from the edges set. This can be done by making use of the vertices predicate as well as the set membership proof for edge identifiers from the edges predicate. The latter is because proving a commitment has two vertex identifiers implicitly proves that it is an edge. The

$\mathsf{disjoint}$ predicate is as follows:

$$PK\Bigg\{((\forall i \in V : \varepsilon_{i_0}, \varepsilon_{i_1}), (\forall (i,j) \in E : \varepsilon_{(i,j)_0}, \varepsilon_{(i,j)_1}, \varepsilon_{(i,j)_2}), \zeta, \rho, \omega, \tau, \gamma, \{\alpha_k\}_{k=0}^{|V'|-1}) :$$

$$\prod_{i \in V} \mathsf{e}\left(W'_i, \prod_{k=0}^{1} X_{0_k}^{\varepsilon_{i_k}}\right) \prod_{(i,j)\in E} \mathsf{e}\left(W'_{(i,j)}, \prod_{k=0}^{2} X_{0_k}^{\varepsilon_{(i,j)_k}}\right) \cdot$$

$$\mathsf{e}\left(h^{\zeta} b^{\rho} c^{\omega} v'^{-\tau}, X_0\right) = \mathsf{e}(v'^{\gamma}, X) \wedge$$

$$\mathsf{vertices}(V) \wedge \varepsilon_{(i,j)}[C_{(i,j)_E}] \in \varXi_{\mathcal{V}} \wedge$$

$$\mathsf{e}\left(a_{0_0}, W_{\mathcal{V}_\ell}\right) = \mathsf{e}\left(W_V, \prod_{k=0}^{|V'|} X_{0_0}^{\mathsf{m}_k}\right) \mathsf{e}\left(\prod_{k=0}^{|V'|-1} a_{0_0}^{\alpha_k}, X_{0_k}\right) \wedge 1_{\mathbb{G}} \neq \prod_{k=0}^{|V'|-1} a_{0_0}^{\alpha_k} \Bigg\}$$

where $\{\mathsf{m}_k\} = \mathsf{MPEncode}(\forall \bar{i} \in V')$ and $W_{\mathcal{V}_\ell}$ is provided by $\mathsf{vertices}$.

Lemma 7. *The $\mathsf{disjoint}$ predicate is perfectly hiding.*

Proof. The proof is similar to that of Lemma 1 and 5. □

In Gross' graph signature scheme [15], the pool of bases are deliberately separated for vertices and edges. If we also apply the same setting, we get a more efficient $\mathsf{disjoint}$ predicate as follows:

$$PK\Bigg\{((\forall i \in V : \varepsilon_{i_0}, \varepsilon_{i_1}), (\forall (i,j) \in E : \varepsilon_{(i,j)_0}, \varepsilon_{(i,j)_1}), \zeta, \rho, \omega, \tau, \gamma, \{\alpha_k\}_{k=0}^{|V'|-1}) :$$

$$\prod_{i \in V} \mathsf{e}\left(W'_i, \prod_{k=0}^{1} X_{0_k}^{\varepsilon_{i_k}}\right) \prod_{(i,j)\in E} \mathsf{e}\left(W_{(i,j)}, \prod_{k=0}^{1} X_{(i,j)_k}^{\varepsilon_{(i,j)_k}}\right) \cdot$$

$$\mathsf{e}\left(h^{\zeta} b^{\rho} c^{\omega} v'^{-\tau}, X_0\right) = \mathsf{e}(v'^{\gamma}, X) \wedge$$

$$\mathsf{e}\left(\prod_{i \in V} W'_i, X_{0_0}\right) = \prod_{i \in V} \mathsf{e}\left(W_i, X_{i_0}\right) \wedge \varepsilon_i[C_i] \in \varXi_{\mathcal{V}} \wedge$$

$$\mathsf{e}\left(a_{0_0}, W_{\mathcal{V}_\ell}\right) = \mathsf{e}\left(W_V, \prod_{k=0}^{|V'|} X_{0_0}^{\mathsf{m}_k}\right) \mathsf{e}\left(\prod_{k=0}^{|V'|-1} a_{0_0}^{\alpha_k}, X_{0_k}\right) \wedge 1_{\mathbb{G}} \neq \prod_{k=0}^{|V'|-1} a_{0_0}^{\alpha_k} \Bigg\}$$

where $\{\mathsf{m}_k\} = \mathsf{MPEncode}(\forall \bar{i} \in V')$.

There is an alternate proof, namely, $\mathsf{disjoint}^*$ whose complexity is independent of total system vertices $|\varXi_{\mathcal{V}}|$. However, $\mathsf{disjoint}^*$ has to prove $|V'| \times |V|$ secret exponents in $\mathbb{G}_1$ instead of $|V'|$ as in $\mathsf{disjoint}$. In order to simplify the notation, we assume the vertex identifier $\bar{i}$, instead of the vertex label, for every $V'_{\bar{i}} \in V'$ does not appear in $\mathcal{G}$. Prover can interact with the verifier to construct the $\mathsf{disjoint}^*$

proof as follows:

$$PK\Bigg\{(\forall i \in V : \varepsilon_{i_0}, \varepsilon_{i_1}), (\forall (i,j) \in E : \varepsilon_{(i,j)_0}, \varepsilon_{(i,j)_1}), (\forall \bar{i} \in V' : \{\alpha_i\}_{i=1}^{|V|}),$$
$$\zeta, \rho, \omega, \tau, \gamma) :$$
$$\prod_{i \in V} \mathsf{e}\left(W_i, \prod_{k=0}^{1} X_{i_k}^{\varepsilon_{i_k}}\right) \prod_{(i,j)\in E} \mathsf{e}\left(W_{(i,j)}, \prod_{k=0}^{1} X_{(i,j)_k}^{\varepsilon_{(i,j)_k}}\right) \cdot$$
$$\mathsf{e}\left(h^{\zeta} b^{\rho} c^{\omega} v'^{-\tau}, X_{0_0}\right) = \mathsf{e}(v'^{\gamma}, X) \wedge$$
$$\forall \bar{i} \in V' :$$
$$\prod_{i \in V} \mathsf{e}\left(W_i, \prod_{k=0}^{1} X_{i_k}^{\varepsilon_{i_k}}\right) =$$
$$\mathsf{e}\left(\prod_{i\in V} W_{\bar{i}}, X_{0_1} X_{0_0}^{\bar{i}}\right) \mathsf{e}\left(\prod_{i\in V} a_{i_0}^{\alpha_i}, X_{0_0}\right) \wedge 1_{\mathbb{G}} \neq a_{i_0}^{\alpha_i}\Bigg\}$$

where $v', \{W_i, W_{(i,j)}\}$ are the public inputs and $\zeta = r \times sk_{\mathsf{U}}, \rho = s', \omega = r, \varepsilon_{i_1} = r_i, \varepsilon_{(i,j)_1} = r_{(i,j)}, \tau = t'$ and $\gamma = y$ for the randomly chosen blinding factors $r, y, r_i, r_{(i,j)} \in \mathbb{Z}_p^*$. The correctness for the second statement is as follows:

$$\begin{aligned}
\mathsf{e}\left(\prod_{i\in V} C_i, X_{0_0}\right) &= \mathsf{e}\left(\prod_{i\in V} a_{i_0}^{ro_i \prod_{k\in V_i}(x'+k)}, X_{0_0}\right) \\
&= \mathsf{e}\left(\prod_{i\in V} a_{i_0}^{q_i(x')(x'+\bar{i})+\bar{r}_i}, X_{0_0}\right)^{ro_i} \\
&= \mathsf{e}\left(\prod_{i\in V} a_{i_0}^{q_i(x')}, X_{0_0}^{x'+\bar{i}}\right)^{ro_i} \mathsf{e}\left(\prod_{i\in V} a_{i_0}^{\bar{r}_i}, X_{0_0}\right)^{ro_i} \\
&= \mathsf{e}\left(\prod_{i\in V} a_{i_0}^{ro_i q_i(x')}, X_{0_0}^{x'+\bar{i}}\right) \mathsf{e}\left(\prod_{i\in V} a_{i_0}^{ro_i \bar{r}_i}, X_{0_0}\right) \\
&= \mathsf{e}\left(\prod_{i\in V} W_i, X_1 X_0^{\bar{i}}\right) \mathsf{e}\left(\prod_{i\in V} a_{i_0}^{\alpha_i}, X_{0_0}\right)
\end{aligned}$$

where $q_i(x')$ is the quotient polynomial and $\alpha_i = \bar{r}_i$ is the remainder. Note that the second last statement should not be stacked together or it conflicts with the last statement. When the verifier queries $\phi_{\mathsf{disjoint}}(E')$ only, the prover can modify the last statement accordingly to prove $(E' \cap \mathcal{G}) = \emptyset$. Note that $\prod_{i\in V} a_{i_0}^{\alpha_i}$ in the second statement is a Pedersen set commitment and Lemma 7 still holds. For the completeness of this argument, we prove the perfectly hiding and binding properties as below.

Lemma 8. *The Pedersen set commitment is perfectly hiding.*

Proof. Given a Pedersen set commitment $C = a_{1_0}^{\alpha_1} a_{2_0}^{\alpha_2} \cdots a_{L_0}^{\alpha_L}$, there are $|\mathbb{Z}_p^*| - 1$ possible pairs of $(\alpha_1, (\alpha_2, \ldots, \alpha_L)) \neq (\alpha'_1, (\alpha'_2, \ldots, \alpha'_L))$ which can result in the same C. Furthermore, for each committed set $(\alpha_2, \ldots, \alpha_L)$, there is a unique α_1 such that:

$$\mathsf{dlog}_{a_{1_0}}(C) = \alpha_1 \prod_{j=2}^{L} \mathsf{dlog}_{a_{1_0}}(a_{j_0})\alpha_j \mod p$$

$$\alpha_1 = \frac{\mathsf{dlog}_{a_{1_0}}(C)}{\prod_{j=2}^{L} \mathsf{dlog}_{a_{1_0}}(a_{j_0})\alpha_j} \mod p$$

Since α_1 is calculated independently of the committed set $(\alpha_2, \ldots, \alpha_L)$, the latter is perfectly hidden. □

Lemma 9. *The Pedersen set commitment is binding if the DLOG problem is hard.*

Proof. Given a DLOG instance $(g, h = g^x)$, we construct a challenger $\mathcal{C}$ that runs the adversary $\mathcal{A}$ of a Pedersen set commitment scheme as the sub-routine to find the solution x. $\mathcal{C}$ sets $a_{1_0} = g$ and $\{a_{i_0} = h^{b_i}\}_{i=1}^{L}$ for randomly chosen $b_i \in \mathbb{Z}_p^*$. $\mathcal{C}$ publishes $\{a_{i_0}\}_{i=0}^{L}$ as the public parameters for the Pedersen set commitment scheme.

If an adversary can output a Pedersen set commitment C for two different sets $(A = \{\alpha_i\}_{i=0}^{L}, A^* = \{\alpha_i^*\}_{i=0}^{L})$ such that $|A \cap A^*| \geq 2$ and:

$$C = a_{1_0}^{\alpha_1} a_{2_0}^{\alpha_2} \cdots a_{L_0}^{\alpha_L} = a_{1_0}^{\alpha_1^*} a_{2_0}^{\alpha_2^*} \cdots a_{L_0}^{\alpha_L^*}$$

$$\Leftrightarrow g^{\alpha_1} h^{\sum_{i=1}^{L} b_i \alpha_i} = g^{\alpha_1^*} h^{\sum_{i=1}^{L} b_i \alpha_i^*},$$

$\mathcal{C}$ can compute:

$$x = \frac{\alpha_1 - \alpha_1^*}{\sum_{i=1}^{L} b_i(\alpha_i^* - \alpha_i)} \mod p$$

to solve the DLOG problem. □

From Table 3, we observe that the complexity of the $\mathsf{disjoint}^*$ predicate is faster than that of the $\mathsf{disjoint}$ predicate only when n' is small. Let the complexity ratio [24] at 128-bit security for a scalar multiplication in $\mathbb{G}_1$ (M_1) to a scalar multiplication in $\mathbb{G}_2$, exponentiation in $\mathbb{G}_T$ and pairing be 2, 6 and 9, respectively. Assuming $l = 1$, the $\mathsf{disjoint}$ predicate has a total of $(29m + n^2/2 + (127n)/2 + 2|\Xi_{\mathcal{V}}| + 5n' + 89)M_1$ while $\mathsf{disjoint}^*$ has a total of $(30m + (5n+32)n' + 29n + 27)M_1$. Setting $n = 100, m = 1000$ and $\Xi_{\mathcal{V}} = 1000$, we get $(5n' + 42439)M_1$ for $\mathsf{disjoint}$ and $(532n' + 8927)M_1$ for $\mathsf{disjoint}^*$. As a result, as long as $n' = |V'| \geq 18$, $\mathsf{disjoint}^*$ is slower than $\mathsf{disjoint}$. On the other hand, at $n = 1000, m = 100$, we get $(5n' + 589389)M_1$ for $\mathsf{disjoint}$ and $(5032n' + 35027)M_1$ for $\mathsf{disjoint}^*$ which pushes the threshold to $n' = |V'| \geq 106$. This shows that disjoint is suitable for graphs with little vertices but large edges while disjoint* is the other way round. X3

6.11 Security

6.11.1 Impersonation Resilience. We establish the security of the graph signature scheme by constructing a reduction to the (co-)SDH problem. To achieve tight security reduction, we make use of Multi-Instance Reset Lemma [17] as the knowledge extractor which requires the adversary $\mathcal{A}$ to run N parallel instances of impersonation under active and concurrent attacks. The challenger $\mathcal{C}$ can fulfill this requirement by simulating the $N-1$ instances from its given SDH instance which is random self-reducible [4]. Since this is obvious, we describe only the simulation for a single instance of impersonation under active and concurrent attacks in the security proofs.

Theorem 5. *If an adversary $\mathcal{A}$ $(t_{\mathsf{imp}}, \varepsilon_{\mathsf{imp}})$-breaks the imp-aca-security of the proposed graph signature scheme, then there exists an algorithm $\mathcal{C}$ which $(t_{\mathsf{cosdh}}, \varepsilon_{\mathsf{cosdh}})$-breaks the co-SDH problem such that:*

$$\frac{\varepsilon_{\mathsf{cosdh}}}{t_{\mathsf{cosdh}}} = \frac{\varepsilon_{\mathsf{imp}}}{t_{\mathsf{imp}}},$$

or an algorithm $\mathcal{C}$ which $(t_{\mathsf{sdh}}, \varepsilon_{\mathsf{sdh}})$-breaks the SDH problem such that:

$$\begin{aligned} \varepsilon_{\mathsf{imp}} &\leq \sqrt[N]{\sqrt{\varepsilon_{\mathsf{sdh}}} - 1} + \frac{1 + (q-1)!/p^{q-2}}{p} + 1, \\ t_{\mathsf{imp}} &\leq t_{\mathsf{sdh}}/2N - T(q^2). \end{aligned}$$

where N is the total adversary instance, $q = Q_{(O,S)} + Q_{(P,V)}$ is the total query made to the Obtain *and* Verify *oracles, while $T(q^2)$ is the time parameterized by q to setup the simulation environment and to extract the SDH solution. Consider the dominant time elements t_{imp} and t_{sdh} only, we have:*

$$\left(1 - \left(1 - \varepsilon_{\mathsf{imp}} + \frac{1 + (q-1)!/p^{q-2}}{p}\right)^N\right)^2 \leq \varepsilon_{\mathsf{sdh}}, 2Nt_{\mathsf{imp}} \approx t_{\mathsf{sdh}}.$$

Let $N = (\varepsilon_{\mathsf{imp}} - \frac{1+(q-1)!/p^{q-2}}{p})^{-1}$, we get $\varepsilon_{\mathsf{sdh}} \geq (1-e^{-1})^2 \geq 1/3$ and the success ratio is:

$$\begin{aligned} \frac{\varepsilon_{\mathsf{sdh}}}{t_{\mathsf{sdh}}} &\geq \frac{1}{3 \cdot 2Nt_{\mathsf{imp}}} \\ \frac{6\varepsilon_{\mathsf{sdh}}}{t_{\mathsf{sdh}}} &\geq \frac{\varepsilon_{\mathsf{imp}}}{t_{\mathsf{imp}}} - \frac{1 + (q-1)!/p^{q-2}}{t_{\mathsf{imp}}p} \end{aligned}$$

which gives a tight reduction.

Similar to the approach in the security proofs for MoniPoly graph signature scheme [24], we categorize the way an adversary impersonates in Table 2.

X3 **XXX TODO:** Why do we need this forward reference to the complexity analysis here? It seems out of place and might confuse the readers.

Table 2. Types of impersonation and the corresponding assumptions.

Type	$\mathcal{G}$	$\mathsf{MPEncode}(\mathcal{G})$	s t v	**Adversary**	**Assumption**	**Lemmas**
0	0	1	* * *	$\mathcal{A}_{bind}$	co-DLOG	Theorem 3
1	0	0	0 0 0	$\mathcal{A}_1$	SDH	1
2	0	0	0 0 1	$\mathcal{A}_1$	DLOG	1
3	0	0	0 1 0	$\mathcal{A}_2$	SDH	2
4	0	0	0 1 1	$\mathcal{A}_2$	DLOG	2
5	0	0	1 0 0	$\mathcal{A}_1$	SDH	1
6	0	0	1 0 1	$\mathcal{A}_1$	DLOG	1
7	0	0	1 1 0	$\mathcal{A}_3$	SDH	3
8	0	0	1 1 1	$\mathcal{A}_3$	DLOG	3
9	1	1	0 0 0	$\mathcal{A}_1$	SDH	1
10	1	1	0 0 1	$\mathcal{A}_1$	DLOG	1
11	1	1	0 1 0	$\mathcal{A}_2$	SDH	2
12	1	1	0 1 1	$\mathcal{A}_2$	DLOG	2
13	1	1	1 0 0	$\mathcal{A}_1$	SDH	1
14	1	1	1 0 1	$\mathcal{A}_1$	N/A	1
15	1	1	1 1 0	$\mathcal{A}_3$	SDH	3
16	1	1	1 1 1	$\mathcal{A}_3$	N/A	3

Note: * = 1 or 0, 1 = queried, 0 = not queried, N/A = not available

We present Lemma 10, 11 and 12 corresponding to the adversaries $\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{A}_3$ as follows.

Lemma 10. *If an adversary $\mathcal{A}_1$ $(t_{\mathsf{imp}}, \varepsilon_{\mathsf{imp}})$-breaks the $\mathsf{imp\text{-}aca}$-security of the proposed graph signature scheme, then there exists an algorithm $\mathcal{C}$ which $(t_{\mathsf{sdh}}, \varepsilon_{\mathsf{sdh}})$-solves the SDH problem such that:*

$$\varepsilon_{\mathsf{imp}} \leq \sqrt[N]{\sqrt{\varepsilon_{\mathsf{sdh}}} - 1} + \frac{1 + (q-1)!/p^{q-2}}{p} + 1,$$

$$t_{\mathsf{imp}} \leq t_{\mathsf{sdh}}/2N - T(q^2).$$

where N is the total of adversary instances, $q = Q_{(O,S)} + Q_{(P,V)}$ is the number of queries made to the Obtain and Verify oracles, while $T(q^2)$ is the time parameterized by q to setup the simulation environment and to extract the SDH solution.

