Efficient Linkable Ring Signature from Compact Commitment to Vector 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 a membership proof protocol 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. It is composed of several public coin arguments that are special honest verifier zero-knowledge and have computational witness-extended emulation. As the base building block which contributes most to the size, we use a black-box pivot argument that proves knowledge of a committed vector. This makes our signature combinable with other proofs with further size reduction. Also, we present an extended version of the EFLRSL signature of size* $2\log_2(n+l+1)+7l+4$*, aliased as Multratug, which simultaneously proves balance and allows for easy multiparty signing. All this takes place in a prime-order group without bilinear parings under the decisional Diffie-Hellman assumption in the random oracle model. Both our signatures are unforgeable w.r.t insider corruption and are also EU-CMA. 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 signatures.*

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

1 INTRODUCTION

In the previous paper [20] we created a log-size linkable threshold ring signature based on the Lin2-Xor lemma, which we proved there. Now we want to know two things, namely, can we generalize the Lin2-Xor lemma using an arbitrary vector commitment argument that has computational witness-extended emulation (WEE) and is special honest verifier zero-knowledge (HVZK)? Also, can we get a linkable threshold ring signature out of it that is more efficient in size and verification time? By vector commitment we mean a weighted sum of orthogonal generators in the group, and by the corresponding argument we mean a proof of knowledge of the weights.

We answer both of the above questions in the affirmative. Lin2-Choice lemma we present herein and its accompanying efficient ring signature seem to be useful findings. Our new ring signature keeps using the linking tag of the form $x^{-1}\mathcal{H}_{\textbf{point}}(xG)$, and also has a version with the linking tag form $x\mathcal{H}_{\textbf{point}}(xG)$, which is time-tested since the work by Liu, Wei, and Wong [15]. Although, both of these linking tags are indistinguishable from the independent uniform randomness, as we have already proved in [20]; for $x\mathcal{H}_{\textbf{point}}(xG)$ this was proved earlier in the work [8].

The signature we present, called EFLRSL, turns out to be extensible; we also introduce an extended version of it, called Multratug, which in addition to proving knowledge of signing keys also proves the sum of hidden amounts. By proof of the 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. To construct the extended version of our signature we provide one more lemma, Lin2-2Choice, as we call it.

We will not repeat common words about signatures from the introduction of [20], they all remain valid. We will keep our presentation concise, taking into account that many explanations can be taken from [20] as well as from the work of Benedikt Bünz et al. [5]. 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 an orthogonal dimension to all transmitted elements, which we learned from the work of Heewon Chung et al. [7]. Although, this method of making a protocol zero-knowledge seems to have been introduced a bit earlier, e.g., in the work of Attema and Cramer [2].

As for notation, we mainly use the notation from [20], supplementing it with notation from [5] and [7] where necessary, more on this in Section 2.1. Also, we use a kind of protocol representation inspired by [7].

Overall, in this paper we assume that a reader has an understanding of the works [5, 7, 20] and possesses an appropriate intuition, so we keep our descriptions and proofs brief, otherwise the paper would be too long. Moreover, since the methods of proving HVZK and WEE properties of protocols are already widely known, e.g., from [5, 7], and the same for unforgeability, anonymity, and other properties of signatures, e.g., from [15, 10, 8, 17], 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 couple of questions we have already outlined at the beginning, our motive in creating this paper is that we see 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 [15]. 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 we mean the possibility of using a single scheme, 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 of hidden amounts), many-out-of-many signing (threshold case, we use the word ‘threshold’ in this sense hereinafter), the case when public keys are formed according to the CryptoNote [22] 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., can be generated ad hoc and be completely malformed, nevertheless the LSAG signature remains secure and anonymous with them). In addition, it is desirable that a signature allows for easy implementation of multiparty signing operations, especially in the blockchain context (multisignature operations, described, e.g., in [11]).

After conducting a kind of pragmatic research, we found that the recently proposed linear-size CLSAG scheme [8], which generalizes and optimizes LSAG, solves all the listed problems, except for the threshold case. So, we take CLSAG for reference and compare the applicability of the know to the date top-performance log-size schemes with it, the results are shown in Table 1.

Table 1: Applicability of signature schemes

	Log-sz	Regular	Linkable	Balance	Thresh.*	Blockchain	General	MP**
CLSAG [8]		✓	✓	✓		✓	✓	✓
Lelantus Spark [11]	✓	✓	✓	✓		✓		✓
Triptych [17]	✓	✓	✓	✓		✓		
RingCT3.0 [23]	✓	✓	✓	✓	✓	✓		
Omniring [13]	✓	✓	✓	✓	✓	✓		✓
DualRing-EC [24]	✓	✓						

* 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 of the considered schemes are log-size, except for the referenced CLSAG, and they all provide the functionality of a regular ring signature. They are roughly ordered by size in the table. Of course, their versions that implement additional check-marked properties contribute extra bytes to the sizes. The most size and verification time efficient DualRing-EC signature [24] doesn't have any linkable version by-design, also its security model requires only properly generated keys, a forgery for the contrary case is shown in Appendix W.

All the other log-size signatures are linkable by-design, however, for each of them, linkability seems to can be eliminated in a trivial way (just for the sake of this comparison). All of them include balance proofs and are compatible with CryptoNote public keys (aka stealth addresses) [22] of the form $B + \mathcal{H}_{\mathbf{scalar}}(rA)G$. Only RingCT3.0 [23] and Omniring [13] substantially save signature space when several signers sign simultaneously. Triptych [17], RingCT3.0, Omniring have linking tags of the form U/x, where U is a predefined generator; this fact deanonymizes them in the general case, as we show in Appendix X.

The fact of having private key x in the tag's denominator also makes it hard to implement multisignature operations. Lelantus Spark [11] has its own subsystem that solves this problem, however, the entire scheme seems too narrowly tied to decentralized anonymous payments to be considered general (we are comparing everything against the general case only for the sake of our own interest).

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 [13] 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 by the Omniring authors, there is no claim that the scheme will be anonymous with malformed keys, like in the scenario described in Appendix X, in which LSAG and CLSAG still remain to be.

So, our second motivation is to create a scheme that covers all the properties specified in Table 1, like shown in Table 2, and also is close to the bottom of the table, i.e., has a relatively good size for typical use cases.

Table 2: Applicability of our scheme

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

For all this, our primary objective, of course, is to determine what can be obtained from the protocols of the Lin2-Choice and Lin2-2Choice lemmas presented in this paper, and how practical that would be. In the most elementary cryptographic group and with minimal additional means, i.e., using a compact argument of knowledge of a committed vector and without even involving the inner product argument.

1.2 COMMITMENT TO VECTOR, VECTOR COMMITMENT, AND ITS ARGUMENT

Our pivotal protocol, to which all proofs of membership and signatures in this paper eventually address, calling it at the last stage only once, is a vector commitment argument. Namely, throughout this paper, by the commitment to a vector or by the vector commitment, 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 in the group, 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 [24], however our implementation is a bit different.

The term vector commitment is already used in the literature for a construction described, e.g., in [6, 14, 9], in relation to groups with bilinear pairings. On the contrary, we denote by this term a construction in a pairings-free group that can be thought of as an extremely simplified form of the construction from [6]. We still decided on the term vector commitment argument because it reflects that prover knows a vector of weights, not a single scalar.

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 independent uniformly sampled, is often called as the Pedersen vector commitment. It is defined in [5] as an extension to Pedersen commitment [18]. In our terminology, the Pedersen vector commitment is a subset 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

The closely resembling argument in terms of its role in the paper and its construction is the compressed pivotal argument by Attema and Cramer in [2]. Our implementation of the vector commitment argument can be regarded as a subset case of the compressed pivot from [2] with the set of connected linear forms $L \equiv \emptyset$. Further in their work, Attema and Cramer obtain results for $L \neq \emptyset$. Meanwhile, we dig from the point $L \equiv \emptyset$ in a perpendicular direction, exploring 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 methods for constructing trusted-setup-free log-size membership proofs and signatures in it. The first of them arises from the identification scheme and its variations by Groth and Kohlweiss [10], and the second from the inner product argument and subsequent proof for an arbitrary arithmetic circuit by Bünz et al. [5].

We have already outlined the recent efficient schemes in Table 1. Thus, Triptych [17] and Lelantus Spark [11] rely on the idea of Groth and Kohlweiss [10] by building on top of it. At the same time, RingCT3.0 [23] and Omniring [13] heavily employ the inner product argument by Bünz et al. [5]. Also, there exist a number of other discrete-log, prime-order, pairings-free, trusted-setup-free, log-size schemes and approaches, which we do not mention because of their lower efficiency compared to the top-performers [23, 13, 17, 11].

The DualRing-EC signature by Tsz Hon Yuen et al. [24] has a rather restrictive security model, nevertheless it advances an elegant idea of better compression. Although we do not use this idea directly, it has inspired us for an optimized version of our vector commitment argument, which ended up being almost the same as the compressed pivot in [2] with the strongest security model.

Our signatures do not use the inner product argument. In addition, they use the underlying proofs of membership, which are quite different from [10]. Therefore, our signatures are not derived from the ones listed in Table 1.

The previous paper [20] represents an approach based on our own identification scheme which is different from [10, 5]. Namely, in [20] we obtained the early results showing what can be achieved by rotating pairs of elements in the base set $\mathbf{P}$ with challenges. However, the signature constructed in [20] is somewhat large in size. In this paper, we will take the idea of [20] and, by reinventing it targeting many-out-of-many proofs, will obtain much more efficient schemes.

Recent work by Russell W. F. Lai et al. [12] and subsequent work by Thomas Attema et al. [3] advance theoretical zero-knowledge frameworks for bilinear group arithmetic, with applications to signatures. These frameworks, at least the first of them, can be implemented in a prime-order group without bilinear pairings. Open questions are still if our signatures are subset cases of these frameworks or not; or how the signatures obtained with these frameworks in a prime-order group without bilinear pairings differ in efficiency from ours. We'd appreciate it if someone could do a comparison and answer these open questions.

1.4 CONTRIBUTION

This work results in the following novel trusted-setup-free pairings-free DDH-based log-size schemes.

1.4.1 LIN2-CHOICE LEMMA'S MEMBERSHIP PROOF

Lin2-Choice lemma is a generalization of the Lin2-Xor lemma [20] to the case of n pairs of elements. Having a ring $\mathbf{P} = \{P_i\}_{i=0}^{n-1}$ of n elements and a commitment Z to an arbitrary element $P_s \in \mathbf{P}$, using the Lin2-Choice lemma protocol it is possible to prove membership of Z in $\mathbf{P}$. This, itself, takes only 1 group elements and 1 scalar, to which the size of an externally employed vector commitment argument is added.

Thus, the lemma provides a concise 1-out-of-many membership proof. It looks to us that the design of the lemma protocol is quite simple. In addition, it 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.

We prove in detail that the lemma's membership proof has computational witness-extended emulation (WEE). We also informally show why it is special honest verifier zero-knowledge (HVZK), referring to the similar design in [7] which is formally proved there.

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 proof of membership, with size

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

This is a simplified version of our Multratug scheme, without balance proof or multiparty signing, with an uncomplicated design and linking tag (aka key image) form $x^{-1}\mathcal{H}_{\mathbf{point}}(xG)$.

Nevertheless, EFLRSL is general-purpose, namely, it is suitable for environments where keys can be generated by signers ad hoc and can be arbitrarily malformed. For example, EFLRSL can be used for implementing whistleblowing or for e-voting systems, for which LSAG [15] 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 simple general-purpose linkable ring signature, this comparison details are shown in Table 10.

Since it is based on a proof of membership which, according to the Lin2-Choice lemma, is HVZK and has WEE, the EFLRSL signature is unforgeable and anonymous. We provide a proof sketch for this, mostly referring to the work in [15, 17, 8, 10, 20], where the situation is similar.

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

Lin2-2Choice lemma is an extended version of the Lin2-Choice lemma; its protocol comprises l instances of the Lin2-Choice lemma 1-out-of-many membership proof, each one extended so that it selects a linear combination of exactly two ring elements instead of one. All together optimized.

This proof 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 commitments $\mathbf{Z} = \{Z_k\}_{k=0}^{l-1}$, using the Lin2-2Choice lemma protocol it is possible to convince verifier that, for each $Z_k \in \mathbf{Z}$, there holds $Z_k = p_k P_{s_k} + v_k V_k$ for some p_k, v_k, s_k known to 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 WEE, and also informally show it is HVZK, referring to the similar design in [7]. The lemma's protocol appears to be so generic that later on we use it to substitute linking tag $x\mathcal{H}_{\mathbf{point}}(xG)$ for $x^{-1}\mathcal{H}_{\mathbf{point}}(xG)$ in the Multratug signature.

1.4.4 HELPER ARGUMENT: RANDOM WEIGHTING FOR T-S TUPLES

Suppose, we have two tuples, possibly blinded. Taking their inner products with a random scalar vector, we wonder if showing that these inner products are proportional to each other will prove that the tuples are elementwise proportional to each other. This question emerged in one of our proofs. We have looked in the existing literature and have not found any answer.

Therefore, we compiled an appropriate argument, defined sufficient conditions, and presented the answer in this paper. It is that, roughly speaking, 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 there holds $\mathbf{D} = a\mathbf{T}$.

1.4.5 MULTRATUG SIGNATURE WITH BALANCE PROOF

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

$$2\lceil \log_2(n+1) \rceil + 6l + 4.$$

We provide a proof sketch for its unforgeability and anonymity, and prove correctness of its balance in detail.

Application area of Multratug includes the EFLRSL domain, extending it with support for hidden amounts and multisignature operations. Multratug is suitable for blockchains. Since modern blockchains usually require multisignature operations, it makes sense to compare Multratug only with signatures that allow them (column 'MP' in Table 1), the full comparison results are shown in Table 8 and Table 9.

1.5 PREVIEW OF CORE PROTOCOLS

1.5.1 LIN2-CHOICE LEMMA'S MEMBERSHIP PROOF

For a ring $\mathbf{P} = \{P_i\}_{i=0}^{n-1}$ and a commitment Z, the Lin2-Choice lemma protocol proves membership of Z in $\mathbf{P}$. In a nutshell, 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 random δ. Finally, the prover convinces the verifier using an arbitrary vector commitment argument that the element $\hat{Z}$ built as

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

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

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

Of course, the vector commitment argument is to be HVZK and has to have WEE. Also, note, the commitment Z and all elements published by prover are blinded, we omit showing the blinding components for simplicity.

It appears to be that the above game completes successfully only if there is some scalar p known to the prover such that $p^{-1}Z \in \mathbf{P}$. The Lin2-Choice lemma guarantees this.

Getting ahead, by adding the linking tag $x^{-1}\mathcal{H}_{\mathbf{point}}(xG)$ into this proof of membership, we obtain the EFLRSL signature.

1.5.2 LIN2-2CHOICE LEMMA'S MEMBERSHIP PROOF

Compared to the Lin2-Choice lemma's simplified game, 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), $k \in [0 \ldots l-1]$, the verifier releases $\mathbf{c} = \{c_i\}_{i=0}^{n+l-1}$, the prover replies with l scalars r_k, $k \in [0 \ldots l-1]$, the verifier releases random δ_1, δ_2. The prover convinces the verifier that, for each $k \in [0 \ldots l-1]$, 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

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

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

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

with random coefficients λ_k's, is the weighted sum of elements from the above set.

The Lin2-2Choice lemma guarantees this game 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 $Z_k \in \mathbf{Z}$, there holds

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

1.5.3 PIVOT: OPTIMIZED VECTOR COMMITMENT ARGUMENT

Our membership proofs call it directly or indirectly, although, of course, they can call any other complete, HVZK, and having WEE implementation of it. Nevertheless, we put its preview here since it is the pivotal protocol in our paper. Although, since it is almost exactly the same as the compressed pivot with $L \equiv \varnothing$ in [2], it is possible to just understand it from [2].

The idea is that initially we build a complete, HVZK, and having WEE linear-size Schnorr-like vector commitment argument that convinces verifier that Y is a weighted sum 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 and prover knows the weights. Namely, the prover publishes an element T as the first message, the verifier issues challenge c, the prover replies with a scalar vector $\boldsymbol{\tau}$, the verifier checks that $\langle \boldsymbol{\tau}, \mathbf{X} \rangle + cY = T$. The game is n Schnorr identification protocol games [19], for each $X_i \in \mathbf{X}$, joined together. This follows from the fact that Y and T are necessarily weighted direct sums of $\mathbf{X}$, otherwise it can be shown that the orthogonality of $\mathbf{X}$ is 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 [5] is used. The proof need not be HVZK, as $\boldsymbol{\tau}$ itself does already reveal nothing. Thus, we obtain a complete, HVZK, and WEE optimized vector commitment argument of size $2\lceil \log_2(n) \rceil + 1$.

A minor addition, hereinafter we always have Y blinded. The blinding generator 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 and denote as H. As H is orthogonal to $\mathbf{X}$, we implicitly append H to $\mathbf{X}$ and, hence, the size of the argument becomes $2\lceil \log_2(n+1) \rceil + 1$.

1.5.4 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 [17]. 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 factors) amounts corresponding to the actual signing indices and, simply put, add them to the end of the ring.

Also, 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, the simplified game for Multratug is that at the start both of the prover and verifier have $\mathbf{P}, \mathbf{A}, \mathbf{A^{tmp}}$, and 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 it can be easily achieved by adding their hashes-to-curve, we omit showing them in this preview. After making an 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, using the Lin2-2Choice lemma membership proof, the prover convinces the verifier that it knows s_k, p_k such that

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

This equality splits into $p_{s_k}^{-1} G = P_{s_k} \in \mathbf{P} \wedge A_{s_k} = A_k^{\mathbf{tmp}}$. Of course, we omit blinding components in this preview. Also, we assume all elements in $\mathbf{P}$ are validated nonzero.

Thus, for all k's, these equalities 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, it only remains to check that $\sum_{k=0}^{l-1} A_k^{\mathbf{tmp}} = A^{\mathbf{sum}}$, and the Multratug signature with the balance proof is ready.

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 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

All our protocols, including the helpers schemes and signatures, perform for a prime-order groups without bilinear pairings in a trustless environment under the DDH assumption in the random oracle model. All the context, namely, the common reference string, trustless setup, assumptions, discrete logarithm (DL) relation assumption, orthogonality definition, non-interactivity through Fiat-Shamir heuristic, special honest verifier zero-knowledge (HVZK) and computational witness-extended emulation (WEE) proof methods, which we use, are exactly the same as in [5, 7]. 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 [5].

As a syntactic sugar we use the shorthands '∼', 'lin', 'ort' defined in [20], although they can be resolved and omitted. Also, we use additive notation for exponentiation of group elements as in [17, 20]. For all protocols, we imply the existence of non-interactive Fiat-Shamir counterparts not mentioning them. We have collected in [20] the existing definitions of linkable ring signature, its variations and security models from various sources; we use these definitions hereinafter, with the only one difference in that what in [20] is called a generic linkable ring signature now we simply call a linkable ring signature.

2.2 NOTATION

Here is a list of basic notations

- $\mathbb{G}$ 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 could be appended, e.g., a, b_{12}, c', ζ', x_k. Also, lowercase italic letters can be used to designate integers used as indices or limits, e.g., n, i, j_1, s_k, this usage is clear from the context. Superscripts, e.g., ϵ^2, denote scalar exponentiation.
- a special case is a lowercase italic letter with a bold superscript, e.g., $d^{\mathbf{\Delta sum}}$, this denotes a regular scalar of $\mathbb{F}_{\bar{\mathsf{p}}}$, and the superscript in bold is purely explanatory.
- 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., $\mathfrak{a}$.
- uppercase italic letters denote elements in $\mathbb{G}$. Apostrophes, hats, and subscript indices could be appended, e.g., A, B_{12}, D', P_{s_k}. Multiplication is used to denote element exponentiation by a scalar, e.g., xG.
- a special case is an uppercase italic letter with a bold superscript, e.g., $A^{\mathbf{sum}}$, this denotes a regular element of $\mathbb{G}$, and the superscript in bold is purely explanatory.
- 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 $\mathbb{F}_{\bar{\mathsf{p}}}$ without the scalar 0, $\mathbb{G}^*$ means $\mathbb{G}$ without the element 0. Substantially, for vectors, if $\mathbf{x} \in \mathbb{F}_{\bar{\mathsf{p}}}^{n*}$, $\mathbf{P} \in \mathbb{G}^{m*}$, 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 M is a bitstring.
- $\mathcal{H}_{\mathbf{scalar}}$ and $\mathcal{H}_{\mathbf{point}}$ are the ideal hash and hash to group element (to curve) functions, respectively.
- $A = \text{lin}(\mathbf{B})$, where $\mathbf{B}$ is a non-empty vector of nonzero elements, means there is a known vector $\mathbf{x}$ such that $A = \langle \mathbf{x}, \mathbf{B} \rangle$. 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 weights in A's representation as a weighted sum of elements in $\mathbf{B}$ cannot be found. Syntactic 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 a non-trivial relation [5] between elements in $\mathbf{S}$ cannot be found. This is in accordance with [20]. 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, $\text{ort}(\mathbf{S})$ actually means that, for each element $E \in \mathbf{S}$, no one in the

system knows weights in E's representation as a weighted sum of elements in $\mathbf{S} \setminus \{E\}$. Note, if $\mathbf{S}$ contains the zero element, then $\mathrm{ort}(\mathbf{S})$ never holds.

