Efficient Linkable Ring Signature from Vector Commitment inexplicably named Multratug

Anton A. Sokolov

acmxddk@gmail.com

Abstract *In this paper we revise the idea of our previous work 'Lin2-Xor lemma and Log-size Linkable Threshold Ring Signature' and introduce another lemma, called Lin2-Choice, which extends the Lin2-Xor lemma. Using an novel zero-knowledge membership proof argument defined in the Lin2-Choice lemma, we create a compact general-purpose trusted-setup-free log-size linkable threshold ring signature called EFLRSL. The signature size is* $2\log_2(n+1) + 3l + 1$*, where n is the ring size and l is the threshold. By extending the membership argument of the Lin2-Choice lemma, we create a multifunctional version of the EFLRSL signature aliased as Multratug, of size* $2\log_2(n+l+1) + 7l + 4$*. In addition to signing a message, Multratug simultaneously proves balance and allows for easy multiparty signing. We use an arbitrary vector commitment argument in the role of the pivotal building block for both versions of our signature, considering it as a black box. Only the black-boxed pivot contributes components that depend on the ring size n into the signature sizes. This makes both of EFLRSL and Multratug combinable with other proofs, with overall size reduction. All this takes place in a prime-order group without bilinear parings under the decisional Diffie-Hellman assumption in the random oracle model. Both versions of our signature are proved unforgeable w.r.t. insider corruption and existentially unforgeable under chosen message attack. They remain anonymous even for non-uniformly distributed and malformed keys, which makes it possible to use them as a log-size drop-in replacement for LSAG-based schemes.*

Keywords: ring signature, membership proof, linkable, log-size, threshold, anonymity, blockchain, hidden amounts, balance proof, zero-knowledge, unforgeability, non-frameability, witness-extended emulation, LSAG

1 INTRODUCTION

In the paper [29] we created a log-size linkable threshold ring signature based on the Lin2-Xor lemma, which we proved there under the decisional Diffie-Hellman (DDH) assumption. Now, we have the following two questions. Can we generalize the Lin2-Xor lemma using an arbitrary vector commitment argument that has computational witness-extended emulation (cWEE) and is special honest verifier zero-knowledge (sHVZK)? And also can we get a pairings-free trusted-setup-free linkable threshold ring signature out of it that is more efficient in size and verification complexity, while remaining under DDH in a prime-order group?

We answer both of these questions in the affirmative. Lin2-Choice lemma and its accompanying efficient ring signature we present herein seem to be useful findings. Our new ring signature keeps using the linking tag of the form $x^{-1}\mathcal{H}_{\mathbf{point}}(xG)$ and, in addition to this, has a version with the linking tag form $x\mathcal{H}_{\mathbf{point}}(xG)$ which is time-tested since the work by Liu, Wei, and Wong [23]. Although, both of these linking tags are indistinguishable from each other and from uniform randomness [29, 13].

By a vector commitment or, equivalently, by a commitment to a vector we mean a weighted sum of orthogonal generators that binds the corresponding scalar weight vector. By a vector commitment argument we mean a proof of knowledge of such a bound weight vector. Most of the arguments in this paper rely on the vector commitment argument, hence we call it a pivotal unit.

The signature we present, called EFLRSL, is a linkable threshold ring signature which may serve as a drop-in replacement for the LSAG signature [23]. EFLRSL is derived from the novel proof of membership protocol that we introduce in the Lin2-Choice lemma. The idea behind this proof of membership is about counterweighting a number of independent randomnesses using a single prover-controlled weight in an expression. The Lin2-Choice lemma establishes the necessary cryptographic properties of the protocol.

Moving forward, in the subsequent Lin2-2Choice lemma we add an extra element to our membership proof and create a signature called Multratug, which in addition to proving knowledge of signing keys also proves the sum of hidden amounts. Thus, our resulting signature is Multratug, which is an extended version of EFLRSL.

By the proof of sum of hidden amounts, proof of balance for short, we mean that prover demonstrates a blinded commitment to some secret amount and proves that this secret amount is equal to the sum of those amounts which correspond to the actual signing keys and are also blinded.

We will not repeat common words about signatures from the introduction of [29], they all remain valid. We will keep our presentation brief, considering that many detailed explanations can be taken from [29] as well as from the work of Benedikt Bünz et al. [7]. As another basic ingredient, we will now use what we think is an elegant way of turning a protocol into zero-knowledge by adding noise in a separate orthogonal dimension, which we found in the works of Attema and Cramer [2] and Heewon Chung et al. [9].

Overall, in this paper we assume that a reader has an understanding of the works [7, 9, 29] and possesses an appropriate intuition, so we keep our descriptions and proofs concise, otherwise the paper would be too long. Moreover, since the methods of proving sHVZK and cWEE properties of protocols are already widely known, e.g., from [7, 9, 2], and the same for unforgeability, anonymity, and other properties of signatures, known, e.g., from [23, 15, 13, 25], we describe only the key points for our proofs, believing that they suffice to reconstruct all the details of interest.

1.1 MOTIVATION

Besides the two questions we have already outlined at the beginning, our motive in creating this paper is that we observe no one among the most prominent log-size ring signatures available nowadays that is as universally applicable as the linear-size schemes originating from AOS [1] and LSAG [23]. Of course, we are considering only the portion of the large number of existing signatures that does not require trusted setups or curve pairings, and is under the types of Diffie-Hellman assumption.

By the universal applicability of a signature scheme we mean the possibility of using it, maybe with some additive modifications, for solving the following list of problems:

- ◇ regular anonymous 1-out-of-many signing,
- ◇ signing only once (linkable ring signature),
- ◇ simultaneous proof of balance (support for hidden amounts),
- ◇ l-out-of-n signing (threshold case, we use the word ‘threshold’ in this sense hereinafter and assume $l \ll n$ for performance comparison; signature size is expected to be less than simply $l\times$1-out-of-many case size),
- ◇ the case when public keys are formed according to the CryptoNote [31] protocol rules (which are adopted in many blockchains these days),
- ◇ and also the most general case when public keys are not restricted by anything (e.g., they can be generated ad hoc and be completely malformed, nevertheless LSAG remains secure and anonymous with them),
- ◇ in addition, especially in the context of blockchains, it is often desirable that a signature allows for easy implementation of multiparty signing operations (multisignature operations, described, e.g., in [16]).

Having conducted a kind of pragmatic research, we found that the recently proposed linear-size CLSAG scheme [13], which generalizes and optimizes LSAG, solves all the listed problems except for the threshold case. We took CLSAG for reference and compared the applicability of the currently known top-performance log-size schemes with it, our results are shown in Table 1.

Table 1: Applicability of signature schemes

	Log-sz	Regular	Linkable	Balance	Thresh.*	Blockchain	General	MP**
CLSAG [13]		✓	✓	✓		✓	✓	✓
Lelantus Spark [16]	✓	✓	✓	✓		✓		✓
Triptych [25]	✓	✓	✓	✓		✓		
RingCT3.0 [32]	✓	✓	✓	✓	✓	✓		
Omniring [21]	✓	✓	✓	✓	✓	✓		✓
DualRing-EC [33]	✓	✓						

* Many-out-of-many size with threshold=l is asymptotically, for big n and l, lower than 1-out-of-many size times l.
** Multiparty signing is easy to implement.

All the examined schemes have logarithmic size, except for the referenced CLSAG, and all of them provide the functionality of a regular ring signature. They are roughly ordered by size in the table. Some of the schemes have versions which implement different subsets of the corresponding check-marked properties, we provide a more elaborated comparison for them in Tables 8, 9, 10.

DualRing-EC [33], which does not have any linkable version by-design, is the most size and verification time efficient among the found signatures. However, its security model requires only properly generated keys, we show a forgery for the contrary case in Appendix X. The other examined log-size signatures are linkable by-design. All of them include balance proofs and are compatible with CryptoNote public keys, aka stealth addresses [31], of the form $B + \mathcal{H}_{\mathbf{scalar}}(rA)G$.

Only the RingCT3.0 [32] and Omniring [21] schemes substantially save space when many signers sign simultaneously. Triptych [25], RingCT3.0, and Omniring have linking tags of the form U/x, where U is a predefined generator, which deanonymizes them in the general case, as we show in Appendix Y. The fact of having private key x in the tag's denominator also makes it hard to implement multisignature operations. Lelantus Spark [16] has its own subsystem that solves this problem, however, the entire scheme seems too narrowly tied to the decentralized payments to be considered general. We shall note that we compare to the general case for our pure interest, whereas originally most of the top-performing schemes are claimed in their papers as blockchain-oriented only.

Omniring has a version with linking tag form $x\mathcal{H}_{\mathbf{point}}(xG)$, the same form is used in CLSAG. This tag is invulnerable to malformed keys and is multisignature-friendly, however the original Omniring paper [21] provides security model only for the less secure U/x tag. So, we have to assume that both versions of the scheme are bound to the CryptoNote stealth addresses regardless of the tag used. As confirmed to us by the Omniring authors, there is no claim that the scheme will remain anonymous when used with malformed keys in the scenario described in Appendix Y, in which LSAG and CLSAG still remain to be.

Therefore, our second motivation in creating a new general-purpose signature is to make an attempt to implement all the properties specified in Table 1 in a single scheme of a relatively good size. As a result, in this paper we present the EFLRSL/Multratug scheme with the properties shown in Table 2, which can be inserted close to the bottom of the table Table 1.

Table 2: Applicability of our scheme

	Log-sz	Regular	Linkable	Balance	Thresh.	Blockchain	General	MP
EFLRSL / Multratug	✓	✓	✓	✓	✓	✓	✓	✓

With all this, our main objective remains to determine what can be obtained from the Lin2-Choice and Lin2-2Choice lemmas presented herein, and how practical that would be. In the most elementary cryptographic group and with minimal additional means, i.e., using a compact vector commitment argument without even involving the inner product argument.

1.2 COMMITMENT TO VECTOR, VECTOR COMMITMENT, AND ITS ARGUMENT

Our pivotal protocol, which all our proofs of membership and signatures ultimately refer to by calling it only once in the last step, is a vector commitment argument. Throughout this paper, by the vector commitment or by the commitment to a vector, we use these terms interchangeably, we mean a published element P such that $P = \langle \mathbf{a}, \mathbf{P} \rangle$, where $\mathbf{P}$ is a vector of orthogonal generators, and $\mathbf{a}$ is a vector of scalar weights, typically large. Vector commitment argument, respectively, is an argument that proves knowledge of all the weights in $\mathbf{a}$ at once. This is similar to the Sum Argument defined in [33], however our implementation is a bit different.

The term vector commitment is already used in the literature for another construction which is described, e.g., in [8, 22, 14], and relates to groups with bilinear pairings. On the contrary, we denote by this term the construction in a pairings-free group that can be thought of as an extremely simplified form of the construction from [8]. In favor of our terminology is, e.g., the construction called vector commitment in [4], which is similar to ours.

A blinded version of the vector commitment of the form $P = \langle \mathbf{a}, \mathbf{P} \rangle + \alpha H$, where H is orthogonal to $\mathbf{P}$, and α is independently uniformly sampled, is commonly called as Pedersen vector commitment. It is defined in [7] as an extension to Pedersen commitment [26]. In our terminology, Pedersen vector commitment is a subset case of the vector commitment. Both of the vector commitment and Pedersen vector commitment are binding, however only the latter is necessarily hiding, and the former becomes hiding only when blinded.

1.3 RELATED WORK

A vector commitment argument closely resembling our pivot, in terms of its role in the larger scheme and its construction, is the compressed pivotal argument by Attema and Cramer in [2]. Although our signature is agnostic to the pivot implementations, we consider some of them when calculating size. The most efficient one we consider can be thought of as a subset case of the compressed pivot from [2] with the empty set of connected linear forms $L \equiv \emptyset$, using definition of L from [2]. Further in their work, Attema and Cramer obtain results for $L \neq \emptyset$. Meanwhile, we investigate the other direction from the point $L \equiv \emptyset$ by studying what happens if the base set of orthogonal generators $\mathbf{P}$ varies with challanges.

For a prime-order group without bilinear pairings, historically there are two main approaches to constructing trusted-setup-free log-size membership proofs and signatures in it. The first of them derives from the identification scheme and its variations by Groth and Kohlweiss [15], and the second one comes from the inner product argument and subsequent proof for an arbitrary arithmetic circuit by Bünz et al. [7]. We do not use either one.

The recently proposed efficient signature schemes are listed in Table 1. Triptych [25] and Lelantus Spark [16] rely on the idea of Groth and Kohlweiss [15] by building on top of it. RingCT3.0 [32] and Omniring [21] heavily employ the inner product argument by Bünz et al. [7]. At the same time, there exist a number of other discrete-log, prime-order, pairings-free, trusted-setup-free, log-size schemes and approaches, we do not mention them because of their lower efficiency compared to the top-performers [32, 21, 25, 16].

The DualRing-EC signature by Tsz Hon Yuen et al. [33] has a restrictive security model, nevertheless it advances an elegant idea of better compression. Although we do not use this idea directly, it inspired us to look for an optimized version of the vector commitment argument, which ended up being almost the same as the compressed pivot in [2] that has a strong security model.

An informal introduction to the theme of commitments and log-size arguments in a prime-order group, as well as a detailed explanation of the work [7] including an overview of the corresponding optimization techniques such as multi-exponentiation and batch verification, can be found in the article by Adam Gibson [12].

In the previous paper [29], we represent an approach based on our own identification scheme, thus providing the early results of what can be obtained by building decoy sets of element pairs and 'rotating' them with challenges. However, the signature constructed in [29] is somewhat large in size. In the current paper, we will reinvent the idea of [29] immediately targeting many-out-of-many proofs and will obtain the much more efficient schemes. In this paper we will still use the definitions of signature properties, such as unforgeability, anonymity, non-frameability, collected in [29].

Recent work by Russell W. F. Lai et al. [20] introduces a method of building succinct arguments for bilinear group arithmetic. The method relies on an enhanced commitment, which in addition to a scalar vector can contain group elements as witnesses to a system of generalized bilinear relations which is further compressed. The method is presented in a group with pairings and can be applied equally well in non-pairing groups, as shown in [19]. Possibility of constructing a variety of signatures using the bilinear group arithmetic also follows from [20].

The subsequent work by Thomas Attema et al. [3] takes a more efficient approach to constructing the bilinear group arithmetic relations while retaining the same type of the enhanced commitment. An efficient transparent setup threshold signature scheme (TSS) is built in [3], giving an idea of its applicability and size. Compared to our current work, first of all, in the TSS terminology 'threshold' means that k signatures can be dynamically merged after creation, which is stronger than our 'threshold' that merely requires to know l signing keys when creating a signature. Second, merged TSS size is independent of k, whereas all versions of our signature have linear by l sizes. Third, for large ring size n the asymptote of TSS is at least $4\lceil\log_2(n)\rceil$, while the asymptote of our signature is $2\lceil\log_2(n)\rceil$. Thus, TSS is more space efficient for big thresholds. Our region of interest, however, is low thresholds with large rings, and our signature is more efficient within it.

1.4 CONTRIBUTION

In this paper, we propose several novel efficient trusted-setup-free pairings-free DDH-based log-size schemes listed in the following subsections, ranging from the Lin2-Choice lemma membership proof protocol to concise general-purpose EFLRSL and blockchain-oriented balance-proof Multratug versions of our signature.

Our schemes are based on a black-boxed arbitrary vector commitment argument called pivot. We show a plain implementation of it, which is a subset case of the inner product argument from [7] with nullified inner product. Also, we show an efficient implementation of it, which is a subset case of the compressed pivot from [2] with the empty set of connected linear forms.

Overall, our EFLRSL and Multratug signatures have such a design that redirects everything associated with ring size to the pivot. They require neither the full inner product argument from [7], nor a bilinear group arithmetic as in [20, 3], and have the different from [15] underlying proving system which we develop here.

The Lin2-Choice lemma is the main lemma of this paper. Its membership proof idea is the keystone of the underlying proving system for our signatures. Since this idea seems to us a rather generic approach, we also point out its other applications. Namely, in addition to our DDH-based log-size membership proofs and signatures, as an extension, in the appendix we sketch out a Q-DLOG-based [17] constant-size membership proof using it.

1.4.1 LIN2-CHOICE LEMMA'S MEMBERSHIP PROOF

The Lin2-Choice lemma is a generalization of the Lin2-Xor lemma [29] to the case of n pairs of elements. In a nutshell, both of these lemmas provide protocols that prove membership, however the first one proves membership in a set of 2 elements whereas the second does the same for n elements. In addition, compared to the Lin2-Xor lemma protocol, the Lin2-Choice lemma's one is substantially refined to separate the ring elements from the auxiliary elements involved in the pairs.

The outcome of this protocol is that having a ring $\mathbf{P} = \{P_i\}_{i=0}^{n-1}$ of n orthogonal elements and a Pedersen commitment Z to an arbitrary element $P_s \in \mathbf{P}$, prover convinces verifier of membership Z in $\mathbf{P}$. This takes only 1 group elements and 1 scalar, to which the size of an externally employed vector commitment argument, i.e., of the pivot, is added. Thus, the Lin2-Choice lemma provides a concise 1-out-of-many membership proof. It has an uncomplicated design and easily extends into a many-out-of-many membership proof. Also, the external vector commitment argument can be shared with other protocols to save space.

In the Lin2-Choice itself, we formally prove in detail that this membership proof has cWEE and, also, we informally show that it is trivially sHVZK by referring to the same blinding design cases formally proved in [2, 9].

1.4.2 EFLRSL SIGNATURE

EFLRSL is a regular linkable threshold ring signature immediately derived from the many-out-of-many version of the Lin2-Choice lemma's proof of membership, with size

$$2\lceil \log_2(n+1) \rceil + 3l + 1.$$

It is a simplified version of our larger Multratug signature without any balance proof or multiparty signing, and with linking tag (aka key image) in the form $x^{-1}\boldsymbol{\mathcal{H}}_{\mathbf{point}}(xG)$.

EFLRSL is general-purpose, i.e., it suits for environments where keys can be generated by signers ad hoc and be arbitrarily malformed. For example, EFLRSL is appropriate for implementing whistleblowing or e-voting systems, for which LSAG [23] used to be chosen. Compared to the streamlined versions of the recent top-performance schemes listed in Table 1, EFLRSL appears to be by far the best sized general-purpose linkable ring signature, the respective comparison is shown in Table 10.

Since EFLRSL is based on the proof of membership which, according to the Lin2-Choice lemma, is sHVZK and has cWEE, the signature is unforgeable and anonymous. To prove this, we use and refer to the techniques from [23, 25, 13, 15, 29], where the situation is the same and the appropriate proofs are provided in full detail.

1.4.3 LIN2-2CHOICE LEMMA'S MEMBERSHIP PROOF WITH ADDITIONAL ELEMENT

The Lin2-2Choice lemma is an evolved version of the Lin2-Choice lemma; its protocol comprises l instances of the Lin2-Choice lemma 1-out-of-many membership proof, each of them extended in such a way as to select a linear combination of exactly two elements of the ring instead of one. All optimized together in a single extended many-out-of-many proof.

It can be introduced by the following example. For the ring $\mathbf{P} \cup \mathbf{V} = \{P_i\}_{i=0}^{n-1} \cup \{V_k\}_{k=0}^{l-1}$ of $(n+l)$ elements and a set of l Pedersen commitments $\mathbf{Z} = \{Z_k\}_{k=0}^{l-1}$, using the Lin2-2Choice lemma protocol, prover convinces verifier that, for each $Z_k \in \mathbf{Z}$, it holds $Z_k = p_k P_{s_k} + v_k V_k$ for some p_k, v_k, s_k known to the prover. This takes only $2l$ group elements and l scalars, plus the size of an external vector commitment argument.

We prove in detail that this extended membership proof has cWEE, and also we informally show it is sHVZK by referring to the same uncomplicated blinding design in [2, 9].

We use this protocol to prove balances in our Multratug signature. Roughly speaking, n signature ring addresses and their associated amounts go to the set $\mathbf{P}$, whereas l actually spent amounts with re-randomized blinding factors go to the set $\mathbf{V}$ in it. Each commitment $Z_k \in \mathbf{Z}$ comprises G and a key image. With additional means we convince verifier that $\forall k : p_k = -v_k$. Thus, having played this protocol, the verifier is convinced that each actually spent amount in $\mathbf{V}$ is fully compensated by some amount in $\mathbf{P}$ and, also, that those unknown addresses in $\mathbf{P}$ which performed the compensation have the known to the prover p_k's such that $G = p_k P_{s_k}$. The latter convinces the verifier that the prover knows the actual signing private keys, and that the key images are built correctly.

Moreover, by properly filling in the input set $\mathbf{P} \cup \mathbf{V}$ for the Lin2-2Choice lemma protocol, we are able to substitute linking tag $x\boldsymbol{\mathcal{H}}_{\mathbf{point}}(xG)$ for $x^{-1}\boldsymbol{\mathcal{H}}_{\mathbf{point}}(xG)$ in the Multratug signature.

1.4.4 HELPER ARGUMENT: RANDOM WEIGHTING FOR T-S TUPLES

Suppose we have two tuples of elements, possibly blinded. Taking their inner products with a random scalar vector, we wonder: if these inner products are shown to be proportional to each other, does this prove that the tuples are elementwise and with the same factor proportional to each other? This question emerged in one of our proofs. We have looked in the existing literature and found no answer.

Therefore, we compiled an appropriate argument, defined sufficient conditions, and presented the answer in this paper. It is that, in brief, for any $\mathbf{T} = \{T_i\}_{i=0}^{n-1}$ and $\mathbf{D} = \{D_i\}_{i=0}^{n-1}$, if for random $\boldsymbol{\xi} = \{\xi_i\}_{i=0}^{n-1}$ prover provides a valid proof of knowledge of a such that $\langle \boldsymbol{\xi}, \mathbf{D} \rangle = a \langle \boldsymbol{\xi}, \mathbf{T} \rangle$, and also if $\mathbf{T}$ contains at least two orthogonal to each other elements, then verifier is convinced that $\mathbf{D} = a\mathbf{T}$ holds.

1.4.5 MULTRATUG SIGNATURE WITH BALANCE PROOF

Multratug is an universally applicable ring signature derived from the Lin2-2Choice lemma protocol. It simultaneously proves knowledge of signing keys and balance. Multratug has linking tag $x\mathcal{H}_{\mathbf{point}}(xG)$ and, also, has all of the properties check-marked in Table 2, its size is

$$2\lceil\log_2(n+l+1)\rceil + 7l + 4.$$

We provide a detailed formal proof of correctness of its balance. We provide only sketches of proofs for its unforgeability and anonymity, since being based on the sHVZK and cWEE properties of the underlying proving system these proofs entirely follow known techniques.

Multratug expands the scope of EFLRSL by adding support for hidden amounts and multisignature operations. It can be used in blockchains, however, is not limited by that. Since the multisignature operations is typically a must-have feature for modern blockchains, it makes sense to compare Multratug only with those signatures that allow them (column 'MP' in Table 1). The full comparison results are shown in Tables 8, 9.

1.5 PREVIEW OF THE CORE PROTOCOLS

1.5.1 LIN2-CHOICE LEMMA'S MEMBERSHIP PROOF

For the orthogonal ring $\mathbf{P} = \{P_i\}_{i=0}^{n-1}$ and Pedersen commitment Z, the Lin2-Choice lemma protocol proves membership of Z in $\mathbf{P}$. It looks as the following game, although we simplify it for this preview.

At the start both of the prover and verifier have Z and $\mathbf{P}$. They jointly pick n helper generators $\mathbf{Q} = \{Q_i\}_{i=0}^{n-1}$ such that all elements of $\mathbf{P} \cup \mathbf{Q}$ are orthogonal to each other. The prover publishes an element F. Then the verifier releases challenges $\mathbf{c} = \{c_i\}_{i=0}^{n-1}$, and the prover replies with a scalar r. Next, the verifier releases a challenge δ. Given all this, finally the prover convinces the verifier using an arbitrary external vector commitment argument that the element $\hat{Z}$ defined as

$$\hat{Z} = Z + \delta r F$$

is a weighted sum, with weights known to the prover, of the elements from the set

$$\{P_i + \delta c_i Q_i\}_{i=0}^{n-1}.$$

The involved external vector commitment argument must be sHVZK and has to have cWEE. Also, the commitment Z and all elements published by prover are properly blinded, we omit showing the blinding components in this preview.

In the Lin2-Choice lemma we prove that the above game succeeds only if either there exists nonzero scalar p known to the prover such that $p^{-1}Z \in \mathbf{P}$, or if it holds that $Z = 0$. As well as the game is sHVZK and has cWEE.

1.5.2 LIN2-2CHOICE LEMMA'S MEMBERSHIP PROOF

Compared to the Lin2-Choice lemma's simplified game in Section 1.5.1, one for the Lin2-2Choice lemma looks as follows. The former ring $\mathbf{P}$ expands to $(n + l)$ entries by the second part $\mathbf{V} = \{V_k\}_{k=0}^{l-1}$ together with the jointly picked helper generators $\mathbf{W} = \{W_k\}_{k=0}^{l-1}$.

So, now at the start both of the prover and verifier have the ring $\mathbf{P} \cup \mathbf{V}$, the set of commitments $\mathbf{Z} = \{Z_k\}_{k=0}^{l-1}$, and the set of helper generators $\mathbf{Q} \cup \mathbf{W}$ such that all elements of $\mathbf{P} \cup \mathbf{V} \cup \mathbf{Q} \cup \mathbf{W}$ are orthogonal to each other. The prover publishes l element pairs (F_k, E_k), $k \in [0 \dots l-1]$, the verifier releases random $\mathbf{c} = \{c_i\}_{i=0}^{n+l-1}$, the prover replies with l scalars r_k, $k \in [0 \dots l-1]$, the verifier releases random δ_1, δ_2. The prover convinces the verifier that, for each $k \in [0 \dots l-1]$, the element $\hat{Z}_k$ built as

$$\hat{Z}_k = Z_k + \delta_1 r_k F_k + \delta_2 c_{n+k} E_k$$

is a weighted sum, with weights known to the prover, of elements from the set

$$\{P_i + \delta_1 c_i Q_i\}_{i=0}^{n-1} \cup \{V_{i-n} + \delta_2 c_i W_{i-n}\}_{i=n}^{n+l-1}\,. \tag{1}$$

Moreover, the proover convinces the verifier that the above holds for all $\hat{Z}_k$'s in one step, by proving that the sum

$$\sum_{k=0}^{l-1} \lambda_k \hat{Z}_k\,,$$

with independently and uniformly sampled coefficients λ_k's, is the weighted sum of elements from the set (1).

The Lin2-2Choice lemma guarantees this game is sHVZK, has cWEE, and completes successfully only if prover knows indices $\mathbf{s} = \{s_k\}_{k=0}^{l-1}$ and scalar factors $\mathbf{p} = \{p_k\}_{k=0}^{l-1}$, $\mathbf{v} = \{v_k\}_{k=0}^{l-1}$ such that, for each $Z_k \in \mathbf{Z}$, it holds

$$Z_k = p_k P_{s_k} + v_k V_k.$$

1.5.3 PIVOT: OPTIMIZED VECTOR COMMITMENT ARGUMENT

Our membership proofs invoke an arbitrary vector commitment argument directly or indirectly at the last steps of their protocols, and we unify the entire variety of such an arbitrary vector commitment argument under the name of a pivotal black box.

As our signatures are built on top of the corresponding membership proofs, to be able to prove their unforgeability we require this black-boxed pivot to be complete, sHVZK, and to have cWEE. We put a preview of one of its possible implementations here, although any other implementation that proves the same having the same properties will do. Note, our pivot is conceptually similar to and can be understood as the compressed pivot with $L \equiv \varnothing$ in [2].

The idea is that initially we build a complete, sHVZK, and having cWEE linear-size Schnorr-like vector commitment argument that convinces verifier that given element Y is a weighted sum, with weights known to the prover, of elements from the vector $\mathbf{X} = \{X_i\}_{i=0}^{n-1}$ such that all X_i's $\in \mathbf{X}$ are orthogonal to each other. It looks as follows. The prover publishes an element T as the first message, the verifier issues a challenge c, the prover replies with a scalar vector $\boldsymbol{\tau}$, the verifier checks that $\langle \boldsymbol{\tau}, \mathbf{X} \rangle + cY = T$. This game comprises n played in parallel Schnorr identification protocol games [27], for each $X_i \in \mathbf{X}$. The fact that Y and T are necessarily weighted direct sums of $\mathbf{X}$ implies all n parallel games are independent of each other, otherwise the orthogonality of $\mathbf{X}$ can be shown to be broken.

Next, for $n > 4$ in this game, instead of replying with $\boldsymbol{\tau}$ the prover replies with a proof of knowledge of $\boldsymbol{\tau}$, which takes only $2\lceil \log_2(n) \rceil$ elements if the reduction from [7] is used. This proof does not need to be sHVZK, as $\boldsymbol{\tau}$ itself already reveals nothing. Thus, we obtain a complete, sHVZK, and cWEE optimized vector commitment argument of size $2\lceil \log_2(n) \rceil + 1$.

When Y is blinded, the blinding generator denoted as H is orthogonal to $\mathbf{X}$, we usually precompute it as a hash to curve $\mathcal{H}_{\mathbf{point}}$ of everything publicly visible at the moment. In this case, we implicitly append H to $\mathbf{X}$ in the above game. This way, the size of the pivotal argument gets increased by 1 under the logarithm and becomes $2\lceil \log_2(n+1) \rceil + 1$.

1.5.4 LINKABLE THRESHOLD RING SIGNATURE EFLRSL

Having a ring of public keys (addresses) $\mathbf{P} = \{P_i\}_{i=0}^{n-1}$, for the first, we orthogonalize it. Namely, we build from it the orthogonal decoy set $(\mathbf{P} + \zeta\mathbf{U})$, where $\mathbf{U} = \{\mathcal{H}_{\mathbf{point}}(P_i)\}_{i=0}^{n-1}$ and ζ is random.

We construct the simplest linkable ring signature EFLRS1, which is for one actual signer, by defining the key image as $I = x^{-1}\mathcal{H}_{\mathbf{point}}(xG)$, where $P_s = xG$ for some index $s \in [0 \dots n-1]$, and by applying the Lin2-Choice lemma's membership proof to the commitment $Z = G + \zeta I$ in the above decoy set.

For l parallel instances of EFLRS1 over the same ring $\mathbf{P}$, we have l instances of the Lin2-Choice lemma's membership proof in them. Using random weights, we merge these membership proofs into one. Thus, we obtain the linkable threshold ring signature EFLRSL, which is for l actual signers and which makes only one call to the Lin2-Choice lemma's membership proof.

1.5.5 MULTRATUG SIGNATURE WITH BALANCE PROOF

Suppose that the ring $\mathbf{P}$ of public keys (addresses) is complemented by the set of hidden (blinded) amounts $\mathbf{A} = \{A_i\}_{i=0}^{n-1}$ such that, for each index i, the hidden amount $A_i \in \mathbf{A}$ is related to the address $P_i \in \mathbf{P}$. Also, suppose, a total hidden amount $A^{\mathbf{sum}}$ is given, and the balance with it should be proved.

We might subtract $A^{\mathbf{sum}}$ from each A_i and prove that for actual signer this difference contains only the blinding component, as it is done, e.g., in [25]. However, this would prevent us from creating an efficient threshold version of the signature. Therefore, we specify the set $\mathbf{A^{tmp}} = \{A_k^{\mathbf{tmp}}\}_{k=0}^{l-1}$ of re-hidden (with re-randomized blinding factor) amounts corresponding to the actual signing indices and, simply put, add them to the end of the ring.

Since we already have in our disposal the Lin2-2Choice lemma's extended membership proof, we adjust it a bit for our needs by making $\mathbf{p} = \mathbf{v}$. This is achieved by adding a new orthogonal generator $K = \mathcal{H}_{\mathbf{point}}(\mathbf{Z}, \mathbf{P}, \mathbf{V}, \dots)$ to each element in $\mathbf{P}$, and subtracting K from each element in $\mathbf{V}$. Further we do not mention K, and consider that our extended membership proof convinces verifier, for all $Z_k \in \mathbf{Z}$, that

$$Z_k = p_k(P_{s_k} + V_k), \quad \text{where } s_k, p_k \text{ are known to prover.}$$

So, for Multratug, the simplified game is that at the start both of the prover and verifier have $\mathbf{P}, \mathbf{A}, \mathbf{A^{tmp}}$, and both of them also have the helper generators $\mathbf{Q}, \mathbf{W}$ required by the Lin2-2Choice lemma protocol.

It is impossible to ensure the orthogonality of regular addresses and hidden amounts taken from a blockchain, however the necessary orthogonality can be easily established by adding the corresponding hashes-to-group to them, e.g., as it is done using the hashes $\mathbf{U}$ in Section 1.5.4, we omit showing them in this preview.

After making the appropriate orthogonalization, for a randomly sampled ω, the prover and verifier have all elements in $(\mathbf{P} - \omega\mathbf{A}) \cup \omega\mathbf{A^{tmp}} \cup \mathbf{Q} \cup \mathbf{W}$ orthogonal to each other. Letting, for each $k \in [0 \dots l-1]$, the commitment

Z_k be equal to G and using the Lin2-2Choice lemma membership proof, the prover convinces the verifier that it knows s_k, p_k such that

$$G = p_k((P_{s_k} - \omega A_{s_k}) + \omega A_k^{\mathbf{tmp}}) \,. \tag{2}$$

This equality splits into $G = p_{s_k} P_{s_k}$ and $A_{s_k} = A_k^{\mathbf{tmp}}$. Of course, we have omitted blinding components here. Also, we assume that all elements in $\mathbf{P}$ are reliably distinct and nonzero.

Thus, for all k's, the equalities (2) prove knowledge of signing private keys at indices s_k's, and also they prove that each $A_k^{\mathbf{tmp}}$ is equal to A_{s_k} to the accuracy of blinding component. After that, to be convinced that the balance is met, it only remains to check that $\sum_{k=0}^{l-1} A_k^{\mathbf{tmp}} = A^{\mathbf{sum}}$ holds to the accuracy of blinding component, that's all.

In addition, the Multratug signature substitutes the $x\boldsymbol{\mathcal{H}}_{\mathbf{point}}(xG)$ key image for the inherited from the EFLRSL signature $x^{-1}\boldsymbol{\mathcal{H}}_{\mathbf{point}}(xG)$ one. For this, the same techinque as for proving the equalities of hidden amounts to their re-hidden counterparts in $\mathbf{A}^{\mathbf{tmp}}$ is used. In Section 9.1.2 we explain this in detail.

2 PRELIMINARIES

We first outline the definitions, assumptions, and methods that we borrow from the base works. Also, we specify the notation and base environment that we use in this paper. Since we construct our signatures from many lesser protocols, we combine the latter under the name of underlying proving system.

2.1 DEFINITIONS AND BASE WORKS

2.1.1 CONTEXT

All of our protocols, including the helpers schemes and signatures, perform for a prime-order group without bilinear pairings in a trustless environment under the decisional Diffie–Hellman (DDH) assumption in the random oracle model, as in [7]. All of our protocols are written as interactive, however, we always imply the existence of their non-interactive Fiat-Shamir counterparts not mentioning them.

All the context, namely, the common reference string, trustless setup, discrete logarithm (DL) relation and DDH assumptions, orthogonality, commitment binding and hiding, non-interactivity through Fiat-Shamir heuristic, perfect completeness (we call it simply completeness), argument of knowledge, special honest verifier zero-knowledge (sHVZK) and computational witness-extended emulation (cWEE) definitions and proof methods which we use are exactly the same as in [7, 9]. Taking them as already well known, we do not quote or explain them in detail to save space, instead referring simply to the fact that they correspond to and can be copied from [7].

2.1.2 COMMON WITH OUR PREVIOUS WORK

As a syntactic sugar we use the shorthands '$\sim$', '$!\sim$', 'lin', 'ort' defined in [29], although they can be resolved and removed. We use additive notation for exponentiation of group elements, as, e.g., in [25, 29]. We refer to [29] for proving some few auxiliary claims, for example, to prove that the linking tags in the forms $x\boldsymbol{\mathcal{H}}_{\mathbf{point}}(xG)$ and $x^{-1}\boldsymbol{\mathcal{H}}_{\mathbf{point}}(xG)$ are statistically indistinguishable from each other.

In [29] we have collected the existing definitions of linkable ring signature and its security models, from various sources. We use these definitions hereinafter with the only one difference in that, what in [29] is called generic linkable ring signature now we simply call linkable ring signature.

2.1.3 RELATIONS AND UNIQUENESS OF WITNESS

We prove soundness of our protocols using the same method as in [7]. Namely, for each of our protocols, we prove that it has cWEE for the corresponding polynomial-time-decidable relation denoted as $\mathcal{R}$. We say that a relation is fulfilled (or proved) by a protocol as a synonym for that the protocol has cWEE for the relation and, consequently, witness of the relation is extractable in a polynomial time.

It should be observed that while the cWEE property implies prover's knowledge of $\mathcal{R}$'s witness, it does not guarantee that the witness is unique. We use the term uniqueness in the same sense as in [7, 29]. For each of our protocols in this paper, corresponding witness is always fed in prover's private input.

In most cases, as in [7], uniqueness of witness follows from the fact that it represents an opening of some binding commitment included in the statement in $\mathcal{R}$. When this is not the case or is not obvious, we prove uniqueness of witness by showing that knowing two different values of it causes breaking the DL relation assumption.

2.2 NOTATION

Here is a list of basic notations and shorthands that we use

- $\mathbb{G}$ is a prime-order group, $\mathbb{F}_{\bar{\mathsf{p}}}$ is its corresponding scalar field.
- $\bar{\mathsf{p}}$ denotes a big prime chosen to be the order of the group $\mathbb{G}$ and, respectively, of its scalar field $\mathbb{F}_{\bar{\mathsf{p}}}$.
- lowercase italic and lowercase Greek letters denote scalars in $\mathbb{F}_{\bar{\mathsf{p}}}$. Apostrophes, hats, and subscript indices can be appended, e.g., a, b_{12}, c', ζ', x_k. Also, lowercase italic and, sometimes, Greek letters denote integers used as indices or limits, e.g., n, i, j_1, s_k, x_π, this usage is clear from context. Superscripts, e.g., ϵ^2, denote scalar exponentiation.
- the special case is the bold superscripts, such as $d^{\mathbf{\Delta sum}}$ or $A^{\mathbf{sum}}$; it stands for regular scalar in $\mathbb{F}_{\bar{\mathsf{p}}}$ or element in $\mathbb{G}$, and the superscript in bold has the purely explanatory meaning.
- bold lowercase italic and bold lowercase Greek letters denote scalar vectors, e.g., $\mathbf{a}$, $\mathbf{b}$, $\boldsymbol{\alpha}$.
- bold lowercase Gothic letters denote scalar matrices, e.g., $\boldsymbol{\mathfrak{a}}$, $\boldsymbol{\mathfrak{b}}$.
- uppercase italic letters denote elements in $\mathbb{G}$. Apostrophes, hats, and subscript indices can be appended, e.g., A, B_{12}, D', P_{s_k}. Multiplication syntax is used to denote element exponentiation by a scalar, e.g., xG.
- bold uppercase italic letters denote element vectors, e.g., $\mathbf{A}$, $\mathbf{P}$.
- $\bar{\mathsf{n}}$ denotes a maximum number of elements in a ring.
- The zero element in $\mathbb{G}$ and the zero scalar in $\mathbb{F}_{\bar{\mathsf{p}}}$ are denoted as 0; it is clear from context which set 0 belongs to. A vector of n zeros is denoted either as $\mathbf{0}^n$ or as $\{0\}^n$, both notations are equivalent.
- asterisk denotes that zero entries are excluded. That is, $\mathbb{F}^*_{\bar{\mathsf{p}}}$ means $\mathbb{F}_{\bar{\mathsf{p}}} \setminus \{0\}$, $\mathbb{G}^*$ means $\mathbb{G} \setminus \{0\}$. Substantially, for vectors, if $\mathbf{x} \in \mathbb{F}^{n*}_{\bar{\mathsf{p}}}$, $\mathbf{P} \in \mathbb{G}^{m*}$, then $\mathbf{x}$ and $\mathbf{P}$ are assumed to contain no zeros in any position.
- star denotes Klein star. For instance, $\mathsf{M} \in \{0, 1\}^\star$ means that M is a bitstring.
- $\mathcal{H}_{\mathbf{scalar}}$ and $\mathcal{H}_{\mathbf{point}}$ are the ideal hash and hash to group (to curve) functions, respectively.
- $A = \text{lin}(\mathbf{B})$, where $\mathbf{B}$ is a non-empty vector of nonzero elements, means that $A = \langle \mathbf{x}, \mathbf{B} \rangle$ for some known vector $\mathbf{x}$. The syntactic sugar $A \sim B$ is equivalent to $A = \text{lin}(\{B\})$.
- $A \mathrel{!=} \text{lin}(\mathbf{B})$, where $\mathbf{B}$ is a non-empty vector of nonzero elements, means that A cannot be represented as a weighted sum of elements in $\mathbf{B}$, except for with negligible probability. The sugar $A \mathrel{!\sim} B$ is equivalent to $A \mathrel{!=} \text{lin}(\{B\})$.
- for any non-empty set $\mathbf{S}$, $\text{ort}(\mathbf{S})$ means that no non-trivial relation between elements in $\mathbf{S}$ can be found. Thus, $\text{ort}(\mathbf{S})$ is a shorthand for the DL relation assumption [7] for $\mathbf{S}$. If $\mathbf{S}$ is a set of $\mathcal{H}_{\mathbf{point}}$ images on different pre-images, then there always holds $\text{ort}(\mathbf{S})$. As an equivalent definition, $\text{ort}(\mathbf{S})$ means that, for each element $E \in \mathbf{S}$, weights for E's representation as a weighted sum of elements in $\mathbf{S} \setminus \{E\}$ cannot be found. Note, if $\mathbf{S}$ contains the zero element, then $\text{ort}(\mathbf{S})$ never holds.
- we say that all elements in $\mathbf{S}$ are orthogonal to each other, iff $\text{ort}(\mathbf{S})$ holds. We emphasize this because 'orthogonal to each other' can be read as pairwise orthogonality, which certainly is a weaker property. Here and elsewhere, by writing that elements in $\mathbf{S}$ are ortogonal to each other we always imply the stronger property, namely, that $\text{ort}(\mathbf{S})$ holds.
- $\text{nz}(\mathbf{B})$ means a subset of $\mathbf{B}$ containing all nonzero elements found in $\mathbf{B}$.
- access to vector and matrix items is performed using Python notation, as in [7]. Also, having a vector $\mathbf{A}$ we imply that A_i denotes i-th item of $\mathbf{A}$, i.e., we imply that A_i is an alias of $\mathbf{A}_{[i]}$ and therefore $A_i = \mathbf{A}_{[i]}$. Often we write 'let $A_i \leftarrow \mathbf{A}_{[i]}$' to preface the use of A_i.
- appending an element into a vector is denoted by comma, e.g., $\hat{\mathbf{X}} \leftarrow [\mathbf{X}, B]$ means that $\hat{\mathbf{X}} = [X_0, \ldots, X_{n-1}, B]$.
- when writing our protocols we mix several assignment styles, they all are construed as imperative assignment. For instance, the expression 'let $x \leftarrow y$' means the same as 'assign $x = y$'. Typically we use 'let $x \leftarrow y$' to indicate that x gets the value of y and both of them won't change.
- as a rule, when we use the letter n to represent an integer, we assume that n is subject to an additional restriction, e.g., that n or $(n + 1)$ is a power of 2. The exact body of this restriction is entirely determined by a concrete vector commitment argument in which this n is directly or indirectly used.
- $\log_2(\ldots)$ is meant as its ceiling $\lceil \log_2(\ldots) \rceil$ everywhere, when used together with integers in formulas.

2.3 COMMONLY AVAILABLE INFORMATION

All the commonly available to both of $\mathcal{P}$ and $\mathcal{V}$ information is shown in Figure 1. This information is also assumed to be accessible in all protocols hereinafter.

Common information
• A big prime number $\bar{\mathsf{p}}$ • Definition of a finite scalar field $\mathbb{F}_{\bar{\mathsf{p}}}$ • Definition of a prime-order group $\mathbb{G}$ over $\mathbb{F}_{\bar{\mathsf{p}}}$ • A generator G of the group $\mathbb{G}$

Figure 1: Information available to each party

2.4 UNDERLYING PROVING SYSTEM

In this paper we construct a number of protocols and use them as building blocks for our signatures. Except for the signatures themselves, each of our protocols is a zero-knowledge argument of knowledge. That is, according to the respective definitions in [7], it is (perfectly) complete , sHVZK, and has cWEE.

Completeness is trivially seen from the code of the protocols, we do not dwell on it. For each argument, we prove that it has cWEE property by constructing a witness extractor. The extractor obtains witness for argument's relation by reading argument's public transcript and making a polynomial number of rewindings on it. For some elementary protocols, instead of explicitly constructing an extractor we refer to the works where it is done in details. For each of our extractors, we also prove that witness obtained by it meets the corresponding relation limits and is unique, we show that otherwise the extractor breaks the DL relation assumption in a polynomial number of steps.

The sHVZK property requires building a simulator in each case. Fortunately, almost (this 'almost' is due to a couple of easy exceptional cases described in Section 2.4.2) all of our arguments can be made zero-knowledge using the concise and currently widely known method presented, e.g., in the works of Attema et al.[2], Chung et al.[9]. Namely, each scalar reply in our public transcripts is by-design masked with an independently and uniformly sampled randomness, whereas each element E in the transcripts is either completely dependent or having the form

$$E = X + \mu H, \tag{3}$$

where X is the value component of E, and μH is the blinding component of E. The blinding generator H is built in such a way to be orthogonal to everything else, and μ is always an independently and uniformly sampled scalar.

It is informally clear why transcripts with these elements reveal no information. Namely, the form (3) is a Pedersen commitment [26, 7], which is perfectly hiding [7]. Formally, we refer to the work [9], where the elements of public transcripts have the same structure and the corresponding simulators are constructed. We will assume that for each of our arguments a simulator is constructed in the same way as in [9], and will not construct it explicitly.

2.4.1 CONNECTION TO SIGNATURES

Having a set of zero-knowledge arguments of knowledge introduced in Section 2.4, which we call an underlying proving system, we build our signatures right on top of it. Since all of the arguments of the underlying proving system are complete, sHVZK, and have cWEE, we can prove security of our signatures using the well known methods.