Proof. Given a q-SDH instance $(g_1, g_1^x, g_1^{x^2}, \ldots, g_1^{x^q}, g_2, g_2^x)$ where $q = Q_{(O,S)} + Q_{(P,V)}$ is the maximum number of queries $\mathcal{A}_1$ can issue to the Obtain and Verify oracles, we show that if $\mathcal{A}_1$ exists, there exists an algorithm $\mathcal{C}$ which can output $(g_1^{\frac{1}{x+t}}, t)$ by acting as the simulator for the graph signature scheme as follows:

Game$_0$. This is the attack by $\mathcal{A}_1$ on the real N instances of graph signature scheme. Let S be the event of a successful impersonation, by assumption, we have:

$$\Pr[S_0] = \varepsilon_{\mathsf{imp}}. \quad (1)$$

Game$_1$. In order to simulate the environment of the graph signature scheme, $\mathcal{C}$ uniformly and randomly selects distinct $t_a, \{t_{a_i}\}_{i=1}^{L}, t_b, t_c, x', t_1, \ldots, t_q \in \mathbb{Z}_p^*$. Next, let $f(x)$ denotes the polynomial $f(x) = \prod_{k=1}^{q}(x + t_k) = \sum_{k=0}^{q} \rho_k x^k$ and $f_u(x)$ denotes the polynomial $f_u(x) = \prod_{k=1,k\neq u}^{q}(x + t_k) = \sum_{k=0}^{q-1} \lambda_k x^k$. Let $g_1^{f(x)} = \prod_{k=0}^{q}(g_1^{x^k})^{\rho_k}$, $\mathcal{C}$ sends $(\mathsf{e}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, \{a_{0_k} = g_1^{f(x)t_a x'^k}, X_{0_k} = g_2^{x'^k}\}_{k=0}^{n}, \{\{a_{i_k}, X_{i_k}\}_{i=1}^{L}\}_{k=0}^{n}, b = g_1^{f(x)t_b}, c = g_1^{f(x)t_c}, d = g_1^{f(x)t_d}, h = g_1^{f(x)t_h}, X = g_2^{x})$ as the public key to $\mathcal{A}_1$. $\mathcal{C}$ creates two empty lists $L_{(FO,FS)}$ and $L_{(P,V)}$ where the former stores the corrupted final signatures simulated during the final signing protocol while the latter stores the non-corrupted signatures simulated during the showing protocol. $\mathcal{C}$ also creates two empty lists $L_{(iO,iS)}, L_{(IO,IS)}$ to store the corrupted initial signatures and intermediary signatures simulated during the initial and intermediary signing protocols, respectively. Since $t_a, t_b, t_c, t_d, t_h, x'$ are uniformly random, the distribution of the simulated public key (and the corresponding random self-reducible [4] $N-1$ instances) is the same as that of the original scheme. So, we have:

$$\Pr[S_1] = \Pr[S_0]. \tag{2}$$

Game$_2$. In this game, $\mathcal{A}_1$ plays the role of multiple users to concurrently interact with the initial signer simulated by $\mathcal{C}$. Without loss of generality, we assume every user u uses different data set $A_u = \{sk_{\mathsf{U}_u}, s'_u, \perp, \perp\}$. If the u-th session of an initial signing protocol ends successfully, $\mathcal{C}$ produces a signature σ_{init} for $\mathcal{A}_1$. Their interaction is as follows:

1. $\mathcal{A}_1$ concurrently initializes the initial signing protocol with $\mathcal{C}$ by running the zero-knowledge protocol:

$$PK\{(\zeta, \rho) : C = h^{\zeta} b^{\rho}\}$$

   Without loss of generality, we assume $\mathcal{A}_1$ always execute this protocol honestly. Therefore, $\mathcal{C}$ always reset successfully and can extract the secret exponents $\zeta = sk_{\mathsf{U}_u}, \rho = s'_u$ used by $\mathcal{A}_1$ in the protocol above.
2. $\mathcal{C}$ chooses a random value $s''_u \in \mathbb{Z}_p^*$ and a graph $\mathcal{G}_{\mathsf{S}_u} = \{V_{\mathsf{S}_u} \cup E_{\mathsf{S}_u}\}$ to set:

$$v_u = h_u^{sk_{\mathsf{U}_u}} \prod_{i \in V_{\mathsf{S}_u}} \prod_{k=0}^{n} a_{u,i_k}^{\mathsf{m}_{u,i_k}} \prod_{(i,j) \in E_{\mathsf{S}_u}} \prod_{k=0}^{n} a_{u,(i,j)_k}^{\mathsf{m}_{u,(i,j)_k}} b_u^{s_u + s''_u} d_u$$

   where $a_{u,i_k} = g_1^{f_u(x)t_{a_i}x'^k}, b_u = g_1^{f_u(x)t_b}, d_u = g_1^{f_u(x)t_d}, \{\mathsf{m}_{u,i_k}\} = \mathsf{MPEncode}(i \in V_{\mathsf{S}_u})$ and $\{\mathsf{m}_{u,(i,j)_k}\} = \mathsf{MPEncode}((i,j) \in E_{\mathsf{S}_u})$. $\mathcal{C}$ adds the record $(t_u, s_u = s_u r_{u,1} + s''_u, v_u, A_u, \mathcal{G}_{\mathsf{S}_u})$ to $L_{(iO,iS)}$ and returns $\sigma'_u = (t_u, s''_u, v_u, \mathcal{G}_{\mathsf{S}_u})$ as the signature to $\mathcal{A}_1$.

Since $\mathcal{C}$'s choices of t_u, s''_u are independent of $\mathcal{A}$'s view, a collision $v_i = v_j$ for some $i, j \leq q$ in $\mathcal{A}$'s concurrent queries happens with a negligible probability of $\Pr[Col] = 1/p$ in which $\mathcal{A}_1$ can compute the discrete logarithm x. Else, $\mathcal{C}$ simulates the $\mathsf{InitSign}$ oracle perfectly for every concurrent query and $\mathcal{A}_1$ can formulate its signature $\sigma_{init_u} = (t_u, s_u = s'_u + s''_u, v_u, \mathcal{G}_{\mathsf{U}_u} = \mathcal{G}_{\mathsf{S}_u})$ as in the original initial signing protocol. This gives:

$$\begin{aligned}\Pr[S_2] &= \Pr[S_1] + \Pr[Col] \\ &\leq \Pr[S_1] + \prod_{i=1}^{q-1} i/p \\ &\leq \Pr[S_1] + (q-1)!/p^{q-1} \qquad (3)\end{aligned}$$

where $\mathcal{A}_1$ can make, at most, another $q-1$ initial signature queries.

Game$_3$. In this game, $\mathcal{A}_1$ plays the role of multiple users to concurrently interact with the intermediary signer simulated by $\mathcal{C}$. Without loss of generality, we assume every user u has different graph $A_u = \{sk_{\mathsf{U}_u}, s'_u, \mathcal{G}_{\mathsf{U}_u}, \{o_{u,i_1}, o_{u,(i,j)_1}\}\}$. If the u-th session of an signing protocol ends successfully, $\mathcal{C}$ produces a signature σ_{interm} for $\mathcal{A}_1$. Their interaction is as follows:

1. $\mathcal{A}_1$ concurrently initializes the signing protocol with $\mathcal{C}$ by running the zero-knowledge protocol:

$$PK\Big\{((\forall i \in V_{\mathsf{U}_u} : \varepsilon_{i_0}, \varepsilon_{i_1}, \varepsilon_i), (\forall (i,j) \in E_{\mathsf{U}_u} : \varepsilon_{(i,j)_0}, \varepsilon_{(i,j)_1}, \varepsilon_{(i,j)}), \zeta, \rho, \omega, \tau, \gamma) :$$
$$\mathsf{e}\left(C_1 C_2 d^{\omega} v'^{-\tau}, X_{0_0}\right) = \mathsf{e}(v'^{\gamma}, X) \ \wedge C_1 C_2 = h^{\zeta} b^{\rho} \prod_{i \in V_{\mathsf{U}_u}} C_i^{\varepsilon_i} \prod_{(i,j) \in E_{\mathsf{U}_u}} C_{(i,j)}^{\varepsilon_{(i,j)}} \wedge$$
$$\mathsf{e}\left(\prod_{i \in V_{\mathsf{U}_u}} C_i \prod_{(i,j) \in E_{\mathsf{U}_u}} C_{(i,j)}, X_{0_0}\right) =$$
$$\prod_{i \in V_{\mathsf{U}_u}} \mathsf{e}\left(W'_i, \prod_{k=0}^{1} X_{0_k}^{\varepsilon_{i_k}}\right) \prod_{(i,j) \in E_{\mathsf{U}_u}} \mathsf{e}\left(W'_{(i,j)}, \prod_{k=0}^{1} X_{0_k}^{\varepsilon_{(i,j)_k}}\right)\Big\}$$

   Without loss of generality, we assume $\mathcal{A}_1$ always execute this protocol honestly. Therefore, $\mathcal{C}$ always reset successfully and can extract the secret exponents $\zeta = sk_{\mathsf{U}_u} r_{u,1}, \rho = s_u r_{u,1}, \omega = r_{u,1}, \tau = t'_u, \gamma = y_u, \varepsilon_{i_1} = r_{u,i}, \varepsilon_{i_0} = r_{u,i} i_u, \varepsilon_i = r_{u,1} o_{u,i_1}^{-1}, \varepsilon_{(i,j)_1} = r_{u,(i,j)}, \varepsilon_{(i,j)_0} = r_{u,(i,j)} i_u, \varepsilon_{(i,j)} = r_{u,1} o_{u,(i,j)_1}^{-1}$ used by $\mathcal{A}_1$ in the protocol above.
2. Using $t_u = t'_u / y_u$ or $s_u = s'_u / r_{u,1}$, $\mathcal{C}$ can search in $L_{(iO,iS)}$ for the $\mathcal{G}_{\mathsf{U}_u}$ of this intermediary signature. Next, $\mathcal{C}$ chooses a random value $s''_u \in \mathbb{Z}_p^*$ and a graph $\mathcal{G}_{\mathsf{S}_u} = \{V_{\mathsf{S}_u} \cup E_{\mathsf{S}_u}\}$ to set:

$$v_u = h_u^{sk_{\mathsf{U}_u} r_{u,1}} \prod_{i \in V_{\mathsf{U}_u}} \left(\prod_{k=0}^{n} a_{u,i_k}^{\mathsf{m}_{u,i_k}}\right)^{o_{u,i_1}} \prod_{(i,j) \in E_{\mathsf{U}_u}} \left(\prod_{k=0}^{n} a_{u,(i,j)_k}^{\mathsf{m}_{u,(i,j)_k}}\right)^{o_{u,(i,j)_1}} \cdot$$
$$\prod_{i \in V_{\mathsf{S}_u}} \prod_{k=0}^{n} a_{u,i_k}^{\mathsf{m}_{u,i_k}} \prod_{(i,j) \in E_{\mathsf{S}_u}} \prod_{k=0}^{n} a_{u,(i,j)_k}^{\mathsf{m}_{u,(i,j)_k}} b_u^{s_u r_{u,1} + s''_u} d_u$$

where $a_{u,i_k} = g_1^{f_u(x)t_{a_i}x'^k}, b_u = g_1^{f_u(x)t_b}, d_u = g_1^{f_u(x)t_d}$. $\mathcal{C}$ adds the record $(t_u, s_u = s_u r_{u,1} + s''_u, v_u, A_u, \mathcal{G}_{\mathsf{S}_u})$ to $L_{(IO,IS)}$ and returns $\sigma'_u = (t_u, s''_u, v_u, \mathcal{G}_{\mathsf{S}_u})$ as the signature to $\mathcal{A}_1$.

Since $\mathcal{C}$'s choices of t_u, s''_u are independent of $\mathcal{A}$'s view, a collision $v_i = v_j$ for some $i, j \le q$ in $\mathcal{A}$'s concurrent queries happens with a negligible probability of $\Pr[Col] = 1/p$ in which $\mathcal{A}_1$ can compute the discrete logarithm x. Else, $\mathcal{C}$ simulates the $\mathsf{IntermSign}$ oracle perfectly for every concurrent query and $\mathcal{A}_1$ can formulate its signature $\sigma_{interm_u} = (t_u, s_u = s_u r_{u,1} + s''_u, v_u, \mathcal{G}_u = \mathcal{G}_{\mathsf{U}_u} + \mathcal{G}_{\mathsf{S}_u}, \{o_{u,i_1}, o_{u,(i,j)_1}\})$ as in the original signing protocol. This gives:

$$\begin{aligned}\Pr[S_3] &= \Pr[S_2] + \Pr[Col] \\ &\le \Pr[S_2] + \prod_{i=1}^{q-2} i/p \\ &\le \Pr[S_2] + (q-2)!/p^{q-2} \end{aligned} \tag{4}$$

where $\mathcal{A}_1$ can make at most, another, $q-2$ intermediary signature queries. The initial signature will not collide with intermediary signature because $\mathcal{C}$ can always avoid this by choosing another s''_u.

Game$_4$. In this game, $\mathcal{A}_1$ plays the role of multiple users to concurrently interact with the final signer simulated by $\mathcal{C}$. Without loss of generality, we assume every user u has different graph $A_u = \{sk_{\mathsf{U}_u}, s'_u, \mathcal{G}_u, \{o_{u,i}, o_{u,(i,j)}\}\}$. If the u-th session of an signing protocol ends successfully, $\mathcal{C}$ produces a graph signature σ for $\mathcal{A}_1$. Their interaction is as follows:

1. $\mathcal{A}_1$ concurrently initializes the signing protocol with $\mathcal{C}$ by running the zero-knowledge protocol:

$$\begin{aligned} PK\Big\{ &((\forall i \in V_u : \varepsilon_{i_0}, \varepsilon_{i_1}, \varepsilon_i), (\forall (i,j) \in E_u : \varepsilon_{(i,j)_0}, \varepsilon_{(i,j)_1}, \varepsilon_{(i,j)}), \zeta, \rho, \omega, \tau, \gamma) : \\ &\mathsf{e}\left(C_1 C_2 d^{\omega} v'^{-\tau}, X_{0_0}\right) = \mathsf{e}(v'^{\gamma}, X) \wedge \\ &C_1 C_2 = h^{\zeta} b^{\rho} \prod_{i \in V_u} C_i^{\varepsilon_i} \prod_{(i,j)\in E_u} C_{(i,j)}^{\varepsilon_{(i,j)}} \wedge \\ &\mathsf{e}\left(\prod_{i\in V_u} C_i \prod_{(i,j)\in E_u} C_{(i,j)}, X_{0_0}\right) = \\ &\qquad \prod_{i\in V_u} \mathsf{e}\left(W'_i, \prod_{k=0}^{1} X_{0_k}^{\varepsilon_{i_k}}\right) \prod_{(i,j)\in E_u} \mathsf{e}\left(W'_{(i,j)}, \prod_{k=0}^{1} X_{0_k}^{\varepsilon_{(i,j)_k}}\right) \Big\} \end{aligned}$$

   Without loss of generality, we assume $\mathcal{A}_1$ always execute this protocol honestly. Therefore, $\mathcal{C}$ always reset successfully and can extract the secret exponents $\zeta = sk_{\mathsf{U}_u} r_{u,2}, \rho = s'_u, \omega = r_{u,2}, \tau = t'_u, \gamma = y_u, \varepsilon_{i_1} = r_{u,i}, \varepsilon_{i_0} = r_{u,i} i_u, \varepsilon_i = r_{u,2} o_{u,i_2}^{-1}, \varepsilon_{(i,j)_1} = r_{u,(i,j)}, \varepsilon_{(i,j)_0} = r_{u,(i,j)} i_u, \varepsilon_{(i,j)} = r_{u,2} o_{u,(i,j)_2}^{-1}$ used by $\mathcal{A}_1$ in the protocol above.

2. Using $t_u = t'_u/y_u$ or $s_u = s'_u/r_{u,2}$, $\mathcal{C}$ can search in $L_{(IO,IS)}$ for $\{\mathcal{G}_u, o_{u,i_1}, o_{u,(i,j)_1}\}$ of this final signature. Next, $\mathcal{C}$ chooses a random value $s''_u \in \mathbb{Z}_p^*$ to set:

$$v_u = h_u^{sk_{\mathsf{U}_u} r_{u,1}} \prod_{i \in V_{\mathsf{U}_u}} \left(\prod_{k=0}^{n} a_{u,i_k}^{\mathsf{m}_{u,i_k}} \right)^{o_{u,i_1} o_{u,i_2}} \prod_{(i,j) \in E_{\mathsf{U}_u}} \left(\prod_{k=0}^{n} a_{u,(i,j)_k}^{\mathsf{m}_{u,(i,j)_k}} \right)^{o_{u,(i,j)_1} o_{u,(i,j)_2}} \cdot \prod_{i \in V_{\mathsf{S}_u}} \left(\prod_{k=0}^{n} a_{u,i_k}^{\mathsf{m}_{u,i_k}} \right)^{o_{u,i_2}} \prod_{(i,j) \in E_{\mathsf{S}_u}} \left(\prod_{k=0}^{n} a_{u,(i,j)_k}^{\mathsf{m}_{u,(i,j)_k}} \right)^{o_{u,(i,j)_2}} b_u^{s'_u + s''_u} c_u$$

where $a_{u,i_k} = g_1^{f_u(x) t_{a_i} x'^k}, b_u = g_1^{f_u(x) t_b}, c_u = g_1^{f_u(x) t_c}$. If $(t_u, s_u, v_u, A_u) \in L_{(P,V)}$, $\mathcal{C}$ removes it from $L_{(P,V)}$. $\mathcal{C}$ adds the record to $L_{(FO,FS)}$. $\mathcal{C}$ returns $\sigma'_u = (t_u, s''_u, v_u)$ as the signature to $\mathcal{A}_1$.

Since $\mathcal{C}$'s choices of t_u, s''_u are independent of $\mathcal{A}$'s view, a collision $v_i = v_j$ for some $i, j \leq q$ in $\mathcal{A}$'s concurrent queries happens with a negligible probability of $\Pr[Col] = 1/p$ in which $\mathcal{A}_1$ can compute the discrete logarithm x. Else, $\mathcal{C}$ simulates the FinalSign oracle perfectly for every concurrent query and $\mathcal{A}_1$ can formulate its signature $\sigma_u = (t_u, s_u = s_u r_{u,2} + s''_u, v_u, \mathcal{G}_u, \{o_{u,i_1} o_{u,i_2}, o_{u,(i,j)_1} o_{u,(i,j)_2}\}, \{o_{u,i_2}, o_{u,(i,j)_2}\})$ as in the original signing protocol. This gives:

$$\begin{aligned} \Pr[S_4] &= \Pr[S_3] + \Pr[Col] \\ &\leq \Pr[S_3] + \prod_{i=1}^{q-2} i/p \\ &\leq \Pr[S_3] + (q-2)!/p^{q-2}. \end{aligned} \tag{5}$$

where $\mathcal{A}_1$ can make, at most, another $q-2$ final signature queries. The final signature will not collide with the initial signature because they use different bases c and d, respectively.

Game$_5$. In this game, $\mathcal{A}_1$ plays the role of multiple provers to concurrently interact with the verifier simulated by $\mathcal{C}$. Without loss of generality, we assume every prover u uses a valid σ_u to run its show proof on ϕ_{stmt_u} such that $\phi_{\mathsf{stmt}_u}(A_u) = 1$. $\mathcal{C}$ always simulates the Verify oracle correctly and this gives:

$$\Pr[S_5] = \Pr[S_4]. \tag{6}$$

Game$_6$. In this game, $\mathcal{A}_1$ plays the role of verifier to concurrently interact with multiple provers simulated by $\mathcal{C}$. When $\mathcal{A}_1$ asks for a show proof on ϕ_{stmt_u}, $\mathcal{C}$ interacts with $\mathcal{A}_1$ using a σ_u such that $\phi_{\mathsf{stmt}_u}(A_u) = 1$. We assume $\mathcal{C}$ already has the appropriate signatures on his hand for these queries. Else, $\mathcal{C}$ simulates a valid σ_u as in the previous games and adds it to $L_{(P,V)}$ before interacting with $\mathcal{A}_1$. This gives:

$$\Pr[S_6] = \Pr[S_5]. \tag{7}$$

Game$_7$. In this game, $\mathcal{A}_1$ wants to impersonate the prover whose data set is $A^* \neq A_i \in L_{(FO,FS)}$ using the predicate ϕ^*_{stmt} such that $\phi^*_{\mathsf{stmt}}(A^*) = 1$ and $\phi^*_{\mathsf{stmt}}(A_i) = 0$. $\mathcal{A}_1$ is still allowed to query the oracles as in previous games but with the restriction $\phi^*_{\mathsf{stmt}}(A_i) \neq 1$ for A_i to the Obtain oracle. Finally, if $\mathcal{A}_1$ completes a show proof for A^* such that $(\mathcal{A}_1^{\mathsf{Prove}}(pk, \cdot, \phi^*_{\mathsf{stmt}}(A^*)), \mathcal{C}^{\mathsf{Verify}}(pk, \phi^*_{\mathsf{stmt}}(A^*))) = 1$, $\mathcal{C}$ resets $\mathcal{A}_1$ to the time where it has just sent the witnesses. If the show is proof verified again, $\mathcal{C}$ can obtain two valid transcripts and recover the secret exponents to extract the signature elements (t^*, s^*, v^*).

Since $\mathcal{A}_1$ must output $t^* \notin \{t_1, \ldots, t_q\}$, if $v^* \notin L_{(FO,FS)} \cup L_{(P,V)}$, $\mathcal{C}$ can construct a polynomial $c(x)$ of degree $n-1$ such that $f(x) = c(x)(x+t^*) + r$ to compute:

$$
\begin{aligned}
&v^{*\,1/(t_h sk_{\mathsf{U}^*} + \sum_{i\in V}(o^*_i t_{a_i} \sum_{k=0}^{n} \mathsf{m}^*_{i_k} x'^k) + o^*_{(i,j)} \sum_{(i,j)\in E}(t_{a_{(i,j)}} \sum_{k=0}^{n} \mathsf{m}^*_{(i,j)_k} x'^k) + t_b s^* + t_c) r}\, g_1^{-\frac{c(x)}{r}} \\
&= g_1^{\frac{((t_h sk_{\mathsf{U}^*} + o^*_i \sum_{i\in V}(t_{a_i} \sum_{k=0}^{n} \mathsf{m}^*_{i_k} x'^k) + \sum_{(i,j)\in E}(o^*_{(i,j)} t_{a_{(i,j)}} \sum_{k=0}^{n} \mathsf{m}^*_{(i,j)_k} x'^k) + t_b s^* + t_c) f(x)}{((t_h sk_{\mathsf{U}^*} + \sum_{i\in V}(o^*_i t_{a_i} \sum_{k=0}^{n} \mathsf{m}^*_{i_k} x'^k) + \sum_{(i,j)\in E}(o^*_{(i,j)} t_{a_{(i,j)}} \sum_{k=0}^{n} \mathsf{m}^*_{(i,j)_k} x'^k) + t_b s^* + t_c)(x+t^*) r} - \frac{c(x)}{r}} \\
&= g_1^{\frac{c(x)(x+t^*)+r}{r(x+t^*)} - \frac{c(x)}{r}} \\
&= g_1^{\frac{1}{x+t^*}}
\end{aligned}
$$

and output $(g^{\frac{1}{x+t^*}}, t^*)$ as the solution for the SDH instance. On the other hand, if we have $v^* \in L_{(FO,FS)} \cup L_{(P,V)}$, $\mathcal{C}$ can extract the discrete logarithm x to break the SDH assumption.