- we say that all elements in $\mathbf{S}$ are orthogonal to each other, iff $\mathrm{ort}(\mathbf{S})$ holds. We emphasize this because 'orthogonal to each other' can be read as pairwise orthogonality, which certainly is a weaker property. Hereinafter, writting that elements in $\mathbf{S}$ are ortogonal to each other we always imply the stronger property, namely, that $\mathrm{ort}(\mathbf{S})$ holds.
- $\mathrm{nz}(\mathbf{B})$ 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 [5]. Also, having a vector, say, $\mathbf{A}$, 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 explicitly 'let $A_i \leftarrow \mathbf{A}_{[i]}$', although the equality is already implied.
- appending an element into a vector is denoted by comma, e.g., $\hat{\mathbf{X}} \leftarrow [\mathbf{X}, B]$ means that $\hat{\mathbf{X}} = [X_0, X_1, \dots, X_{n-1}, B]$.
- writing down our protocols we mix several assignment styles, they all are construed as an imperative assignment. That is, e.g., 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 value of y and both 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.
- everywhere $\log_2(\dots)$ is meant as its ceiling $\lceil \log_2(\dots) \rceil$ when used together with integers in formulas.

2.3 COMMONLY AVAILABLE INFORMATION

With the above notation all the common information available from the beginning to both $\mathcal{P}$ and $\mathcal{V}$ and silently presented in all protocols is shown in Figure 1.

`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, which we then use as building blocks for our signatures. For each of the protocols, we are interested in the three properties, namely, in completeness, HVZK, and WEE.

Completeness is seen from code of the protocols, we do not dwell on it. The HVZK property requires building a simulator; and each of our protocol has a by-design property which simplifies things. Namely, besides the fact that all scalars of the protocol transcripts are masked with independent and uniformly sampled summands, each element of the transcripts, if it is not a completely dependent one, has the form

$$X + \mu H, \tag{1}$$

where X is a semantic component of the element, H is a blinding generator built in such a way as to be clearly orthogonal to everything else, and μ is always an independent and uniformly sampled scalar. Therefore, we refer to the work [7], where the situation is the same and a simulator is constructed. The intuition here is that the form (1) is the Pedersen commitment [18], which is perfectly hiding [5]. We imply that for each of our protocols a simulator is constructed in the same way as in [7]. Also, as far as we can see, e.g. from [2], these days it is a common method of rendering a protocol to zero-knowledge, so we do not dwell on it either.

For each of our protocols, we prove its WEE property in detail by constructing an extractor that restores witness by performing polynomial number of rewindings. For some elementary protocols, we instead refer to the works where such the details can be found. We also prove that the obtained witness meets the limits specified in protocol's relation, otherwise the extractor breaks the DL relation assumption in a polynomial number of steps.

2.4.1 CONNECTION TO SIGNATURES

Thus, by the above, each of our signatures relies on a complete, HVZK, and WEE underlying proving system. Therefore, to establish unforgeability, anonymity, and other properties of our signatures, we refer to the work in [15, 8, 20] where these properties are obtained from the HVZK and WEE properties of the undrlying proving systems for the signatures with key images $x\mathcal{H}_{\mathbf{point}}(xG)$ and $x^{-1}\mathcal{H}_{\mathbf{point}}(xG)$.

2.4.2 EXCEPTIONAL HVZK CASES

We have the only two exceptional HVZK protocols in this paper, which does not follow the form (1) for their public transcript elements. The first of them is the two-element Schnorr-like scheme in Section 3.2.1, which splits into two Schnorr-id protocols and, hence, can be proven HVZK by combining outputs of two Schnorr-id simulators.

The second one is the optimized version of our pivot vector commitment argument in Figure 27, previewed in Section 1.5.3. It is HVZK 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 HVZK with open $\boldsymbol{\tau}$, and the replacement of $\boldsymbol{\tau}$ with its proof of knowledge does not revoke HVZK property of the entire argument. More details can be found in the proof of the HVZK property for the pivot scheme in [2].

3 RUDIMENTARY PROTOCOLS

Here we present simple protocols which prove knowledge of witnesses for basic relations. We will then use them to construct our lemmas and signatures. Although, generally speaking, they can be used independently or as parts of other systems. And, concrete implementations of those in Section 3.1.1 and in Section 3.1.2 are not decisive; any other implementations can be used for both of them, as long as they prove the same relations and are complete, HVZK, and have WEE.

In the section names below, we often omit the word ‘argument’ when we match a type of commitment with its corresponding argument.

3.1 OVERVIEW

3.1.1 TWO ELEMENT COMMITMENT

The first helper sub-protocol is a two-element commitment argument. We denote it as

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

In this notation, the elements X, H, Y are common input for prover and verifier. And x, h are prover's private input, they are witnesses known only for it. The $\texttt{zk2ElemComm}(X, H, Y;\ x, h)$ argument proves the relation

$$\mathcal{R} = \{\, X, H \in \mathbb{G}^*, Y \in \mathbb{G};\ x, h \in \mathbb{F}_{\bar{\mathsf{p}}} \mid Y = xX + hH \,\}, \tag{2}$$

where X and H are orthogonal to each other. Also, we require the argument to be HVZK and WEE. In Figure 2 we provide an uncomplicated implementation for it.

Overall, $\texttt{zk2ElemComm}(X, H, Y;\ x, h)$ convinces verifier that prover knows a representation of element Y as a weighted sum of orthogonal generators X and H with weights known to prover. We use a two-generator extension of the Schnorr identification scheme as an implementation of this proof. Its size is one element in $\mathbb{G}$ and two scalars in $\mathbb{F}_{\bar{\mathsf{p}}}$.

3.1.2 BASIC VECTOR COMMITMENT

Vector commitment argument, which will be a pivot in this paper,

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

provides a proof for the relation

$$\mathcal{R} = \{\, \mathbf{X} \in \mathbb{G}^{n*}, H \in \mathbb{G}^*, Y \in \mathbb{G};\ \mathbf{a} \in \mathbb{F}^n_{\bar{\mathsf{p}}}, \alpha \in \mathbb{F}_{\bar{\mathsf{p}}} \mid Y = \langle \mathbf{a}, \mathbf{X} \rangle + \alpha H \,\}, \tag{3}$$

where all generators from the set $\mathbf{X} \cup \{H\}$ are orthogonal to each other. That is, $\texttt{zkVC}_n$ 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 turn the protocol into zero-knowledge, as in [7].

Our implementation of $\texttt{zkVC}_n$ in Figure 3 is based on the inner product argument implementation from [5] for the relation

$$\mathcal{R} = \{\, \mathbf{G}, \mathbf{H} \in \mathbb{G}^{n*}, U, P \in \mathbb{G};\ \mathbf{a}, \mathbf{b} \in \mathbb{F}^n_{\bar{\mathsf{p}}} \mid P = \langle \mathbf{a}, \mathbf{G} \rangle + \langle \mathbf{b}, \mathbf{H} \rangle + \langle \mathbf{a}, \mathbf{b} \rangle\, U \,\}, \tag{4}$$

which we modify as follows. First, since we do actually not need in the inner product argument, just only in its vector commitment part, we zero out the vector **b** in the relation (4), making the inner product $\langle \mathbf{a}, \mathbf{b} \rangle$ equal to zero everywhere and leave only the vector commitment, i.e., only the argument for the relation

$$\mathcal{R} = \{\, \mathbf{G} \in \mathbb{G}^{n*},\ P \in \mathbb{G};\ \mathbf{a} \in \mathbb{F}_{\bar{p}}^n \mid P = \langle \mathbf{a}, \mathbf{G} \rangle \,\}. \tag{5}$$

Second, we add zero-knowledge property to the inner product argument not the way it is done in [5], instead we add it in a straighter way, as in [7]. That is, we respectively add the blinding summands αH, βH, and γH to the vector commitment P and to all the L and R elements that are transmitted during the reduction taken from [5]. The secret factors α, β, γ are uniformly sampled from $\mathbb{F}_{\bar{p}}^*$, the generator H is chosen independently, and thus P and all the transmitted L's and R's appear indistinguishable from random noise. We rename the vector **G** and the commitment P in the relation (5) as **X** and Y in the relation (3), respectively. The blinding summand αH is taken into account in the relation (3).

Third, for the case $n = 1$ we use our own Schnorr-like HVZK and WEE protocol, which is different from sub-protocols used in [5] and [7]. Namely, we use zk2ElemComm for the case, and this does not alter the properties of the entire $\mathtt{zkVC}_n$ protocol. In any case, any complete, HVZK and WEE protocol that proves $Y = \mathrm{lin}(X_0, H)$ will do instead of zk2ElemComm for $n = 1$ in $\mathtt{zkVC}_n$.

Thus, our $\mathtt{zkVC}_n$ implementation of the vector commitment argument in Figure 3 has the same properties as the implementation of the inner product argument in [5] with $\mathbf{b} = \mathbf{0}^n$, plus it is HVZK and, of course, it remains to be having WEE.

If we compare our $\mathtt{zkVC}_n$ protocol with the weighted inner product argument from [7], which is also based on the inner product argument from [5], then just as in the comparison with the inner product argument from [5] we zero out the vector **b**, thus making the weighted inner product $\mathbf{a} \odot_y \mathbf{b}$ equal to zero. In doing so, we assume the weight y equal to 1 everywhere, and also use zk2ElemComm for the case $n = 1$.

Note, actually our implementation of $\mathtt{zkVC}_n$ is not based on the weighted inner product argument of [7], since we use neither 'weighted' in the sense of [7] nor 'inner product'. From the work in [7] we only use the way we turn our argument into zero-knowledge, type of notation that we find concise and convenient, and also we borrowed from [7] the idea of using a custom Schnorr-like protocol for $n = 1$.

In sum, our zero-knowledge vector commitment argument $\mathtt{zkVC}_n$ size is $2\lceil \log_2(n) \rceil + 1$ elements in $\mathbb{G}$ and 2 scalar in $\mathbb{F}_{\bar{p}}$. Here and elsewhere, when we use this implementation of $\mathtt{zkVC}_n$ we consider n is a power of 2. Although, as we have already noted, we are not generally bound to a particular realization of $\mathtt{zkVC}_n$, hence when we use an optimized implementation of it, such as that defined in Section 10, this requirement for n changes.

3.1.3 RANDOM WEIGHTING FOR 3-TUPLES

Another auxiliary argument,

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

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 first triplet can be zero, in which case the other two elements of the triplet (P, Q, R) must be orthogonal to each other. So, the protocol zk3ElemRW proves the following relation

$$\mathcal{R} = \left\{ \begin{array}{l} P \in \mathbb{G}^*,\ Q, R \in \mathbb{G},\ H \in \mathbb{G}^*,\ Z, F, E \in \mathbb{G}; \\ a, \alpha, \beta, \gamma \in \mathbb{F}_{\bar{p}} \end{array} \middle| \begin{array}{l} Z\ = aP + \alpha H\ \wedge \\ F\ = aQ + \beta H\ \wedge \\ E\ = aR + \gamma H \end{array} \right\}, \tag{6}$$

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

There are two sampled challenges δ_1 and δ_2 within the protocol zk3ElemRW. The two sums X and Y together with total blinding factor $\hat{\alpha}$ are constructed with these challenges

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

As the second step, using an arbitrary complete, HVZK, and having WEE argument it is proved that Y is a weighted sum of X and H with some known to prover weights. Thus, the relation (6) is proved.

In terms of [20], in the second step of zk3ElemRW a proof of $Y = \mathrm{lin}(X, H)$ for prover is somehow obtained (in an HVZK and WEE way). We will be often omitting everything connected with H as a technical blinding detail, so writting down this shortly as $Y \sim X$ (to the accuracy of H). The WEE property of the protocol zk3ElemRW can be proved exactly the same way, as for the RandomWeighting-WEE lemma protocol in [20]. The extreme case, when one of the elements Q or R is zero, is not problematic.

3.1.4 SIMMETRIC VECTOR COMMITMENT

We will also need an argument to convince verifier that several, e.g., two or three, vector commitments share the same weights, with the only exclusion for blinding factors, known to prover. That is, we will need an argument

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

for the following relation

$$\mathcal{R} = \left\{ \begin{array}{l} \mathbf{P} \in \mathbb{G}^{n*}, \mathbf{Q}, \mathbf{R} \in \mathbb{G}^{n}, H \in \mathbb{G}^{*}, Z, F, E \in \mathbb{G}; \\ \mathbf{a} \in \mathbb{F}_{\bar{\mathsf{p}}}^{n}, \alpha, \beta, \gamma \in \mathbb{F}_{\bar{\mathsf{p}}} \end{array} \middle| \begin{array}{l} Z\ = \langle \mathbf{a}, \mathbf{P} \rangle + \alpha H\ \wedge \\ F = \langle \mathbf{a}, \mathbf{Q} \rangle + \beta H\ \wedge \\ E\ = \langle \mathbf{a}, \mathbf{R} \rangle + \gamma H \end{array} \right\}, \tag{7}$$

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 as

$$\operatorname{ort}(\mathbf{P} \cup \operatorname{nz}(\mathbf{Q}) \cup \operatorname{nz}(\mathbf{R}) \cup \{H\}),$$

and where for any index $i \in [0 \dots n-1]$ at least one of two elements $\mathbf{Q}_{[i]}$ and $\mathbf{R}_{[i]}$ is nonzero, denoted as

$$(\mathbf{Q} + \mathbf{R}) \in \mathbb{G}^{*}.$$

The relation (7) states that the three different vector commitments Z, F, E are sort of 'symmetrical' to each other by their common weights $\mathbf{a}$, which apply to the three different bases $\mathbf{P}, \mathbf{Q}, \mathbf{R}$, respectively. The protocol $\mathtt{zkSVC}_{3,n}$ is shown in Figure 5.

Note, that we require all elements in $\mathbf{P}$ to be nonzero, while vectors $\mathbf{Q}$ and $\mathbf{R}$ can 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 one imposed by `zk3ElemRW` in Section 3.1.3.

Using random weights similar to the way they are used in Section 3.1.3, we reduce the argument $\mathtt{zkSVC}_{3,n}$ to the vector commitment argument $\mathtt{zkVC}_n$. 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

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

After $\mathtt{zkVC}_n$ successful completion, as a result, we see that by this n instances of the protocol `zk3ElemRW` have been successfully performed, for all the indices $i \in [0 \dots n-1]$. This means that the relation (6) is fulfilled for each triplet pair (P_i, Q_i, R_i) and $(Z_{P_i}, F_{Q_i}, E_{R_i})$ and, therefore, the relation in question (7) is fulfilled.

Here 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 prover. Of course, upon successful completion of $\mathtt{zkSVC}_{3,n}$, verifier is also convinced of this. 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 `zk2ElemComm` *in Figure 2 is a complete, HVZK argument having WEE for the relation (2).*

Proof: Appendix A.
Overview: Section 3.1.1.

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

Relation $\mathcal{R} = \{\, X, H \in \mathbb{G}^*, Y \in \mathbb{G};\ x, h \in \mathbb{F}_{\bar{\mathsf{p}}} \mid Y = xX + hH \,\}$ // (2)

// 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

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

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 there holds $\mathrm{ort}(\mathbf{X} \cup \{H\})$, for an element $Y \in \mathbb{G}$, the protocol $\texttt{zkVC}_n$ in Figure 3 is a complete, HVZK argument having WEE for the relation (3).

Proof: Appendix B.
Overview: Section 3.1.2.

```
$\boxed{\mathtt{zkVC}_n(\mathbf{X}, H, Y;\ \mathbf{a}, \alpha)}$
Relation $\mathcal{R} = \{\, \mathbf{X} \in \mathbb{G}^{n*}, H \in \mathbb{G}^*, Y \in \mathbb{G};\ \mathbf{a} \in \mathbb{F}_{\bar{p}}^n, \alpha \in \mathbb{F}_{\bar{p}} \mid Y = \langle \mathbf{a}, \mathbf{X} \rangle + \alpha H \,\}$   // (3)
  // $\mathbf{X}, H$ 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 there holds $\mathrm{ort}(\mathrm{nz}(P, Q, R, H))$ *and at least one of the two elements Q, R is nonzero, the protocol* `zk3ElemRW` *in Figure 4 is a complete, HVZK argument having WEE for the relation (6).*

Proof: Appendix C.
Overview: 3.1.3.

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

$$\text{Relation } \mathcal{R} = \left\{ \begin{array}{l|l} P \in \mathbb{G}^*,\ Q, R \in \mathbb{G},\ H \in \mathbb{G}^*,\ Z, F, E \in \mathbb{G}; & Z\ = aP + \alpha H\ \wedge \\ a, \alpha, \beta, \gamma \in \mathbb{F}_{\bar{\mathsf{p}}} & F\ = aQ + \beta H\ \wedge \\ & E\ = aR + \gamma H \end{array} \right\} \quad // \, (6)$$

// 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, HVZK, and WEE protocol that convinces $\mathcal{V}$ that

a, α, β, γ at $\mathcal{P}$'s privare input connect X and Y so that $Y = aX + \hat{\alpha} H$

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 there holds $\mathrm{ort}(\mathbf{P} \cup \mathrm{nz}(\mathbf{Q}) \cup \mathrm{nz}(\mathbf{R}) \cup \{H\})$, *for three elements $Z, F, E \in \mathbb{G}$, the protocol $\texttt{zkSVC}_{3,n}$ in Figure 5 is a complete, HVZK argument having WEE for the relation (7).*

Proof: Appendix D.
Overview: 3.1.4.

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

$$\text{Relation } \mathcal{R} = \left\{ \begin{array}{l|l} \mathbf{P} \in \mathbb{G}^{n*}, \mathbf{Q}, \mathbf{R} \in \mathbb{G}^n, H \in \mathbb{G}^*, Z, F, E \in \mathbb{G}; & Z\ = \langle \mathbf{a}, \mathbf{P} \rangle + \alpha H\ \wedge \\ \mathbf{a} \in \mathbb{F}^n_{\bar{\mathsf{p}}}, \alpha, \beta, \gamma \in \mathbb{F}_{\bar{\mathsf{p}}} & F\ = \langle \mathbf{a}, \mathbf{Q} \rangle + \beta H\ \wedge \\ & E\ = \langle \mathbf{a}, \mathbf{R} \rangle + \gamma H \end{array} \right\} \quad // \, (7)$$

// $\mathbf{P}, \mathbf{Q}, \mathbf{R}, H$ 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, HVZK, and WEE protocol for the relation (3)

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

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