Namely, to establish unforgeability, anonymity, and other signature properties we refer to the work in [23, 13, 29], where these properties are obtained from the sHVZK and cWEE properties of the underlying proving systems, for the linkable signatures with key image forms $x\mathcal{H}_{\mathbf{point}}(xG)$ or $x^{-1}\mathcal{H}_{\mathbf{point}}(xG)$. We keep in mind that, as we proved in [29], key images in these two forms are indistinguishable from each other.

2.4.2 EXCEPTIONAL SHVZK CASES

We have the only two exceptional sHVZK arguments which do not follow the form (3) for their public transcript elements. Anyway, their sHVZK can be easily established different ways. The first of them is the two-element Schnorr-like scheme in Figure 2, which splits into two Schnorr-id protocols and, hence, can be proven sHVZK by combining outputs of two Schnorr-id simulators.

The second one is the optimized version of our pivot vector commitment argument in Figure 28, previewed in Section 1.5.3. It is sHVZK since its first message T is the sum of elements, each randomized according to the Schnorr-id scheme. At the same time, the scalar vector $\boldsymbol{\tau}$ in it needs not to be hidden. That is, the argument is already sHVZK with open $\boldsymbol{\tau}$, and the replacement of $\boldsymbol{\tau}$ with its proof of knowledge does not revoke the sHVZK property of the entire argument.

As another way of proving the above, it suffices to recall that the argument in Figure 28 is a subset case (with cosmetic differences) of the compressed pivot in [2]. Hence, the proof of sHVZK for the argument in Figure 28 can be borrowed from [2]. Moreover, since the argument in Figure 2 is a subset case of the argument in Figure 28, for both of our exceptional sHVZK cases we can simply refer to the proof in [2].

2.5 INFORMAL INTERPRETATION

Proving the cWEE property for protocols is a necessary and one of the difficult steps when designing a cryptosystem under the DL assumption. However, when creating a completely new protocol, neither cWEE nor sHVZK property definitions give an idea of what it should look like. Fortunately, we can use the following metaphor when constructing the protocols we need. This metaphor allows us to guess what those protocols might be, for which we are likely having a chance to prove that they have cWEE.

The metaphor is that all elements in $\mathbb{G}$ can be thought of as vectors of an infinite-dimensional linear space with countable base $\mathfrak{L}$ over $\mathbb{F}_{\bar{p}}$. A set of orthogonal elements in a protocol corresponds to a set of linearly independent vectors in $\mathfrak{L}$ which determine a linear subspace in it. Note, others vectors of the protocol are not assumed to be belonging to this subspace by default. Addition and multiplication by a scalar in $\mathfrak{L}$ are the same as in $\mathbb{G}$. Calculating the dot product between two vectors in $\mathfrak{L}$ is assumed hard, which corresponds to the DL assumption in $\mathbb{G}$. This metaphor allows for a geometrical interpretation of the protocols.

For example, the well-known Schnorr-id scheme can be interpreted as the following game in $\mathfrak{L}$. For two given vectors G and Y, prover $\mathcal{P}$ must convince verifier $\mathcal{V}$ that Y is collinear to G. Note that $\mathcal{V}$ itself cannot check whether this is the case by taking the dot product between G and Y. So, $\mathcal{P}$ publishes some vector T, then $\mathcal{V}$ issues a challenge c and $\mathcal{P}$ replies with the factor r such that $rG = T - cY$, thus showing that the vector $(T - cY)$ is collinear to G. Since c is random, this convinces $\mathcal{V}$ that both of T and Y are collinear to G.

As another example, consider the simplest case of the reduction by Bünz et al. [7], where $\mathcal{P}$ proves that the given Y belongs to the plane spanned by X_0 and X_1 by demonstrating some L and R such that $\hat{Y} = Y + \epsilon^2 L + \epsilon^{-2} R$ is true for a random ϵ, provided that for $\hat{Y}$ it is already shown that it belongs to the plane spanned by X_0 and X_1. It is easy to see that the vector $(\epsilon^2 L + \epsilon^{-2} R)$ is randomly sampled in the plane spanned by L and R. Therefore, if Y does not belong to the same plane, then $\hat{Y}$ will not be in any predetermined plane. However, as defined right above, it is shown that $\hat{Y}$ belongs to the predetermined plane which is the one spanned by X_0 and X_1. So Y belongs to the plane of L and R and, hence, $\hat{Y}$ belongs to it too. However, $\hat{Y}$ belongs to to the plane of X_0 and X_1, which means that Y, L, R also belong to the plane of X_0 and X_1.

Since this is an informal method, we will not mention it further in the text, except for a few informal explanations. And, of course, we neither consider it as a formal argument nor use it in the formal proofs. Anyway, keeping this metaphor in mind can be helpful in understanding our protocols.

3 ELEMENTARY PROTOCOLS

We begin with the simple protocols, each representing an sHVZK argument of knowledge for the corresponding basic relation. We will use these arguments later in our lemmas and signatures. Although, generally speaking, they can be used independently or as the parts of other systems.

Concrete implementations of the arguments `zk2ElemComm` (Figure 2) and $\mathtt{zkVC}_n$ (Figure 3) are not decisive; other implementations will do, as long as they support the same relations and are complete, sHVZK, and have cWEE. Moreover, in the optimized implementations of our signatures we replace $\mathtt{zkVC}_n$ by $\mathtt{zkVC}_n^{\mathbf{opt}}$ (Figure 28).

Some of the relations given below clearly can be interpreted as definitions of binding commitments, and we call their respective elements commitments. For most of them, their binding property follows directly from binding of Pedersen vector commitment [7]. In any case, for all of our arguments, we prove that their relations have unique witnesses when this is not trivially seen, as we have already pointed out in Section 2.1.3.

As for the hiding property of those elements considered as commitments, we do not require it by default; the sHVZK property of the corresponding arguments suffices for our needs.

3.1 OVERVIEW

3.1.1 TWO ELEMENT COMMITMENT

We call the first helper protocol a two-element commitment argument, and denote it as

$$\mathtt{zk2ElemComm}(X, H, Y;\, x, h).$$

In this notation, the elements X, H, Y are common input for prover and verifier. And the pair of scalars x, h is the prover's private input, it is the witness known only to the prover. The protocol `zk2ElemComm` is an argument for the relation

$$\mathcal{R} = \{ X, H \in \mathbb{G}^*, Y \in \mathbb{G};\ x, h \in \mathbb{F}_{\bar{p}} \mid Y = xX + hH \}, \tag{4}$$

where X and H are orthogonal to each other.

We require `zk2ElemComm` to be sHVZK and to have cWEE. Additionally, we require the witness (x, h) of the relation (4) to be proved unique, which fortunately is trivial. In Figure 2 we provide an uncomplicated implementation of this argument.

Overall, `zk2ElemComm` convinces verifier that prover knows weights in the representation of the element Y as a weighted sum of the orthogonal generators X and H. We implement it as a two-generator extension of the Schnorr identification scheme [27]. Its size is one element in $\mathbb{G}$ and two scalars in $\mathbb{F}_{\bar{p}}$.

The input element Y can be regarded as a commitment that binds its opening (x, h). When h is sampled independently and uniformly, Y becomes hiding as Pedersen commitment. Notable, the `zk2ElemComm` protocol itself remains sHVZK for any distribution of h, including $h = 0$.

3.1.2 BASIC VECTOR COMMITMENT

Vector commitment argument, which will be playing the pivotal role in our paper, is

$$\mathtt{zkVC}_n(\mathbf{X}, H, Y;\ \mathbf{a}, \alpha).$$

It proves knowledge of an unique witness for the relation

$$\mathcal{R} = \{ \mathbf{X} \in \mathbb{G}^{n*}, H \in \mathbb{G}^*, Y \in \mathbb{G};\ \mathbf{a} \in \mathbb{F}_{\bar{p}}^n, \alpha \in \mathbb{F}_{\bar{p}} \mid Y = \langle \mathbf{a}, \mathbf{X} \rangle + \alpha H \}, \tag{5}$$

where all generators in the set $\mathbf{X} \cup \{H\}$ are orthogonal to each other.

This argument convinces verifier that prover knows $(n + 1)$ weights, namely, $\mathbf{a}$ and α, in the decomposition of Y by the generators $\mathbf{X} \cup \{H\}$. The genearator H together with its corresponding weight α is used here to make this argument zero-knowledge, as in [9, 2].

Our implementation of $\mathtt{zkVC}_n$ in Figure 3 is based on the inner product argument implementation from [7], which is provided for the following relation in the original paper

$$\mathcal{R} = \{ \mathbf{G}, \mathbf{H} \in \mathbb{G}^{n*}, U, P \in \mathbb{G};\ \mathbf{a}, \mathbf{b} \in \mathbb{F}_{\bar{p}}^n \mid P = \langle \mathbf{a}, \mathbf{G} \rangle + \langle \mathbf{b}, \mathbf{H} \rangle + \langle \mathbf{a}, \mathbf{b} \rangle\, U \}. \tag{6}$$

We modify this relation and the implementation from [7] the next way. First, since we do not actually need the inner product argument, just only its vector commitment part, we zero out the vector $\mathbf{b}$ in the relation (6). Thus, the inner product $\langle \mathbf{a}, \mathbf{b} \rangle$ becomes equal to zero everywhere. This leaves only the vector commitment argument, i.e., only the argument for the relation

$$\mathcal{R} = \{ \mathbf{G} \in \mathbb{G}^{n*},\ P \in \mathbb{G};\ \mathbf{a} \in \mathbb{F}_{\bar{p}}^n \mid P = \langle \mathbf{a}, \mathbf{G} \rangle \}. \tag{7}$$

Second, we make this argument zero-knowledge not the way it is done in [7], instead we use the straighter way that is as in [9]. Namely, we respectively add the blinding summands αH, βH, and γH to the vector commitment P and to all the L and R elements transmitted during the reduction in [7]. The secret blinding factors β, γ are sampled independently and uniformly by $\mathcal{P}$, the blinding generator H is chosen to be orthogonal to $\mathbf{G}$, hence all the transmitted L's and R's appear to be indistinguishable from random noise. We rename the vector $\mathbf{G}$ and the commitment P in the relation (7) to $\mathbf{X}$ and Y in the relation (5), respectively. The blinding summand αH is taken into account in the relation (5).

Third, for the case $n = 1$ we use our own Schnorr-like sHVZK and cWEE protocol, which is different from sub-protocols used in [7] and [9]. Namely, we use `zk2ElemComm` instead, and this does not alter the properties of the entire $\mathtt{zkVC}_n$ protocol.

Overall, our implementation of $\mathtt{zkVC}_n$ is shown in Figure 3. It has the same properties as the implementation of the inner product argument in [7] with $\mathbf{b} = \mathbf{0}^n$, plus it is sHVZK and, of course, it remains to be having cWEE. Compared to the implementations in [7, 9] our $\mathtt{zkVC}_n$ contains no inner product proof. It proves knowledge of the opening $(\mathbf{a}, \alpha)$ of the vector commitment Y only.

Size of $\mathtt{zkVC}_n$ is $2\lceil \log_2(n) \rceil + 1$ elements in $\mathbb{G}$ and 2 scalar in $\mathbb{F}_{\bar{p}}$. Here and elsewhere, when using this implementation we consider n is a power of 2. Although, as we have already mentioned, our protocols will not be generally bound to a particular realization of $\mathtt{zkVC}_n$ and, hence, when we use its optimized version defined in Section 10, this requirement for n will change.

3.1.3 RANDOM WEIGHTING FOR 3-TUPLES

Another auxiliary argument,

$$\texttt{zk3ElemRW}(P, Q, R, H, Z, F, E;\ a, \alpha, \beta, \gamma)$$

shown in Figure 4, connects a triplet of orthogonal elements (P, Q, R) with a triplet of arbitrary elements (Z, F, E). One of the two elements Q and R in the triplet (P, Q, R) can be zero, in which case the other two elements of the triplet must remain orthogonal to each other. So, the protocol zk3ElemRW is an argument for the relation

$$\mathcal{R} = \left\{ \begin{array}{l} P \in \mathbb{G}^*,\ Q, R \in \mathbb{G},\ H \in \mathbb{G}^*,\ Z, F, E \in \mathbb{G}; \\ a, \alpha, \beta, \gamma \in \mathbb{F}_{\bar{p}} \end{array} \middle| \begin{array}{l} Z\ = aP + \alpha H\ \wedge \\ F\ = aQ + \beta H\ \wedge \\ E\ = aR + \gamma H \end{array} \right\}, \tag{8}$$

where all nonzero elements in the set $\{P, Q, R, H\}$ are required to be orthogonal to each other, which is denoted by $\mathrm{ort}(\mathrm{nz}(P, Q, R, H))$. Also, at least one of Q and R must be nonzero, which is denoted by $(Q + R) \in \mathbb{G}^*$.

The implementation of zk3ElemRW is as follows. $\mathcal{V}$ samples two challenges δ_1 and δ_2, and both $\mathcal{P}$ and $\mathcal{V}$ build the sums X and Y using these challenges. Also, $\mathcal{P}$ builds the total blinding factor $\hat{\alpha}$

$$\begin{aligned} X &= P + \delta_1 Q + \delta_2 R, \\ Y &= Z + \delta_1 F + \delta_2 E, \\ \hat{\alpha} &= \alpha + \delta_1 \beta + \delta_2 \gamma\,. \end{aligned}$$

As the second step, $\mathcal{P}$ proves to $\mathcal{V}$ using an arbitrary external complete, sHVZK, and having cWEE argument that Y is a weighted sum of X and H, with known to $\mathcal{P}$ weights. In course of the proof of Theorem 3 we will show that this suffices to extract unique witness for the relation (8).

Using the shorthands defined in [29], we can also say that a proof of $Y = \mathrm{lin}(X, H)$ holds on $\mathcal{P}$'s side is somehow obtained in the second step of zk3ElemRW. We will often omit everything connected with H as a technical blinding detail, writting this shortly as $Y \sim X$ (to the accuracy of H component).

The cWEE property of zk3ElemRW can be proved the same way as it is done for the RandomWeighting-WEE lemma protocol in [29]. Also, in the proof of Theorem 3 we consider the extreme case, when one of the elements Q and R is zero.

Our requirement $(Q + R) \in \mathbb{G}^*$ may seem excessive, however without it the protocol is not sound (does not have cWEE). For instance, suppose both of Q and R are equal to zero, then $\mathcal{P}$ can let $Z = P, F = 2P, E = 0$. The protocol succeeds on this input, yet witnesses such as a, β remain unknown. Thus, the requirement $(Q + R) \in \mathbb{G}^*$ is highly significant.

3.1.4 SIMMETRIC VECTOR COMMITMENT

We also need an argument to convince verifier that several, e.g., two or three, vector commitments share the same known to prover weights, with the only exclusion for blinding factors which are not shared. That is, we need the argument

$$\texttt{zkSVC}_{3,n}(\mathbf{P}, \mathbf{Q}, \mathbf{R}, H, Z, F, E;\ \mathbf{a}, \alpha, \beta, \gamma)$$

shown in Figure 5 for the following relation

$$\mathcal{R} = \left\{ \begin{array}{l} \mathbf{P} \in \mathbb{G}^{n*}, \mathbf{Q}, \mathbf{R} \in \mathbb{G}^n, H \in \mathbb{G}^*, Z, F, E \in \mathbb{G}; \\ \mathbf{a} \in \mathbb{F}_{\bar{p}}^n, \alpha, \beta, \gamma \in \mathbb{F}_{\bar{p}} \end{array} \middle| \begin{array}{l} Z\ = \langle \mathbf{a}, \mathbf{P} \rangle + \alpha H\ \wedge \\ F = \langle \mathbf{a}, \mathbf{Q} \rangle + \beta H\ \wedge \\ E\ = \langle \mathbf{a}, \mathbf{R} \rangle + \gamma H \end{array} \right\}, \tag{9}$$

where all nonzero elements from the set $\mathbf{P} \cup \mathbf{Q} \cup \mathbf{R} \cup \{H\}$ are orthogonal to each other, which is denoted by $\mathrm{ort}(\mathbf{P} \cup \mathrm{nz}(\mathbf{Q}) \cup \mathrm{nz}(\mathbf{R}) \cup \{H\})$. Also, for each index $i \in [0 \ldots n-1]$, at least one of two elements $\mathbf{Q}_{[i]}$ and $\mathbf{R}_{[i]}$ must be nonzero, which we denote by $(\mathbf{Q} + \mathbf{R}) \in \mathbb{G}^*$.

The relation (9) asserts that the three different vector commitments Z, F, E are sort of 'symmetrical' to each other due to the common weights $\mathbf{a}$ which apply to three different bases $\mathbf{P}, \mathbf{Q}, \mathbf{R}$, respectively. Note, that we require all elements in $\mathbf{P}$ to be nonzero, while vectors $\mathbf{Q}$ and $\mathbf{R}$ are allowed to contain zero elements, provided that for each index there is at least one nonzero element at that index in them. This condition is similar to the requirement $(Q + R) \in \mathbb{G}^*$ imposed by the relation (8) to (P, Q, R) in Section 3.1.3.

Using random weights similar to the way they are used in Section 3.1.3, we reduce the argument $\texttt{zkSVC}_{3,n}$ to the vector commitment argument $\texttt{zkVC}_n$. Namely, for random δ_1 and δ_2, we construct

$$\begin{aligned} \mathbf{X} &= \mathbf{P} + \delta_1 \mathbf{Q} + \delta_2 \mathbf{R}, \\ Y &= Z + \delta_1 F + \delta_2 E, \\ \hat{\alpha} &= \alpha + \delta_1 \beta + \delta_2 \gamma\,, \end{aligned}$$

and call

$$\mathtt{zkVC}_n(\mathbf{X}, H, Y;\ \mathbf{a}, \hat{\alpha}).$$

Upon successful completion of $\mathtt{zkVC}_n$ we see that, by this, n instances of the protocol $\mathtt{zk3ElemRW}$ have been successfully performed, for all the indices $i \in [0 \ldots n-1]$. This means that the relation (8) is fulfilled for each pair of triplets $(\,(P_i, Q_i, R_i),\ (Z_{P_i}, F_{Q_i}, E_{R_i})\,)$ and, therefore, the relation in question (9) is fulfilled. Also, uniqueness of witness of the relation (9) follows from the uniqueness of witness of the relation (8).

In the above, Z_{P_i} denotes P_i's component in a decomposition of Z by the base $\mathbf{P}$, the same for F_{Q_i}, E_{R_i}. We have implicitly assumed that Z, F, E are weighted direct sums of $\mathbf{P}, \mathbf{Q}, \mathbf{R}$, respectively, with weights known to the prover. This is a strong assumption which cannot be made out of thin air. Fortunately, upon successful completion of $\mathtt{zkSVC}_{3,n}$ verifier is primarily convinced that Z, F, E are the mentioned weighted direct sums. Otherwise the protocol witness extractor would be able to break the DL relation assumption.

3.2 FORMAL PRESENTATION

3.2.1 TWO ELEMENT COMMITMENT

Theorem 1:
For two nonzero elements $X, H \in \mathbb{G}^*$ such that they are orthogonal to each other, for an element $Y \in \mathbb{G}$, the protocol $\mathtt{zk2ElemComm}$ in Figure 2 is a complete, sHVZK argument having cWEE for the relation (4) with unique witness.

Proof: Appendix A.
Overview: Section 3.1.1.

$\boxed{\mathtt{zk2ElemComm}(X, H, Y;\ x, h)}$

Relation $\mathcal{R} = \{\, X, H \in \mathbb{G}^*, Y \in \mathbb{G};\ x, h \in \mathbb{F}_{\bar{\mathsf{p}}} \mid Y = xX + hH \,\}$ // (4)

// X, H in $\mathcal{R}$ satisfy $\mathrm{ort}(X, H)$.

$\mathcal{P}$'s input : $(X, H, Y; x, h)$

$\mathcal{V}$'s input : (X, H, Y)

$\mathcal{P}$'s output : none

$\mathcal{V}$'s output: *Accept* or *Reject*

$\boxed{\mathcal{P}}$: $\phi, \psi \leftarrow\!\$\ \mathbb{F}^*_{\bar{\mathsf{p}}}$ and computes $T = \phi X + \psi H$

$\boxed{\mathcal{P} \to \mathcal{V}}$: T

$\boxed{\mathcal{V}}$: $c \leftarrow\!\$\ \mathbb{F}^*_{\bar{\mathsf{p}}}$

$\boxed{\mathcal{V} \to \mathcal{P}}$: c

$\boxed{\mathcal{P}}$: computes
$$\tau = \phi - cx$$
$$\eta = \psi - ch$$

$\boxed{\mathcal{P} \to \mathcal{V}}$: τ, η

$\boxed{\mathcal{V}}$: **return**s *Accept* iff the following holds

$$T \stackrel{?}{=} \tau X + \eta H + cY$$

Figure 2: Zero-knowledge argument for two element commitment relation

3.2.2 BASIC VECTOR COMMITMENT

Theorem 2:
For $n \in \mathbb{N}^*$ such that n is a power of 2, for a vector of nonzero elements $\mathbf{X} \in \mathbb{G}^{n*}$, for a nonzero element $H \in \mathbb{G}^*$ such that $\mathrm{ort}(\mathbf{X} \cup \{H\})$ holds, for an element $Y \in \mathbb{G}$, the protocol $\mathtt{zkVC}_n$ in Figure 3 is a complete, sHVZK argument having cWEE for the relation (5) with unique witness.

Proof: Appendix B.
Overview: Section 3.1.2.

$\boxed{\mathtt{zkVC}_n(\mathbf{X}, H, Y;\ \mathbf{a}, \alpha)}$

Relation $\mathcal{R} = \{\, \mathbf{X} \in \mathbb{G}^{n*}, H \in \mathbb{G}^*, Y \in \mathbb{G};\ \mathbf{a} \in \mathbb{F}_{\bar{p}}^n, \alpha \in \mathbb{F}_{\bar{p}} \mid Y = \langle \mathbf{a}, \mathbf{X} \rangle + \alpha H \,\}$ // (5)

// $\mathbf{X}, H$ in $\mathcal{R}$ satisfy $\mathrm{ort}(\mathbf{X} \cup \{H\})$, n is a power of 2 everytime.

$\mathcal{P}$'s input : $(\mathbf{X}, H, Y; \mathbf{a}, \alpha)$

$\mathcal{V}$'s input : $(\mathbf{X}, H, Y)$

$\mathcal{P}$'s output : none

$\mathcal{V}$'s output: *Accept* or *Reject*

---

if $n > 1$ **then**

$\boxed{\mathcal{P}}$: $\beta, \gamma \leftarrow\!\!\$\ \mathbb{F}_{\bar{p}}^*$ and computes $\hat{n} = n/2$

$$L = \left\langle \mathbf{a}_{[:\hat{n}]}, \mathbf{X}_{[\hat{n}:]} \right\rangle + \beta H$$
$$R = \left\langle \mathbf{a}_{[\hat{n}:]}, \mathbf{X}_{[:\hat{n}]} \right\rangle + \gamma H$$

$\boxed{\mathcal{P} \to \mathcal{V}}$: L, R

$\boxed{\mathcal{V}}$: $e \leftarrow\!\!\$\ \mathbb{F}_{\bar{p}}^*$

$\boxed{\mathcal{V} \to \mathcal{P}}$: e

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: compute $\hat{\mathbf{X}} = e^{-1}\mathbf{X}_{[:\hat{n}]} + e\,\mathbf{X}_{[\hat{n}:]}$

$$\hat{Y} = Y + e^2 L + e^{-2} R$$

$\boxed{\mathcal{P}}$: computes

$$\hat{\mathbf{a}} = e\,\mathbf{a}_{[:\hat{n}]} + e^{-1}\mathbf{a}_{[\hat{n}:]}$$
$$\hat{\alpha} = \alpha + e^2\beta + e^{-2}\gamma$$

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: run $\mathtt{zkVC}_{\hat{n}}(\hat{\mathbf{X}}, H, \hat{Y};\ \hat{\mathbf{a}}, \hat{\alpha})$ // run recursively until n=1

else // n=1

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: let $X_0 \leftarrow \mathbf{X}_{[0]}$

and run $\mathtt{zk2ElemComm}(X_0, H, Y;\ a_0, \alpha)$

endif

Figure 3: Zero-knowledge argument for vector commitment relation

3.2.3 RANDOM WEIGHTING FOR 3-TUPLES

Theorem 3:
For a nonzero element $P \in \mathbb{G}^*$, for a pair of elements $Q, R \in \mathbb{G}$, for a nonzero element $H \in \mathbb{G}^*$ such that $\mathrm{ort}(\mathrm{nz}(P, Q, R, H))$ *holds and at least one of the two elements Q, R is nonzero, the protocol* `zk3ElemRW` *in Figure 4 is a complete, sHVZK argument having cWEE for the relation (8) with unique witness.*

Proof: Appendix C.
Overview: 3.1.3.

$\boxed{\texttt{zk3ElemRW}(P, Q, R, H, Z, F, E;\ a, \alpha, \beta, \gamma)}$

$$\text{Relation } \mathcal{R} = \left\{ \begin{array}{l|l} P \in \mathbb{G}^*,\ Q, R \in \mathbb{G},\ H \in \mathbb{G}^*,\ Z, F, E \in \mathbb{G}; & Z\ = aP + \alpha H\ \wedge \\ a, \alpha, \beta, \gamma \in \mathbb{F}_{\bar{\mathsf{p}}} & F\ = aQ + \beta H\ \wedge \\ & E\ = aR + \gamma H \end{array} \right\} \quad // \ (8)$$

// P, Q, R, H in $\mathcal{R}$ satisfy $\mathrm{ort}(\mathrm{nz}(P, Q, R, H))$ and $(Q + R) \in \mathbb{G}^*$

$\mathcal{P}$'s input : $(P, Q, R, H, Z, F, E; a, \alpha, \beta, \gamma)$

$\mathcal{V}$'s input : (P, Q, R, H, Z, F, E)

$\mathcal{P}$'s output : none

$\mathcal{V}$'s output: $Accept$ or $Reject$

$\boxed{\mathcal{V}}$: $\delta_1, \delta_2 \leftarrow\$ \mathbb{F}^*_{\bar{\mathsf{p}}}$

$\boxed{\mathcal{V} \to \mathcal{P}}$: δ_1, δ_2

$\boxed{\mathcal{P}}$: computes $\hat{\alpha} = \alpha + \delta_1 \beta + \delta_2 \gamma$

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: compute $X = P + \delta_1 Q + \delta_2 R$

$Y = Z + \delta_1 F + \delta_2 E$

and run any complete, sHVZK, and cWEE protocol that convinces $\mathcal{V}$ that the pair $(a, \hat{\alpha})$ is a known to $\mathcal{P}$ witness of the relation (4), that is, that X and Y are connected as $Y = aX + \hat{\alpha} H$

Figure 4: Zero-knowledge argument for two 3-tuples proportional to each other

3.2.4 SIMMETRIC VECTOR COMMITMENT

Theorem 4:
For $n \in \mathbb{N}^*$, for a vector of nonzero elements $\mathbf{P} \in \mathbb{G}^{n*}$, and for a pair of vectors of elements $\mathbf{Q}, \mathbf{R} \in \mathbb{G}^n$ such that $(\mathbf{Q} + \mathbf{R}) \in \mathbb{G}^{n*}$, for a nonzero element $H \in \mathbb{G}^*$ such that $\mathrm{ort}(\mathbf{P} \cup \mathrm{nz}(\mathbf{Q}) \cup \mathrm{nz}(\mathbf{R}) \cup \{H\})$ holds, for three elements $Z, F, E \in \mathbb{G}$, the protocol $\texttt{zkSVC}_{3,n}$ in Figure 5 is a complete, sHVZK argument having cWEE for the relation (9) with unique witness.

Proof: Appendix D.
Overview: 3.1.4.

$\boxed{\texttt{zkSVC}_{3,n}(\mathbf{P}, \mathbf{Q}, \mathbf{R}, H, Z, F, E;\ \mathbf{a}, \alpha, \beta, \gamma)}$

$$\text{Relation } \mathcal{R} = \left\{ \begin{array}{l|l} \mathbf{P} \in \mathbb{G}^{n*}, \mathbf{Q}, \mathbf{R} \in \mathbb{G}^n, H \in \mathbb{G}^*, Z, F, E \in \mathbb{G}; & Z\ = \langle \mathbf{a}, \mathbf{P} \rangle + \alpha H\ \wedge \\ \mathbf{a} \in \mathbb{F}^n_{\bar{\mathsf{p}}}, \alpha, \beta, \gamma \in \mathbb{F}_{\bar{\mathsf{p}}} & F\ = \langle \mathbf{a}, \mathbf{Q} \rangle + \beta H\ \wedge \\ & E\ = \langle \mathbf{a}, \mathbf{R} \rangle + \gamma H \end{array} \right\} \quad // \ (9)$$

// $\mathbf{P}, \mathbf{Q}, \mathbf{R}, H$ in $\mathcal{R}$ satisfy $\mathrm{ort}(\mathbf{P} \cup \mathrm{nz}(\mathbf{Q}) \cup \mathrm{nz}(\mathbf{R}) \cup \{H\})$ and $(\mathbf{Q} + \mathbf{R}) \in \mathbb{G}^{n*}$

$\mathcal{P}$'s input : $(\mathbf{P}, \mathbf{Q}, \mathbf{R}, H, Z, F, E; \mathbf{a}, \alpha, \beta, \gamma)$

$\mathcal{V}$'s input : $(\mathbf{P}, \mathbf{Q}, \mathbf{R}, H, Z, F, E)$

$\mathcal{P}$'s output : none

$\mathcal{V}$'s output: $Accept$ or $Reject$

$\boxed{\mathcal{V}}$: $\delta_1, \delta_2 \leftarrow\$ \mathbb{F}^*_{\bar{\mathsf{p}}}$

$\boxed{\mathcal{V} \to \mathcal{P}}$: δ_1, δ_2

$\boxed{\mathcal{P}}$: computes $\hat{\alpha} = \alpha + \delta_1 \beta + \delta_2 \gamma$

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: compute $\mathbf{X} = \mathbf{P} + \delta_1 \mathbf{Q} + \delta_2 \mathbf{R}$

$Y = Z + \delta_1 F + \delta_2 E$

and run $\texttt{zkVC}_n(\mathbf{X}, H, Y;\ \mathbf{a}, \hat{\alpha})$, or run any other complete, sHVZK, and cWEE protocol for the relation (5)

Figure 5: Zero-knowledge argument for 3 vector commitments with shared weights

As a subset case of the $\texttt{zkSVC}_{3,n}$ protocol in Figure 5, for $\mathbf{R} = \mathbf{0}^n$, we define the protocol $\texttt{zkSVC}_{2,n}$ in Figure 6, requiring for it that all elements in $\mathbf{Q}$ be nonzero.

$$\boxed{\texttt{zkSVC}_{2,n}(\mathbf{P}, \mathbf{Q}, H, P, Q;\ \mathbf{a}, \alpha, \beta)}$$

$$\texttt{zkSVC}_{2,n}(\mathbf{P}, \mathbf{Q}, H, Z, F;\ \mathbf{a}, \alpha, \beta) = \texttt{zkSVC}_{3,n}(\mathbf{P}, \mathbf{Q}, \mathbf{0}^n, H, Z, F, 0;\ \mathbf{a}, \alpha, \beta, 0)$$

// where $\mathbf{P}, \mathbf{Q} \in \mathbb{G}^{n*},\ H \in \mathbb{G}^*,\ Z, F \in \mathbb{G};\ \mathbf{a} \in \mathbb{F}_{\bar{p}}^n,\ \alpha, \beta, \gamma \in \mathbb{F}_{\bar{p}}$

Figure 6: Zero-knowledge argument for 2 vector commitments with shared weights

4 LIN2-CHOICE LEMMA

In this section we present the Lin2-Choice lemma featuring $\texttt{zkLin2Choice}_n$ one-out-of-many proof of membership, which we will use later to create the ring signatures. This lemma is main in our paper.

4.1 OVERVIEW

4.1.1 A LOOK INTO OUR PREVIOUS WORK

In [29] we proved the Lin2-Xor lemma which, informally, allows one to select a pair of elements from two pairs of elements. Formally, it provides an argument for the relation

$$\mathcal{R} = \left\{ \ \mathbf{P}, \mathbf{Q} \in \mathbb{G}^{2*}, Z \in \mathbb{G}^*; s \in [0 \dots 1],\ p, q \in \mathbb{F}_{\bar{p}} \ \middle|\ Z = pP_s + qQ_s \ \right\}, \tag{10}$$

where all generators in $\mathbf{P} \cup \mathbf{Q}$ are orthogonal to each other.

Intuition here is that in the first round of the Lin2-Xor lemma protocol both of the prover and verifier multiply one element in each of the two original pairs (P_0, Q_0) and (P_1, Q_1) by a random challenge, so that each of these two pairs becomes a compound element with its own random 'rotation'. Namely, they become

$$(P_0 + c_0 Q_0) \ \text{ and } \ (P_1 + c_1 Q_1)\,. \tag{11}$$

Here we use the notation and indexing from [29].

In the second round of the Lin2-Xor protocol, the prover and verifier play a sub-protocol convincing the verifier that the element $(Z + r_1 H_1)$ in [29] is a weighted sum of the two compound elements (11) which carry their random 'rotations' c_0 and c_1. It turns out that this weighted sum can have no more than one nonzero weight out of two, otherwise the DL relation assumption would be broken.

In fact, since $P_0, Q_0, P_1, Q_1, Z, H_1$ are fixed from the beginning, and as they are orthogonal to each other, the element $(Z + r_1 H_1)$ has at most one 'degree of freedom' parameterized by r_1. At the same time, each of the elements (11) has exactly one degree of freedom defined by the parameters c_0 and c_1, respectively. Hence, if both of the coefficients a, b in the weighted sum

$$Z + r_1 H_1 = a(P_0 + c_0 Q_0) + b(P_1 + c_1 Q_1) \tag{12}$$

are not equal to zero, then the right-hand side of the equality (12), which has two 'degrees of freedom' with the random parameters c_0 and c_1, is balanced out by one 'degree of freedom' of the left-hand side with the prover-controlled parameter r_1. However, this is impossible without breaking orthogonality of P_0, Q_0, P_1, Q_1 and, therefore, at least one of the two weights a, b must be zero.

Also, in [29], by successive application of the Lin2-Xor lemma $\log_2(n)$ times we proved the Lin2-Selector lemma, which allows to select a pair of elements from n pairs of elements. The Lin2-Selector lemma provides an argument for the relation

$$\mathcal{R} = \left\{ \ \mathbf{P}, \mathbf{Q} \in \mathbb{G}^{n*}, Z \in \mathbb{G}^*; s \in [0 \dots n-1],\ p, q \in \mathbb{F}_{\bar{p}} \ \middle|\ Z = pP_s + qQ_s \ \right\}. \tag{13}$$

4.1.2 LIN2-CHOICE LEMMA PROOF OF MEMBERSHIP

After some consideration, we concluded that instead of proving the relation (13) with the Lin2-Selector lemma protocol, as we did in [29], it would be better to prove it directly, as if the Lin2-Xor lemma were applied to n pairs of elements at once while making an auxiliary call to some external vector commitment argument. We implement this method in the Lin2-Choice lemma now, it is more efficient in size and also leaves more room for optimizing the verification complexity.

In line with the intuition in Section 4.1.1, in the first round we can take n pairs of elements and turn them into n compound elements with random 'rotations'. After that, in the second round, we can prove that $(Z + r_1 H_1)$ is a linear combination of these n compound elements. As a result, exactly the same way as for the linear combination (12), we let the compound element $(Z + r_1 H_1)$ with one 'degree of freedom' controlled by prover with r_1 balance out n 'degrees of freedom' of a weighted sum comprising n compound elements of the form $P_i + c_i Q_i$. That is, we let the following equality hold

$$Z + r_1 H_1 = \sum_{i=0}^{n-1} a_i (P_i + c_i Q_i). \tag{14}$$

The equality (14) can hold only if the vector of coefficients $\mathbf{a} = \{a_i\}_{i=0}^{n-1}$ is one-hot. We skip the edge case $\mathbf{a} = \mathbf{0}^n$ here and will discuss it a bit later. Thus, we obtain an argument for the relation (13) as the two-round game, where in the first round r_1 is chosen in response to n challenges $\{c_i\}_{i=0}^{n-1}$, and in the second round the pivotal vector commitment argument is played as

$$\mathtt{zkVC}_n(\,\{P_i + c_i Q_i\}_{i=0}^{n-1},\ H,\ Z + r_1 H_1\ ;\ \mathbf{a}, \alpha\,).$$

Here H_1 is fixed as in [29], H is an independent orthogonal blinding generator, α is the blinding factor, and $\mathbf{a}$ is one-hot.

Also, since the vector $\mathbf{Q}$ carries only a technical role in the relation (13), now we are getting rid of Q_s in (13) by adding a proof that $q = 0$ everywhere in the signatures. Namely, we are including a proof of $q = 0$ in our current argument. With all this in mind, the Lin2-Choice lemma (Theorem 5) provides the protocol

$$\mathtt{zkLin2Choice}_n(\mathbf{P}, \mathbf{Q}, H, Z; s, p, \alpha)$$

shown in Figure 8, which is sHVZK, has cWEE, and is an argument for the relation

$$\mathcal{R} = \left\{ \begin{array}{l} \mathbf{P}, \mathbf{Q} \in \mathbb{G}^{n*}, H \in \mathbb{G}^*, Z \in \mathbb{G}; \\ s \in [0 \ldots n-1], p, \alpha \in \mathbb{F}_{\bar{p}} \end{array} \;\middle|\; Z = pP_s + \alpha H \right\}, \tag{15}$$

where all elements in $\mathbf{P}, \mathbf{Q}, H$ are orthogonal, i.e., $\mathrm{ort}(\,\mathbf{P} \cup \mathbf{Q} \cup \{H\})$ holds.

Thus, our Lin2-Choice lemma allows to choose exactly one element from the set of orthogonal elements $\mathbf{P} \in \mathbb{G}^{n*}$. Addressing the details, with a simultaneous proof of $q = 0$, the Lin2-Choice lemma protocol $\mathtt{zkLin2Choice}_n$ for the relation (15) performs as follows

- The first $\mathcal{P}$'s message is an element F, which plays the same role as H_1 in [29]. After the first message, both of $\mathcal{P}$ and $\mathcal{V}$ have the elements Z and F.
- All n elements in $\mathbf{Q}$ are multiplied by the challenges $\{c_i\}_{i=0}^{n-1}$, thus $\mathcal{P}$ and $\mathcal{V}$ obtain the vector $\hat{\mathbf{Q}} = \{c_i Q_i\}_{i=0}^{n-1}$.
- $\mathcal{P}$ replies with r, which plays the same role as r_1 in [29].
- $\mathcal{P}$ and $\mathcal{V}$ play $\mathtt{zkSVC}_{2,n}(\mathbf{P}, \hat{\mathbf{Q}}, H, Z, rF;\ \mathbf{a}, \alpha, r\beta)$, where $\mathbf{a}$ is one-hot, H is an orthogonal blinding generator, α and β are blinding factors of Z and F respectively.

In this protocol, we can see that if $\mathbf{a}$ has more than one hot entry, then $\mathtt{zkSVC}_{2,n}$ played in the last step will not complete successfully for the same reason as the equality (14) will not hold for such $\mathbf{a}$. To be precise, the following equality is checked inside $\mathtt{zkSVC}_{2,n}$, and it guarantees $\mathbf{a}$ is one-hot

$$Z + \delta_1 r F = \sum_{i=0}^{n-1} a_i (P_i + \delta_1 c_i Q_i). \tag{16}$$

In addition to this, if $\mathtt{zkSVC}_{2,n}$ completes successfully, then Z's decomposition by the input generators cannot contain elements from $\mathbf{Q}$, as $\mathtt{zkSVC}_{2,n}$ guarantees $Z = \mathrm{lin}(\mathbf{P} \cup \{H\})$.

Now it is a time to discuss the edge cases that are about completely zero weights in the linear combinations. The case $a = b = 0$ for the equality (12) is settled in [29] by some extra checks. Extra checks would also resolve the edge case for the equality (14), however we do not use the latter at all. Instead of (14), our current Lin2-Choice lemma protocol $\mathtt{zkLin2Choice}_n$ resorts to the equality (16), which has the additional random factor δ_1, making any extra checks unnecessary. Actually, if $\mathbf{a} = \mathbf{0}^n$ in the equality (16), then it holds

$$Z + \delta_1 r F = 0\,,$$

where δ_1 is sampled after Z, F, r are published; this proves without any extra checks that Z is equal to zero. To be precise, in this case Z is proved having only the blinding component, recalling all the above equalities are written to the accuracy of H component. Thus, the edge case $\mathbf{a} = \mathbf{0}^n$ in the equality (16) naturally corresponds to the case $p = 0$ in the relation (15).

4.1.3 GENERIC IDEA OF LIN2-CHOICE LEMMA

The Lin2-Choice lemma's membership proof is shown in Figure 8, informally overviewed in Section 4.1.2, and formally presented in Section 4.2. We can take a look at it from a different angle and informally obtain a more generic view of this protocol. This view will clarify a bit the main idea of the Lin2-Choice lemma, and may make it easier to further understand its formal proof. As we usually do in informal explanations, we omit blinding.

The $\mathtt{zkSVC}_{2,n}$ sub-protocol played in the last step of the Lin2-Choice lemma's membership proof, by Theorem 4, convinces verifier that the following system of two equalities holds

$$\begin{cases} Z = \sum_{i=0}^{n-1} a_i P_i & \text{(17a)} \\ rF = \sum_{i=0}^{n-1} a_i c_i Q_i & \text{(17b)} \end{cases}$$

As follows from the lemma's premise, both sets $\mathbf{P} = \{P_i\}_{i=0}^{n-1}$ and $\mathbf{Q} = \{Q_i\}_{i=0}^{n-1}$ are orthogonal. Therefore, the equality (17a) expresses the fact that the weights $\mathbf{a} = \{a_i\}_{i=0}^{n-1}$ are bound by the commitment Z, and thus they are fixed before the challenges $\mathbf{c} = \{c_i\}_{i=0}^{n-1}$ are sampled by the verifier.

With the fixed weights $\mathbf{a}$ and sampled independently and uniformly challenges $\mathbf{c}$, the verifier is convinced that the second set of weights $\mathbf{a} \circ \mathbf{c} = \{a_i c_i\}_{i=0}^{n-1}$, which participates in the equality (17b), has the following structure. For each weight $a_i c_i$, it holds with overwhelming probability that

$$\begin{cases} a_i c_i = 0 \ \text{ iff } \ a_i = 0\,, & \text{(18a)} \\ a_i c_i \ \text{ is distributed independently and uniformly at random otherwise, i.e., when } \ a_i \neq 0 & \text{(18b)} \end{cases}$$

Let us consider two trivial cases of the set $\mathbf{a}$, namely, the zero case and one-hot case. If $\mathbf{a} = \mathbf{0}^n$, then prover easily makes the equality (17b) hold by replying with $r = 0$ or by sending $F = 0$ in the first message of the protocol. In the case of one-hot $\mathbf{a}$, the prover trivially counterweights the sole nonzero weight $a_i c_i$ in the right-hand side of (17b) with r in the left-hand side. Of course, the prover must properly choose F for the first message in this case.

The Lin2-Choice lemma (Theorem 5) states that there is no non-trivial cases for $\mathbf{a}$ in this protocol; the two trivial cases above are the only possible ones.

Now let's make a generalization. As the Lin2-Choice lemma's protocol in Figure 8 invokes $\mathtt{zkSVC}_{2,n}$ only to convince the verifier of the system (17), we can relax this call and require that (17) be proved to the verifier in any suitable way (meaning cWEE and sHVZK), not exclusively by calling to $\mathtt{zkSVC}_{2,n}$. Also, we can weaken the requirement on the weights $\mathbf{a}$ to be precisely bound with the commitment (17a), leaving only the condition that they must be fixed before sampling the challenges $\mathbf{c}$. With these two relaxations, the game of the Lin2-Choice lemma protocol can be viewed as shown in Figure 7.

`Generic idea of the Lin2-Choice lemma protocol`

- On input, both of $\mathcal{P}$ and $\mathcal{V}$ have the set of helper generators $\mathbf{Q} \in \mathbb{G}^{n*}$. Also, $\mathcal{P}$ has the set $\mathbf{a} \in \mathbb{F}_{\bar{p}}^{n}$.
- $\mathcal{P}$ sends to $\mathcal{V}$ an element $F \in \mathbb{G}$ as the first message. At the same time, $\mathcal{P}$ convinces $\mathcal{V}$ by any means, e.g., using a binding commitment, that the set $\mathbf{a}$ is fixed and will not change.
- $\mathcal{V}$ samples the set of challenges $\mathbf{c} \leftarrow\$ \mathbb{F}_{\bar{p}}^{n*}$ and sends them to $\mathcal{P}$.
- $\mathcal{P}$ replies with $r \in \mathbb{F}_{\bar{p}}$.
- $\mathcal{P}$ convinces $\mathcal{V}$ by any (cWEE and sHVZK) means that the equality (17b) holds namely, that $rF = \sum_{i=0}^{n-1} a_i c_i Q_i$.

Figure 7: Generalized view of the Lin2-Choice lemma's argument from Figure 8

Informally, we claim that if the game depicted in Figure 7 completes successfully, then $\mathbf{a}$ on $\mathcal{P}$'s input contains no more than one nonzero scalar. Our rationale for this is based on the multi-dimensional linear space metaphor described in Section 2.5.

That is, the left-hand side of the equality (17b), namely, rF such that F is fixed, represents an 1-dimensional linear subspace (line) of a multi-dimensional linear space. As r is prover-controlled, $\mathcal{P}$ is able to select any point on that line by picking r. At the same time, the right-hand side of (17b), namely, $\sum_{i=0}^{n-1} a_i c_i Q_i$, represents, in accordance with the found structure of the weights $a_i c_i$ (18), an evenly distributed random point in a t-dimensional linear subspace of the same space. According to (18), the number of dimensions t is equal to the number of nonzeros in $\mathbf{a}$. Since for a t-dimensional space such that $t > 1$, there is only a negligible probability that a random point in it happens to be on a given line, we have $t \leqslant 1$, which corresponds to the fact that the number of nonzeros in $\mathbf{a}$ does not exceed 1.