Let $\Pr[Acc]$ be the probability of $\mathcal{C}$ outputs 1 in the showing protocol with $\mathcal{A}_1$, and $\Pr[Res]$ be the probability of $\mathcal{C}$ resets successfully, by Multi-Instance Reset Lemma [17], we have:

$$
\begin{aligned}
\Pr[S_7] &\leq \Pr[S_6] + \Pr[Acc] \\
&\leq \Pr[S_6] + \sqrt[N]{\Pr[Res] - 1} + 1/p + 1 \\
&\leq \Pr[S_6] + \sqrt[N]{\sqrt{\varepsilon_{\mathsf{sdh}}} - 1} + 1/p + 1. \qquad (8)
\end{aligned}
$$

Summing up the probability from (1) to (8), we have $\varepsilon_{\mathsf{imp}} \leq \sqrt[N]{\sqrt{\varepsilon_{\mathsf{sdh}}} - 1} + 1/p + 1 + (q-1)!/p^{q-1}$ as required. The time taken by $\mathcal{C}$ is at least $2Nt_{\mathsf{imp}}$ due to reset and interacting with N parallel impersonation instances, in addition to the environment setup and the final SDH solution extraction that cost $T(q^2)$. □

Lemma 11. *If an adversary $\mathcal{A}_2$ $(t_{\mathsf{imp}}, \varepsilon_{\mathsf{imp}})$-breaks the* imp-aca*-security of the proposed graph signature scheme, then there exists an algorithm $\mathcal{C}$ which $(t_{\mathsf{sdh}}, \varepsilon_{\mathsf{sdh}})$-solves the SDH problem such that:*

$$
\begin{aligned}
&\varepsilon_{\mathsf{imp}} \leq \sqrt[N]{\sqrt{\varepsilon_{\mathsf{sdh}}} - 1} + \frac{1 + (q-1)!/p^{q-2}}{p} + 1, \\
&t_{\mathsf{imp}} \leq t_{\mathsf{sdh}}/2N - T(q^2).
\end{aligned}
$$

where N is the total of adversary instances, $q = Q_{(O,S)} + Q_{(P,V)}$ is the number of queries made to the Obtain *and* Verify *oracles, while $T(q^2)$ is the time parameterized by q to setup the simulation environment and to extract the SDH solution.*

Proof. Given a q-SDH instance $(g_1, g_1^x, g_1^{x^2}, \ldots, g_1^{x^q}, g_2, g_2^x)$ where $q = Q_{(O,S)} + Q_{(P,V)}$ is the maximum number of queries $\mathcal{A}_2$ can issue to the Obtain and Verify oracles, there exists an algorithm $\mathcal{C}$ which can output $(g_1^{\frac{1}{x+t}}, t)$ by acting as the simulator for the graph signature scheme as follows:

Game$_0$. This is the same as the **Game$_0$** in Lemma 10 where we have:

$$\Pr[S_0] = \varepsilon_{\mathsf{imp}}. \tag{9}$$

Game$_1$. This is the same as the **Game$_1$** in Lemma 10 except that $\mathcal{C}$ additionally checks whether $X = g_2^{t_u}$ for $u \in \{1, \ldots, q\}$. If such t_u is found, $\mathcal{C}$ outputs the solution of the SDH instance using the discrete logarithm $x = t_u$. Let $f_{u',u}(x)$ denotes the polynomial $f_{u',u}(x) = \prod_{k=1,k\neq u',u}^{q}(x + t_k) = \sum_{k=0}^{q-2} \gamma_k x^k$. $\mathcal{C}$ uniformly selects random distinct $s_1, \ldots, s_q \in \mathbb{Z}_p^*$ and sends $(\mathsf{e}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, \{a_{0_k} = g_1^{f(x)t_a x'^k}, X_{0_k} = g_2^{x'^k}\}_{k=0}^{n}, \{\{a_{i_k}, X_{i_k}\}_{i=1}^{L}\}_{k=0}^{n}, b = g_1^{f(x)t_b - \sum_{u=1}^{q} f_u(x)}, c = g_1^{f(x)t_c + \sum_{u=1}^{q} s_u f_u(x)}, d = g_1^{f(x)t_d + \sum_{u=1}^{q} s_u f_u(x)}, h = g_1^{f(x)t_h}, X = g_2^x)$ as the public key to $\mathcal{A}_2$. This gives:

$$\Pr[S_1] \leq \Pr[S_0]. \tag{10}$$

Game$_2$. This is the same as the **Game$_2$** in Lemma 10 except that, after resetting $\mathcal{A}_2$, $\mathcal{C}$ simulates the signature $\sigma'_u = (t_u, s''_u, v_u)$ for $A_u = \{sk_{\mathsf{U}_u}, s'_u, \perp, \perp\}$ and $\mathcal{G}_{\mathsf{S}_u} = \{V_{\mathsf{S}_u} \cup E_{\mathsf{S}_u}\}$ such that:

$$v_i = h_u^{sk_{\mathsf{U}_u} r_{u,1}} \prod_{i \in V_{\mathsf{S}_u}} \prod_{k=0}^{n} a_{u,i_k}^{\mathsf{m}_{u,i_k}} \prod_{(i,j) \in E_{\mathsf{S}_u}} \prod_{k=0}^{n} a_{u,(i,j)_k}^{\mathsf{m}_{u,(i,j)_k}} b_u^{s'_u + (s_u - s'_u)} d_u$$

where $s''_u = s_u - s'_u$. When the protocol ends, $\mathcal{A}_2$ can compile the intermediary signature as $\sigma_{init_u} = (t_u, s_u = s'_u + s''_u, v_u, \mathcal{G}_{\mathsf{U}_u} = \mathcal{G}_{\mathsf{S}_u})$. As $\mathcal{C}$ simulates the InitSign oracle perfectly, we have:

$$\Pr[S_2] \leq \Pr[S_1] + (q-1!)/p^{q-1}. \tag{11}$$

where $\mathcal{A}_1$ can make, at most, another $q-1$ initial signature queries.

Game$_3$. This is the same as the **Game$_3$** in Lemma 10 except that, after resetting $\mathcal{A}_2$, $\mathcal{C}$ simulates the signature $\sigma'_u = (t_u, s''_u, v_u)$ for $A_u = \{sk_{\mathsf{U}_u}, s'_u, \mathcal{G}_{\mathsf{U}_u}, \{o_{u,i_1}, o_{u,(i,j)_1}\}\}$ and $\mathcal{G}_{\mathsf{S}_u}$ such that:

$$v_i = h_u^{sk_{\mathsf{U}_u} r_{u,1}} \prod_{i \in V_{\mathsf{U}_u}} \left(\prod_{k=0}^{n} a_{u,i_k}^{\mathsf{m}_{u,i_k}} \right)^{o_{u,i_1}} \prod_{(i,j) \in E_{\mathsf{U}_u}} \left(\prod_{k=0}^{n} a_{u,(i,j)_k}^{\mathsf{m}_{u,(i,j)_k}} \right)^{o_{u,(i,j)_1}} \cdot \prod_{i \in V_{\mathsf{S}_u}} \prod_{k=0}^{n} a_{u,i_k}^{\mathsf{m}_{u,i_k}} \prod_{(i,j) \in E_{\mathsf{S}_u}} \prod_{k=0}^{n} a_{u,(i,j)_k}^{\mathsf{m}_{u,(i,j)_k}} b_u^{s'_u + (s_u - s'_u)} d_u$$

where $s''_u = s_u - s'_u$. When the protocol ends, $\mathcal{A}_2$ can compile the intermediary signature as $\sigma_{interm_u} = (t_u, s_u = s'_u + s''_u, v_u, \mathcal{G}_{\mathsf{U}_u}, \{o_{u,i_1}, o_{u,(i,j)_1}\}, \mathcal{G}_{\mathsf{S}_u})$. As $\mathcal{C}$ simulates the $\mathsf{IntermSign}$ oracle perfectly, we have:

$$\Pr[S_3] \leq \Pr[S_2] + (q-2!)/p^{q-2}. \tag{12}$$

where $\mathcal{A}_1$ can make, at most, another $q-2$ intermediary signature queries.

Game$_4$. This is the same as the **Game**$_4$ in Lemma 10 except that, after resetting $\mathcal{A}_2$, $\mathcal{C}$ simulates the signature $\sigma'_u = (t_u, s''_u, v_u)$ for $A_u = \{sk_{\mathsf{U}_u}, s'_u, \mathcal{G}_u, \{o_{u,i}, o_{u,(i,j)}\}\}$ such that:

$$v_i = h_u^{sk_{\mathsf{U}_u} r_{u,1}} \prod_{i \in V_{\mathsf{U}_u}} \left(\prod_{k=0}^{n} a_{u,i_k}^{\mathsf{m}_{u,i_k}} \right)^{o_{u,i_1} o_{u,i_2}} \prod_{(i,j) \in E_{\mathsf{U}_u}} \left(\prod_{k=0}^{n} a_{u,(i,j)_k}^{\mathsf{m}_{u,(i,j)_k}} \right)^{o_{u,(i,j)_1} o_{u,(i,j)_2}} \cdot$$
$$\prod_{i \in V_{\mathsf{S}_u}} \left(\prod_{k=0}^{n} a_{u,i_k}^{\mathsf{m}_{u,i_k}} \right)^{o_{u,i_2}} \prod_{(i,j) \in E_{\mathsf{S}_u}} \left(\prod_{k=0}^{n} a_{u,(i,j)_k}^{\mathsf{m}_{u,(i,j)_k}} \right)^{o_{u,(i,j)_2}} b_u^{s'_u + (s_u - s'_u)} c_u$$

where $s''_u = s_u - s'_u$. When the protocol ends, $\mathcal{A}_2$ can compile the final signature as $\sigma_u = (t_u, s_u = s'_u + s''_u, v_u, \mathcal{G}_u, \{o_{u,i}, o_{u,(i,j)}\})$. As $\mathcal{C}$ simulates the $\mathsf{FinalSign}$ oracle perfectly, we have:

$$\Pr[S_4] \leq \Pr[S_3] + (q-2!)/p^{q-2}. \tag{13}$$

where $\mathcal{A}_1$ can make, at most, another $q-2$ final signature queries.

Game$_5$. This is the same as the **Game**$_5$ in Lemma 10 and we have:

$$\Pr[S_5] = \Pr[S_4]. \tag{14}$$

Game$_6$. This is the same as the **Game**$_6$ in Lemma 10 and we have:

$$\Pr[S_6] = \Pr[S_5]. \tag{15}$$

Game$_7$. Similar to the **Game**$_7$ in Lemma 10, $\mathcal{C}$ can reset $\mathcal{A}_2$ to extract the elements (t^*, s^*, v^*) of σ^* where v^* has the form:

$$v^* = \Bigg(g_1^{f(x)(t_h sk_{\mathsf{U}^*} + \sum_{i \in V} (o_i^* t_{a_i} \sum_{k=0}^{n} \mathsf{m}_{i_k}^* x'^k) + \sum_{(i,j) \in E} (o_{(i,j)}^* t_{a_{(i,j)}} \sum_{k=0}^{n} \mathsf{m}_{(i,j)_k}^* x'^k) + s^* t_b + t_c)} \cdot$$
$$g_1^{\sum_{u'=1, u' \neq u}^{q} (s_{u'} - s^*) f_{u'}(x) + (s_u - s^*) f_u(x)} \Bigg)^{1/(x+t_u)}.$$

Since $\mathcal{A}_2$ must output $t^* = t_u \in \{t_1, \ldots, t_q\}$ but $s^* \neq s_u \in \{s_1, \ldots, s_q\}$ for an $u \in \{1, \ldots, q\}$, $\mathcal{C}$ proceeds to compute $c(x)$ of degree $q-2$ and $r \in \mathbb{Z}_p^*$ from the knowledge of $\{t_1, \ldots, t_q\}$ such that $f_u(x) = c(x)(x + t_u) + r$. Moreover, it will be the case $v^* \notin L_{(FO,FS)} \cup L_{(P,V)}$ or $\mathcal{C}$ already found $x = t_u$ during **Game**$_1$. Subsequently, $\mathcal{C}$ calculates:

$$\left(v^*/g_1^{f_u(x)(t_h sk_{\mathsf{U}^*}+\sum_{i\in V}(o_i^* t_{a_i}\sum_{k=0}^{n}\mathsf{m}_{i_k}^* x'^k)+\sum_{(i,j)\in E}(o_{(i,j)}^* t_{a_{(i,j)}}\sum_{k=0}^{n}\mathsf{m}_{(i,j)_k}^* x'^k)+s^* t_b+t_c)}\cdot\right.$$
$$\left. g_1^{\sum_{u'=1,u'\neq u}^{q}(s_{u'}-s^*)f_{u',u}(x)+c(x)(s_u-s^*)}\right)^{\frac{1}{r(s_u-s^*)}}$$
$$= g_1^{\frac{(f_u(x)-c(x)(x+t_u))(s_u-s^*)}{r(s_u-s^*)(x+t_u)}}$$
$$= g_1^{\frac{1}{x+t_u}}$$

and outputs $(g_1^{\frac{1}{x+t_u}}, t_u)$ as the solution for the SDH instance. Therefore, we have:

$$\Pr[S_5] \le \Pr[S_4] + \sqrt[N]{\sqrt{\varepsilon_{\mathsf{sdh}}} - 1} + 1/p + 1 \tag{16}$$

and summing up the probability from (10) to (16), we have $\varepsilon_{\mathsf{imp}} \le \sqrt[N]{\sqrt{\varepsilon_{\mathsf{sdh}}} - 1} + 1/p + 1 + (q-1)!/p^{q-1}$ as required. The time taken by $\mathcal{C}$ is at least $2Nt_{\mathsf{imp}}$ due to reset and interacting with N parallel impersonation instances, in addition to the environment setup and the final SDH solution extraction that cost $T(q^2)$.

□

Lemma 12. *If an adversary $\mathcal{A}_3$ $(t_{\mathsf{imp}}, \varepsilon_{\mathsf{imp}})$-breaks the imp-aca-security of the proposed graph signature system, then there exists an algorithm $\mathcal{C}$ which $(t_{\mathsf{sdh}}, \varepsilon_{\mathsf{sdh}})$-solves the SDH problem such that:*

$$\varepsilon_{\mathsf{imp}} \le \sqrt[N]{\sqrt{\varepsilon_{\mathsf{sdh}}} - 1} + \frac{(q-1)!/p^{q-2}}{p} + 1,$$
$$t_{\mathsf{imp}} \le t_{\mathsf{sdh}}/2N - T(q^2).$$

where N is the total of adversary instances, $q = Q_{(O,S)} + Q_{(P,V)}$ is the number of queries made to the Obtain *and* Verify *oracles, while $T(q^2)$ is the time parameterized by q to setup the simulation environment and to extract the SDH solution.*

Proof. Given a q-SDH instance $(g_1, g_1^x, g_1^{x^2}, \ldots, g_1^{x^q}, g_2, g_2^x)$ where $q = Q_{(O,S)} + Q_{(P,V)}$ is the maximum number of queries $\mathcal{A}_3$ can make to the Obtain and Verify oracles, there exists an algorithm $\mathcal{C}$ which can output $(g_1^{\frac{1}{x+t}}, t)$ by acting as the simulator for the graph signature scheme as follows:

Game$_0$. This is the same as the **Game**$_0$ in Lemma 10 and we have:

$$\Pr[S_0] = \varepsilon_{\mathsf{imp}}. \tag{17}$$

Game$_1$. The precomputations and checking are the same as the **Game**$_1$ in Lemma 11 but $(\mathsf{e}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, \{a_{0_k} = g_1^{f(x)t_0 - \sum_{u=1}^{q} f_u(x)}, X_{0_k} = g_2^{x'^k}\}_{k=0}^{n}, \{\{a_{i_k},$

$X_{i_k}\}_{i=1}^{L}\}_{k=0}^{n}, b = g_1^{f(x)t_b - \sum_{u=1}^{q} f_u(x)}, c = g_1^{f(x)t_c + \sum_{u=1}^{q} \mathsf{z}_u f_u(x)}, X = g_2^x, X_0 = g_2, X_1 = X_0^{x'}, \ldots, X_n = X_0^{x'^n})$ as the public key to $\mathcal{A}_3$ where the random $z_1, \ldots, z_q \in \mathbb{Z}_p^*$ are uniformly distributed. This gives:

$$\Pr[S_1] \leq \Pr[S_0]. \tag{18}$$

Game$_2$. This is the same as the **Game**$_2$ in Lemma 10 except that, after resetting $\mathcal{A}_3$, $\mathcal{C}$ simulates the signature $\sigma'_u = (t_u, s''_u, v_u)$ for $A_u = \{sk_{\mathsf{U}_u}, s'_u, \perp, \perp\}$ and $\mathcal{G}_{\mathsf{S}_u} = \{V_{\mathsf{S}_u} \cup E_{\mathsf{S}_u}\}$ by letting

$$s_u = z_u - \prod_{i \in V_{\mathsf{S}_u}} (x' + i) \prod_{w \in f_{\mathcal{V}}(i)} (x' + w) - \prod_{(i,j) \in E_{\mathsf{S}_u}} (x' + i)(x' + j) \prod_{w \in f_{\mathcal{E}}(i,j)} (x' + w)$$

where:

$$v_i = g_1^{f(x)(t_h sk_{\mathsf{U}^*} + \sum_{i \in V}(t_{a_i} \sum_{k=0}^{n} \mathsf{m}^*_{i_k} x'^k) + \sum_{(i,j) \in E}(t_{a_{(i,j)}} \sum_{k=0}^{n} \mathsf{m}^*_{(i,j)_k} x'^k) + s^* t_b + t_c)} \cdot g_1^{\sum_{u'=1, u' \neq u_2}^{q} (z_{u'} - z_{u_2}) f_{u', u_2}(x)}$$

$$= h_u^{sk_{\mathsf{U}_u} r_{u,1}} \prod_{i \in V_{\mathsf{S}_u}} \prod_{k=0}^{n} a_{u,i_k}^{\mathsf{m}_{u,i_k}} \prod_{(i,j) \in E_{\mathsf{S}_u}} \prod_{k=0}^{n} a_{u,(i,j)_k}^{\mathsf{m}_{u,(i,j)_k}} b_u^{s'_u + (s_u - s'_u)} d_u$$

where $s''_u = s_u - s'_u$. When the protocol ends, $\mathcal{A}_3$ compiles the signature as $\sigma'_{init_u} = (t_u, s_u = s'_u + s''_u, v_u, \mathcal{G}_{\mathsf{U}_u} = \mathcal{G}_{\mathsf{S}_u})$. As $\mathcal{C}$ simulates the $\mathsf{InitSign}$ oracle perfectly, we have:

$$\Pr[S_2] \leq \Pr[S_1] + (q-1)!/p^{q-1}. \tag{19}$$

where $\mathcal{A}_1$ can make, at most, another $q-1$ initial signature queries.

Game$_3$. This is the same as the **Game**$_3$ in Lemma 10 except that, after resetting $\mathcal{A}_3, \mathcal{C}$ simulates the signature $\sigma'_u = (t_u, s''_u, v_u)$ for $A_u = \{sk_{\mathsf{U}_u}, s'_u, \mathcal{G}_{\mathsf{U}_u}, \{o_{u,i_1}, o_{u,(i,j)_1}\}\}$ and $\mathcal{G}_{\mathsf{S}_u}$ by letting

$$\begin{aligned} s_u = z_u - & \prod_{i \in V_{\mathsf{U}_u}} o_{u,i_1}(x' + i) \prod_{w \in f_{\mathcal{V}}(i)} (x' + w) - \\ & \prod_{(i,j) \in E_{\mathsf{U}_u}} o_{u,(i,j)_1}(x' + i)(x' + j) \prod_{w \in f_{\mathcal{E}}(i,j)} (x' + w) - \\ & \prod_{i \in V_{\mathsf{S}_u}} (x' + i) \prod_{w \in f_{\mathcal{V}}(i)} (x' + w) - \prod_{(i,j) \in E_{\mathsf{S}_u}} (x' + i)(x' + j) \prod_{w \in f_{\mathcal{E}}(i,j)} (x' + w) \end{aligned}$$

where:

$$v_i = h_u^{sk_{\mathsf{U}_u} r_{u,2}} \prod_{i \in V_{\mathsf{U}_u}} \left(\prod_{k=0}^{n} a_{u,i_k}^{\mathsf{m}_{u,i_k}} \right)^{o_{u,i_1}} \prod_{(i,j) \in E_{\mathsf{U}_u}} \left(\prod_{k=0}^{n} a_{u,(i,j)_k}^{\mathsf{m}_{u,(i,j)_k}} \right)^{o_{u,(i,j)_1}} \cdot$$

$$\prod_{i \in V_{\mathsf{S}_u}} \prod_{k=0}^{n} a_{u,i_k}^{\mathsf{m}_{u,i_k}} \prod_{(i,j) \in E_{\mathsf{S}_u}} \prod_{k=0}^{n} a_{u,(i,j)_k}^{\mathsf{m}_{u,(i,j)_k}} b_u^{s'_u + (s_u - s'_u)} d_u$$

and $s''_u = s_u - s'_u$. When the protocol ends, $\mathcal{A}_3$ compiles the signature as $\sigma'_{interm_u} = (t_u, s_u = s'_u + s''_u, v_u, \mathcal{G}_{\mathsf{U}_u}, \{o_{u,i_1}, o_{u,(i,j)_1}\}, \mathcal{G}_{\mathsf{S}_u})$. As $\mathcal{C}$ simulates the $\mathsf{IntermSign}$ oracle perfectly, we have:

$$\Pr[S_3] \leq \Pr[S_2] + (q-2)!/p^{q-2}. \tag{20}$$

where $\mathcal{A}_1$ can make, at most, another $q-2$ intermediary signature queries.