```
$\boxed{\mathtt{zkSVC}_{2,n}(\mathbf{P}, \mathbf{Q}, H, P, Q;\ \mathbf{a}, \alpha, \beta)}$
$\mathtt{zkSVC}_{2,n}(\mathbf{P}, \mathbf{Q}, H, Z, F;\ \mathbf{a}, \alpha, \beta) = \mathtt{zkSVC}_{3,n}(\mathbf{P}, \mathbf{Q}, \mathbf{0}^n, H, Z, F, 0;\ \mathbf{a}, \alpha, \beta, 0)$
  // where $\mathbf{P}, \mathbf{Q} \in \mathbb{G}^{n*},\ H \in \mathbb{G}^*,\ Z, F \in \mathbb{G};\ \mathbf{a} \in \mathbb{F}_{\bar{\mathrm{p}}}^n,\ \alpha, \beta, \gamma \in \mathbb{F}_{\bar{\mathrm{p}}}$
```

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 1-out-of-many proof of membership $\mathtt{zkLin2Choice}_n$, which we will use later to create ring signatures.

4.1 OVERVIEW

In [20] we proved the Lin2-Xor lemma which, informally, allows one to select a pair of elements from two pairs of elements, i.e., it provides an argument for the relation

$$\mathcal{R} = \left\{\ \mathbf{P}, \mathbf{Q} \in \mathbb{G}^{2*}, Z \in \mathbb{G}^*; s \in [0 \ldots 1],\ p, q \in \mathbb{F}_{\bar{\mathrm{p}}}\ \middle|\ Z = pP_s + qQ_s\ \right\}, \tag{8}$$

where the generators of $\mathbf{P} \cup \mathbf{Q}$ are orthogonal to each other. Also, in [20] by successive application of the Lin2-Xor lemma $\log_2(n)$ times we proved the Lin2-Selector lemma, which allows to select one pair of elements from n pairs of elements. In other words, the Lin2-Selector lemma [20] provides an argument for the relation

$$\mathcal{R} = \left\{\ \mathbf{P}, \mathbf{Q} \in \mathbb{G}^{n*}, Z \in \mathbb{G}^*; s \in [0 \ldots n-1],\ p, q \in \mathbb{F}_{\bar{\mathrm{p}}}\ \middle|\ Z = pP_s + qQ_s\ \right\}. \tag{9}$$

However, after some consideration, we concluded that instead of proving the relation (9) by the Lin2-Selector lemma protocol, it is 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 vector commitment argument. This way is more efficient in size, and also gives more opportunities to optimize the verification complexity.

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 the two original pairs becomes a compound element with its random 'rotation'. Namely, they become $P_0 + c_0Q_0$ and $P_1 + c_1Q_1$. Here we use the notation and indexing from [20]. In the second round of the Lin2-Xor protocol, the prover and verifier play a sub-protocol convincing the verifier that the element $Z + r_1H_1$ is a linear combination of the two compound elements, which carry their random 'rotations' c_0 and c_1. It then turns out that this linear combination can be only one-hot, otherwise the DL relation assumption would be broken.

Indeed, 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 $Z + r_1H_1$ has at most one 'degree of freedom' parameterized by r_1. At the same time, each of the elements $P_0 + c_0Q_0$ and $P_1 + c_1Q_1$ 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 linear combination

$$Z + r_1H_1 = a(P_0 + c_0Q_0) + b(P_1 + c_1Q_1) \tag{10}$$

are not equal to zero, then the right-hand side of the equality (10), which has two 'degrees of freedom' with the random parameters c_0 and c_1, is balanced by one 'degree of freedom' of the left-hand side with the controlled parameter r_1, which is impossible without breaking orthogonality of P_0, Q_0, P_1, Q_1.

In line with this intuition, we can take n pairs of elements and turn them into n compound elements with random 'rotations' in the first round. After that, in the second round, we can prove that $Z + r_1H_1$ is a linear combination of these n compound elements. As a result, exactly the same way as for the linear combination (10), we obtain that the compound element $Z + r_1H_1$ with one 'degree of freedom' r_1 must balance the weighted sum of the compound elements of the form $P_i + c_iQ_i$, each adding one 'degree of freedom' to the right side of the equality

$$Z + r_1H_1 = \sum_{i=0}^{n-1} a_i(P_i + c_iQ_i), \tag{11}$$

which is possible only if the vector of coefficients $\{a_i\}_{i=0}^{n-1}$ is one-hot. Thus, we obtain an argument for the relation (9) as a 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

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

is played. Here H_1 is fixed as in [20], H is an independent generator for blinding, α is the blinding factor, and **a** is one-hot.

Also, since the vector **Q** carries only a technical role in the relation (9), in particular in [20] we get rid of Q_s by adding a proof of that $q = 0$ everywhere in the signatures, we now include a proof of $q = 0$ in our argument. Taking everything into account, in the Lin2-Choice lemma (Theorem 5) we provide the protocol

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

shown in Figure 7, which is HVZK, has WEE, and proves (knowledge of witness) for the following relation

$$\mathcal{R} = \left\{ \begin{array}{l} \mathbf{P}, \mathbf{Q} \in \mathbb{G}^{n*}, H \in \mathbb{G}^*, Z \in \mathbb{G}; \\ s \in [0 \ldots n-1], p, \alpha \in \mathbb{F}_{\bar{p}} \end{array} \;\middle|\; Z = pP_s + \alpha H \right\}, \tag{12}$$

where $\mathbf{P}, \mathbf{Q}, H$ meet $\mathrm{ort}(\mathbf{P} \cup \mathbf{Q} \cup \{H\})$. Thus, our Lin2-Choice lemma allows to choose exactly one element from the set of orthogonal elements $\mathbf{P} \in \mathbb{G}^{n*}$.

Turning to the details, with the simultaneous proof of $q = 0$, the Lin2-Choice lemma protocol $\mathtt{zkLin2Choice}_n$ for the relation (12) is constructed as follows

- After the first $\mathcal{P}$'s message both $\mathcal{P}$ and $\mathcal{V}$ have elements Z and F, where F plays the same role as H_1 in [20].
- All n elements of **Q** are multiplied by the challenges $\{c_i\}_{i=0}^{n-1}$ respectively, so $\mathcal{P}$ and $\mathcal{V}$ build a vector of elements $\hat{\mathbf{Q}} = \{c_i Q_i\}_{i=0}^{n-1}$.
- $\mathcal{P}$ replies with r, which plays the same role as r_1 in [20].
- $\mathcal{P}$ and $\mathcal{V}$ play $\mathtt{zkSVC2}_{2,n}(\mathbf{P}, \hat{\mathbf{Q}}, H, Z, rF; \mathbf{a}, \alpha, r\beta)$, where **a** is one-hot, H is an orthogonal blinding generator, α and β are blinding factors of Z and F respectively.

Informally, we can see that if **a** has more than one hot entry, then $\mathtt{zkSVC}_{2,n}$ will not complete successfully for the same reason as the equality (11) will not hold for such **a**. To be precise, the following equality is checked within $\mathtt{zkSVC}_{2,n}$, and it guarantees **a** is one-hot

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

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

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 there holds $\mathrm{ort}(\mathbf{P} \cup \mathbf{Q} \cup \{H\})$, for an element $Z \in \mathbb{G}$, the protocol $\mathtt{zkLin2Choice}_n$ in Figure 7 is a complete, HVZK argument having WEE for the relation (12).

Proof: Appendix E.
Overview: Section 4.1.

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

$$\text{Relation } \mathcal{R} = \left\{ \begin{array}{l} \mathbf{P},\mathbf{Q} \in \mathbb{G}^{n*}, H \in \mathbb{G}^*, Z \in \mathbb{G}\,;\\ s \in [0\dots n-1], p, \alpha \in \mathbb{F}_{\bar{\mathsf{p}}} \end{array} \;\middle|\; Z = pP_s + \alpha H \right\} \quad // \text{ (12)}$$

// $\mathbf{P},\mathbf{Q},H$ in $\mathcal{R}$ satisfy 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*

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

$F = qQ_s + \beta H$

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

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

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

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: 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 $r \leftarrow\!\!\$\ \mathbb{F}^*_{\bar{\mathsf{p}}}$,

assigns **if** $p \neq 0$ **then** $r = c_s p/q$ **endif**

$\hat{\beta} = r\beta$,

and lets $\mathbf{a} = \left\{ \begin{array}{l} a_s = p \quad // \text{ that is, } p \text{ is at } s\text{'th position in one-hot } \mathbf{a} \text{ (or, if } p = 0\text{, then } \mathbf{a} = \mathbf{0}^n\text{)} \\ a_i = 0 \text{ for all } i \in [0\dots n-1], i \neq s \end{array} \right.$

$\boxed{\mathcal{P} \to \mathcal{V}}$: 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 7: 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 1 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 8. Using it both of the prover and verifier have identical definitions of the hash $\mathcal{H}_{\mathbf{scalar}}$ and hash-to-group $\mathcal{H}_{\mathbf{point}}$ functions, as well as a common set of orthogonal generators $\mathbf{G}$.

$\boxed{\texttt{Additional common information}}$

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

Figure 8: Additional information available to each party

The hash function $\mathcal{H}_{\mathbf{scalar}}$ models the random oracle, whereas the hash-to-group (-to-curve) function $\mathcal{H}_{\mathbf{point}}$ is used to generate a brand new element orthogonal to the set of already exposed ones. Using the predefined set of genarators $\mathbf{G}$ we reduce the signature verification complexity.

All public keys of the signatures can be known to all participants, and there are no additional restrictions on them. That is, in fact, we do not impose any rules on public keys, which is reflected in Figure 9.

`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 the signatures.

Figure 9: Public keys seen to all parties

5.2 OVERVIEW

Having the zero-knowledge argument $\texttt{zkLin2Choice}_n$ for the relation (12), it is easy to build a ring signature, we call it EFLRS1 (Efficient linkable ring signature for 1 actual signer), its interactive scheme is shown in Figure 10,

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

By the ring we mean a set of n public keys

$$\mathbf{P} = \{P_i\}_{i=0}^{n-1}\,, \tag{13}$$

where $n \geqslant 1$. The 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 are no assumptions about the public keys in $\mathbf{P}$, all they can be regarded as adversarially chosen.

By the corresponding to the ring decoy set, technically called so, we will mean a set of n pairs of the form

$$\{\,(\,P_i + \zeta\mathcal{H}_{\mathbf{point}}(P_i),\ Q_i\,)\,\}_{i=0}^{n-1}\,, \tag{14}$$

where P_i is a public key in the ring, ζ is a random weight, $\mathcal{H}_{\mathbf{point}}$ is the hash-to-curve function, and $Q_i \in \mathbf{Q}$, where $\mathbf{Q}$ is a set of auxiliary orthogonal generators that can be prepared in advance, provided that $\mathcal{H}_{\mathbf{point}}$ always generates elements orthogonal to $\mathbf{Q}$.

At the same time, key image is defined as

$$I = x^{-1}\mathcal{H}_{\mathbf{point}}(P_s), \tag{15}$$

where x is a private key for the public key $P_s \in \mathbf{P}$ such that there holds $P_s = xG$.

To obtain a signature it remains to define Z as

$$Z = G + \zeta I, \tag{16}$$

pick the blinding generator H as to be orthogonal to all the other used generators, and to apply the protocol of the Lin2-Choice lemma as follows

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

thus producing the signature of size $2\lceil\log_2(n)\rceil + 6$.

When calculating the signature size we assume that bitwise representation of an element from $\mathbb{G}$ takes as much space as bitwise representation of a scalar from $\mathbb{F}_{\bar{p}}$. We take into account all elements and scalars transmitted from prover to verifier, including the key image I. We ignore 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 the signature signs an input message M for the first place, we use the well-known method of binding a signature to a message, described, e.g., in [10]. Namely, we assume that the signature's random oracle depends on the input message, and thus the entire series of random values in each signature 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 10 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 $\mathcal{P} \rightarrow \mathcal{V}$ 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 $I \leftarrow p\mathcal{H}_{\mathbf{point}}(P_s)$
$\boxed{\mathcal{P} \rightarrow \mathcal{V}}$: $I$
$\boxed{\mathcal{V}}$: $\epsilon, \zeta \leftarrow\!\!\$\ \mathbb{F}^*_{\bar{p}}$
$\boxed{\mathcal{V} \rightarrow \mathcal{P}}$: $\epsilon, \zeta$
$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: 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, 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 10: EFLRS1 signing and verification

In the signature schemes we always imply presence of one more procedure, `Link`, although we do not specify it explicitly. It is constructed trivially, as a comparison of key images I, just as in [15, 8, 20].

5.4 SIZE AND VERIFICATION COMPLEXITY

When the protocol $\texttt{EFLRS1.SignAndVerify}_{1,n}$ in Figure 10 runs, the series of nested subprotocols is executed up to calling `zk2ElemComm`, as shown in the top box in Figure 11. As a result, assuming that verifier postpones all calculations on its side until the end of the message exchange with prover, the verifier has only to check one expanded equality shown in Figure 11.

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

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

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

Figure 11: 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 common ring of n public keys. We consider l signatures in order to compare their summary size and complexity against a threshold variant presented further on. To see the size and verification complexity of the single signature, simply let $l = 1$.

To verify the batch, verifier combines l instances of the equality in Figure 11 together using random weighting. As in [5, 7, 20], the verifier computes all the scalar weights with scalar-scalar multiplications, which are considered consuming negligibly time, and then performs the single multi-exponentiation as per Figure 11, the resulting complexity is shown in Table 3.

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

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

6 LINKABLE THRESHOLD RING SIGNATURE

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

6.1 OVERVIEW

6.1.1 MULTIPLE VECTOR COMMITMENTS

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

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

that, for a given element vector $\mathbf{Y} \in \mathbb{G}^l$, proves that every $Y_i \in \mathbf{Y}$ is a vector commitment over the vector of orthogonal generators $\mathbf{X} \cup \{H\} \in \mathbb{G}^{n*} \times \mathbb{G}^*$ with weights known to prover. It is shown in Figure 12, $\mathtt{zkMVC}_{l,n}$ is a protocol for the relation

$$\mathcal{R} = \{\, \mathbf{X} \in \mathbb{G}^{n*}, H \in \mathbb{G}^*, \mathbf{Y} \in \mathbb{G}^l;\ \mathfrak{a} \in \mathbb{F}_{\mathrm{p}}^{l\times n}, \boldsymbol{\alpha} \in \mathbb{F}_{\mathrm{p}}^l \mid \mathbf{Y} = \mathfrak{a}\cdot\mathbf{X} + \boldsymbol{\alpha}\cdot H \,\}\,. \tag{17}$$

The structure of this protocol is quite simple. All l elements from the vector $\mathbf{Y}$ are combined into one element Y with random weights, then the protocol $\mathtt{zkVC}_n$ 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 one vector commitment proof.

6.1.2 MANY-OUT-OF-MANY PROOF

The $\mathtt{zkMVC}_{l,n}$ protocol, according to the relation (17), proves the same as l $\mathtt{zkVC}_n$ protocols prove. Using it, now we will construct an efficient many-out-of-many proof of membership

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

shown in Figure 13, for the following relation

$$\mathcal{R} = \left\{ \begin{array}{l} \mathbf{P}, \mathbf{Q} \in \mathbb{G}^{n*}, H \in \mathbb{G}^*, \mathbf{Z} \in \mathbb{G}^l; \\ \mathbf{s} \in [0 \dots n-1]^l, \mathbf{p}, \boldsymbol{\alpha} \in \mathbb{F}_{\mathrm{p}}^l \end{array} \;\middle|\; \begin{array}{l} \forall k \in [0 \dots l-1] : \\ Z_k = p_k P_{s_k} + \alpha_k H \end{array} \right\}, \tag{18}$$

where $\mathbf{P}, \mathbf{Q}, H$ satisfy $\text{ort}(\,\mathbf{P} \cup \mathbf{Q} \cup \{H\})$. The many-out-of-many proof of membership $\texttt{zkLin2mChoice}_{n,l}$ in Figure 13 proves the same as l concurrent insances of the one-out-of-many proof of membership $\texttt{zkLin2Choice}_n$ in Figure 7 prove, at the price of one instance.

All the l instances of $\texttt{zkLin2Choice}_n$ are played as a sequence of nested sub-protocol calls invoked simultaneously, we depict this as the following 'invokation stack'

$$l \times \texttt{zkLin2Choice}_n \hookrightarrow l \times \texttt{zkSVC}_{2,n} \hookrightarrow \; l \times \texttt{zkVC}_n \,. \tag{19}$$

Since each of these l concurrent $\texttt{zkLin2Choice}_n$ instances is completely independent of each other, we let all the 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 (19) are needed only to prove 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

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

Therefore, we can replace these $l \times \texttt{zkVC}_n$ calls with one call to $\texttt{zkMVC}_{l,n}$, thus making the invocation stack (19) 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 scheme in Figure 10 that we constructed in Section 5.2 is, in sum, about that prover builds a key image I of type (15), then publishes it, then verifier sends a challenge ζ. Then using the one-out-of-many proof of membership $\texttt{zkLin2Choice}_n$ the prover convinces the verifier that Z, which is built by the formula (16), belongs to the decoy set built by the formula (14), namely, to the set of pairs

$$(\,\mathbf{P} + \zeta\mathbf{U},\ \mathbf{Q}\,)\,,\ \text{where}\ \mathbf{U} = \{\mathcal{H}_{\textbf{point}}(P_i)\}_{i=0}^{n-1}.$$

Suppose, prover publishes a vector of l key images

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

of type (15) each, corresponding to l different indices $\mathbf{s} = \{s_k\}_{k=0}^{l-1}$, which we call actual signing indices or 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 (16), 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 the decoy set by the formula (14). After that, as the last step, they play the $\texttt{zkLin2Choice}_n$ 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

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

Although, instead of playing the one-out-of-many proof protocol l times, they can play as well the many-out-of-many proof protocol $\texttt{zkLin2mChoice}_{n,l}$ once. By doing so, they obtain a threshold version of the signature, which we call EFLRSL (Efficient linkable ring signature for l actual signers), its scheme

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

is shown in Figure 14. Its size is $2\lceil\log_2(n)\rceil + 3l + 3$. The key image vector $\{I_k\}_{k=0}^{l-1}$ is taken into account in the calculation. 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 there holds $\text{ort}(\mathbf{X} \cup \{H\})$, for a vector of elements $\mathbf{Y} \in \mathbb{G}^l$, the protocol $\texttt{zkMVC}_{l,n}$ in Figure 12 is a complete, HVZK argument having WEE for the relation (17).

Proof: Appendix G.
Overview: Section 6.1.1.

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

Relation $\mathcal{R} = \{\, \mathbf{X} \in \mathbb{G}^{n*}, H \in \mathbb{G}^{*}, \mathbf{Y} \in \mathbb{G}^{l};\ \mathfrak{a} \in \mathbb{F}_{\bar{\mathsf{p}}}^{l\times n}, \boldsymbol{\alpha} \in \mathbb{F}_{\bar{\mathsf{p}}}^{l} \mid \mathbf{Y} = \mathfrak{a} \cdot \mathbf{X} + \boldsymbol{\alpha} \cdot H \,\}$ // (17)

// $\mathbf{X}, H$ 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*

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

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

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

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

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: compute $Y = \langle \boldsymbol{\xi}, \mathbf{Y} \rangle$

and run $\texttt{zkVC}_{n}(\mathbf{X}, H, Y;\ \mathbf{a}, \alpha)$

Figure 12: 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 there holds $\mathrm{ort}(\mathbf{P} \cup \mathbf{Q} \cup \{H\})$, for a vector of elements $\mathbf{Z} \in \mathbb{G}^{l}$, the protocol $\texttt{zkLin2mChoice}_{n,l}$ in Figure 13 is a complete, HVZK argument having WEE for the relation (18).

Proof: Appendix H.
Overview: Section 6.1.2.

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

$$\text{Relation } \mathcal{R} = \left\{ \begin{array}{l|l} \mathbf{P}, \mathbf{Q} \in \mathbb{G}^{n*}, H \in \mathbb{G}^{*}, \mathbf{Z} \in \mathbb{G}^{l}; & \forall k \in [0 \ldots l-1]: \\ \mathbf{s} \in [0 \ldots n-1]^{l}, \mathbf{p}, \alpha \in \mathbb{F}_{\bar{p}}^{l} & Z_k = p_k P_{s_k} + \alpha_k H \end{array} \right\} \quad // \ (18)$$

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

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

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

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

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

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

Figure 13: Zero-knowledge argument for multiple element choice relation

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 14 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.
Overview: Section 6.1.3.

```
$\boxed{\texttt{EFLRSL.SignAndVerify}_{l,n}(\mathsf{M}, \mathbf{P}; \mathbf{s}, \mathbf{x})}$
$\mathcal{P}$'s input  : $(\mathsf{M} \in \{0,1\}^\star, \mathbf{P} \in \mathbb{G}^{n*}; \mathbf{s} \in [0 \dots n-1]^l, \mathbf{x} \in \mathbb{F}_{\bar{\mathsf{p}}}^{l*})$
$\mathcal{V}$'s input  : $(\mathsf{M} \in \{0,1\}^\star, \mathbf{P} \in \mathbb{G}^{n*})$
$\mathcal{P}$'s output : $Signature$   // 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 $k \in [0 \dots l-1]$
                      assert $\mathbf{x}_{[k]} \neq 0$
                      $\mathbf{p}_{[k]} = \mathbf{x}_{[k]}^{-1}$
                  lets $(s_k, p_k) \leftarrow (\mathbf{s}_{[k]}, \mathbf{p}_{[k]})$,
                      $\mathbf{I}_{[k]} = p_k \, \mathbf{U}_{[s_k]}$   // 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
    $\epsilon, \zeta \leftarrow\$ \mathbb{F}_{\bar{\mathsf{p}}}^{*}$
$\boxed{\mathcal{V} \to \mathcal{P}}$: $\epsilon, \zeta$
$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: 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 14: 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 15. 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 15: Unfolded equality for EFLRSL, verifier checks it

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

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

[*] Optimized size is shown in Table 7.

Comparing Table 4 and Table 3, we find that the treshold variant of the signature is asymptotically l times more compact. Also, the verification of the treshold variant is asymptotically slightly faster.

7 LIN2-2CHOICE LEMMA

The Lin2-Choice lemma protocol in Figure 7 made it possible to us to choose one element Z from the set $\mathbf{P}$. Now we are going to extend this protocol so that we can select from $\mathbf{P}$ two elements at a time instead of one. That is, 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 remain anonymous, however we want its weight to be 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 16, and in the Simplified Lin2-2Choice lemma (Theorem 10) will prove its properties as a 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 18.

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 a pair of scalars p, v and there holds $Z = Z_P + Z_V \wedge p^{-1}Z_P \in \mathbf{P} \wedge v^{-1}Z_V \in \mathbf{V}$. Naturally, this argument turns into a regular 1-out-of-many membership proof in the case $\mathbf{V} = \varnothing, Z_V = 0$. All elements in $\mathbf{P} \cup \mathbf{V}$ are assumed to be pre-validated nonzero, so the inversion of p and v is always assumed to be possible.

The protocol $\mathtt{zkLin22sChoice}_{n,m}(\mathbf{P}, \mathbf{Q}, \mathbf{V}, \mathbf{W}, H, Z, t; s, p, v, \alpha)$ in Figure 16 is such an argument. Formally, it convinces verifier that prover knows witness (s, p, v, α) for the relation

$$\left\{ \begin{array}{l} \mathbf{P}, \mathbf{Q} \in \mathbb{G}^{n*}, \mathbf{V}, \mathbf{W} \in \mathbb{G}^{m*}, H \in \mathbb{G}^*, Z \in \mathbb{G}, t \in [0 \dots m-1]\,; \\ s \in [0 \dots n-1], p, v, \alpha \in \mathbb{F}_{\bar{p}} \end{array} \;\middle|\; Z = pP_s + vV_t + \alpha H \right\}. \tag{20}$$

As usual, we account for blinding and for zero factors. Also, for $V_t \in \mathbf{V}$ we hide only its factor v, not its index t. The vectors $\mathbf{P}, \mathbf{Q} \in \mathbb{G}^{n*}$, $\mathbf{V}, \mathbf{W} \in \mathbb{G}^{m*}$ in (20) 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, 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.