Thus, the above is a brief informal statement and proof of the very idea of the Lin2-Choice lemma. The formal statement of the lemma can be found in the next section.

4.2 FORMAL PRESENTATION

Theorem 5 (Lin2-Choice lemma)**:**
For $n \in \mathbb{N}^*$, for two vectors of nonzero elements $\mathbf{P}, \mathbf{Q} \in \mathbb{G}^{n*}$, for a nonzero element $H \in \mathbb{G}^*$ such that $\mathrm{ort}(\mathbf{P} \cup \mathbf{Q} \cup \{H\})$ *holds, for an element $Z \in \mathbb{G}$, the protocol* $\texttt{zkLin2Choice}_n$ *in Figure 8 is a complete, sHVZK argument having cWEE for the relation (15) with unique witness.*

Proof: Appendix E.
Overview: Section 4.1.2.

For the protocol $\texttt{zkLin2Choice}_n$ in Figure 8, we consider (p, α) as a witness, with the auxiliary index s always recoverable from $(p \neq 0, \alpha)$ in a polynomial time. For $p = 0$, the index s is undefined.

$\boxed{\texttt{zkLin2Choice}_n(\mathbf{P}, \mathbf{Q}, H, Z; s, p, \alpha)}$

Relation $\mathcal{R} = \left\{ \begin{array}{l} \mathbf{P}, \mathbf{Q} \in \mathbb{G}^{n*}, H \in \mathbb{G}^*, Z \in \mathbb{G} \,; \\ s \in [0 \ldots n-1], p, \alpha \in \mathbb{F}_{\bar{p}} \end{array} \middle| \; Z = pP_s + \alpha H \right\}$ // (15)

// $\mathbf{P}, \mathbf{Q}, H$ in $\mathcal{R}$ satisfy $\mathrm{ort}(\mathbf{P} \cup \mathbf{Q} \cup \{H\})$.

$\mathcal{P}$'s input : $(\mathbf{P}, \mathbf{Q}, H, Z;\ s, p, \alpha)$

$\mathcal{V}$'s input : $(\mathbf{P}, \mathbf{Q}, H, Z)$

$\mathcal{P}$'s output : none

$\mathcal{V}$'s output: $Accept$ or $Reject$

$\boxed{\mathcal{P}}$: $q, \beta \leftarrow\!\$\; \mathbb{F}^*_{\bar{p}}$ and assigns **if** $p = 0$ **then** $q = 0$ **endif**
$F = qQ_s + \beta H$

$\boxed{\mathcal{P} \to \mathcal{V}}$: F

$\boxed{\mathcal{V}}$: $\mathbf{c} \leftarrow\!\$\; \mathbb{F}^{n*}_{\bar{p}}$

$\boxed{\mathcal{V} \to \mathcal{P}}$: $\mathbf{c}$

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: compute $\hat{\mathbf{Q}} = \mathbf{c} \circ \mathbf{Q}$

$\boxed{\mathcal{P}}$: takes scalar c_s at index s in $\mathbf{c}$, that is, lets $c_s \leftarrow \mathbf{c}_{[s]}$,
samples $r \leftarrow\!\$\; \mathbb{F}^*_{\bar{p}}$,
assigns **if** $p \neq 0$ **then** $r = c_s p / q$ **endif**
$\hat{\beta} = r\beta$,
and lets $\mathbf{a} = \left\{ \begin{array}{l} a_s = p \quad \text{// that is, } p \text{ is at } s\text{'th position in one-hot } \mathbf{a} \text{ (or, if } p = 0, \text{ then } \mathbf{a} = \mathbf{0}^n) \\ a_i = 0 \text{ for all } i \in [0 \ldots n-1], i \neq s \end{array} \right.$

$\boxed{\mathcal{P} \to \mathcal{V}}$: r

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: let $\hat{F} \leftarrow rF$
and run $\texttt{zkSVC}_{2,n}(\mathbf{P}, \hat{\mathbf{Q}}, H, Z, \hat{F}; \mathbf{a}, \alpha, \hat{\beta})$

Figure 8: Zero-knowledge argument for one element choice relation

5 LINKABLE RING SIGNATURE FOR ONE ACTUAL SIGNER

An immediate practical result of the Lin2-Choice lemma is the linkable ring signature for one signer described in this section.

5.1 ADDITIONAL DEFINITIONS

To create the signature we extend the common information in Figure 1 with the information in Figure 9. It supplies both of the prover and verifier with identical definitions of the scalar hash $\mathcal{H}_{\mathbf{scalar}}$ and hash-to-group $\mathcal{H}_{\mathbf{point}}$ functions, as well as with a common set of orthogonal generators $\mathbf{G}$.

`Additional common information`

- Maximum number of elements in a ring $\bar{\mathsf{n}}$
- Definition of an ideal hash fiction $\mathcal{H}_{\mathbf{scalar}} : \{0,1\}^{\star} \to \mathbb{F}_{\bar{\mathsf{p}}}^{*}$
- Definition of an ideal hash fiction $\mathcal{H}_{\mathbf{point}} : \{0,1\}^{\star} \to \mathbb{G}^{*}$
- A vector of generators $\mathbf{G} = \{G_0, G_1, G_2, \ldots, G_{\bar{\mathsf{n}}-1}\} \in \mathbb{G}^{\bar{\mathsf{n}}*}$
  such that for any set $\mathbf{H}$ of $\mathcal{H}_{\mathbf{point}}$ images on different pre-images it holds $\operatorname{ort}(\mathbf{H} \cup \{G\} \cup \mathbf{G})$

Figure 9: Additional information available to each party

The random oracle is modeled with the scalar hash $\mathcal{H}_{\mathbf{scalar}}$. The hash-to-group (-to-curve) function $\mathcal{H}_{\mathbf{point}}$ is supposed to generate brand new orthogonal elements. The predefined set of orthogonal genarators $\mathbf{G}$ is used in all signature instances, thus reducing verification time when they are verified in a batch.

All public keys used in the signatures can be known to all participants, and there are no additional restrictions on them. That is, as shown in Figure 10, we do not impose any rules on public keys.

`Public keys`

- There is an unlimited amount of public keys generated in the system.
- Each public key is considered visible to all parties, although this is not required.
- Each public key P is assumed to have the following relation to its private key x
  $$P = xG\,,$$
  although this is not required, i.e., there can exist public keys without any known x in this relation.
  There can exist adversarial public keys generated with purpose of breaking signatures.

Figure 10: Public keys seen to all parties

For all of our signature schemes in this paper, we show their unified `Sign` and `Verify` procedures in corresponding figures. Also, for each of them, we always imply presence of one more procedure, `Link`. We do not specify it explicitly since it is constructed trivially, as a comparison of key images I, just as in [23, 13, 29].

5.2 OVERVIEW

Using the argument $\mathtt{zkLin2Choice}_n$ for the relation (15), we construct a ring signature, calling it EFLRS1 (Efficient linkable ring signature for 1 actual signer). Its interactive scheme is shown in Figure 11,

$$\mathtt{EFLRS1.SignAndVerify}_{1,n}(\mathsf{M}, \mathbf{P}; s, x)\,.$$

By the ring we mean a set of $n \geqslant 1$ public keys

$$\mathbf{P} = \{P_i\}_{i=0}^{n-1}\,. \tag{19}$$

Our signature convinces verifier that signer knows a scalar x such that the equality $P_s = xG$ holds for some $s \in [0 \ldots n-1]$. There is no assumption about the public keys in $\mathbf{P}$, except for all they must be different and nonzero which can be easily checked by verifier. Other than that, they can all be regarded as maliciously chosen.

By the decoy set, technically called so, we mean a set of n pairs of the form

$$\{\, (\, P_i + \zeta\mathcal{H}_{\mathbf{point}}(P_i),\ Q_i\,)\,\}_{i=0}^{n-1}\,, \tag{20}$$

where ζ is a random weight. The set $\mathbf{Q}$ of size n contains auxiliary orthogonal generators that can be prepared in advance, provided that $\mathcal{H}_{\mathbf{point}}$ always generates elements which are orthogonal to $\mathbf{Q}$.

Prover publishes key image I defined as

$$I = x^{-1}\mathcal{H}_{\mathbf{point}}(P_s), \tag{21}$$

where x is a private key for the public key $P_s \in \mathbf{P}$ such that $P_s = xG$ holds. Note, the random ζ used in the decoy set above and in Z below is sampled after I is published.

Both of the prover and verifier define the input element Z for the relation (15) as

$$Z = G + \zeta I\,, \tag{22}$$

and sample the blinding generator H to be orthogonal to all the other used generators. Due to the random ζ generated after G and I are published, the element Z defined by (22) always contains nonzero value component, which excludes the case $p = 0$ in the relation (15).

To obtain the signature, it remains to call the protocol of the Lin2-Choice lemma as follows

$$\texttt{zkLin2Choice}_n(\{P_i + \zeta\mathcal{H}_{\mathbf{point}}(P_i)\}_{i=0}^{n-1}, \mathbf{Q}, H, G + \zeta I; s, x^{-1}, 0). \tag{23}$$

It results in the signature of size $2\lceil\log_2(n)\rceil + 6$. When calculating this size, we assume that bitwise representation of an element from $\mathbb{G}$ takes as much space as bitwise representation of a scalar from $\mathbb{F}_{\bar{\mathsf{p}}}$. We count all elements and scalars transmitted from prover to verifier, including the key image I and ignoring the ring of public keys $\{P_i\}_{i=0}^{n-1}$, which is assumed to be known beforehand to both of the prover and verifier.

Also, recalling that any signature is supposed to sign input message M, we implicitly use the well-known method of binding a signature to M which is described, e.g., in [15]. Namely, we assume that our signature's random oracle depends on the input message, and thus the entire series of random values in each of our signatures is bound to M.

5.3 FORMAL PRESENTATION

Theorem 6:
For $n \in \mathbb{N}^*$, for a vector of nonzero elements $\mathbf{P} \in \mathbb{G}^{n*}$ which is considered as a ring of public keys, the protocol EFLRS1 in Figure 11 is a linkable ring signature with the following properties

1. *perfect correctness,*
2. *existential unforgeability against adaptive chosen message / public key attackers,*
3. *unforgeability w.r.t. insider corruption,*
4. *anonymity,*
5. *anonymity w.r.t. chosen public key attackers,*
6. *linkability,*
7. *non-frameability,*
8. *and non-frameability w.r.t. chosen public key attackers.*

Proof: Appendix F.
Overview: Section 5.2.

$\boxed{\texttt{EFLRS1.SignAndVerify}_{1,n}(\mathsf{M}, \mathbf{P}; s, x)}$

$\mathcal{P}$'s input : $(\mathsf{M} \in \{0,1\}^\star, \mathbf{P} \in \mathbb{G}^{n*};\ s \in [0 \ldots n-1], x \in \mathbb{F}_{\bar{p}}^*)$

$\mathcal{V}$'s input : $(\mathsf{M} \in \{0,1\}^\star, \mathbf{P} \in \mathbb{G}^{n*})$

$\mathcal{P}$'s output : *Signature* // signature is a list of all $\mathcal{P} \to \mathcal{V}$ messages from this and nested protocols

$\mathcal{V}$'s output: *Accept* or *Reject*

$\boxed{\mathcal{P}}$: lets $P_s \leftarrow \mathbf{P}_{[s]}$,
assert $x \neq 0$
lets $p \leftarrow x^{-1}$
lets $I \leftarrow p\mathcal{H}_{\mathbf{point}}(P_s)$

$\boxed{\mathcal{P} \to \mathcal{V}}$: I

$\boxed{\mathcal{V}}$: $\epsilon, \zeta \leftarrow\!\!\$\ \mathbb{F}_{\bar{p}}^*$

$\boxed{\mathcal{V} \to \mathcal{P}}$: ϵ, ζ

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: **assert** all elements in $\mathbf{P}$ are nonzero and different
let $\mathbf{U} \leftarrow \{\mathcal{H}_{\mathbf{point}}(\mathbf{P}_{[i]})\}_{i=0}^{n-1}$,
$H \leftarrow \mathcal{H}_{\mathbf{point}}(\epsilon)$ // thus, $\mathrm{ort}(H, \mathbf{G}, \mathbf{P}, \mathbf{U}, Z, I)$ holds
compute $\hat{\mathbf{P}} = \mathbf{P} + \zeta\mathbf{U}$
$Z = G + \zeta I$,
and run $\texttt{zkLin2Choice}_n(\hat{\mathbf{P}}, \mathbf{G}_{[:n]}, H, Z; s, p, 0)$

Figure 11: EFLRS1 signing and verification

5.4 SIZE AND VERIFICATION COMPLEXITY

At runtime, the protocol $\texttt{EFLRS1.SignAndVerify}_{1,n}$ in Figure 11 runs the series of nested subprotocols up to calling zk2ElemComm, as shown in the top box in Figure 12. As a result, assuming that verifier postpones all calculations on its side until the end of the message exchange, the verifier has only to check one expanded equality shown in Figure 12.

$\boxed{\texttt{SignAndVerify}_{1,n} \hookrightarrow \texttt{zkLin2Choice}_n \hookrightarrow \texttt{zkSVC}_{2,n} \hookrightarrow \texttt{zkVC}_n \hookrightarrow \texttt{zk2ElemComm}}$

// Function $\texttt{bitAtPos}(i, j)$ returns j-th bit of binary representation of i

$$c\left(G + \zeta I + \delta_1 rF + \sum_{j=0}^{\log_2(n)-1} (e_j^2 L_j + e_j^{-2} R_j)\right) + \eta H - T + \tau \sum_{i=0}^{n-1} \left(\prod_{j=0}^{\log_2(n)-1} e_j^{2\cdot\texttt{bitAtPos}(i,j)-1}\right)(P_i + \zeta U_i + \delta_1 c_i G_i) = 0$$

Figure 12: Unfolded equality for EFLRS1, verifier checks it

Table 3 shows the size and verification complexity of a batch of l EFLRS1 signatures that are created using a shared ring of n public keys. We consider l signatures in order to compare their summary size and complexity against a threshold variant presented later in this paper. To see the size and verification complexity of a single signature, simply let $l = 1$.

To verify the batch, verifier combines l instances of the equality in Figure 12 together using random weighting. As in [7, 9, 29], the verifier computes all the scalar weights with scalar-scalar multiplications, which are assumed consuming negligibly time, and then performs the single multi-exponentiation to calculate at once l randomly weighted instances of the left-hand side of the equality in Figure 12.

Table 3: **EFLRS1** signature size and verification complexity

	Size	Verification complexity
EFLRS1	$l\,(2\lceil\log_2(n)\rceil + 6)$	***mexp*** $(3n + 2l\log_2(n) + 3l + 2) + (n+1)\mathbf{H_{pt}}$

6 LINKABLE THRESHOLD RING SIGNATURE

To create a threshold version of the EFLRS1 signature, we will define an auxiliary protocol $\texttt{zkMVC}_{l,n}$ that proves the same as l instances of $\texttt{zkVC}_n$ prove. Then, by running in parallel l instances of $\texttt{zkLin2Choice}_n$ and by substituting for l nested in them calls of $\texttt{zkVC}_n$ one call of $\texttt{zkMVC}_{l,n}$, we will get a many-out-of-many proof of membership, from which we will create the threshold version of EFLRS1, calling it EFLRSL.

6.1 OVERVIEW

6.1.1 MULTIPLE VECTOR COMMITMENTS

To obtain the necessary many-out-of-many proof, we need one more helper zero-knowledge argument, namely, a proof of multiple vector commitments

$$\texttt{zkMVC}_{l,n}(\mathbf{X}, H, \mathbf{Y};\ \mathfrak{a}, \boldsymbol{\alpha}).$$

For a given element vector $\mathbf{Y} \in \mathbb{G}^l$, it proves that every $Y_i \in \mathbf{Y}$ is a vector commitment over the vector of orthogonal generators $\mathbf{X} \cup \{H\} \in \mathbb{G}^{n*} \times \mathbb{G}^*$. It is shown in Figure 13 and, formally, is an argument for the relation

$$\mathcal{R} = \{\, \mathbf{X} \in \mathbb{G}^{n*}, H \in \mathbb{G}^*, \mathbf{Y} \in \mathbb{G}^l;\ \mathfrak{a} \in \mathbb{F}_{\bar{p}}^{l\times n}, \boldsymbol{\alpha} \in \mathbb{F}_{\bar{p}}^{l} \mid \mathbf{Y} = \mathfrak{a} \cdot \mathbf{X} + \boldsymbol{\alpha} \cdot H \,\}\,. \tag{24}$$

The relation (24) is a union of l instances of the relation (5). The structure of the $\texttt{zkMVC}_{l,n}$ protocol is quite simple. All l elements in the vector $\mathbf{Y}$ are combined into one element Y using random weights. Then, the argument $\texttt{zkVC}_n$ proves that Y is a vector commitment over the generators $\mathbf{X} \cup \{H\}$, thus convincing verifier that, due to the random weights, every $Y_i \in \mathbf{Y}$ is a vector commitment over $\mathbf{X} \cup \{H\}$.

This way we obtain a proof for a set of vector commitments at the price (space) of a proof for one vector commitment. A similar construction can be found in [3]. This effect, where multiplication by random weights yields multiple proofs for the price of one, propagates to the other relations such as (25), (36). Although, of course, this effect itself as well as its propagation must be formally proved, which we do onward.

6.1.2 MANY-OUT-OF-MANY PROOF

According to the relation (24), the protocol $\texttt{zkMVC}_{l,n}$ proves the same as l $\texttt{zkVC}_n$ protocols prove. Using it, in Figure 14 we construct an efficient many-out-of-many proof of membership

$$\texttt{zkLin2mChoice}_{n,l}(\mathbf{P}, \mathbf{Q}, H, \mathbf{Z}; \mathbf{s}, \mathbf{p}, \boldsymbol{\alpha}),$$

which is an argument for the relation

$$\mathcal{R} = \left\{ \begin{array}{l} \mathbf{P}, \mathbf{Q} \in \mathbb{G}^{n*}, H \in \mathbb{G}^*, \mathbf{Z} \in \mathbb{G}^l; \\ \mathbf{s} \in [0 \ldots n-1]^l, \mathbf{p}, \boldsymbol{\alpha} \in \mathbb{F}_{\bar{p}}^l \end{array} \;\middle|\; \begin{array}{l} \forall k \in [0 \ldots l-1] : \\ Z_k = p_k P_{s_k} + \alpha_k H \end{array} \right\}, \tag{25}$$

where $\mathbf{P}, \mathbf{Q}, H$ satisfy $\mathrm{ort}(\, \mathbf{P} \cup \mathbf{Q} \cup \{H\})$.

The many-out-of-many proof of membership $\texttt{zkLin2mChoice}_{n,l}$ proves the same as l concurrent insances of the one-out-of-many proof of membership $\texttt{zkLin2Choice}_n$ prove, at the price of one instance. All of these l concurrent instances of $\texttt{zkLin2Choice}_n$ are considered invoking all their nested sub-protocols simultaneously. We depict this as the following invocation stack

$$l \times \texttt{zkLin2Choice}_n \hookrightarrow l \times \texttt{zkSVC}_{2,n} \hookrightarrow\ l \times \texttt{zkVC}_n\,. \tag{26}$$

Since all of these l running instances of $\texttt{zkLin2Choice}_n$ are completely independent of each other, we let all the protocol challenges be shared between them, provided that the random oracle which generates the challenges takes into account all the filled in parts of the common transcript.

The final $l \times \texttt{zkVC}_n$ calls on the invocation stack (26) are made only for the sake of proving that each of l vector commitments, namely, each element of the set

$$\{Z_k + \delta_1 r_k F_k\}_{k=0}^{l-1},$$

is constructed over the common set of orthogonal generators

$$\{P_i + \delta_1 c_i Q_i\}_{i=0}^{n-1}.$$

Hence, we can replace all these $l \times \texttt{zkVC}_n$ calls, which altogether prove l instances of the relation (5), with one call to $\texttt{zkMVC}_{l,n}$ which proves the relation (24). After that, the invocation stack (26) starts to look as

$$l \times \texttt{zkLin2Choice}_n \hookrightarrow l \times \texttt{zkSVC}_{2,n} \hookrightarrow\ \texttt{zkMVC}_{l,n}\,.$$

6.1.3 SIGNATURE EFLRSL

The EFLRS1 signature in Figure 11 is a game in which prover builds the key image I of type (21), publishes it, then verifier issues the challenge ζ. Then, using the one-out-of-many proof of membership $\mathtt{zkLin2Choice}_n$, the prover convinces the verifier that Z built by the formula (22) belongs to the decoy set built by the formula (20), namely, to the set

$$(\mathbf{P} + \zeta\mathbf{U})\,,\ \text{ where } \mathbf{U} = \{\mathcal{H}_{\mathbf{point}}(P_i)\}_{i=0}^{n-1} \text{ and, also, the auxiliary set } \mathbf{Q} \text{ is omitted.}$$

Now, instead of one key image I, let the prover publish the following vector of l key images of type (21) each

$$\mathbf{I} = \{I_k\}_{k=0}^{l-1}\,,$$

which correspond to l different indices $\mathbf{s} = \{s_k\}_{k=0}^{l-1}$. We call $\mathbf{s}$ actual signing indices or, equivalently, actual signers in the ring. The corresponding signing private keys $\mathbf{x} = \{x_k\}_{k=0}^{l-1}$ are assumed to be known to the prover.

Taking a randomly sampled ζ both of the prover and verifier construct l values of Z by the formula (22), i.e., they construct the vector

$$\mathbf{Z} = \{Z_k\}_{k=0}^{l-1} = \{G\}^l + \zeta\,\mathbf{I} = \{G + \zeta I_k\}_{k=0}^{l-1}\,.$$

And, also, they build a decoy set by the formula (20). After that, as the last step, they play the $\mathtt{zkLin2Choice}_n$ one-out-of-many proof protocol l times, for the same decoy set and for each $Z_k,\ k \in [0 \ldots l-1]$. We depict this as

$$l \times \mathtt{zkLin2Choice}_n\,.$$

As shown in Section 6.1.2, instead of playing the one-out-of-many proof protocol l times, they can play as well the many-out-of-many proof protocol $\mathtt{zkLin2mChoice}_{n,l}$ only once. By doing so, they obtain the threshold version of the signature, which we call EFLRSL (Efficient linkable ring signature for l actual signers). Its scheme

$$\mathtt{EFLRSL.SignAndVerify}_{l,n}(\mathsf{M}, \mathbf{P}; \mathbf{s}, \mathbf{x})$$

is shown in Figure 15. Its size is $2\lceil \log_2(n) \rceil + 3l + 3$, where the key image vector $\{I_k\}_{k=0}^{l-1}$ is also counted. The ring $\mathbf{P}$ is, as usual, assumed to be known beforehand for both of the prover and verifier.

6.2 FORMAL PRESENTATION

6.2.1 MULTIPLE VECTOR COMMITMENTS

Theorem 7:
For $n, l \in \mathbb{N}^*$, for a vector of nonzero elements $\mathbf{X} \in \mathbb{G}^{n*}$, for a nonzero element $H \in \mathbb{G}^*$ such that $\mathrm{ort}(\mathbf{X} \cup \{H\})$ holds, for a vector of elements $\mathbf{Y} \in \mathbb{G}^l$, the protocol $\mathtt{zkMVC}_{l,n}$ in Figure 13 is a complete, sHVZK argument having cWEE for the relation (24) with unique witness.

Proof: Appendix H.
Overview: Section 6.1.1.

$\boxed{\mathtt{zkMVC}_{l,n}(\mathbf{X}, H, \mathbf{Y};\ \mathfrak{a}, \boldsymbol{\alpha})}$

Relation $\mathcal{R} = \{\, \mathbf{X} \in \mathbb{G}^{n*}, H \in \mathbb{G}^*, \mathbf{Y} \in \mathbb{G}^l;\ \mathfrak{a} \in \mathbb{F}_{\bar{\mathsf{p}}}^{l \times n}, \boldsymbol{\alpha} \in \mathbb{F}_{\bar{\mathsf{p}}}^{l} \mid \mathbf{Y} = \mathfrak{a} \cdot \mathbf{X} + \boldsymbol{\alpha} \cdot H \,\}$ // (24)

// $\mathbf{X}, H$ in $\mathcal{R}$ satisfy $\mathrm{ort}(\mathbf{X} \cup \{H\})$.

$\mathcal{P}$'s input : $(\mathbf{X}, H, \mathbf{Y}; \mathfrak{a}, \boldsymbol{\alpha})$

$\mathcal{V}$'s input : $(\mathbf{X}, H, \mathbf{Y})$

$\mathcal{P}$'s output : none

$\mathcal{V}$'s output: *Accept* or *Reject*

$\boxed{\mathcal{V}}$: $\boldsymbol{\xi} \leftarrow\!\$\ \mathbb{F}_{\bar{\mathsf{p}}}^{l*}$

$\boxed{\mathcal{V} \to \mathcal{P}}$: $\boldsymbol{\xi}$

$\boxed{\mathcal{P}}$: computes $\quad \mathbf{a}^\top = \boldsymbol{\xi}^\top \cdot \mathfrak{a}$

$\qquad\qquad\qquad\quad \alpha = \langle \boldsymbol{\xi}, \boldsymbol{\alpha} \rangle$

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: compute $Y = \langle \boldsymbol{\xi}, \mathbf{Y} \rangle$
and run $\mathtt{zkVC}_n(\mathbf{X}, H, Y;\ \mathbf{a}, \alpha)$

Figure 13: Zero-knowledge argument for multiple vector commitments

6.2.2 MANY-OUT-OF-MANY PROOF

Theorem 8:
For $n \in \mathbb{N}^*$, for two vectors of nonzero elements $\mathbf{P}, \mathbf{Q} \in \mathbb{G}^{n*}$, for a nonzero element $H \in \mathbb{G}^*$ such that $\mathrm{ort}(\mathbf{P} \cup \mathbf{Q} \cup \{H\})$ *holds, for a vector of elements $\mathbf{Z} \in \mathbb{G}^l$, the protocol $\mathtt{zkLin2mChoice}_{n,l}$ in Figure 14 is a complete, sHVZK argument having cWEE for the relation (25) with unique witness.*

Proof: Appendix I.
Overview: Section 6.1.2.

$\boxed{\mathtt{zkLin2mChoice}_{n,l}(\mathbf{P}, \mathbf{Q}, H, \mathbf{Z}; \mathbf{s}, \mathbf{p}, \boldsymbol{\alpha})}$

$$\text{Relation } \mathcal{R} = \left\{ \begin{array}{l|l} \mathbf{P}, \mathbf{Q} \in \mathbb{G}^{n*}, H \in \mathbb{G}^*, \mathbf{Z} \in \mathbb{G}^l; & \forall k \in [0 \ldots l-1] : \\ \mathbf{s} \in [0 \ldots n-1]^l, \mathbf{p}, \boldsymbol{\alpha} \in \mathbb{F}_{\bar{p}}^l & Z_k = p_k P_{s_k} + \alpha_k H \end{array} \right\} \quad // \text{ (25)}$$

// $\mathbf{P}, \mathbf{Q}, H$ in $\mathcal{R}$ satisfy $\mathrm{ort}(\mathbf{P} \cup \mathbf{Q} \cup \{H\})$.

$\mathcal{P}$'s input : $(\mathbf{P}, \mathbf{Q}, H, \mathbf{Z}; \mathbf{s}, \mathbf{p}, \boldsymbol{\alpha})$
$\mathcal{V}$'s input : $(\mathbf{P}, \mathbf{Q}, H, \mathbf{Z})$
$\mathcal{P}$'s output : none
$\mathcal{V}$'s output: *Accept* or *Reject*

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: allocate $\hat{\mathbf{X}} \in \mathbb{G}^n$, $\mathbf{Y} \in \mathbb{G}^l$, $\mathfrak{a} \in \mathbb{F}_{\bar{p}}^{l \times n}$, $\hat{\boldsymbol{\alpha}} \in \mathbb{F}_{\bar{p}}^l$,
and run the following block, depicted as foreach, in l parallel threads (with shared challenges),
using common $\hat{\mathbf{X}}, \mathbf{Y}, \mathfrak{a}, \hat{\boldsymbol{\alpha}}$
foreach $k \in [0 \ldots l-1]$ // execute in parallel
let $(Z_k, s_k, p_k, \alpha_k) \leftarrow (\mathbf{Z}_{[k]}, \mathbf{s}_{[k]}, \mathbf{p}_{[k]}, \boldsymbol{\alpha}_{[k]})$,
run $\mathtt{zkLin2Choice}_n(\mathbf{P}, \mathbf{Q}, H, Z_k; s_k, p_k, \alpha_k)$ without calling nested $\mathtt{zkVC}_n(\mathbf{X}, H, Y; \mathbf{a}, \hat{\alpha})$ in it,
instead assign $\hat{\mathbf{X}} = \mathbf{X}$ // $\mathbf{X}$ is the same in all threads
$\mathbf{Y}_{[k]} = Y$
$\mathfrak{a}_{[k]} = \mathbf{a}$
$\hat{\boldsymbol{\alpha}}_{[k]} = \hat{\alpha}$.
endforeach
run $\mathtt{zkMVC}_{l,n}(\hat{\mathbf{X}}, H, \mathbf{Y}; \mathfrak{a}, \hat{\boldsymbol{\alpha}})$

Figure 14: Zero-knowledge argument for multiple element choice relation

By the same reason as for the protocol $\mathtt{zkLin2Choice}_n$ in Figure 8, we consider $(\mathbf{p}, \boldsymbol{\alpha})$ as witness for the protocol $\mathtt{zkLin2mChoice}_{n,l}$ in Figure 14. The auxiliary indices $\mathbf{s}$ are recoverable from the witness in a polynomial time.

6.2.3 SIGNATURE EFLRSL

Theorem 9:
For $n, l \in \mathbb{N}^*$ such that $l \leqslant n$, for a vector of nonzero elements $\mathbf{P} \in \mathbb{G}^{n*}$ which is considered as a ring of public keys, the protocol EFLRSL in Figure 15 is a linkable threshold ring signature with the following properties

1. *perfect correctness,*
2. *existential unforgeability against adaptive chosen message / public key attackers,*
3. *unforgeability w.r.t. insider corruption,*
4. *anonymity,*
5. *anonymity w.r.t. chosen public key attackers,*
6. *linkability,*
7. *non-frameability,*
8. *non-frameability w.r.t. chosen public key attackers.*

Proof: Appendix J.2.
Overview: Section 6.1.3.

$\texttt{EFLRSL.SignAndVerify}_{l,n}(\mathsf{M}, \mathbf{P}; \mathbf{s}, \mathbf{x})$

$\mathcal{P}$'s input : $(\mathsf{M} \in \{0,1\}^{\star}, \mathbf{P} \in \mathbb{G}^{n*};\ \mathbf{s} \in [0 \ldots n-1]^{l}, \mathbf{x} \in \mathbb{F}_{\bar{\mathsf{p}}}^{l*})$

$\mathcal{V}$'s input : $(\mathsf{M} \in \{0,1\}^{\star}, \mathbf{P} \in \mathbb{G}^{n*})$

$\mathcal{P}$'s output : $\mathit{Signature}$ // signature is a list of all $\mathcal{P} \to \mathcal{V}$ messages from this and nested protocols

$\mathcal{V}$'s output: Accept or Reject

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: **assert** all elements in $\mathbf{P}$ are nonzero and different

let $\mathbf{U} \leftarrow \{\mathcal{H}_{\mathbf{point}}(\mathbf{P}_{[i]})\}_{i=0}^{n-1}$

$\boxed{\mathcal{P}}$: allocates $\mathbf{I} \in \mathbb{G}^{l*}$, $\mathbf{p} \in \mathbb{F}_{\bar{\mathsf{p}}}^{l*}$,

initializes **foreach** $k \in [0 \ldots l-1]$

assert $\mathbf{x}_{[k]} \neq 0$

$\mathbf{p}_{[k]} = \mathbf{x}_{[k]}^{-1}$

lets $(s_k, p_k) \leftarrow (\mathbf{s}_{[k]}, \mathbf{p}_{[k]})$,

$\mathbf{I}_{[k]} = p_k\, \mathbf{U}_{[s_k]}$ // vector $\mathbf{I}$ is filled in here

endforeach

$\boxed{\mathcal{P} \to \mathcal{V}}$: $\mathbf{I}$

$\boxed{\mathcal{V}}$: **assert** all elements in $\mathbf{I}$ are different // here $\mathcal{V}$ makes sure there is no actual signer signing twice

$\epsilon, \zeta \leftarrow\!\!\$\ \mathbb{F}_{\bar{\mathsf{p}}}^{*}$

$\boxed{\mathcal{V} \to \mathcal{P}}$: ϵ, ζ

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: let $H \leftarrow \mathcal{H}_{\mathbf{point}}(\epsilon)$ // thus, $\mathrm{ort}(H, \mathbf{G}, \mathbf{P}, \mathbf{U}, \mathbf{Z}, \mathbf{I})$ holds

compute $\hat{\mathbf{P}} = \mathbf{P} + \zeta \mathbf{U}$,

$\mathbf{Z} = \{G\}^{l} + \zeta \mathbf{I}$

run $\texttt{zkLin2mChoice}_{n,l}(\hat{\mathbf{P}}, \mathbf{G}_{[:n]}, H, \mathbf{Z};\ \mathbf{s}, \mathbf{p}, \{0\}^{l})$

Figure 15: EFLRSL signing and verification

6.3 SIZE AND COMPLEXITY

The only equality that verifier has to check in order to verify authenticity of the EFLRSL signature, is shown in Figure 16. The signature size and verification complexity are provided in Table 4.

$\boxed{\texttt{SignAndVerify}_{l,n} \hookrightarrow\ \times \texttt{zkLin2Choice}_{n} \hookrightarrow l \times \texttt{zkSVC}_{2,n} \hookrightarrow \texttt{zkMVC}_{l,n} \hookrightarrow \texttt{zkVC}_{n} \hookrightarrow \texttt{zk2ElemComm}}$

// Function $\texttt{bitAtPos}(i, j)$ returns j-th bit of binary representation of i

$$c\left(\sum_{k=0}^{l-1} \xi_k (G + \zeta I_k + \delta_1 r_k F_k) + \sum_{j=0}^{\log_2(n)-1} (e_j^2 L_j + e_j^{-2} R_j)\right) + \eta H - T +$$
$$+ \tau \sum_{i=0}^{n-1} \left(\prod_{j=0}^{\log_2(n)-1} e_j^{2 \cdot \texttt{bitAtPos}(i,j) - 1}\right)(P_i + \zeta U_i + \delta_1 c_i G_i) = 0$$

Figure 16: Unfolded equality for EFLRSL, verifier checks it

Table 4: **EFLRSL** signature size and verification complexity

	Size	Verification complexity
EFLRSL[*]	$2\lceil \log_2(n) \rceil + 3l + 3$	$\boldsymbol{mexp}\left(3n + 2\log_2(n) + 2l + 3\right) + (n+1)\mathbf{H_{pt}}$

[*] Optimized size is shown in Table 7.

By comparing Table 4 and Table 3, we may observe that the treshold variant of the signature is asymptotically

l times more compact in size. Also, it is asymptotically slightly faster in verification.

7 LIN2-2CHOICE LEMMA

The Lin2-Choice lemma protocol in Figure 8 made it possible to us to select one element Z from the set $\mathbf{P}$ of orthogonal elements. Now, we are going to extend this protocol so that we can select two elements from $\mathbf{P}$ at a time, instead of one. That is, now we want Z to be a weighted sum of two elements from $\mathbf{P}$. We do not require the index of the second chosen element to be anonymous, however we want its weight to remain securely hidden.

For this purpose, we need to extend the $\mathtt{zkLin2Choice}_n$ protocol with a part that will be responsible for the second element. We will introduce such an extension in Figure 17, and in the Simplified Lin2-2Choice lemma (Theorem 10) will prove its properties as an one-out-of-many proof with an additional element. Next, like with the transition from $\mathtt{zkLin2Choice}_n$ to $\mathtt{zkLin2mChoice}_{n,l}$ in Section 6.1.2, we will proceed to the many-out-of-many proof represented by the Lin2-2Choice lemma (Theorem 12) protocol in Figure 19.

7.1 OVERVIEW

7.1.1 SIMPLIFIED LIN2-2CHOICE LEMMA

By 1-out-of-many membership proof with an additional element we mean an argument about the element in question Z being the sum of two elements Z_P and Z_V such that prover knows the scalar pair (p, v) and the following three equalities hold

$$\begin{cases} Z & = Z_P + Z_V \\ Z_P & = pP_s \ , \quad \text{where } P_s \in \mathbf{P} \\ Z_V & = vV_t \ , \quad \text{where } V_t \in \mathbf{V} \end{cases} .$$

All elements in the set $\mathbf{P} \cup \mathbf{V}$ are assumed to be orthogonal.

The protocol $\mathtt{zkLin22sChoice}_{n,m}(\mathbf{P}, \mathbf{Q}, \mathbf{V}, \mathbf{W}, H, Z, t; s, p, v, \alpha)$ in Figure 17 is such an argument. Formally, it convinces verifier that prover knows witness (s, p, v, α) for the relation

$$\left\{ \begin{array}{l} \mathbf{P}, \mathbf{Q} \in \mathbb{G}^{n*}, \mathbf{V}, \mathbf{W} \in \mathbb{G}^{m*}, H \in \mathbb{G}^*, Z \in \mathbb{G}, t \in [0 \dots m-1] \ ; \\ s \in [0 \dots n-1], p, v, \alpha \in \mathbb{F}_{\bar{p}} \end{array} \ \middle| \ Z = pP_s + vV_t + \alpha H \right\} . \tag{27}$$

In this relation, for $V_t \in \mathbf{V}$, we hide only its weight v, not its index t. The vectors $\mathbf{P}, \mathbf{Q} \in \mathbb{G}^{n*}$, $\mathbf{V}, \mathbf{W} \in \mathbb{G}^{m*}$ are the common prover and verifier input. All $2(n+m)$ elements in these four vectors are orthogonal to each other. The vectors $\mathbf{Q}$ and $\mathbf{W}$ are for technical purposes only, while the vectors $\mathbf{P}$ and $\mathbf{V}$ are used to compose the element $Z = pP_s + vV_t$, where s, p, v are secret, and t is public.

Naturally, in the case $m = 0$, $\mathbf{V} = \mathbf{W} = \varnothing$, $Z_V = 0$, the protocol $\mathtt{zkLin22sChoice}_{n,m}$ turns into the regular 1-out-of-many membership proof $\mathtt{zkLin2Choice}_n$ provided by the Lin2-Choice lemma in Section 4.

Now let's move on to the design of the protocol $\mathtt{zkLin22sChoice}_{n,m}$ in Figure 17. It is constructed from $\mathtt{zkLin2Choice}_n$ in Figure 8 as follows.

- $\mathcal{P}$ hands over the following pair of elements to $\mathcal{V}$, instead of the single element F in $\mathtt{zkLin2Choice}_n$,

$$F \ \text{ and } \ E \ . \tag{28}$$

- $\mathcal{V}$ generates a set of $(n+m)$ challenges $\{c_i\}_{i=0}^{n+m-1}$.
- $\mathcal{P}$ and $\mathcal{V}$ construct a decoy set comprising two parts, of total size $n+m$. The first part of the decoy set, of size n, contains the following triplets

$$\{(P_i,\ c_i Q_i,\ 0)\}_{i=0}^{n-1} \ , \tag{29}$$

whereas the second one, which is new, of size m, contains the following triplets

$$\{(V_i,\ 0,\ c_{n+i} W_i)\}_{i=0}^{m-1} \ . \tag{30}$$

- $\mathcal{P}$ replies with the scalar r, as in $\mathtt{zkLin2Choice}_n$, and then the following two elements are constructed

$$rF \ , \ c_{n+t} E \ . \tag{31}$$

- As the last step, $\mathcal{P}$ and $\mathcal{V}$ play $\mathtt{zkSVC}_{3,(n+m)}$, instead of $\mathtt{zkSVC}_{2,n}$, and thus $\mathcal{V}$ gets convinced that $\mathcal{P}$ knows weights for the following decompositions

$$\begin{cases} Z = \text{lin}(\mathbf{P}, \mathbf{V}) \\ F = \text{lin}(\mathbf{Q}) \\ E = \text{lin}(\mathbf{W}) \end{cases} . \tag{32}$$

Here we omit mentioning blinding with H, which is always implied performed before transmitting elements from prover to verifier.

An informal explanation of the $\mathtt{zkLin22sChoice}_{n,m}$ protocol is that considering the triplet of elements

$$(Z,\ rF,\ c_{n+t}E) \tag{33}$$

we prove with $\mathtt{zkSVC}_{3,(n+m)}$ that the first, second, and third elements of the triplet (33) are linear combinations with the same coefficients of $(n+m)$ elements of, respectively, the first, second, and third dimensions of the decoy set composed of the parts (29) and (30). We observe that thereby all the steps of the $\mathtt{zkLin2Choice}_n$ and $\mathtt{zkLin2Choice}_m$ protocols are actually performed for Z's 'projections' on $\mathbf{P}$ and on $\mathbf{V}$, respectively. That is, the following holds

$$Z = Z_P + Z_V,\ \text{ where } Z_P = \mathrm{lin}(\mathbf{P}),\ Z_V = \mathrm{lin}(\mathbf{V})\ . \tag{34}$$

Thus, we get to the conclusion that all the steps of $\mathtt{zkLin2Choice}_n$ in Figure 8 have been performed for

- Z_P and the first part of the decoy set comprising n triples (29). The actual index s remains hidden because the response r is randomized, as in the Lin2-Choice lemma protocol $\mathtt{zkLin2Choice}_n$.
- Z_V and the second part of the decoy set comprising m triples (30). The actual index t in this part is not hidden because the implied 'reply' c_{n+t} clearly reveals it. Nevertheless, this does not wreck the Lin2-Choice lemma argument, just makes it non-zero-knowledge by t.

Hence, by the Lin2-Choice lemma, verifier is convinced that the following holds for prover

$$\begin{cases} Z_P \sim P_s & \text{, where } s \text{ is secret} \\ Z_V \sim V_t & \text{, where } t \text{ is public} \end{cases}\qquad , \tag{35}$$

and therefore $Z = pP_s + vV_t$ for some p and v known to the prover.

7.1.2 MULTIPLE SIMMETRIC VECTOR COMMITMENTS

We need one more auxiliary zero-knowledge argument, it is shown in Figure 18,

$$\mathtt{zkMSVC}_{l,3,n}(\mathbf{P}, \mathbf{Q}, \mathbf{R}, H, \mathbf{Z}, \mathbf{F}, \mathbf{E};\ \mathfrak{a}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})\ ,$$

which proves the same as l simultaneously played instances of the $\mathtt{zkSVC}_{3,n}$ argument (Figure 5) prove. Namely, this is an argument for the following relation

$$\mathcal{R} = \left\{ \begin{array}{l} \mathbf{P} \in \mathbb{G}^{n*}, \mathbf{Q}, \mathbf{R} \in \mathbb{G}^n, H \in \mathbb{G}^*, \mathbf{Z}, \mathbf{F}, \mathbf{E} \in \mathbb{G}^l; \\ \mathfrak{a} \in \mathbb{F}_{\bar{p}}^{l\times n}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma} \in \mathbb{F}_{\bar{p}}^l \end{array} \ \middle|\ \begin{array}{l} \mathbf{Z} = \mathfrak{a}\cdot\mathbf{P} + \boldsymbol{\alpha}\cdot H\ \wedge \\ \mathbf{F} = \mathfrak{a}\cdot\mathbf{Q} + \boldsymbol{\beta}\cdot H\ \wedge \\ \mathbf{E} = \mathfrak{a}\cdot\mathbf{R} + \boldsymbol{\gamma}\cdot H \end{array} \right\}, \tag{36}$$

where all generators $\mathbf{P}, \mathbf{Q}, \mathbf{R}, H$ are orthogonal to each other. This relation is l instances of the relation (9) merged together. The other surrounding conditions for it are the same as for the relation (9).

We implement $\mathtt{zkMSVC}_{l,3,n}$ by merging l instances of $\mathtt{zkSVC}_{3,n}$ together using the shared random scalars δ_1 and δ_2. The following two vectors are built with these random scalars

$$\begin{aligned} \mathbf{X} &= \mathbf{P} + \delta_1\mathbf{Q} + \delta_2\mathbf{R} \\ \mathbf{Y} &= \mathbf{Z} + \delta_1\mathbf{F} + \delta_2\mathbf{E}\,. \end{aligned}$$

Then, instead of invoking $\mathtt{zkVC}_n(\mathbf{X}, H, Y_j\ ;\ \mathfrak{a}_{[j,:]}, \alpha_j + \delta_1\beta_j + \delta_2\gamma_j)$ for each $j \in [0\dots l-1]$, we invoke $\mathtt{zkMVC}_{l,n}$ (Figure 13) for $\mathbf{X}, \mathbf{Y}$. Thus, we get a proof for the relation (36) at the price of one $\mathtt{zkMVC}_{l,n}$ call and, therefore, at the price of one $\mathtt{zkVC}_n$ call.

7.1.3 LIN2-2CHOICE LEMMA

Now we can construct the protocol in Figure 19,

$$\mathtt{zkLin22Choice}_{l,n,m}(\mathbf{P}, \mathbf{Q}, \mathbf{V}, \mathbf{W}, H, \mathbf{Z}; \mathbf{s}, \mathbf{p}, \mathbf{v}, \boldsymbol{\alpha}),$$

and prove the Lin2-2Choice lemma which states that $\mathtt{zkLin22Choice}_{l,n,m}$ is an argument for the relation

$$\mathcal{R} = \left\{ \begin{array}{l} \mathbf{P}, \mathbf{Q} \in \mathbb{G}^{n*}, \mathbf{V}, \mathbf{W} \in \mathbb{G}^{m*}, H \in \mathbb{G}^*, \mathbf{Z} \in \mathbb{G}^l; \\ \mathbf{s} \in [0\dots n-1]^l, \mathbf{p}, \mathbf{v}, \boldsymbol{\alpha} \in \mathbb{F}_{\bar{p}}^l \end{array} \ \middle|\ \begin{array}{l} \forall k \in [0\dots l-1] : \\ Z_k = p_k P_{s_k} + v_k V_k + \alpha_k H \end{array} \right\}, \tag{37}$$

where the generators $\mathbf{P}, \mathbf{Q}, \mathbf{V}, \mathbf{W}, H$ are orthogonal to each other and $l \leqslant m$.