Game$_4$. This is the same as the **Game**$_4$ in Lemma 10 except that, after resetting $\mathcal{A}_3$, $\mathcal{C}$ simulates the signature $\sigma'_u = (t_u, s''_u, v_u)$ for $\{sk_{\mathsf{U}_u}, s'_u, \mathcal{G}_u, \{o_{u,i}, o_{u,(i,j)}\}\}$ by letting

$$\begin{aligned} s_u = z_u - & \prod_{i \in V_{\mathsf{U}_u}} o_{u,i_1} o_{u,i_2} (x'+i) \prod_{w \in f_{\mathcal{V}}(i)} (x'+w) - \\ & \prod_{(i,j) \in E_{\mathsf{U}_u}} o_{u,(i,j)_1} o_{u,(i,j)_2} (x'+i)(x'+j) \prod_{w \in f_{\mathcal{E}}(i,j)} (x'+w) - \\ & \prod_{i \in V_{\mathsf{S}_u}} o_{u,i_2} (x'+i) \prod_{w \in f_{\mathcal{V}}(i)} (x'+w) - \prod_{(i,j) \in E_{\mathsf{S}_u}} o_{u,(i,j)_2} (x'+i)(x'+j) \prod_{w \in f_{\mathcal{E}}(i,j)} (x'+w) \end{aligned}$$

where:

$$v_i = h_u^{sk_{\mathsf{U}_u} r_{u,2}} \prod_{i \in V_{\mathsf{U}_u}} \left(\prod_{k=0}^{n} a_{u,i_k}^{\mathsf{m}_{u,i_k}} \right)^{o_{u,i_1} o_{u,i_2}} \prod_{(i,j) \in E_{\mathsf{U}_u}} \left(\prod_{k=0}^{n} a_{u,(i,j)_k}^{\mathsf{m}_{u,(i,j)_k}} \right)^{o_{u,(i,j)_1} o_{u,(i,j)_2}} \cdot$$

$$\prod_{i \in V_{\mathsf{S}_u}} \left(\prod_{k=0}^{n} a_{u,i_k}^{\mathsf{m}_{u,i_k}} \right)^{o_{u,i_2}} \prod_{(i,j) \in E_{\mathsf{S}_u}} \left(\prod_{k=0}^{n} a_{u,(i,j)_k}^{\mathsf{m}_{u,(i,j)_k}} \right)^{o_{u,(i,j)_2}} b_u^{s'_u + (s_u - s'_u)} d_u$$

and $s''_u = s_u - s'_u$. When the protocol ends, $\mathcal{A}_3$ compiles the signature as $\sigma'_u = (t_u, s_u = s'_u + s''_u, v_u, \mathcal{G}_u, \{o_{u,i}, o_{u,(i,j)}\})$. As $\mathcal{C}$ simulates the $\mathsf{FinalSign}$ oracle perfectly, we have:

$$\Pr[S_4] \leq \Pr[S_3] + (q-2)!/p^{q-2}. \tag{21}$$

where $\mathcal{A}_1$ can make, at most, another $q-2$ final signature queries.

Game$_5$. This is the same as the **Game**$_5$ in Lemma 10 and we have:

$$\Pr[S_4] = \Pr[S_4]. \tag{22}$$

Game$_6$. This is the same as the **Game**$_6$ in Lemma 10 and we have:

$$\Pr[S_6] = \Pr[S_6]. \tag{23}$$

Game$_7$. By definition, $\mathcal{A}_3$ must output $t^* = t_u \in \{t_1, \ldots, t_q\}$ and $s^* = s_u \in \{s_1, \ldots, s_q\}$ for a $u \in \{1, \ldots, q\}$. Note that it must be the case $v^* \notin L_{(FO,FS)} \cup L_{(P,V)}$ or $x = t_u$ has been found during **Game**$_1$. In the unlikely case of Type 16 forgery $A^* \in L_{(P,V)}$ which happens with probability $1/p$, $\mathcal{C}$ aborts. Similar to the **Game**$_7$ in Lemma 10, $\mathcal{C}$ can reset $\mathcal{A}_3$ to extract the signature elements (t^*, s^*, v^*). $\mathcal{C}$ proceeds to compute $c(x)$ of degree $q-2$ and the remainder $r \in \mathbb{Z}_p^*$ from the knowledge of $\{t_1, \ldots, t_q\}$ such that $f_u(x) = c(x)(x+t_u) + r$. Subsequently, $\mathcal{C}$ calculates:

$$\left(v^*/g_1^{f_u(x)(t_h sk_{\mathsf{U}^*} + \sum_{i \in V}(t_{a_i} \sum_{k=0}^{n} \mathsf{m}^*_{i_k} x'^k) + \sum_{(i,j)\in E}(t_{a_{(i,j)}} \sum_{k=0}^{n} \mathsf{m}^*_{(i,j)_k} x'^k) + s^* t_b + t_c)} \cdot \right.$$
$$\left. g_1^{\sum_{u'=1, u' \neq u}^{q} (z_{u'} - z^*) f_{u', u_2}(x) + (z_{u'} - z^*) c(x)} \right)^{r(z_u - z^*)}$$
$$= g_1^{\frac{(f_u(x) - c(x)(x+t_u))(z_u - z^*)}{(x+t_u) r (z_u - z^*)}}$$
$$= g_1^{\frac{1}{x+t_u}}$$

and outputs $(g_1^{\frac{1}{x+t_u}}, t_u)$ as the solution for the SDH instance. Therefore, we have:

$$\Pr[S_7] \leq \Pr[S_6] + \sqrt[N]{\sqrt{\varepsilon_{\mathsf{sdh}}} - 1} + 1 \tag{24}$$

and summing up the probability from (17) to (24), we have $\varepsilon_{\mathsf{imp}} \leq \sqrt[N]{\sqrt{\varepsilon_{\mathsf{sdh}}} - 1} + 1 + (q-1)!/p^{q-1}$ as required. The time taken by $\mathcal{C}$ is at least $2Nt_{\mathsf{imp}}$ due to reset and interacting with N parallel impersonation instances, in addition to the environment setup and the final SDH solution extraction which cost $T(q^2)$. $\square$

Combining Theorem 2, Lemmas 10, 11, and 12 gives Theorem 5 as required.

6.11.2 Unlinkability.

Next, we prove the unlinkability of the proposed graph signature scheme. It is sufficient to show that the witnesses, the committed graph in the signing protocols and showing protocols are perfectly hiding. Then, we demonstrate that every instance of the protocols is uniformly distributed due to the random self-reducibility property. This implies that even when $\mathcal{A}$ is given access to the Obtain, Prove, Verify and Corrupt oracles, it does not has advantage in guessing the challenged graphs.

Lemma 13. *The initialization of the signing protocol in the graph signature scheme has random self-reducibility.*

Proof. We use initial signing protocol as an example. Let $Gen = \mathsf{KeyGen}$, $P = user$, $V = signer$, $pk = C$ and $sk = (sk_{\mathsf{U}_u}, s')$, we define the algorithms Rerand, Derand and Tran as follows:

- Rerand(C) randomly selects $\rho \in \mathbb{Z}_p^*$ and outputs $C' = C^\rho$ where $C = h^{sk_{\mathsf{U}}} b^{s'}$ is the commitment generated by $user$. For all (C, sk_{U}, s'), $(C', sk'_{\mathsf{U}}, s'')$ has the same uniform distribution as another $(C'', sk''_{\mathsf{U}}, s''')$ which would have been generated by $user$.
- Derand$(C, C', sk'_{\mathsf{U}}, s''), \rho)$ outputs $(sk_{\mathsf{U}}, s') = (\{sk'_{\mathsf{U}}/\rho\}, s''/\rho)$ for all $(C', \rho) \in$ Rerand(C).

– $\mathsf{Tran}(C, C', \rho, (\tilde{C}', e', \{\hat{sk_U}'\}, \hat{s}''))$ outputs $(\tilde{C} = \tilde{C}'^{1/\rho}, \{\hat{sk_U}\} = \{\hat{sk_U}'/\rho\}, \hat{s}' = \hat{s}''/\rho$ for all $(C', \rho) \in \mathsf{Rerand}(C)$. The transcript $(\tilde{C}, e', \{\hat{sk_U}\}, \hat{s}')$ is valid wrt. C if $(\tilde{C}', e', \{\hat{sk_U}'\}, \hat{s}'')$ is valid wrt. C'.

□

Lemma 14. *The blinded graph signature is perfectly hiding.*

Proof. This has been proven in the previous work [24]. □

Lemma 15. *The showing protocol of the graph signature scheme offers random self-reducibility.*

Proof. The proof is similar to that of Lemma 13 as well as to the previous work [24]. □

Theorem 6. *If the signing protocols are performed in secure channels and the showing protocols have random self-reducibility, and their witnesses, committed graph as well as the blinded graph signature are perfectly hiding, the graph signature scheme is* gunl-aca*-secure.*

Proof. We show that an adversary $\mathcal{A}$ can win the aunl-aca-security game only with a negligible advantage $\varepsilon_{\mathsf{gunl}}$, with respect to the graph signature scheme simulator $\mathcal{C}$.

Game$_0$. This is an attack on the original graph signature scheme. Let S_0 denotes a successful distinguishing attempt, by definition we have:

$$\Pr[S_0] \leq \varepsilon_{\mathsf{gunl}} + \frac{1}{2}. \tag{25}$$

Game$_1$. $\mathcal{C}$ generates (pk, sk) as in the original algorithm and forwards to $\mathcal{A}$ so that the latter can play the role of user and signer. In addition, $\mathcal{C}$ maintains four list $L_{(iO,iS)}, L_{(IO,IS)}, L_{(FO,FS)}, L_{(P,V)}$ for corrupted signing protocols and showing protocols, respectively. Since $\mathcal{C}$ does not alter the key generation algorithm, this gives:

$$\Pr[S_1] = \Pr[S_0]. \tag{26}$$

Game$_2$. When $\mathcal{A}$ act as the user to concurrently interact with signers, $\mathcal{C}$ simulates Obtain oracle to produce the corresponding signature σ_i for the user in the i-th session, as well as adding the related information into the corresponding lists $L_{(iO,iS)}, L_{(IO,IS)}, L_{(FO,FS)}$. Without lost of generality, we assume every user uses different data set A_i. Since our graph signature works in the secure channel, it is clear that every transcript is well hidden. If it is the hybrid ElGamal encryption that is hiding the transcripts, its ciphertext indistinguishability security also hides the transcripts from the adversary. Moreover, every protocol session is uniformly distributed by Lemma 13. This gives:

$$\Pr[S_2] = \Pr[S_1]. \tag{27}$$

$\mathbf{Game}_3$. Comparing to the previous games, $\mathcal{A}$ additionally queries the signing protocol transcript of the i-th session to the $\mathsf{Corrupt}$ oracle. $\mathcal{C}$ searches in $L_{(iO,iS)}, L_{(IO,IS)}, L_{(FO,FS)}$ to return the internal state and the random exponents used in completing the protocol. By Lemma 13, for any two signing transcripts returned by $\mathsf{Corrupt}$, the distribution of their transcripts are identical to each other from the view of $\mathcal{A}$. Following Lemma 5 to Lemma 16, this is true even for the non-uniformly distributed graphs $\mathcal{G}_i$ which have been hidden by o_i and $o_{(i,j)}$. Since $\mathcal{A}$ does not gain any advantage, we have:

$$\Pr[S_3] = \Pr[S_2]. \tag{28}$$

$\mathbf{Game}_4$. Now $\mathcal{A}$ also acts as the verifier to concurrently interact with $\mathcal{C}$ as the provers for multiple signatures. $\mathcal{C}$ runs the i-th session of a showing protocol for $\sigma_i = (t_i, s_i, v_i, \mathcal{G}_i, \{o_i, o_{(i,j)}\})$. Without loss of generality, we assume $\mathcal{A}$ always requests for successful show proofs where $\phi_{\mathsf{stmt}}(A_i) = 1$.

The interaction is the same as in the original show proof from the view of $\mathcal{A}$. Moreover, Lemma 14 shows that the blinded signature is perfectly hiding and Lemma 15 indicates every protocol session is uniformly distributed. This gives:

$$\Pr[S_4] = \Pr[S_3]. \tag{29}$$

$\mathbf{Game}_5$. In contrast to the previous games, $\mathcal{A}$ also queries the showing protocol transcript of the i-th session to the $\mathsf{Corrupt}$ oracle. $\mathcal{C}$ searches in $L_{(P,V)}$ to return the internal state and the random exponents used in completing the protocol. By Lemma 15, for any two witness sets in a showing protocol returned by $\mathsf{Corrupt}$, the distribution of their transcripts are identical from the view of $\mathcal{A}$. Following Lemma 14, this is true even if $\mathcal{A}$ knows some final signatures σ from the signing protocols, which now have been perfectly hidden by (r_i, y_i). $\mathcal{A}$ can also act as a prover in which it does not gain useful information. We have:

$$\Pr[S_5] = \Pr[S_4] \tag{30}$$

where $\mathcal{A}$ does not gain any advantage from the query.

$\mathbf{Game}_6$. When $\mathcal{A}$ decides two data sets A_0 and A_1 as well as the predicate ϕ^*_{stmt} which he wishes to challenge such that $\phi^*_{\mathsf{stmt}}(A_0) = \phi^*_{\mathsf{stmt}}(A_1) = 1$, $\mathcal{C}$ randomly decides a bit $b \in \{0,1\}$ and play the user role to run the challenge signing protocols with $\mathcal{A}$ for A_b and A_{1-b}, respectively. When the final signing protocols are completed, $\mathcal{C}$ obtains two signatures σ_b and σ_{1-b}. In the same order, $\mathcal{C}$ uses σ_b and σ_{1-b} to complete the challenge show proof with $\mathcal{A}$ as the verifier. $\mathcal{A}$ can request polynomially many times of show proofs. From time to time, $\mathcal{A}$ still can query the oracles as before with the restriction of querying the challenge transcripts to $\mathsf{Corrupt}$. Finally, if $\mathcal{A}$ makes a correct guess $b' = b$, it breaks the graph unlinkability with the probability:

$$\begin{aligned}\Pr[S_6] &= \Pr[S_5] \\ &= \Pr[b' = b] \\ &= \frac{1}{2} + \varepsilon_{\mathsf{gunl}}.\end{aligned} \tag{31}$$

Combining the probability from equation (25) to (31), we have a negligible $\varepsilon_{\mathsf{gunl}}$ as required and $\mathcal{A}$ runs in time t_{gunl}. □

Using the similar approach, we show that the security of protocol unlinkability also holds for the proposed graph signature scheme.

Theorem 7. *If the showing protocols have random self-reducibility, and their witnesses, committed graph as well as the blinded graph signature are perfectly hiding, the graph signature scheme is* punl-aca*-secure.*

Proof. The proof is the same as that of Theorem 6 except **Game**$_6$:

Game$_6$**.** When $\mathcal{A}$ decides two graphs A_0 and A_1 as well as the access policy ϕ^*_{stmt} which he wishes to challenge such that $\phi^*_{\mathsf{stmt}}(A_0) = \phi^*_{\mathsf{stmt}}(A_1) = 1$, $\mathcal{C}$ randomly decides a bit $b_1 \in \{0, 1\}$ and play the user role to run the challenge signing protocols with $\mathcal{A}$ for A_{b_1} and A_{1-b_1}, respectively. When the final signing protocols are completed, $\mathcal{C}$ obtains two signatures σ_{b_1} and σ_{1-b_1}. $\mathcal{C}$ randomly decides another bit $b_2 \in \{0, 1\}$ and uses σ_{b_2} and σ_{1-b_2} to complete the challenge showing protocol with $\mathcal{A}$ as the verifier. $\mathcal{A}$ can request polynomially many times of show proofs. From time to time, $\mathcal{A}$ still can query the oracles as before with the restriction of querying the challenge transcripts to Corrupt. Finally, if $\mathcal{A}$ makes a correct guess $(\pi_{(O,I)}, \pi_{(P,V)})$ such that $\sigma_{\pi_{(O,I)}} = \sigma_{\pi_{(P,V)}}$, it breaks the full protocol unlinkability of the graph signature scheme with the probability:

$$\begin{aligned}\Pr[S_6] &= \Pr[S_5] \\ &= \Pr[\sigma_{\pi_{(O,I)}} = \sigma_{\pi_{(P,V)}}] \\ &= \frac{1}{2} + \varepsilon_{\mathsf{punl}}. \end{aligned} \tag{32}$$

Therefore, we have a negligible $\varepsilon_{\mathsf{punl}}$ as required and $\mathcal{A}$ runs in time t_{punl}. □

7 Arguments on Graph Properties

In this section, we present the details for the statements edge, connected and isolated.

7.1 edge(i, j)

The edge(i, j) statement is a special case of set proof such that the query is a single edge identifier $E' = (i^*, j^*)$ as below:

$$\begin{aligned} PK\Bigg\{ &(\forall i \in V : \varepsilon_{i_0}, \varepsilon_{i_1}), (\forall (i,j) \in E \setminus (i^*, j^*) : \varepsilon_{(i,j)_0}, \varepsilon_{(i,j)_1}), \zeta, \rho, \omega, \tau, \gamma) : \\ &\prod_{i \in V} \mathsf{e}\left(W'_i, X_{0_1}^{\varepsilon_{i_1}} X_{0_0}^{\varepsilon_{i_0}}\right) \prod_{(i,j) \in E \setminus E'} \mathsf{e}\left(W'_{(i,j)}, X_{0_1}^{\varepsilon_{(i,j)_1}} X_{0_0}^{\varepsilon_{(i,j)_0}}\right) \cdot \\ &\mathsf{e}(W'_{(i^*,j^*)}, X_2 X_1^{i^*+j^*} X_0^{i^* \cdot j^*}) \mathsf{e}\left(h^\zeta b^\rho c^\omega v'^{-\tau}, X_{0_0}\right) = \mathsf{e}(v'^\gamma, X) \Bigg\} \end{aligned}$$

where $\{W'_i, W'_{(i,j)}\}$ and $W'_{(i^*,j^*)} = \prod_{k=0}^{|E_{(i^*,j^*)}|-2} a_{(i,j)_k}^{r \prod_{w \in f_{\mathcal{E}}(i^*,j^*)}(x'+w)}$ are public inputs. If $E'_{(i^*,j^*)} \in \mathcal{G}_{\mathsf{U}}$, it is $W'_{(i^*,j^*)} = \prod_{k=0}^{|E_{(i^*,j^*)}|-2} a_{(i,j)_k}^{o_{(i^*,j^*)} r \prod_{w \in f_{\mathcal{E}}(i^*,j^*)}(x'+w)}$ instead.