The protocol $\mathtt{zkLin22sChoice}_{n,m}$ is constructed from $\mathtt{zkLin2Choice}_n$ as follows.

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

$$F \text{ and } E\,. \tag{21}$$

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

$$\{(P_i,\ c_iQ_i,\ 0)\}_{i=0}^{n-1}\,, \tag{22}$$

whereas the second part, which is new, of size m, contains the following ones

$$\{(V_i,\ 0,\ c_{n+i}W_i)\}_{i=0}^{m-1}\,. \tag{23}$$

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

$$rF\,,\ c_{n+t}E\,. \tag{24}$$

- 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

$$\begin{cases} Z = \mathrm{lin}(\mathbf{P}, \mathbf{V}) \\ F = \mathrm{lin}(\mathbf{Q}) \\ E = \mathrm{lin}(\mathbf{W}) \end{cases}. \tag{25}$$

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

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

$$(Z,\ rF,\ c_{n+t}E) \tag{26}$$

we prove with $\texttt{zkSVC}_{3,(n+m)}$ that the first, second, and third elements of the triplet (26) 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 (22) and (23). We observe that thereby all the steps of the $\texttt{zkLin2Choice}_n$ and $\texttt{zkLin2Choice}_m$ protocols are actually performed for Z's 'projections' on $\mathbf{P}$ and on $\mathbf{V}$, respectively. That is, we observe that

$$Z = Z_P + Z_V,\ \text{where } Z_P = \text{lin}(\mathbf{P}),\ Z_V = \text{lin}(\mathbf{V})\ . \tag{27}$$

Thus, we find out that all the steps of the Lin2-Choice lemma protocol have been performed for

- Z_P and the first part of the decoy set comprising n triples (22). The actual index s remains hidden because the response r is randomized, as in the Lin2-Choice lemma protocol.
- Z_V and the second part of the decoy set comprising m triples (23). 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

$$\begin{cases} Z_P \sim P_s & \text{, where } s \text{ is secret} \\ Z_V \sim V_t & \text{, where } t \text{ is public} \end{cases}, \tag{28}$$

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

7.1.2 MULTIPLE SIMMETRIC VECTOR COMMITMENTS

We need one more auxiliary zero-knowledge protocol,

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

It is shown in Figure 17 and proves the same thing as l simultaneously played instances of the $\texttt{zkSVC}_{3,n}$ protocol (Figure 5) prove. Namely, this is a protocol for the following relation

$$\mathcal{R} = \left\{ \begin{array}{l} \mathbf{P} \in \mathbb{G}^{n*}, \mathbf{Q}, \mathbf{R} \in \mathbb{G}^{n}, H \in \mathbb{G}^{*}, \mathbf{Z}, \mathbf{F}, \mathbf{E} \in \mathbb{G}^{l}; \\ \mathfrak{a} \in \mathbb{F}_{\bar{p}}^{l\times n}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma} \in \mathbb{F}_{\bar{p}}^{l} \end{array} \middle| \begin{array}{l} \mathbf{Z} = \mathfrak{a}\cdot\mathbf{P} + \boldsymbol{\alpha}\cdot H \ \wedge \\ \mathbf{F} = \mathfrak{a}\cdot\mathbf{Q} + \boldsymbol{\beta}\cdot H \ \wedge \\ \mathbf{E} = \mathfrak{a}\cdot\mathbf{R} + \boldsymbol{\gamma}\cdot H \end{array} \right\}, \tag{29}$$

where all generators $\mathbf{P},\mathbf{Q},\mathbf{R},H$ are orthogonal to each other. This relation is l instances of the relation (7) merged together. The other accompanying requirements for it are the same as for (7).

We implement the $\texttt{zkMSVC}_{l,3,n}$ protocol the same way as l instances of $\texttt{zkSVC}_{3,n}$ would be implemented using 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 $\texttt{zkVC}_n(\mathbf{X}, H, Y_j;\ \mathfrak{a}_{[j,:]}, \alpha_j + \delta_1\beta_j + \delta_2\gamma_j)$ for each $j \in [0\ldots l-1]$, we invoke the $\texttt{zkMVC}_{l,n}$ protocol (Figure 12) for $\mathbf{X},\mathbf{Y}$. Thus, we get a proof for the relation (29) at the price (i.e., size) of one protocol $\texttt{zkMVC}_{l,n}$ call, and hence, at the price of one $\texttt{zkVC}_n$ call.

7.1.3 LIN2-2CHOICE LEMMA

We can now construct the protocol

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

shown in Figure 18, and prove the Lin2-2Choice lemma which states that $\texttt{zkLin22Choice}_{l,n,m}$ is a complete, zero-knowledge argument having computational witness-extended emulation for the relation

$$\mathcal{R} = \left\{ \begin{array}{l} \mathbf{P},\mathbf{Q} \in \mathbb{G}^{n*}, \mathbf{V},\mathbf{W} \in \mathbb{G}^{m*}, H \in \mathbb{G}^{*}, \mathbf{Z} \in \mathbb{G}^{l}; \\ \mathbf{s} \in [0\ldots n-1]^{l}, \mathbf{p},\mathbf{v},\boldsymbol{\alpha} \in \mathbb{F}_{\bar{p}}^{l} \end{array} \middle| \begin{array}{l} \forall k \in [0\ldots l-1]: \\ Z_k = p_k P_{s_k} + v_k V_k + \alpha_k H \end{array} \right\}, \tag{30}$$

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

The relation (30) is essentially the relation (20) repeated for the first l elements of the decoy set's second part (23). Having such a correspondence between the relations (30) and (20), the $\mathtt{zkLin22Choice}_{l,n,m}$ protocol is l instances of the protocol $\mathtt{zkLin22sChoice}_{n,m}$ run in parallel, with the only one refinement.

The refinement is that all the l instances of the $\mathtt{zkLin22sChoice}_{n,m}$ 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 $\mathtt{zkSVC}_{3,n}$ are called. All these l calls of $\mathtt{zkSVC}_{3,n}$, are, in turn, replaced by one call to $\mathtt{zkMSVC}_{l,3,n}$, 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 there holds $\mathrm{ort}(\mathbf{P} \cup \mathbf{Q} \cup \mathbf{V} \cup \mathbf{W} \cup \{H\})$, for an element $Z \in \mathbb{G}$, the protocol $\mathtt{zkLin22sChoice}_{n,m}$ in Figure 16 is a complete, HVZK argument having WEE for the relation (20).

Proof: Appendix K.
Overview: Section 7.1.1.

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

$$\text{Relation } \mathcal{R} = \left\{ \begin{array}{l} \mathbf{P}, \mathbf{Q} \in \mathbb{G}^{n*}, \mathbf{V}, \mathbf{W} \in \mathbb{G}^{m*}, H \in \mathbb{G}^*, Z \in \mathbb{G}, t \in [0 \ldots m-1]; \\ s \in [0 \ldots n-1], p, v, \alpha \in \mathbb{F}_{\bar{p}} \end{array} \;\middle|\; Z = pP_s + vV_t + \alpha H \right\} \quad \text{// (20)}$$

// $\mathbf{P}, \mathbf{Q}, \mathbf{V}, \mathbf{W}, H$ 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**

$$F = qQ_s + \beta H$$
$$E = vW_t + \gamma H$$

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

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

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

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

$$\hat{\beta} = r\beta$$
$$\hat{\gamma} = c_{n+t}\gamma \ ,$$

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

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

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

let $\hat{F} \leftarrow rF$
$\hat{E} \leftarrow \mathbf{c}_{[n+t]}E$,
and run $\mathtt{zkSVC}_{3,(n+m)}(\hat{\mathbf{P}}, \hat{\mathbf{Q}}, \hat{\mathbf{R}}, H, Z, \hat{F}, \hat{E}; \mathbf{a}, \alpha, \hat{\beta}, \hat{\gamma})$

Figure 16: Simplified Lin2-2Choice lemma protocol, zero-knowledge argument for two-element choice relation

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 17 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 there holds $\mathrm{ort}(\mathbf{P} \cup \mathrm{nz}(\mathbf{Q}) \cup \mathrm{nz}(\mathbf{R}) \cup \{H\})$, for three vectors of elements $\mathbf{Z}, \mathbf{F}, \mathbf{E} \in \mathbb{G}^l$, the protocol $\texttt{zkMSVC}_{l,3,n}$ in Figure 17 is a complete, HVZK argument having WEE for the relation (29).

Proof: Appendix L.
Overview: Section 7.1.2.

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

$$\text{Relation } \mathcal{R} = \left\{ \begin{array}{l|l} \mathbf{P} \in \mathbb{G}^{n*}, \mathbf{Q}, \mathbf{R} \in \mathbb{G}^n, H \in \mathbb{G}^*, \mathbf{Z}, \mathbf{F}, \mathbf{E} \in \mathbb{G}^l; & \mathbf{Z} = \mathfrak{a} \cdot \mathbf{P} + \boldsymbol{\alpha} \cdot H \ \wedge \\ \mathfrak{a} \in \mathbb{F}_{\bar{p}}^{l \times n}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma} \in \mathbb{F}_{\bar{p}}^l & \mathbf{F} = \mathfrak{a} \cdot \mathbf{Q} + \boldsymbol{\beta} \cdot H \ \wedge \\ & \mathbf{E} = \mathfrak{a} \cdot \mathbf{R} + \boldsymbol{\gamma} \cdot H \end{array} \right\} \quad // \ (29)$$

// $\mathbf{P}, \mathbf{Q}, \mathbf{R}, H$ 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*

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

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

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

$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: 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 17: 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 there holds $\mathrm{ort}(\mathbf{P} \cup \mathbf{Q} \cup \mathbf{V} \cup \mathbf{W} \cup \{H\})$, for a vector of elements $\mathbf{Z} \in \mathbb{G}^l$, the protocol $\texttt{zkLin22Choice}_{l,n,m}$ in Figure 18 is a complete, HVZK argument having WEE for the relation (30).

Proof: Appendix M.
Overview: Section 7.1.3.

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

$$\text{Relation } \mathcal{R} = \left\{ \begin{array}{l|l} \mathbf{P},\mathbf{Q} \in \mathbb{G}^{n*}, \mathbf{V},\mathbf{W} \in \mathbb{G}^{m*}, H \in \mathbb{G}^{*}, \mathbf{Z} \in \mathbb{G}^{l}; & \forall k \in [0\dots l-1]: \\ \mathbf{s} \in [0\dots n-1]^{l}, \mathbf{p},\mathbf{v},\boldsymbol{\alpha} \in \mathbb{F}_{\bar{p}}^{l} & Z_k = p_k P_{s_k} + v_k V_k + \alpha_k H \end{array} \right\} \quad \text{// (30)}$$

// $\mathbf{P},\mathbf{Q},\mathbf{V},\mathbf{W},H$ 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*

```
[P]: $\mathbf{q},\boldsymbol{\beta},\boldsymbol{\gamma} \leftarrow\!\$\ \mathbb{F}_{\bar{p}}^{l*}$ , allocates $\mathbf{F},\mathbf{E} \in \mathbb{G}^{l*}$ ,
    initializes    foreach $k \in [0\dots l-1]$
                       let $(s_k, p_k, v_k, \beta_k, \gamma_k) \leftarrow (\mathbf{s}_{[k]}, \mathbf{p}_{[k]}, \mathbf{v}_{[k]}, \boldsymbol{\beta}_{[k]}, \boldsymbol{\gamma}_{[k]})$ ,
                           if $p_k = 0$ then $\mathbf{q}_{[k]} = 0$ endif ,
                       let $q_k \leftarrow \mathbf{q}_{[k]}$ ,
                           $\mathbf{F}_{[k]} = q_k Q_{s_k} + \beta_k H$   // $\mathbf{F}$ is filled in, note random $q_k$'s are nullified when $p_k = 0$
                           $\mathbf{E}_{[k]} = v_k W_k + \gamma_k H$   // $\mathbf{E}$ is filled in
                   endforeach
[P → V]: $\mathbf{F},\mathbf{E}$
[V]: $\mathbf{c} \leftarrow\!\$\ \mathbb{F}_{\bar{p}}^{(n+m)*}$
[V → P]: $\mathbf{c}$
[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 $k \in [0\dots l-1]$
                           let $c_{s_k} \leftarrow \mathbf{c}_{[s_k]}$
                               if $p_k \neq 0$ then $\mathbf{r}_{[k]} = c_{s_k} p_k / q_k$ endif   // $\mathbf{r}$ is filled in here
                       endforeach ,
    continues initialization   $\hat{\boldsymbol{\beta}} = \mathbf{r} \circ \boldsymbol{\beta}$
                               $\hat{\boldsymbol{\gamma}} = \mathbf{c}_{[n:(n+l)]} \circ \boldsymbol{\gamma}$ ,
    lets $\mathfrak{a} = \{a_{k\in[0\dots l-1], i\in[0\dots n+m-1]}\} = \begin{cases} a_{k,s_k} = p_k & \text{// that is, } p_k \text{ is at } s_k\text{'th position in } k\text{'th row} \\ a_{k,n+k} = v_k & \text{// that is, } v_k \text{ is at } (n+k)\text{'th position in } k\text{'th row} \\ a_{k,i} = 0 \text{ if } i \neq s_k \wedge i \neq (n+k) & \text{// zeros for all the rest} \end{cases}$
[P → V]: $\mathbf{r}$
[P and 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 $\hat{\mathbf{F}} \leftarrow \mathbf{r} \circ \mathbf{F}$
        $\hat{\mathbf{E}} \leftarrow \mathbf{c}_{[n:(n+l)]} \circ \mathbf{E}$ ,
    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 18: 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. We assume that each public key in the signature ring has an associated hidden amount in the form of Pedersen commitment [18]. When 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 another amount, which is hidden behind another commitment known beforehand to both of the prover and verifier. We are going to make the prover providing a zero-knowledge proof of this balance along with the signature.

8.1 ADDITIONAL DEFINITIONS

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

$$\{(P_i, A_i)\}_{i=0}^{n-1}\text{ , where }\mathbf{P} = \{P_i\}_{i=0}^{n-1} \ \wedge \ \mathbf{A} = \{A_i\}_{i=0}^{n-1}. \tag{31}$$

In the honest case we assume 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

$$P_i = p_i G\,, \tag{32}$$

$$A_i = b_i B + d_i D\,. \tag{33}$$

In general, as usual, we assume the case is dishonest, i.e., the equalities (33) and (32) may not hold and, moreover, some or all P_i's and A_i's in the ring may be adversarially chosen. However, hereinafter we will assume that for all A_i's in the ring there are some valid proofs of (33) supplied along with the signature. With this precondition, assuming verifier checks the supplied proofs for A_i's when it verifies the signature, in the worst case the involved P_i's can have adversarially chosen p_i's or can have unknown relation to G, whereas A_i's can only have adversarially chosen b_i's and d_i's.

In Figure 19 we summarize the above definition of how the hidden amounts are represented in the system.

`Hidden amounts`

- Each public key P is accompanied by a hidden amount A in the system. Each ring has the form (31).
- Each hidden amount A in a ring is assumed having the decomposition (33) 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 (33) for it in the system.

Figure 19: Hidden amounts seen to all parties

We also need to supplement the common information, which is available to all parties according to Figure 1 and Figure 8, with an extended set of predefined orthogonal generators, and to update the function $\mathcal{H}_{\mathbf{point}}$ one more time, as in Figure 20, so that it will respect orthogonality of the additional generators.

`Updated common information`

- A couple of generators $B, D \in \mathbb{G}^*$ and the enlarged vector $\mathbf{G} = \{G_0, G_1, G_2, \dots, G_{2\bar{\mathsf{n}}-1}\} \in \mathbb{G}^{2\bar{\mathsf{n}}*}$ such that, for any set $\mathbf{H}$ of $\mathcal{H}_{\mathbf{point}}$ images on different pre-images, there 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 20: Updated common information available to each party

8.2 OVERVIEW

8.2.1 SIGNATURE EFLRSLWB

Efficient linkable threshold ring signature EFLRSLWB (Efficient linkable ring signature for l actual signers with balance proof) is shown in Figure 21. Here is an informal introduction to how it works.

Having a ring of the form (31) prover publishes l key images, for each of the actually signing indices $\mathbf{s} \in [0 \dots n-1]^l$,

$$\mathbf{I} = \{I_k\}_{k=0}^{l-1} = \{x_k^{-1} \mathcal{H}_{\mathbf{point}}(P_{s_k})\}_{k=0}^{l-1}. \tag{34}$$

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

$$A^{\mathbf{sum}} = \sum\nolimits_{k=0}^{l-1} A_{s_k}\,. \tag{35}$$

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

$$\mathbf{A}^{\mathbf{tmp}} = \{A_{s_k} + \mu_k H\}_{k=0}^{l-1}\,, \ \text{ where } \mu_k \leftarrow\!\$\ \mathbb{F}^*_{\bar{\mathsf{p}}}\,. \tag{36}$$

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

$$\mathbf{J} = \{x_k^{-1}\mathcal{H}_{\mathbf{point}}(H, A_k^{\mathbf{tmp}}) + \upsilon_k H\}_{k=0}^{l-1}, \ \text{ where } \upsilon_k \leftarrow\!\$\ \mathbb{F}^*_{\bar{\mathsf{p}}}\,. \tag{37}$$

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 they are orthogonal to H even in the dishonest case.

In addition to this, prover and verifier generate one more orthogonal generator, K, as a hash to group of everything they have in common after $\mathbf{J}$ is published.

Now, using random weights ζ, ω, χ prover and verifier define the following three vectors

$$\mathbf{X} = \mathbf{P}\ - \{K\}^n\ + \zeta\,\{\mathcal{H}_{\mathbf{point}}(P_i)\}_{i=0}^{n-1}\ - \omega\mathbf{A}\,, \tag{38}$$

$$\mathbf{V} = \{K\}^l\ + \omega\mathbf{A}^{\mathbf{tmp}}\ + \chi\,\{\mathcal{H}_{\mathbf{point}}(H, A_k^{\mathbf{tmp}})\}_{k=0}^{l-1}\,, \tag{39}$$

$$\mathbf{Z} = \{G\}^l + \zeta\mathbf{I} + \chi\mathbf{J}\,, \tag{40}$$

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

$$\mathtt{zkLin22Choice}_{l,n,l}(\mathbf{X}, \mathbf{Q}, \mathbf{V}, \mathbf{W}, H, \mathbf{Z};\ \ \mathbf{s}, \mathbf{x}^{-1}, \mathbf{x}^{-1},\ \boldsymbol{\alpha}_H\ )\,, \tag{41}$$

where $\mathbf{Q}, \mathbf{W}$ are auxiliary orthogonal generators prepared in advance. Here all elements in $\mathbf{Q}, \mathbf{W}$ are also orthogonal to the elements in $\mathbf{X}$ (38) and in $\mathbf{V}$ (39), this is because of $\mathcal{H}_{\mathbf{point}}$ 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 (41) successfully completes, by Theorem 12 (Lin2-2Choice lemma) verifier is convinced that, for each $k \in [0 \dots l-1]$, prover knows a scalar pair (p_k, v_k) such that there holds, to the accuracy of H component

$$Z_k =\ p_k X_{s_k} + v_k V_k\ . \tag{42}$$

Inserting (38), (39), (40) into (42) the verifier obtains

$$G + \zeta I_k + \chi J_k = p_k(\,P_{s_k} - K + \zeta\mathcal{H}_{\mathbf{point}}(P_{s_k}) - \omega A_{s_k}\,) + v_k(\,K + \omega A_k^{\mathbf{tmp}} + \chi\mathcal{H}_{\mathbf{point}}(H, A_k^{\mathbf{tmp}})\,)\ , \tag{43}$$

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. Reducing (43), the verifier gets

$$G + \zeta I_k + \chi J_k = p_k(\,P_{s_k} + \zeta\mathcal{H}_{\mathbf{point}}(P_{s_k}) + \chi\mathcal{H}_{\mathbf{point}}(H, A_k^{\mathbf{tmp}})\,) + p_k(\,\omega A_k^{\mathbf{tmp}} - \omega A_{s_k}\,)\ . \tag{44}$$

Since $\mathcal{H}_{\mathbf{point}}(H, A_k^{\mathbf{tmp}})$ is orthogonal to everything else in the right-hand side of (44) and since at least P_{s_k} in it is nonzero, by Theorem 3 verifier gets convinced that the following hold for some known to the prover scalar p_k, to the accuracy of blinding with H,

$$\begin{cases} G \ = p_k P_{s_k} & \text{(45a)} \\ I_k \ = p_k\mathcal{H}_{\mathbf{point}}(P_{s_k}) & \text{(45b)} \\ J_k \ = p_k\mathcal{H}_{\mathbf{point}}(H, A_k^{\mathbf{tmp}}) & \text{(45c)} \\ A_{s_k} = A_k^{\mathbf{tmp}}\ \ . & \text{(45d)} \end{cases}$$

The equalities (45a), (45b) are strict, as all elements in them are included into the pre-image of H. Thus, they convince verifier that the signing is correct and the linking tag is valid. At the same time, (45d) convinces verifier that $A_k^{\mathbf{tmp}}$ is the hidden amount corresponding to the signing key, to the accuracy of H.

Keeping in mind there exist l equalities (45d) for all actually signing keys in $\mathbf{s}$, after the call to

$$\mathtt{zk2ElemComm}(D, H, A^{\mathbf{sum}} - \sum_{k=0}^{l-1} A_k^{\mathbf{tmp}}\ ;\ \dots\,), \tag{46}$$

the verifier is convinced that $A^{\mathbf{sum}}$ is a 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 $A^{\mathbf{sum}}$ and $\mathbf{A} \supseteq \{A_{s_k}\}_{k=0}^{l-1}$ are in the pre-image of H, the call (46) convinces the verifier in the stronger assertion, namely, that $A^{\mathbf{sum}}$ is a sum of $\{A_{s_k}\}_{k=0}^{l-1}$ to the accuracy of only D component.

Thus, the verifier is convinced that the signature is correct and also that there holds, to the accuracy of D, the equality (35). This is all it gets from the signature.

8.2.2 IMMEDIATE IMPLICATION

The verifier then reasons from the properties of the system in Figure 19, as follows. Recalling that according to Figure 19, for each element in $\mathbf{A}$, there exists a proof of the decomposition (33) in the system, having checked these proofs the verifier makes sure that $\mathbf{A}$ contains some hidden amounts, and not something else, namely, that there holds

$$\{A_{s_k}\}_{k=0}^{l-1} = \{b_{s_k}B + d_{s_k}D\}_{k=0}^{l-1} \subseteq \mathbf{A}\,, \text{ where all } b_{s_k}\text{'s and } d_{s_k}\text{'s are known to someones in the system.} \tag{47}$$

From the decompositions (47) and from the proved equality (35), it gets convinced that

$$A^{\mathbf{sum}} = b^{\mathbf{sum}}B + d^{\mathbf{sum}}D\,, \text{ where } b^{\mathbf{sum}} \text{ and } d^{\mathbf{sum}} \text{ can be reconstructed in the system.} \tag{48}$$

Finally, from (48), (47), (35) it gets convinced that

$$b^{\mathbf{sum}} = \sum_{k=0}^{l-1} b_{s_k}\,, \tag{49}$$

Thus, by verifying the EFLRSLWB signature and the corresponding proofs of the form (33) for all hidden amounts in the signature ring, the verifier gets convinced that prover knows signing private keys, and that the sum of the corresponding hidden amounts is balanced with the given hidden amount $A^{\mathbf{sum}}$.

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 21 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 8.2.1.

Note, Theorem 13 doesn't impose any requirement on elements of the vector $\mathbf{A}$ and on $A^{\mathbf{sum}}$, i.e., there is no assumption like (33) about their decompositions. At the same time, it's easy to see that if the property 9) holds, then the proof of balance (49) immediately follows from the proofs of the decomposition (33) for all $A_k \in \mathbf{A}$. Therefore, if these proofs are obtained by any other means together with EFLRSLWB, and all of them are successfully verified, then the proof of balance (49) is thus obtained.