The relation (37) is essentially the relation (27) repeated for the first l elements of the decoy set's second part (30). Having such a correspondence between the relations (37) and (27), the $\texttt{zkLin22Choice}_{l,n,m}$ protocol is l instances of the protocol $\texttt{zkLin22sChoice}_{n,m}$ run in parallel, with the only one refinement which follows.

The refinement is that all the l instances of the $\texttt{zkLin22sChoice}_{n,m}$ protocol are played in sync and independently of each other (except for the common challenges, as for EFLRSL in Section 6.1.3) up to the last step, where l instances of $\texttt{zkSVC}_{3,n}$ are called. All these l calls of $\texttt{zkSVC}_{3,n}$, in turn, are replaced with one call to $\texttt{zkMSVC}_{l,3,n}$, which gives significant reduction in the transcript size.

7.2 FORMAL PRESENTATION

7.2.1 SIMPLIFIED LIN2-2CHOICE LEMMA

Theorem 10:
For $n, m \in \mathbb{N}^*$, for four vectors of nonzero elements $\mathbf{P}, \mathbf{Q} \in \mathbb{G}^{n*}$, $\mathbf{V}, \mathbf{W} \in \mathbb{G}^{m*}$, for a nonzero element $H \in \mathbb{G}^*$ such that $\mathrm{ort}(\mathbf{P} \cup \mathbf{Q} \cup \mathbf{V} \cup \mathbf{W} \cup \{H\})$ holds, for an element $Z \in \mathbb{G}$, the protocol $\texttt{zkLin22sChoice}_{n,m}$ in Figure 17 is a complete, sHVZK argument having cWEE for the relation (27) with unique witness.

Proof: Appendix K.
Overview: Section 7.1.1.

$\boxed{\texttt{zkLin22sChoice}_{n,m}(\mathbf{P}, \mathbf{Q}, \mathbf{V}, \mathbf{W}, H, Z, t; s, p, v, \alpha)}$

Relation $\mathcal{R} = \left\{ \begin{array}{l} \mathbf{P}, \mathbf{Q} \in \mathbb{G}^{n*}, \mathbf{V}, \mathbf{W} \in \mathbb{G}^{m*}, H \in \mathbb{G}^*, Z \in \mathbb{G}, t \in [0 \ldots m-1]; \\ s \in [0 \ldots n-1], p, v, \alpha \in \mathbb{F}_{\bar{p}} \end{array} \middle| \; Z = pP_s + vV_t + \alpha H \right\}$ // (27)

// $\mathbf{P}, \mathbf{Q}, \mathbf{V}, \mathbf{W}, H$ in $\mathcal{R}$ satisfy $\mathrm{ort}(\mathbf{P} \cup \mathbf{Q} \cup \mathbf{V} \cup \mathbf{W} \cup \{H\})$.

$\mathcal{P}$'s input : $(\mathbf{P}, \mathbf{Q}, \mathbf{V}, \mathbf{W}, H, Z, t;\ s, p, v, \alpha)$
$\mathcal{V}$'s input : $(\mathbf{P}, \mathbf{Q}, \mathbf{V}, \mathbf{W}, H, Z, t)$
$\mathcal{P}$'s output : none
$\mathcal{V}$'s output: *Accept* or *Reject*

$\boxed{\mathcal{P}}$: $q, \beta, \gamma \leftarrow\!\!\$\ \mathbb{F}^*_{\bar{p}}$ and assigns **if** $p = 0$ **then** $q = 0$ **endif**
$F = qQ_s + \beta H$
$E = vW_t + \gamma H$

$\boxed{\mathcal{P} \to \mathcal{V}}$: F, E

$\boxed{\mathcal{V}}$: $\mathbf{c} \leftarrow\!\!\$\ \mathbb{F}^{(n+m)*}_{\bar{p}}$

$\boxed{\mathcal{V} \to \mathcal{P}}$: $\mathbf{c}$

$\boxed{\mathcal{P}}$: takes scalars c_s, c_{n+t} at indices s and $n+t$ in $\mathbf{c}$, that is, lets $c_s \leftarrow \mathbf{c}_{[s]}$, $c_{n+t} \leftarrow \mathbf{c}_{[n+t]}$,
samples $r \leftarrow\!\!\$\ \mathbb{F}^*_{\bar{p}}$,
assigns **if** $p \neq 0$ **then** $r = c_s p/q$ **endif**
$\hat{\beta} = r\beta$
$\hat{\gamma} = c_{n+t}\gamma$,

and lets $\mathbf{a} = \left\{ \begin{array}{ll} a_s = p & \text{// that is, } p \text{ is at } s\text{'th position in } \mathbf{a} \\ a_{n+t} = v & \text{// thus, } \mathbf{a} \text{ contains at most two hot entries} \\ a_i = 0 \text{ for all } i \in [0 \ldots n+m-1], i \neq s \wedge i \neq (n+t) & \end{array} \right.$

$\boxed{\mathcal{P} \to \mathcal{V}}$: r

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: allocate $\hat{\mathbf{P}} \in \mathbb{G}^{(n+m)*}$, $\hat{\mathbf{Q}}, \hat{\mathbf{R}} \in \mathbb{G}^{(n+m)}$,
assign $\hat{\mathbf{P}}_{[:n]} = \mathbf{P}$, $\hat{\mathbf{P}}_{[n:]} = \mathbf{V}$
$\hat{\mathbf{Q}}_{[:n]} = \mathbf{c}_{[:n]} \circ \mathbf{Q}$, $\hat{\mathbf{Q}}_{[n:]} = \mathbf{0}^m$
$\hat{\mathbf{R}}_{[:n]} = \mathbf{0}^n$, $\hat{\mathbf{R}}_{[n:]} = \mathbf{c}_{[n:]} \circ \mathbf{W}$,
let $\hat{F} \leftarrow rF$
$\hat{E} \leftarrow \mathbf{c}_{[n+t]} E$,
and run $\texttt{zkSVC}_{3,(n+m)}(\hat{\mathbf{P}}, \hat{\mathbf{Q}}, \hat{\mathbf{R}}, H, Z, \hat{F}, \hat{E}; \mathbf{a}, \alpha, \hat{\beta}, \hat{\gamma})$

Figure 17: Simplified Lin2-2Choice lemma protocol, zero-knowledge argument for two-element choice relation

Similar to $\texttt{zkLin2Choice}_n$ in Figure 8, we consider (p, v, α) as witness for $\texttt{zkLin22sChoice}_{n,m}$ in Figure 17. The auxiliary index s is recoverable from the witness in a polynomial time.

7.2.2 MULTIPLE SIMMETRIC VECTOR COMMITMENTS

To advance from the one-out-of-many proof to the many-out-of-many one, in Figure 18 we define a helper protocol.

Theorem 11:
For $n, l \in \mathbb{N}^*$, for a vector of nonzero elements $\mathbf{P} \in \mathbb{G}^{n*}$, and for a pair of vectors of elements $\mathbf{Q}, \mathbf{R} \in \mathbb{G}^n$ such that $(\mathbf{Q} + \mathbf{R}) \in \mathbb{G}^{n*}$, for a nonzero element $H \in \mathbb{G}^*$ such that $\mathrm{ort}(\mathbf{P} \cup \mathrm{nz}(\mathbf{Q}) \cup \mathrm{nz}(\mathbf{R}) \cup \{H\})$ holds, for three vectors of elements $\mathbf{Z}, \mathbf{F}, \mathbf{E} \in \mathbb{G}^l$, the protocol $\texttt{zkMSVC}_{l,3,n}$ in Figure 18 is a complete, sHVZK argument having cWEE for the relation (36) with unique witness.

Proof: Appendix L.
Overview: Section 7.1.2.

$$\boxed{\texttt{zkMSVC}_{l,3,n}(\mathbf{P}, \mathbf{Q}, \mathbf{R}, H, \mathbf{Z}, \mathbf{F}, \mathbf{E};\ \mathfrak{a}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})}$$

$$\text{Relation } \mathcal{R} = \left\{ \begin{array}{l} \mathbf{P} \in \mathbb{G}^{n*}, \mathbf{Q}, \mathbf{R} \in \mathbb{G}^n, H \in \mathbb{G}^*, \mathbf{Z}, \mathbf{F}, \mathbf{E} \in \mathbb{G}^l; \\ \mathfrak{a} \in \mathbb{F}_{\bar{p}}^{l \times n}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma} \in \mathbb{F}_{\bar{p}}^l \end{array} \middle| \begin{array}{l} \mathbf{Z} = \mathfrak{a} \cdot \mathbf{P} + \boldsymbol{\alpha} \cdot H \ \wedge \\ \mathbf{F} = \mathfrak{a} \cdot \mathbf{Q} + \boldsymbol{\beta} \cdot H \ \wedge \\ \mathbf{E} = \mathfrak{a} \cdot \mathbf{R} + \boldsymbol{\gamma} \cdot H \end{array} \right\} \quad // \ (36)$$

// $\mathbf{P}, \mathbf{Q}, \mathbf{R}, H$ in $\mathcal{R}$ satisfy $\mathrm{ort}(\mathbf{P} \cup \mathrm{nz}(\mathbf{Q}) \cup \mathrm{nz}(\mathbf{R}) \cup \{H\})$ and $(\mathbf{Q} + \mathbf{R}) \in \mathbb{G}^{n*}$

$\mathcal{P}$'s input : $(\mathbf{P}, \mathbf{Q}, \mathbf{R}, H, \mathbf{Z}, \mathbf{F}, \mathbf{E};\ \mathfrak{a}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$
$\mathcal{V}$'s input : $(\mathbf{P}, \mathbf{Q}, \mathbf{R}, H, \mathbf{Z}, \mathbf{F}, \mathbf{E})$
$\mathcal{P}$'s output : none
$\mathcal{V}$'s output: *Accept* or *Reject*

$\boxed{\mathcal{V}}$: $\delta_1, \delta_2 \leftarrow\$ \ \mathbb{F}_{\bar{p}}^*$

$\boxed{\mathcal{V} \to \mathcal{P}}$: δ_1, δ_2

$\boxed{\mathcal{P}}$: computes $\quad \hat{\boldsymbol{\alpha}} = \boldsymbol{\alpha} + \delta_1 \boldsymbol{\beta} + \delta_2 \boldsymbol{\gamma}$

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: compute $\mathbf{X} = \mathbf{P} + \delta_1 \mathbf{Q} + \delta_2 \mathbf{R}$
$\mathbf{Y} = \mathbf{Z} + \delta_1 \mathbf{F} + \delta_2 \mathbf{E}$
and run $\texttt{zkMVC}_{l,n}(\mathbf{X}, H, \mathbf{Y};\ \mathfrak{a}, \hat{\boldsymbol{\alpha}})$

Figure 18: Zero-knowledge argument for multiple 3-vector commitments with shared weights

7.2.3 LIN2-2CHOICE LEMMA. MULTIPLE TWO-ELEMENT CHOICES

Theorem 12 (Lin2-2Choice lemma)**:**
For $n, m, l \in \mathbb{N}^*$ such that $l \leqslant m$, for four vectors of nonzero elements $\mathbf{P}, \mathbf{Q} \in \mathbb{G}^{n*}$, $\mathbf{V}, \mathbf{W} \in \mathbb{G}^{m*}$, for a nonzero element $H \in \mathbb{G}^*$ such that $\mathrm{ort}(\mathbf{P} \cup \mathbf{Q} \cup \mathbf{V} \cup \mathbf{W} \cup \{H\})$ holds, for a vector of elements $\mathbf{Z} \in \mathbb{G}^l$, the protocol $\texttt{zkLin22Choice}_{l,n,m}$ in Figure 19 is a complete, sHVZK argument having cWEE for the relation (37) with unique witness.

Proof: Appendix M.
Overview: Section 7.1.3.

Like for the protocol $\texttt{zkLin2Choice}_n$ in Figure 8, we consider $(\mathbf{p}, \mathbf{v}, \boldsymbol{\alpha})$ as witness for $\texttt{zkLin22Choice}_{l,n,m}$ in Figure 19. The auxiliary indices $\mathbf{s}$ are recoverable from the witness in a polynomial time.

$$\boxed{\texttt{zkLin22Choice}_{l,n,m}(\mathbf{P},\mathbf{Q},\mathbf{V},\mathbf{W},H,\mathbf{Z};\mathbf{s},\mathbf{p},\mathbf{v},\boldsymbol{\alpha})}$$

$$\text{Relation } \mathcal{R} = \left\{ \begin{array}{l} \mathbf{P},\mathbf{Q} \in \mathbb{G}^{n*}, \mathbf{V},\mathbf{W} \in \mathbb{G}^{m*}, H \in \mathbb{G}^{*}, \mathbf{Z} \in \mathbb{G}^{l}; \\ \mathbf{s} \in [0\ldots n-1]^{l}, \mathbf{p},\mathbf{v},\boldsymbol{\alpha} \in \mathbb{F}_{\bar{p}}^{l} \end{array} \middle| \begin{array}{l} \forall k \in [0\ldots l-1]: \\ Z_k = p_k P_{s_k} + v_k V_k + \alpha_k H \end{array} \right\} \quad // \; (37)$$

// $\mathbf{P},\mathbf{Q},\mathbf{V},\mathbf{W},H$ in $\mathcal{R}$ satisfy ort($\mathbf{P} \cup \mathbf{Q} \cup \mathbf{V} \cup \mathbf{W} \cup \{H\}$) .

$\mathcal{P}$'s input : $(\mathbf{P},\mathbf{Q},\mathbf{V},\mathbf{W},H,\mathbf{Z};\ \mathbf{s},\mathbf{p},\boldsymbol{\alpha})$

$\mathcal{V}$'s input : $(\mathbf{P},\mathbf{Q},\mathbf{V},\mathbf{W},H,\mathbf{Z})$

$\mathcal{P}$'s output : none

$\mathcal{V}$'s output: *Accept* or *Reject*

$\boxed{\mathcal{P}}$: $\mathbf{q},\boldsymbol{\beta},\boldsymbol{\gamma} \leftarrow\!\!\$\ \mathbb{F}_{\bar{p}}^{l*}$, allocates $\mathbf{F},\mathbf{E} \in \mathbb{G}^{l*}$,

initializes **foreach** $k \in [0\ldots l-1]$

$$\begin{array}{l} \text{let } (s_k, p_k, v_k, \beta_k, \gamma_k) \leftarrow (\mathbf{s}_{[k]}, \mathbf{p}_{[k]}, \mathbf{v}_{[k]}, \boldsymbol{\beta}_{[k]}, \boldsymbol{\gamma}_{[k]}) , \\ \quad \textbf{if } p_k = 0 \textbf{ then } \mathbf{q}_{[k]} = 0 \textbf{ endif} , \\ \text{let } q_k \leftarrow \mathbf{q}_{[k]} , \\ \quad \mathbf{F}_{[k]} = q_k Q_{s_k} + \beta_k H \quad \text{// } \mathbf{F} \text{ is filled in, note random } q_k\text{'s are nullified when } p_k = 0 \\ \quad \mathbf{E}_{[k]} = v_k W_k + \gamma_k H \quad \text{// } \mathbf{E} \text{ is filled in} \end{array}$$

endforeach

$\boxed{\mathcal{P} \to \mathcal{V}}$: $\mathbf{F},\mathbf{E}$

$\boxed{\mathcal{V}}$: $\mathbf{c} \leftarrow\!\!\$\ \mathbb{F}_{\bar{p}}^{(n+m)*}$

$\boxed{\mathcal{V} \to \mathcal{P}}$: $\mathbf{c}$

$\boxed{\mathcal{P}}$: allocates $\hat{\boldsymbol{\beta}},\hat{\boldsymbol{\gamma}} \in \mathbb{F}_{\bar{p}}^{l*}$, $\mathfrak{a} \in \mathbb{F}_{\bar{p}}^{l\times(n+m)}$, samples $\mathbf{r} \leftarrow\!\!\$\ \mathbb{F}_{\bar{p}}^{l*}$,

and initializes **foreach** $k \in [0\ldots l-1]$

$$\begin{array}{l} \text{let } c_{s_k} \leftarrow \mathbf{c}_{[s_k]} \\ \quad \textbf{if } p_k \neq 0 \textbf{ then } \mathbf{r}_{[k]} = c_{s_k} p_k / q_k \textbf{ endif} \quad \text{// } \mathbf{r} \text{ is filled in here} \end{array}$$

endforeach ,

continues initialization

$$\begin{array}{l} \hat{\boldsymbol{\beta}} = \mathbf{r} \circ \boldsymbol{\beta} \\ \hat{\boldsymbol{\gamma}} = \mathbf{c}_{[n:(n+l)]} \circ \boldsymbol{\gamma} , \end{array}$$

$$\text{lets } \mathfrak{a} = \{a_{k\in[0\ldots l-1], i\in[0\ldots n+m-1]}\} = \left\{ \begin{array}{ll} a_{k,s_k} = p_k & \text{// that is, } p_k \text{ is at } s_k\text{'th position in } k\text{'th row} \\ a_{k,n+k} = v_k & \text{// that is, } v_k \text{ is at } (n+k)\text{'th position in } k\text{'th row} \\ a_{k,i} = 0 \text{ if } i \neq s_k \wedge i \neq (n+k) & \text{// zeros for all the rest} \end{array} \right.$$

$\boxed{\mathcal{P} \to \mathcal{V}}$: $\mathbf{r}$

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: allocate $\hat{\mathbf{P}} \in \mathbb{G}^{(n+m)*}$, $\hat{\mathbf{Q}},\hat{\mathbf{R}} \in \mathbb{G}^{(n+m)}$,

$$\begin{array}{lll} \text{assign } & \hat{\mathbf{P}}_{[:n]} = \mathbf{P}, & \hat{\mathbf{P}}_{[n:]} = \mathbf{V} \\ & \hat{\mathbf{Q}}_{[:n]} = \mathbf{c}_{[:n]} \circ \mathbf{Q}, & \hat{\mathbf{Q}}_{[n:]} = \mathbf{0}^{m} \\ & \hat{\mathbf{R}}_{[:n]} = \mathbf{0}^{n}, & \hat{\mathbf{R}}_{[n:]} = \mathbf{c}_{[n:]} \circ \mathbf{W} , \end{array}$$

$$\begin{array}{l} \text{let } \hat{\mathbf{F}} \leftarrow \mathbf{r} \circ \mathbf{F} \\ \quad \hat{\mathbf{E}} \leftarrow \mathbf{c}_{[n:(n+l)]} \circ \mathbf{E} , \end{array}$$

and run $\texttt{zkMSVC}_{l,3,(n+m)}(\hat{\mathbf{P}},\hat{\mathbf{Q}},\hat{\mathbf{R}},H,\mathbf{Z},\hat{\mathbf{F}},\hat{\mathbf{E}};\ \mathfrak{a},\boldsymbol{\alpha},\hat{\boldsymbol{\beta}},\hat{\boldsymbol{\gamma}})$

Figure 19: Lin2-2Choice lemma protocol, zero-knowledge argument for multiple two-element choices relation

8 SIGNATURE EFLRSLWB WITH BALANCE PROOF

Now we are going to append a proof of the balance to the EFLRSL signature described in Section 6.2.3. For the honest case, we assume that each public key in the signature ring has an associated hidden amount in the form of Pedersen commitment [26]. When an honest prover signs, it knows the signing indices and thus knows those commitments associated with them. The sum of their openings, namely, the sum of the respective amounts, is to be equal to the total amount that is hidden in another commitment known beforehand to both of the prover and verifier. We will make the prover submit a zero-knowledge proof of this balance along with the signature.

For the dishonest case, we do not require the public keys and their associated hidden amounts in the ring to be in any specific form, all they can be adversarially generated. However, we will show that no prover can generate a valid signature without knowing signing private keys or without having the sum of hidden amount commitments associated with signing indices be equal to the total amount commitment, to the accuracy of blinding component.

8.1 ADDITIONAL DEFINITIONS

Let there be two additional predefined group generators B, D, and let the ring be composed of n pairs

$$\{(P_i, A_i)\}_{i=0}^{n-1}\ \text{, where}\ \ \mathbf{P} = \{P_i\}_{i=0}^{n-1}\ \wedge\ \mathbf{A} = \{A_i\}_{i=0}^{n-1}. \tag{38}$$

In the honest case we assume that the following two assertions hold, for each $i \in [0 \dots n-1]$, with the scalars p_i, b_i, d_i known to at least one player in the system

$$P_i = g_i G\,, \tag{39}$$

$$A_i = b_i B + d_i D\,. \tag{40}$$

In general, as usual, we assume that the case is dishonest, i.e., the equalities (40) and (39) may not hold and, moreover, some or all P_i's and A_i's in the ring can be adversarially chosen.

Nevertheless, we will use the following quite reasonable minimal assumption about the hidden amounts. Hereinafter we will assume that, for all A_i's in the ring, there already exist some validated proofs of the decomposition (40). These proofs can be submitted, e.g., along with other signatures that introduce these A_i's into system. In the case of blockchain, this means that validators must verify them along with the signatures.

This minimal assumption simply expresses the fact that no hidden amount enters the system in a free form. All of them are somehow examined to be at least in the form (40). This is unlike addresses whose form in the system is not examined when they are published. With this minimal assumption, in the ring, in the worst case, P_i's may have adversarially chosen g_i's or may have an unknown relation to G, whereas A_i's are guaranteed to be in the form (40) and may have only adversarially chosen b_i's and d_i's.

Besides, we are not going to make much use of this minimal assumption. Our plan is to construct a signature that will convince verifier that prover knows the signing indicies $\mathbf{s} = \{s_k\}_{k=0}^{l-1}$, the signing private keys $\mathbf{x} = \{x_k\}_{k=0}^{l-1}$ such that $\forall k : P_{s_k} = x_k G$, and also that $A^{\mathbf{sum}} = \sum_{k=0}^{l-1} A_{s_k}$ holds to the accuracy of D component. $A^{\mathbf{sum}}$ denotes the total hidden amount here. When the verifier validate such a signature, our minimal assumption will have its only use, namely, it will immediately imply the balance.

In Figure 20 we summarize the above definitions and assumptions about how the addresses and hidden amounts are represented in the system.

`Addresses and hidden amounts`

- Each public key P is accompanied by a hidden amount A in the system. Each ring has the form (38).
- Each hidden amount A in a ring is assumed having the decomposition (40) by the predefined generators B, D, i.e.,

  $$A = bB + dD\,,$$

  where b is the amount and d is the amount's blinding factor. That is, it is assumed that as soon as A is included in the ring, there already exists an available valid proof of the decomposition (40) for it in the system.

Figure 20: Addresses and hidden amounts seen to all parties

We have to update the common information available to all parties in Figures 1, 9 with an extended set of predefined orthogonal generators, and to amend $\mathcal{H}_{\mathbf{point}}$ again to respect orthogonality of the additional generators, as shown in Figure 21.

`Updated common information`

- A couple of generators $B, D \in \mathbb{G}^*$ and the enlarged vector $\mathbf{G} = \{G_0, G_1, G_2, \dots, G_{2\bar{n}-1}\} \in \mathbb{G}^{2\bar{n}*}$ such that, for any set $\mathbf{H}$ of $\mathcal{H}_{\mathbf{point}}$ images on different pre-images, it holds $\operatorname{ort}(\mathbf{H} \cup \{G, B, D\} \cup \mathbf{G})$.
- $\mathcal{H}_{\mathbf{point}} : \{0,1\}^{\star} \to \mathbb{G}^*$ is updated in such a way, so that the above $\operatorname{ort}(\mathbf{H} \cup \{G, B, D\} \cup \mathbf{G})$ holds.

Figure 21: Updated common information available to each party

8.2 OVERVIEW

Efficient linkable threshold ring signature EFLRSLWB (Efficient linkable ring signature for l actual signers with balance proof) is shown in Figure 22. Here is an informal introduction to how it works.

Having a ring of the form (38), $\mathcal{P}$ publishes l key images which correspond to the actually signing indices $\mathbf{s} \in [0 \ldots n-1]^l$

$$\mathbf{I} = \{I_k\}_{k=0}^{l-1} = \{x_k^{-1} \mathcal{H}_{\mathbf{point}}(P_{s_k})\}_{k=0}^{l-1}. \tag{41}$$

Also, it publishes an element $A^{\mathbf{sum}}$ and declares that, to the accuracy of a summand which is proportional to the hidden amount blinding generator D, the following holds

$$A^{\mathbf{sum}} = \sum_{k=0}^{l-1} A_{s_k} \, . \tag{42}$$

Next, $\mathcal{P}$ and $\mathcal{V}$ play the following game. They choose an orthogonal blinding generator H as an $\mathcal{H}_{\mathbf{point}}$ image of everything they have in common, and $\mathcal{P}$ publishes vector $\mathbf{A^{tmp}}$ of l hidden amounts, which correspond to the actual signing keys and are additionally blinded with H, i.e.,

$$\mathbf{A^{tmp}} = \{A_{s_k} + \mu_k H\}_{k=0}^{l-1} , \text{ where } \mu_k \hookleftarrow\$ \ \mathbb{F}_{\bar{p}}^* \, . \tag{43}$$

Then, $\mathcal{P}$ publishes a set of l what we call 'pseudo key images' $\mathbf{J}$, which are constructed as follows

$$\mathbf{J} = \{x_k^{-1} \mathcal{H}_{\mathbf{point}}(H, A_k^{\mathbf{tmp}}) + \upsilon_k H\}_{k=0}^{l-1}, \text{ where } \upsilon_k \hookleftarrow\$ \ \mathbb{F}_{\bar{p}}^* \, . \tag{44}$$

The term 'pseudo key image' comes from the fact that each J_k is structurally similar to I_k, except for that I_k takes $\mathcal{H}_{\mathbf{point}}$ of P_{s_k}, whereas J_k takes $\mathcal{H}_{\mathbf{point}}$ of $(H, \mathbf{A}_k^{\mathbf{tmp}})$ and is additionally blinded. Apparently, J_k cannot be used in the role of the real key image I_k for linking actual signers, as J_k is not unique due to the blinding. Note, that all I_k's are published before H is generated, so that they remain orthogonal to H even in the dishonest case.

In addition to this, $\mathcal{P}$ and $\mathcal{V}$ generate one more orthogonal generator, K, as an $\mathcal{H}_{\mathbf{point}}$ image of everything they have in common after $\mathbf{J}$ is published.

Now, $\mathcal{P}$ and $\mathcal{V}$ define the following three vectors using random weights ζ, ω, χ

$$\mathbf{X} = \mathbf{P} - \{K\}^n + \zeta \, \{\mathcal{H}_{\mathbf{point}}(P_i)\}_{i=0}^{n-1} - \omega \mathbf{A} \, , \tag{45}$$

$$\mathbf{V} = \{K\}^l + \omega \mathbf{A^{tmp}} + \chi \, \{\mathcal{H}_{\mathbf{point}}(H, A_k^{\mathbf{tmp}})\}_{k=0}^{l-1} \, , \tag{46}$$

$$\mathbf{Z} = \{G\}^l + \zeta \mathbf{I} + \chi \mathbf{J} \, , \tag{47}$$

and make a call to the Lin2-2Choice lemma protocol for them, as follows

$$\texttt{zkLin22Choice}_{l,n,l}(\mathbf{X}, \mathbf{Q}, \mathbf{V}, \mathbf{W}, H, \mathbf{Z};\ \ \mathbf{s}, \mathbf{x}^{-1}, \mathbf{x}^{-1},\ \boldsymbol{\alpha}_H \,) \, , \tag{48}$$

where $\mathbf{Q}, \mathbf{W}$ are auxiliary orthogonal generators prepared in advance. All elements in $\mathbf{Q}, \mathbf{W}$ are also orthogonal to the elements in $\mathbf{X}$ (45) and in $\mathbf{V}$ (46), since $\mathcal{H}_{\mathbf{point}}$ is defined in such a way that all its images are orthogonal to the predefined $\mathbf{Q}, \mathbf{W}$. The vector $\boldsymbol{\alpha}_H$ comprises the summary weights accumulated by the corresponding H components within the protocol.

When the call (48) successfully completes, by Theorem 12 (Lin2-2Choice lemma) $\mathcal{V}$ is convinced that, for each k, $\mathcal{P}$ knows scalar pair (p_k, v_k) such that, to the accuracy of H component, it holds

$$Z_k = \ p_k X_{s_k} + v_k V_k \ . \tag{49}$$

By inserting (45), (46), (47) into (49), $\mathcal{V}$ obtains

$$G + \zeta I_k + \chi J_k = p_k(\, P_{s_k} - K + \zeta \mathcal{H}_{\mathbf{point}}(P_{s_k}) - \omega A_{s_k} \,) + v_k(\, K + \omega A_k^{\mathbf{tmp}} + \chi \mathcal{H}_{\mathbf{point}}(H, A_k^{\mathbf{tmp}}) \,) \ , \tag{50}$$

which immediately yields $p_k = v_k$, as otherwise the $\mathcal{H}_{\mathbf{point}}$ image K gets decomposed by the components of its pre-image. By reducing (50), $\mathcal{V}$ gets

$$G + \zeta I_k + \chi J_k = p_k(\, P_{s_k} + \zeta \mathcal{H}_{\mathbf{point}}(P_{s_k}) + \chi \mathcal{H}_{\mathbf{point}}(H, A_k^{\mathbf{tmp}}) \,) + p_k(\, \omega A_k^{\mathbf{tmp}} - \omega A_{s_k} \,) \ . \tag{51}$$

Since $\mathcal{H}_{\mathbf{point}}(H, A_k^{\mathbf{tmp}})$ is orthogonal to everything else in the right-hand side of (51) and since at least P_{s_k} in it is nonzero, by Theorem 3 $\mathcal{V}$ gets convinced that, for each k, the following hold for some known to $\mathcal{P}$ scalar p_k, to the accuracy of the blinding H component,

$$\begin{cases} G \ = p_k P_{s_k} & \text{(52a)} \\ I_k \ = p_k \mathcal{H}_{\mathbf{point}}(P_{s_k}) & \text{(52b)} \\ J_k \ = p_k \mathcal{H}_{\mathbf{point}}(H, A_k^{\mathbf{tmp}}) & \text{(52c)} \\ A_{s_k} = A_k^{\mathbf{tmp}} \ . & \text{(52d)} \end{cases}$$

The equalities (52a), (52b) are strict, as all elements in them are included into the pre-image of H. Thus, they convince $\mathcal{V}$ that the signing is correct and the linking tag is properly calculated. At the same time, (52d) convinces $\mathcal{V}$ that $A_k^{\mathbf{tmp}}$ is the hidden amount corresponding to the signing key, to the accuracy of H.

Keeping in mind that the equality (52d) holds for each of l actually signing keys in $\mathbf{s}$, after the call to

$$\texttt{zk2ElemComm}(D,\, H,\, A^{\mathbf{sum}} - \sum_{k=0}^{l-1} A_k^{\mathbf{tmp}}\ ;\ \ldots) \tag{53}$$

$\mathcal{V}$ is convinced that $A^{\mathbf{sum}}$ is the sum of all the hidden amounts $\{A_{s_k}\}_{k=0}^{l-1}$ corresponding to the signing keys, to the accuracy of a linear by H and D component. Moreover, as both of $A^{\mathbf{sum}}$ and $\mathbf{A} \supseteq \{A_{s_k}\}_{k=0}^{l-1}$ are in the pre-image of H, the call (53) convinces $\mathcal{V}$ of the stronger assertion, namely, that $A^{\mathbf{sum}}$ is a sum of $\{A_{s_k}\}_{k=0}^{l-1}$ to the accuracy of D component only.

Thus, $\mathcal{V}$ is convinced that $\mathcal{P}$ knows the actually signing private keys, the linking tags are properly calculated, and also that, to the accuracy of D component, the equality (42) holds. This is all $\mathcal{V}$ gets from the signature.

8.3 FORMAL PRESENTATION

Theorem 13:
For $n, l \in \mathbb{N}^*$ such that $l \leqslant n$, for a vector of nonzero elements $\mathbf{P} \in \mathbb{G}^{n*}$ together with a vector of elements $\mathbf{A} \in \mathbb{G}^n$ which are considered a ring of (public key, hidden amount) pairs, for an element $A^{\mathbf{sum}}$, for a nonzero element D which is considered as a blinding generator for hidden amounts, the protocol in Figure 22 is a linkable threshold ring signature with the following properties

1. *perfect correctness,*
2. *existential unforgeability against adaptive chosen message / public key attackers,*
3. *unforgeability w.r.t. insider corruption,*
4. *anonymity,*
5. *anonymity w.r.t. chosen public key attackers,*
6. *linkability,*
7. *non-frameability,*
8. *non-frameability w.r.t. chosen public key attackers,*
9. *it is a proof of that $A^{\mathbf{sum}}$ is a sum of A's of the actual signing keys, to the accuracy of the blinding component proportional to D.*

Proof: Appendix O.
Overview: Section 8.2.

$\boxed{\texttt{EFLRSLWB.SignAndVerify}_{l,n}(\mathsf{M}, \mathbf{P}, \mathbf{A}, A^{\mathbf{sum}}, D; \mathbf{s}, \mathbf{x}, d^{\Delta\mathbf{sum}})}$

$\mathcal{P}$'s input : $(\mathsf{M} \in \{0,1\}^\star, \mathbf{P} \in \mathbb{G}^{n*}, \mathbf{A} \in \mathbb{G}^n, A^{\mathbf{sum}} \in \mathbb{G}, D \in \mathbb{G}^*;\ \mathbf{s} \in [0 \ldots n-1]^l, \mathbf{x} \in \mathbb{F}_{\bar{\mathsf{p}}}^{l*}, d^{\Delta\mathbf{sum}} \in \mathbb{F}_{\bar{\mathsf{p}}})$

$\mathcal{V}$'s input : $(\mathsf{M} \in \{0,1\}^\star, \mathbf{P} \in \mathbb{G}^{n*}, \mathbf{A} \in \mathbb{G}^n, A^{\mathbf{sum}} \in \mathbb{G}, D \in \mathbb{G}^*)$

$\mathcal{P}$'s output : *Signature* // signature is a list of all $\mathcal{P} \to \mathcal{V}$ messages from this and nested protocols

$\mathcal{V}$'s output: *Accept* or *Reject*

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: **assert** all elements in $\mathbf{P}$ are nonzero and different

let $\mathbf{U} \leftarrow \{\mathcal{H}_{\mathbf{point}}(\mathbf{P}_{[i]})\}_{i=0}^{n-1}$

$\boxed{\mathcal{P}}$: allocates $\mathbf{I} \in \mathbb{G}^{l*}$, $\mathbf{p} \in \mathbb{F}_{\bar{\mathsf{p}}}^{l*}$,

initializes **foreach** $k \in [0 \ldots l-1]$

assert $\mathbf{x}_{[k]} \neq 0$

$\mathbf{p}_{[k]} = \mathbf{x}_{[k]}^{-1}$

lets $(s_k, p_k) \leftarrow (\mathbf{s}_{[k]}, \mathbf{p}_{[k]})$,

$\mathbf{I}_{[k]} = p_k \mathbf{U}_{[s_k]}$ // vector $\mathbf{I}$ is filled in here

endforeach

$\boxed{\mathcal{P} \to \mathcal{V}}$: $\mathbf{I}$

$\boxed{\mathcal{V}}$: **assert** all elements in $\mathbf{I}$ are nonzero and different // $\mathcal{V}$ makes sure there is no zero I and no signer signing twice

$\epsilon \hookleftarrow\$ \mathbb{F}_{\bar{\mathsf{p}}}^*$

$\boxed{\mathcal{V} \to \mathcal{P}}$: ϵ

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: let $H \leftarrow \mathcal{H}_{\mathbf{point}}(\epsilon)$ // thus, H is orthogonal to all known so far elements, i.e., $\mathrm{ort}(H, G, \mathbf{P}, \mathbf{A}, \mathbf{U}, \mathbf{I}, A^{\mathbf{sum}}, D)$

$\boxed{\mathcal{P}}$: $\boldsymbol{\mu}, \boldsymbol{\upsilon} \hookleftarrow\$ \mathbb{F}_{\bar{\mathsf{p}}}^{l*}$, allocates $\mathbf{A}^{\mathbf{tmp}} \in \mathbb{G}^{l*}$, $\boldsymbol{\alpha} \in \mathbb{F}_{\bar{\mathsf{p}}}^{l*}$,

initializes **foreach** $k \in [0 \ldots l-1]$

lets $\mu_k \leftarrow \boldsymbol{\mu}_{[k]}$,

$\mathbf{A}^{\mathbf{tmp}}_{[k]} = \mathbf{A}_{[s_k]} + \mu_k H$ // $\mathbf{A}^{\mathbf{tmp}}$ is filled in, amounts get double blinded (with D and with H)

$\boldsymbol{\alpha}_{[k]} = p_k\, \mu_k$ // $\boldsymbol{\alpha}$ is initialized here, it contains reduced $\mathbf{A}^{\mathbf{tmp}}$'s second blinding factors

endforeach

$\boxed{\mathcal{P} \to \mathcal{V}}$: $\mathbf{A}^{\mathbf{tmp}}$

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: let $\hat{\mathbf{U}} \leftarrow \{\mathcal{H}_{\mathbf{point}}(H, \mathbf{A}^{\mathbf{tmp}}_{[k]})\}_{k=0}^{l-1}$

$\boxed{\mathcal{P}}$: lets $\mathbf{J} \leftarrow \{p_k \hat{\mathbf{U}}_{[k]} + \upsilon_k H\}_{k=0}^{l-1}$ // vector $\mathbf{J}$ is initialized here, it contains 'pseudo key images' built using $\hat{\mathbf{U}}$

$\boxed{\mathcal{P} \to \mathcal{V}}$: $\mathbf{J}$

$\boxed{\mathcal{V}}$: **assert** all elements in $\mathbf{A}^{\mathbf{tmp}}$, $\mathbf{J}$ are nonzero and different // $\mathcal{V}$ makes sure $\hat{\mathbf{U}}$ is orthogonal and there is no zero J

$\hat{\epsilon}, \zeta, \omega, \chi \hookleftarrow\$ \mathbb{F}_{\bar{\mathsf{p}}}^*$

$\boxed{\mathcal{V} \to \mathcal{P}}$: $\hat{\epsilon}, \zeta, \omega, \chi$

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: let $K \leftarrow \mathcal{H}_{\mathbf{point}}(\hat{\epsilon})$ // thus, $\mathrm{ort}(K, H, G, \mathbf{P}, \mathbf{A}, \mathbf{U}, \mathbf{I}, A^{\mathbf{sum}}, \mathbf{A}^{\mathbf{tmp}}, \hat{\mathbf{U}}, \mathbf{J})$ holds

allocate $\mathbf{X} \in \mathbb{G}^{n*}$, $\mathbf{V}, \mathbf{Z} \in \mathbb{G}^{l*}$, $S \in \mathbb{G}$,

assign $\mathbf{X} = \mathbf{P} - \{K\}^n + \zeta\mathbf{U} - \omega\mathbf{A}$, $\mathbf{V} = \{K\}^l + \omega\mathbf{A}^{\mathbf{tmp}} + \chi\hat{\mathbf{U}}$,

$\mathbf{Z} = \{G\}^l + \zeta\mathbf{I} + \chi\mathbf{J}$

assign $S = A^{\mathbf{sum}} - \sum_{k=0}^{l-1} \mathbf{A}^{\mathbf{tmp}}_{[k]}$

run $\texttt{zk2ElemComm}(D, H, S;\ d^{\Delta\mathbf{sum}}, -\sum_{k=0}^{l-1} \mu_k)$

run $\texttt{zkLin22Choice}_{l,n,l}(\mathbf{X}, \mathbf{G}_{[:n]}, \mathbf{V}, \mathbf{G}_{[n:(n+l)]}, H, \mathbf{Z};\ \mathbf{s}, \mathbf{p}, \mathbf{p},\ -\omega\boldsymbol{\alpha} + \chi\boldsymbol{\upsilon})$

Figure 22: EFLRSLWB signing and verification

8.4 SIZE AND COMPLEXITY

To verify the EFLRSLWB signature, $\mathcal{V}$ needs only to check the equalities (*) and (**) in Figure 23. By combining the equalities (*) and (**) with random weights and then using the multi-exponetiation technique, $\mathcal{V}$ performs the verifiacation in the time shown in Table 5, where signature size is also shown.

$\boxed{\texttt{SignAndVerify}_{l,n,u} \hookrightarrow \texttt{zkLin22Choice}_{l,n,l} \hookrightarrow \texttt{zkMSVC}_{l,3,(n+l)} \hookrightarrow \texttt{zkMVC}_{l,(n+l)} \hookrightarrow \texttt{zkVC}_{(n+l)} \hookrightarrow \texttt{zk2ElemComm}}$

// Function $\texttt{bitAtPos}(i,j)$ returns j-th bit of binary representation of i

$$c\left(\sum_{k=0}^{l-1}\xi_k(G+\zeta I_k+\chi J_k+\delta_1 r_k F_k+\delta_2 c_{(n+k)}E_k)+\sum_{j=0}^{\log_2(n+l)-1}(e_j^2 L_j+e_j^{-2}R_j)\right)+\eta H-T+$$
$$+\ \tau\left(\sum_{i=0}^{n-1}\left(\prod_{j=0}^{\log_2(n+l)-1}e_j^{2\cdot\texttt{bitAtPos}(i,j)-1}\right)(P_i+\zeta U_i-\omega A_i+K+\delta_1 c_i G_i)+\right. \tag{*}$$
$$\left.+\sum_{i=n}^{n+l-1}\left(\prod_{j=0}^{\log_2(n+l)-1}e_j^{2\cdot\texttt{bitAtPos}(i,j)-1}\right)(\omega A^{\mathbf{tmp}}_{(i-n)}+\chi\hat{U}_{(i-n)}-K+\delta_2 c_i G_i)\right)=0$$

and

$$\hat{\tau}D+\hat{\eta}H+\hat{c}S-\hat{T}=0 \tag{**}$$

Figure 23: EFLRSLWB unfolded equality, verifier checks it

Table 5: **EFLRSLWB** signature size and verification complexity

	Size	Verification complexity
EFLRSLWB	$2\lceil\log_2(n+l)\rceil+6l+6$	$\boldsymbol{mexp}(\,4n+2\log_2(n+l)+7l+7\,)+(n+l+2)\mathbf{H_{pt}}$

8.5 IMMEDIATE IMPLICATION

Having verified an instance of EFLRSLWB, $\mathcal{V}$ proceeds from the system properties in Figure 20 as follows. Since a proof of the decomposition (40) exists for each element in $\mathbf{A}$, having checked that all of these proofs are already verified in the system, $\mathcal{V}$ makes sure that $\mathbf{A}$ contains some hidden amounts, and not anything else. Namely, $\mathcal{V}$ gets convinced that it holds

$$\{A_{s_k}\}_{k=0}^{l-1}=\{b_{s_k}B+d_{s_k}D\}_{k=0}^{l-1}\subseteq\mathbf{A}\,,\ \text{where all } b_{s_k}\text{'s and } d_{s_k}\text{'s are known to someones in the system.} \tag{54}$$

From the decompositions (54) and from the proved equality (42), which follows from successful signature verification, $\mathcal{V}$ gets convinced that

$$A^{\mathbf{sum}}=b^{\mathbf{sum}}B+d^{\mathbf{sum}}D\,,\quad \text{where } b^{\mathbf{sum}} \text{ and } d^{\mathbf{sum}} \text{ can be reconstructed in the system.} \tag{55}$$

Finally, from (55), (54), (42) $\mathcal{V}$ gets convinced that

$$b^{\mathbf{sum}}=\sum_{k=0}^{l-1}b_{s_k}\,. \tag{56}$$

Thus, by verifying an instance of EFLRSLWB and by making sure that the corresponding proofs of the form (40) have already been checked for all the hidden amounts in the signature ring, $\mathcal{V}$ gets convinced that $\mathcal{P}$ knows signing private keys and, also, that the sum of the corresponding hidden amounts is balanced out by the given hidden amount $A^{\mathbf{sum}}$, to the accuracy of blinding with D.

9 SIGNATURE MULTRATUG

The signature EFLRSLWB has key image $\mathcal{H}_{\mathbf{point}}(P)/x$ with private key x in the denominator. In some applications it is desirable to have key image in a linear form by private key. This form, namely, the form

$x\mathcal{H}_{\mathbf{point}}(P)$, is used in the LSAG [23], CLSAG [13], CryptoNote [31] schemes. Consequently, the multiparty signing operations can be easily implemented for them.

Now we will move x from the denominator to the numerator in the EFLRSLWB's key image. Thus we will obtain a version of the EFLRSLWB signature with key image $x\mathcal{H}_{\mathbf{point}}(P)$, called EFLRSLWBLI (Efficient linkable ring signature with balance proof and linear key image) and aliased as Multratug.

Our idea of this x's movement is quite simple and does not require any new steps in the protocol, just only the few modifications to it which are outlined below. Although, to prove that this movement of x is correct, for the first, we will have to generalize Theorem 3 about three-element tuples to element tuples of greater length.

9.1 OVERVIEW

9.1.1 RANDOM WEIGHTING FOR T-S-TUPLES

Suppose, we have two tuples $\mathbf{T}, \mathbf{D}$ of $(t + s + 1)$ elements each, we call them t-s-tuples, such that

$$\mathbf{T} = (P,\ Q_0,\ Q_1, \ldots, Q_{t-1},\ S_0,\ S_1, \ldots, S_{s-1})\ , \tag{57}$$

$$\mathbf{D} = (Z,\ F_0,\ F_1,\ \ldots, F_{t-1},\ 0,\ \ 0,\ \ldots)\ , \tag{58}$$

where $P \in \mathbb{G}^*$, $\mathbf{Q} \in \mathbb{G}^t$, $\mathbf{S} \in \mathbb{G}^s$, $Z \in \mathbb{G}$, $\mathbf{F} \in \mathbb{G}^t$, for some $t > 0$, $s \geqslant 0$. The structure of these tuples is as follows. The element Z corresponds to the element P, the elements in $\mathbf{F}$ correspond to the elements with the same indices in $\mathbf{Q}$, and s zeros correspond to the elements in $\mathbf{S}$.