7.2 $\mathsf{connected}(i,j,\ell)$

The $\mathsf{connected}(i,j,\ell)$ statement shows that two vertices i and j are connected by ℓ edges. This is an extension of $\mathsf{edge}(i,j)$ statement, which is a simplified ℓ times edges within a set statement. The $\mathsf{connected}$ statement for ℓ edges E' connecting i and j is as below:

$$PK\Bigg\{(\forall i \in V : \varepsilon_{i_0}, \varepsilon_{i_1}), (\forall (i,j) \in E \setminus E'\} : \varepsilon_{(i,j)_0}, \varepsilon_{(i,j)_1}), \{\varepsilon_{l,0}\}_{l=2}^{\ell-1}, \{\varepsilon_{l,1}\}_{l=1}^{\ell-1},$$

$$\{\varepsilon_{l,2}\}_{l=1}^{\ell}, \zeta, \rho, \omega, \tau, \gamma) :$$
$$\prod_{i \in V} \mathsf{e}\left(W'_i, X_{0_1}^{\varepsilon_{i_1}} X_{0_0}^{\varepsilon_{i_0}}\right) \prod_{(i,j) \in E \setminus E'} \mathsf{e}\left(W'_{(i,j)}, X_{0_1}^{\varepsilon_{(i,j)_1}} X_{0_0}^{\varepsilon_{(i,j)_0}}\right) \cdot$$
$$\mathsf{e}\left(W'_1, X_{0_2}^{\varepsilon_{1,2}} (X_{0_1}^{i^*})^{\varepsilon_{1,2}} X_{0_1}^{\varepsilon_{1,1}} (X_{0_0}^{i^*})^{\varepsilon_{1,1}}\right) \mathsf{e}\left(W'_2, X_{0_2}^{\varepsilon_{2,2}} X_{0_1}^{\varepsilon_{1,1}} X_{0_1}^{\varepsilon_{2,1}} X_{0_0}^{\varepsilon_{2,0}}\right) \cdots$$
$$\cdots \mathsf{e}\left(W'_{\ell-1}, X_{0_2}^{\varepsilon_{\ell-1,2}} X_{0_1}^{\varepsilon_{\ell-2,1}} X_{0_1}^{\varepsilon_{\ell-1,1}} X_{0_0}^{\varepsilon_{\ell-1,0}}\right) \mathsf{e}\left(W'_\ell, X_{0_2}^{\varepsilon_{\ell,2}} X_{0_1}^{\varepsilon_{\ell-1,1}} (X_{0_1}^{j^*})^{\varepsilon_{\ell,2}} (X_{0_0}^{j^*})^{\varepsilon_{\ell-1,1}}\right) \cdot$$
$$\mathsf{e}\left(h^\zeta b^\rho c^\omega v'^{-\tau}, X_{0_0}\right) = \mathsf{e}(v'^\gamma, X)\Bigg\}$$

where $\{W'_i, W'_{(i,j)}\}$ and $W'_1, \ldots, W'_\ell$ are public inputs. The details for the secret exponents are listed below:

- $\varepsilon_{1,2} = \varepsilon_{2,2} = \cdots = \varepsilon_{\ell,2} = r_{(i^*,j^*)}$,
- $\varepsilon_{1,1} = r_{(i^*,j^*)} \cdot j_1 = r_{(i^*,j^*)} \cdot i_2$,
- $\varepsilon_{2,1} = r_{(i^*,j^*)} \cdot j_2$,
- $\varepsilon_{2,0} = r_{(i^*,j^*)} \cdot i_2 j_2$,
  $\ldots$
- $\varepsilon_{\ell-1,1} = r_{(i^*,j^*)} \cdot j_{\ell-1} = r_{(i^*,j^*)} \cdot i_\ell$

such that all ℓ edges share the same randomness $r_{(i^*,j^*)} \in \mathbb{Z}_p^*$.

Lemma 16. *The randomization of the ℓ-path in the* $\mathsf{connected}$ *predicate is perfectly hiding.*

Proof. The proof is similar to that of Lemma 1. □

7.3 $\mathsf{isolated}(i,j)$

The $\mathsf{isolated}$ statement allows a prover to prove the possession of i and j as well as their disjointness. In brief, a prover shows that his graph consists of a vertex set and some sub-graphs that are represented by edge sets, with i and j fall in different sub-graphs. Specifically, a prover proves the vertex composition to separate the vertex set from the edge sets. Subsequently, a bi-partition variant of edges is executed to show that i and j are disjoint. Let E_{i^*} and $E_{j^*} = E \setminus E_{i^*}$ represent the disjoint edge sets which consist of the queried i^* and j^* respectively. We describe the $\mathsf{isolated}$ statement as the following protocol:

$$PK\Bigg\{((\forall i \in V : \varepsilon_{i_0}, \varepsilon_{i_1}), (\forall (i,j) \in E : \varepsilon_{(i,j)_0}, \varepsilon_{(i,j)_1}, \varepsilon_{(i,j)_2}), \zeta, \rho, \omega, \tau, \gamma,$$
$$\{\alpha_k\}_{k=0}^{\min(|E_{i^*}|,|E_{j^*}|)-1}) :$$
$$\prod_{i \in V} \mathsf{e}\left(W_i', X_{0_1}^{\varepsilon_{i_1}} X_{0_0}^{\varepsilon_{i_0}}\right) \prod_{(i,j)\in E} \mathsf{e}\left(W'_{(i,j)}, \prod_{k=0}^{2} X_{0_k}^{\varepsilon_{(i,j)_k}}\right) \cdot$$
$$\mathsf{e}\left(h^\zeta b^\rho c^\omega v'^{-\tau}, X_0\right) = \mathsf{e}(v'^\gamma, X) \wedge$$
$$\mathsf{vertices}(V) \wedge \mathsf{edges}(E_{i^*}) \wedge i^* \in E_{i^*} \wedge \mathsf{edges}(E_{j^*}) \wedge j^* \in E_{j^*} \wedge \phi_{\mathsf{disjoint}(E_{i^*})}(E_{j^*})\Bigg\}.$$

The $\mathsf{isolated}$ proof above is still valid if the prover runs edges without the set membership proof for the edge labels. By the graph encoding setting, a commitment with two vertex identifiers must be an edge. Furthermore, if the base pool setting for $\mathsf{disjoint}$ is applied here, the $\mathsf{vertices}$ predicate can be skipped. The simplified proof is as below:

$$PK\Bigg\{((\forall i \in V : \varepsilon_{i_0}, \varepsilon_{i_1}), (\forall (i,j) \in E : \varepsilon_{(i,j)_0}, \varepsilon_{(i,j)_1}, \varepsilon_{(i,j)_2}), \zeta, \rho, \omega, \tau, \gamma,$$
$$\{\alpha_k\}_{k=0}^{\min(|E_{i^*}|,|E_{j^*}|)-1}) :$$
$$\prod_{i \in V} \mathsf{e}\left(W_i, \prod_{k=0}^{1} X_{i_k}^{\varepsilon_{i_k}}\right) \prod_{(i,j)\in E} \mathsf{e}\left(W'_{(i,j)}, \prod_{k=0}^{2} X_{0_k}^{\varepsilon_{(i,j)_k}}\right) \cdot$$
$$\mathsf{e}\left(h^\zeta b^\rho c^\omega v'^{-\tau}, X_0\right) = \mathsf{e}(v'^\gamma, X) \wedge$$
$$\varepsilon_{(i,j)}[C_{(i,j)_{E_{l_{i^*}}}}]_{l_{i^*}=1}^{\ell_{i^*}} \in \Xi_{\mathcal{V}} \wedge i^* \in E_{i^*} \wedge$$
$$\varepsilon_{(i,j)}[C_{(i,j)_{E_{l_{j^*}}}}]_{l_{j^*}=1}^{\ell_{i^*}} \in \Xi_{\mathcal{V}} \wedge j^* \in E_{j^*} \wedge$$
$$\phi_{\mathsf{disjoint}(E_{j^*})}(E_{i^*})\Bigg\}$$

where the last statement is changed to $\phi_{\mathsf{disjoint}(E_{i^*})}(E_{j^*})$ if $|E_{j^*}| > |E_{i^*}|$.

8 Performance

We list the show proofs complexity in Table 3 and that of the optimized version from Appendix A in Table 4. In order to ease the presentation, we assume every vertex $V_i \in V$ and edge $E_i \in E$ has the same label length l.

Table 3. Complexity for our proofs of knowledge statements.

Statement	Point Add. ($\mathbb{G}_1$)	Point Mult. ($\mathbb{G}_1$)	Point Add. ($\mathbb{G}_2$)	Point Mult. ($\mathbb{G}_2$)	Mult. ($\mathbb{G}_T$)	Pairing
vertices	$(l+2)n + 2(\|\varXi_{\mathcal{V}}\| + \|\varXi_{\mathcal{L}}\|) + \sum_{k=1}^{n-1} k$	$(l+2)n + 2(\|\varXi_{\mathcal{V}}\| + \|\varXi_{\mathcal{L}}\| + 1) + \sum_{k=1}^{n-1} (k+1)$	$5n + \sum_{k=1}^{n} kl - 1$	$7n + \sum_{k=1}^{n} (kl+1) + 1$	$3n$	$3n+5$
edges	$(l+3)m + 2\|\varXi_{\mathcal{L}}\| + \sum_{i=1}^{\ell}(2\|\varXi_{\mathcal{V}}\| + \sum_{k=1}^{\|E_i\|-1} k)$	$4m + ml + 2\|\varXi_{\mathcal{L}}\| + \sum_{i=1}^{\ell}(2\|\varXi_{\mathcal{V}}\| - 2m + \sum_{k=1}^{\|E_i\|-1} (k+1) + 1) + 1$	$7m + \sum_{i=1}^{\ell} \|E_i\| + \sum_{k=1}^{m} kl - 1$	$9m + \sum_{i=1}^{\ell}(\|E_i\|+1) + \sum_{k=1}^{m}(kl+1)$	$2m + \sum_{i=1}^{\ell} \|E_i\| + 1$	$2m + \sum_{i=1}^{\ell}(\|E_i\|+1) + 4$
possession	$l(n+m)+9$	$(l+1)n + (l+2)m + 15$	$3(n+m)$	$6(n+m)$	$n+m$	$n+m+2$
set	$l(n-m') + (l+1)m + (1-l)n' + 9$	$(l+1)n + (1-l)n' + (l+2)m - lm' + 15$	$3(n+m) + (l-2)n' + (l-1)m'$	$6(n+m) + (l-5)n' + (l-4)m'$	$n+m$	$n+m+2$
disjoint	$(2l+2)n + ml + 3n' + 2\|\varXi_{\mathcal{V}}\| + \sum_{k=1}^{n-1} k + 1$	$(2l+2)n + (l+1)m + 3n' + 2\|\varXi_{\mathcal{V}}\| + \sum_{k=1}^{n-1}(k+1) + 16$	$4n + 3m + n'$	$7n + 6m + n' + 1$	$3n+m$	$3n+m+8$
disjoint*	$nl + ml + m + n'(nl + 3n - 3) + 9$	$n(l+1) + m(l+2) + n'(n(l+1) + 3n) + 15$	$n+m+n'$	$6(n+m)+n'$	$n+m+2n'-1$	$n+m+2n'+2$
cover	$l(n+n''+m) + n' + m' + (l+1)m'' + 9$	$(l+1)(n+n'') + (l+2)(m + m'') + 2(n'+m') + 15$	$3(n+m+n''+m'') + (l+1)n' + (l+2)m'$	$6(n+m+n''+m'') + (l+1)n' + (l+2)m'$	$n+m+n'+m'-1$	$n+m+n'+m'+n''+m''+2$
edge	$ln + (l+1)m + 8$	$(l+1)n + (l+2)m + 14$	$3(n+m)-1$	$6(n+m)-4$	$n+m$	$n+m+2$
connected	$nl + (l+1)m - \ell + 9$	$(l+1)n + (l+2)m - \ell + 15$	$3(n+m) + 6\ell$	$6(n+m) + 3\ell + 19$	$n+m$	$n+m+2$
isolated	$nl + ml + m + \sum_{l_{i^*}=1}^{\ell_{i^*}}(2\|\varXi_{\mathcal{V}}\| + \sum_{k=1}^{\|E_{l_{i^*}}\|-1} k) + \sum_{l_{j^*}=1}^{\ell_{j^*}}(2\|\varXi_{\mathcal{V}}\| + \sum_{k=1}^{\|E_{l_{j^*}}\|-1} k) + \sum_{l_{i^*}=1}^{\ell_{i^*}}(l_{i^*}) + \sum_{l_{j^*}=1}^{\ell_{j^*}}(l_{j^*}) + 3\min(\|E_{i^*}\|, \|E_{j^*}\|) + 4$	$nl + n + ml + 2m + \sum_{l_{i^*}=1}^{\ell_{i^*}}(2\|\varXi_{\mathcal{V}}\| - \|E_{i^*}\| + \sum_{k=1}^{\|E_{l_{i^*}}\|-1}(k+1) + 1) + \sum_{l_{j^*}=1}^{\ell_{j^*}}(2\|\varXi_{\mathcal{V}}\| - \|E_{j^*}\| + \sum_{k=1}^{\|E_{l_{j^*}}\|-1}(k + 1) + 1) + 3\min(\ell_{i^*}, \ell_{j^*}) + 15$	$3n + 6m + \sum_{l_{i^*}=1}^{\ell_{i^*}} \|E_{l_{i^*}}\| + \sum_{l_{j^*}=1}^{\ell_{j^*}} \|E_{l_{j^*}}\| + \max(\|E_{i^*}\|, \|E_{j^*}\|) - \min(\|E_{i^*}\|, \|E_{j^*}\|) + 2$	$6n + 9m + \sum_{l_{i^*}=1}^{\ell_{i^*}}(\|E_{l_{i^*}}\| + 1) + \sum_{l_{j^*}=1}^{\ell_{j^*}}(\|E_{l_{j^*}}\| + 1) + \max(\|E_{i^*}\|, \|E_{j^*}\|) - \min(\|E_{i^*}\|, \|E_{j^*}\|) + 5$	$n + m + \sum_{l_{i^*}=1}^{\ell_{i^*}} \|E_{l_{i^*}}\| + \sum_{l_{j^*}=1}^{\ell_{j^*}} \|E_{l_{j^*}}\| + \ell_{i^*} + \ell_{j^*} - 1$	$n + m + \sum_{l_{i^*}=1}^{\ell_{i^*}}(\|E_{l_{i^*}}\| + 1) + \sum_{l_{j^*}=1}^{\ell_{j^*}}(\|E_{l_{j^*}}\| + 1) + \ell_{i^*} + \ell_{j^*} + 13$

Note: l: label length, n: $|V|$, m: $|E|$, m_0: $|E_0|$, m_1: $|E_1|$, n': $|V'|$, m': $|E'|$, n'': $|V'-V|$, m'': $|E'-E|$, ℓ: threshold, *: the slower alternative.

8.1 Efficiency Comparison

We compare the efficiency of show proofs of our proposed graph signature to that of Groß' graph signature [14, 15] in Table 5. We first recall the notations from Groß' graph signature scheme. Let n be the total vertices and $m \leq \frac{n(n-1)}{2}$ be the total edges in a graph. Also, let k be the number of sets considered in a statement where $k \leq l \leq n$ and $O(kl) = O(n)$. We follow the approach in Groß' work in computing the asymptotic complexity where we consider only the complexity of the core computations to realize a statement. For instance, the $O(1)$ for edge in Groß' signature does not include the complexity of possession, and neither does ours.

Our graph signature outperforms Groß' graph signature in some statements because the set membership proofs in vertices and edges implicitly prove pair-wise disjointness. This allows our graph signature to enjoy a more efficient graph well-formedness proof as well as realizing the disjoint statement using the set statement instead of the more complicated partition statement [14, 15]. Our proofs can be further optimized to reduce its complexity as shown in Appendix A. The optimization transfers the computation cost from pairing to the point multiplication in $\mathbb{G}_1$ for every statement and achieves better efficiency. For instance, the optimization on possession reduces $n+m+2$ pairings to only 4 while added $n(|f_{\mathcal{V}}(i)|+7)+m(|f_{\mathcal{E}}(i)|+8)$ point multiplications in $\mathbb{G}_1$. Moreover, Table 5 shows that the optimization on set reduces the complexity from $O(|V'|)$ to $O(n_{\max})$ where $n_{\max} = \max(|V'_1|, \ldots, |V'_{|V'|}|)$, a number significantly smaller than $|V'|$

Table 4. Complexity for our optimized proofs of knowledge statements.

Statement	Point Add. ($\mathbb{G}_1$)	Point Mult. ($\mathbb{G}_1$)	Point Add. ($\mathbb{G}_2$)	Point Mult. ($\mathbb{G}_2$)	Mult. ($\mathbb{G}_T$)	Pairing
vertices	$(l+13)n+2(\lvert\Xi_{\mathcal{V}}\rvert+\lvert\Xi_{\mathcal{L}}\rvert)+\sum_{k=1}^{n-1}k-5$	$(l+16)n+2(\lvert\Xi_{\mathcal{V}}\rvert+2\lvert\Xi_{\mathcal{L}}\rvert)+\sum_{k=1}^{n-1}(k+1)+2$	$2n+\sum_{k=1}^{n}kl-1$	$n+\sum_{k=1}^{n}(kl+1)+7$	$n+4$	$n+10$
edges	$8m+ml+2\lvert\Xi_{\mathcal{V}}\rvert+\sum_{i=1}^{\ell}(2(\lvert E_i\rvert+\lvert\Xi_{\mathcal{V}}\rvert)+\sum_{k=1}^{\lvert E_i\rvert-1}k-2)+6\sum_{i=1}^{\ell}(\lvert E_i\rvert-1)-7$	$13m+ml+2\lvert\Xi_{\mathcal{L}}\rvert+\sum_{i=1}^{\ell}(2\lvert\Xi_{\mathcal{V}}\rvert+\sum_{k=1}^{\lvert E_i\rvert-1}(k+1)+8\lvert E_i\rvert+1)+1$	$\sum_{i=1}^{\ell}\lvert E_i\rvert+\sum_{k=1}^{m}kl+m-1$	$\sum_{k=1}^{m}(kl+1)+\sum_{i=1}^{\ell}(\lvert E_i\rvert+1)+9$	$m+3\ell+3$	$m+4\ell+7$
possession	$(2l+6)n+(2l+8)m-1$	$(2l+8)n+(2l+10)m+15$	0	6	2	4
set	$(2l+4)n-(2l-6)n'+(2l+6)m-(2l-5)m'+5$	$(2l+8)n-(2l-6)n'+(2l+10)m-(2l-8)m'+15$	$n'+m'$	$n'+m'+4$	$n'+m'+1$	$n'+m'+4$
set[#]	$(2l+4)(n-n')+(2l+6)(m-m')+n_{\max}(2n'+2m'-1)+5$	$(2l+8)(n-n')+(2l+10)(m-m')+3n_{\max}(n'+m')+15$	0	2	$n_{\max}+1$	$n_{\max}+4$
disjoint	$(l+14)n+ml+3n'+2\lvert\Xi_{\mathcal{V}}\rvert+\sum_{k=1}^{n-1}k-9$	$2nl+16n+ml+m+3n'+\lvert\Xi_{\mathcal{V}}\rvert+\sum_{k=1}^{n-1}(k+1)+15$	$n+4m+n'$	$n+6m+n'+8$	$n+m+5$	$n+m+12$
disjoint*	$lm+lnn'+ln+m+3nn'-2n'+9$	$n(l+1)+m(l+2)+n'(n(l+1)+3n+1)+15$	$n+m+n'$	$6(n+m)$	$n+m+n'$	$n+m+n'+3$
cover	$(2l+6)(n+n'')+(2l+8)(m+m'')+n'+m'-6$	$(2l+8)(n+n'')+(2l+10)(m+m'')+2n'+2m'+15$	$n'+m'$	$n'+m'+8$	$n'+m'+3$	$n'+m'+6$
cover[#]	$(2l+6)(n+n'')+(2l+8)(m+m'')+n_{\max}(2n'+2m'-1)-6$	$(2l+8)(n+n'')+(2l+10)(m+m'')+15+n_{\max}(2n'+n'+2m'+m')$	0	6	$3+n_{\max}$	$6+n_{\max}$
edge	$(2l+4)n+(2l+6)m-l-3$	$(2l+8)n+(2l+10)m-l+6$	2	8	3	5
connected	$(2l+6)n+(2l+8)m-(l+2)\ell+3$	$(2l+8)n+(2l+10)m+(3-l)\ell+17$	0	9	5	7
isolated	$nl+ml+4m+\sum_{l_{i^*}=1}^{\ell_{i^*}}(2\lvert E_{l_{i^*}}\rvert+2\lvert\Xi_{\mathcal{V}}\rvert+\sum_{k=1}^{\lvert E_{l_{i^*}}\rvert-1}k-2)+\sum_{l_{j^*}=1}^{\ell_{j^*}}(2\lvert E_{l_{j^*}}\rvert+2\lvert\Xi_{\mathcal{V}}\rvert+\sum_{k=1}^{\lvert E_{l_{j^*}}\rvert-1}k-2)+3\min(\lvert E_{i^*}\rvert,\lvert E_{j^*}\rvert)+6\sum_{l_{i^*}=1}^{\ell_{i^*}}(\lvert E_{l_{i^*}}\rvert-1)+6\sum_{l_{j^*}=1}^{\ell_{j^*}}(\lvert E_{l_{j^*}}\rvert-1)+1$	$nl+n+ml+11m+\sum_{l_{i^*}=1}^{\ell_{i^*}}(2\lvert\Xi_{\mathcal{V}}\rvert+8\lvert E_{l_{i^*}}\rvert+\sum_{k=1}^{\lvert E_{l_{i^*}}\rvert-1}(k+1)+1)+\sum_{l_{j^*}=1}^{\ell_{j^*}}(2\lvert\Xi_{\mathcal{V}}\rvert+8\lvert E_{l_{j^*}}\rvert+\sum_{k=1}^{\lvert E_{l_{j^*}}\rvert-1}(k+1)+1)+\sum_{l_{i^*}=1}^{\ell_{i^*}}(l_{i^*}+1)+\sum_{l_{j^*}=1}^{\ell_{j^*}}(l_{j^*}+1)+3(\min(\lvert E_{i^*}\rvert,\lvert E_{j^*}\rvert))+15$	$3n+\sum_{l_{i^*}=1}^{\ell}\lvert E_{l_{i^*}}\rvert+\sum_{l_{j^*}=1}^{\ell}\lvert E_{l_{j^*}}\rvert+2$	$6n+\sum_{l_{i^*}=1}^{\ell}(\lvert E_{l_{i^*}}\rvert+1)+\sum_{l_{j^*}=1}^{\ell}(\lvert E_{l_{j^*}}\rvert+1)+\max(\lvert E_{i^*}\rvert,\lvert E_{j^*}\rvert)-\min(\lvert E_{i^*}\rvert,\lvert E_{j^*}\rvert)+14$	$n+4(\ell_{i^*}+\ell_{j^*})+2$	$n+5(\ell_{i^*}+\ell_{j^*})+16$

Note: l: label length, n: $\lvert V\rvert$, m: $\lvert E\rvert$, m_0: $\lvert E_0\rvert$, m_1: $\lvert E_1\rvert$, n': $\lvert V'\rvert$, m': $\lvert E'\rvert$, n'': $\lvert V'-V\rvert$, m'': $\lvert E'-E\rvert$, ℓ: threshold, *: the slower alternative, #: the $n_{\max}$ alternative.