```
$\boxed{\texttt{EFLRSLWB.SignAndVerify}_{l,n}(\mathsf{M}, \mathbf{P}, \mathbf{A}, A^{\mathbf{sum}}, D; \mathbf{s}, \mathbf{x}, d^{\Delta\mathbf{sum}})}$
$\mathcal{P}$'s input  : $(\mathsf{M} \in \{0,1\}^{\star}, \mathbf{P} \in \mathbb{G}^{n*}, \mathbf{A} \in \mathbb{G}^{n}, A^{\mathbf{sum}} \in \mathbb{G}, D \in \mathbb{G}^{*}; \mathbf{s} \in [0 \ldots n-1]^{l}, \mathbf{x} \in \mathbb{F}_{\bar{p}}^{l*}, d^{\Delta\mathbf{sum}} \in \mathbb{F}_{\bar{p}})$
$\mathcal{V}$'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 $\mathbf{I} \in \mathbb{G}^{l*}$, $\mathbf{p} \in \mathbb{F}_{\bar{p}}^{l*}$,
    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
    $\epsilon \leftarrow\$ \mathbb{F}_{\bar{p}}^{*}$
$\boxed{\mathcal{V} \to \mathcal{P}}$: $\epsilon$
$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: let $H \leftarrow \mathcal{H}_{\mathbf{point}}(\epsilon)$    // thus, $H$ is orthogonal to all known so far elements, i.e., $\mathrm{ort}(H, G, \mathbf{P}, \mathbf{A}, \mathbf{U}, \mathbf{I}, A^{\mathbf{sum}}, D)$
$\boxed{\mathcal{P}}$: $\boldsymbol{\mu}, \boldsymbol{\upsilon} \leftarrow\$ \mathbb{F}_{\bar{p}}^{l*}$, allocates $\mathbf{A}^{\mathbf{tmp}} \in \mathbb{G}^{l*}$, $\boldsymbol{\alpha} \in \mathbb{F}_{\bar{p}}^{l*}$,
    initializes      foreach $k \in [0 \ldots l-1]$
                       lets $\mu_k \leftarrow \boldsymbol{\mu}_{[k]}$,
                         $\mathbf{A}_{[k]}^{\mathbf{tmp}} = \mathbf{A}_{[s_k]} + \mu_k H$    // $\mathbf{A}^{\mathbf{tmp}}$ is filled, 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 $\hat{\mathbf{U}} \leftarrow \{\mathcal{H}_{\mathbf{point}}(H, \mathbf{A}_{[k]}^{\mathbf{tmp}})\}_{k=0}^{l-1}$
$\boxed{\mathcal{P}}$: 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 $J$
    $\hat{\epsilon}, \zeta, \omega, \chi \leftarrow\$ \mathbb{F}_{\bar{p}}^{*}$
$\boxed{\mathcal{V} \to \mathcal{P}}$: $\hat{\epsilon}, \zeta, \omega, \chi$
$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: 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 $S = A^{\mathbf{sum}} - \sum_{k=0}^{l-1} \mathbf{A}_{[k]}^{\mathbf{tmp}}$
            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 21: 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 22. By combining the equalities (*) and (**) with random weighting and 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

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

and

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

Figure 22: EFLRSLWB unfolded equality, verifier checks it

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

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

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 [15], CLSAG [8], CryptoNote [22] schemes and, for instance, multiparty signing operations can be easily implemented with it 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 a few modifications to it, which are outlined below. Although, for the first, we will have to generalize Theorem 3, which is about 3-element tuples, to element tuples of greater length to prove that this movement of x is correct.

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, such that

$$\mathbf{T} = (P,\ Q_0,\ Q_1, \dots, Q_{t-1},\ S_0,\ S_1, \dots, S_{s-1})\ , \tag{50}$$

$$\mathbf{D} = (Z,\ F_0,\ F_1,\ \dots, F_{t-1},\ 0,\ \ 0,\ \dots)\ , \tag{51}$$

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 inner products of our tuples with this scalar vector $\boldsymbol{\xi}$. Namely, we build X, Y such that

$$X = \langle \boldsymbol{\xi}, \mathbf{T} \rangle = P + \xi_1 Q_0 + \xi_2 Q_1 + \dots + \xi_{t+1} S_0 + \xi_{t+2} S_1 + \dots\ , \tag{52}$$

$$Y = \langle \boldsymbol{\xi}, \mathbf{D} \rangle = Z + \xi_1 F_0 + \xi_2 F_1 + \dots\ , \tag{53}$$

$$\text{where } \boldsymbol{\xi} = [1, \delta_0, \delta_1, \dots, \delta_{t-1}, \sigma_0, \sigma_1, \dots, \sigma_{s-1}]\ . \tag{54}$$

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, HVZK, and having WEE argument that convinces us that $Y \sim X$ to the accuracy of H component. Here H is assumed as a blinding generator 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 $\mathbf{T}$, there necessarily exists a factor a known to prover that connects all the corresponding element pairs in $\mathbf{T}$ and $\mathbf{D}$. Namely, the following relation (55), protocol $\texttt{zkTElemRW}_{t,s}(P, \mathbf{Q}, \mathbf{S}, H, Z, \mathbf{F};\ a, \alpha, \boldsymbol{\beta}, \boldsymbol{\gamma})$ in Figure 23, and Theorem 14, formalize the game and sufficient conditions for the existence of such a factor.

$$\left\{ \begin{array}{l|l} P \in \mathbb{G}^*,\ \mathbf{Q} \in \mathbb{G}^t, \mathbf{S} \in \mathbb{G}^s, H \in \mathbb{G}^*, Z \in \mathbb{G},\ \mathbf{F} \in \mathbb{G}^t; & Z\ = aP + \alpha H\ \wedge \\ a, \alpha \in \mathbb{F}_{\bar{\mathsf{p}}},\ \boldsymbol{\beta} \in \mathbb{F}_{\bar{\mathsf{p}}}^t, \boldsymbol{\gamma} \in \mathbb{F}_{\bar{\mathsf{p}}}^s & \mathbf{F}\ = a\mathbf{Q} + \boldsymbol{\beta} H\ \wedge \\ & \{0\}^s\ = a\mathbf{S} + \boldsymbol{\gamma} H \end{array} \right\} \tag{55}$$

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 21 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 21, prover builds a vector of the linear key images $\hat{\mathbf{I}}$ as

$$\hat{\mathbf{I}} = \{\, x_k U_{s_k} \,\}_{k=0}^{l-1}\ . \tag{56}$$

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

$$\mathbf{U}^{\mathbf{tmp}} = \{\, U_{s_k} \,\}_{k=0}^{l-1} + \hat{\boldsymbol{\mu}} H, \quad \text{where } \hat{\boldsymbol{\mu}} \leftarrow\!\!\$\ \mathbb{F}_{\bar{\mathsf{p}}}^{l*}, \tag{57}$$

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

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

$$\mathbf{X} = \mathbf{P} - \{K\}^n + \zeta \mathbf{U} - \omega \mathbf{A}\,, \tag{58}$$

$$\mathbf{V} = \{K\}^l + \omega \mathbf{A}^{\mathbf{tmp}} - \zeta \mathbf{U}^{\mathbf{tmp}} + \theta \hat{\mathbf{I}} + \chi \hat{\mathbf{U}}\,, \tag{59}$$

$$\mathbf{Z} = \{G\}^l + \theta \mathbf{U}^{\mathbf{tmp}} + \chi \mathbf{J}\,. \tag{60}$$

Then both of them proceed with executing the protocol to the completion. Of course, the prover adjusts the total blinding factor at the private input of $\texttt{zkLin22Choice}_{l,n,l}$ with respect to new $\hat{\boldsymbol{\mu}}$ sampled in (57).

Since $\mathbf{X}, \mathbf{V}, \mathbf{Z}$ are now defined by (58), (59), (60) instead of (38), (39), (40), by Theorem 12 (Lin2-2Choice lemma) the verifier obtains l following equalies, instead of l equalities (44), for each $k \in [0 \dots l-1]$, to the accuracy of H component

$$G + \theta U_k^{\mathbf{tmp}} + \chi J_k = p_k (P_{s_k} + \theta \hat{I}_k + \chi \hat{U}_k) + p_k (\omega A_k^{\mathbf{tmp}} - \omega A_{s_k} + \zeta U_{s_k} - \zeta U_k^{\mathbf{tmp}})\,, \quad \text{where } p_k = x_k^{-1}\,. \tag{61}$$

By Theorem 14, from (61) it 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 R

$$\begin{cases} G\ = p_k P_{s_k} & \text{(62a)} \\ U_{s_k}\ = U_k^{\mathbf{tmp}} & \text{(62b)} \\ U_{s_k}\ = p_k \hat{I}_k & \text{(62c)} \\ J_k\ = p_k \hat{U}_k & \text{(62d)} \\ A_{s_k}\ = A_k^{\mathbf{tmp}}\ . & \text{(62e)} \end{cases}$$

From (62a) and (62c), which are strict (have zero H component, as H is a hash image of all their elements), the verifier 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 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 there holds* $\mathrm{nz}(\mathbf{Q}) \neq \varnothing \;\wedge\; P \mathrel{!=} \mathrm{lin}(\mathrm{nz}(\mathbf{Q}) \cup \mathrm{nz}(\mathbf{S}) \cup \{H\}) \;\wedge\; H \mathrel{!=} \mathrm{lin}(\mathrm{nz}(\mathbf{Q}) \cup \{P\})$*, the protocol* $\mathtt{zkTElemRW}_{t,s}$ *in Figure 23 is a complete, HVZK argument having WEE for the relation (55).*

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

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

$$\text{Relation } \mathcal{R} = \left\{ \begin{array}{l} P \in \mathbb{G}^*,\ \mathbf{Q} \in \mathbb{G}^t, \mathbf{S} \in \mathbb{G}^s, H \in \mathbb{G}^*, Z \in \mathbb{G},\ \mathbf{F} \in \mathbb{G}^t; \\ a, \alpha \in \mathbb{F}_{\bar{\mathsf{p}}},\ \boldsymbol{\beta} \in \mathbb{F}_{\bar{\mathsf{p}}}^t, \boldsymbol{\gamma} \in \mathbb{F}_{\bar{\mathsf{p}}}^s \end{array} \middle| \begin{array}{l} Z \;= aP + \alpha H \;\wedge \\ \mathbf{F} \;= a\mathbf{Q} + \boldsymbol{\beta} H \;\wedge \\ \{0\}^s \;= a\mathbf{S} + \boldsymbol{\gamma} H \end{array} \right\} \quad \text{// (55)}$$

// Precondition: $P, \mathbf{Q}, H$ in $\mathcal{R}$ satisfy $\mathrm{nz}(\mathbf{Q}) \neq \varnothing \wedge$

// $P \mathrel{!=} \mathrm{lin}(\mathrm{nz}(\mathbf{Q}) \cup \mathbf{S} \cup \{H\}) \;\wedge\; H \mathrel{!=} \mathrm{lin}(\mathrm{nz}(\mathbf{Q}) \cup \{P\})$

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

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

$\boxed{\mathcal{P}}$: 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, HVZK, and WEE protocol
that convinces $\mathcal{V}$ that $\mathcal{P}$ knows $a, \hat{\alpha}$ such
that there holds $Y = aX + \hat{\alpha} H$

Figure 23: Random weighting for two t-tuples

9.2.2 SIGNATURE MULTRATUG

Theorem 15:
The scheme in Figure 24 obtained from the scheme in Figure 21 by appending the element vector $\mathbf{U}^{\mathbf{tmp}}$ *and substituting the new key image vector* $\hat{\mathbf{I}}$ *for the vector* $\mathbf{I}$ *in it, as shown in Figure 24, is a linkable threshold ring signature retaining the properties 1. . . 9) of the scheme in Figure 21 listed in Theorem 13.*

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

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

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

$\mathcal{P}$'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 : $Signature$    // 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
    $\epsilon \leftarrow\$ \mathbb{F}_{\bar{\mathsf{p}}}^*$
$\boxed{\mathcal{V} \to \mathcal{P}}$: $\epsilon$
$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: let $H \leftarrow \mathcal{H}_{\mathbf{point}}(\epsilon)$    // thus, $H$ is orthogonal to all known so far elements, i.e., $\text{ort}(H, G, \mathbf{P}, \mathbf{A}, \mathbf{U}, \hat{\mathbf{I}}, A^{\mathbf{sum}}, D)$
$\boxed{\mathcal{P}}$: $\boldsymbol{\mu}, \hat{\boldsymbol{\mu}}, \boldsymbol{\upsilon} \leftarrow\$ \mathbb{F}_{\bar{\mathsf{p}}}^{l*}$, 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 $(\mu_k, \hat{\mu}_k) \leftarrow (\boldsymbol{\mu}_{[k]}, \hat{\boldsymbol{\mu}}_{[k]})$,
                            $\mathbf{A}_{[k]}^{\mathbf{tmp}} = \mathbf{A}_{[s_k]} + \mu_k H$    // $\mathbf{A}^{\mathbf{tmp}}$ is filled, amounts get double blinded (with $D$ and with $H$)
                            $\boldsymbol{\alpha}_{[k]} = p_k \, \mu_k$    // $\alpha$ is initialized here, it contains reduced $\mathbf{A}^{\mathbf{tmp}}$'s second blinding factors
                            $\mathbf{U}_{[k]}^{\mathbf{tmp}} = \mathbf{U}_{[s_k]} + \hat{\mu}_k H$    // $\mathbf{U}^{\mathbf{tmp}}$ is filled, $U$'s get blinded with $H$
                            $\hat{\boldsymbol{\alpha}}_{[k]} = p_k \, \hat{\mu}_k$    // $\hat{\alpha}$ 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 $\hat{\mathbf{U}} \leftarrow \{\, \mathcal{H}_{\mathbf{point}}(H, \mathbf{U}^{\mathbf{tmp}}, \mathbf{A}_{[k]}^{\mathbf{tmp}}) \,\}_{k=0}^{l-1}$
$\boxed{\mathcal{P}}$: 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 $J$
    $\hat{\epsilon}, \zeta, \omega, \chi, \theta \leftarrow\$ \mathbb{F}_{\bar{\mathsf{p}}}^*$
$\boxed{\mathcal{V} \to \mathcal{P}}$: $\hat{\epsilon}, \zeta, \omega, \chi, \theta$
$\boxed{\mathcal{P} \text{ and } \mathcal{V}}$: let $K \leftarrow \mathcal{H}_{\mathbf{point}}(\hat{\epsilon})$    // thus, $\text{ort}(K, H, G, \mathbf{P}, \mathbf{A}, \mathbf{U}, \mathbf{I}, A^{\mathbf{sum}}, \mathbf{A}^{\mathbf{tmp}}, \hat{\mathbf{U}}, \mathbf{J})$ holds
            allocate $\mathbf{X} \in \mathbb{G}^{n*}$, $\mathbf{V}, \mathbf{Z} \in \mathbb{G}^{l*}$, $S \in \mathbb{G}$,
            assign $\mathbf{X} = \mathbf{P} - \{K\}^n + \zeta\mathbf{U} - \omega\mathbf{A}$,        $\mathbf{V} = \{K\}^l + \omega\mathbf{A}^{\mathbf{tmp}} - \zeta\mathbf{U}^{\mathbf{tmp}} + \theta\hat{\mathbf{I}} + \chi\hat{\mathbf{U}}$,
                $\mathbf{Z} = \{G\}^l + \theta\mathbf{U}^{\mathbf{tmp}} + \chi\mathbf{J}$
            assign $S = A^{\mathbf{sum}} - \sum_{k=0}^{l-1} \mathbf{A}_{[k]}^{\mathbf{tmp}}$
            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 24: Multratug with $\hat{I} = x\mathcal{H}_{\mathbf{point}}(P)$ signing and verification

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 $\mathbf{I}$ changes neither of the size and 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.

10 BETTER ARGUMENT FOR VECTOR COMMITMENT

As we have already noted, 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, with the same properties of completeness, HVZK, and WEE. 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 HVZK and having WEE 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 would suffice. Moreover, this proof does not have to be HVZK, a complete argument having WEE would be enough.

10.2 FORMAL PRESENTATION

For the first, we take the following vector commitment relation, which is actually the relation (5) with the items renamed and, at the same time, is the relation (3) with the blinding generator H moved to the vector $\mathbf{X}$.

$$\mathcal{R} = \{\, \mathbf{X} \in \mathbb{G}^{n*},\ Y \in \mathbb{G};\ \mathbf{x} \in \mathbb{F}_{\mathrm{p}}^{n} \mid Y = \langle \mathbf{x}, \mathbf{X} \rangle \,\}\,, \tag{63}$$

and define the following custom Schnorr-like protocol for it.

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

Relation $\mathcal{R} = \{\, \mathbf{X} \in \mathbb{G}^{n*},\ Y \in \mathbb{G};\ \mathbf{x} \in \mathbb{F}_{\bar{\mathsf{p}}}^{n} \mid Y = \langle \mathbf{x}, \mathbf{X} \rangle \,\}$ // (63)

// 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*

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

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

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

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

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

Figure 25: Zero-knowledge argument for n element commitment relation

Properties of the protocol $\texttt{zkNElemComm}_n$ in Figure 25 are specified in the next theorem. Note that, for $n = 2$, $\texttt{zkNElemComm}_2$ is equivalent to $\texttt{zk2ElemComm}$ in Figure 2.

Theorem 16:
For $n \in \mathbb{N}^*$, for a vector of nonzero elements $\mathbf{X} \in \mathbb{G}^{n*}$ such that there holds $\mathrm{ort}(\mathbf{X})$*, for an element $Y \in \mathbb{G}$, the protocol* $\texttt{zkNElemComm}_n$ *in Figure 25 is a complete, HVZK argument having WEE for the relation (63).*

Proof: is in Appendix T.