Now, we sample a random scalar vector $\boldsymbol{\xi}$ of length $(t + s + 1)$ and build the inner products of our tuples with this scalar vector $\boldsymbol{\xi}$. Namely, we build X, Y such that

$$X = \langle \boldsymbol{\xi}, \mathbf{T} \rangle = P + \xi_1 Q_0 + \xi_2 Q_1 + \cdots + \xi_{t+1} S_0 + \xi_{t+2} S_1 + \ldots\ , \tag{59}$$

$$Y = \langle \boldsymbol{\xi}, \mathbf{D} \rangle = Z + \ \xi_1 F_0 + \ \xi_2 F_1 + \ldots\ , \tag{60}$$

$$\text{where } \boldsymbol{\xi} = [1, \delta_0, \delta_1, \ldots, \delta_{t-1},\ \sigma_0, \sigma_1, \ldots, \sigma_{s-1}]\ . \tag{61}$$

Without limiting generality, we let the first element of the random vector $\boldsymbol{\xi}$ be equal to 1.

In addition to the above, suppose we have a complete, sHVZK, and having cWEE argument that convinces verifier of $Y \sim X$ to the accuracy of H component. Here H performs the role of a blinding generator, it is chosen in such a way as to be orthogonal to all the elements in $\mathbf{T}$, except for maybe those in its part $\mathbf{S}$. The question is what we can say about $\mathbf{T}$ and $\mathbf{D}$ under these conditions.

Theorem 14 answers this question so that as long as $\mathbf{Q}$ contains at least one nonzero element and P is orthogonal to $\mathbf{T} \setminus \{P\}$, there necessarily exists an unique factor a known to prover that connects all the corresponding elements from $\mathbf{T}$ and $\mathbf{D}$. The following relation (62), protocol $\mathtt{zkTElemRW}_{t,s}(P, \mathbf{Q}, \mathbf{S}, H, Z, \mathbf{F};\ a, \alpha, \boldsymbol{\beta}, \boldsymbol{\gamma})$ in Figure 24, and Theorem 14, formalize the game and sufficient conditions for the existence of such an unique factor.

$$\left\{ \begin{array}{l} P \in \mathbb{G}^*,\ \mathbf{Q} \in \mathbb{G}^t, \mathbf{S} \in \mathbb{G}^s, H \in \mathbb{G}^*, Z \in \mathbb{G},\ \mathbf{F} \in \mathbb{G}^t; \\ a, \alpha \in \mathbb{F}_{\bar{\mathrm{p}}},\ \boldsymbol{\beta} \in \mathbb{F}_{\bar{\mathrm{p}}}^t, \boldsymbol{\gamma} \in \mathbb{F}_{\bar{\mathrm{p}}}^s \end{array} \middle| \begin{array}{l} Z\ = aP + \alpha H\ \wedge \\ \mathbf{F}\ = a\mathbf{Q} + \boldsymbol{\beta} H\ \wedge \\ \{0\}^s\ = a\mathbf{S} + \boldsymbol{\gamma} H \end{array} \right\} \tag{62}$$

9.1.2 MULTRATUG: MOVING X TO THE NUMERATOR

Our idea of this x's movement is about building $\mathbf{X}, \mathbf{V}$, and $\mathbf{Z}$ in Figure 22 a bit differently, as follows. So, instead of the key image vector $\mathbf{I} = \{\, {x_k}^{-1} U_{s_k}\, \}_{k=0}^{l-1}$ in Figure 22, $\mathcal{P}$ builds a vector of the linear key images $\hat{\mathbf{I}}$ as

$$\hat{\mathbf{I}} = \{\, x_k U_{s_k}\, \}_{k=0}^{l-1}\,. \tag{63}$$

Then, $\mathcal{P}$ builds a blinded copy of the corresponding subset of $\mathbf{U}$ as

$$\mathbf{U^{tmp}} = \{\, U_{s_k}\, \}_{k=0}^{l-1} + \hat{\boldsymbol{\mu}} H, \ \ \text{where } \hat{\boldsymbol{\mu}} \hookleftarrow\!\$\ \mathbb{F}_{\bar{\mathrm{p}}}^{l*}, \tag{64}$$

and sends it to $\mathcal{V}$ together with $\mathbf{A^{tmp}}$. The vector $\mathbf{U^{tmp}}$ (along with $\mathbf{A^{tmp}}$) gets into the pre-images of all the hashes that are generated in the protocol from this moment on.

Finally, using the vectors $\hat{\mathbf{I}}, \mathbf{U^{tmp}}$, and an additional random scalar θ, both of $\mathcal{P}$ and $\mathcal{V}$ build $\mathbf{X}, \mathbf{V}, \mathbf{Z}$ as

$$\mathbf{X} = \mathbf{P} - \{K\}^n + \zeta \mathbf{U} - \omega \mathbf{A}\,, \tag{65}$$

$$\mathbf{V} = \{K\}^l + \omega \mathbf{A^{tmp}} - \zeta \mathbf{U^{tmp}} + \theta \hat{\mathbf{I}} + \chi \hat{\mathbf{U}}\,, \tag{66}$$

$$\mathbf{Z} = \{G\}^l + \theta \mathbf{U^{tmp}} + \chi \mathbf{J}\,. \tag{67}$$

Then, $\mathcal{P}$ and $\mathcal{V}$ proceed with executing the protocol to the completion. Of course, $\mathcal{P}$ adjusts the total blinding factor at the private input of $\texttt{zkLin22Choice}_{l,n,l}$ with respect to the new $\hat{\boldsymbol{\mu}}$ sampled in (64).

Since $\mathbf{X}, \mathbf{V}, \mathbf{Z}$ are now defined by (65), (66), (67) instead of (45), (46), (47), by Theorem 12 (Lin2-2Choice lemma) $\mathcal{V}$ obtains l following equalies instead of l equalities (51), for each $k \in [0 \dots l-1]$, to the accuracy of H component

$$G + \theta U_k^{\mathbf{tmp}} + \chi J_k = p_k(P_{s_k} + \theta \hat{I}_k + \chi \hat{U}_k) + p_k(\omega A_k^{\mathbf{tmp}} - \omega A_{s_k} + \zeta U_{s_k} - \zeta U_k^{\mathbf{tmp}}), \quad \text{where } p_k = x_k^{-1}. \tag{68}$$

By Theorem 14, from (68) $\mathcal{V}$ gets convinced that the following system of equalities holds, for each k, to the accuracy of H component, this is explained in detail in Appendix S

$$\begin{cases} G = p_k P_{s_k} & \text{(69a)} \\ U_{s_k} = U_k^{\mathbf{tmp}} & \text{(69b)} \\ U_{s_k} = p_k \hat{I}_k & \text{(69c)} \\ J_k = p_k \hat{U}_k & \text{(69d)} \\ A_{s_k} = A_k^{\mathbf{tmp}}\ . & \text{(69e)} \end{cases}$$

From (69a) and (69c), which are strict (have zero H component, as H is a hash image of all their elements), $\mathcal{V}$ gets convinced that the signing is correct and that the linear linking tags are valid, respectively. The balance proof and all the other points of the Theorem 13 proof remain the same as for EFLRSLWB with the former linking tag. Thus, the transition to the linear linking tag is performed, with all the EFLRSLWB properties moved unaffected to EFLRSLWBI in Figure 25, aliased as Multratug.

9.2 FORMAL PRESENTATION

9.2.1 RANDOM WEIGHTING FOR T-S-TUPLES

Theorem 14 (Random weighting for t-s-tuples)**:**
For $t \in \mathbb{N}^*$, $s \in \mathbb{N}$, for two nonzero elements $P, H \in \mathbb{G}^*$, for two element vectors $\mathbf{Q} \in \mathbb{G}^t$, $\mathbf{S} \in \mathbb{G}^s$ such that $\mathrm{nz}(\mathbf{Q}) \neq \varnothing \ \wedge\ P \mathrel{!=} \mathrm{lin}(\mathrm{nz}(\mathbf{Q}) \cup \mathrm{nz}(\mathbf{S}) \cup \{H\}) \ \wedge\ H \mathrel{!=} \mathrm{lin}(\mathrm{nz}(\mathbf{Q}) \cup \{P\})$ *holds, the protocol* $\texttt{zkTElemRW}_{t,s}$ *in Figure 24 is a complete, sHVZK argument having cWEE for the relation (62) with unique witness.*

Proof: is in Appendix P.
Overview: Section 9.1.1.

$\boxed{\texttt{zkTElemRW}_{t,s}(P, \mathbf{Q}, \mathbf{S}, H, Z, \mathbf{F};\ a, \alpha, \boldsymbol{\beta}, \boldsymbol{\gamma})}$

$$\text{Relation } \mathcal{R} = \left\{ \begin{array}{l|l} P \in \mathbb{G}^*,\ \mathbf{Q} \in \mathbb{G}^t, \mathbf{S} \in \mathbb{G}^s, H \in \mathbb{G}^*, Z \in \mathbb{G},\ \mathbf{F} \in \mathbb{G}^t; & Z = aP + \alpha H\ \wedge \\ a, \alpha \in \mathbb{F}_{\bar{p}},\ \boldsymbol{\beta} \in \mathbb{F}_{\bar{p}}^t, \boldsymbol{\gamma} \in \mathbb{F}_{\bar{p}}^s & \mathbf{F} = a\mathbf{Q} + \boldsymbol{\beta} H\ \wedge \\ & \{0\}^s = a\mathbf{S} + \boldsymbol{\gamma} H \end{array} \right\} \quad // \text{(62)}$$

// Precondition: $P, \mathbf{Q}, H$ in $\mathcal{R}$ satisfy $\mathrm{nz}(\mathbf{Q}) \neq \varnothing \ \wedge\ P \mathrel{!=} \mathrm{lin}(\mathrm{nz}(\mathbf{Q}) \cup \mathbf{S} \cup \{H\}) \ \wedge\ H \mathrel{!=} \mathrm{lin}(\mathrm{nz}(\mathbf{Q}) \cup \{P\})$

$\boxed{\mathcal{V}}$: $\boldsymbol{\delta} \leftarrow\!\!\$\ \mathbb{F}_{\bar{p}}^{t*}, \boldsymbol{\sigma} \leftarrow\!\!\$\ \mathbb{F}_{\bar{p}}^{s*}$

$\boxed{\mathcal{V} \to \mathcal{P}}$: $\boldsymbol{\delta}, \boldsymbol{\sigma}$

$\boxed{\mathcal{P}}$: computes $\hat{\alpha} = \alpha + \langle \boldsymbol{\delta}, \boldsymbol{\beta} \rangle + \langle \boldsymbol{\sigma}, \boldsymbol{\gamma} \rangle$

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: compute $X = P + \langle \boldsymbol{\delta}, \mathbf{Q} \rangle + \langle \boldsymbol{\sigma}, \mathbf{S} \rangle$
$Y = Z + \langle \boldsymbol{\delta}, \mathbf{F} \rangle$
and run any complete, sHVZK, and cWEE protocol that convinces $\mathcal{V}$
that $\mathcal{P}$ knows witness $(a, \hat{\alpha})$ for the relation (4), i.e.,
that X and Y are connected as $Y = aX + \hat{\alpha} H$

Figure 24: Random weighting for two t-s-tuples

Note, the premise of Theorem 14 introduces a couple of preconditions in the form $A \mathrel{!=} \mathrm{lin}(\mathbf{B})$ which easily implements as $A = \mathcal{H}_{\mathbf{point}}(\mathbf{B})$. This form of precondition is weaker than $\mathrm{ort}(\{A\} \cup \mathbf{B})$ which is a shorthand of the DL relation assumption [7] for $\{A\} \cup \mathbf{B}$. Thus, a theorem having the precondition $A \mathrel{!=} \mathrm{lin}(\mathbf{B})$ is stronger than a theorem with the precondition $\mathrm{ort}(\{A\} \cup \mathbf{B})$.

Since we do not have a separate assumption for premises in the form $A\,!=\text{lin}(\mathbf{B})$, only for those in the form $\text{ort}(\mathbf{C})$, here is a rule for how the first form translates to the second. If $\text{ort}(\mathbf{B})$ holds, then $A\,!=\text{lin}(\mathbf{B})$ is equivalent to $\text{ort}(\{A\}\cup\mathbf{B})$ and thus the translation is done. Otherwise, if $\text{ort}(\mathbf{B})$ does not hold, then there exists a set $\hat{\mathbf{B}}\subset\mathbf{B}$ together with some coefficients known to prover such that $\text{ort}(\hat{\mathbf{B}})\ \wedge\ \forall B\in\mathbf{B} : B=\text{lin}(\hat{\mathbf{B}})$. Thus, $A\,!=\text{lin}(\mathbf{B})$ translates to $\text{ort}(\{A\}\cup\hat{\mathbf{B}})$ in this case. Having defined such a translation, we have shown that Theorem 14 remains under the DL relation assumption.

9.2.2 SIGNATURE MULTRATUG

Theorem 15:
The scheme in Figure 25 obtained from the scheme in Figure 22 by appending the element vector $\mathbf{U^{tmp}}$ *and substituting the new key image vector* $\hat{\mathbf{I}}$ *for the vector* $\mathbf{I}$ *in it, as shown in Figure 25, is a linkable threshold ring signature retaining the properties 1. . . 9) of the scheme in Figure 22 listed in Theorem 13.*

Proof: is in Appendix T.
Overview: Section 9.1.2.

Thus, we have created the Multratug signature scheme and proved that it has all the properties shown in Table 2.

9.3 SIZE AND COMPLEXITY

The size of Multratug increases by l compared to EFLRSLWB because of the appended vector $\mathbf{U^{tmp}}$. Also, for the same reason, its verification complexity increases by l under the multi-exponent. The substitution of $\hat{\mathbf{I}}$ for $\mathbf{I}$ affects neither the size nor complexity. The totals are shown in Table 6.

Table 6: **Multratug** signature size and verification complexity

	Size	Verification complexity
Multratug[*]	$2\lceil\log_2(n+l)\rceil+7l+6$	$\boldsymbol{mexp}(\,4n+2\log_2(n+l)+8l+7\,)+(n+l+2)\mathbf{H_{pt}}$

[*] Optimized size is shown in Table 7.

$\boxed{\texttt{EFLRSLWBLI.SignAndVerify}_{l,n}(\mathsf{M}, \mathbf{P}, \mathbf{A}, A^{\mathbf{sum}}, D; \mathbf{s}, \mathbf{x}, d^{\Delta\mathbf{sum}})}$

$\mathcal{P}$'s input : $(\mathsf{M} \in \{0,1\}^\star, \mathbf{P} \in \mathbb{G}^{n*}, \mathbf{A} \in \mathbb{G}^n, A^{\mathbf{sum}} \in \mathbb{G}, D \in \mathbb{G}^*; \mathbf{s} \in [0 \dots n-1]^l, \mathbf{x} \in \mathbb{F}_{\bar{\mathsf{p}}}^{l*}, d^{\Delta\mathbf{sum}} \in \mathbb{F}_{\bar{\mathsf{p}}})$

$\mathcal{V}$'s input : $(\mathsf{M} \in \{0,1\}^\star, \mathbf{P} \in \mathbb{G}^{n*}, \mathbf{A} \in \mathbb{G}^n, A^{\mathbf{sum}} \in \mathbb{G}, D \in \mathbb{G}^*)$

$\mathcal{P}$'s output : $\mathit{Signature}$ // signature is a list of all $\mathcal{P} \to \mathcal{V}$ messages from this and nested protocols

$\mathcal{V}$'s output: Accept or Reject

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: **assert** all elements in $\mathbf{P}$ are nonzero and different

let $\mathbf{U} \leftarrow \{\mathcal{H}_{\mathbf{point}}(\mathbf{P}_{[i]})\}_{i=0}^{n-1}$

$\boxed{\mathcal{P}}$: allocates $\hat{\mathbf{I}} \in \mathbb{G}^{l*}$, $\mathbf{p} \in \mathbb{F}_{\bar{\mathsf{p}}}^{l*}$,

initializes **foreach** $k \in [0 \dots l-1]$

assert $\mathbf{x}_{[k]} \neq 0$

$\mathbf{p}_{[k]} = \mathbf{x}_{[k]}^{-1}$

lets $(s_k, p_k) \leftarrow (\mathbf{s}_{[k]}, \mathbf{p}_{[k]})$,

$\hat{\mathbf{I}}_{[k]} = x_k \mathbf{U}_{[s_k]}$ // vector $\hat{\mathbf{I}}$ is filled in here

endforeach

$\boxed{\mathcal{P} \to \mathcal{V}}$: $\hat{\mathbf{I}}$

$\boxed{\mathcal{V}}$: **assert** all elements in $\hat{\mathbf{I}}$ are nonzero and different // $\mathcal{V}$ makes sure there is no zero I and no signer signing twice

$\epsilon \hookleftarrow\$ \mathbb{F}_{\bar{\mathsf{p}}}^*$

$\boxed{\mathcal{V} \to \mathcal{P}}$: ϵ

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: let $H \leftarrow \mathcal{H}_{\mathbf{point}}(\epsilon)$ // thus, H is orthogonal to all known so far elements, i.e., $\mathrm{ort}(H, G, \mathbf{P}, \mathbf{A}, \mathbf{U}, \hat{\mathbf{I}}, A^{\mathbf{sum}}, D)$

$\boxed{\mathcal{P}}$: $\boldsymbol{\mu}, \hat{\boldsymbol{\mu}}, \boldsymbol{\upsilon} \hookleftarrow\$ \mathbb{F}_{\bar{\mathsf{p}}}^{l*}$, allocates $\mathbf{A}^{\mathbf{tmp}}, \mathbf{U}^{\mathbf{tmp}} \in \mathbb{G}^{l*}$, $\boldsymbol{\alpha}, \hat{\boldsymbol{\alpha}} \in \mathbb{F}_{\bar{\mathsf{p}}}^{l*}$,

initializes **foreach** $k \in [0 \dots l-1]$

lets $(\mu_k, \hat{\mu}_k) \leftarrow (\boldsymbol{\mu}_{[k]}, \hat{\boldsymbol{\mu}}_{[k]})$,

$\mathbf{A}^{\mathbf{tmp}}_{[k]} = \mathbf{A}_{[s_k]} + \mu_k H$ // $\mathbf{A}^{\mathbf{tmp}}$ is filled in, amounts get double blinded (with D and with H)

$\boldsymbol{\alpha}_{[k]} = p_k\, \mu_k$ // $\boldsymbol{\alpha}$ is initialized here, it contains reduced $\mathbf{A}^{\mathbf{tmp}}$'s second blinding factors

$\mathbf{U}^{\mathbf{tmp}}_{[k]} = \mathbf{U}_{[s_k]} + \hat{\mu}_k H$ // $\mathbf{U}^{\mathbf{tmp}}$ is filled in, U's get blinded with H

$\hat{\boldsymbol{\alpha}}_{[k]} = p_k\, \hat{\mu}_k$ // $\hat{\boldsymbol{\alpha}}$ is initialized here, it contains reduced $\mathbf{U}^{\mathbf{tmp}}$'s blinding factors

endforeach

$\boxed{\mathcal{P} \to \mathcal{V}}$: $\mathbf{A}^{\mathbf{tmp}}$, $\mathbf{U}^{\mathbf{tmp}}$

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: let $\hat{\mathbf{U}} \leftarrow \{\, \mathcal{H}_{\mathbf{point}}(H, \mathbf{U}^{\mathbf{tmp}}, \mathbf{A}^{\mathbf{tmp}}_{[k]})\,\}_{k=0}^{l-1}$

$\boxed{\mathcal{P}}$: lets $\mathbf{J} \leftarrow \{p_k \hat{\mathbf{U}}_{[k]} + \upsilon_k H\}_{k=0}^{l-1}$ // vector $\mathbf{J}$ is initialized here, it contains 'pseudo key images' built using $\hat{\mathbf{U}}$

$\boxed{\mathcal{P} \to \mathcal{V}}$: $\mathbf{J}$

$\boxed{\mathcal{V}}$: **assert** all elements in $\mathbf{A}^{\mathbf{tmp}}$, $\mathbf{J}$ are nonzero and different // $\mathcal{V}$ makes sure $\hat{\mathbf{U}}$ is orthogonal and there is no zero J

$\hat{\epsilon}, \zeta, \omega, \chi, \theta \hookleftarrow\$ \mathbb{F}_{\bar{\mathsf{p}}}^*$

$\boxed{\mathcal{V} \to \mathcal{P}}$: $\hat{\epsilon}, \zeta, \omega, \chi, \theta$

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: let $K \leftarrow \mathcal{H}_{\mathbf{point}}(\hat{\epsilon})$ // thus, $\mathrm{ort}(K, H, G, \mathbf{P}, \mathbf{A}, \mathbf{U}, \mathbf{I}, A^{\mathbf{sum}}, \mathbf{A}^{\mathbf{tmp}}, \hat{\mathbf{U}}, \mathbf{J})$ holds

allocate $\mathbf{X} \in \mathbb{G}^{n*}$, $\mathbf{V}, \mathbf{Z} \in \mathbb{G}^{l*}$, $S \in \mathbb{G}$,

assign $\mathbf{X} = \mathbf{P} - \{K\}^n + \zeta \mathbf{U} - \omega \mathbf{A}$, $\quad \mathbf{V} = \{K\}^l + \omega \mathbf{A}^{\mathbf{tmp}} - \zeta \mathbf{U}^{\mathbf{tmp}} + \theta \hat{\mathbf{I}} + \chi \hat{\mathbf{U}}$,

$\mathbf{Z} = \{G\}^l + \theta \mathbf{U}^{\mathbf{tmp}} + \chi \mathbf{J}$

assign $S = A^{\mathbf{sum}} - \sum_{k=0}^{l-1} \mathbf{A}^{\mathbf{tmp}}_{[k]}$

run $\texttt{zk2ElemComm}(D, H, S;\ d^{\Delta\mathbf{sum}}, -\sum_{k=0}^{l-1} \mu_k)$

run $\texttt{zkLin22Choice}_{l,n,l}(\mathbf{X}, \mathbf{G}_{[:n]}, \mathbf{V}, \mathbf{G}_{[n:(n+l)]}, H, \mathbf{Z};\ \mathbf{s}, \mathbf{p}, \mathbf{p},\ -\omega \boldsymbol{\alpha} + \zeta \hat{\boldsymbol{\alpha}} + \theta \hat{\boldsymbol{\mu}} + \chi \boldsymbol{\upsilon})$

Figure 25: Multratug with $\hat{I} = x \mathcal{H}_{\mathbf{point}}(P)$ signing and verification

10 BETTER ARGUMENT FOR VECTOR COMMITMENT

The implementation of our pivotal vector commitment argument $\mathtt{zkVC}_n$ in Figure 3 is not decisive. We will now present a shorter implementation of it, called $\mathtt{zkVC}_n^{\mathbf{opt}}$, with the same properties of completeness, sHVZK, and cWEE. This our implementation utilizes the same ideas as the compressed pivot implementation in [2].

10.1 OVERVIEW

The idea is that, for any $n \geqslant 1$, it is always possible to construct an sHVZK and having cWEE custom Schnorr-like protocol of size $n+1$, that proves a commitment Y is a weighted sum of n orthogonal generators $\mathbf{X}$ with weights known to the prover.

In this protocol, prover sends an element T as the first message. Then, verifier challenges with random scalar c, and the prover replies with n scalars $\boldsymbol{\tau}$ by which the orthogonal generators $\mathbf{X}$ are then multiplied. The final check is the same as for the Schnorr id protocol, the only difference is that now the inner product $\langle \boldsymbol{\tau}, \mathbf{X} \rangle$ is taken instead of the basic generator multiplied by the scalar replied in the Schnorr id scheme.

However, it is excessive to transmit all n scalars in $\boldsymbol{\tau}$; a proof of their knowledge suffices. Moreover, this proof does not have to be sHVZK, a complete argument having cWEE is enough.

10.2 FORMAL PRESENTATION

$\boxed{\mathtt{zkNElemComm}_n(\mathbf{X}, Y;\, \mathbf{x})}$

Relation $\mathcal{R} = \{\, \mathbf{X} \in \mathbb{G}^{n*},\ Y \in \mathbb{G};\ \mathbf{x} \in \mathbb{F}_{\bar{p}}^{n} \mid Y = \langle \mathbf{x}, \mathbf{X} \rangle \,\}$ // (70)

// X in $\mathcal{R}$ satisfies $\mathrm{ort}(X)$.

$\mathcal{P}$'s input : $(\mathbf{X}, Y; \mathbf{x})$

$\mathcal{V}$'s input : $(\mathbf{X}, Y)$

$\mathcal{P}$'s output : none

$\mathcal{V}$'s output: *Accept* or *Reject*

$\boxed{\mathcal{P}}$: $\boldsymbol{\phi} \leftarrow\!\!\$\ \mathbb{F}_{\bar{p}}^{n*}$ and computes $T = \langle \boldsymbol{\phi}, \mathbf{X} \rangle$

$\boxed{\mathcal{P} \to \mathcal{V}}$: T

$\boxed{\mathcal{V}}$: $c \leftarrow\!\!\$\ \mathbb{F}_{\bar{p}}^{*}$

$\boxed{\mathcal{V} \to \mathcal{P}}$: c

$\boxed{\mathcal{P}}$: computes $\boldsymbol{\tau} = \boldsymbol{\phi} - c\mathbf{x}$

$\boxed{\mathcal{P} \to \mathcal{V}}$: $\boldsymbol{\tau}$

$\boxed{\mathcal{V}}$: **return**s *Accept* iff the following holds

$$T - cY \stackrel{?}{=} \langle \boldsymbol{\tau}, \mathbf{X} \rangle$$

Figure 26: Zero-knowledge argument for n element commitment relation

For the first, we define the protocol $\mathtt{zkNElemComm}_n$ in Figure 26. It has the Schnorr-like design. This protocol is an argument for the relation

$$\mathcal{R} = \{\, \mathbf{X} \in \mathbb{G}^{n*},\ Y \in \mathbb{G};\ \mathbf{x} \in \mathbb{F}_{\bar{p}}^{n} \mid Y = \langle \mathbf{x}, \mathbf{X} \rangle \,\}\,. \tag{70}$$

The relation (70) is actually the relation (7) with the items renamed and, at the same time, is the relation (5) with the blinding generator H moved to the vector $\mathbf{X}$.

The $\mathtt{zkNElemComm}_n$ protocol properties are specified in the next theorem. Note that, for $n = 2$, $\mathtt{zkNElemComm}_2$ is equivalent to $\mathtt{zk2ElemComm}$ in Figure 2.

Theorem 16:
For $n \in \mathbb{N}^*$, for a vector of nonzero elements $\mathbf{X} \in \mathbb{G}^{n*}$ such that $\mathrm{ort}(\mathbf{X})$ holds, for an element $Y \in \mathbb{G}$, the protocol $\mathtt{zkNElemComm}_n$ in Figure 26 is a complete, sHVZK argument having cWEE for the relation (70) with unique witness.

Proof: is in Appendix U.

For the second, in Figure 27 we define a log-size vector commitment argument $\texttt{argVC}_n$ for the same relation (70). We use the blinding generator H neither in $\texttt{zkNElemComm}_n$ nor in $\texttt{argVC}_n$. Also, note that $\texttt{zkNElemComm}_n$ is sHVZK, whereas $\texttt{argVC}_n$ is not. The properties of $\texttt{argVC}_n$ are specified in the following theorem.

Theorem 17:
For $n \in \mathbb{N}^*$ such that n is a power of 2, for a vector of nonzero elements $\mathbf{X} \in \mathbb{G}^{n*}$ such that $\mathrm{ort}(\mathbf{X})$ holds, for an element $Y \in \mathbb{G}$, the protocol $\texttt{argVC}_n$ in Figure 27 is a complete argument having cWEE for the relation (70) with unique witness.

Proof: is in Appendix V.

$\boxed{\texttt{argVC}_n(\mathbf{X}, Y;\, \mathbf{x})}$

Relation $\mathcal{R} = \{\, \mathbf{X} \in \mathbb{G}^{n*},\ Y \in \mathbb{G};\ \mathbf{x} \in \mathbb{F}_{\bar{p}}^{n} \mid Y = \langle \mathbf{x}, \mathbf{X} \rangle \,\}$ // (70)

// $\mathbf{X}$ in $\mathcal{R}$ satisfies $\mathrm{ort}(\mathbf{X})$, n is a power of 2 everytime.

$\mathcal{P}$'s input : $(\mathbf{X}, Y;\, \mathbf{x})$

$\mathcal{V}$'s input : $(\mathbf{X}, Y)$

$\mathcal{P}$'s output : none

$\mathcal{V}$'s output: *Accept* or *Reject*

if $n > 4$ **then**

$\boxed{\mathcal{P}}$: lets $\hat{n} \leftarrow n/2$ and computes $L = \left\langle \mathbf{x}_{[:\hat{n}]}, \mathbf{X}_{[\hat{n}:]} \right\rangle$

$R = \left\langle \mathbf{x}_{[\hat{n}:]}, \mathbf{X}_{[:\hat{n}]} \right\rangle$

$\boxed{\mathcal{P} \rightarrow \mathcal{V}}$: L, R

$\boxed{\mathcal{V}}$: $e \leftarrow\$\ \mathbb{F}_{\bar{p}}^{*}$

$\boxed{\mathcal{V} \rightarrow \mathcal{P}}$: e

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: compute $\hat{\mathbf{X}} = e^{-1}\mathbf{X}_{[:\hat{n}]} + e\,\mathbf{X}_{[\hat{n}:]}$

$\hat{Y} = Y + e^{2}L + e^{-2}R$

$\boxed{\mathcal{P}}$: computes $\hat{\mathbf{x}} = e\,\mathbf{x}_{[:\hat{n}]} + e^{-1}\mathbf{x}_{[\hat{n}:]}$

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: run $\texttt{argVC}_{\hat{n}}(\hat{\mathbf{X}}, \hat{Y};\, \hat{\mathbf{x}})$ // run recursively until n=4

else // $n \leqslant 4$

$\boxed{\mathcal{P} \rightarrow \mathcal{V}}$: $\mathbf{x}$

$\boxed{\mathcal{V}}$: **return**s *Accept* iff the following holds

$$Y \stackrel{?}{=} \langle \mathbf{x}, \mathbf{X} \rangle$$

endif

Figure 27: Efficient argument for vector commitment

Third, we combine $\texttt{zkNElemComm}_n$ with $\texttt{argVC}_n$ into the single proof, as follows.

$\boxed{\mathtt{zkVC}_n^{\mathbf{opt}}(\mathbf{X}, H, Y;\ \mathbf{a}, \alpha)}$

Relation $\mathcal{R} = \{\, \mathbf{X} \in \mathbb{G}^{n*}, H \in \mathbb{G}^*, Y \in \mathbb{G};\ \mathbf{a} \in \mathbb{F}_{\bar{p}}^{n}, \alpha \in \mathbb{F}_{\bar{p}} \mid Y = \langle \mathbf{a}, \mathbf{X} \rangle + \alpha H \,\}$ // (5)

// $\mathbf{X}, H$ in $\mathcal{R}$ satisfy $\mathrm{ort}(\mathbf{X} \cup \{H\})$, and also $(n+1)$ is a power of 2 everytime.

$\mathcal{P}$'s input : $(\mathbf{X}, H, Y; \mathbf{a}, \alpha)$

$\mathcal{V}$'s input : $(\mathbf{X}, H, Y)$

$\mathcal{P}$'s output : none

$\mathcal{V}$'s output: *Accept* or *Reject*

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: let $\hat{\mathbf{X}} \leftarrow [\mathbf{X}, H]$

$\boxed{\mathcal{P}}$: $\boldsymbol{\phi} \leftarrow\!\$\ \mathbb{F}_{\bar{p}}^{(n+1)*}$, lets $\hat{\mathbf{x}} \leftarrow [\mathbf{x}, \alpha]$, and computes $T = \left\langle \boldsymbol{\phi}, \hat{\mathbf{X}} \right\rangle$

$\boxed{\mathcal{P} \rightarrow \mathcal{V}}$: T

$\boxed{\mathcal{V}}$: $c \leftarrow\!\$\ \mathbb{F}_{\bar{p}}^{*}$

$\boxed{\mathcal{V} \rightarrow \mathcal{P}}$: c

$\boxed{\mathcal{P}}$: computes $\boldsymbol{\tau} = \boldsymbol{\phi} - c\hat{\mathbf{x}}$

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: run $\mathtt{argVC}_{n+1}(\hat{\mathbf{X}},\ T - cY;\ \boldsymbol{\tau})$

Figure 28: Efficient zero-knowledge argument for vector commitment

Theorem 18:
For a nonzero element $H \in \mathbb{G}^*$, for $n \in \mathbb{N}^*$ such that $(n+1)$ is a power of 2, for a vector of nonzero elements $\mathbf{X} \in \mathbb{G}^{n*}$ such that $\mathrm{ort}(\mathbf{X} \cup \{H\})$ holds, for an element $Y \in \mathbb{G}$, the protocol $\mathtt{zkVC}_n^{\mathbf{opt}}$ in Figure 28 is a complete, sHVZK argument having cWEE for the relation (5) with unique witness.

Proof: is in Appendix W.

10.3 SIZES AND COMPLEXITIES

As a result, we obtain the argument $\mathtt{zkVC}_n^{\mathbf{opt}}$ of size $2\lceil \log_2(n+1) \rceil + 1$.We replace $\mathtt{zkVC}_n$ with $\mathtt{zkVC}_n^{\mathbf{opt}}$ in Multratug and EFLRSL. After this replacement, new sizes of the signatures are shown in Table 7. Their verification complexities do not change much, so we do not recalculate them. For comparison, the former sizes and times are shown in Table 4 and Table 6. Also, from now on we require $(n+l+1)$ and $(n+1)$ to be powers of 2, respectively.

Table 7: Optimized characteristics of the **Multratug** and **EFLRSL** schemes

	Size	Verification complexity
Multratug	$2\lceil \log_2(n+l+1) \rceil + 7l + 4$	$\boldsymbol{mexp}(\,4n + 8l + \ldots) + (n+l+2)\mathbf{H_{pt}}$
EFLRSL	$2\lceil \log_2(n+1) \rceil + 3l + 1$	$\boldsymbol{mexp}\,(\,3n + 2l + \ldots) + (n+1)\mathbf{H_{pt}}$

. . . Insignificant summands are omitted.

11 APPLICATIONS

11.1 REGULAR RING SIGNATURE

EFLRSL, which was obtained in the first chapters of this paper, can be regarded as a streamlined version of Multratug. It is a regular linkable threshold ring signature which, in terms of Table 2, has Log-sz, Regular, Linkable, Thresh., General properties check-marked. Consequently, EFLRSL can be used in a wide range of cryptographic systems and scenarios, including electronic voting or whistleblowing described, e.g., in [23].

If an application requires a signature to be unlinkable, then an unlinkable version of EFLRSL can be easily constructed. For instance, it can be done by blinding the EFLRSL key images. To blind the key images it suffices to exclude them from the arguments of $\mathcal{H}_{\mathbf{point}}$ call that creates the blinding generator H, and to add randomly sampled H components to them.

11.2 SIGNATURE IN BLOCKCHAIN

Suppose that Multratug is used to sign transactions in an UTXO blockchain like, e.g., [24, 31]. Suppose the blockchain public keys, hidden amounts, hash functions, and predefined generators follow the rules in Figures 1, 9, 20, 21. There is nothing unusual for a blockchain in these requirements. Futhermore, the blockchain does not have to follow the CryptoNote rules for stealth addresses [31], although it can.

For every transaction, its sender $\mathcal{P}$ performs as follows.

- Picks n pairs of the form (P, A) from the ledger, they become transaction inputs, and makes the ring (38) of them.
- Generates and places into the transaction m pairs of the form (P, A), which become the transaction outputs. For convenience, it considers all m hidden amounts A of these outputs as the vector $\mathbf{A^{out}}$. Note, knowing the actual signing keys in the ring and their corresponding hidden amounts, $\mathcal{P}$ distributes the value parts of elements in $\mathbf{A^{out}}$ in such a way as to be in balance with the signing amounts in the ring. $\mathcal{P}$ samples the blinding parts of the elements in $\mathbf{A^{out}}$ independently and uniformly.
- Lets $A^{\mathbf{sum}} = \sum_{k=0}^{m-1} A_k^{\mathbf{out}}$.
- Knowing the actual signing private keys whose corresponding public key indices are in the vector $\mathbf{s}$, $\mathcal{P}$ signs the transaction with the Multratug signature.
- $\mathcal{P}$ proves ranges of all elements in $\mathbf{A^{out}}$, for example, using the aggregate range proof from [9] which is combinable with Multratug, as shown in Section 12.3.
- Proves that each $A_k^{\mathbf{out}} \in \mathbf{A^{out}}$ has the decomposition (40) with known to $\mathcal{P}$ coefficients. Notably, if the ranges of elements in $\mathbf{A^{out}}$ are already proved by the protocols from [7, 9], then, for all $A_k^{\mathbf{out}} \in \mathbf{A^{out}}$, their decompositions (40) are proved by this.

Thus, the transaction contains the proofs of the form (40) for all of the output hidden amounts in $\mathbf{A^{out}}$. Also, the transaction contains an instance of the Multratug signature which provides the proof that $\sum_{k=0}^{m-1} A_k^{\mathbf{out}}$ is equal to the sum $\sum_{k=0}^{l-1} A_{s_k}$ of all hidden amounts related to the signing indices $\mathbf{s}$, to the accuracy of D.

Taking into account that all A_{s_k}'s are a subset of all hidden amounts $\mathbf{A}$ in the ring, and for the latter it is assumed that they have already been verified to have the form (40), it follows that the sum of amounts corresponding to the actual signing keys is equal to the sum of the output amounts, i.e.,

$$\sum_{k=0}^{l-1} b_{s_k} = \sum_{k=0}^{m-1} b_k^{\mathbf{out}}.$$

At the same time, the same instance of Multratug proves that $\mathcal{P}$ knows private keys for the actual signing public keys at indices in $\mathbf{s}$. Also, this instance of Multratug delivers key images $\mathbf{I}$'s, thus blocking reuse of those public keys that actually signed the transaction.

12 EXTENSION AND IMPROVEMENTS

12.1 USING RING OF SIZE N·L

It is possible to slightly reduce the size of the Multratug signature by not using the Lin2-2Choice lemma and instead by growing the ring l times as to comprise l replicas of itself, each for its hidden amount $A_k^{\mathbf{tmp}}$. In this case, after the appropriate optimizations, the signature size would be

$$2\log_2(nl) + 5l + O(1).$$

However, we still prefer the version with the Lin2-2Choice lemma, since not using it implies that the ring grows to nl size. This would require to add more generators to keep all the ring elements linearly independent of each other and, hence, will correspondingly increase l times the verification complexity.

12.2 BATCH VERIFICATION

Multratug signature batch verification can be performed by checking only one equality, by combining the equalities (*) and (**) in Figure 23 of all signatures in a batch using random weighting. Of course, the equality (*) slightly changes when $\mathtt{zkVC}_n^{\mathbf{opt}}$ is used in place of $\mathtt{zkVC}_n$, this is a minor detail and we do not show the change here.

In any case, for batches, the asymptotic verification complexity by ring size n decreases from $4n$ to $3n$ under the multi-exponent. This happens due to the fact that all the Multratug signature batch instances use the same vector of predefined generators $\mathbf{G}$. The same can be stated about EFLRSL, referring to Figure 16 and finding there a reduction from $3n$ to $2n$ under the multi-exponent.

12.3 COMBINING WITH OTHER PROOFS

Multratug relies upon the pivotal vector commitment argument and is independent of implementation of the pivot. Consequently, Multratug can be combined with any other proof which uses the vector commitment argument. For instance, it can be combined with the inner product argument implemented according to [7] or [9].

In this way, Multratug can be combined with the single or aggregate range proofs from [9], and they will share the component responsible for the sum

$$\sum_{j=0}^{\log_2(n+l+n^{\mathbf{rangeproof}})-1} (e_j^2 L_j + e_j^{-2} R_j),$$

where $n^{\mathbf{rangeproof}}$ is equal to bitsize of the range times number of proofs aggregated.

12.4 DOWNGRADING TO U/X KEY IMAGE

In the case of using our signature in a blockchain confined to the stealth address format of CryptoNote [31], it is possible to replace the key image form $x^{-1}\mathcal{H}_{\mathbf{point}}(xG)$ with the form $x^{-1}U$, where U is a predefined orthogonal generator. This can be performed for the EFLRSLWB version of the signature defined in Section 8.

Of course, such a replacement would require expanding the vector $\mathbf{G}$ of predefined orthogonal generators so as to use them instead of $\mathcal{H}_{\mathbf{point}}(P_i)$'s in the ring. The size of the signature will not change after that. However, the batch verification time will be significantly reduced.

12.5 MULTIPLE HIDDEN AMOUNTS PER ACCOUNT

In the context of blockchain, particularly in the scenario described in Section 11.2, as well as in other cases, we can consider a setup where several hidden amounts are associated with a public key, instead of one. To be precise, we can assume that for each i-th address (39) in the ring, $i \in [0 \dots n-1]$, instead of the hidden amount A_i defined by the formula (40) there are u hidden amounts $\{A_{ij}\}_{j=0}^{u-1}$ defined by the following formula

$$A_{ij} = b_{ij} B_j + d_{ij} D\,. \tag{71}$$

According to this new formula (71) which replaces (40), now, for each signing index s_k, $k \in [0 \dots l-1]$, prover $\mathcal{P}$ is required to know u amounts $\{b_{s_k,j}\}_{j=0}^{u-1}$ along with u blinding factors $\{d_{s_k,j}\}_{j=0}^{u-1}$. Also, according to (71), now there are u orthogonal generators $\{B_j\}_{j=0}^{u-1}$ instead of the single generator B in the system, hence the common information in Figure 21 is assumed extended with them. Each j-th hidden amount is encoded with the corresponding generator B_j. The blinding generator D remains intact and is used for all of the amounts.

Finally, for this setup, each of the output hidden amounts $A \in \mathbf{A}^{\mathbf{out}}$ is replaced with u new hidden amounts of the form (71), with $i \in [0 \dots m-1]$, $j \in [0 \dots u-1]$ for them. It is assumed that some external range proofs are provided for all of the output hidden amounts as well.

With this setup, the signature Multratug needs no modification to convince $\mathcal{V}$ that u balances are kept. For this, all u hidden amounts of each address are convolved back into the single element A_i, just as follows,

$$A_i = \sum_{j=0}^{u-1} b_{ij} B_j + \sum_{j=0}^{u-1} d_{ij} D\,,$$

and the same is for the output hidden amounts. After that, the signature Multratug is signed and published for them. It is easy to see that, since for each j the amount b_{ij} is encoded with the corresponding orthogonal generator B_j, the amounts for different j's do not intermix. Thus, all u balances get proved at the price of one, as in Table 7.

12.6 APPLICATION TO KZG COMMITMENTS

The Kate-Zaverucha-Goldberg (KZG) commitment scheme [17] allows to commit to polynomials of some predefined and typically large degree and then to construct succinct arguments of knowledge of values of these polynomials at points. KZG commitments are used in modern proof systems, e.g., in PLONK [11] by A. Gabizon, Z. J. Williamson, and O. Ciobotaru. For a quick dive into this topic, we refer to the explanatory overview [5] by D. Boneh.

Now let us recall that the generic idea of the Lin2-Choice lemma outlined in Section 4.1.3 is that, for a given set of n fixed linearly independent elements, we, informally, construct a linear expression with n degrees of freedom from them. Given one more fixed element, which we consider as a commitment, we construct another linear expression with one degree of freedom from it. In addition, we have a private mask $\mathbf{a}$ of n scalars, which is also considered fixed. We randomize the first linear expression in all its n dimensions and, also, overlap it with the mask $\mathbf{a}$. Finally, we show that the part of the randomized first expression that is 'seen' through the mask $\mathbf{a}$ is equal to the second expression, for which we control its single degree of freedom. This convinces the verifier that our private mask $\mathbf{a}$ is one-hot, and hence the given commitment corresponds to exactly one member of the given set of n elements.

In Appendix Z, we show how this key idea can be applied to KZG commitments. In fact, instead of dealing with elements and commitments in a prime-order group under DDH, we might try to use linearly independent objects and committments of a different nature under slightly different assumptions. For instance, the given above set of linearly independent elements and commitment might be replaced by a set of linearly independent polynomials and a KZG commitment.

As another example, the set can be an image of a polynomial on some subdomain $\mathbf{\Omega}$ such that the polynomial is nonzero on $\mathbf{\Omega}$. The commitment can be a KZG commitment to another polynomial. In this case, using the same considerations we can argue that the opening of the commitment is one-hot on $\mathbf{\Omega}$.

13 COMPARISON

We compare our optimized Multratug and EFLRSL (Table 7) with the best performing signatures listed in Table 1, namely, with Lelantus Spark [16], Omniring [21], RingCT3.0 [32], Triptych [25], and DualRing-EC [33], taking linear-size CLSAG [13] for the base.

We distinguish two gradations of scheme anonymity inherently bound to the two key image (linking tag) forms used. In general, if a scheme has a key image or another public element in the form $x^{-1}U$, then it has lower anonymity unless a compensatory restriction is imposed on the keys. Key images in the forms $x^{-1}\boldsymbol{\mathcal{H}}_{\mathbf{point}}(P)$ and $x\boldsymbol{\mathcal{H}}_{\mathbf{point}}(P)$ are stronger and entail no key restrictions, however, it is still required that the scheme has no other public elements in the form $x^{-1}U$. More on this in Appendix Y.

13.1 FOR MULTRATUG

The signatures with balance proofs are compared in Table 8. Notation is as follows. $\mathbf{H_{sc}}$ is the time of taking a scalar hash, it is omitted when its multiplier is logarithmic or less. $\mathbf{H_{pt}}$ is the time of taking a hash to curve, $\boldsymbol{mexp}(N)$ is the time of multi-exponentiation of N summands.

The schemes with 'Any keys=Yes' operate with arbitrary keys; those with 'Any keys=No' require special key format, e.g., as in [31]. Our signature receives 'Any keys=Yes', as according to Theorem 15 and, hence, by Theorem 13 it has the EU_CMA/CPA, anonymity w.r.t. CPA, non-frameability w.r.t. CPA properties.