Table 5. Efficiency comparison for proofs of knowledge statements.

Statement	Groß [14, 15]		Ours			
			Non-optimized		Optimized	
	Basis	O	**Basis**	$O_{\mathbb{G}_T}$	**Basis**	$O_{\mathbb{G}_1}$
possession	-	$O(n+m)$	-	$O_{\mathbb{G}_T}(n+m)$	-	$O_{\mathbb{G}_1}(n+m)$
vertices	possession	$O(n)$	set	$O_{\mathbb{G}_T}(n)$	set	$O_{\mathbb{G}_1}(n)$
edges	possession	$O(m)$	set	$O_{\mathbb{G}_T}(m)$	set	$O_{\mathbb{G}_1}(m)$
set	vertices	$O(n')$	-	$O_{\mathbb{G}_T}(n')$	-	$O_{\mathbb{G}_1}(n_{\max})$
disjoint	vertices	$O(k^2+n)$	-	$O_{\mathbb{G}_T}(n')$	-	$O_{\mathbb{G}_1}(n')$
cover	vertices	$O(kl)$	-	$O_{\mathbb{G}_T}(n'+n'')$	-	$O_{\mathbb{G}_1}(n_{\max}+n'')$
partition	vertices	$O(k^2+n)$	set	$O_{\mathbb{G}_T}(n)$	set	$O_{\mathbb{G}_1}(n)$
edge	possession	$O(1)$	set	$O_{\mathbb{G}_T}(1)$	set	$O_{\mathbb{G}_1}(1)$
connected	edges	$O(\ell)$	set	$O_{\mathbb{G}_T}(\ell)$	set	$O_{\mathbb{G}_1}(\ell)$
isolated	edges	$O(m)$	edges	$O_{\mathbb{G}_T}(m)$	edges	$O_{\mathbb{G}_1}(m)$

Note: n: $\lvert V\rvert$, $m \leq \frac{n(n-1)}{2}$, $k \leq l \leq n$, n': $\lvert V'\rvert$, n'': $\lvert V'-V\rvert$, ℓ: threshold.

in the practice. Since a point multiplication operation is faster than a pairing, whose operating time is comparable to a modular exponentiation in RSA, our proposed graph signature scheme is more efficient than Groß' in general.

9 Conclusion

In this paper, we proposed a SDH-based graph signature scheme based on MoniPoly encoding. We rigorously proved the security of our graph signature scheme in the standard model with tight reduction. Our graph signature scheme provides more efficient show proofs compared to that of the RSA-based graph signature scheme.

Acknowledgment

This work was supported by the ERC Starting Grant CASCAde (GA n°716980).

Bibliography

[1] Man Ho Au, Willy Susilo, and Yi Mu. Constant-size dynamic k-taa. In Roberto De Prisco and Moti Yung, editors, *Security and Cryptography for Networks*, pages 111–125, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[2] Mihir Bellare and Gregory Neven. Transitive signatures based on factoring and rsa. In *Advances in Cryptology–ASIACRYPT 2002*, pages 397–414. Springer, 2002.

[3] Manuel Blum. How to prove a theorem so no one else can claim it. In *Proceedings of the International Congress of Mathematicians*, volume 1, page 2, 1986.

[4] Dan Boneh and Xavier Boyen. Short signatures without random oracles and the sdh assumption in bilinear groups. *Journal of Cryptology*, 21(2):149–177, Apr 2008.

[5] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matt Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, pages 41–55, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[6] Jan Camenisch and Thomas Groß. Efficient attributes for anonymous credentials. In *Proceedings of the 15th ACM conference on Computer and communications security*, pages 345–356. ACM, 2008.

[7] Jan Camenisch and Thomas Groß. Efficient attributes for anonymous credentials. *ACM Trans. Inf. Syst. Secur.*, 15(1):4:1–4:30, March 2012.

[8] Jan Camenisch and Anna Lysyanskaya. A signature scheme with efficient protocols. In Stelvio Cimato, Giuseppe Persiano, and Clemente Galdi, editors, *Security in Communication Networks*, pages 268–289, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[9] Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In Matt Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, pages 56–72, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[10] Ran Canetti, Omer Paneth, Dimitrios Papadopoulos, and Nikos Triandopoulos. Verifiable set operations over outsourced databases. In Hugo Krawczyk, editor, *Public-Key Cryptography – PKC 2014*, pages 113–130, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[11] Sanjit Chatterjee and Alfred Menezes. On cryptographic protocols employing asymmetric pairings - the role of ψ revisited. *Discrete Applied Mathematics*, 159(13):1311 – 1322, 2011.

[12] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM*, 38(3):690–728, 1991.

[13] T. Groß and Ioannis Sfyrakis. Specification of the graph signature cryptographic library and the prismacloud topology certifi-

cation version 0.9.2. Newcastle University ePrint Archive, 2018. https://www.ncl.ac.uk/media/wwwnclacuk/schoolofcomputingscience/files/trs/1523.pdf.

[14] Thomas Groß. Signatures and efficient proofs on committed graphs and np-statements. In Rainer Böhme and Tatsuaki Okamoto, editors, *Financial Cryptography and Data Security*, pages 293–314, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[15] Thomas R. Groß. Efficient certification and zero-knowledge proofs of knowledge on infrastructure topology graphs. In *Proceedings of the 6th Edition of the ACM Workshop on Cloud Computing Security*, CCSW '14, pages 69–80, New York, NY, USA, 2014. Association for Computing Machinery.

[16] IBM. Specification of the identity mixer cryptographic library, v. 2.3.40. Specification, IBM Research, 2013.

[17] Eike Kiltz, Daniel Masny, and Jiaxin Pan. Optimal security proofs for signatures from identification schemes. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016*, pages 33–61, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[18] Kalikinkar Mandal, Basel Alomair, and Radha Poovendran. Outsourcing graph databases with label-constrain query verification, 2015. http://labs.ece.uw.edu/nsl/papers/GDB-full-15.pdf.

[19] Silvio Micali and Ronald L Rivest. Transitive signature schemes. In *Topics in Cryptology-CT–RSA 2002*, pages 236–243. Springer, 2002.

[20] Charalampos Papamanthou, Roberto Tamassia, and Nikos Triandopoulos. Optimal verification of operations on dynamic sets. In Phillip Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, pages 91–110, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[21] Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In Joan Feigenbaum, editor, *Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings*, volume 576 of *Lecture Notes in Computer Science*, pages 129–140. Springer, 1991.

[22] Sven Schäge. Tight proofs for signature schemes without random oracles. In Kenneth G. Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011*, pages 189–206, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[23] Syh-Yuan Tan and Thomas Groß. Monipoly—an expressive q-SDH-based anonymous attribute-based credential system. In *To appear in proceedings of ASIACRYPT 2020*, May 2020.

[24] Syh-Yuan Tan and Thomas Groß. Monipoly—an expressive q-SDH-based anonymous attribute-based credential system [extended version]. Cryptology ePrint Archive Report 2020/587 (ia.cr/2020/587), IACR, May 2020.

[25] Y. Zhang, J. Katz, and C. Papamanthou. An expressive (zero-knowledge) set accumulator. In *2017 IEEE European Symposium on Security and Privacy (EuroS P)*, pages 158–173, April 2017.

[26] Yupeng Zhang, Charalampos Papamanthou, and Jonathan Katz. Alitheia: Towards practical verifiable graph processing. In *In CCS*, 2014.

[27] Yupeng Zhang, Charalampos Papamanthou, and Jonathan Katz. Verifiable graph processing. *ACM Trans. Priv. Secur.*, 21(4):20:1–20:23, October 2018.

A Optimized Proofs

We present the optimized proofs for the proposed graph signature scheme. The exponents which represent the encoded vertices and edges at $\mathbb{G}_2$ bases $(X_{i_k}, X_{(i,j)_k})$ in a pairing are moved to the $\mathbb{G}_1$ bases $(W_i, W_{(i,j)})$. Therefore, every pairing can now be compressed together by sharing the same X_{i_k}, i.e., reduces the number of pairings from linear to W_i, to linear to X_{i_k}. For instance, the vertex decomposition in the graph statement can be rewritten as:

$$\begin{aligned}\prod_{i \in V} \mathsf{e}\left(a_{i_0}^{o_i r_i^{-1} \prod_{w \in f_{\mathcal{V}}(i_l)}(x'+w)}, X_{0_0}^{r_i(x'+i)}\right) &= \prod_{i \in V} \mathsf{e}\left(W_i', X_{0_1}^{r_i} X_{0_0}^{r_i \cdot i}\right) \\ &= \prod_{i \in V} \mathsf{e}\left(W_i', X_{0_1}^{r_i}\right) \prod_{i \in V} \mathsf{e}\left(W_i', X_{0_0}^{r_i i}\right) \\ &= \prod_{i \in V} \mathsf{e}\left(W_i'^{r_i}, X_{0_1}\right) \prod_{i \in V} \mathsf{e}\left(W_i'^{r_i i}, X_{0_0}\right) \\ &= \mathsf{e}\left(\prod_{i \in V} W_i'^{r_i u_1^{-1}}, X_{0_1}^{u_1}\right) \mathsf{e}\left(\prod_{i \in V} W_i'^{r_i i u_0^{-1}}, X_{0_0}^{u_0}\right)\end{aligned}$$

where $|V|$ pairings are compressed to only two pairings.

Lemma 17. *The randomization in the optimization gives perfectly hiding property.*

Proof. The random exponents u_0, u_1 turns the $\mathbb{G}_T$ elements

$$\mathsf{e}\left(\prod_{i \in V} W_i'^{r_i u_1^{-1}}, X_{0_1}\right)^{u_1} \mathsf{e}\left(\prod_{i \in V} W_i'^{r_i i u_0^{-1}}, X_{0_0}\right)^{u_0}$$

into a Pedersen set commitment (Lemma 8) which is perfectly hiding. □

A.1 bootstrap Proof

The bootstrap statement can be optimized as follows:

$$\begin{aligned}&PK\Bigg\{(\forall i : \{\varepsilon_{i_l}\}_{l=0}^{1}, \forall (i,j) : \{\varepsilon_{(i,j)_l}\}_{l=0}^{2}, \mu_0, \mu_1, \mu_2) : \\ &\mathsf{e}\left(C \prod_{i \in V} C_i \prod_{(i,j) \in E} C_{(i,j)}, X_{0_0}\right) = \prod_{k=0}^{1} \mathsf{e}\left(\prod_{i \in V} (W_i' W_i)^{\varepsilon_{i_k}}, X_{0_k}^{\mu_k}\right) \cdot \\ &\qquad \prod_{k=0}^{2} \mathsf{e}\left(\prod_{(i,j) \in E} (W_{(i,j)}' W_{(i,j)})^{\varepsilon_{(i,j)_k}}, X_{0_k}^{\mu_k}\right) \wedge \\ &\mathsf{e}\left(\prod_{i \in V} W_i' \prod_{(i,j) \in E} W_{(i,j)}', X_{0_0}\right) = \prod_{i \in V} \mathsf{e}\left(W_i, X_{i_0}\right) \prod_{(i,j) \in E} \mathsf{e}\left(W_{(i,j)}, X_{(i,j)_0}\right)\Bigg\}\end{aligned}$$

where the number of pairings are reduced from $2+2(|V|+|E|)$ to $7+|V|+|E|$. We can further compress the proof to save another 3 pairings but we present it in the current form for clarity purposes.

A.2 vertices Proof

Let $V = \bigcup\{V_1, \ldots, V_\ell\}$ be the vertex set in a graph $\mathcal{G}$, the prover runs the following protocol:

$$
\begin{aligned}
&PK\Bigg\{(\{\varepsilon_{l_0}, \varepsilon_{l_1}\}_{l=1}^{\ell}, \mu_0, \mu_1) : \mathsf{e}\left(\prod_{i\in V} C_i, X_{0_0}\right) = \mathsf{e}\left(\prod_{l=1}^{\ell} W_i^{\varepsilon_{i_1}}, X_{0_1}^{\mu_1}\right)\mathsf{e}\left(\prod_{l=1}^{\ell} W_i^{\varepsilon_{i_0}}, X_{0_0}^{\mu_0}\right) \wedge \\
&\mathsf{e}\left(W_{\mathcal{V}_1}, X_{0_0}\right) = \mathsf{e}\left(a_{0_0}^{\varepsilon_{1_1}}, X_{0_1}^{\mu_1}\right)\mathsf{e}\left(a_{0_0}^{\varepsilon_{1_0}}, X_{0_0}^{\mu_0}\right) \wedge \mathsf{e}\left(a_{0_0}, W_{\mathcal{L}_1}\right) = \mathsf{e}\left(W_1, X_{0_0}\right) \wedge \\
&\mathsf{e}\left(W_{\mathcal{V}_2}, X_{0_0}\right) = \mathsf{e}\left(W_{\mathcal{V}_1}^{\varepsilon_{2_1}}, X_{0_1}^{\mu_1}\right)\mathsf{e}\left(W_{\mathcal{V}_1}^{\varepsilon_{2_0}}, X_{0_0}^{\mu_0}\right) \wedge \\
&\mathsf{e}\left(W_{\mathcal{V}_3}, X_{0_0}\right) = \mathsf{e}\left(W_{\mathcal{V}_2}^{\varepsilon_{3_1}}, X_{0_1}^{\mu_1}\right)\mathsf{e}\left(W_{\mathcal{V}_2}^{\varepsilon_{3_0}}, X_{0_0}^{\mu_0}\right) \wedge \cdots \\
&\cdots \wedge \mathsf{e}\left(a_{0_0}, W_{\mathcal{V}_\ell}\right) = \mathsf{e}\left(W_{\mathcal{V}_{\ell-1}}^{\varepsilon_{\ell_1}}, X_{0_1}^{\mu_1}\right)\mathsf{e}\left(W_{\mathcal{V}_{\ell-1}}^{\varepsilon_{\ell_0}}, X_{0_0}^{\mu_0}\right) \wedge \\
&\mathsf{e}\left(\prod_{k=0}^{|\varXi_{\mathcal{V}}|} a_{0_k}^{\mathsf{m}_{\mathcal{V}_k}}, X_{0_0}\right) = \mathsf{e}\left(W_{\varXi_{\mathcal{V}}\setminus V}, W_{\mathcal{V}_\ell}\right) \wedge \\
&\mathsf{e}\left(a_{0_0}, W_{\mathcal{L}_2}\right) = \mathsf{e}\left(W_2, W_{\mathcal{L}_1}\right) \wedge \mathsf{e}\left(a_{0_0}, W_{\mathcal{L}_3}\right) = \mathsf{e}\left(W_3, W_{\mathcal{L}_2}\right) \wedge \\
&\cdots \wedge \mathsf{e}\left(a_{0_0}, W_{\mathcal{L}_\ell}\right) = \mathsf{e}\left(W_\ell, W_{\mathcal{L}_{\ell-1}}\right) \wedge \mathsf{e}\left(\prod_{k=0}^{|\varXi_{\mathcal{L}}|} a_{0_k}^{\mathsf{m}_{\mathcal{L}_k}}, X_{0_0}\right) = \mathsf{e}\left(W_{\varXi_{\mathcal{L}}\setminus f_{\mathcal{V}}(V)}, W_{\mathcal{L}_\ell}\right) \\
&\Bigg\}
\end{aligned}
$$

and the correctness can be observed from the following:

$$
\begin{aligned}
\mathsf{e}(C_{i_l}, X_{0_0}) &= \mathsf{e}\left(a_{0_0}^{o_l(x'+i_l)\prod_{w\in f_{\mathcal{V}}(i_l)}(x'+w)}, X_{0_0}\right) \\
&= \mathsf{e}\left(a_{0_0}^{o_l\cdot r_l^{-1}\prod_{w\in f_{\mathcal{V}}(i_l)}(x'+w)}, X_{0_0}^{r_l(x'+i_l)}\right) \\
&= \mathsf{e}\left(W_l, X_{0_1}^{r_l} X_{0_0}^{r_l i_l}\right) \\
&= \mathsf{e}(W_l^{r_l u_1^{-1}}, X_{0_1}^{u_1})\mathsf{e}(W_l^{r_l i_l u_0^{-1}}, X_{0_0}^{u_0}) \\
&= \mathsf{e}\left(W_l^{\varepsilon_{l_1}}, X_{0_1}^{\mu_1}\right)\mathsf{e}\left(W_l^{\varepsilon_{l_0}}, X_{0_0}^{\mu_0}\right)
\end{aligned}
$$

and,

$$\begin{aligned}
\mathsf{e}\left(W_{\mathcal{V}_l}, X_{0_0}\right) &= \mathsf{e}\left(a_{0_0}^{r_l(x'+i_l)\prod_{k=1}^{l-1} r_k(x'+i_k)}, X_{0_0}\right)\\
&= \mathsf{e}\left(a_{0_0}^{\prod_{k=1}^{l-1} r_k(x'+l_k)}, X_{0_0}^{r_l(x'+i_l)}\right)\\
&= \mathsf{e}\left(W_{\mathcal{V}_{l-1}}, X_{0_1}^{r_l x'} X_{0_0}^{r_l i_l}\right)\\
&= \mathsf{e}\left(W_{\mathcal{V}_{l-1}}^{r_l u_1^{-1}}, X_{0_1}^{u_1}\right) \mathsf{e}\left(W_{\mathcal{V}_{l-1}}^{r_l i_l u_0^{-1}}, X_{0_0}^{u_0}\right)\\
&= \mathsf{e}\left(W_{\mathcal{V}_{l-1}}^{\varepsilon_{l_1}}, X_{0_1}^{\mu_1}\right) \mathsf{e}\left(W_{\mathcal{V}_{l-1}}^{\varepsilon_{l_0}}, X_{0_0}^{\mu_0}\right)
\end{aligned}$$

and,

$$\begin{aligned}
\mathsf{e}\left(a_{0_0}, W_{\mathcal{L}_l}\right) &= \mathsf{e}\left(a_{0_0}, X_{0_0}^{o_l \cdot r_l^{-1} \prod_{w\in f_{\mathcal{V}}(i_l)}(x'+w) \prod_{k=1}^{l-1} o_k \cdot r_k^{-1} \prod_{w\in f_{\mathcal{V}}(i_k)}(x'+w)}\right)\\
&= \mathsf{e}\left(a_{0_0}^{o_l \cdot r_l^{-1} \prod_{w\in f_{\mathcal{V}}(i_l)}(x'+w)}, X_{0_0}^{\prod_{k=1}^{l-1} o_k \cdot r_k^{-1} \prod_{w\in f_{\mathcal{V}}(i_k)}(x'+w)}\right)\\
&= \mathsf{e}\left(W_l, W_{\mathcal{L}_{l-1}}\right)
\end{aligned}$$

where $\{\mathsf{m}_{\mathcal{V}k}\} = \mathsf{MPEncode}(\mathit{\Xi}_{\mathcal{V}}), \{\mathsf{m}_{\mathcal{L}_{\mathcal{V}}k}\} = \mathsf{MPEncode}(\mathit{\Xi}_{\mathcal{L}_{\mathcal{V}}}), W_l = \prod_{k=0}^{|f_{\mathcal{V}}(i_l)|} a_{0_k}^{\mathsf{m}_{l_k}}$, $\varepsilon_{l_0} = r_l i_l u_0^{-1}, \varepsilon_{l_1} = r_l u_1^{-1}$ and $\{\mathsf{m}_{l_k}\} = o_l \cdot r_l^{-1} \times \mathsf{MPEncode}(f_{\mathcal{V}}(i_l))$ for randomly selected $o_l, r_l, u_1 \in \mathbb{Z}_p^*$. The public inputs $(W_1, \ldots, W_\ell)$ are witnesses for the vertex labels, $(W_{\mathcal{V}_1}, \ldots, W_{\mathcal{V}_\ell}, W_{\mathit{\Xi}_{\mathcal{V}}\setminus V})$ are witnesses for the cumulative product of vertex identifiers while $(W_{\mathcal{L}_1}, \ldots, W_{\mathcal{L}_\ell}, W_{\mathit{\Xi}_{\mathcal{L}}\setminus f_{\mathcal{V}}(V)})$ are witnesses for the cumulative product of vertex labels. Subsequently, simplifying the pairing notations in the proof above gives:

$$\begin{aligned}
&PK\Bigg\{(\{\varepsilon_{l_0}, \varepsilon_{l_1}\}_{l=1}^{\ell}, \mu_0, \mu_1) : \mathsf{e}\left(\prod_{i\in V} C_i, X_{0_0}\right) = \prod_{k=0}^{1} \mathsf{e}\left(\prod_{l=1}^{\ell} W_l^{\varepsilon_{l_k}}, X_{0_k}^{\mu_k}\right) \wedge\\
&\mathsf{e}\left(\prod_{k=0}^{|\mathit{\Xi}_{\mathcal{V}}|} a_{0_k}^{\mathsf{m}_{\mathcal{V}k}} \prod_{l=1}^{\ell-1} W_{\mathcal{V}_l}, X_{0_0}\right) =\\
&\qquad\qquad \prod_{k=0}^{1} \mathsf{e}\left(\prod_{l=1}^{\ell} W_{\mathcal{V}_{l-1}}^{\varepsilon_{l_k}}, X_{0_k}^{\mu_k}\right) \mathsf{e}\left(a_{0_0}^{-1} W_{\mathit{\Xi}_{\mathcal{V}}\setminus V}, W_{\mathcal{V}_\ell}\right) \wedge\\
&\mathsf{e}\left(\prod_{k=0}^{|\mathit{\Xi}_{\mathcal{L}}|} a_{0_k}^{\mathsf{m}_{\mathcal{L}k}}, X_{0_0}\right) \mathsf{e}\left(a_{0_0}, \prod_{l=1}^{\ell} W_{\mathcal{L}_l}\right) =\\
&\qquad\qquad \prod_{l=1}^{\ell} \mathsf{e}\left(W_l, W_{\mathcal{L}_{l-1}}\right) \mathsf{e}\left(W_{\mathit{\Xi}_{\mathcal{L}}\setminus f_{\mathcal{V}}(V)}, W_{\mathcal{L}_\ell}\right)\Bigg\}
\end{aligned}$$

as the $\mathsf{vertices}(V)$ statement where $W_{\mathcal{V}_0} = a_{0_0}$. Notice that this optimization decreases the number of pairing in $\mathsf{vertices}(V)$ from $5+3n$ into $10+n$ only.