For the second, in Figure 26 we define a log-size vector commitment argument $\texttt{argVC}_n$ for the same relation (63). Note, we do use the blinding generator H neither in $\texttt{zkNElemComm}_n$ nor in $\texttt{argVC}_n$. Also, note that $\texttt{zkNElemComm}_n$ is HVZK, whereas $\texttt{argVC}_n$ is not. Its properties 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 there holds $\mathrm{ort}(\mathbf{X})$*, for an element $Y \in \mathbb{G}$, the protocol* $\texttt{argVC}_n$ *in Figure 26 is a complete argument having WEE for the relation (63).*

Proof: is in Appendix U.

```
$\boxed{\mathtt{argVC}_n(\mathbf{X}, Y;\ \mathbf{x})}$
Relation $\mathcal{R} = \{\, \mathbf{X} \in \mathbb{G}^{n*},\ Y \in \mathbb{G};\ \mathbf{x} \in \mathbb{F}_{\bar{p}}^{n} \mid Y = \langle \mathbf{x}, \mathbf{X} \rangle \,\}$   // (63)
  // $\mathbf{X}$ in $\mathcal{R}$ satisfies 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 $\mathtt{argVC}_{\hat{n}}(\hat{\mathbf{X}}, \hat{Y};\ \hat{\mathbf{x}})$   // run recursively until n=4
else   // n $\leqslant$ 4
    $\boxed{\mathcal{P} \rightarrow \mathcal{V}}$: $\mathbf{x}$
    $\boxed{\mathcal{V}}$: returns $Accept$ iff the following holds
                            $Y \overset{?}{=} \langle \mathbf{x}, \mathbf{X} \rangle$
endif
```