Lelantus Spark [16] has key image $x^{-1}U$, nevertheless, according to the original paper it has a subsystem that facilitates multiparty signing, so we set 'MP=Yes' for it. The other schemes receive 'MP=Yes' only if their key images are linear by x. Also, for Lelantus Spark, we only count the size of its parallel 1-out-of-many proof from the section '7 Efficiency' in [16], so its actual size may have a few extra bytes.

For this comparison, we exclude key images together with input/output accounts which occupy the same space for all schemes. Also, we do not include the output range proofs assuming they are separated into distinct units, although according to Section 12.3 our scheme effectively integrates with them, as does Omniring [21].

Batch verification time, which is explained for our scheme in Section 12.2, is generally 25%. . . 50% less for all log-size schemes due to common generators merging, we do not show it. Verification complexities of the schemes with the key images $x^{-1}\boldsymbol{\mathcal{H}}_{\mathbf{point}}(P)$ or $x\boldsymbol{\mathcal{H}}_{\mathbf{point}}(P)$ have an additional summand of roughly $n\mathbf{H_{pt}}$, which reflects the fact that $\boldsymbol{\mathcal{H}}_{\mathbf{point}}$ must be called at least once for every public key in the ring.

Multratug is represented by its version with optimized vector commitment argument, with characteristics taken from Table 7; we have subtracted l from its size, since the key images are not counted. The CLSAG, Triptych, and Lelantus Spark schemes have no threshold versions, hence, to compare them with those having threshold ones, their sizes in Table 8 are to be multiplied by l. RingCT3.0 size is taken from the corresponding paper [32]. The same is for Omniring, its size is taken from the section '6.3 Performance Comparison' of [21]. Note, according to its paper, Omniring has $O\log_2(nl + \dots)$ size, whereas in the section *'D Comparison with Omniring'* in [32] it reads as $O\log_2(n + \dots)$, we hold to the first one.

For the ring size $n = 2^5 \dots 2^{10}$ and number of inputs limited to, say, $l \leqslant 5$, which is in accordance to [32, 21, 25], inserted into the corresponding formulas in Table 8, our Multratug looks performing on par with the best-performing signature schemes.

As for applicability in blockchains, we should probably only consider signatures that allow for easy signing by multiple parties, since this seems to be a must-have attribute for a modern blockchain. Therefore, only Lelantus Spark, Omniring version with $x\boldsymbol{\mathcal{H}}_{\mathbf{point}}(P)$, and our signature are to be compared. Table 9 shows their sizes (excluding key images and range proofs) in bytes computed in the mentioned above region of interest. We assume an element in $\mathbb{G}$ and a scalar in $\mathbb{F}_{\bar{p}}$ take 32 bytes each.

Table 8: Comparison of LRS schemes that simultaneously prove the balance

	Size	Verification complexity	Key image	Any keys	MP
CLSAG*	$n+2$	$(n+2)\mathbf{H_{sc}} + 2n\,\boldsymbol{mexp}(3) + n\mathbf{H_{pt}}$	$x\,\mathcal{H}_{\mathbf{point}}(P)$	Yes	Yes
Triptych*	$3\lceil\log_2(n)\rceil + 8$	$\boldsymbol{mexp}(\,2n + \ldots\,)$	$x^{-1}U$	No	No
Lelantus Spark*	$3\lceil\log_2(n)\rceil + 5$	$\boldsymbol{mexp}(\,2n + \ldots\,)$	$x^{-1}U$	No	Yes
RingCT3.0	$2\lceil\log_2(n\,l)\rceil + l + 17$	$\boldsymbol{mexp}(\,2\,n\,l + \ldots\,) + \boldsymbol{mexp}(l+1) + \ldots$	$x^{-1}U$	No	No
Omniring	$2\lceil\log_2(n\,l + n + 3l + 3)\rceil + 9$	$\boldsymbol{mexp}(\,2\,n\,l + \ldots\,)$	$x^{-1}U$	No	No
Omniring	$2\lceil\log_2(n\,l + n + 3l + 3)\rceil + 9$	***	$x\,\mathcal{H}_{\mathbf{point}}(P)$	No	Yes
Multratug**	$2\lceil\log_2(n+l+1)\rceil + 6l + 4$	$\boldsymbol{mexp}(\,4n + 8l + \ldots\,) + (n+l+2)\mathbf{H_{pt}}$	$x\,\mathcal{H}_{\mathbf{point}}(P)$	Yes	Yes

* Authors did not specify any optimized threshold version, assuming it takes up l times the size.
** Scheme version with linear linking tag, Section 9, and optimized vector commitment argument, Section 10.3 .
*** Authors did not specify formula, we assume the quantity is average in its class, about the same as for the version with $x^{-1}U$.
. . . Insignificant summands are omitted.

Table 9: Comparison of LRS schemes with balance that are suitable for blockchain

	$l=1$		$l=2$		$l=3$		$l=4$		$l=5$	
	$n=2^5$	$n=2^{10}$	$n=2^5$	$n=2^{10}$	$n=2^5$	$n=2^{10}$	$n=2^5$	$n=2^{10}$	$n=2^5$	$n=2^{10}$
Lelantus Spark	640	1120	1120	2080	1600	3040	2080	4000	2560	4960
Omniring	704	1024	736	1056	768	1088	768	1088	800	1120
Multratug	672	992	864	1184	1056	1376	1248	1568	1440	1760

Notably, Multratug is the only log-size signature with balance proof of all the listed, which is applicable in blockchains as well as in other environments where keys do not stick to the [31] rules, and where the keys are also allowed to be generated ad-hoc and be malformed as, e.g., in [23, 24, 13].

13.2 FOR EFLRSL

In Table 10 we compare the simplest versions of the signature schemes, which are the ring signatures with one actual signer. So, we take our EFLRSL signature for $l = 1$ with the optimized vector commitment argument (Table 7). We also include in the comparison the DualRing-EC [33] signature which, according to the survey in [33], is the most space-efficient known so far. For this comparison, we don't distinguish between the regular ring signatures and the linkable ones. When both versions are available, we take the regular one. The sizes of DualRing-EC and streamlined versions of RingCT3.0, Omniring are taken from *'Table 1: O(log n)-size DL-based ring signature schemes for n public keys . . . '* in [33].

According to Table 10, for large rings such that $\lceil\log_2(n+1)\rceil = \lceil\log_2(n)\rceil$ almost everytime, both the DualRing-EC and EFLRSL signatures have the best size among the others. However, EFLRSL has a stronger security model, which is explained in Appendix X. Thus, it turns out that the EFLRSL signature for $l = 1$ is the shortest one known to date among the signatures for environments in which malformed keys are allowed.

Table 10: Comparison of DL-based ring signatures

	Size	Verification complexity
CLSAG	$n+1$	$n\,\mathbf{H_{sc}} + n\,\boldsymbol{mexp}(2)$
RingCT3.0	$2\lceil\log_2(n)\rceil + 14$	$\boldsymbol{mexp}(\,2n + \ldots\,) + \ldots$
Omniring	$2\lceil\log_2(n+2)\rceil + 9$	$\boldsymbol{mexp}(\,2\,n\,l + \ldots\,)$
EFLRSL*	$2\lceil\log_2(n+1)\rceil + 4$	$\boldsymbol{mexp}(\,3n + \ldots\,) + (n+1)\mathbf{H_{pt}}$
DualRing-EC**	$2\lceil\log_2(n)\rceil + 4$	$\boldsymbol{mexp}(\,n + \ldots\,)$

* Only linkable version of the ring signature is available.
** See comments in Appendix X.
. . . Insignificant summands are omitted.

14 CONCLUSION

In this paper we presented two novel efficient membership proofs in a prime-order group without bilinear pairings, under the DDH assumption. In the lemmas called Lin2-Choice and Lin2-2Choice we proved these membership proofs are complete, special honest verifier zero-knowledge, and have computational witness-extended emulation. Using our membership proofs we created a trusted-setup-free, pairings-free, DDH-based log-size linkable threshold ring signature with balance proof called Multratug. To illustrate, for a ring of 2^{10} addresses with associated hidden amounts, and for 5 actual signing keys in it, Multratug occupies less than 2KBytes of space, as shown in Table 9.

In addition to its quite moderate size and built-in balance proof, the Multratug signature makes it easy to implement multi-party signing operations with it. Thus, it can be used for signing confidential transactions in a modern blockchain. Multratug can operate securely with any addresses, not only with those which follow the CryptoNote stealth address paradigm. This trait along with the above properties makes the signature applicable in wide range of cryptographic systems. Therefore, Multratug may serve as a log-size drop-in replacement for the well-known linear-size LSAG scheme and its extensions.

We made a comparison which showed that among the existing pairings-free trusted-setup-free log-size signatures under DDH, for large rings and medium thresholds, only a version of the Omniring scheme comprises almost the same set of useful features (Table 1, Table 2) at the minimal size (Table 8). However, the Multratug's security model is proved to be stronger against malformed keys.

For the case when a cryptographic system requires neither a balance proof nor any other additional properties from a signature, just the minimal possible size and a security model strong enough to accept ad hoc generated and malformed keys, we provide a streamlined version of our signature called EFLRSL. It is the most compact signature with the strong security model to date (Table 10), as far as we can find.

Lin2-Choice is the main lemma of this paper, it provides a concise method for proving membership in a linearly independent set. Having proved this lemma in the DDH setting, we have shown as an extension that this lemma's key idea can likely be valid under other assumptions as well.

Our membership proofs and signatures are built upon an arbitrary vector commitment argument (Section 1.2) viewed as a black box. They effectively combine with other arguments such as range proofs to further reduce the overall size. The design of our membership proofs and signatures is modular. We compose them from elementary protocols, and for each one we prove that it is special honest verifier zero-knowledge and has computational witness-extended emulation. We represent in full detail the crucial parts of our proofs, for the other parts we provide the sketches and refer to the works where necessary details can be found. Due to the modular design, it suffices to check all the elementary protocols individually in order to understand and verify our resulting schemes. It should be noted that some of these protocols, such as the main lemma's argument and random weighting for t-s-tuples argument, are far from trivial and may have an independent application.

Although signatures and other cryptographic solutions using additional or more complex assumptions such as bilinear pairings may give better performance, we think that the efficient signatures constructed for the simplest prime-order group herein may be interesting in two aspects. First, they show in purely theoretical terms how much can be achieved on the simplest foundation. Second, just as the Bulletproofs protocol originally formulated for a prime-order group was later instantiated in a post-quantum setting using lattice hardness assumptions, we have some hope that something similar can be done for our protocols in the future.

ACKNOWLEDGEMENTS

Author thanks Tsz Hon Yuen et al. for the work [33], which inspired some ideas for improvement in the current paper, and also for our brief email conversation about cryptographic games and definitions. The author appreciate the comments provided by anonymous reviewers and is grateful to the readers who advised the better wording.

REFERENCES

[1] Masayuki Abe, Miyako Ohkubo, and Koutarou Suzuki. "1-out-of-n signatures from a variety of keys". In: *ASIACRYPT 2002*. Springer-Verlag. 2002, pp. 415–432.

[2] Thomas Attema and Ronald Cramer. *Compressed Σ-Protocol Theory and Practical Application to Plug & Play Secure Algorithmics*. Cryptology ePrint Archive, Paper 2020/152. 2020. URL: https://eprint.iacr.org/2020/152.

[3] Thomas Attema, Ronald Cramer, and Matthieu Rambaud. *Compressed Σ-Protocols for Bilinear Group Arithmetic Circuits and Application to Logarithmic Transparent Threshold Signatures*. Cryptology ePrint Archive, Paper 2020/1447. 2020. DOI: 10.1007/978-3-030-92068-5. URL: https://eprint.iacr.org/2020/1447.

[4] Thomas Attema et al. *Vector Commitments over Rings and Compressed Σ-Protocols*. Cryptology ePrint Archive, Paper 2022/181. 2022. URL: https://eprint.iacr.org/2022/181.

[5] Dan Boneh. *ZKP MOOC Lecture 5: The Plonk SNARK*. 2023. URL: `https://youtu.be/A0oZVEXav24?si=UfLt2z9R4e6Fv7GC`.

[6] Dan Boneh and Victor Shoup. *A Graduate Course in Applied Cryptography*. Dan Boneh's publications web page, `http://crypto.stanford.edu/~dabo/pubs/abstracts/bookShoup.html`. `https://toc.cryptobook.us/book.pdf`. 2020.

[7] Benedikt Bünz et al. "Bulletproofs: Short proofs for confidential transactions and more". In: *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2018, pp. 315–334.

[8] Dario Catalano and Dario Fiore. *Vector Commitments and their Applications*. Cryptology ePrint Archive, Paper 2011/495. 2011. URL: `https://eprint.iacr.org/2011/495`.

[9] Heewon Chung et al. *Bulletproofs+: Shorter Proofs for Privacy-Enhanced Distributed Ledger*. Cryptology ePrint Archive, Report 2020/735. `https://ia.cr/2020/735`. 2020.

[10] Richard A. Demillo and Richard J. Lipton. "A probabilistic remark on algebraic program testing". In: *Information Processing Letters* 7.4 (1978), pp. 193–195. DOI: `https://doi.org/10.1016/0020-0190(78)90067-4`.

[11] Ariel Gabizon, Zachary J. Williamson, and Oana Ciobotaru. *PLONK: Permutations over Lagrange-bases for Oecumenical Noninteractive arguments of Knowledge*. Cryptology ePrint Archive, Paper 2019/953. 2019. URL: `https://eprint.iacr.org/2019/953`.

[12] Adam Gibson. *From Zero (Knowledge) to Bulletproofs*. Github. 2022. URL: `https://github.com/AdamISZ/from0k2bp`.

[13] Brandon Goodell, Sarang Noether, and RandomRun. *Concise Linkable Ring Signatures and Forgery Against Adversarial Keys*. Cryptology ePrint Archive, Report 2019/654. `https://ia.cr/2019/654`. 2019.

[14] Sergey Gorbunov et al. *Pointproofs: Aggregating Proofs for Multiple Vector Commitments*. Cryptology ePrint Archive, Paper 2020/419. 2020. URL: `https://eprint.iacr.org/2020/419`.

[15] Jens Groth and Markulf Kohlweiss. "One-out-of-many proofs: Or how to leak a secret and spend a coin". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2015, pp. 253–280.

[16] Aram Jivanyan and Aaron Feickert. *Lelantus Spark: Secure and Flexible Private Transactions*. Cryptology ePrint Archive, Paper 2021/1173. `https://eprint.iacr.org/2021/1173`. 2021.

[17] Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg. "Constant-Size Commitments to Polynomials and Their Applications". In: *Advances in Cryptology - ASIACRYPT 2010*. Ed. by Masayuki Abe. Springer Berlin Heidelberg, 2010, pp. 177–194.

[18] Veronika Kuchta and Joseph K. Liu. *Non-Slanderability of Linkable Spontaneous Anonymous Group Signature (LSAG)*. Cryptology ePrint Archive, Paper 2021/1406. 2021. URL: `https://eprint.iacr.org/2021/1406`.

[19] Russell W. F. Lai. "Succinct Arguments: Constructions and Applications". doctoralthesis. Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), 2022.

[20] Russell W. F. Lai, Giulio Malavolta, and Viktoria Ronge. *Succinct Arguments for Bilinear Group Arithmetic: Practical Structure-Preserving Cryptography*. Cryptology ePrint Archive, Paper 2019/969. 2019. DOI: `10.1145/3319535.3354262`. URL: `https://eprint.iacr.org/2019/969`.

[21] Russell WF Lai et al. "Omniring: Scaling private payments without trusted setup". In: *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 2019, pp. 31–48.

[22] Benoît Libert and Moti Yung. "Concise Mercurial Vector Commitments and Independent Zero-Knowledge Sets with Short Proofs". In: *Theory of Cryptography*. Ed. by Daniele Micciancio. Springer Berlin Heidelberg, 2010, pp. 499–517.

[23] Joseph K Liu, Victor K Wei, and Duncan S Wong. "Linkable Spontaneous Anonymous Group Signature for Ad Hoc Groups (Extended Abstract)". In: *Proc. Ninth Australasian Conf. Information Security and Privacy (ACISP)*. 2004.

[24] Satoshi Nakamoto. *Bitcoin: A peer-to-peer electronic cash system*. `https://bitcoin.org/bitcoin.pdf`. 2008.

[25] Sarang Noether and Brandon Goodell. *Triptych: logarithmic-sized linkable ring signatures with applications*. Cryptology ePrint Archive, Report 2020/018. `https://ia.cr/2020/018`. 2020.

[26] Torben Pryds Pedersen. "Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing". In: *Advances in Cryptology — CRYPTO '91*. Ed. by Joan Feigenbaum. Springer Berlin Heidelberg, 1992, pp. 129–140.

[27] Claus-Peter Schnorr. "Efficient Signature Generation by Smart Cards". In: *J. Cryptology* 4.3 (1991), pp. 161–174.

[28] J. T. Schwartz. "Fast Probabilistic Algorithms for Verification of Polynomial Identities". In: *J. ACM* 27.4 (1980), pp. 701–717. DOI: 10.1145/322217.322225.

[29] Anton A. Sokolov. *Lin2-Xor Lemma and Log-size Linkable Threshold Ring Signature*. Cryptology ePrint Archive, Report 2020/688. https://ia.cr/2020/688. 2020.

[30] Patrick P. Tsang et al. *Separable Linkable Threshold Ring Signatures*. Cryptology ePrint Archive, Report 2004/267. https://ia.cr/2004/267. 2004.

[31] Nicolas Van Saberhagen. *CryptoNote v 2.0*. https://cryptonote.org/whitepaper.pdf. 2013.

[32] Tsz Hon Yuen et al. *RingCT 3.0 for Blockchain Confidential Transaction: Shorter Size and Stronger Security*. Tech. rep. Cryptology ePrint Archive, Report 2019/508, 2019. https://eprint.iacr.org/2019/508, 2019.

[33] Tsz Hon Yuen et al. *DualRing: Generic Construction of Ring Signatures with Efficient Instantiations*. Cryptology ePrint Archive, Paper 2021/1213. https://eprint.iacr.org/2021/1213. 2021.

[34] Richard Zippel. "Probabilistic algorithms for sparse polynomials". In: *Symbolic and Algebraic Computation*. Ed. by Edward W. Ng. Springer Berlin Heidelberg, 1979, pp. 216–226.

A PROOF OF 2-ELEMENT COMMITMENT

Proof: [Theorem 1] Completeness of the protocol can easily be seen from its code. Also, from the protocol code it is seen that, in the case if T is a direct weighted sum of $\{X, H\}$, then the protocol splits into two independent Schnorr identification schemes [27] with the same challenge. Thus, if this is the case, then the sHVZK and cWEE properties of the protocol in Figure 2 are proved the same way as for the Schnorr id scheme.

Suppose this is not the case, i.e., suppose prover sends T without knowing its relation to $\{X, H\}$ or, using shorthands, having $T \mathrel{!}= \text{lin}(X, H)$. Then, for the prover, if it has $Y = \text{lin}(X, H)$, then upon successful completion of the protocol it has $T = \text{lin}(X, H)$, which contradicts to the supposition. Otherwise, if $Y \mathrel{!}= \text{lin}(X, H)$ holds, then by rewinding the protocol and excluding T it obtains $Y = \text{lin}(X, H)$, which is a contradiction again.

Thus, we have shown that the sHVZK and cWEE properies of the protocol must hold. Formally, in full detail, these properties can be proved the same way as for the other Schnorr-like protocols in [2, 6, 9, 29].

Also, uniqueness of the witness (x, h) follows from the fact that Y is a Pedersen commitment, which is binding according to the definition in [7].

B PROOF OF VECTOR COMMITMENT

Proof: [Theorem 2] The protocol $\texttt{zkVC}_n$ in Figure 3 is a modified subset version of the Bulletproofs logarithmic inner product argument from [7]. There are the following three modifications to the inner product argument

- The inner product argument in [7] has no sHVZK property, we append this property to it the same way as it is done in [9], namely, by adding a blinding component to all transmitted elements. We omit providing a proof of sHVZK for our $\texttt{zkVC}_n$ protocol here; it is identical to the sHVZK proof for the improved inner product argument in [9].
- With the above modification, our $\texttt{zkVC}_n$ in Figure 3 is a subset case, namely, for $\mathbf{b} = \mathbf{0}^n$, of the inner product argument from [7] for the relation (6). Thus, our protocol is an argument for the relation (5).
- For the case $n = 1$, in $\texttt{zkVC}_n$ we use the custom zero-knowledge $\texttt{zk2ElemComm}$ protocol, which is complete, sHVZK, and has cWEE by Theorem 1.

Each of the above three modifications clearly does not override the completeness and cWEE properties of the Bulletproofs logarithmic inner product argument. Also, the first modification adds the sHVZK property. Thus, $\texttt{zkVC}_n$ in Figure 3 is a complete, sHVZK argument having cWEE for the relation (5).

Uniqueness of the witness $(\mathbf{a}, \alpha)$ follows from the fact that Y is a Pedersen vector commitment, which is binding.

C PROOF OF 3-TUPLE RANDOM WEIGHTING

Proof: [Theorem 3] The completeness and sHVZK properties of the `zk3ElemRW` protocol in Figure 4 follow from the fact that `zk3ElemRW` adds nothing to transcript of a protocol called in the last step, which in its turn is complete and sHVZK by the premise.

cWEE property of the `zk3ElemRW` protocol is also easy to establish, we do not provide a detailed proof here to save space, only the following sketch. In any case, `zk3ElemRW` is a subset case of our bigger argument shown in Figure 24, for which a detailed proof of cWEE can be found in the proof of Theorem 14.

The blinding generator H is orthogonal to all other generators by the premise, components proportional to H of all participating elements can be considered separately and be omitted in the main consideration.

In any case, first, for the related to H part of witness of the sub-protocol called in the last step, it suffices to calculate the factor $\hat{\alpha}$ as

$$\hat{\alpha} = \alpha + \delta_1\beta + \delta_2\gamma\,.$$

Second, witness extraction can be accomplished the same way as in the proof of the RandomWeighting-WEE lemma in [29].

Third, to ascertain that the witness a has only one possible value in this protocol, we can write Z, F, E as

$$\begin{cases} Z = z_P P + z_Q Q + z_R R \\ F = f_P P + f_Q Q + f_R R \\ E = e_P P + e_Q Q + e_R R \end{cases}, \tag{72}$$

since it is clear that, when H is already excluded from the consideration, the elements Z, F, E cannot have components outside the linear span of P, Q, R without breaking the DL assumption. Inserting the decomposition (72) into the equality $Y = aX$, we obtain

$$\operatorname{rank}\left(\begin{bmatrix} 1 & \delta_1 \text{ or } 0, \text{if } Q = 0 & \delta_2 \text{ or } 0, \text{if } R = 0 \\ z_P + \delta_1 f_P + \delta_2 e_P & z_Q + \delta_1 f_Q + \delta_2 e_Q & z_R + \delta_1 f_R + \delta_2 e_R \end{bmatrix}\right) < 2\,, \tag{73}$$

which immediately yields the sought relation, namely, that for some unique witness a, to the accuracy of H components, it holds

$$\begin{cases} Z = aP \\ F = aQ \\ E = aR \end{cases}.$$

Also, from the condition (73) it can be understood why we require for $P \neq 0 \wedge (Q \neq 0 \vee R \neq 0)$.

D PROOF OF SIMMETRIC VECTOR COMMITMENT

Proof: [Theorem 4] The protocol $\texttt{zkSVC}_{3,n}$ in Figure 5 adds nothing to transcript of a complete, sHVZK, and cWEE protocol called in its last step (it can be, say, $\texttt{zkVC}_n$), thus inheriting the sHVZK property from the latter. Completeness of the protocol $\texttt{zkSVC}_{3,n}$ is trivial. cWEE property of the protocol is easy to establish, the sketch follows.

First of all, we exclude H from all considerations for the same reason as in Appendix C. Then, because of orthogonality of all nonzero elements in $\mathbf{P} \cup \mathbf{Q} \cup \mathbf{R}$, each of the elements Z, F, and E decomposes into a weighted direct sum of $\mathbf{P}, \mathbf{Q}, \mathbf{R}$, respectively. Therefore, to prove the cWEE property of $\texttt{zkSVC}_{3,n}$ it suffices to prove cWEE for $\texttt{zkSVC}_{3,1}$.

In its turn, $\texttt{zkSVC}_{3,1}$ is equivalent to the protocol `zk3ElemRW` in Figure 4, hence $\texttt{zkSVC}_{3,1}$ has cWEE by Theorem 3. Thus we obtain cWEE for $\texttt{zkSVC}_{3,n}$.

Uniqueness of the witness $(\mathbf{a}, \alpha, \beta, \gamma)$ follows from the fact that each of Z, F, E is a Pedersen vector commitment, which is binding.

E PROOF OF LIN2-CHOICE LEMMA

Proof: [Theorem 5] Completeness and sHVZK of the $\texttt{zkLin2Choice}_n$ protocol are seen trivially from Figure 8. We exclude H from all considerations for the same reason as in Appendix C.

Let's prove the cWEE property of the $\texttt{zkLin2Choice}_n$ protocol by constructing a PPT witness extractor for it. In the last step of $\texttt{zkLin2Choice}_n$ there is a call to

$$\texttt{zkSVC}_{2,n}(\mathbf{P}, \mathbf{c} \circ \mathbf{Q}, H, Z, rF; \mathbf{a}, \alpha, \hat{\beta}),$$

and hence by Theorem 4 the following relation holds

$$\begin{cases} Z = \langle \mathbf{a}, \mathbf{P} \rangle \\ rF = \langle \mathbf{a}, \mathbf{c} \circ \mathbf{Q} \rangle \end{cases} , \tag{74}$$

where $\mathbf{a} \in \mathbb{F}_{\mathrm{p}}^n$ is obtained using $\mathtt{zkSVC}_{2,n}$ protocol witness extractor.

Thus, if $\mathbf{a}$ contains only one nonzero scalar, say, under index j, then the sought witness p is extracted together with the index s, namely, $p = a_j$, $s = j$. If $\mathbf{a} = \{0\}^n$ is the case, then the witness p is extracted as zero, the index s has no meaning.

Let's show that $\mathbf{a}$ cannot contain more than one nonzero scalar, otherwise the $\mathtt{zkLin2Choice}_n$ protocol witness extractor would be able to break the DL assumption. Suppose that $\mathbf{a}$ contains at least two nonzeros, a_j and a_k, under the indices j and k such that $j \neq k$. By writing Z and rF as weighted direct sums of $\mathbf{P}$ and $\mathbf{Q}$, respectively, we see that, according to the equalities (74), by unwinding the $\mathtt{zkSVC}_{2,n}$ call the extractor can obtain $\mathbf{a}$ such that the following two equalities hold for the known $Z, F, \mathbf{c}, r, \mathbf{a}$

$$Z = \sum_{i=0}^{n-1} a_i P_i , \tag{75}$$

$$rF = \sum_{i=0}^{n-1} a_i c_i Q_i , \tag{76}$$

where $r \neq 0$, otherwise the equality (76) would immediately produce a contradiction to $\mathrm{ort}(\mathbf{Q})$.

Let the extractor unwinds to the point where the challenges $\mathbf{c}$ were generated and resumes, thus obtaining new $\mathbf{c}', r', \mathbf{a}'$. Due to the equality (76), it holds $r' \neq 0$. Recalling $\mathrm{ort}(\mathbf{P})$, due to the equality (75), it holds $\mathbf{a}' = \mathbf{a}$. By excluding F from the equality (76) the extractor obtains

$$0 = \sum_{i=0}^{n-1} a_i \left(\frac{c_i}{r} - \frac{c'_i}{r'}\right) Q_i . \tag{77}$$

Since $\mathrm{ort}(\mathbf{Q})$ holds, all weights of Q_i's in the equality (77) must be zero, otherwise the extractor breaks the DL assumption. According to our supposition, $a_j \neq 0$ and $a_k \neq 0$, hence we write out two equations for the zero weights of Q_j and Q_k

$$\begin{cases} 0 = \frac{c_j}{r} - \frac{c'_j}{r'} \\ 0 = \frac{c_k}{r} - \frac{c'_k}{r'} \end{cases} , \tag{78}$$

where we have already performed division by nonzero a_j and a_k. As $r \neq 0$ and $r' \neq 0$, the system (78) reduces to

$$\frac{c_k}{c'_k} = \frac{c_j}{c'_j} , \tag{79}$$

which holds only with negligible probability. Therefore, if there is more than one nonzero element in $\mathbf{a}$, then the extractor with overwhelming probability obtains one or more nonzero weights of Q_i's in the equality (77). Thus, under our supposition, the extractor breaks the DL assumption by expressing Q_j through the elements of $\mathbf{Q} \setminus \{Q_j\}$, therefore our supposition is incorrect.

By this we have proved that the PPT extractor with overwhelming probability finds witness for the relation (15) and, thus, the protocol $\mathtt{zkLin2Choice}_n$ has cWEE.

As for uniqueness of witness (p, α), it trivially follows from subtracting two different decompositions of Z from each other and breaking the DL relation assumption, in the case if witness is not unique.

F SIGNATURE EFLRS1

Proof: [Theorem 6] As follows from Figure 11, EFLRS1 is a linkable ring signature by definition (we assume the `EFLRS1.Link` method is defined the usual way by matching key images, e.g., as in [23]).

All the listed properties 1. . . 8) of the EFLRS1 signature are proved by well-known methods, such as in [23, 13, 15, 29], which rely on the key image of the form of $x^{\pm 1}\mathcal{H}_{\mathbf{point}}(P)$ and on completeness, sHVZK, and cWEE of the underlying proving system. We do not describe these proofs here due to their volume; instead, we refer the interested reader to the referenced papers.

Anyway, as an example, here is a proof sketch of the property 2). Definition of the existential unforgeability against adaptive chosen message / public key attackers is provided in [23], it is also can be taken from [29]. In this sketch, for the sake of simplicity we combine the approaches introduced in [23, 15]. We will build a PPT master algorithm $\mathcal{M}$ that breaks the DL assumption by calling a PPT adversary $\mathcal{A}$ that forges EFLRS1.

Let $\mathcal{L}$ be a list of public keys of which each key is generated according to the description in [23] or, equivalently, according to the definition in [29]. Neither $\mathcal{M}$ nor $\mathcal{A}$ knows any of private keys for $\mathcal{L}$. First of all, $\mathcal{M}$ substitutes a new implementation for $\mathcal{H}_{\mathbf{point}}$, which for an input element L samples a random r and returns rL. This new $\mathcal{H}_{\mathbf{point}}$ implementation memorizes the sampled r's and, thus, remains deterministic and indistinguishable from the original $\mathcal{H}_{\mathbf{point}}$ outside $\mathcal{M}$.

Second, $\mathcal{M}$ simulates the signing oracle $\mathcal{SO}$ the following way. For an input ring $\mathbf{L} \subset \mathcal{L}$, it uniformly picks an index π and simulates signing with L_π. Without knowing private key x_π such that $L_\pi = x_\pi G$, it constructs key image as $I = rG$ using r memorized by $\mathcal{H}_{\mathbf{point}}$ for L_π. Thus, the $\texttt{zkLin2Choice}_n$ call (23) at the end of the simulated EFLRS1 takes the form

$$\texttt{zkLin2Choice}_n(\{L_i + \zeta \mathcal{H}_{\mathbf{point}}(L_i)\}_{i=0}^{n-1}, \mathbf{G}_{[:n]}, H, G + \zeta rG; \pi, \ldots, 0) .$$

Since $\texttt{zkLin2Choice}_n$ is sHVZK by Theorem 5, $\mathcal{M}$ builds a simulated transcript of it with back patching $\mathcal{H}_{\mathbf{scalar}}$. Namely, without knowing x_π, $\mathcal{M}$ uniformly samples the random oracle replies to be used as known-in-advance challenges in the signature simulation and feeds them to $\mathcal{SO}$. The latter builds corresponding random oracle queries using the fed replies and patches $\mathcal{H}_{\mathbf{scalar}}$ so that it returns these replies in response to the built queries. As a result, the simulated signature gets indistinguishable from a real one.

Then, $\mathcal{M}$ feeds $\mathcal{L}$, $\mathcal{SO}$, $\mathcal{H}_{\mathbf{point}}$, and $\mathcal{H}_{\mathbf{scalar}}$ to $\mathcal{A}$, letting the latter produce forgeries whose rings are not spotted in calls to $\mathcal{SO}$. Finally, starting with an arbitrary successfully forged transcript, $\mathcal{M}$ unwinds and forks it the necessary amount of times, thus building a transcript tree with successful forgeries as leaves. Since $\texttt{zkLin2Choice}_n$ has cWEE by Theorem 5, from this transcript tree $\mathcal{M}$ restores witness x_π that breaks the DL assumption for one of the public keys in $\mathcal{L}$.

That's the sketch. It misses the non-trivial part a full proof should posess that is about the implication from $\mathcal{A}$'s non-negligible probability of generating successful forgeries to $\mathcal{M}$'s non-negligible ability of building the forged transcript tree or a dynamic equivalent of it. Formal methods of proving this implication can be found, e.g., in [23, 15]. Besides, here is the following brief intuition for this in Appendix G.

G MASTER CAPABLE OF BUILDING FORGED TREE

Suppose, $\mathcal{A}$ produces forgeries with a non-negligible probability and, nevertheless, $\mathcal{M}$ has only a negligible probability of successfully building the forged tree. Then $\mathcal{M}$ is always able to start a new tree with a new forgery generated by $\mathcal{A}$, however it never succeeds in obtaining the necessary amount of successful leaves from $\mathcal{A}$. This means that since $\mathcal{M}$ rewinds, forks, and resumes $\mathcal{A}$, at some point of this process $\mathcal{M}$ always gets stuck in the situation that it has a successfully built subtree with forged leaves for the first fork with some challenges generated at that point, yet for one of its subsequent forks with other challenges from the same point $\mathcal{M}$ cannot complete building a forged subtree anymore.

This situation would not be possible if these forks were identical and completely independent, only reading different random tapes. Indeed, if they were, they would be indistinguishable from each other and, therefore, would have equal probabilities of success. However they are not, as being identical they still share the same instances of $\mathcal{SO}$ and simulated $\mathcal{H}_{\mathbf{point}}$, $\mathcal{H}_{\mathbf{scalar}}$. Now we will demonstrate how to convert $\mathcal{SO}$, $\mathcal{H}_{\mathbf{point}}$, $\mathcal{H}_{\mathbf{scalar}}$ to such a form that the forks become identical to each other. Thus we will informally prove that $\mathcal{M}$ does not fall into the above situation, and hence our assumption is not true, which means that $\mathcal{M}$ has a non-negligible probability of constructing the complete forged tree.

Apparently, the simulated $\mathcal{H}_{\mathbf{point}}$ is not a problem, as it is indistinguishable from the stateless deterministic function, and hence it can be kept as is. The only problem is $\mathcal{H}_{\mathbf{scalar}}$, which is back patched for some queries occured in $\mathcal{SO}$. To make $\mathcal{H}_{\mathbf{scalar}}$ look stateless deterministic, let it crash when an attempt is made to back patch it for a query it has already been called with before. This makes $\mathcal{H}_{\mathbf{scalar}}$ indistinguishable from a deterministic stateless function, unless it crashes. With this modification, the first executed fork of $\mathcal{A}$ always has a greater or equal chance of success than the subsequent forks, as the latter may crash when trying to patch queries made by the first one; if they do not crash, then all of them succeed in building their forged subtrees.

So, to avoid these crashes, let's make the following change to $\mathcal{SO}$. Let $\mathcal{SO}$ check each time before applying back patch to $\mathcal{H}_{\mathbf{scalar}}$ for a query to see if it will crash. If so, let $\mathcal{SO}$ uniformly resample the challenges and build the query again. The queries are linearly challenge-dependent, so the uniform challenge resampling changes the query as if the latter were resampled uniformly. Therefore, it would take no more than a polynomial number of resamplings to avoid the crashes at all. Thus, we have shown that $\mathcal{M}$ is capable of constructing a complete forged tree as soon as $\mathcal{A}$ produces forgeries with non-negligible probability.

H PROOF OF MULTIPLE VECTOR COMMITMENTS

Proof: [Theorem 7] As can be seen from Figure 13, the protocol $\texttt{zkMVC}_{l,n}$ adds nothing to the transcript of the protocol $\texttt{zkVC}_n$, thus inheriting the sHVZK property. Completeness of the protocol $\texttt{zkMVC}_{l,n}$ is clear. Let's prove the cWEE property of the protocol.

This time, to show an example, we will not exclude the generator H from our consideration. We append H to $\mathbf{X}$, obtaining the expanded vector $\bar{\mathbf{X}} \in \mathbb{G}^{n+1}$

$$\bar{\mathbf{X}} = \begin{bmatrix} \mathbf{X} \\ H \end{bmatrix} .$$

At the same time, we attach the vector of blinding factors $\boldsymbol{\alpha} \in \mathbb{F}_{\bar{p}}^{l}$ to the witness matrix $\mathfrak{a} \in \mathbb{F}_{\bar{p}}^{l \times n}$, and thus define the expanded witness matrix $\bar{\mathfrak{a}} \in \mathbb{F}_{\bar{p}}^{l \times (n+1)}$ as

$$\bar{\mathfrak{a}} = [\mathfrak{a} \;\; \boldsymbol{\alpha}] .$$

Also, we combine $\mathbf{a} \in \mathbb{F}_{\bar{p}}^{n}$ with $\alpha \in \mathbb{F}_{\bar{p}}$, and thus define $\bar{\mathbf{a}} \in \mathbb{F}_{\bar{p}}^{n+1}$

$$\bar{\mathbf{a}} = \begin{bmatrix} \mathbf{a} \\ \alpha \end{bmatrix} .$$

Extractor obtains $\bar{\mathbf{a}}$ by unwinding the $\texttt{zkVC}_n$ call. As a result, for each i-th column $\mathfrak{a}_{[:,i]}$ of the matrix $\mathfrak{a}$, the following equality holds

$$\bar{\mathbf{a}}_{[i]} = \boldsymbol{\xi}^{\intercal} \cdot \bar{\mathfrak{a}}_{[:,i]} . \tag{80}$$

The extractor repeats the unwinding l times with re-sampled challenges $\boldsymbol{\xi}$. This way the equality (80) repeated l times turns into a matrix equation with random matrix of size $l \times l$, from which the extractor recovers each i'th column $\bar{\mathfrak{a}}_{[:,i]}$, $i \in [0 \ldots n]$ of the matrix $\bar{\mathfrak{a}}$. Thus, the extractor recovers the sought witness $\bar{\mathfrak{a}}$.

As for uniqueness of the witness $(\mathfrak{a}, \boldsymbol{\alpha})$, it trivially follows from subtracting two different decompositions of $\mathbf{Y}$ from each other and, thus, breaking the DL relation assumption.

I PROOF OF THE PROPERTIES OF MANY-OUT-OF-MANY PROOF

Proof: [Theorem 8] Completeness and sHVZK of the $\texttt{zkLin2mChoice}_{n,l}$ protocol in Figure 14 are clear from its design. Let's prove the cWEE property of the protocol. We will consider H this time.

First, extractor uses the $\texttt{zkMVC}_{l,n}$ protocol extractor, which exists by Theorem 7, and restores witness $(\mathfrak{a}, \hat{\boldsymbol{\alpha}})$ from the $\texttt{zkMVC}_{l,n}$ call in the last step of $\texttt{zkLin2mChoice}_{n,l}$. After that, for every $k \in [0 \ldots l-1]$, it assigns

$$(\mathbf{a}, \hat{\alpha}) \leftarrow (\mathfrak{a}_{[k]}, \hat{\boldsymbol{\alpha}}_{[k]}) ,$$

and proceeds with the extraction using the $\texttt{zkLin2Choice}_n$ protocol extractor, which exists by Theorem 5, as though the values of $\mathbf{a}, \hat{\alpha}$ were obtained from $\texttt{zkVC}_n$ in the last step of $\texttt{zkLin2Choice}_n$. This way the extractor obtains witness (p, α), and maps it to k-th positions in $\mathbf{p}$ and $\boldsymbol{\alpha}$, respectively.

We have shown how the extractor restores witness $(\mathbf{p}, \boldsymbol{\alpha})$ for the relation (25) and, hence, the $\texttt{zkLin2mChoice}_{n,l}$ protocol has cWEE.

Uniqueness of the witness $(\mathbf{p}, \boldsymbol{\alpha})$ immediately follows from uniqueness of the witness (p, α) for the protocol $\texttt{zkLin2Choice}_n$ in Figure 8, which is by Theorem 5.

J SIGNATURE EFLRSL

J.1 EFLRSL FOR L=1

As can be seen from Figure 15, for $l = 1$, the EFLRSL protocol is equivalent to the EFLRS1 protocol in Figure 11, with the variables and calls renamed. The overwhelmingly nonzero multiplier ξ_0, which is applied simultaneously to the commitment and witness in the nested $\texttt{zkVC}_n$ call, doesn't distort the equivalence. Thus, by Theorem 6, for $l = 1$, all the properties listed in Theorem 9 hold.

J.2 EFLRSL FOR L $\geqslant$ 1

Proof: [Theorem 9] A proof for the case $l = 1$ is provided in Appendix J.1.

The EFLRSL protocol is a linkable threshold ring signature by-design, this can be seen from Figure 15. We assume the `EFLRSL.Link` method is defined the usual way, i.e., by matching key images.

All of the listed in Theorem 9 properties of the EFLRSL signature can be proved by assuming that any of them does not hold and reducing to the case of $l = 1$, that is, by inferring a contradiction to what has already been proved in Appendix J.1. The key image form $x^{\pm 1}\mathcal{H}_{\mathbf{point}}(P)$ along with the completeness, sHVZK, and cWEE properties of the underlying proving system make the reduction to the $l = 1$ case possible.

As an alternative method, it is also possible to prove the listed properties with the notion of non-slanderability using the techniques provided in [30, 13, 18], which we do not describe here due to their volume.

K PROOF OF SIMPLIFIED LIN2-2CHOICE LEMMA

Proof: [Theorem 10] Completeness and sHVZK properties of the $\mathtt{zkLin22sChoice}_{n,m}$ protocol in Figure 17 are clear. We exclude H from the consideration for the same reason as in Appendix C.

Let's prove the protocol cWEE property. In the last step of $\mathtt{zkLin22sChoice}_{n,m}$ there is a call to

$$\mathtt{zkSVC}_{3,n}\left(\begin{bmatrix}\mathbf{P}\\ \mathbf{V}\end{bmatrix}, \begin{bmatrix}\mathbf{c}_{[:n]} \circ \mathbf{Q}\\ \mathbf{0}^m\end{bmatrix}, \begin{bmatrix}\mathbf{0}^n\\ \mathbf{c}_{[n:]} \circ \mathbf{W}\end{bmatrix}, H, Z, rF, c_{n+t}E; \mathbf{a}, \alpha, \hat{\beta}, \hat{\gamma}\right) ,$$

and hence, by Theorem 4, the following relation holds

$$\begin{cases} Z &= \langle \mathbf{a}_{[:n]}, \mathbf{P}\rangle \quad + \quad \langle \mathbf{a}_{[n:]}, \mathbf{V}\rangle \\ rF &= \langle \mathbf{a}_{[:n]}, \mathbf{c}_{[:n]} \circ \mathbf{Q}\rangle \\ c_{n+t}E &= \qquad\qquad\qquad\quad \langle \mathbf{a}_{[n:]}, \mathbf{c}_{[n:]} \circ \mathbf{W}\rangle \end{cases} , \tag{81}$$

with the witness $\mathbf{a} \in \mathbb{F}_{\bar{\mathsf{p}}}^{n+m}$ restored by witness extractor of the $\mathtt{zkSVC}_{3,n}$ protocol.

Due to $\mathrm{ort}(\mathbf{P}, \mathbf{V}, \mathbf{Q}, \mathbf{W})$, having $Z = Z_P + Z_V$ according to the formula (34), the system (81) splits into two subsystems

$$\begin{cases} Z_P &= \langle \mathbf{a}_{[:n]}, \mathbf{P}\rangle \\ rF &= \langle \mathbf{a}_{[:n]}, \mathbf{c}_{[:n]} \circ \mathbf{Q}\rangle \end{cases} , \tag{82}$$

$$\begin{cases} Z_V &= \langle \mathbf{a}_{[n:]}, \mathbf{V}\rangle \\ c_{n+t}E &= \langle \mathbf{a}_{[n:]}, \mathbf{c}_{[n:]} \circ \mathbf{W}\rangle \end{cases} . \tag{83}$$

Each of the systems (82), (83) is similar to the system (74) and, therefore, by applying to each of them the same reasoning as in the proof of the cWEE property of the Lin2-Choice lemma in Appendix E, we obtain the following two equalities, respectively

$$Z_P = pP_s \ , \tag{84}$$

$$Z_V = vV_{n+\tilde{s}} \ , \tag{85}$$

where p and v are scalars known to prover, and $s, \tilde{s}$ are indices also known to it. (If $p = 0$ or $v = 0$, then respectively s or $\tilde{s}$ is undefined.)

One more detail, when obtaining the equality (84) from the subsystem (82), we take r as a response to the challenges $\mathbf{c}_{[:n]}$, whereas obtaining the equality (85) from the subsystem (83), we take c_{n+t} as the response to the challenges $\mathbf{c}_{[n:]}$.

If $v \neq 0$ and $\tilde{s} \neq t$, then the extractor breaks the DL assumption by establishing a linear relationship between at least two different elements from the orthogonal set $\mathbf{R}$, hence we let $\tilde{s} = t$ for $v \neq 0$ and write the equality (85) as

$$Z_V = vV_{n+t} \ . \tag{86}$$

Now, recalling that Z decomposes into the sum $Z = Z_P + Z_V$ by the formula (34) which is discussed in Section 7.1.1, the extractor comes to the conclusion that the restored by the formulas (84), (86) values of (p, v, s) are the sought witnesses for the relation (27). Thus, we have proved the cWEE property of $\mathtt{zkLin22sChoice}_{n,m}$.

As for uniqueness of witness (p, v, α), it trivially follows from subtracting two different decompositions of Z from each other and, thus, breaking the DL relation assumption.

L PROOF OF MULTIPLE SIMMETRIC VECTOR COMMITMENTS

Proof: [Theorem 11] As can be seen from Figure 18, the $\texttt{zkMSVC}_{l,3,n}$ protocol adds nothing to the transcript of the $\texttt{zkMVC}_{l,n}$ protocol, thus inheriting the sHVZK property. Completeness of the $\texttt{zkMSVC}_{l,3,n}$ protocol is clear from Figure 18. We exclude H from all considerations for the same reason as in Appendix C.