A.3 edges Proof

Let $E = \bigcup\{E_1, \dots, E_\ell\}$ be the edge set in a graph $\mathcal{G}$, the protocol below establishes the optimized statement $\mathsf{edges}(E)$:

$$
\begin{aligned}
&PK\Bigg\{ (\forall (i,j) \in E : \varepsilon_{(i,j)_0}, \varepsilon_{(i,j)_1}, \varepsilon_{(i,j)_2}, \mu_0, \mu_1, \mu_2) : \\
&\mathsf{e}\left(\prod_{(i,j)\in E} C_{(i,j)}, X_{0_0} \right) = \prod_{k=0}^{2} \mathsf{e}\left(\prod_{(i,j)\in E} W_{(i,j)}^{\varepsilon_{(i,j)_k}}, X_{0_k}^{\mu_k} \right) \wedge \\
&\mathsf{e}\left(\prod_{l=1}^{\ell} \left(\prod_{k=0}^{|\Xi_\mathcal{V}|} a_{0_k}^{\mathsf{m}_{\mathcal{V}_k}} \prod_{w_l=1}^{|E_l|-1} W_{\mathcal{E}_{w_l}} \right), X_{0_0} \right) = \\
&\qquad \prod_{l=1}^{\ell} \prod_{k=0}^{2} \mathsf{e}\left(\prod_{w_l=1}^{|E_l|} W_{\mathcal{E}_{w_1-1}}^{\varepsilon_{(i,j)_k}}, X_{0_k}^{\mu_k} \right) \mathsf{e}\left(a_{0_0}^{-1} W_{\Xi_\mathcal{V} \setminus E_l}, W_{\mathcal{E}_{|E_l|}} \right) \wedge \\
&\mathsf{e}\left(\prod_{k=0}^{|\Xi_\mathcal{L}|} a_{0_k}^{\mathsf{m}_{\mathcal{L}_k}}, X_{0_0} \right) \mathsf{e}\left(a_{0_0}, \prod_{l=(2,3)}^{(\ell-2,\ell-1)} W_{\mathcal{L}_l} \right) = \\
&\qquad \mathsf{e}\left(W_{(2,3)}, X_{0_0} \right) \prod_{l=(4,5)}^{(\ell-2,\ell-1)} \mathsf{e}\left(W_l, W_{\mathcal{L}_{l-1}} \right) \mathsf{e}\left(W_{\Xi_\mathcal{L} \setminus f_\mathcal{E}(E_1)}, W_{\mathcal{L}_{(\ell-2,\ell-1)}} \right) \Bigg\}
\end{aligned}
$$

and can be rewritten as:

$$
\begin{aligned}
&PK\Bigg\{ (\forall (i,j) \in E : \varepsilon_{(i,j)_0}, \varepsilon_{(i,j)_1}, \varepsilon_{(i,j)_2}, \mu_0, \mu_1, \mu_2) : \\
&\quad \mathsf{e}\left(\prod_{(i,j)\in E} C_{(i,j)}, X_{0_0} \right) = \prod_{k=0}^{2} \mathsf{e}\left(\prod_{(i,j)\in E} W_{(i,j)}^{\varepsilon_{(i,j)_k}}, X_{0_k}^{\mu_k} \right) \wedge \\
&\quad \varepsilon_{(i,j)}[C_{(i,j)_{E_1}}] \in \Xi_\mathcal{V} \wedge \cdots \wedge \varepsilon_{(i,j)}[C_{(i,j)_{E_l}}] \in \Xi_\mathcal{V} \wedge \\
&\quad W_{(i,j)}[C_{(i,j)}] \in \Xi_\mathcal{L} \qquad \Bigg\}
\end{aligned}
$$

for undirected acyclic graph. On the other hand, the proof for a directed acyclic graph is as follows:

$$PK\Big\{((\forall(i,j,j) \in E : \varepsilon_{(i,j,j)_0}, \varepsilon_{(i,j,j)_1}, \varepsilon_{(i,j,j)_2}, \varepsilon_{(i,j,j)_3}, \varepsilon_{(i,j)_0}, \varepsilon_{(i,j)_1}, \varepsilon_{(i,j)_2}, \varepsilon_{j_0}, \varepsilon_{j_1}),$$

$$\mu_0, \mu_1, \mu_2, \mu_3) :$$

$$\mathsf{e}\left(\prod_{(i,j,j)\in E} C_{(i,j,j)}, X_{0_0}\right) = \prod_{k=0}^{3} \mathsf{e}\left(\prod_{(i,j,j)\in E} W_{(i,j,j)}^{\varepsilon_{(i,j,j)_k}}, X_{0_k}^{\mu_k}\right) \wedge$$

$$W_{(i,j,j)}[C_{(i,j,j)}] \in \Xi_{\mathcal{L}} \wedge$$

$$\prod_{k=0}^{3} \mathsf{e}\left(\prod_{(i,j,j)\in E} a_{0_0}^{\varepsilon_{(i,j,j)_k}}, X_{0_k}\right) = \prod_{k=0}^{1} \mathsf{e}\left(\prod_{(i,j,j)\in E} W_{(i,j)}^{\varepsilon_{j_k}}, X_{0_k}^{\mu_k}\right) \wedge$$

$$W_{(i,j)}[C_{(i,j,j)_{E_1}}] \in \Xi_{\mathcal{V}} \wedge \cdots \wedge W_{(i,j)}[C_{(i,j,j)_{E_l}}] \in \Xi_{\mathcal{V}} \wedge \varepsilon_j[C_{(i,j,j)}] \in \Xi_{\mathcal{V}}\Big\}$$

where we also prove the extra vertex identifiers in every edge identifier of a directed acyclic graph. Similar to the original version, the optimized proof above is applicable to the directed cyclic graph.

A.4 Intermediary Signing

The optimized intermediary signing protocol uses the optimized proof of representation during the first step:

1. User randomly selects $r_1, y, o_{i_1}, o_{(i,j)_1}, r_i, r_{(i,j),\mu_0,\mu_1} \in \mathbb{Z}_p^*$ and interacts with the signer to prove the possession of σ_{init} and the representation of his hidden graph $\mathcal{G}_{\mathsf{U}}$:

$$PK\Big\{((\forall i \in V_{\mathsf{U}} : \varepsilon_{i_0}, \varepsilon_{i_1}, \varepsilon_i), (\forall (i,j) \in E_{\mathsf{U}} : \varepsilon_{(i,j)_0}, \varepsilon_{(i,j)_1}, \varepsilon_{(i,j)}),$$

$$\zeta, \rho, \omega, \tau, \gamma, \mu_0, \mu_1) :$$

$$\mathsf{e}\left(C_1 C_2 c^{\omega} v'^{-\tau}, X_{0_0}\right) = \mathsf{e}(v'^{\gamma}, X) \wedge$$

$$C_1 C_2 = h^{\zeta} b^{\rho} \prod_{i\in V_{\mathsf{U}}} C_i^{\varepsilon_i} \prod_{(i,j)\in E_{\mathsf{U}}} C_{(i,j)}^{\varepsilon_{(i,j)}} \wedge$$

$$\mathsf{e}\left(\prod_{i\in V_{\mathsf{U}}} C_i \prod_{(i,j)\in E_{\mathsf{U}}} C_{(i,j)}, X_{0_0}\right) =$$

$$\prod_{k=0}^{1} \mathsf{e}\left(\prod_{i\in V_{\mathsf{U}}} W_i'^{\varepsilon_{i_k}} \prod_{i\in E_{\mathsf{U}}} W_{(i,j)}'^{\varepsilon_{(i,j)_k}}, X_{0_k}^{\mu_k}\right)\Big\}$$

where $\varepsilon_{i_1} = r_i \mu_1^{-1}, \varepsilon_{i_0} = r_i \mu_0^{-1} i, \varepsilon_{(i,j)_1} = r_{(i,j)} \mu_1^{-1}, \varepsilon_{(i,j)_1} = r_{(i,j)} \mu_0^{-1} i$ while the witnesses are

$$W_i' = a_{i_0}^{o_{i_1} r_i^{-1} \prod_{w\in f_{\mathcal{V}}(i)}(x'+w)}$$

and

$$W'_{(i,j)} = a_{(i,j)_0}^{o_{(i,j)_1} r_{(i,j)}^{-1} (x'+j) \prod_{w \in f_{\mathcal{E}}(i,j)} (x'+w)}.$$

The same changes also appear in the final signing protocol.

A.5 Proof of Possession

The optimized proof for the possession statement is as follows:

$$PK\Bigg\{ (\forall i \in V : \varepsilon_{i_0}, \varepsilon_{i_1}), (\forall (i,j) \in E : \varepsilon_{(i,j)_0}, \varepsilon_{(i,j)_1}), \zeta, \rho, \omega, \tau, \gamma, \mu_0, \mu_1) :$$
$$\prod_{k=0}^{1} \mathsf{e}\left(\prod_{i \in V} W_i'^{\varepsilon_{i_k}} \prod_{(i,j) \in E} W_{(i,j)}'^{\varepsilon_{(i,j)_k}}, X_{0_k}^{\mu_k} \right) \mathsf{e}\left(h^{\zeta} b^{\rho} c^{\omega} v'^{-\tau}, X_{0_0}\right) = \mathsf{e}(v'^{\gamma}, X) \Bigg\}$$

where only 4 pairings are involved, instead of $|V| + |E| + 2$ pairings.

A.6 set Proof

The optimized set proof is as follows:

$$PK\Bigg\{ ((\forall i \in V \setminus V' : \varepsilon_{i_0}, \varepsilon_{i_1}), (\forall (i,j) \in E \setminus E' : \varepsilon_{(i,j)_0}, \varepsilon_{(i,j)_1}), \zeta, \rho, \omega, \tau, \gamma, \mu_0, \mu_1) :$$
$$\prod_{k=0}^{1} \mathsf{e}\left(\prod_{i \in V \setminus V'} W_i'^{\varepsilon_{i_k}} \prod_{(i,j) \in E \setminus E'} W_{(i,j)}'^{\varepsilon_{(i,j)_k}}, X_{0_k}^{\mu_k} \right) \cdot$$
$$\prod_{i \in V'} \mathsf{e}\left(W_i', \prod_{k=0}^{|V_i'|} X_{0_k}^{\mathsf{m}_{i_k}} \right) \prod_{(i,j) \in E'} \mathsf{e}\left(W_{(i,j)}', \prod_{k=0}^{|E'_{(i,j)}|} X_{0_k}^{\mathsf{m}_{(i,j)_k}} \right) \cdot$$
$$\mathsf{e}\left(h^{\zeta} b^{\rho} c^{\omega} v'^{-\tau}, X_{0_0}\right) = \mathsf{e}(v'^{\gamma}, X) \Bigg\}$$

where $\{\mathsf{m}_{i_j}\} = \mathsf{MPEncode}(V_i'), \{\mathsf{m}_{(i,j)_j}\} = \mathsf{MPEncode}(E'_{(i,j)}), \zeta = r \times sk_{\mathsf{U}}, \rho = s', \omega = r, \varepsilon_{i_1} = r_i, \varepsilon_{(i,j)_1} = r_{(i,j)}, \tau = t', \gamma = y$ for the randomly chosen blinding factors $r, y, r_i, r_{(i,j)} \in \mathbb{Z}_p^*$.

Noticed that the second line in the set proof above is the same as that in the original proof. If we apply the optimization mechanism on it, we can compress the $|V'| + |E'|$ pairings into a possibly smaller number of pairings which is linear to the size of the largest encoded vertex or edge in $\mathcal{G}'$. Let $n_{\max} = \max(\max(|V_1'|, \ldots, |V'_{|V'|}|), \max(|E_1'|, \ldots, |E'_{|E'|}|))$, if $n_{\max} < |\mathcal{G}'|$, the verifier can choose to verify the set proof as below:

$$PK\Bigg\{((\forall i \in V \setminus V' : \varepsilon_{i_0}, \varepsilon_{i_1}), (\forall (i,j) \in E \setminus E' : \varepsilon_{(i,j)_0}, \varepsilon_{(i,j)_1}), \zeta, \rho, \omega, \tau, \gamma, \mu_0, \mu_1) :$$

$$\prod_{k=0}^{1} \mathsf{e}\left(\prod_{i \in V \setminus V'} W_i'^{\varepsilon_{i_k}} \prod_{(i,j) \in E \setminus E'} W_{(i,j)}'^{\varepsilon_{(i,j)_k}}, X_{0_k}^{\mu_k} \right) \cdot$$

$$\prod_{k=0}^{n_{\max}} \mathsf{e}\left(\prod_{i \in V'} W_i'^{\mathsf{m}_{i_k}} \prod_{(i,j) \in E'} W_{(i,j)}'^{\mathsf{m}_{(i,j)_k}}, X_{0_k} \right) \cdot$$

$$\mathsf{e}\left(h^{\zeta} b^{\rho} c^{\omega} v'^{-\tau}, X_{0_0}\right) = \mathsf{e}(v'^{\gamma}, X)\Bigg\}$$

where $\mathsf{m}_{i_k} = 0, \mathsf{m}_{(i,j)_k} = 0$ whenever $|V_i'| < k$ and $|E_{(i,j)}'| < k$, respectively. The correctness for $\mathcal{G}'$ can be verified from the following:

$$\begin{aligned}
&\mathsf{e}\left(\prod_{i \in V'} C_i \prod_{(i,j) \in E'} C_{(i,j)}, X_{0_0} \right) \\
&= \mathsf{e}\left(\prod_{i \in V'} a_{i_0}^{r(x'+i)\prod_{w \in f_{\mathcal{V}}(i)}(x'+w)} \prod_{(i,j) \in E'} a_{(i,j)_0}^{r(x'+i)(x'+j)\prod_{w \in f_{\mathcal{E}}(i,j)}(x'+w)}, X_{0_0} \right) \\
&= \prod_{i \in V'} \mathsf{e}\left(a_{i_0}^{r}, X_{0_0}^{(x'+i)\prod_{w \in f_{\mathcal{V}}(i)}(x'+w)} \right) \prod_{(i,j) \in E'} \mathsf{e}\left(a_{(i,j)_0}^{r}, X_{0_0}^{(x'+i)(x'+j)\prod_{w \in f_{\mathcal{E}}(i,j)}(x'+w)} \right) \\
&= \prod_{i \in V'} \mathsf{e}\left(W_i', \prod_{k=0}^{|V_i'|} X_{0_k}^{\mathsf{m}_{i_k}} \right) \prod_{(i,j) \in E'} \mathsf{e}\left(W_{(i,j)}', \prod_{k=0}^{|E_{(i,j)}'|} X_{0_k}^{\mathsf{m}_{(i,j)_k}} \right) \\
&= \prod_{i \in V'} \prod_{k=0}^{|V_i'|} \mathsf{e}\left(W_i'^{\mathsf{m}_{i_k}}, X_{0_k} \right) \prod_{(i,j) \in E'} \prod_{k=0}^{|E_{(i,j)}'|} \mathsf{e}\left(W_{(i,j)}'^{\mathsf{m}_{(i,j)_k}}, X_{0_k} \right) \\
&= \prod_{k=0}^{\max(|V_1'|,\ldots,|V_{|V'|}'|)} \mathsf{e}\left(\prod_{i \in V'} W_i'^{\mathsf{m}_{i_k}}, X_{0_k} \right) \prod_{k=0}^{\max(|E_1'|,\ldots,|E_{|E'|}'|)} \mathsf{e}\left(\prod_{(i,j) \in E'} W_{(i,j)}'^{\mathsf{m}_{(i,j)_k}}, X_{0_k} \right) \\
&= \prod_{k=0}^{n_{\max}} \mathsf{e}\left(\prod_{i \in V'} W_i'^{\mathsf{m}_{i_k}} \prod_{(i,j) \in E'} W_{(i,j)}'^{\mathsf{m}_{(i,j)_k}}, X_{0_k} \right)
\end{aligned}$$

and we have $\{W_i' = a_{i_0}^{o_i r}, W_{(i,j)}' = a_{(i,j)_0}^{o_{(i,j)} r}\}$ if $\{V_i', E_{(i,j)}'\} \in \mathcal{G}_{\mathsf{U}}$.

A.7 cover Proof

The optimized protocol can be executed as follows:

$$PK\Bigg\{(\forall i \in V : \varepsilon_{i_0}, \varepsilon_{i_1}), (\forall (i,j) \in E : \varepsilon_{(i,j)_0}, \varepsilon_{(i,j)_1}), (\forall i \in V' - V : \varepsilon_{i_0}, \varepsilon_{i_1}),$$
$$(\forall (i,j) \in E' - E : \varepsilon_{(i,j)_0}, \varepsilon_{(i,j)_1}), \zeta, \rho, \omega, \tau, \gamma, \mu_0, \mu_1) :$$
$$\prod_{k=0}^{1} \mathsf{e}\left(\prod_{i \in V} W_i'^{\varepsilon_{i_k}} \prod_{(i,j) \in E} W_{(i,j)}'^{\varepsilon_{(i,j)_k}}, X_{0_k}^{\mu_k}\right) \mathsf{e}\left(h^{\zeta} b^{\rho} c^{\omega} v'^{-\tau}, X_{0_0}\right) = \mathsf{e}(v'^{\gamma}, X) \wedge$$
$$\prod_{k=0}^{1} \mathsf{e}\left(\prod_{i \in V'} W_i'^{\varepsilon_{i_k}} \prod_{(i,j) \in E'} W_{(i,j)}'^{\varepsilon_{(i,j)_k}}, X_{0_k}^{\mu_k}\right) =$$
$$\prod_{i \in V'} \mathsf{e}\left(W_i'', \prod_{k=0}^{|V_i'|} X_{0_k}^{\mathsf{m}_{i_k}}\right) \prod_{(i,j) \in E'} \mathsf{e}\left(W_{(i,j)}'', \prod_{k=0}^{|E_{(i,j)}'|} X_{0_k}^{\mathsf{m}_{(i,j)_k}}\right)\Bigg\}$$

where $\{\mathsf{m}_{i_k}\} = \mathsf{MPEncode}(V_i'), \{\mathsf{m}_{(i,j)_k}\} = \mathsf{MPEncode}(E_{(i,j)}'), \zeta = r \times sk_{\mathsf{U}}, \rho = s', \omega = r, \varepsilon_{i_1} = r_i u_1^{-1}, \varepsilon_{(i,j)_1} = r_{(i,j)} u_1^{-1}, \tau = t', \gamma = y, \mu_0 = u_0, \mu_1 = u_1$ for the randomly chosen blinding factors $r, y, r_i, r_{(i,j)}, u_0, u_1 \in \mathbb{Z}_p^*$. If $\mathcal{G}' \in \mathcal{G}_{\mathsf{U}}$ and $n_{\max} < |\mathcal{G}'|$ where $n_{\max} = \max(\max(|V_1'|, \ldots, |V_{|V'|}'|), \max(|E_1'|, \ldots, |E_{|E'|}'|))$, the verifier can replace the right hand side of the last statement with

$$\prod_{k=0}^{n_{\max}} \mathsf{e}\left(\prod_{i \in V'} W_i''^{\mathsf{m}_{i_k}} \prod_{(i,j) \in E'} W_{(i,j)}''^{\mathsf{m}_{(i,j)_k}}, X_{0_k}\right)$$

during verification.