Figure 26: Efficient argument for vector commitment

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

```
$\boxed{\mathtt{zkVC}_n^{\mathbf{opt}}(\mathbf{X}, H, Y;\ \mathbf{a}, \alpha)}$
Relation $\mathcal{R} = \{\, \mathbf{X} \in \mathbb{G}^{n*}, H \in \mathbb{G}^{*}, Y \in \mathbb{G};\ \mathbf{a} \in \mathbb{F}_{\bar{p}}^{n}, \alpha \in \mathbb{F}_{\bar{p}} \mid Y = \langle \mathbf{a}, \mathbf{X} \rangle + \alpha H \,\}$   // (3)
  // $\mathbf{X}$, $H$ in $\mathcal{R}$ satisfy 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 27: 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 there holds $\operatorname{ort}(\mathbf{X} \cup \{H\})$*, for an element $Y \in \mathbb{G}$, the protocol $\mathtt{zkVC}_n^{\mathbf{opt}}$ in Figure 27 is a complete, HVZK argument having WEE for the relation (3).*

Proof: is in Appendix V.

Note, that the magic of compressing the scalars $\boldsymbol{\tau}$ within the vector commitment argument $\mathtt{zkVC}_n^{\mathbf{opt}}$ looks similar to the one in the work [24]. However, our method is different. We compress prover's reply that looks random, whereas true randomness sampled by prover is compressed in [24]. In connection with this, providing an argument of knowledge for such the scalars is sufficient in our case. Whereas the cases as in [24], in our view, may either imply a weaker security model or require an additional proof that the scalars are indeed random, otherwise prover may tamper with them, more on this in Appendix W.

10.3 SIZES AND COMPLEXITIES

As a result, $\mathtt{zkVC}_n^{\mathbf{opt}}$ size is $2\lceil\log_2(n+1)\rceil + 1$. Substituting $\mathtt{zkVC}_n^{\mathbf{opt}}$ for $\mathtt{zkVC}_n$ in the Multratug and EFLRSL schemes, the new sizes of them are shown in Table 7. Their verification times do not change much, so we do not recalculate. For comparison, the former sizes and times are in Table 4 and Table 6. One more change is that 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	$2\lceil\log_2(n+l+1)\rceil + 7l + 4$	$\boldsymbol{mexp}(\,4n+8l+\dots) + (n+l+2)\mathbf{H}_{\mathbf{pt}}$
EFLRSL	$2\lceil\log_2(n+1)\rceil + 3l + 1$	$\boldsymbol{mexp}\,(\,3n+2l+\dots) + (n+1)\mathbf{H}_{\mathbf{pt}}$

. . . Insignificant summands are omitted.

11 IMPROVEMENTS

11.1 USING RING OF SIZE N·L

It is possible to slightly reduce the size of the Multratug scheme by not using the Lin2-2Choice lemma and instead repeating the ring l times, each time for k-th amount $A_k^{\mathbf{tmp}}$. In this case, after appropriate optimizations, the signature size would be

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

Nevertheless, we still prefer the version with the Lin2-2Choice lemma, because not using it implies the nl case, which 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 time.

11.2 BATCH VERIFICATION

Verification of a batch of Multratug signatures can be accomplished with checking just one equality, by combining Figure 22's equalities (*) and (**) of all signatures in the batch using random weighting. Of course, (*) 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 now.

In any case, the asymptotic verification complexity by ring size n under the multi-exponent decreases from $4n$ to $3n$ due to the fact, that all instances of the Multratug signature use the same vector of predefined generators $\mathbf{G}$. The same can be stated about EFLRSL by referring to Figure 15 and finding there a decrease from $3n$ to $2n$ under the multi-exponent.

11.3 COMBINING WITH OTHER PROOFS

Multratug is rooted in a single vector commitment argument and does not depend on the realization of the argument. Hence, Multratug can be combined with any other argument that uses a proof of vector commitment. For instance, it can be combined with the inner product argument implemented according to [5] or [7].

Namely, Multratug can be combined with single or aggregate range proofs from [7], and they will share the component

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

where $n^{\mathbf{rangeproof}}$ is equal to bitsize of the range times number of proofs aggregated.

12 APPLICATIONS

12.1 SIGNATURE IN BLOCKCHAIN

Suppose, Multratug is used to sign a transactions in an UTXO blockchain like, e.g., [16, 22]. Suppose, the blockchain public keys, hidden amounts, hash functions, and predefined generators follow the rules in Figures 1, 8, 19, 20. There is nothing unusual in these requirements for a blockchain. Moreover, the blockchain does not necessarily have to follow the CryptoNote rules for stealth addresses [22], although it may follow them.

For every transaction, its sender $\mathcal{P}$ executes the next procedure.

- Picks from the ledger n pairs of the form (P, A), which become transaction inputs, and makes the ring (31) 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 $\mathbf{A^{out}}$.
- Lets $A^{\mathbf{sum}} = \sum_{k=0}^{m-1} A_k^{\mathbf{out}}$.
- Signs the transaction with the Multratug signature, knowing actually signing private keys at indices in the vector $\mathbf{s}$.
- Proves ranges for all elements in $\mathbf{A^{out}}$, for example, using the aggregate range proof from [7] which is combinable with Multratug, as shown in Section 11.3.
- Proves that each $A_k^{\mathbf{out}} \in \mathbf{A^{out}}$ has a decomposition (33) with known to $\mathcal{P}$ coefficients. Notable, in the case if ranges for elements in $\mathbf{A^{out}}$ are proved with a protocol from [5, 7], then, for all $A_k^{\mathbf{out}}$'s, their decompositions (33) are proved by that.

Thus, the transaction contains proofs of the form (33) for each of the output hidden amounts $\mathbf{A^{out}}$. Also, the transaction contains Multratug, which contains a proof of 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 constitute a subset of all hidden amounts $\mathbf{A}$ in the ring, and the latter are assumed already proved having the form (33), from all these proofs it follows that the sum of amounts related to the actual signing keys is equal to the sum of the output amounts

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

Finally, at the same time, the same Multratug proves that $\mathcal{P}$ knows private keys for actually signing public keys at indices in $\mathbf{s}$ in the ring. It also provides the key images $\mathbf{I}$ which exclude reuse of those actually signing public keys for signing other transactions.

One may ask about how the assumption that all the ring's hidden amounts $\mathbf{A}$ have the form (33) is practically fulfilled. The answer is that, as it is accepted in the UTXO blockchains, each element $A \in \mathbf{A}$ already exists in the ledger and, hence, A belongs to $\mathbf{A^{out}}$ of some other transaction, and the latter contains a proof of (33) for A.

12.2 REGULAR RING SIGNATURE

EFLRSL is a simple linkable threshold ring signature, which, in terms of Table 2, has Log-sz, Regular, Linkable, Thresh., General check-marked. Consequently, it can be used for implementing various systems and scenarios, including electronic voting or whistleblowing described, e.g., in [15].

As for a not-linkable version of EFLRSL, it can be easily implemented by blinding the EFLRSL key images. This can be accomplished by creating the key images after defining the blinding generator H and adding random H-components to them.

13 COMPARISON

Now we will compare our optimized Multratug and EFLRSL schemes (Table 7) with the best performing ones listed in Table 1, namely, with Lelantus Spark [11], Omniring [13], RingCT3.0 [23], Triptych [17], and DualRing-EC [24], taking linear-size CLSAG [8] for the base.

We distinguish two gradations of scheme anonymity inherently bound to two key image (linking tag) forms used. In general, if a scheme has key image or another public element of the form $x^{-1}U$, then it has lower anonymity, unless a compensatory restriction is imposed on the keys. Key images in the forms $x^{-1}\mathcal{H}_{\mathbf{point}}(P)$ and $x\mathcal{H}_{\mathbf{point}}(P)$ do not require any restrictions for the keys. However, it is still required that the scheme has no other elements of the form $x^{-1}U$. More on this in Appendix X.

13.1 FOR MULTRATUG

In Table 8 we compare the schemes with balance proofs. Notation is following: $\mathbf{H_{sc}}$ 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-exponentition 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 [22].

Our signature receives 'Any keys=Yes', since according to Theorem 15, and hence according to Theorem 6, it has the EU_CMA/CPA, anonymity w.r.t. CPA, non-frameability w.r.t. CPA properties, whose proofs are identical to their proofs for LSAG or CLSAG.

Lelantus Spark [11] has key image $x^{-1}U$, nevertheless, according to its 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 count only the size of its parallel 1-out-of-many proof from the section '7 Efficiency' of [11], the actual size may have a few extra bytes.

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 by Section 11.3 our scheme effectively integrates with them, as does Omniring [13]. Batch verification time, for our scheme explained in Section 11.2, is generally 25%. . . 50% less for all log-size schemes due to common generators merging, we do not show it.

So of course, the schemes with key image forms $x^{-1}\mathcal{H}_{\mathbf{point}}(P)$ and $x\mathcal{H}_{\mathbf{point}}(P)$ have an additional summand of roughly $n\mathbf{H_{pt}}$ in their verification complexity formulas.

Multratug is represented by its version with optimized vector commitment argument, with characteristics taken from Table 7. Note, we subtracted l from its size, since key images are not counted. The CLSAG, Triptych, and Lelantus Spark schemes have no threshold versions, so, for the comparison 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 [23]. The same is for Omniring, its size is taken from the section '6.3 Performance Comparison' of [13]. Note, according to its paper, Omniring has $O\log_2(nl + \dots)$ size, whereas in the section 'D Comparison with Omniring' in [23] it reads as $O\log_2(n + \dots)$, we hold to the first one.

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*	$3\lceil\log_2(n)\rceil + 8$	$\boldsymbol{mexp}(\,2n + \dots)$	$x^{-1}U$	No	No
Lelantus Spark*	$3\lceil\log_2(n)\rceil + 5$	$\boldsymbol{mexp}(\,2n + \dots)$	$x^{-1}U$	No	Yes
RingCT3.0	$2\lceil\log_2(n\,l)\rceil + l + 17$	$\boldsymbol{mexp}(\,2\,n\,l + \dots) + \boldsymbol{mexp}(l+1) + \dots$	$x^{-1}U$	No	No
Omniring	$2\lceil\log_2(n\,l + n + 3l + 3)\rceil + 9$	$\boldsymbol{mexp}(\,2\,n\,l + \dots)$	$x^{-1}U$	No	No
Omniring	$2\lceil\log_2(n\,l + n + 3l + 3)\rceil + 9$	***	$x\,\mathcal{H}_{\mathbf{point}}(P)$	No	Yes
Multratug**	$2\lceil\log_2(n + l + 1)\rceil + 6l + 4$	$\boldsymbol{mexp}(\,4n + 8l + \dots) + (n + l + 2)\mathbf{H_{pt}}$	$x\,\mathcal{H}_{\mathbf{point}}(P)$	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.

According to Table 8, assuming ring size is, say, $n = 2^5 \dots 2^{10}$, and the number of inputs is limited by, say, $l \leqslant 5$ which is in line with [23, 13, 17], Multratug performs on par with the best schemes.

As for applicability in blockchain, we should probably only consider signatures that allow for easy signing by multiple parties, since this seems to be a must-have attribute for modern blockchains. Therefore, only Lelantus Spark, Omniring version with $x\mathcal{H}_{\mathbf{point}}(P)$, and our signature are to be compared. Table 9 shows their sizes (excluding key images and range proofs) in bytes computed in the region of interest, assuming an element in $\mathbb{G}$ and a scalar in $\mathbb{F}_{\bar{p}}$ take 32 bytes each.

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

In addition to this, for a blockchain or another system that requires an anonymous log-size signature with balance proofs, multiparty signing, and arbitrary public keys (namely, keys that do not follow [22], such as generated ad-hoc as in [16]), of all the currently known schemes only ours will do.

13.2 FOR EFLRSL

In Table 10 we compare the simplest versions of the schemes, which are 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 [24] signature, which, according to the survey in [24], is the shortest 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, in this case the linkable version usually takes up one more element of space. The sizes of DualRing-EC and of the streamlined RingCT3.0 and Omniring versions are taken from 'Table 1: O(log n)-size DL-based ring signature schemes for n public keys . . . ' in [24].

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 shortest size. However, EFLRSL has a stronger security model, which is explained in Appendix W. Thus, it appears that the EFLRSL signature for $l = 1$ is the shortest one known to date for environments in which malformed keys are permitted.

Table 10: Comparison of DL-based ring signatures

	Size	Verification complexity
CLSAG	$n + 1$	$n\,\mathbf{H_{sc}} + n\,\boldsymbol{mexp}(2)$
RingCT3.0	$2\lceil \log_2(n) \rceil + 14$	$\boldsymbol{mexp}(\,2n + \dots) + \dots$
Omniring	$2\lceil \log_2(n + 2) \rceil + 9$	$\boldsymbol{mexp}(\,2\,n\,l + \dots)$
EFLRSL[*]	$2\lceil \log_2(n + 1) \rceil + 4$	$\boldsymbol{mexp}(\,3n + \dots) + (n + 1)\mathbf{H_{pt}}$
DualRing-EC[**]	$2\lceil \log_2(n) \rceil + 4$	$\boldsymbol{mexp}(\,n + \dots)$

[*] Only linkable version of the ring signature is available.

[**] See comments in Appendix W.

. . . Insignificant summands are omitted.

14 CONCLUSION

In this paper we presented two novel efficient membership proofs in a group without bilinear pairings, under DDH assumption. These membership proofs are proved complete, special honest verifier zero-knowledge, and having computational witness-extended emulation in the lemmas called Lin2-Choice and Lin2-2Choice.

With 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 actually signing keys in it, Multratug occupies less than 2KBytes of space, as shown in Table 9.

In addition to quite moderate size and balance proof, our 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 Multratug applicable to various cryptographic systems, including and not limited by blockchain. Therefore, Multratug may serve as a log-size replacement for the well-known linear-size LSAG scheme and its extensions.

Our survey has shown that among the existing log-size schemes, for large rings and medium thresholds, only a version of the Omniring scheme comprises almost the same wide set of useful features (Table 1, Table 2) at a minimal size (Table 8). However, the Multratug scheme still tolerates malformed keys better.

Apart from blockchains, for the case if a cryptographic system requires neither a balance proof nor the other modern 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).

It should be noted that since our membership proofs, and hence our signatures, rely on the simplest vector commitment argument (as we define it for prime-order group in Section 1.2), they effectively combine with other arguments, such as range proofs, to further reduce total proof 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 appropriate works where the necessary details can be found.

Because of the modular design, it is sufficient to check all the blocks individually in order to understand and verify our schemes. Although, some of these elementary protocols, such as the random weighting for t-s-tuples argument or the optimized vector commitment argument, are far from trivial and may have independent value.

ACKNOWLEDGEMENTS

Author thanks Tsz Hon Yuen et al. for the work [24], which inspired some ideas for improvement, and also for our brief email conversation about cryptographic games and definitions. Plus, I appreciate the valuable comments of an anonymous reviewer and am grateful to the readers who advised the necessary 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] 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.

[5] 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.

[6] Dario Catalano and Dario Fiore. *Vector Commitments and their Applications*. Cryptology ePrint Archive, Paper 2011/495. 2011. URL: https://eprint.iacr.org/2011/495.

[7] Heewon Chung et al. *Bulletproofs+: Shorter Proofs for Privacy-Enhanced Distributed Ledger*. Cryptology ePrint Archive, Report 2020/735. https://ia.cr/2020/735. 2020.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] Satoshi Nakamoto. *Bitcoin: A peer-to-peer electronic cash system*. https://bitcoin.org/bitcoin.pdf. 2008.

[17] 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.

[18] 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.

[19] Claus-Peter Schnorr. "Efficient Signature Generation by Smart Cards". In: *J. Cryptology* 4.3 (1991), pp. 161–174.

[20] 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.

[21] Patrick P. Tsang et al. *Separable Linkable Threshold Ring Signatures*. Cryptology ePrint Archive, Report 2004/267. `https://ia.cr/2004/267`. 2004.

[22] Nicolas Van Saberhagen. *CryptoNote v 2.0*. `https://cryptonote.org/whitepaper.pdf`. 2013.

[23] 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.

[24] 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.

A PROOF OF 2-ELEMENT COMMITMENT

Proof: [Theorem 1] Completeness of the protocol is seen from its code. Also, in the case if T is a direct weighted sum of $\{X, H\}$, then the protocol splits into two independent Schnorr identification schemes [19] with the same challenge. Thus, if this is the case, then HVZK, and WEE of the protocol in Figure 2 is proved the same way as for the Schnorr id scheme.

Suppose, this is not the case, i.e., prover sends T without knowing its relation to $\{X, H\}$, in other words, having $T\ != \text{lin}(X, H)$. Then, for the prover, if it has $Y = \text{lin}(X, H)$, then after successful completion of the protocol it has $T = \text{lin}(X, H)$, which contradicts to the supposition. Otherwise, if it has $Y = \ != \text{lin}(X, H)$, then having rewound the protocol and excluded T, it obtains $Y = \text{lin}(X, H)$, which is a contradiction again.

Thus, the HVZK and WEE properies of the protocol are proved. They also can be proved the same way as for the other Schnorr-like protocols in [2, 4, 7, 20].

B PROOF OF VECTOR COMMITMENT

Proof: [Theorem 2] The `zkVC`$_n$ protocol in Figure 3 is a slightly modified subset version of the Bulletproofs logarithmic inner product argument from [5]. There are following three modifications to it

- The inner product argument described in [5] has no HVZK property, we append this property to it the same way as this is done in [7], namely, by adding a blinding component to all transmitted elements. We omit providing a proof of HVZK for our `zkVC`$_n$ protocol here; it is identical to the HVZK proof in [7].
- With the above modification, the `zkVC`$_n$ protocol in Figure 3 is a subset case, namely $\mathbf{b} = \mathbf{0}^n$, of the inner product argument from [5] for the relation (4). Thus, our protocol is an argument for the relation (3).
- For the case $n = 1$ in `zkVC`$_n$ we use the custom zero-knowledge `zk2ElemComm` protocol, which is complete, HVZK, and has WEE by Theorem 1.

Each of the above three modifications clearly does not override the completeness and WEE properties of the Bulletproofs logarithmic inner product argument. Also, the first modification adds the HVZK property. Thus, our protocol `zkVC`$_n$ in Figure 3 is a complete, HVZK argument having WEE for the relation (3).

C PROOF OF 3-TUPLE RANDOM WEIGHTING

Proof: [Theorem 3] Completeness and HVZK 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 of it, which in its turn is complete and HVZK by the premise.

WEE property of the `zk3ElemRW` protocol is also easy to establish, we will not provide a detailed proof here to save space, only the following sketch.

First, note that due to orthogonality of H to all other generators, components proportional to H of all participating elements can be considered separately and be omitted in the main consideration. For the H components of the protocol it is enough to calculate the factor $\hat{\alpha}$ as $\hat{\alpha} = \alpha + \delta_1\beta + \delta_2\gamma$ only.

Second, witness extraction can be accomplished in a well-known way, e.g., as in the proof of the RandomWeighting-WEE lemma in [20].

Third, to ascertain that the witness a has only one possible value in this protocol, we can write Z, F, E as

$$\begin{cases} Z = z_P P + z_Q Q + z_R R \\ F = f_P P + f_Q Q + f_R R \\ E = e_P P + e_Q Q + e_R R \end{cases} , \tag{64}$$

since it is clear that, when H is already excluded from the consideration, the elements Z, F, E cannot have components beyond the linear span of P, Q, R without breaking the DL assumption. Inserting the decomposition (64) into the equality $Y = aX$, we obtain

$$\operatorname{rank}\left(\begin{bmatrix} 1 & \delta_1 \text{ or } 0, \text{if } Q = 0 & \delta_2 \text{ or } 0, \text{if } R = 0 \\ z_P + \delta_1 f_P + \delta_2 e_P & z_Q + \delta_1 f_Q + \delta_2 e_Q & z_R + \delta_1 f_R + \delta_2 e_R \end{bmatrix}\right) < 2 ,$$

which immediately yields the sought relation, namely, that for some unique (and, hence, extractable) witness a there holds, to the accuracy of H components,

$$\begin{cases} Z = aP \\ F = aQ \\ E = aR \end{cases} ,$$

and from where it can be understood why we are demanding $P \neq 0 \wedge (Q \neq 0 \vee R \neq 0)$.

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, HVZK, and WEE protocol called in the last step of it (it can be, say, $\texttt{zkVC}_n$), thus inheriting the HVZK property from the latter. Completeness of the protocol $\texttt{zkSVC}_{3,n}$ is clear. WEE 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 WEE property of $\texttt{zkSVC}_{3,n}$ it suffices to prove WEE for $\texttt{zkSVC}_{3,1}$.

In its turn, $\texttt{zkSVC}_{3,1}$ is equivalent to the protocol zk3ElemRW in Figure 4, so by Theorem 3 $\texttt{zkSVC}_{3,1}$ has WEE. Thus we obtain WEE for $\texttt{zkSVC}_{3,n}$.

E PROOF OF LIN2-CHOICE LEMMA

Proof: [Theorem 5] Completeness and HVZK of the $\texttt{zkLin2Choice}_n$ protocol in Figure 7 are clear. We exclude H from all considerations for the same reason as in Appendix C.

Let's prove the WEE property of the protocol. 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 there holds the relation

$$\begin{cases} Z = \langle \mathbf{a}, \mathbf{P} \rangle \\ rF = \langle \mathbf{a}, \mathbf{c} \circ \mathbf{Q} \rangle \end{cases} , \tag{65}$$

where $\mathbf{a} \in \mathbb{F}_{\bar{p}}^n$ is extracted by the $\texttt{zkSVC}_{2,n}$ protocol 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 **a** cannot contain more than one nonzero scalar, otherwise the $\texttt{zkLin2Choice}_n$ protocol extractor is 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 $j \neq k$. Writing out Z and rF as weighted direct sums of $\mathbf{P}$ and $\mathbf{Q}$, respectively, according to the equalities (65) we obtain that having unwound the $\texttt{zkSVC}_{2,n}$ call the extractor has $Z, F, \mathbf{c}, r, \mathbf{a}$ such that the following two equalities hold

$$Z = \sum_{i=0}^{n-1} a_i P_i \,, \tag{66}$$

$$rF = \sum_{i=0}^{n-1} a_i c_i Q_i \,, \tag{67}$$

where $r \neq 0$, otherwise the equality (67) would immediately produce a contradiction with $\mathrm{ort}(\mathbf{Q})$.

Let the extractor unwinds to the point where the challenges $\mathbf{c}$ were generated, and resumes obtaining new $\mathbf{c}', r', \mathbf{a}'$. Thus, by the equality (67) there holds $r' \neq 0$, and by the equality (66) there holds $\mathbf{a}' = \mathbf{a}$. By excluding F from the equality (67) the extractor obtains

$$0 = \sum_{i=0}^{n-1} a_i \left(\frac{c_i}{r} - \frac{c'_i}{r'}\right) Q_i \,. \tag{68}$$

Due to $\mathrm{ort}(\mathbf{Q})$ all weights of Q_i's in the equality (68) must be zero, otherwise the extractor breaks the DL assumption.

According to our supposition, $a_j \neq 0$ and $a_k \neq 0$, so we write out two equations for the weights of Q_j and Q_k

$$\begin{cases} 0 = & \frac{c_j}{r} - \frac{c'_j}{r'} \\ 0 = & \frac{c_k}{r} - \frac{c'_k}{r'} \end{cases}, \tag{69}$$

where we have already performed division by nonzero a_j and a_k. Since $r \neq 0$ and $r' \neq 0$, the system (69) reduces to

$$\frac{c_k}{c'_k} = \frac{c_j}{c'_j}, \tag{70}$$

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 (68). Thus, under our supposition, the extractor breaks the DL assumption by expressing Q_j through the elements of $\mathbf{Q} \setminus \{Q_j\}$, hence our supposition is incorrect.

By this we have proved that the extractor with overwhelming probability finds witness for the relation (12) and, thus, the protocol $\mathtt{zkLin2Choice}_n$ has WEE.

F SIGNATURE EFLRS1

Proof: [Theorem 6] As follows from Figure 10, EFLRS1 is a linkable ring signature by definition (we imply the EFLRS1.Link method is defined usual way, i.e., matching key images, e.g., as in [15]).

All the listed properties of the EFLRS1 signature are proved by well-known methods, such as, for example, in [15, 8, 20], which rely on the key image of the form of $x^{\pm 1}\mathcal{H}_{\mathbf{point}}(P)$ and on completeness, HVZK, and WEE 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.

G PROOF OF MULTIPLE VECTOR COMMITMENTS

Proof: [Theorem 7] As can be seen from Figure 12, the protocol $\mathtt{zkMVC}_{l,n}$ adds nothing to the transcript of the protocol $\mathtt{zkVC}_n$, thus inheriting the HVZK property. Completeness of the protocol $\mathtt{zkMVC}_{l,n}$ is clear. Let's prove the protocol WEE property.

This time, to show an example, we will not exclude the generator H from our consideration. We add 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 $\bar{\mathbf{a}} \in \mathbb{F}_{\bar{p}}^{n+1}$

$$\bar{\mathbf{a}} = \begin{bmatrix} \mathbf{a} \\ \alpha \end{bmatrix}.$$

Having unwound the $\mathtt{zkVC}_n$ call, extractor obtains $\bar{\mathbf{a}}$. As a result, for each i-th column $\mathfrak{a}_{[:,i]}$ of the matrix $\mathfrak{a}$ there holds the equality

$$\bar{\mathbf{a}}_{[i]} = \boldsymbol{\xi}^\top \cdot \bar{\mathfrak{a}}_{[:,i]} \,. \tag{71}$$

The extractor repeats the unwinding l times with re-sampled challenges $\boldsymbol{\xi}$. This way the equality (71) 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 $\bar{\mathfrak{a}}$.

H PROOF OF THE PROPERTIES OF MANY-OUT-OF-MANY PROOF

Proof: [Theorem 8] Completeness and HVZK of the $\texttt{zkLin2mChoice}_{n,l}$ protocol in Figure 13 are clear. Let's prove the WEE 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 $(\mathfrak{a}, \hat{\alpha})$ from the $\texttt{zkMVC}_{l,n}$ call in the last step of $\texttt{zkLin2mChoice}_{n,l}$. After that, for every $k \in [0 \dots l-1]$, it assigns

$$(\mathbf{a}, \hat{\alpha}) \leftarrow (\mathfrak{a}_{[k]}, \hat{\alpha}_{[k]}) ,$$

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 (18) and, hence, the $\texttt{zkLin2mChoice}_{n,l}$ protocol has WEE.

I SIGNATURE EFLRSL FOR L=1

As can be seen from Figure 14, for $l = 1$ the EFLRSL protocol is the same as the EFLRS1 protocol in Figure 10, with the variables and calls renamed. Although, the multiplier ξ_0 is applied to both commitment and witness in the nested $\texttt{zkVC}_n$ call. Anyway, this doesn't distort the correspondence. Thus, by Theorem 6, for $l = 1$, all the properties listed in Theorem 9 hold.

J SIGNATURE EFLRSL FOR L $\geqslant$ 1

Proof: [Theorem 9] A proof for the case $l = 1$ is provided in Appendix I.

As can be seen from Figure 14, the EFLRSL protocol is a linkable threshold ring signature by definition (we imply the EFLRSL.Link method is defined usual way, i.e., matching key images).

All the listed properties of the EFLRSL signature can be proved by well-known methods, for example, by assuming that any of the properties does not hold, and reducing this case to the case $l = 1$, i.e., to the contradiction with the already proven in Appendix I. In this case, similar to, e.g., [15, 21, 8], key image form $x^{\pm 1}\mathcal{H}_{\mathbf{point}}(P)$ and completeness, HVZK, and WEE of the underlying proving system are used.

We do not present the proofs here because of their volume, referring the interested reader to the referenced publications.

K PROOF OF SIMPLIFIED LIN2-2CHOICE LEMMA