Let's prove the cWEE property of the protocol. Having unwound the $\texttt{zkMVC}_{l,n}$ call, extractor obtains a matrix $\mathfrak{a} \in \mathbb{F}_{\bar{\mathrm{p}}}^{l\times n}$ such that according to the relation (24)

$$\mathbf{Y} = \mathfrak{a} \cdot \mathbf{X} . \tag{87}$$

Thus, for each element $Y_j = \mathbf{Y}_{[j]}, j \in [0 \dots l-1]$, and for the corresponding row $\mathfrak{a}_{[j,:]}$ of the matrix $\mathfrak{a}$, it holds

$$Y_j = \mathfrak{a}_{[j,:]} \cdot \mathbf{X} . \tag{88}$$

At the same time, due to the equalities (88), the $\texttt{zkMVC}_{l,n}$ protocol can be viewed as l independent, except for the common challenges (δ_1, δ_2), instances of the $\texttt{zkSVC}_{3,n}$ protocol. Therefore, by Theorem 4, the restored by the extractor matrix $\mathfrak{a}$ is the sought witness.

Uniqueness of the witness is due to the same reasons as in Appendix H.

M PROOF OF LIN2-2CHOICE LEMMA

Proof: [Theorem 12] Completeness and sHVZK of the protocol $\texttt{zkLin22Choice}_{l,n,m}$ in Figure 19 are clear. Particularly, note that the vectors $\mathbf{F}$ and $\mathbf{E}$ do not reveal any information since their elements are blinded with H. We further exclude H from all considerations for the same reason as in Appendix C.

Let's prove the protocol cWEE property. In the last step of $\texttt{zkLin22Choice}_{l,n,m}$ there is a call to

$$\texttt{zkMSVC}_{l,3,(n+m)}\left(\begin{bmatrix}\mathbf{P}\\ \mathbf{V}\end{bmatrix}, \begin{bmatrix}\mathbf{c}_{[:n]} \circ \mathbf{Q}\\ \mathbf{0}^m\end{bmatrix}, \begin{bmatrix}\mathbf{0}^n\\ \mathbf{c}_{[n:]} \circ \mathbf{W}\end{bmatrix}, H, \mathbf{Z}, \mathbf{r} \circ \mathbf{F}, \mathbf{c}_{[n:(n+l)]} \circ \mathbf{E};\ \mathfrak{a}, \alpha, \hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\gamma}}\right),$$

and hence, by Theorem 11, the following system of equalities holds

$$\begin{cases} \mathbf{Z} & = \mathfrak{a} \cdot \begin{bmatrix}\mathbf{P}\\ \mathbf{V}\end{bmatrix} \\ \mathbf{r} \circ \mathbf{F} & = \mathfrak{a} \cdot \begin{bmatrix}\mathbf{c}_{[:n]} \circ \mathbf{Q}\\ \mathbf{0}^m\end{bmatrix} \\ \mathbf{c}_{[n:(n+l)]} \circ \mathbf{E} & = \mathfrak{a} \cdot \begin{bmatrix}\mathbf{0}^n\\ \mathbf{c}_{[n:]} \circ \mathbf{W}\end{bmatrix} \end{cases}, \tag{89}$$

where the matrix $\mathfrak{a} \in \mathbb{F}_{\bar{\mathrm{p}}}^{l\times(n+m)}$ is the witness restored by the $\texttt{zkMSVC}_{l,3,(n+m)}$ protocol extractor.

Furthermore, the system (89) is l systems of the form (81), with proper renaming, for each row $\mathfrak{a}_{[t,:]}, t \in [0 \dots l-1]$ of the matrix $\mathfrak{a}$. Namely, the system (89) is the following l systems

$$\begin{cases} Z_t & = \langle \mathfrak{a}_{[t,:n]}, \mathbf{P}\rangle + \langle \mathfrak{a}_{[t,n:]}, \mathbf{V}\rangle \\ r_t F_t & = \langle \mathfrak{a}_{[t,:n]}, \mathbf{c}_{[:n]} \circ \mathbf{Q}\rangle \\ c_{n+t} E_t & = \langle \mathfrak{a}_{[t,n:]}, \mathbf{c}_{[n:]} \circ \mathbf{W}\rangle \end{cases}, \tag{90}$$

for each $t \in [0 \dots l-1]$.

The $\texttt{zkLin22Choice}_{l,n,m}$ protocol in Figure 19 comprises, up to the point of calling $\texttt{zkMSVC}_{l,3,(n+m)}$ and with the appropriate renaming, l parallel instances of the protocol $\texttt{zkLin22sChoice}_{n,m}$ from Figure 17. Hence, given l parallel systems (90) for $t \in [0 \dots l-1]$, the extractor performs the same calculations as in Appendix K l times, for each t. This way it obtains l witnesses (p_t, v_t, s_t), $t \in [0 \dots l-1]$ for l instances of the relation (27). That is, for each extracted tuple (p_t, v_t, s_t), it holds

$$Z_t = p_t P_{s_t} + v_t V_t , \tag{91}$$

that means witnesses for the relation (37) are found and, hence, cWEE property of the $\texttt{zkLin22Choice}_{l,n,m}$ protocol is proven.

Uniqueness of the witness is due to the same reasons as in Appendix K.

N PROOF OF CLAIM ABOUT LIN2-2CHOICE PROTOCOL CALL

Proof: [Claim 1] By Theorem 12, the call

$$\texttt{zkLin22Choice}_{l,n,l}((\mathbf{X}, \mathbf{G}_{[:n]}, \mathbf{V}, \mathbf{G}_{[n:(n+l)]}, H, \mathbf{Z}; \dots)$$

in the last step of the EFLRSLWB scheme in Figure 22 proves the relation (37). That is, it has an extractor that restores unique witness for the relation.

Let's demonsrate that this call also proves that $\mathbf{v} = \mathbf{p}$ in the relation (37), where $\mathbf{X}, \mathbf{V}, \mathbf{Z}$ are defined according to the EFLRSLWB scheme. Copying their definitions from Figure 22 here

$$\begin{aligned}
\mathbf{X} &= \mathbf{P} - \{K\}^n + \zeta \mathbf{U} - \omega \mathbf{A}\,, \\
\mathbf{V} &= \{K\}^l + \omega \mathbf{A}^{\mathbf{tmp}} + \chi \hat{\mathbf{U}}\,, \\
\mathbf{Z} &= \{G\}^l + \zeta \mathbf{I} + \chi \mathbf{J}\,.
\end{aligned}$$

Suppose the opposite, i.e., that for some $k \in [0 \dots l-1]$ it holds that $v_k \neq p_k$. Then the $\texttt{zkLin22Choice}_{l,n,m}$ protocol witness extractor extracts $\mathbf{v}, \mathbf{p}$ and, according to the relation (37), for some index s_k, holds

$$G + \zeta I_k + \chi J_k = p_k(P_{s_k} - K + \zeta U_{s_k} - \omega A_{s_k}) + v_k(K + \omega A_k^{\mathbf{tmp}} + \chi \hat{U}_k)\ . \tag{92}$$

Note that we omit showing the H component for the same reason as in Appendix C. However, it is always implied present, and the factor of H is implied extracted by the extractor for this and for the following equalities. Method of this extraction is straightforward.

Moving the K component to the left-hand side of the (92) equality the extractor gets

$$(p_k - v_k)K = -G - \zeta I_k - \chi J_k + p_k(P_{s_k} + \zeta U_{s_k} - \omega A_{s_k}) + v_k(\omega A_k^{\mathbf{tmp}} + \chi \hat{U}_k)\ , \tag{93}$$

that is, it expresses K as a linear combination (93) of $G, I_k, J_k, P_{s_k}, U_{s_k}, A_{s_k}, A_k^{\mathbf{tmp}}, \hat{U}_k, H$. However, according to the EFLRSLWB scheme, all these elements are a part of the pre-image of K and, hence, K is orthogonal to all of them. Thus, under the supposition $\mathbf{v} \neq \mathbf{p}$ the extractor breaks the DL assumption, which is impossible. Therefore, the supposition is incorrect and the following holds

$$\mathbf{v} = \mathbf{p}\,. \tag{94}$$

Using the equality (94), the equality (92) rewrites as

$$G + \zeta I_k + \chi J_k = p_k(\, P_{s_k} + \zeta U_{s_k} + \chi \hat{U}_k + \omega(A_k^{\mathbf{tmp}} - A_{s_k})\,)\ . \tag{95}$$

Note that in the equality (95) the following holds for p_k's

$$p_k \neq 0 \ \text{ for each } k \in [0 \dots l-1]\,. \tag{96}$$

In fact, $p_k = 0$ for some k requires that the left-hand side of the equality (95) be equal to zero, however the left-hand side contains nonzero element G alongside with the randomly weighted elements I_k, J_k, and, hence, there is only negligible probability for it to be equal to zero. The implicit presence of H component in the equality (95) does not change the case; if the assertion (96) does not hold then the extractor breaks the DL assumption.

All elements in the right-hand part of the relation (95), namely, $P_{s_k}, U_{s_k}, A_k^{\mathbf{tmp}}, A_{s_k}, H$, are in the pre-image of $\hat{U}_k$. Thus, $\hat{U}_k$ is orthogonal to all of them and, hence, due to the random weighting by χ, to the accuracy of H, the following equality holds

$$G + \zeta I_k = p_k(\, P_{s_k} + \zeta U_{s_k} + \omega(A_k^{\mathbf{tmp}} - A_{s_k})\,)\ . \tag{97}$$

In other words, the equality (97) follows from the equality (95) by Theorem 3, where the triplets are taken as

$$(\, P_{s_k} + \zeta U_{s_k} + \omega(A_k^{\mathbf{tmp}} - A_{s_k}),\ \hat{U}_k,\ 0\,) \ \text{ and } \ (\, G + \zeta I_k,\ J_k,\ 0\,)\ .$$

Suppose that $(A_k^{\mathbf{tmp}} - A_{s_k}) \neq 0$. By unwinding and resuming the $\texttt{zkLin22Choice}_{l,n,l}$ call with different ω' the extractor obtains different p'_k and, by subtracting two instances of the equality (97) from each other, obtains

$$0 = p_k(\, P_{s_k} + \zeta U_{s_k} + \omega(A_k^{\mathbf{tmp}} - A_{s_k})\,) - p'_k(\, P_{s_k} + \zeta U_{s_k} + \omega'(A_k^{\mathbf{tmp}} - A_{s_k})\,)\ ,$$

which rewrites as

$$(p'_k - p_k)(P_{s_k} + \zeta U_{s_k}) = (p_k\omega - p'_k\omega')(A_k^{\mathbf{tmp}} - A_{s_k})\ . \tag{98}$$

Due to the orthogonality of P_{s_k} and U_{s_k} in the EFLRSLWB scheme, it holds

$$(P_{s_k} + \zeta U_{s_k}) \neq 0 .$$

If $p'_k = p_k$, then the left-hand side of the equality (98) is zero and, hence, $\omega' = \omega$ that holds only with negligible probability. So, with overwhelming probability $p'_k \neq p_k$ and the extractor divides the equality (98) by $(p'_k - p_k)$, calculating scalar factor a as follows

$$P_{s_k} + \zeta U_{s_k} = a\,(A_k^{\mathbf{tmp}} - A_{s_k})\,, \text{ where } a = \frac{p_k\omega - p'_k\omega'}{p'_k - p_k}\,. \tag{99}$$

Unwinding and resuming the $\texttt{zkLin22Choice}_{l,n,l}$ call with different ζ' a couple of times, the extractor calculates factor a' such that

$$P_{s_k} + \zeta' U_{s_k} = a'\,(A_k^{\mathbf{tmp}} - A_{s_k})\,. \tag{100}$$

By subtracting the equality (99) from the equality (100) and dividing by $(\zeta' - \zeta)$, which is nonzero with overwhelming probability, the extractor obtains

$$U_{s_k} = \frac{a' - a}{\zeta' - \zeta}\,(A_k^{\mathbf{tmp}} - A_{s_k})\,. \tag{101}$$

Also, it obtains from the equalities (99) and (101)

$$P_{s_k} = \left(a - \zeta\,\frac{a' - a}{\zeta' - \zeta}\right)\,(A_k^{\mathbf{tmp}} - A_{s_k})\,. \tag{102}$$

After that, as $U_{s_k} \neq 0$ and, hence, $(a' - a) \neq 0$ in the equality (101), the extractor expresses $(A_k^{\mathbf{tmp}} - A_{s_k})$ through P_{s_k} in (101) and inserts $(A_k^{\mathbf{tmp}} - A_{s_k})$ into the equality (102), thus obtaining

$$P_{s_k} = \left(a - \zeta\,\frac{a' - a}{\zeta' - \zeta}\right)\,\frac{\zeta' - \zeta}{a' - a}\,U_{s_k}\,. \tag{103}$$

Recalling P_{s_k} and U_{s_k} are orthogonal to each other, the extractor breaks the DL assumption with the equality (103); thus the supposition is wrong and the following holds

$$A_k^{\mathbf{tmp}} = A_{s_k}\,. \tag{104}$$

In accordance with the equality (104), the equality (97) which is obtained by the extractor after unwinding the $\texttt{zkLin22Choice}_{l,n,l}$ call, rewrites as

$$G + \zeta I_k = p_k(P_{s_k} + \zeta U_{s_k})\,, \tag{105}$$

where p_k is known to the extractor. Thus, the $\texttt{zkLin22Choice}_{l,n,l}$ call is an argument having cWEE property for the relation (106). The witness p_k is unique, as the opposite breaks the DL assumption between P_{s_k} and $U_{s_k} = \boldsymbol{\mathcal{H}}_{\mathbf{point}}(P_{s_k})$ in the equality (105).

At the same time, according to the obtained by the extractor equality (104), the same $\texttt{zkLin22Choice}_{l,n,l}$ call is an argument having cWEE for the relation (107) for the same s_k, which implies the same $\mathbf{s}$ for the both relations. Completeness and sHVZK of the $\texttt{zkLin22Choice}_{l,n,l}$ call follow from Theorem 12.

Uniqueness of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ is trivially seen, as the opposite breaks the DL relation assumption. Claim 1 is proven.

O SIGNATURE EFLRSLWB FOR L ⩾ 1

Proof: [Theorem 13] We first make the following claim.

Claim 1:
The call to $\texttt{zkLin22Choice}_{l,n,l}$ in the last step of the EFLRSLWB scheme in Figure 22 is a complete, sHVZK argument having cWEE for the relation (25) with appropriate input renaming, i.e., for the relation

$$\mathcal{R} = \left\{ \begin{array}{l} (\mathbf{P} + \zeta\mathbf{U}),\ \mathbf{G}_{[:n]} \in \mathbb{G}^{n*}, H \in \mathbb{G}^*,\ (\{G\}^l + \zeta\mathbf{I}) \in \mathbb{G}^l; \\ \mathbf{s} \in [0 \dots n-1]^l, \mathbf{p}, \boldsymbol{\alpha} \in \mathbb{F}_{\mathfrak{p}}^l \end{array} \middle| \begin{array}{l} \forall k \in [0 \dots l-1]: \\ G + \zeta I_k = p_k(P_{s_k} + \zeta U_{s_k}) + \alpha_k H \end{array} \right\} \tag{106}$$

with unique witness $(\mathbf{p}, \boldsymbol{\alpha})$, and is also a complete, sHVZK argument having cWEE for the relation

$$\mathcal{R}' = \left\{ \begin{array}{l} \mathbf{A} \in \mathbb{G}^n,\ \mathbf{A^{tmp}} \in \mathbb{G}^l,\ H \in \mathbb{G}^*; \\ \mathbf{s} \in [0 \dots n-1]^l,\ \boldsymbol{\beta} \in \mathbb{F}_{\mathfrak{p}}^l \end{array} \middle| \begin{array}{l} \forall k \in [0 \dots l-1]: \\ A_k^{\mathbf{tmp}} = A_{s_k} + \beta_k H \end{array} \right\} \tag{107}$$

with unique witness $\boldsymbol{\beta}$, such that the private input $\mathbf{s}$ is common for both relations (106) and (107).

Proof: is in Appendix N.

Note that the vectors $\mathbf{A^{tmp}}$ and $\mathbf{J}$ in Figure 22 are indistinguishable from white noise, because all their elements contain independent blinding components with randomized factors from, respectively, $\boldsymbol{\mu}$ and $\boldsymbol{\upsilon}$.

The Claim 1 asserts that in the last step of the EFLRSLWB scheme there is a call to the complete, sHVZK, and having cWEE proving system $\mathtt{zkLin22Choice}_{l,n,l}$ that produces a proof of the relation (106), which is actually the relation (25) with proper renaming. Also, as we can see in Figure 22, all previous steps of the EFLRSLWB scheme do all the play of the EFLRSL scheme from Figure 15 up to the proof of the relation (25). As for the vectors $\mathbf{A^{tmp}}$ and $\mathbf{J}$ which are all indistinguishable from white noise, they can be discarded as uninfluential when considering the relation (106). Thus, we see that the EFLRSLWB scheme is the EFLRSL scheme with the substituted underlying proving system, which is also complete, sHVZK, and having cWEE.

Therefore, the EFLRSLWB scheme is a linkable threshold ring signature with the properties 1. . . 8), which hold due to exactly the same reasons as the properties 1. . . 8) of the EFRLSL scheme in Theorem 9.

The property 9) comes as a result of calling $\mathtt{zk2ElemComm}$ in the last step of the EFLRSLWB scheme. By Theorem 1, it holds

$$A^{\mathbf{sum}} = \sum_{k=0}^{l-1} \mathbf{A}^{\mathbf{tmp}}_{[k]} + f_H H + f_D D \ , \tag{108}$$

where f_H, f_D are scalars known to prover. At the same time, by Claim 1 according to the relation (107), the equality (108) unfolds as

$$A^{\mathbf{sum}} = \sum_{k=0}^{l-1} A_{s_k} + \left(f_H + \sum_{k=0}^{l-1} \beta_k \right) H + f_D D \ . \tag{109}$$

Recalling that according to the EFLRSLWB scheme the generator H is an $\mathcal{H}_{\mathbf{point}}$ image of the $A^{\mathbf{sum}}, \mathbf{A}, D$ elements, the equality (109) reduces to

$$A^{\mathbf{sum}} = \sum_{k=0}^{l-1} A_{s_k} + f_D D \ ,$$

which is exactly what the property 9) is. Theorem 13 is proven.

P PROOF OF RANDOM WEIGHTING FOR T-S-TUPLES

Proof: [Theorem 14] Completeness and sHVZK properties of the $\mathtt{zkTElemRW}_{t,s}$ protocol are trivially seen from Figure 24. Turning to the cWEE property, we start with the following claim.

Claim 2:
Under the conditions of Theorem 14, if a PPT witness extractor for the protocol $\mathtt{zkTElemRW}_{t,s}$ in Figure 24 extracts two different values of the factor a in the relation $Y = aX + \hat{\alpha}H$ for two different random challenge sets $(\boldsymbol{\delta}, \boldsymbol{\sigma})$ in the last step of the protocol, then a PPT algorithm that breaks the DL relation assumption can be constructed.

Proof: is in Appendix Q.

Having the Claim 2 proved, let's construct a witness extractor for $\mathtt{zkTElemRW}_{t,s}$. The extractor restores the factors $(a, \hat{\alpha})$ in the equality

$$Y = aX + \hat{\alpha}H \ \text{ in Figure 24.} \tag{110}$$

According to Figure 24, the equality (110) itself represents the relation (4) with the renamed entries. To accomplish the extraction, the extractor uses the cWEE property of the protocol that proves the relation (4) in the last step of $\mathtt{zkTElemRW}_{t,s}$. Namely, it uses another witness extractor which extracts witness for (4).

By inserting into the equality (110) X, Y defined a step above in Figure 24 and moving aX to the left-hand side, the extractor obtains

$$(Z - aP) + \langle \boldsymbol{\delta}, \mathbf{F} - a\mathbf{Q} \rangle - \langle \boldsymbol{\sigma}, a\mathbf{S} \rangle = \hat{\alpha}H \ . \tag{111}$$

By unwinding and running the $\mathtt{zkTElemRW}_{t,s}$ protocol $(t + s)$ more times with different $\boldsymbol{\delta}, \boldsymbol{\sigma}$, the extractor gets, in sum, $(t + s + 1)$ equalities of type (111), which have common $Z, P, \mathbf{F}, \mathbf{Q}, \mathbf{S}, H, a$ and different $\boldsymbol{\delta}, \boldsymbol{\sigma}, \hat{\alpha}$. The factor a is common to all of them, as the opposite breaks the DL relation assumption by Claim 2. The extractor writes down all these $(t + s + 1)$ equalities in a matrix form, as follows,

$$\mathfrak{a} \cdot \mathbf{B} = \hat{\boldsymbol{\alpha}} H, \tag{112}$$

where

$$\mathfrak{a} = \begin{bmatrix} 1 & \delta_{0,0} & \dots & \delta_{(t-1),0} & \sigma_{0,0} & \dots & \sigma_{(s-1),0} \\ 1 & \delta_{0,1} & \dots & \delta_{(t-1),1} & \sigma_{0,1} & \dots & \sigma_{(s-1),1} \\ 1 & \delta_{0,2} & \dots & \delta_{(t-1),2} & \sigma_{0,2} & \dots & \sigma_{(s-1),2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \delta_{0,(t+s)} & \dots & \delta_{(t-1),(t+s)} & \sigma_{0,(t+s)} & \dots & \sigma_{(s-1),(t+s)} \end{bmatrix}, \ \mathbf{B} = \begin{bmatrix} Z - aP \\ F_0 - aQ_0 \\ \vdots \\ F_{t-1} - aQ_{t-1} \\ -aS_0 \\ \vdots \\ -aS_{s-1} \end{bmatrix}, \ \hat{\boldsymbol{\alpha}} = \begin{bmatrix} \hat{\alpha}_0 \\ \hat{\alpha}_1 \\ \hat{\alpha}_2 \\ \vdots \\ \hat{\alpha}_{t+s} \end{bmatrix}. \quad (113)$$

Then, it solves the matrix equation (112) for **B**. Taking into account that $\mathfrak{a}$ is composed of uniformly random scalars together with the first column of 1's and, hence, with overwhelming probability $\det(\mathfrak{a}) \neq 0$, it expresses each element of **B** as H multiplied by a corresponding scalar from the vector $\mathfrak{a}^{-1} \cdot \hat{\boldsymbol{\alpha}}$

$$\mathbf{B} = \mathfrak{a}^{-1} \cdot \hat{\boldsymbol{\alpha}} H . \quad (114)$$

Now, let us show that the witness a in the relation (62), which is fed at $\mathcal{P}$'s private input, is equal to the factor a restored for the equality (110), which is just found by the extractor and used in the definition of **B** in (113). Suppose the opposite, then here is an algorithm that breaks the DL relation assumption, it looks as follows.

It honestly runs $\texttt{zkTElemRW}_{t,s}$ knowing the input $a, \alpha, \boldsymbol{\beta}, \boldsymbol{\gamma}$, which is the witness for the relation (62). Then, it extracts a different a for the equality (110). Then, the breaker algorithm takes the equality for the first element of **B** in (114) and the equality for Z in (62). Eliminating Z from the both, keeping in mind the multipliers of P are different in them, the breaker expresses P through H and, thus, breaks the premise $P \mathrel{!=} \mathrm{lin}(\mathrm{nz}(\mathbf{Q}) \cup \mathrm{nz}(\mathbf{S}) \cup \{H\})$.

Thus, we have proved the witness a found by the extractor is the sought witness part a for the relation (62). Finally, it is easy to see how the extractor can restore the blinding factor $\alpha, \boldsymbol{\beta}, \boldsymbol{\gamma}$ component of the witness in (62). That is, it puts the $(t + s + 1)$ blinding factors $\alpha, \boldsymbol{\beta}, \boldsymbol{\gamma}$ together into a vector and calculates them from (114), (113), (62) as

$$\begin{bmatrix} \alpha \\ \beta_0 \\ \vdots \\ \beta_{t-1} \\ \gamma_0 \\ \vdots \\ \gamma_{s-1} \end{bmatrix} = \mathfrak{a}^{-1} \cdot \hat{\boldsymbol{\alpha}} .$$

We have built an extractor that finds the witness $(a, \alpha, \boldsymbol{\beta}, \boldsymbol{\gamma})$ for the relation (62). Uniqueness of a is already proved by Claim 2. Uniqueness of $\alpha, \boldsymbol{\beta}, \boldsymbol{\gamma}$ is trivial, as the opposite breaks the DL assumption. Thus, Theorem 14 is proved.

Q PROOF OF CLAIM ABOUT THE SAME FACTOR

Proof: [Claim 2] This proof is going to be a bit nontrivial, so, for the first, let's understand how the witness a in the equality (110) extracted in the last step of the protocol $\texttt{zkTElemRW}_{t,s}$ in Figure 24 depends on the challanges. We keep in mind a is a witness for the relation (4) which is represented by the equality (110).

For convenience, we rewrite the equality (110) in the matrix form, as follows, using the formulas (57), (58), (59), (60), (61), assuming $\boldsymbol{\xi}$ is a row vector, and **T**, **D** are column vectors

$$\boldsymbol{\xi} \cdot \mathbf{D} = a\boldsymbol{\xi} \cdot \mathbf{T} + \hat{\alpha} H . \quad \text{Note, this equality represents the relation (4).} \quad (115)$$

Let the extractor perform $(t + s + 1)$ rewindings and, thus, let it have $(t + s + 1)$ instances of the relation (115) for $(t + s + 1)$ instances of the challange vector $\boldsymbol{\xi}$. The extractor puts these $(t + s + 1)$ instances of $\boldsymbol{\xi}$ into the matrix

$$\mathfrak{a} = \begin{bmatrix} \boldsymbol{\xi}_0 \\ \boldsymbol{\xi}_1 \\ \boldsymbol{\xi}_2 \\ \vdots \\ \boldsymbol{\xi}_{(t+s)} \end{bmatrix} = \begin{bmatrix} 1 & \delta_{0,0} & \dots & \delta_{(t-1),0} & \sigma_{0,0} & \dots & \sigma_{(s-1),0} \\ 1 & \delta_{0,1} & \dots & \delta_{(t-1),1} & \sigma_{0,1} & \dots & \sigma_{(s-1),1} \\ 1 & \delta_{0,2} & \dots & \delta_{(t-1),2} & \sigma_{0,2} & \dots & \sigma_{(s-1),2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \delta_{0,(t+s)} & \dots & \delta_{(t-1),(t+s)} & \sigma_{0,(t+s)} & \dots & \sigma_{(s-1),(t+s)} \end{bmatrix}. \quad (116)$$

Since $\mathfrak{a}$ is a random matrix, with overwhelming probability it holds that $\det(\mathfrak{a}) \neq 0$ and, thus, $\mathfrak{a}$ is a basis in the $(t+s+1)$-dimensional scalar vector challenge space. Also, let the extractor map the corresponding $(t+s+1)$ witness pairs $(a, \hat{\alpha})$ extracted in the last step of the protocol into the following two vectors

$$\mathbf{a} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{t+s} \end{bmatrix} , \quad \hat{\boldsymbol{\alpha}} = \begin{bmatrix} \hat{\alpha}_0 \\ \hat{\alpha}_1 \\ \hat{\alpha}_2 \\ \vdots \\ \hat{\alpha}_{t+s} \end{bmatrix} , \tag{117}$$

and rewrite $(t+s+1)$ instances of the equality (115) for these vectors in the matrix form, as follows,

$$\mathfrak{a} \cdot \mathbf{D} = \mathrm{diag}(\mathbf{a}) \cdot \mathfrak{a} \cdot \mathbf{T} + \hat{\boldsymbol{\alpha}} H , \quad \text{where } \mathrm{diag}(\mathbf{a}) = \begin{bmatrix} a_0 & & 0 \\ & \ddots & \\ 0 & & a_{t+s} \end{bmatrix} . \tag{118}$$

Let the extractor rewind one more time and obtain $(a', \hat{\alpha}')$ for a new challenge vector $\boldsymbol{\xi}'$. The matrix $\mathfrak{a}$ is a basis in the challenge space, so $\boldsymbol{\xi}'$ decomposes by it. Denote the corresponding row vector of weights as $\mathbf{b}$ such that

$$\boldsymbol{\xi}' = \mathbf{b} \cdot \mathfrak{a} . \tag{119}$$

Next, multiplying the decomposition (119) by $\mathbf{D}$ and unfolding both sides of it using the formulas (115) and (118), respectively, the extractor obtains the following equality

$$(a'\boldsymbol{\xi}' - \mathbf{b} \cdot \mathrm{diag}(\mathbf{a}) \cdot \mathfrak{a}) \cdot \mathbf{T} = (\mathbf{b} \cdot \hat{\boldsymbol{\alpha}} - \hat{\alpha}')\, H . \tag{120}$$

Recalling that by the definition (57) $\mathbf{T}$ is a column vector of $\{P\} \cup \mathbf{Q} \cup \mathbf{S}$, the equality (120) takes on the meaning of a decomposition of 0 into a weighted sum of $\{P\} \cup \mathbf{Q} \cup \mathbf{S} \cup \{H\}$ with known to the extractor weights. In the case if the weight of P in (120) is nonzero, the extractor obtains weights for the decomposition $P = \mathrm{lin}(\mathrm{nz}(\mathbf{Q}) \cup \mathrm{nz}(\mathbf{S}) \cup \{H\})$, which contradicts to the premise of the Theorem 14.

Namely, if the weight of P in (120) is nonzero, then the extractor has a known decomposition of P by $\mathbf{Q} \cup \mathbf{S} \cup \{H\}$ and thus breaks the DL relation assumption. Therefore, the weight of P in (120) must be zero. The extractor calculates it from (120) using (116), (61), (117) as

$$0 = a' - \langle \mathbf{b}, \mathbf{a} \rangle .$$

This way, the extractor obtains the following transformation rule for the witness a depending on the challenge vector $\boldsymbol{\xi}'$

$$a' = \langle \mathbf{b}, \mathbf{a} \rangle , \quad \text{where } \mathbf{b} = \boldsymbol{\xi}' \cdot \mathfrak{a}^{-1}. \tag{121}$$

Note, the vector $\mathbf{b}$ in the rule (121), as well as in the formulas (119), (120), meets the condition $\left\langle \mathbf{b}, \{1\}^{t+s+1} \right\rangle = 1$, which guarantees that 1 is always at the first position in $\boldsymbol{\xi}'$.

To sum up, the rule (121) states the following. If the extractor already has a challenge space base defined by matrix $\mathfrak{a}$, and if it also has the corresponding witnesses collected in vector $\mathbf{a}$, then, for any new random vector $\boldsymbol{\xi}'$, value of the newly extracted witness a' is equal to the value defined by the formula (121). Otherwise, if the extractor gets a value for a' other than (121), then it breaks the DL relation assumption.

Now, let the extractor perform $(t+s+1)$ more rewindings and, thus, let it obtain another challenge space base

$$\mathfrak{c} = \begin{bmatrix} \boldsymbol{\xi}'_0 \\ \boldsymbol{\xi}'_1 \\ \boldsymbol{\xi}'_2 \\ \vdots \\ \boldsymbol{\xi}'_{(t+s)} \end{bmatrix} = \begin{bmatrix} 1 & \delta'_{0,0} & \dots & \delta'_{(t-1),0} & \sigma'_{0,0} & \dots & \sigma'_{(s-1),0} \\ 1 & \delta'_{0,1} & \dots & \delta'_{(t-1),1} & \sigma'_{0,1} & \dots & \sigma'_{(s-1),1} \\ 1 & \delta'_{0,2} & \dots & \delta'_{(t-1),2} & \sigma'_{0,2} & \dots & \sigma'_{(s-1),2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \delta'_{0,(t+s)} & \dots & \delta'_{(t-1),(t+s)} & \sigma'_{0,(t+s)} & \dots & \sigma'_{(s-1),(t+s)} \end{bmatrix} . \tag{122}$$

Note, the equality (115) holds as well for the new $(t+s+1)$ instances of the challange vector $\boldsymbol{\xi}'$ written as rows of the matrix $\mathfrak{c}$. By this, the transition matrix between the bases $\mathfrak{a}$ and $\mathfrak{c}$ is

$$\mathfrak{b} = \begin{bmatrix} \mathbf{b}_0 \\ \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_{(t+s)} \end{bmatrix} = \mathfrak{c} \cdot \mathfrak{a}^{-1} , \quad \text{where each row } \mathbf{b}_i \text{ is a weight vector of } \boldsymbol{\xi}'_i\text{'s decomposition by } \boldsymbol{\xi}_i\text{'s.} \tag{123}$$

Note that for $\mathfrak{b}$ there always holds $\{1\}^{t+s+1} = \mathfrak{b} \cdot \{1\}^{t+s+1}$, as the first columns in $\mathfrak{a}$ and $\mathfrak{c}$ are equal to $\{1\}^{t+s+1}$.

Apparently, as $\mathfrak{c}$ is a random matrix, with overwhelming probability it holds that $\det(\mathfrak{c}) \neq 0$ and, hence, $\det(\mathfrak{b}) \neq 0$. And, by the definition (123), $\mathfrak{c} = \mathfrak{b} \cdot \mathfrak{a}$ holds. The witness transformation rule (121) written in the matrix form for the base vectors in $\mathfrak{c}$ becomes

$$\mathbf{a}' = \mathfrak{b} \cdot \mathbf{a}\,, \quad \text{where } \mathfrak{b} = \mathfrak{c} \cdot \mathfrak{a}^{-1}. \tag{124}$$

Looking closer at $\mathfrak{b}$ and $\mathbf{a}$ we make the following three simple claims about their items distributions. Hereinafter, for any two scalar sets $\mathbf{x}$ and $\mathbf{y}$, we say $\mathbf{x}$ is considered in isolation of $\mathbf{y}$ if neither direct nor indirect dependencies or correlates of $\mathbf{y}$, except for maybe $\mathbf{x}$ itself, are involved in the consideration of $\mathbf{x}$.

Claim 3:
For any two random bases $\mathfrak{a}$ and $\mathfrak{c}$ defined by the formulas (116) and (122), respectively, with all their items picked independently and uniformly at random, except for the items in the first columns which are 1's, the transition matrix $\mathfrak{b}$ defined by (123) and considered in isolation of $\mathfrak{c}$ has the following two properties for its items

a) all items in $\mathfrak{b}$ are distributed uniformly

b) for each row $\mathbf{b}_i \in \mathfrak{b}$, there are $(t+s)$ independent items and one dependent item in it. The dependency is determined by the equality

$$\left\langle \mathbf{b}_i, \{1\}^{t+s+1} \right\rangle = 1\,. \tag{125}$$

Proof: is in Appendix R.1.

Claim 4:
If the extractor knows two randomly sampled challenge vectors $\dot{\boldsymbol{\xi}}, \ddot{\boldsymbol{\xi}}$ along with the corresponding witnesses $\dot{a}, \ddot{a}$ such that $\dot{\boldsymbol{\xi}} \neq \ddot{\boldsymbol{\xi}}, \dot{a} \neq \ddot{a}$, and the equality (115) holds for them, then, for any new random base $\mathfrak{a}$ constructed by the formula (116), the corresponding witness vector $\mathbf{a}$ built by (117) and considered in isolation of $\mathfrak{a}$ has all witnesses in it distributed independently and uniformly.

Proof: is in Appendix R.2.

Claim 5:
For any arithmetic expression which contains only $\mathbf{a}$ built by (117) and $\mathfrak{b}$ built by (123), all the scalars in $\mathbf{a}$ and $\mathfrak{b}$ can be viewed as distributed independently and uniformly, except for the scalars in the first column of $\mathfrak{b}$ which are completely dependent and are determined by the equality (125).

Proof: is in Appendix R.3.

Now, by reverting to the equality (120) and rewriting it for each $\boldsymbol{\xi}_i \in \mathfrak{c}$, the extractor obtains the following matrix equation

$$(\, \mathrm{diag}(\mathbf{a}') \cdot \mathfrak{c} - \mathfrak{b} \cdot \mathrm{diag}(\mathbf{a}) \cdot \mathfrak{a} \,) \cdot \mathbf{T} = (\, \mathfrak{b} \cdot \hat{\alpha} - \hat{\alpha}' \,)\, H\,. \tag{126}$$

Using the definitions of $\mathfrak{b}$ (123) and $\mathbf{a}'$ (124), the extractor rewrites (126) as

$$(\, \mathrm{diag}(\mathfrak{b} \cdot \mathbf{a}) \cdot \mathfrak{b} - \mathfrak{b} \cdot \mathrm{diag}(\mathbf{a}) \,) \cdot \mathfrak{a} \cdot \mathbf{T} = (\, \mathfrak{b} \cdot \hat{\alpha} - \hat{\alpha}' \,)\, H\,. \tag{127}$$

All the entries on both sides of the matrix equation (127) are known to the extractor, so it may wish to express the vector column $\mathbf{T}$ (57) through H by solving (127) as a linear system. However, all the weights of $P \in \mathbf{T}$ are equal to zero in the linear system (127) due to the same reason as for the transformation rule (121). In fact, the matrix within the brackets on the left-hand side of (127), let's call it $\mathfrak{s}$,

$$\mathfrak{s} = \mathrm{diag}(\mathfrak{b} \cdot \mathbf{a}) \cdot \mathfrak{b} - \mathfrak{b} \cdot \mathrm{diag}(\mathbf{a})\,, \tag{128}$$

has the non-empty kernel. Namely, there exsists at least one nonzero vector, $\{1\}^{t+s+1}$, such that

$$\mathfrak{s} \cdot \{1\}^{t+s+1} = \{0\}^{t+s+1}\,.$$

Thus, $\det(\mathfrak{s}) = 0$ and, hence, $\det(\mathfrak{s} \cdot \mathfrak{a}) = 0$, which means the matrix equation (127) cannot be resolved for $\mathbf{T}$ by taking $(\mathfrak{s} \cdot \mathfrak{a})^{-1}$. Anyway, the matrix $\mathfrak{s}$ (128) contains $\mathfrak{b}$ and $\mathbf{a}$ only, which makes Claim 5 applicable to it; so the extractor is going to use $\mathfrak{s}$ in a different way, as follows.

Since finding P from (127) is not possible due to $\det(\mathfrak{s} \cdot \mathfrak{a}) = 0$ (the underlying reason is that this would mean $P \sim H$, which would break the DL relation assumption), the extractor constructs the following truncated version of

(127). It removes P from $\mathbf{T}$ which is at the first position there, thus leaving the truncated vector column

$$\tilde{\mathbf{T}} = \begin{bmatrix} Q_0 \\ \vdots \\ Q_{t-1} \\ S_0 \\ \vdots \\ S_{s-1} \end{bmatrix} . \tag{129}$$

Also, it removes the first column of the matrix $(\mathfrak{s} \cdot \mathfrak{a})$ which is of zeros (it contains the weights of P, all of which are zeros), denoting the resulting $(t+s) \times (t+s+1)$ matrix as $\mathfrak{m}$. The extractor calculates the column vector of $(t+s+1)$ scalars on the right-hand side of (127) as

$$\mathbf{h} = \begin{bmatrix} h_0 \\ h_1 \\ \vdots \\ h_{t+s} \end{bmatrix} = (\, \mathfrak{b} \cdot \hat{\alpha} - \hat{\alpha}' \,) \; . \tag{130}$$

Finally, the truncated version of (127) takes the form of

$$\mathfrak{m} \cdot \tilde{\mathbf{T}} = \mathbf{h} H \, . \tag{131}$$

We make the following claim about $\mathfrak{m}$.

Claim 6:
The $(t+s) \times (t+s+1)$ matrix $\mathfrak{m}$, which is constructed by removal of the first column from the matrix $(\mathfrak{s} \cdot \mathfrak{a})$ where $\mathfrak{a}$ is defined by (116) and $\mathfrak{s}$ is defined by (128), with overwhelming probability has rank $(t+s)$.

Proof: is in Appendix R.4.

Once $\mathfrak{m}$ has rank $(t+s)$, according to Claim 6, it has at least one submatrix of rank $(t+s)$. As there are only $(t+s+1)$ submatrices of size $(t+s) \times (t+s)$ in $\mathfrak{m}$, the extractor finds the one with rank $(t+s)$ among them, denote it as $\mathfrak{r}$, by simply iterating and checking that the determinant is nonzero.

Let the found $(t+s) \times (t+s)$ submatrix $\mathfrak{r}$ of rank $(t+s)$ be $\mathfrak{m}$ with r'th row removed, with $r \in [0 \ldots t+s]$ found by the extractor. The extractor removes r'th item from $\mathbf{h}$ (130) as well, denoting the reduced vector as $\acute{\mathbf{h}}$. Thus, it obtains the equation

$$\mathfrak{r} \cdot \tilde{\mathbf{T}} = \acute{\mathbf{h}} H \, , \tag{132}$$

where $\det(\mathfrak{r}) \neq 0$.

The extractor solves (132) for $\tilde{\mathbf{T}}$

$$\tilde{\mathbf{T}} = \mathfrak{r}^{-1} \cdot \acute{\mathbf{h}} H \tag{133}$$

and, hence, it has every $Q_j \in \tilde{\mathbf{T}}$, $j \in [0 \ldots t-1]$, expressed as H multiplied by a known scalar, which breaks the DL relation assumption. Namely, according to Theorem 14 premise, there is at least one nonzero Q_j for some $j \in [0 \ldots t-1]$, and also it holds that $H \mathrel{!=} \mathrm{lin}(\mathrm{nz}(\mathbf{Q}) \cup \{P\})$, however, according to (133), the extractor has found a scalar such that $Q_j \sim H$.

Thus, under the premise of Claim 2, we have built an algorithm that breaks the DL relation assumption. The Claim 2 is proved.

R SAME FACTOR SUBCLAIM PROOFS

R.1 SCALAR DISTRIBUTIONS IN THE TRANSITION MATRIX B

Proof: [Claim 3] The property a) is trivial. The property b) follows from the fact that $\det(\mathfrak{a}^{-1}) \neq 0$, both of $\mathfrak{a}$ and $\mathfrak{a}^{-1}$ are completely independent of $\mathfrak{c}$, and, hence, $(t+s)$ independent randomnesses in $\xi'_i \in \mathfrak{c}$ map one-to-one to $(t+s+1)$ randomnesses in $\mathbf{b}_i$ by the formula $\mathbf{b}_i = \xi'_i \cdot \mathfrak{a}^{-1}$, with the additional constraint $\left\langle \mathbf{b}_i, \{1\}^{t+s+1} \right\rangle = 1$ which follows from the equality to 1 of all items in the first columns of $\mathfrak{a}$ and $\mathfrak{c}$.

R.2 SCALAR DISTRIBUTIONS IN THE WITNESS VECTOR A

Proof: [Claim 4] Before sampling the base $\mathfrak{a}$, let the extractor construct the random base $\mathfrak{e}$ that includes $\dot{\xi}, \ddot{\xi}$ and has all the other its base vectors sampled independently and uniformly, as in (116). Also, let the extractor perform $(t+s+1)$ rewindings and obtain the witnesses a in (115) for the base $\mathfrak{e}$, collecting them into the vector $\mathbf{e}$. As $\dot{\xi}, \ddot{\xi} \in \mathfrak{e}$, the vector $\mathbf{e} \ni \dot{a}, \ddot{a}$ contains at least two different scalars. Note, since $\dot{\xi}, \ddot{\xi} \in \mathfrak{e}$ are not collinear and the other base vectors in $\mathfrak{e}$ are random, it holds that $\det(\mathfrak{e}) \neq 0$.

For the newly sampled random base $\mathfrak{a}$, let the extractor obtain the witness vector $\mathbf{a}$ by making $(t+s+1)$ more rewindings. According to the transformation rule (124), the vectors $\mathbf{e}$ and $\mathbf{a}$ are connected as

$$\mathbf{a} = \mathfrak{d} \cdot \mathbf{e}\,, \quad \text{where } \mathfrak{d} = \mathfrak{a} \cdot \mathfrak{e}^{-1}. \tag{134}$$

For an isolated of $\mathfrak{a}$ consideration of $\mathfrak{d}$, according to the Claim 3, each row $\mathbf{d}_i \in \mathfrak{d}$ has all its items distributed uniformly at random, with $(t+s)$ of them independent and one of them, say, d_{i0}, completely determined by the equality $\left\langle \mathbf{d}_i, \{1\}^{t+s+1} \right\rangle = 1$. At the same time, for this consideration, as the vector $\mathbf{e}$ is defined before $\mathfrak{a}$ is sampled, and hence $\mathbf{e}$ is independent of $\mathfrak{a}$, the vector $\mathbf{e}$ is independent of $\mathfrak{d}$. Thus, in this consideration, each item $a_i \in \mathbf{a}$ calculated by the formula (134) as

$$a_i = \langle \mathbf{d}_i, \mathbf{e} \rangle \tag{135}$$

is the inner product of the uniformly distributed vector $\mathbf{d}_i$, which has $(t+s)$ independent items $d_{ij} \in \mathbf{d}_i \setminus \{d_{i0}\}$ and one dependent item d_{i0} calculated as

$$d_{i0} = 1 - \sum_{j=1}^{t+s} d_{ij}\,, \tag{136}$$

with the independent and not necessarily uniformly distributed vector $\mathbf{e}$ which has at least two different items. Inserting (136) into (135), a_i gets the form

$$a_i = e_0 + \sum_{j=1}^{t+s} (e_j - e_0) d_{ij}\,, \tag{137}$$

which makes a_i look uniformly random in an isolated of $\mathfrak{a}$ consideration of it, namely, in isolation of $\mathfrak{a}$ and, hence, without $\mathfrak{a}$'s dependency $\mathfrak{d}$, and with at least two different e_k's $\in \mathbf{e}$.

For each index $i \in [0 \ldots t+s]$, the scalar $a_i \in \mathbf{a}$ is independent of the other scalars in $\mathbf{a}$ since, according to (137), they are built using different and completely independent sources of randomness $\mathbf{d}_i$.

R.3 INDEPENDENCE OF SCALARS IN AN EXPRESSION CONTAINING ONLY B AND A

Proof: [Claim 5] According to the Claim 3, since neither the matrix $\mathfrak{c}$ nor its dependencies participate in the expression in question, all scalars in the matrix $\mathfrak{b}$ can be considered as independent and uniformly random, except for the ones in the first column which can be found from the equality (125).