A.8 disjoint Proof

Prover can interact with the verifier to construct the optimized $\mathsf{disjoint}$ proof as follows:

$$PK\Bigg\{((\forall i \in V : \varepsilon_{i_0}, \varepsilon_{i_1}), (\forall (i,j) \in E : \varepsilon_{(i,j)_0}, \varepsilon_{(i,j)_1}), \zeta, \rho, \omega, \tau, \gamma, \mu_0, \mu_1, \{\alpha_k\}_{k=0}^{|V'|-1}) :$$

$$\prod_{k=0}^{1} \mathsf{e}\left(\prod_{i \in V} W_i'^{\varepsilon_{i_k}}, X_{0_k}^{\mu_k}\right) \prod_{(i,j)\in E} \mathsf{e}\left(W_{(i,j)}, \prod_{k=0}^{1} X_{(i,j)_k}^{\varepsilon_{(i,j)_k}}\right) \cdot$$

$$\mathsf{e}\left(h^{\zeta} b^{\rho} c^{\omega} v'^{-\tau}, X_0\right) = \mathsf{e}(v'^{\gamma}, X) \wedge$$

$$\mathsf{e}\left(\prod_{i\in V} W_i', X_{0_0}\right) = \prod_{i \in V} \mathsf{e}\left(W_i, X_{i_0}\right) \wedge \varepsilon_i[C_i] \in \Xi_{\mathcal{V}} \wedge$$

$$\mathsf{e}\left(a_{0_0}, W_{\mathcal{V}_\ell}\right) = \mathsf{e}\left(W_V, \prod_{k=0}^{|V'|} X_{0_0}^{\mathsf{m}_k}\right) \mathsf{e}\left(\prod_{k=0}^{|V'|-1} a_{0_0}^{\alpha_k}, X_{0_0}\right) \wedge$$

$$1_{\mathbb{G}} \neq W_V \wedge 1_{\mathbb{G}} \neq \prod_{k=0}^{|V'|-1} a_{0_0}^{\alpha_k} \Bigg\}$$

where $\{\mathsf{m}_k\} = \mathsf{MPEncode}(\forall \bar{i} \in V')$. Next, the optimized $\mathsf{disjoint}^*$ proof is as follows:

$$PK\Bigg\{(\forall i \in V : \varepsilon_{i_0}, \varepsilon_{i_1}), (\forall (i,j) \in E : \varepsilon_{(i,j)_0}, \varepsilon_{(i,j)_1}), (\forall \bar{i} \in V' : \{\alpha_i\}_{i=1}^{|V|}),$$

$$\zeta, \rho, \omega, \tau, \gamma) :$$

$$\prod_{i\in V} \mathsf{e}\left(W_i, \prod_{k=0}^{1} X_{i_k}^{\varepsilon_{i_k}}\right) \prod_{(i,j)\in E} \mathsf{e}\left(W_{(i,j)}, \prod_{k=0}^{1} X_{(i,j)_k}^{\varepsilon_{(i,j)_k}}\right) \cdot$$

$$\mathsf{e}\left(h^{\zeta} b^{\rho} c^{\omega} v'^{-\tau}, X_{0_0}\right) = \mathsf{e}(v'^{\gamma}, X) \wedge$$

$$\forall \bar{i} \in V' :$$

$$\prod_{i\in V} \mathsf{e}\left(W_i, \prod_{k=0}^{1} X_{i_k}^{\varepsilon_{i_k}}\right) =$$

$$\mathsf{e}\left(\prod_{i\in V} W_i, X_{0_1}\right) \mathsf{e}\left(\left(\prod_{i\in V} W_i\right)^{\bar{i}} a_{i,0}^{\alpha_i}, X_{0_0}\right) \wedge 1_{\mathbb{G}} \neq a_{i_0}^{\alpha_i} \Bigg\}$$

where $v', \{W_i, W_{(i,j)}\}$ are the public inputs and $\zeta = r \times sk_{\mathsf{U}}, \rho = s', \omega = r, \varepsilon_{i_1} = r_i, \varepsilon_{(i,j)_1} = r_{(i,j)}, \tau = t'$ and $\gamma = y$ for the randomly chosen blinding factors $r, y, r_i, r_{(i,j)} \in \mathbb{Z}_p^*$.

The right hand side of second statement in the original proof requires $2|V'|$ pairings yet this optimized version requires only $1 + |V'|$ pairings. This is because $\mathsf{e}\left(\prod_{i\in V} W_i, X_{0_1}\right)$ can be accumulated for every $\bar{i}$ while the $|V'|$ elements of

$\mathsf{e}\left(\left(\prod_{i\in V} W_i\right)^{\bar{i}} a_{i,0}^{\alpha_i}, X_{0_0}\right)$ are sufficient to avoid the conflicts with the last statements.

Table 4 shows a similar complexity relationship for disjoint* and disjoint predicates. Using the same setting from Section 6.10, the disjoint* has a total of $(30m + (5n+23)n' + 29n + 18)M_1$ while disjoint has a total of $(27m + 1/2n(n+71) + |\varXi_{\mathcal{V}}| + 5n' + 192)M_1$. At $n = 100, m = 1000$, we get $(5n' + 36742)M_1$ for disjoint and $(516n' + 32942)M_1$ for disjoint*. The disjoint predicate is faster than disjoint* when $n' = |V'| \geq 7$. On the other hand, at $n = 1000, m = 100$, we get $(5n' + 539392)M_1$ for disjoint and $(5016n' + 32042)M_1$ for disjoint* which then produces a threshold $n' \geq 101$.

A.9 edge(i, j)

Let $E' = (i^*, j^*)$, the edge(i, j) statement can be proved as below:

$$PK\Bigg\{(\forall i \in V : \varepsilon_{i_0}, \varepsilon_{i_1}), (\forall (i,j) \in E \setminus (i^*, j^*) : \varepsilon_{(i,j)_0}, \varepsilon_{(i,j)_1}), \zeta, \rho, \omega, \tau, \gamma, \mu_0, \mu_1) :$$

$$\prod_{k=0}^{1} \mathsf{e}\left(\prod_{i\in V} W_i'^{\varepsilon_{i_k}} \prod_{(i,j)\in E\setminus E'} W_{(i,j)}'^{\varepsilon_{(i,j)_k}}, X_{0_k}^{\mu_k}\right) \mathsf{e}(W'_{(i^*,j^*)}, X_2 X_1^{i^*+j^*} X_0^{i^*\cdot j^*}) \cdot$$

$$\mathsf{e}\left(h^\zeta b^\rho c^\omega v'^{-\tau}, X_{0_0}\right) = \mathsf{e}(v'^\gamma, X)\Bigg\}$$

where $\{W_i', W_{(i,j)}'\}$ and $W'_{(i^*,j^*)}$ are public inputs.

A.10 connected(i, j, ℓ)

The optimized connected statement for ℓ edges E' connecting i and j is as below:

$$PK\Bigg\{(\forall i \in V : \varepsilon_{i_0}, \varepsilon_{i_1}), (\forall (i,j) \in E \setminus E' : \varepsilon_{(i,j)_0}, \varepsilon_{(i,j)_1}), \{\varepsilon_{l,0}\}_{l=1}^{\ell}, \{\varepsilon_{l,1}\}_{l=1}^{\ell}, \varepsilon_{1,2},$$

$$\{\varepsilon_{l,3}\}_{l=1}^{\ell}, \zeta, \rho, \omega, \tau, \gamma, \mu_0, \mu_1) :$$

$$\prod_{k=0}^{1} \mathsf{e}\left(\prod_{i\in V} W_i'^{\varepsilon_{i_k}} \prod_{(i,j)\in E\setminus E'} W_{(i,j)}'^{\varepsilon_{(i,j)_k}}, X_{0_k}^{\mu_k}\right) \cdot$$

$$\mathsf{e}\left((W_1'^{i^*})^{\varepsilon_{1,0}} \prod_{l=2}^{\ell-1} W_l'^{\varepsilon_{l,0}} (W_\ell'^{j^*})^{\varepsilon_{\ell,0}}, X_{0_0}^{\mu_0}\right) \cdot$$

$$\mathsf{e}\left((W_1'^{i^*})^{\varepsilon_{1,2}} W_1'^{\varepsilon_{1,1}} \prod_{l=2}^{\ell-1} W_l'^{\varepsilon_{l-1,1}} W_l'^{\varepsilon_{l,1}} W_\ell'^{\varepsilon_{\ell-1,1}} (W_\ell'^{j^*})^{\varepsilon_{\ell,1}}, X_{0_1}^{\mu_1}\right) \cdot$$

$$\mathsf{e}\left(\prod_{l=1}^{\ell} W_l'^{\varepsilon_{l,3}}, X_{0_2}^{\mu_2}\right) \mathsf{e}\left(h^\zeta b^\rho c^\omega v'^{-\tau}, X_{0_0}\right) = \mathsf{e}(v'^\gamma, X)\Bigg\}$$

where $\{W'_i, W'_{(i,j)}\}$ and $W'_1, \ldots, W'_\ell$ are public inputs. The correctness can be verified from the following:

$$\mathsf{e}\left(W'_1, X_{0_2}^{\varepsilon_{1,3}}(X_{0_1}^{i^*})^{\varepsilon_{1,2}}X_{0_1}^{\varepsilon_{1,1}}(X_{0_0}^{i^*})^{\varepsilon_{1,0}}\right)\mathsf{e}\left(W'_2, X_{0_2}^{\varepsilon_{2,3}}X_{0_1}^{\varepsilon_{1,1}}X_{0_1}^{\varepsilon_{2,1}}X_{0_0}^{\varepsilon_{2,0}}\right)\cdots$$
$$\cdots\mathsf{e}\left(W'_{\ell-1}, X_{0_2}^{\varepsilon_{\ell-1,3}}X_{0_1}^{\varepsilon_{\ell-2,1}}X_{0_1}^{\varepsilon_{\ell-1,1}}X_{0_0}^{\varepsilon_{\ell-1,0}}\right)\mathsf{e}\left(W'_\ell, X_{0_2}^{\varepsilon_{\ell,3}}X_{0_1}^{\varepsilon_{\ell-1,1}}(X_{0_1}^{j^*})^{\varepsilon_{\ell,1}}(X_{0_0}^{j^*})^{\varepsilon_{\ell,0}}\right)$$
$$= \mathsf{e}\left(W_1'^{\varepsilon_{1,3}}, X_{0_2}^{u_2}\right)\mathsf{e}\left((W_1'^{i^*})^{\varepsilon_{1,2}}W_1'^{\varepsilon_{1,1}}, X_{0_1}^{u_1}\right)\mathsf{e}\left((W_1'^{i^*})^{\varepsilon_{1,0}}, X_{0_0}^{u_0}\right)\cdot$$
$$\mathsf{e}\left(W_2'^{\varepsilon_{2,3}}, X_{0_2}^{u_2}\right)\mathsf{e}\left(W_2'^{\varepsilon_{1,1}}W_2'^{\varepsilon_{2,1}}, X_{0_1}^{u_1}\right)\mathsf{e}\left(W_2'^{\varepsilon_{2,0}}, X_{0_0}^{u_0}\right)\cdots$$
$$\cdots\mathsf{e}\left(W_{\ell-1}'^{\varepsilon_{\ell-1,3}}, X_{0_2}^{u_2}\right)\mathsf{e}\left(W_{\ell-1}'^{\varepsilon_{\ell-2,1}}W_{\ell-1}'^{\varepsilon_{\ell-1,1}}, X_{0_1}^{u_1}\right)\mathsf{e}\left(W_{\ell-1}'^{\varepsilon_{\ell-1,0}}, X_{0_0}^{u_0}\right)\cdot$$
$$\mathsf{e}\left(W_\ell'^{\varepsilon_{\ell,3}}, X_{0_2}^{u_2}\right)\mathsf{e}\left(W_\ell'^{\varepsilon_{\ell-1,1}}(W_\ell'^{j^*})^{\varepsilon_{\ell,2}}, X_{0_1}^{u_1}\right)\mathsf{e}\left((W_\ell'^{j^*})^{\varepsilon_{\ell,0}}, X_{0_0}^{u_0}\right)$$
$$= \mathsf{e}\left((W_1'^{i^*})^{\varepsilon_{1,0}}\prod_{l=2}^{\ell-1}W_l'^{\varepsilon_{l,0}}(W_\ell'^{j^*})^{\varepsilon_{\ell,0}}, X_{0_0}^{u_0}\right)\cdot$$
$$\mathsf{e}\left((W_1'^{i^*})^{\varepsilon_{1,2}}W_1'^{\varepsilon_{1,1}}\prod_{l=2}^{\ell-1}W_l'^{\varepsilon_{l-1,1}}W_l'^{\varepsilon_{l,1}}W_\ell'^{\varepsilon_{\ell-1,1}}(W_\ell'^{j^*})^{\varepsilon_{\ell,1}}, X_{0_1}^{u_1}\right)\mathsf{e}\left(\prod_{l=1}^{\ell}W_l'^{\varepsilon_{l,3}}, X_{0_2}^{u_2}\right)$$

which is an extension of the original connected proof. This optimization reduces the number of pairings used to represent the connection from ℓ to only 3 pairings.

A.11 isolated(i, j)

Let $E' = ((i^*, j), (i, j^*))$, the optimized isolated proof is as below:

$$PK\Bigg\{((\forall i \in V : \varepsilon_{i_0}, \varepsilon_{i_1}), (\forall (i,j) \in E : \varepsilon_{(i,j)_0}, \varepsilon_{(i,j)_1}, \varepsilon_{(i,j)_2}), \zeta, \rho, \omega, \tau, \gamma,$$
$$\mu_0, \mu_1, \mu_2, \{\alpha_k\}_{k=0}^{\min(|E_{i^*}|,|E_{j^*}|)-1}) :$$
$$\prod_{i\in V}\mathsf{e}\left(W'_i, \prod_{k=0}^{1}X_{i_k}^{\varepsilon_{i_k}}\right)\prod_{k=0}^{2}\mathsf{e}\left(\prod_{(i,j)\in E}W_{(i,j)}'^{\varepsilon_{(i,j)_k}}, X_{0_k}^{\mu_k}\right)\cdot$$
$$\mathsf{e}\left(h^\zeta b^\rho c^\omega v'^{-\tau}, X_0\right) = \mathsf{e}(v'^\gamma, X)\wedge$$
$$\varepsilon_{(i,j)}[C_{(i,j)_{E_{l_{i^*}}}}]_{l_{i^*}=1}^{\ell_{i^*}} \in \Xi_{\mathcal{V}} \wedge i^* \in E_{i^*}\wedge$$
$$\varepsilon_{(i,j)}[C_{(i,j)_{E_{l_{j^*}}}}]_{l_{j^*}=1}^{\ell_{j^*}} \in \Xi_{\mathcal{V}} \wedge j^* \in E_{j^*}\wedge$$
$$\phi_{\mathsf{disjoint}(E_{j^*})}(E_{i^*})\Bigg\}$$

and the expanded form is as follows:

$$
\begin{aligned}
PK\Bigg\{ & ((\forall i \in V : \varepsilon_{i_0}, \varepsilon_{i_1}), (\forall (i,j) \in E : \varepsilon_{(i,j)_0}, \varepsilon_{(i,j)_1}, \varepsilon_{(i,j)_2}), \zeta, \rho, \omega, \tau, \gamma, \mu_0, \mu_1, \mu_2, \\
& \{\alpha_k\}_{k=0}^{\min(|E_{i^*}|,|E_{j^*}|)-1}) : \\
& \prod_{i \in V} \mathsf{e}\left(W'_i, \prod_{k=0}^{1} X_{i_k}^{\varepsilon_{i_k}}\right) \prod_{k=0}^{2} \mathsf{e}\left(\prod_{(i,j)\in E} W'^{\varepsilon_{(i,j)_k}}_{(i,j)}, X_{0_k}^{\mu_k}\right) \cdot \\
& \qquad \mathsf{e}\left(h^{\zeta} b^{\rho} c^{\omega} v'^{-\tau}, X_0\right) = \mathsf{e}(v'^{\gamma}, X) \wedge \\
& \mathsf{e}\left(\prod_{l_{i^*}=1}^{\ell_{i^*}} \left(\prod_{k=0}^{|\Xi_{\mathcal{V}}|} a_{0_k}^{\mathsf{m}_{\mathcal{V}_k}} \prod_{w_{l_{i^*}}=1}^{|E_{l_{i^*}}|-1} W_{\mathcal{E}_{w_{l_{i^*}}}}\right), X_{0_0}\right) = \\
& \qquad \prod_{l_{i^*}=1}^{\ell_{i^*}} \prod_{k=0}^{2} \mathsf{e}\left(\prod_{w_{l_{i^*}}=1}^{|E_{l_{i^*}}|} W^{\varepsilon_{(i,j)_k}}_{\mathcal{E}_{w_{l_{i^*}}-1}}, X_{0_k}^{\mu_k}\right) \mathsf{e}\left(a_{0_0}^{-1} W_{\Xi_{\mathcal{V}} \setminus E_{l_{i^*}}}, W_{\mathcal{E}_{|E_{l_{i^*}}|}}\right) \wedge \\
& \mathsf{e}\left(\prod_{l_{i^*}=1}^{\ell_{i^*}} W'_{\mathcal{E}_{l_{i^*}}}, X_{0_0}\right) = \prod_{l_{i^*}=1}^{\ell_{i^*}} \mathsf{e}\left(W'_{\mathcal{E}_{l_{i^*}-1}}, W_{\mathcal{E}_{|E_{l_{i^*}}|}}\right) \wedge \\
& \mathsf{e}\left(W'_{\mathcal{E}_{\ell_{i^*}}}, X_{0_0}\right) = \mathsf{e}\left(W_{i^*}, X_{0_1} X_{0_0}^{i^*}\right) \wedge \\
& \mathsf{e}\left(\prod_{l_{j^*}=1}^{\ell_{j^*}} \left(\prod_{k=0}^{|\Xi_{\mathcal{V}}|} a_{0_k}^{\mathsf{m}_{\mathcal{V}_k}} \prod_{w_{l_{j^*}}=1}^{|E_{l_{j^*}}|-1} W_{\mathcal{E}_{w_{l_{j^*}}}}\right), X_{0_0}\right) = \\
& \qquad \prod_{l_{j^*}=1}^{\ell_{j^*}} \prod_{k=0}^{2} \mathsf{e}\left(\prod_{w_{l_{j^*}}=1}^{|E_{l_{j^*}}|} W^{\varepsilon_{(i,j)_k}}_{\mathcal{E}_{w_{l_{j^*}}-1}}, X_{0_k}^{\mu_k}\right) \mathsf{e}\left(a_{0_0}^{-1} W_{\Xi_{\mathcal{V}} \setminus E_{l_{j^*}}}, W_{\mathcal{E}_{|E_{l_{j^*}}|}}\right) \wedge \\
& \mathsf{e}\left(\prod_{l_{j^*}=1}^{\ell_{j^*}} W'_{\mathcal{E}_{l_{j^*}}}, X_{0_0}\right) = \prod_{l_{j^*}=1}^{\ell_{j^*}} \mathsf{e}\left(W'_{\mathcal{E}_{l_{j^*}-1}}, W_{\mathcal{E}_{|E_{l_{j^*}}|}}\right) \wedge \\
& \mathsf{e}\left(W'_{\mathcal{E}_{\ell_{j^*}}}, X_{0_0}\right) = \mathsf{e}\left(W_{j^*}, X_{0_1} X_{0_0}^{j^*}\right) \wedge \\
& \mathsf{e}\left(W'_{\mathcal{E}_{\ell_{i^*}}}, X_{0_0}\right) = \mathsf{e}\left(W'_{\mathcal{E}_{\ell_{j^*}}}, \bar{W}_{j^*}\right) \mathsf{e}\left(\prod_{k=0}^{|E_{j^*}|-1} a_{0_0}^{\alpha_k}, X_{0_0}\right) \wedge \\
& 1_{\mathbb{G}} \neq \prod_{k=0}^{|E_{j^*}|-1} a_{0_0}^{\alpha_k} \Bigg\}.
\end{aligned}
$$

If $|E_{j^*}| > |E_{i^*}|$, the last three statements are changed to:

$$\mathsf{e}\left(W'_{\mathcal{E}_{\ell_{j^*}}}, X_{0_0}\right) = \mathsf{e}\left(W'_{\mathcal{E}_{\ell_{i^*}}}, \bar{W}_{i^*}\right) \mathsf{e}\left(\prod_{k=0}^{|E_{i^*}|-1} a_{0_0}^{\alpha_k}, X_{0_0}\right) \wedge$$

$$1_{\mathbb{G}} \neq \prod_{k=0}^{|E_{i^*}|-1} a_{0_0}^{\alpha_k}.$$