As for the vector $\mathbf{a}$, its items are independent of the items in $\mathfrak{b}$ by the above, according to the Claim 3. In addition to this, by the Claim 4, the items in $\mathbf{a}$ are distributed uniformly at random and independently of each other.

R.4 RANK OF M

Proof: [Claim 6] Rank of the $(t+s) \times (t+s+1)$ matrix $\mathfrak{m}$ is equal to rank of the $(t+s+1) \times (t+s+1)$ matrix $(\mathfrak{s} \cdot \mathfrak{a})$, as the former is obtained from the latter by removing a column which contains only zeros (the first column).

Rank of the square matrix $(\mathfrak{s} \cdot \mathfrak{a})$ is equal to rank of the $(t+s+1) \times (t+s+1)$ square matrix $\mathfrak{s}$ (128), as the former is built as a product of the latter with an invertible matrix, namely, with the $(t+s+1) \times (t+s+1)$ square matrix $\mathfrak{a}$ (116) which has $\det(\mathfrak{a}) \neq 0$ as a random one. Thus,

$$\operatorname{rank}(\mathfrak{m}) = \operatorname{rank}(\mathfrak{s})\,. \tag{138}$$

Let us consider a submatrix of $\mathfrak{s}$ which is obtained by removing both the first column and row from $\mathfrak{s}$. We denote it as $\tilde{\mathfrak{s}}$ below. According to (128), each item $s_{ij} \in \tilde{\mathfrak{s}}$, where $i, j \in [1 \ldots t+s]$, has the form

$$s_{ij} = \langle \mathbf{b}_i, \mathbf{a} \rangle\, b_{ij} - a_j b_{ij}\,. \tag{139}$$

Recalling (125), the equality (139) rewrites as

$$s_{ij} = \left(a_0 \left(1 - \sum_{k=1}^{t+s} b_{ik} \right) + \sum_{k=1}^{t+s} a_k b_{ik} - a_j \right) b_{ij} = \left(a_0 + \sum_{k=1}^{t+s} (a_k - a_0) b_{ik} - a_j \right) b_{ij}\,. \tag{140}$$

The matrix $\mathfrak{s}$ comprises $\mathfrak{b}$ and $\mathbf{a}$ only, so Claim 5 applies to it. The same is true for $\tilde{\mathfrak{s}} \subset \mathfrak{s}$. Moreover, according to (140), each item $s_{ij} \in \tilde{\mathfrak{s}}$ is represented by a multivariate polynomial of total degree 3 of the set of variables $(\mathbf{b}_i \setminus \{b_{i0}\}) \cup \mathbf{a}$, each of which can be regarded, according to Claim 5, as distributed independently and uniformly.

Let us consider $\det(\tilde{\mathfrak{s}})$ constructed by Leibniz's formula as a sum of signed products of s_{ij}'s. This way, by (140), $\det(\tilde{\mathfrak{s}})$ is a multivariate polynomial of the independent and uniformly distributed random variables $(\mathbf{b}_i \setminus \{b_{i0}\}) \cup \mathbf{a}$. We rewrite (140) as follows, separating the $a_j b_{ij}^2$ summand in it,

$$s_{ij} = \left(a_0 + \sum_{k=1\ldots(t+s),\ k\neq j} (a_k - a_0) b_{ik} - a_j \right) b_{ij} - a_0 b_{ij}^2 + a_j b_{ij}^2 \,. \tag{141}$$

Consider the $\prod_{i=1}^{t+s} s_{ii}$ signed product component of $\det(\tilde{\mathfrak{s}})$. According to (141), it contributes the $\prod_{i=1}^{t+s} a_i b_{ii}^2$ summand to $\det(\tilde{\mathfrak{s}})$. As follows from (141), there is no other signed product in $\det(\tilde{\mathfrak{s}})$ which contributes any other summand containing $\prod_{i=1}^{t+s} a_i b_{ii}^2$. Thus, the multivariate polynomial representing $\det(\tilde{\mathfrak{s}})$ contains the uncompensated $\prod_{i=1}^{t+s} a_i b_{ii}^2$ and, therefore, $\det(\tilde{\mathfrak{s}})$ has total degree not less than $3(t+s)$.

By the Schwartz–Zippel lemma [10, 34, 28], having total degree greater than zero, $\det(\tilde{\mathfrak{s}})$ has only negligible probability to be zero and, thus, with overwhelming probability it holds that

$$\operatorname{rank}(\mathfrak{s}) = (t+s) \,, \tag{142}$$

which implies, by (138), that with overwhelming probability

$$\operatorname{rank}(\mathfrak{m}) = (t+s) \,.$$

The claim is proved.

S RANDOMLY WEIGHTED SUMS IMPLY THE SYSTEM IN MULTRATUG

When moving from the equality (51) to the system (52) in EFLRSLWB, we implicitly used Theorem 3. More details about this are proveded in the proof of Theorem 13, particularly in Appendix N, where the equality (51) corresponds to the equality (95).

However, in Multratug, verifier has the equality (68) instead of (51). The transition from (68) to the system (69) in Multratug may not seem apparent. Newertheless, with Theorem 14, which is a generalization of Theorem 3 to $(t+s+1)$-element tuples, the transition from (68) to (69) becomes easy, details are in the proof of the following claim.

Claim 7:
If the Multratug protocol in Figure 25 completes successfully, then verifier is convinced that the equality (68) implies the system (69) in it.

Proof: Let

$$\begin{aligned}
P &= \hat{U}_k, \\
\mathbf{Q} &= \{P_{s_k},\ \hat{I}_k\}, \\
\mathbf{S} &= \{A_k^{\mathbf{tmp}} - A_{s_k},\ U_{s_k} - U_k^{\mathbf{tmp}}\}, \\
H &= H, \\
Z &= J_k, \\
\mathbf{F} &= \{G,\ U_k^{\mathbf{tmp}}\}.
\end{aligned}$$

The right-hand sides of these equalities contain the elements from the Multratug scheme in Figure 25, whereas the left-hand sides contain ones from the protocol of Theorem 14 in Figure 24. By the formulas (57) and (58), respectively, the t-s-tuples become

$$\mathbf{T} = (\ \hat{U}_k,\ P_{s_k},\ \hat{I}_k,\quad A_k^{\mathbf{tmp}} - A_{s_k},\ U_{s_k} - U_k^{\mathbf{tmp}}\ ) \,, \tag{143}$$

$$\mathbf{D} = (\ J_k,\ G,\ U_k^{\mathbf{tmp}},\ 0,\qquad 0\ ) \,. \tag{144}$$

Also, in accordance to Figure 25, the random scalar vector $\boldsymbol{\xi}$ in the formula (61) becomes

$$\boldsymbol{\xi} = [1,\ \chi^{-1},\ \chi^{-1}\theta,\ \chi^{-1}\omega,\ \chi^{-1}\zeta] \,. \tag{145}$$

By Theorem 12, due to the $\texttt{zkLin22Choice}_{l,n,l}$ call in Figure 25, verifier is convinced that prover knows p_k, v_k such that the following equality holds to the accuracy of H component, for each $k \in [0 \ldots l-1]$

$$G + \theta U_k^{\mathbf{tmp}} + \chi J_k = p_k(P_{s_k} - K + \zeta U_{s_k} - \omega A_{s_k}) + v_k(K + \omega A_k^{\mathbf{tmp}} - \zeta U_k^{\mathbf{tmp}} + \theta \hat{I}_k + \chi \hat{U}_k) , \quad (146)$$

which becomes the equality (68) after elimianing the hash to group K. The elimination is performed the same way as for (92) in Appendix N. Namely, since K is orthogonal to everything else, it collapses guaranteeing $p_k = v_k$.

As a result, for X, Y calculated by the formulas (59), (60) using (143), (144), (145), the equality (68) rewrites as

$$\chi Y = \chi p_k X . \quad (147)$$

Everything to the accuracy of H. Since χ is a nonzero scalar known to both of the prover and verifier prior to applying the Theorem 12 protocol, both sides of (147) can be divided by it, and (147) rewrites as

$$Y = p_k X , \quad (148)$$

which means verifier is convinced that prover knows some a, namely, $a = p_k$, and $\hat{\alpha}$ such that $Y = aX + \alpha H$ holds. Moreover, by the above this connection between Y and X is established by a complete, sHVZK, and cWEE protocol of Theorem 12 (Lin2-2Choice lemma), which proves the relation (37).

Also, according to Figure 25 the following holds. The element $\hat{U}_k$ in the tuple $\mathbf{T}$ (143) is nonzero and is orthogonal to all the other nonzero elements of $\mathbf{T}$ and to the blinding generator H, i.e., $\hat{U}_k$!= lin(nz($P_{s_k}, \hat{I}_k$), nz($A_k^{\mathbf{tmp}} - A_{s_k}, U_{s_k} - U_k^{\mathbf{tmp}}$), H). The nonzero element H is ortogonal to all nonzero elements of the set $\{P_{s_k}, \hat{I}_k, \hat{U}_k\}$, i.e., H != lin(nz($P_{s_k}, \hat{I}_k$), $\hat{U}_k$). The element P_{s_k} is guaranteed nonzero.

Thus, all steps of the $\texttt{zkTElemRW}_{2,2}$ protocol in Figure 24 have been performed and the premise of Theorem 14 is met. Therefore, by Theorem 14 the verifier is convinced that the relation (62) holds, and, hence, the tuples (143), (144) are elementwise proportional to each other, to the acccuracy of H, which is equivalent to the system (69).

T SIGNATURE MULTRATUG

Proof: [Theorem 15] According to Figure 25, as the new vectors $\mathbf{U^{tmp}}, \hat{\mathbf{I}}$ are defined by the formulas (64), (63), all proofs of Theorem 13 for the EFLRSLWB scheme in Figure 22 transfer to the Multratug scheme in Figure 25.

In fact, $\mathbf{U^{tmp}}$ is indistinguishable from the independent uniform randomness due to the blinding components $\hat{\mu}H$ in it (64), hence $\mathbf{U^{tmp}}$ does not change anything. The same is for $\hat{\mathbf{I}}$ (63), which is indistinguishable from the independent uniform randomness and from the former $\mathbf{I}$ (41). This is proved in [29], and also can be proved using the method of [13]. Also, the new vectors $\hat{\mathbf{I}}$ and $\mathbf{U^{tmp}}$ get into $\hat{\mathbf{U}}$'s pre-image, however this does not change anything, only depricates any linear dependency of $\hat{\mathbf{U}}$'s with $\hat{\mathbf{I}}$'s and $\mathbf{U^{tmp}}$'s. The same is for the blinding generator H, which gets the new vectors into its pre-image.

Note, Theorem 14, which we use for Multratug instead of Theorem 3 for EFLRSLWB, does not require in the premise $\hat{\mathbf{I}}$'s and $\mathbf{U^{tmp}}$'s to be proved linearly independent of each other, only $\hat{\mathbf{U}}$'s and H are required to be proved linearly independent of $\hat{\mathbf{I}}$'s and $\mathbf{U^{tmp}}$'s.

With the former $\mathbf{I}$, EFLRSLWB has (51) and gets (52) from it. With the new $\mathbf{U^{tmp}}, \hat{\mathbf{I}}$, Multratug has (68) instead of (51), and gets (69) from it by Claim 7 in Appendix S, instead of (52). As (52) is a subset of (69), with $\hat{\mathbf{I}}$ substituted for $\mathbf{I}$, all the subsequent EFRLSLWB proofs use $\hat{\mathbf{I}}$ instead of $\mathbf{I}$ and thus translate to Multratug proofs.

This way, Multratug appears to be proved a linkable threshold ring signature, provided that EFLRSLWB is proved to be such. And, all the properties listed in Theorem 13 for the linkable threshold ring signature EFLRSLWB in Figure 22 transfer to the linkable threshold ring signature Multratug in Figure 25.

U VECTOR SCHNORR ARGUMENT

Proof: [Theorem 16] Design of the protocol in Figure 26 is clearly Schnorr-like. Hence, its completeness, sHVZK, and cWEE can be proved in the standard way, so we do not include a detailed proof here, clarifications are the same as for `zk2ElemComm` in Appendix A.

In addition to this, all the explanatory details can be found in [2], where the sHVZK and cWEE properties are proved for quite a similar protocol.

V NON-ZK LOG-SIZE VECTOR COMMITMENT ARGUMENT

Proof: [Theorem 17] For $n > 4$, the protocol in Figure 27 comprises the reductions used in the inner product argument [7] with $\mathbf{b} = \{0\}^n$ and, hence, it is complete and has cWEE for these reductions. For $n \leqslant 4$, $\mathcal{P}$ simply opens the witness to $\mathcal{V}$ and the latter checks the relation. Thus, for $n \geqslant 1$, the protocol is complete and has cWEE.

Also, in [2] the sHVZK and cWEE properties are proved for a similar protocol.

W OPTIMIZED ZK LOG-SIZE VECTOR COMMITMENT ARGUMENT

Proof: [Theorem 18] Completeness is by-design. The $\mathtt{argVC}_{n+1}$ call in the last step of $\mathtt{zkVC}_n^{\mathbf{opt}}$ has cWEE by Theorem 17. Having extracted the witness $\boldsymbol{\tau}$ from it, the protocol turns out to be $\mathtt{zkNElemComm}_{n+1}$, which has cWEE by Theorem 16. Thus, $\mathtt{zkVC}_n^{\mathbf{opt}}$ has cWEE. Even with the opened $\boldsymbol{\tau}$ the protocol remains sHVZK by Theorem 16, so partially hiding it inside $\mathtt{argVC}_{n+1}$ doesn't make $\mathtt{zkVC}_n^{\mathbf{opt}}$ less zero-knowledge. Thus, $\mathtt{zkVC}_n^{\mathbf{opt}}$ is sHVZK.

Also, in [2] such a composition is proved to be having sHVZK and cWEE properties.

X NOTES ABOUT DUALRING-EC

The DualRing-EC signature, according to its security model in [33], requires all keys in the ring to be honestly generated, i.e., it does not work with malformed ones. In contrast, our security model defined by Theorem 9 allows malformed keys to appear in the rings. We have tried to assess, whether an environment in which EFLRSL remains secure can be used for DualRing-EC, and discovered the following attack to DualRing-EC, of course, with reference to our security model.

Let a dishonest $\mathcal{P}$ want to sign with DualRing-EC using a ring of four malformed public keys, none of which it knows secret key for. Knowing no secret keys for Q, R, K and knowing secret key for P, it creates the four-element ring as $\{Q, R, P+K, P-K\}$. Then $\mathcal{P}$ performs as though it signs honestly with P's secret key using three-element ring $\{Q, R, P\}$. However, it still hashes the four-element ring to create the challenge. Instead of creating the Sum Argument [33] for three challenges c_0, c_1, c_2, which correspond to Q, R, P, it splits c_2 into two halves and includes the Sum Argument for four challenges $c_0, c_1, c_2/2, c_2/2$ into the forgery. After that, honest $\mathcal{V}$ accepts this signature.

Y LOW ANONYMITY OF U/X

Let us show some anonymity implications of having an element of the form $x^{-1}U$ in a public transcript such that U is a fixed generator and x is a private key. The element may not be necessarily a linking tag, such element may appear, for instance, in a part of the scheme proving the balance.

Consider a rather possible case of non-uniform distribution of x's. Let the distribution have a probability peak for pairs of private keys (x_1, x_2) such that $x_2 = 2x_1$. Consequently, there will be non-negligible probability to randomly pick two signatures which were signed with keys from the same pair. These two signatures will be linked together by simply checking whether the element $x_2^{-1}U$ multiplyed by 2 is equal to its counterpart.

The obvious objection to this case is that the system may by-design forbid such the non-uniform distributions or other tightly coupled keys. This is, for example, the case in [31], where private keys behind the public keys in the rings have the form $x = b + r$ with hidden b and with independently and uniformly distributed r which may even be known to adversary. Thus, the element in question takes the form

$$(b+r)^{-1}U\ , \quad \text{where } r \text{ is known to the adversary, and always is independently and uniformly distributed.}$$

According to [21, 32, 25], this form makes it impossible to break anonymity, even if the adversary is diligently observing r.

Takeaway from this is that if a scheme conatains an element of the form $x^{-1}U$, then it is not anonymous w.r.t. chosen public key attackers. Also, in this case it seems not possible to follow the usual methods for proving existential unforgeability against adaptive chosen message / public key attackers, even if the scheme possesses this property.

Z APPLICATION TO KZG COMMITMENTS AND PLONK

Let us sketch out a couple of schemes implementing the key idea of the Lin2-Choice lemma described in Section 4.1.3. These schemes are shown in Figure 32 and Figure 33, and are about the application to KZG commitments which we started to discuss in Section 12.6.

To keep things brief, we sketch out these schemes without soundness proofs, although it seems to be not difficult to prove it. Also, we present these schemes only as arguments of knowledge, without zero-knowledge, considering the latter can be added in the number of ways which exist for KZG commitments, e.g., as in [11].

We do not investigate practical value of these two schemes, providing the sketches mainly to illustrate applicability of the key idea of our main lemma to the other type of commitments. Although, apparently, these schemes can immediately lead to, e.g., a kind of constant-size ring signature and a full membership proof in blockchains. They seem implementable either standalone or as custom gates for PLONK architecture [11].

Adhering to the notation from [5], for a polynomial $\boldsymbol{f}$ we denote a KZG commitment to it as $\boxed{\boldsymbol{f}}$, assuming the following things hold and are known to both $\mathcal{P}$ and $\mathcal{V}$ beforehand

- $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t$ are three prime-order groups of the same cardinality with the corresponding scalar field $\mathbb{F}_{\bar{p}}$. Q-DLOG assumption [11] holds for them. $G \in \mathbb{G}_1$ and $S \in \mathbb{G}_2$ are the group generators.
- There is an efficiently computable non-trivial pairing function $e : \mathbb{G}_1 \times \mathbb{G}_2 \mapsto \mathbb{G}_t$.
- Predefined maximal degree is d. A ring of univariate polynomials over $\mathbb{F}_{\bar{p}}$ of degree less than d is $\mathbb{F}_{<d}[X]$. We denote by X the sole variable of the polynomials.
- The secret scalar that was chosen during the trusted setup phase and then forgotten is τ. The global parameters in relation to this are

$$\mathbf{H} = (H_0 = G,\ H_1 = \tau G,\ \ldots,\ H_d = \tau^{d-1} G) \in \mathbb{G}_1^d, \quad \mathbf{K} = (K_0 = S,\ K_1 = \tau S) \in \mathbb{G}_2^2 .$$

- A primitive k-th root of unity is $\omega \in \mathbb{F}_{\bar{p}}$ such that $\omega^k = 1$. It is assumed that $1 \ll k \ll d$. The corresponding subgroup of points is $\boldsymbol{\Omega} = \{1, \omega, \ldots, \omega^{k-1}\} \in \mathbb{F}_{\bar{p}}^{k*}$. We let the number n of elements in decoy sets in our sketches to be not greater than k.
- Lagrange basis in $\mathbb{F}_{<k}[X] \subset \mathbb{F}_{<d}[X]$ is defined over $\boldsymbol{\Omega}$ as

$$\boldsymbol{L} = \{\lambda_i\}_{i=0}^{k-1} = \left\{ \prod_{\substack{0 \leqslant j < k\,, \\ j \neq i}} \frac{(X - \omega^j)}{(\omega^i - \omega^j)} \right\}_{i=0}^{k-1} .$$

- Thus, the Lagrange form $\mathbf{L}$ of the global parameters $\mathbf{H}$ built using $\boldsymbol{L}$ and τ is

$$\mathbf{L} = \{\lambda_i(\tau) G\}_{i=0}^{k-1} = \left\{ \left(\prod_{\substack{0 \leqslant j < k\,, \\ j \neq i}} \frac{(\tau - \omega^j)}{(\omega^i - \omega^j)} \right) G \right\}_{i=0}^{k-1} .$$

Z.1 HELPER GADGETS

We will need three helper gadgets built using the standard proof primitives that already exist for KZG commitments. The first of them is shown in Figure 29, it is an argument for the following relation

$$\mathcal{R} = \{ \boxed{\boldsymbol{f}} \in \mathbb{G}_1,\ u, v \in \mathbb{F}_{\bar{p}}\,;\ \boldsymbol{f} \in \mathbb{F}_{<d}[X] \mid \boldsymbol{f}(u) = v \} . \tag{149}$$

It proves that the value of the opening $\boldsymbol{f}$ of the commitment $\boxed{\boldsymbol{f}}$ at the point u is v. The technique of this proof is informally introduced in [5].

$\boxed{\texttt{kzgPoFV}(\boxed{\boldsymbol{f}}, u, v\,;\ \boldsymbol{f})}$

Relation $\mathcal{R} = \{ \boxed{\boldsymbol{f}} \in \mathbb{G}_1,\ u, v \in \mathbb{F}_{\bar{p}}\,;\ \boldsymbol{f} \in \mathbb{F}_{<d}[X] \mid \boldsymbol{f}(u) = v \}$ // (149)

$\mathcal{P}$'s input : ($\boxed{\boldsymbol{f}}, u, v\,;\ \boldsymbol{f}$)

$\mathcal{V}$'s input : ($\boxed{\boldsymbol{f}}, u, v$)

$\mathcal{P}$'s output : none

$\mathcal{V}$'s output: *Accept* or *Reject*

$\boxed{\mathcal{P}}$: constructs $\boldsymbol{q} \in \mathbb{F}_{<d}[X]$ such that $(X - u)\boldsymbol{q}(X) = \boldsymbol{f}(X) - v$
computes $\boxed{\boldsymbol{q}}$

$\boxed{\mathcal{P} \to \mathcal{V}}$: $\boxed{\boldsymbol{q}}$

$\boxed{\mathcal{V}}$: **return**s *Accept* iff the following holds

$$e(\boxed{\boldsymbol{q}},\ K_1 - uK_0) \stackrel{?}{=} e(\boxed{\boldsymbol{f}} - vH_0,\ K_0)$$

Figure 29: KZG proof of polynomial value at point

Our second helper gadget is shown in Figure 30, it is an extended version of the ZeroTest gadget from [5] which proves that a committed polynomial $\boldsymbol{f}$ is zero on $\boldsymbol{\Omega}$. Our second gadget proves the same for the polynomial $(\boldsymbol{f} - \boldsymbol{a} \cdot \boldsymbol{m})$, for which $\mathcal{V}$ knows the commitments $\boxed{\boldsymbol{f}}$ and $\boxed{\boldsymbol{a}}$, and also knows the polynomial $\boldsymbol{m}$ in explicit form. This gadget is an argument for the relation

$$\mathcal{R} = \left\{ \begin{array}{l} \boxed{\boldsymbol{f}}, \boxed{\boldsymbol{a}} \in \mathbb{G}_1,\ \boldsymbol{m} \in \mathbb{F}_{<d}[X]\,; \\ \boldsymbol{f} \in \mathbb{F}_{<d}[X],\ \boldsymbol{a} \in \mathbb{F}_{<d}[X] \end{array} \ \middle|\ \forall \alpha \in \boldsymbol{\Omega} : \boldsymbol{f}(\alpha) - \boldsymbol{a}(\alpha)\boldsymbol{m}(\alpha) = 0 \right\}. \tag{150}$$

Note that in some cases the actual degrees of the polynomials $\boldsymbol{a}$ and $\boldsymbol{m}$ must be somewhat less than indicated by the limit d. Otherwise $\mathcal{P}$ will not be able to build $\boxed{\boldsymbol{q}}$. Notwithstanding this, as follows from the implementation in Figure 30 the protocol remains sound in any case, this is enough for sketching our idea.

`kzgZeroTestEx`$_{\boldsymbol{\Omega}}(\boxed{\boldsymbol{f}}, \boxed{\boldsymbol{a}}, \boldsymbol{m}\,;\, \boldsymbol{f}, \boldsymbol{a})$

Relation $\mathcal{R} = \left\{ \begin{array}{l} \boxed{\boldsymbol{f}}, \boxed{\boldsymbol{a}} \in \mathbb{G}_1,\ \boldsymbol{m} \in \mathbb{F}_{<d}[X]\,; \\ \boldsymbol{f} \in \mathbb{F}_{<d}[X],\ \boldsymbol{a} \in \mathbb{F}_{<d}[X] \end{array} \ \middle|\ \forall \alpha \in \boldsymbol{\Omega} : \boldsymbol{f}(\alpha) - \boldsymbol{a}(\alpha)\boldsymbol{m}(\alpha) = 0 \right\}$ // (150)

$\mathcal{P}$'s input : $(\boxed{\boldsymbol{f}}, \boxed{\boldsymbol{a}}, \boldsymbol{m}\,;\, \boldsymbol{f}, \boldsymbol{a})$

$\mathcal{V}$'s input : $(\boxed{\boldsymbol{f}}, \boxed{\boldsymbol{a}}, \boldsymbol{m})$

$\mathcal{P}$'s output : none

$\mathcal{V}$'s output: *Accept* or *Reject*

$\boxed{\mathcal{P}}$: constructs $\boldsymbol{q} \in \mathbb{F}_{<d}[X]$ such that $(X^k - 1)\boldsymbol{q}(X) = \boldsymbol{f}(X) - \boldsymbol{a}(X)\boldsymbol{m}(X)$

computes $\boxed{\boldsymbol{q}}$ // — using **H** or **L**

$\boxed{\mathcal{P} \to \mathcal{V}}$: $\boxed{\boldsymbol{q}}$

$\boxed{\mathcal{V}}$: $x \leftarrow\$ \mathbb{F}^*_{\bar{p}}$

$\boxed{\mathcal{V} \to \mathcal{P}}$: x

$\boxed{\mathcal{P}}$: computes $q_x = \boldsymbol{q}(x)$, $f_x = \boldsymbol{f}(x)$, $a_x = \boldsymbol{a}(x)$

$\boxed{\mathcal{P} \to \mathcal{V}}$: q_x, f_x, a_x

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: run `kzgPoFV`$(\boxed{\boldsymbol{q}}, x, q_x\,;\, \boldsymbol{q})$, // — or play a batched

`kzgPoFV`$(\boxed{\boldsymbol{f}}, x, f_x\,;\, \boldsymbol{f})$ // — version of these

`kzgPoFV`$(\boxed{\boldsymbol{a}}, x, a_x\,;\, \boldsymbol{a})$ // — three proofs

$\boxed{\mathcal{V}}$: **return**s *Accept* iff the following holds

$$(x^k - 1)q_x \stackrel{?}{=} f_x - a_x\boldsymbol{m}(x)$$

Figure 30: KZG zero test on a set of roots of unity

Our third helper gadget is an extension of the SumCheck gadget from [5]. It is shown in Figure 31 and is an argument for the relation

$$\mathcal{R} = \{ \boxed{\boldsymbol{f}} \in \mathbb{G}_1,\ \boldsymbol{m} \in \mathbb{F}_{<d}[X],\ v \in \mathbb{F}_{\bar{p}}\,;\, \boldsymbol{f} \in \mathbb{F}_{<d}[X] \ \mid\ \sum_{\alpha \in \boldsymbol{\Omega}} \boldsymbol{f}(\alpha)\boldsymbol{m}(\alpha) = v \}. \tag{151}$$

The difference between our gadget and SumCheck from [5] is that, instead of proving that the sum of a polynomial $\boldsymbol{f}$ on $\boldsymbol{\Omega}$ is zero, we prove that the sum of the polynomial $\boldsymbol{f} \cdot \boldsymbol{m}$, where $\boldsymbol{m}$ is explicitly known, is equal to the given value v. We use the same method as SumCheck to prove this. Our note about the actual degrees of the polynomials and soundness applies here, too.

$$\texttt{kzgSumCheckEx}_{\boldsymbol{\Omega}}(\,\boxed{\boldsymbol{f}}\,,\boldsymbol{m},v\,;\,\boldsymbol{f}\,)$$

Relation $\mathcal{R} = \{\,\boxed{\boldsymbol{f}} \in \mathbb{G}_1,\ \boldsymbol{m} \in \mathbb{F}_{<d}[X],\ v \in \mathbb{F}_{\bar{\mathsf{p}}}\,;\ \boldsymbol{f} \in \mathbb{F}_{<d}[X] \ \mid\ \sum_{\alpha\in\boldsymbol{\Omega}} \boldsymbol{f}(\alpha)\boldsymbol{m}(\alpha) = v\,\}$ // (151)

$\mathcal{P}$'s input : ($\boxed{\boldsymbol{f}}$, $\boldsymbol{m}, v$; $\boldsymbol{f}$)

$\mathcal{V}$'s input : ($\boxed{\boldsymbol{f}}$, $\boldsymbol{m}, v$)

$\mathcal{P}$'s output : none

$\mathcal{V}$'s output: *Accept* or *Reject*

// ——— the target equality in $\mathcal{R}$ rewrites as $\sum_{\alpha\in\boldsymbol{\Omega}} \boldsymbol{g}(\alpha) = 0$, where $\forall\,\alpha \in \Omega : \boldsymbol{g}(\alpha) = \boldsymbol{f}(\alpha)\boldsymbol{m}(\alpha) - v/k$ ———

// ——— $\sum_{\alpha\in\boldsymbol{\Omega}} \boldsymbol{g}(\alpha) = 0$ is proved the standard way using $\boldsymbol{h} \in \mathbb{F}_{<d}[X]$ such that $\forall\,\alpha \in \Omega : \boldsymbol{h}(\omega\alpha) = \boldsymbol{h}(\alpha) + \boldsymbol{g}(\omega\alpha)$ ——

// ——— thus, it suffices to prove that $\boldsymbol{h}(\omega^{k-1}) = 0$ and $\forall\,\alpha \in \Omega : \boldsymbol{h}(\alpha) - \boldsymbol{h}(\omega\alpha) + \boldsymbol{f}(\omega\alpha)\boldsymbol{m}(\omega\alpha) - v/k = 0$ ———

$\boxed{\mathcal{P}}$: defines $\boldsymbol{g} \in \mathbb{F}_{<d}[X]$ as $\boldsymbol{g}(X) = \boldsymbol{f}(X)\boldsymbol{m}(X) - v/k$

constructs $\boldsymbol{h} \in \mathbb{F}_{<d}[X]$ such that $\boldsymbol{h}(\omega^{k-1}) = 0$ and $\forall\alpha \in \Omega : \boldsymbol{h}(\omega\alpha) = \boldsymbol{h}(\alpha) + \boldsymbol{g}(\omega\alpha)$

constructs $\boldsymbol{q} \in \mathbb{F}_{<d}[X]$ such that $(X^k - 1)\boldsymbol{q}(X) = \boldsymbol{h}(X) - \boldsymbol{h}(\omega X) + \boldsymbol{g}(\omega X)$

computes $\boxed{\boldsymbol{h}}$, $\boxed{\boldsymbol{q}}$ // — using $\mathbf{H}$ or $\mathbf{L}$

$\boxed{\mathcal{P} \to \mathcal{V}}$: $\boxed{\boldsymbol{h}}$, $\boxed{\boldsymbol{q}}$

$\boxed{\mathcal{V}}$: $x \leftarrow\$\ \mathbb{F}^*_{\bar{\mathsf{p}}}$

$\boxed{\mathcal{V} \to \mathcal{P}}$: x

$\boxed{\mathcal{P}}$: computes $q_x = \boldsymbol{q}(x)$, $h_x = \boldsymbol{h}(x)$, $h_{\omega^{k-1}} = \boldsymbol{h}(\omega^{k-1})$,
$h_{\omega x} = \boldsymbol{h}(\omega x)$, $f_{\omega x} = \boldsymbol{f}(\omega x)$

$\boxed{\mathcal{P} \to \mathcal{V}}$: $q_x,\ h_x,\ h_{\omega x},\ f_{\omega x},\ h_{\omega^{k-1}}$

$\boxed{\mathcal{P}\text{ and }\mathcal{V}}$: run $\texttt{kzgPoFV}(\,\boxed{\boldsymbol{q}}\,, x, q_x\,;\,\boldsymbol{q}\,)$, $\texttt{kzgPoFV}(\,\boxed{\boldsymbol{h}}\,, x, h_x\,;\,\boldsymbol{h}\,)$, // — or play a batched

$\texttt{kzgPoFV}(\,\boxed{\boldsymbol{h}}\,, \omega x, h_{\omega x}\,;\,\boldsymbol{h}\,)$, $\texttt{kzgPoFV}(\,\boxed{\boldsymbol{f}}\,, \omega x, f_{\omega x}\,;\,\boldsymbol{f}\,)$, // — version of these

$\texttt{kzgPoFV}(\,\boxed{\boldsymbol{h}}\,, \omega^{k-1}, h_{\omega^{k-1}}\,;\,\boldsymbol{h}\,)$ // — five proofs

$\boxed{\mathcal{V}}$: **return**s *Accept* iff the following holds

$$h_{\omega^{k-1}} \stackrel{?}{=} 0 \quad\wedge\quad (x^k - 1)q_x \stackrel{?}{=} h_x - h_{\omega x} + f_{\omega x}\boldsymbol{m}(\omega x) - v/k$$

Figure 31: KZG proof of polynomial sum

Z.2 PROOF OF MEMBERSHIP IN A SET OF KZG COMMITMENTS

The sketch protocol $\texttt{kzgPoMforComm}_{n\leqslant k}$ in Figure 32 is a proof of membership, an argument for the relation

$$\mathcal{R} = \{\,\boldsymbol{P} \in \mathbb{F}^{n*}_{<d}[X],\ \boxed{z} \in \mathbb{G}_1;\ z \in \mathbb{F}_{<d}[X],\ s \in [0\ldots n-1], p \in \mathbb{F}_{\bar{\mathsf{p}}} \mid z = p\,\boldsymbol{p}_s\,\}\,, \tag{152}$$

where all elements in $\boldsymbol{P}$ are linearly independent of each other.

The relation (152) reads as follows. For the set of explicitly known polynomials $\boldsymbol{P} = \{\boldsymbol{p}_i\}_{i=0}^{n-1}$, which is considered as a decoy set, and for a given KZG commitment $\boxed{z}$, prover knows an index s and a factor p such that the opening z of $\boxed{z}$ turns out to be $z = p\,\boldsymbol{p}_s$.

We consider only decoy sets of size n which is not greater than the cardinality k of the set $\boldsymbol{\Omega}$. By the linear independence of polynomials in $\boldsymbol{P}$ we mean that no entry $\boldsymbol{p}_i \in \boldsymbol{P}$ is representable as a weighted sum of the polynomials from $\boldsymbol{P} \setminus \{\boldsymbol{p}_i\}$. We denote this property, as usual, as $\mathrm{ort}(\boldsymbol{P})$.

Our membership proof in Figure 32 works as follows. For the first, $\mathcal{P}$ builds the masking polynomial $\boldsymbol{a}$ which is one-hot on $\in \boldsymbol{\Omega}$ with the only nonzero value at $\omega^s \in \boldsymbol{\Omega}$. Without loss of generality, we omit consideration of the case $p = 0$.

Second, $\mathcal{P}$ sends the commitment $\boxed{\boldsymbol{a}}$ to $\mathcal{V}$. We assume that $\boxed{\boldsymbol{a}}$ does not reveal any information about the parameters h, p, s, although we omit showing this in the protocol. We assume $\boxed{\boldsymbol{a}}$ is made zero knowledge with one of the standard for KZG commitments ways.

Third, both of $\mathcal{P}$ and $\mathcal{V}$ multiply each polynomial in $\boldsymbol{p}_i \in \boldsymbol{P}$ by its own random challenge c_i and take the scalar product of the mask $\{\boldsymbol{a}(\omega^i)\}_{i=0}^{n-1}$ and the set $\{c_i \boldsymbol{p}_i\}_{i=0}^{n-1}$

$$\sum_{i=0}^{n-1} \boldsymbol{a}(\omega^i) c_i \boldsymbol{p}_i \ . \tag{153}$$

Finally, $\mathcal{P}$ demonstrates that the scalar product (153) is equal to the polynomial in question $\boldsymbol{z}$ multiplied by some scalar r, thus convincing $\mathcal{V}$ that $\{\boldsymbol{a}(\omega^i)\}_{i=0}^{n-1}$ is one-hot, which implies $\boldsymbol{z}$ is equal to one of $\boldsymbol{p}_i$'s up to a scalar multiplier.

$\boxed{\texttt{kzgPoMforComm}_{n \leqslant k}(\boldsymbol{P}, \boxed{z}\,; z, s, p)}$

Relation $\mathcal{R} = \{\, \boldsymbol{P} \in \mathbb{F}^{n*}_{<d}[X],\ \boxed{z} \in \mathbb{G}_1;\ z \in \mathbb{F}_{<d}[X],\ s \in [0 \ldots n-1], p \in \mathbb{F}_{\bar{p}} \mid z = p\, \boldsymbol{p}_s \,\}$ // (152)

// $\boldsymbol{P}$ in $\mathcal{R}$ satisfy $\mathrm{ort}(\boldsymbol{P})$.

$\mathcal{P}$'s input : $(\boldsymbol{P}, \boxed{z}\,; z, s, p)$

$\mathcal{V}$'s input : $(\boldsymbol{P}, \boxed{z}\,)$

$\mathcal{P}$'s output : none

$\mathcal{V}$'s output: *Accept* or *Reject*

$\boxed{\mathcal{P}}$: $h \leftarrow\!\$\ \mathbb{F}^*_{\bar{p}}$

// —— constructs $\boldsymbol{a} \in \mathbb{F}_{<k}[X]$ which is one-hot on $\boldsymbol{\Omega}$ ——————

lets $\boldsymbol{a} = \begin{cases} \boldsymbol{a}(\omega^s) = hp & \text{// that is, } \boldsymbol{a} \text{ has the only hot value } hp \text{ at } \omega^s \in \boldsymbol{\Omega} \\ \boldsymbol{a}(\omega^i) = 0 \ \text{ for all } i \in [0 \ldots k-1], i \neq s \end{cases}$

lets $\boxed{\boldsymbol{a}} = hp\, \mathbf{L}_{[s]}$ // —- where $\mathbf{L}_{[s]} \in \mathbf{L}$ is the Lagrange form element that is nonzero at ω^s

$\boxed{\mathcal{P} \to \mathcal{V}}$: $\boxed{\boldsymbol{a}}$

$\boxed{\mathcal{V}}$: $\mathbf{c} \leftarrow\!\$\ \mathbb{F}^{n*}_{\bar{p}}$

$\boxed{\mathcal{V} \to \mathcal{P}}$: $\mathbf{c}$

$\boxed{\mathcal{P}}$: lets $r = hc_s$

$\boxed{\mathcal{P} \to \mathcal{V}}$: r

// ———— below $\mathcal{P}$ convinces $\mathcal{V}$ that the equality $r\boldsymbol{z} = \sum_{i=0}^{n-1} \boldsymbol{a}(\omega^i) c_i \boldsymbol{p}_i$ holds ——————-

$\boxed{\mathcal{V}}$: $\xi \leftarrow\!\$\ \mathbb{F}_{\bar{p}}$

$\boxed{\mathcal{V} \to \mathcal{P}}$: ξ

$\boxed{\mathcal{P}}$: computes $v = z(\xi)$

$\boxed{\mathcal{P} \to \mathcal{V}}$: v

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: run $\texttt{kzgPoFV}(\boxed{z}, \xi, v\,; z\,)$

let $y = rv$

// —— construct $\boldsymbol{m} \in \mathbb{F}_{<k}[X]$ by k points using the Lagrange basis $\boldsymbol{L}$ such that ——————

// —— $\forall i \in [0 \ldots n-1] : \boldsymbol{m}(\omega^i) = c_i \boldsymbol{p}_i(\xi) \quad \wedge \quad \forall i \in [n \ldots k-1] : \boldsymbol{m}(\omega^i) = 0$ ——

let $\{m_i\}_{i=0}^{k-1} = \{c_i \boldsymbol{p}_i(\xi)\}_{i=0}^{n-1} \cup \{0\}_{i=n}^{k-1}$

let $\boldsymbol{m} = \sum_{i=0}^{k-1} m_i \lambda_i$

// ——— thus, it remains for $\mathcal{P}$ to convince $\mathcal{V}$ that $y = \sum_{i=0}^{k-1} \boldsymbol{a}(\omega^i) \boldsymbol{m}(\omega^i)$ holds ————

run $\texttt{kzgSumCheckEx}_{\boldsymbol{\Omega}}(\boxed{\boldsymbol{a}}, \boldsymbol{m}, y\,; \boldsymbol{a}\,)$

Figure 32: Membership proof for a set of KZG commitments

The reason why $\mathcal{V}$ is convinced of this is, informally, the same as in our main lemma. Namely, suppose there are at least two nonzero scalars in $\{\boldsymbol{a}(\omega^i)\}_{i=0}^{n-1}$. The commitment $\boxed{\boldsymbol{a}}$ makes $\mathcal{V}$ sure that these scalars do not change to suit the challenges c_i's. Therefore, the scalar product (153) turns out to be dependent on at least two independent random challenges. However, $\mathcal{P}$ manages to balance out these two randomnesses in (153) with the single scalar r, thus convincing $\mathcal{V}$ that the supposition is incorrect and, hence, $\{\boldsymbol{a}(\omega^i)\}_{i=0}^{n-1}$ cannot contain more than one nonzero scalar.

Z.3 PROOF THAT A COMMITTED POLYNOMIAL IS ONE-HOT OR ZERO

The second sketch protocol $\texttt{kzgCommIsOneHot}_{n\leqslant k}$, which is shown in Figure 33, is an argument for the relation

$$\mathcal{R} = \{\, \boxed{\boldsymbol{f}} \in \mathbb{G}_1;\ \boldsymbol{f} \in \mathbb{F}_{<d}[X],\ s \in [0\ldots n-1] \ \mid\ \forall i \in [0\ldots k-1], i \neq s\ :\ \boldsymbol{f}(\omega^i) = 0 \,\}\,. \tag{154}$$

This protocol proves that a polynomial $\boldsymbol{f}$ represented by the commitment $\boxed{\boldsymbol{f}}$ has at most one nonzero value on the set $\boldsymbol{\Omega}$. Moreover, this nonzero value is located among the first n points of $\boldsymbol{\Omega}$.

The design of the protocol is similar to the previous one, so we briefly describe it in the following few words. $\mathcal{P}$ sends to $\mathcal{V}$ a KZG commitment to the one-hot masking polynomial $\boldsymbol{a}$ that is nonzero at the same point as $\boldsymbol{f}$ on $\boldsymbol{\Omega}$. Next, $\mathcal{P}$ shows that the product $\boldsymbol{a} \cdot \boldsymbol{e}$ of this masking polynomial with a random polynomial $\boldsymbol{e}$ is equal, up to a scalar multiplier, to the polynomial in question $\boldsymbol{f}$ on $\boldsymbol{\Omega}$. This convinces $\mathcal{V}$ that both of $\boldsymbol{a}$ and $\boldsymbol{f}$ are one-hot.

$\boxed{\texttt{kzgCommIsOneHot}_{n\leqslant k}(\boxed{\boldsymbol{f}}\,;\,\boldsymbol{f}, s)}$

Relation $\mathcal{R} = \{\, \boxed{\boldsymbol{f}} \in \mathbb{G}_1;\ \boldsymbol{f} \in \mathbb{F}_{<d}[X],\ s \in [0\ldots n-1] \ \mid\ \forall i \in [0\ldots k-1], i \neq s\ :\ \boldsymbol{f}(\omega^i) = 0 \,\}$ // (154)

$\mathcal{P}$'s input : $(\boxed{\boldsymbol{f}}\,;\,\boldsymbol{f}, s)$

$\mathcal{V}$'s input : $(\boxed{\boldsymbol{f}})$

$\mathcal{P}$'s output : none

$\mathcal{V}$'s output: *Accept* or *Reject*

$\boxed{\mathcal{P}}$: computes $p = \boldsymbol{f}(\omega^s)$ // —- $\mathcal{P}$ assigns to p the only nonzero value of $\boldsymbol{f}$ on the set $\boldsymbol{\Omega}$

$h \leftarrow\$\ \mathbb{F}_{\mathsf{p}}^*$

// ——— constructs $\boldsymbol{a} \in \mathbb{F}_{<k}[X]$ which is one-hot on $\boldsymbol{\Omega}$ ———————

lets $\boldsymbol{a} = \begin{cases} \boldsymbol{a}(\omega^s) = hp & \text{// that is, } \boldsymbol{a} \text{ has the only hot value } hp \text{ at } \omega^s \in \boldsymbol{\Omega} \\ \boldsymbol{a}(\omega^i) = 0 \text{ for all } i \in [0\ldots k-1], i \neq s \end{cases}$

lets $\boxed{\boldsymbol{a}} = hp\,\mathbf{L}_{[s]}$ // —- where $\mathbf{L}_{[s]} \in \mathbf{L}$ is the Lagrange form element that is nonzero at ω^s

$\boxed{\mathcal{P} \to \mathcal{V}}$: $\boxed{\boldsymbol{a}}$

$\boxed{\mathcal{V}}$: $\mathbf{c} \leftarrow\$\ \mathbb{F}_{\mathsf{p}}^{n*}$

$\boxed{\mathcal{V} \to \mathcal{P}}$: $\mathbf{c}$

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: // —- construct $\boldsymbol{e} \in \mathbb{F}_{<k}[X]$ such that $\forall i \in [0\ldots n-1] : \boldsymbol{e}(\omega^i) = c_i \ \wedge\ \forall i \in [n\ldots k-1] : \boldsymbol{e}(\omega^i) = 0$

let $\boldsymbol{e} = \sum_{i=0}^{n-1} c_i \boldsymbol{\lambda}_i$

$\boxed{\mathcal{P}}$: lets $r = hc_s$

$\boxed{\mathcal{P} \to \mathcal{V}}$: r

// ———— now $\mathcal{P}$ convinces $\mathcal{V}$ that $\forall \alpha \in \boldsymbol{\Omega}\ :\ r\boldsymbol{f}(\alpha) = \boldsymbol{a}(\alpha)\boldsymbol{e}(\alpha)$ holds ——-

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: run $\texttt{kzgZeroTestEx}_{\boldsymbol{\Omega}}(r\,\boxed{\boldsymbol{f}}, \boxed{\boldsymbol{a}}, \boldsymbol{e}\,;\,\boldsymbol{f}, \boldsymbol{a})$

Figure 33: Proof that KZG commitment is one-hot