Proof: [Theorem 10] Completeness and HVZK properties of the $\texttt{zkLin22sChoice}_{n,m}$ protocol in Figure 16 are clear. We exclude H from the consideration for the same reason as in Appendix C.

Let's prove the protocol WEE property. In the last step of $\texttt{zkLin22sChoice}_{n,m}$ there is a call to

$$\texttt{zkSVC}_{3,n}\left(\begin{bmatrix}\mathbf{P}\\ \mathbf{V}\end{bmatrix}, \begin{bmatrix}\mathbf{c}_{[:n]} \circ \mathbf{Q}\\ \mathbf{0}^m\end{bmatrix}, \begin{bmatrix}\mathbf{0}^n\\ \mathbf{c}_{[n:]} \circ \mathbf{W}\end{bmatrix}, H, Z, rF, c_{n+t}E; \mathbf{a}, \alpha, \hat{\beta}, \hat{\gamma}\right) ,$$

and hence by Theorem 4 there holds the relation

$$\begin{cases} Z &= \langle \mathbf{a}_{[:n]}, \mathbf{P}\rangle \quad + \quad \langle \mathbf{a}_{[n:]}, \mathbf{V}\rangle \\ rF &= \langle \mathbf{a}_{[:n]}, \mathbf{c}_{[:n]} \circ \mathbf{Q}\rangle \\ c_{n+t}E &= \qquad\qquad\qquad\quad \langle \mathbf{a}_{[n:]}, \mathbf{c}_{[n:]} \circ \mathbf{W}\rangle \end{cases} , \tag{72}$$

with the witness $\mathbf{a} \in \mathbb{F}_{\mathrm{p}}^{n+m}$ restored by the $\texttt{zkSVC}_{3,n}$ protocol extractor.

Due to ort$(\mathbf{P}, \mathbf{V}, \mathbf{Q}, \mathbf{W})$, having $Z = Z_P + Z_V$ according to the formula (27), the system (72) breaks down into two subsystems

$$\begin{cases} Z_P &= \langle \mathbf{a}_{[:n]}, \mathbf{P}\rangle \\ rF &= \langle \mathbf{a}_{[:n]}, \mathbf{c}_{[:n]} \circ \mathbf{Q}\rangle \end{cases} , \tag{73}$$

$$\begin{cases} Z_V &= \langle \mathbf{a}_{[n:]}, \mathbf{V}\rangle \\ c_{n+t}E &= \langle \mathbf{a}_{[n:]}, \mathbf{c}_{[n:]} \circ \mathbf{W}\rangle \end{cases} . \tag{74}$$

Each of the systems (73), (74) is similar to the system (65) and, therefore, by applying the same reasons to each of them as in the proof of the WEE property of the Lin2-Choice lemma in Appendix E, we obtain the following two equations respectively

$$Z_P = pP_s \ , \tag{75}$$

$$Z_V = vV_{n+\tilde{s}} \ , \tag{76}$$

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). Furthermore, when obtaining the equality (75) from the subsystem (73), we take r as a response to the challenges $\mathbf{c}_{[:n]}$, whereas obtaining the equality (76) from the subsystem (74), 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 (76) as

$$Z_V = vV_{n+t} \ . \tag{77}$$

Now, recalling that Z decomposes into the sum $Z = Z_P + Z_V$ by the formula (27) which is discussed in Section 7.1.1, the extractor comes to the conclusion that the restored by the formulas (75), (77) values of (p, v, s) are the sought witnesses for the relation (20). Thus, we have proved the WEE property of $\mathtt{zkLin22sChoice}_{n,m}$.

L PROOF OF MULTIPLE SIMMETRIC VECTOR COMMITMENTS

Proof: [Theorem 11] As can be seen from Figure 17, the $\mathtt{zkMSVC}_{l,3,n}$ protocol adds nothing to the transcript of the $\mathtt{zkMVC}_{l,n}$ protocol, thus inheriting the HVZK property. Completeness of the $\mathtt{zkMSVC}_{l,3,n}$ protocol is clear from Figure 17. We exclude H from all considerations for the same reason as in Appendix C.

Let's prove the WEE property of the protocol. Having unwound the $\mathtt{zkMVC}_{l,n}$ call, extractor obtains a matrix $\mathfrak{a} \in \mathbb{F}_{\bar{\mathrm{p}}}^{l\times n}$ such that according to the relation (17)

$$\mathbf{Y} = \mathfrak{a} \cdot \mathbf{X} \, . \tag{78}$$

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 $\mathfrak{a}$, there holds

$$Y_j = \mathfrak{a}_{[j,:]} \cdot \mathbf{X} \, . \tag{79}$$

At the same time, due to the equalities (79), the $\mathtt{zkMVC}_{l,n}$ protocol can be viewed as l independent, except for the common challenges (δ_1, δ_2), instances of the $\mathtt{zkSVC}_{3,n}$ protocol. Therefore, by Theorem 4, the restored by the extractor matrix $\mathfrak{a}$ is the sought witness.

M PROOF OF LIN2-2CHOICE LEMMA

Proof: [Theorem 12] Completeness and HVZK of the protocol $\mathtt{zkLin22Choice}_{l,n,m}$ in Figure 18 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 WEE property. In the last step of $\mathtt{zkLin22Choice}_{l,n,m}$ there is a call to

$$\mathtt{zkMSVC}_{l,3,(n+m)}\left(\begin{bmatrix}\mathbf{P}\\ \mathbf{V}\end{bmatrix}, \begin{bmatrix}\mathbf{c}_{[:n]} \circ \mathbf{Q}\\ \mathbf{0}^m\end{bmatrix}, \begin{bmatrix}\mathbf{0}^n\\ \mathbf{c}_{[n:]} \circ \mathbf{W}\end{bmatrix}, H, \mathbf{Z}, \mathbf{r} \circ \mathbf{F}, \mathbf{c}_{[n:(n+l)]} \circ \mathbf{E};\ \mathfrak{a}, \alpha, \hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\gamma}}\right) ,$$

and hence, by Theorem 11, there holds the following system of equalities

$$\begin{cases} \mathbf{Z} & = \mathfrak{a} \cdot \begin{bmatrix}\mathbf{P}\\ \mathbf{V}\end{bmatrix} \\ \mathbf{r} \circ \mathbf{F} & = \mathfrak{a} \cdot \begin{bmatrix}\mathbf{c}_{[:n]} \circ \mathbf{Q}\\ \mathbf{0}^m\end{bmatrix} \\ \mathbf{c}_{[n:(n+l)]} \circ \mathbf{E} & = \mathfrak{a} \cdot \begin{bmatrix}\mathbf{0}^n\\ \mathbf{c}_{[n:]} \circ \mathbf{W}\end{bmatrix} \end{cases} , \tag{80}$$

where the matrix $\mathfrak{a} \in \mathbb{F}_{\bar{\mathrm{p}}}^{l\times(n+m)}$ is the witness restored by the $\mathtt{zkMSVC}_{l,3,(n+m)}$ protocol extractor.

Furthermore, the system (80) is l systems of the form (72), with proper renaming, for each row $\mathfrak{a}_{[t,:]}, t \in [0 \dots l-1]$ of the matrix $\mathfrak{a}$. Namely, the system (80) is the following l systems

$$\begin{cases} Z_t & = \langle \mathfrak{a}_{[t,:n]}, \mathbf{P} \rangle \quad + \quad \langle \mathfrak{a}_{[t,n:]}, \mathbf{V} \rangle \\ r_t F_t & = \langle \mathfrak{a}_{[t,:n]}, \mathbf{c}_{[:n]} \circ \mathbf{Q} \rangle \\ c_{n+t} E_t & = \qquad\qquad\qquad \langle \mathfrak{a}_{[t,n:]}, \mathbf{c}_{[n:]} \circ \mathbf{W} \rangle \end{cases}, \tag{81}$$

for each $t \in [0 \dots l-1]$.

The $\texttt{zkLin22Choice}_{l,n,m}$ protocol in Figure 18 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 16. Hence, given l parallel systems (81) for $t \in [0 \dots l-1]$, the extractor performs l times, for each t, the same calculations as in Appendix K. This way it obtains l witnesses (p_t, v_t, s_t), $t \in [0 \dots l-1]$ for l instances of the relation (20). That is, for each extracted tuple (p_t, v_t, s_t) there holds

$$Z_t = p_t P_{s_t} + v_t V_t , \tag{82}$$

that means witnesses for the relation (30) are found and, hence, WEE property of the $\texttt{zkLin22Choice}_{l,n,m}$ protocol is proven.

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 21 proves the relation (30).

Let's demonsrate that this call also proves that $\mathbf{v} = \mathbf{p}$ in the relation (30), where $\mathbf{X}, \mathbf{V}, \mathbf{Z}$ are defined according to the EFLRSLWB scheme. Writing out their definitions 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 $k \in [0 \dots l-1]$ there holds $v_k \neq p_k$. Then the $\texttt{zkLin22Choice}_{l,n,m}$ protocol extractor extracts such $\mathbf{v}, \mathbf{p}$, and for some index s_k there holds, according to relation (30)

$$G + \zeta I_k + \chi J_k = p_k (P_{s_k} - K + \zeta U_{s_k} - \omega A_{s_k}) + v_k (K + \omega A_k^{\mathbf{tmp}} + \chi \hat{U}_k) . \tag{83}$$

Note that we omit writting out 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 the extraction is straightforward.

By moving the K component to the left-hand side of the (83) equality, the extractor gets

$$(p_k - v_k) K = -G - \zeta I_k - \chi J_k + p_k (P_{s_k} + \zeta U_{s_k} - \omega A_{s_k}) + v_k (\omega A_k^{\mathbf{tmp}} + \chi \hat{U}_k) , \tag{84}$$

that is, it expresses K as a linear combination (84) 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 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, thus the supposition is incorrect and there holds

$$\mathbf{v} = \mathbf{p} . \tag{85}$$

Using the equality (85), the equality (83) rewrites as

$$G + \zeta I_k + \chi J_k = p_k (P_{s_k} + \zeta U_{s_k} + \chi \hat{U}_k + \omega (A_k^{\mathbf{tmp}} - A_{s_k})) . \tag{86}$$

Note that in the equality (86) the following holds for p_k's

$$p_k \neq 0 \quad \text{for each } k \in [0 \dots l-1] . \tag{87}$$

In fact, $p_k = 0$ for some k requires that the left-hand side of the equality (86) 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 (86) does not change the case; if the assertion (87) does not hold then the extractor breaks the DL assumption.

All elements in the right-hand part of the relation (86), 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 random weighting by χ, to the accuracy of H, the following equality holds

$$G + \zeta I_k = p_k(\, P_{s_k} + \zeta U_{s_k} + \omega(A_k^{\mathbf{tmp}} - A_{s_k})\,)\ . \tag{88}$$

In other words, the equality (88) follows from the equality (86) 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, subtracting two instances of the equality (88) 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

$$(p'_k - p_k)(P_{s_k} + \zeta U_{s_k}) = (p_k\omega - p'_k\omega')(A_k^{\mathbf{tmp}} - A_{s_k})\ . \tag{89}$$

Due to the orthogonality of P_{s_k} and U_{s_k} in the EFLRSLWB scheme, there holds

$$(P_{s_k} + \zeta U_{s_k}) \neq 0.$$

If $p'_k = p_k$ then the left-hand side of the equality (89) 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 (89) by $(p'_k - p_k)$, calculating scalar factor a as follows

$$P_{s_k} + \zeta U_{s_k} = a\,(A_k^{\mathbf{tmp}} - A_{s_k})\,,\ \text{where } a = \frac{p_k\omega - p'_k\omega'}{p'_k - p_k}\ . \tag{90}$$

Unwinding and resuming the $\texttt{zkLin22Choice}_{l,n,l}$ call with different ζ' a couple of times, the extractor calculates factor a' such that

$$P_{s_k} + \zeta' U_{s_k} = a'\,(A_k^{\mathbf{tmp}} - A_{s_k})\,. \tag{91}$$

Subtracting the equality (90) from the equality (91) and dividing by $(\zeta' - \zeta)$, which is nonzero with overwhelming probability, the extractor obtains

$$U_{s_k} = \frac{a' - a}{\zeta' - \zeta}\,(A_k^{\mathbf{tmp}} - A_{s_k})\,. \tag{92}$$

Also, it obtains from the equalities (90) and (92)

$$P_{s_k} = \left(a - \zeta\,\frac{a' - a}{\zeta' - \zeta}\right)\,(A_k^{\mathbf{tmp}} - A_{s_k})\,. \tag{93}$$

After that, as $U_{s_k} \neq 0$, and hence $(a' - a) \neq 0$ in the equality (92), the extractor expresses $(A_k^{\mathbf{tmp}} - A_{s_k})$ through P_{s_k} with it and inserts $(A_k^{\mathbf{tmp}} - A_{s_k})$ into the equality (93), thus obtaining

$$P_{s_k} = \left(a - \zeta\,\frac{a' - a}{\zeta' - \zeta}\right)\,\frac{\zeta' - \zeta}{a' - a}\,U_{s_k}\ . \tag{94}$$

Recalling P_{s_k} and U_{s_k} are orthogonal to each other the extractor breaks the DL assumption with the equality (94), thus the supposition is wrong and there holds

$$A_k^{\mathbf{tmp}} = A_{s_k}\ . \tag{95}$$

In accordance with the equality (95) the equality (88), which is obtained by the extractor after unwinding the $\texttt{zkLin22Choice}_{l,n,l}$ call, rewrites as

$$G + \zeta I_k = p_k(P_{s_k} + \zeta U_{s_k})\ , \tag{96}$$

where p_k is known to the extractor. Thus the $\texttt{zkLin22Choice}_{l,n,l}$ call is an argument having WEE property for the relation (97).

At the same time, according to the obtained by the extractor equality (95) the same $\texttt{zkLin22Choice}_{l,n,l}$ call is an argument having WEE for the relation (98). Completeness and HVZK of the call follow from Theorem 12. Claim 1 is proven.

O SIGNATURE EFLRSLWB FOR L $\geqslant$ 1

Proof: [Theorem 13] We first make the following claim.

Claim 1:
The call to `zkLin22Choice`$_{l,n,l}$ *in the last step of the EFLRSLWB scheme in Figure 21 is a complete, HVZK argument having WEE for the relation (18) with appropriate input renaming, i.e., for the relation*

$$\mathcal{R} = \left\{ \begin{array}{l} (\mathbf{P} + \zeta\mathbf{U}),\ \mathbf{G}_{[:n]} \in \mathbb{G}^{n*}, H \in \mathbb{G}^*,\ (\{G\}^l + \zeta\mathbf{I}) \in \mathbb{G}^l; \\ \mathbf{s} \in [0\dots n-1]^l, \mathbf{p}, \boldsymbol{\alpha} \in \mathbb{F}_{\bar{p}}^l \end{array} \middle| \begin{array}{l} \forall k \in [0\dots l-1] : \\ G + \zeta I_k = p_k(P_{s_k} + \zeta U_{s_k}) + \alpha_k H \end{array} \right\}, \tag{97}$$

and is also a complete, HVZK argument having WEE for the relation

$$\mathcal{R}' = \left\{ \begin{array}{l} \mathbf{A} \in \mathbb{G}^n,\ \mathbf{A}^{\mathbf{tmp}} \in \mathbb{G}^l,\ H \in \mathbb{G}^* ; \\ \mathbf{s} \in [0\dots n-1]^l,\ \boldsymbol{\beta} \in \mathbb{F}_{\bar{p}}^l \end{array} \middle| \begin{array}{l} \forall k \in [0\dots l-1] : \\ A_k^{\mathbf{tmp}} = A_{s_k} + \beta_k H \end{array} \right\}, \tag{98}$$

such that witness **s** *is common for both relations (97) and (98).*

Proof: is in Appendix N.

Note, that the vectors $\mathbf{A}^{\mathbf{tmp}}$ and $\mathbf{J}$ in Figure 21 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, HVZK, and WEE proving system `zkLin22Choice`$_{l,n,l}$ that produces a proof of the relation (97), which is actually the relation (18) with proper renaming. Also, as we can see in Figure 21, all previous steps of the EFLRSLWB scheme do all the play of the EFLRSL scheme from Figure 14 up to the proof of the relation (18). As for the vectors $\mathbf{A}^{\mathbf{tmp}}$ and $\mathbf{J}$ which are all indistinguishable from white noise, they can be discarded as uninfluential when considering the relation (97). Thus, we see that the EFLRSLWB scheme is the EFLRSL scheme with the substituted underlying proving system, which is also complete, HVZK, and WEE.

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) holds due to the `zk2ElemComm` call in the last step of the EFLRSLWB scheme. By Theorem 1 there holds

$$A^{\mathbf{sum}} = \sum_{k=0}^{l-1} \mathbf{A}_{[k]}^{\mathbf{tmp}} + f_H H + f_D D \ , \tag{99}$$

where f_H, f_D are scalars known to prover. At the same time, by Claim 1 according to the relation (98), the equality (99) unfolds as

$$A^{\mathbf{sum}} = \sum_{k=0}^{l-1} A_{s_k} + \left(f_H + \sum_{k=0}^{l-1} \beta_k \right) H + f_D D \ . \tag{100}$$

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 (100) 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 HVZK properties of the `zkTElemRW`$_{t,s}$ protocol are seen from Figure 23. Proceeding to the WEE property of the protocol, we start by stating the following.

Claim 2:
Under the conditions of Theorem 14, if a PPT witness extractor for the protocol `zkTElemRW`$_{t,s}$ *in Figure 23 extracts two different values of the factor a in the relation $Y = aX + \alpha H$ for different challenges $\boldsymbol{\delta}, \boldsymbol{\sigma}$ 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.

Let's construct a witness extractor for `zkTElemRW`$_{t,s}$. The extractor fetches the factor a in the relation

$$Y = aX + \hat{\alpha}H \, , \tag{101}$$

using another extractor which corresponds to a protocol that proves (101) in the last step of $\texttt{zkTElemRW}_{t,s}$. Inserting X, Y in (101) and moving aX to the left-hand side, the extractor obtains the equation

$$(Z - aP) + \langle \boldsymbol{\delta}, \mathbf{F} - a\mathbf{Q} \rangle - \langle \boldsymbol{\sigma}, a\mathbf{S} \rangle = \hat{\alpha} H \,. \tag{102}$$

By unwinding and running the $\texttt{zkTElemRW}_{t,s}$ protocol $(t + s)$ more times with different $\boldsymbol{\delta}, \boldsymbol{\sigma}$, the extractor gets, in sum, $(t + s + 1)$ equations of type (102), 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, as the opposite breaks the DL relation assumption by Claim 2. Writing down all these $(t + s + 1)$ equations in a matrix form as follows,

$$\mathfrak{a} \cdot \mathbf{B} = \hat{\boldsymbol{\alpha}} H, \tag{103}$$

where

$$\mathfrak{a} = \begin{bmatrix} 1 & \delta_{0,0} & \dots & \delta_{(t-1),0} & \sigma_{0,0} & \dots & \sigma_{(s-1),0} \\ 1 & \delta_{0,1} & \dots & \delta_{(t-1),1} & \sigma_{0,1} & \dots & \sigma_{(s-1),1} \\ 1 & \delta_{0,2} & \dots & \delta_{(t-1),2} & \sigma_{0,2} & \dots & \sigma_{(s-1),2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \delta_{0,(t+s)} & \dots & \delta_{(t-1),(t+s)} & \sigma_{0,(t+s)} & \dots & \sigma_{(s-1),(t+s)} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} Z - aP \\ F_0 - aQ_0 \\ \vdots \\ F_{t-1} - aQ_{t-1} \\ -aS_0 \\ \vdots \\ -aS_{s-1} \end{bmatrix}, \quad \hat{\boldsymbol{\alpha}} = \begin{bmatrix} \hat{\alpha}_0 \\ \hat{\alpha}_1 \\ \hat{\alpha}_2 \\ \vdots \\ \hat{\alpha}_{t+s} \end{bmatrix}, \tag{104}$$

and solving it for $\mathbf{B}$, 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 $\det(\mathfrak{a}) \neq 0$,

$$\mathbf{B} = \mathfrak{a}^{-1} \cdot \hat{\boldsymbol{\alpha}} H, \tag{105}$$

the extractor expresses each element of $\mathbf{B}$ as H multiplied by a corresponding scalar from the vector $\mathfrak{a}^{-1} \cdot \hat{\boldsymbol{\alpha}}$.

If witness a in the relation's (55) which was fed at $\mathcal{P}$'s private input is not equal to the witness a for the relation (101), which is found by the extractor and is used in the definition of $\mathbf{B}$ in (104), then it is possible to break the DL relation assumption. For this case, a breaker alghorithm honestly run the $\texttt{zkTElemRW}_{t,s}$ knowing $a, \alpha, \boldsymbol{\beta}, \boldsymbol{\gamma}$ in (55), and then extracts a in (101). Then, the breaker takes the equality for the first element of $\mathbf{B}$ in (105) and the equality for Z in (55). Eliminating Z from the both, keeping in mind 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, the witness a found by the extractor is the sought witness for (55).

Having the witness for (55) found, drawing the $(t + s + 1)$ blinding factors $\alpha, \boldsymbol{\beta}, \boldsymbol{\gamma}$ together into a vector, the extractor calculates them from (105), (104), (55) 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 $a, \alpha, \boldsymbol{\beta}, \boldsymbol{\gamma}$ for (55) and, thus, Theorem 14 is proven.

Q PROOF OF CLAIM ABOUT THE SAME FACTOR

Proof: [Claim 2] The proof is going to be a bit non-trivial, so let's first understand how the witness a extracted for the relation (101) in the last step of the protocol $\texttt{zkTElemRW}_{t,s}$ in Figure 23 depends on the challanges.

For convenience, we rewrite the relation (101) in matrix form using the formulas (50), (51), (52), (53), (54), taking $\boldsymbol{\xi}$ as a row vector and $\mathbf{T}$, $\mathbf{D}$ as column vectors, as

$$\boldsymbol{\xi} \cdot \mathbf{D} = a\boldsymbol{\xi} \cdot \mathbf{T} + \hat{\alpha} H \,. \tag{106}$$

We count that the extractor has already performed $(t + s + 1)$ rewindings and has $(t + s + 1)$ instances of the relation

(106) for $(t+s+1)$ instances of the challange vector $\boldsymbol{\xi}$. We write these $(t+s+1)$ instances of $\boldsymbol{\xi}$ as matrix

$$\mathfrak{a} = \begin{bmatrix} \boldsymbol{\xi}_0 \\ \boldsymbol{\xi}_1 \\ \boldsymbol{\xi}_2 \\ \vdots \\ \boldsymbol{\xi}_{(t+s)} \end{bmatrix} = \begin{bmatrix} 1 & \delta_{0,0} & \dots & \delta_{(t-1),0} & \sigma_{0,0} & \dots & \sigma_{(s-1),0} \\ 1 & \delta_{0,1} & \dots & \delta_{(t-1),1} & \sigma_{0,1} & \dots & \sigma_{(s-1),1} \\ 1 & \delta_{0,2} & \dots & \delta_{(t-1),2} & \sigma_{0,2} & \dots & \sigma_{(s-1),2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \delta_{0,(t+s)} & \dots & \delta_{(t-1),(t+s)} & \sigma_{0,(t+s)} & \dots & \sigma_{(s-1),(t+s)} \end{bmatrix}. \tag{107}$$

Since $\mathfrak{a}$ is a random matrix, with overwhelming probability there holds $\det(\mathfrak{a}) \neq 0$ and, thus, $\mathfrak{a}$ is a basis in the challenge space. Also, the extractor maps the corresponding $(t+s+1)$ witness pairs extracted in the last step of the protocol into two vectors as

$$\mathbf{A} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{t+s} \end{bmatrix}, \quad \hat{\alpha} = \begin{bmatrix} \hat{\alpha}_0 \\ \hat{\alpha}_1 \\ \hat{\alpha}_2 \\ \vdots \\ \hat{\alpha}_{t+s} \end{bmatrix}, \tag{108}$$

and has $(t+s+1)$ instances of the equality (106) for them written in the matrix form as

$$\mathfrak{a} \cdot \mathbf{D} = \begin{bmatrix} a_0 & & 0 \\ & \ddots & \\ 0 & & a_{t+s} \end{bmatrix} \cdot \mathfrak{a} \cdot \mathbf{T} + \hat{\alpha} H\,. \tag{109}$$

Let the extractor rewinds one more time and obtains $(a, \hat{\alpha})$ for a new challenge vector $\boldsymbol{\xi}$. The matrix $\mathfrak{a}$ is a basis in the space of $(t+s+1)$-dimensional scalar vectors, so $\boldsymbol{\xi}$ decomposes by it. Denote the corresponding row vector of weights as $\mathbf{B}$ such that

$$\boldsymbol{\xi} = \mathbf{B} \cdot \mathfrak{a}\,. \tag{110}$$

Next, multiplying the decomposition (110) by $\mathbf{D}$ and unfolding both sides of it using the formulas (106) and (109), respectively, the extractor obtains the following equality

$$\left(a\boldsymbol{\xi} - \mathbf{B} \cdot \begin{bmatrix} a_0 & & 0 \\ & \ddots & \\ 0 & & a_{t+s} \end{bmatrix} \cdot \mathfrak{a} \right) \cdot \mathbf{T} = (\mathbf{B} \cdot \hat{\alpha} - \hat{\alpha})\, H\,. \tag{111}$$

Recalling $\mathbf{T}$, by definition (50), is a vector of $\{P\} \cup \mathbf{Q} \cup \mathbf{S}$, the equality (111) represents 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 (111) is nonzero, the extractor has $P = \text{lin}(\text{nz}(\mathbf{Q}) \cup \text{nz}(\mathbf{S}) \cup \{H\})$, which contradicts to the premise of the Theorem 14. Namely, if the weight of P in (111) 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.

We have come to the conclusion that the weight of P in (111) must be zero. The extractor calculates it using (107), (54), (108) as

$$0 = a - \mathbf{B} \cdot \mathbf{A},$$

obtaining this way the sought transformation rule for the witness a depending on the challenge vector ξ

$$a = \mathbf{B} \cdot \mathbf{A}\,, \quad \text{where } \mathbf{B} = \xi \cdot \mathfrak{a}^{-1}. \tag{112}$$

Note, the rule (112) requires $\mathbf{B}$ to meet the condition $\left\langle \mathbf{B}, \{1\}^{t+s+1} \right\rangle = 1$, which keeps 1 at the first position in ξ.

Now, let's suppose the vector $\mathbf{A}$ contains at least two different scalars. In this case the extractor is able to build another basis $\mathfrak{a}'$ in the challenge space, instead of $\mathfrak{a}$, such that the corresponding witness vector $\mathbf{A}'$ is one-hot. Here is how the extractor builds $\mathfrak{a}'$. For the first, it defines helper matrices $\mathfrak{s}$ and $\mathfrak{e}$ as follows

$$\mathfrak{s} = \begin{bmatrix} 1 & a_0 & \tau_{0,0} & \dots & \tau_{0,(t+s-2)} \\ 1 & a_1 & \tau_{1,0} & \dots & \tau_{1,(t+s-2)} \\ 1 & a_2 & \tau_{2,0} & \dots & \tau_{2,(t+s-2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & a_{t+s} & \tau_{(t+s),0} & \dots & \tau_{(t+s),(t+s-2)} \end{bmatrix}, \quad \mathfrak{e} = \begin{bmatrix} 1 & 1 & \epsilon_{0,0} & \dots & \epsilon_{0,(t+s-2)} \\ 1 & 0 & \epsilon_{1,0} & \dots & \epsilon_{1,(t+s-2)} \\ 1 & 0 & \epsilon_{2,0} & \dots & \epsilon_{2,(t+s-2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & \epsilon_{(t+s),0} & \dots & \epsilon_{(t+s),(t+s-2)} \end{bmatrix}, \tag{113}$$

where all the scalars $\tau_{i,j}, \epsilon_{i,j},\ i \in [0, \dots, t+s], j \in [0, \dots, t+s-2]$ are uniformly random.

Namely, both $\mathfrak{s}$ and $\mathfrak{e}$ have the first columns of ones, the second colum of $\mathfrak{s}$ is $\mathbf{A}$, whereas the second colum of $\mathfrak{e}$ is $\mathbf{A}'$, i.e., one-hot

$$\mathbf{A}' = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} . \tag{114}$$

All the rest items in both matrices are picked uniformly at random. Apparently, $\det(\mathfrak{e}) \neq 0$ and, recalling there are at least two different items in $\mathbf{A}$, $\det(\mathfrak{s}) \neq 0$.

Next, the extractor defines matrix $\mathfrak{b}$ as

$$\mathfrak{b} = \mathfrak{e} \cdot \mathfrak{s}^{-1} , \tag{115}$$

thus having for it the equality

$$\mathfrak{b} \cdot \mathfrak{s} = \mathfrak{e} . \tag{116}$$

Finally, the extractor defines $\mathfrak{a}'$ as

$$\mathfrak{a}' = \mathfrak{b} \cdot \mathfrak{a} . \tag{117}$$

Since $\det(\mathfrak{b}) \neq 0$, which is seen from its definition (115), $\mathfrak{a}'$ is a new basis of the challenge space, its rows are the new basis vectors. Due to the equality (116), each new basis vector has 1 at the first position. Moreover, due to (116), (112) the corresponding values of the witness a are equal to 0 for all the new basis vectors, except for the first one, for which the witness a is 1.

In addition to this, the extractor builds one more basis $\mathfrak{a}''$ the following way

$$\mathfrak{a}'' = \frac{1}{2} \begin{bmatrix} 2 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ 1 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \dots & 1 \end{bmatrix} \cdot \mathfrak{a}' = \frac{1}{2}\mathfrak{g}' + \frac{1}{2}\mathfrak{a}' , \text{ where } \mathfrak{g}' = \begin{bmatrix} \xi'_0 \\ \xi'_0 \\ \xi'_0 \\ \vdots \\ \xi'_0 \end{bmatrix} . \tag{118}$$

Namely, in $\mathfrak{a}''$ it combines each vector in $\mathfrak{a}'$ having witness 0 with the first vector of $\mathfrak{a}'$, which has witness 1. By the rule (112), witnesses for the vectors in $\mathfrak{a}''$ are

$$\mathbf{A}'' = \begin{bmatrix} 1 \\ 1/2 \\ 1/2 \\ \vdots \\ 1/2 \end{bmatrix} . \tag{119}$$

Making $2(t+s)+1$ calls to witness extractor of subprotocol in the last step of $\mathtt{zkTElemRW}_{t,s}$, which is complete and WEE by the premise, the extractor complements the values of A' and A'' with values of $\hat{\alpha}'$ and $\hat{\alpha}''$ for all the vectors in $\mathfrak{a}'$ and $\mathfrak{a}''$.

As a result, denoting as $\bar{\mathfrak{g}}'$ the $(t+s) \times (t+s+1)$ matrix obtained from $\mathfrak{g}'$ by removing the first row in it; as $\hat{\alpha}'_0$ the common first scalar in $\hat{\alpha}'$ and $\hat{\alpha}''$; as $\bar{\mathfrak{a}}', \bar{\mathfrak{a}}''$ the two $(t+s) \times (t+s+1)$ matrices obtained from $\mathfrak{a}', \mathfrak{a}''$ by removing the first row in them; as $\bar{\hat{\alpha}}', \bar{\hat{\alpha}}''$ the vectors obtained from $\hat{\alpha}', \hat{\alpha}''$ by removing the first scalar in them, according to (106), (109) the extractor has the following equalities

$$\bar{\mathfrak{g}}' \cdot \mathbf{D} = \bar{\mathfrak{g}}' \cdot \mathbf{T} + \hat{\alpha}'_0 \mathbf{1}^{t+s} H , \tag{120}$$

$$\bar{\mathfrak{a}}' \cdot \mathbf{D} = \bar{\hat{\alpha}}' H , \tag{121}$$

$$\bar{\mathfrak{a}}'' \cdot \mathbf{D} = \frac{1}{2}\bar{\mathfrak{a}}'' \cdot \mathbf{T} + \bar{\hat{\alpha}}'' H . \tag{122}$$

By substituting the definition of $\mathfrak{a}''$ (118) into the equality (122) the extractor gets

$$(\bar{\mathfrak{g}}' + \bar{\mathfrak{a}}') \cdot \mathbf{D} = \frac{1}{2}(\bar{\mathfrak{g}}' + \bar{\mathfrak{a}}') \cdot \mathbf{T} + 2\bar{\hat{\alpha}}'' H . \tag{123}$$

Eliminating $\mathbf{D}$ from (123) using (120) and (121) it gets

$$(\bar{\mathfrak{a}}' - \bar{\mathfrak{g}}') \cdot \mathbf{T} = 2\left(\hat{\alpha}'_0 \mathbf{1}^{t+s} + \bar{\hat{\alpha}}' - 2\bar{\hat{\alpha}}''\right) H . \tag{124}$$

The right-hand side of the equality (124) is a column vector of known scalars, denote it as $\boldsymbol{\beta}$, multiplied by H. The left-hand side of the equality is the $(t+s) \times (t+s+1)$ matrix $(\bar{\mathfrak{a}}' - \bar{\mathfrak{g}}')$ multiplied by the column vector $\mathbf{T}$, which is a column made from (50).

Since $(\bar{\mathfrak{a}}' - \bar{\mathfrak{g}}')$ is made from $\mathfrak{a}'$, which has $\mathrm{rank}(\mathfrak{a}') = (t+s+1)$, by subtracting its first row from all their other rows with subsequent removing the first one, we have $\mathrm{rank}(\bar{\mathfrak{a}}' - \bar{\mathfrak{g}}') = (t+s)$. Recalling the first columns of all the three bases $\mathfrak{a}$, $\mathfrak{a}'$, $\mathfrak{a}''$ in the challenge space are by-design equal to $\mathbf{1}^{t+s+1}$, the first column of $(\bar{\mathfrak{a}}' - \bar{\mathfrak{g}}')$ is $\mathbf{0}^{t+s}$. Thus, the extractor removes the first column from $(\bar{\mathfrak{a}}' - \bar{\mathfrak{g}}')$, denoting the resulting $(t+s) \times (t+s)$ matrix as $\mathfrak{m}$, and has $\mathrm{rank}(\mathfrak{m}) = (t+s)$, i.e., $\det(\mathfrak{m}) \neq 0$. This way the equality (124) rewrites as

$$\mathfrak{m} \cdot \bar{\mathbf{T}} = \boldsymbol{\beta} H\,, \text{ where } \bar{\mathbf{T}} = \begin{bmatrix} Q_0 \\ \vdots \\ Q_{t-1} \\ S_0 \\ \vdots \\ S_{s-1} \end{bmatrix}, \tag{125}$$

and, thus, the extractor has each element of $\bar{\mathbf{T}}$ expressed through H

$$\bar{\mathbf{T}} = \mathfrak{m}^{-1} \cdot \boldsymbol{\beta} H\,. \tag{126}$$

According to the Theorem 14 premise, there is $i \in [0 \ldots t-1]$ such that $Q_i \neq 0$, and also there holds $H \mathrel{!}= \mathrm{lin}(\mathrm{nz}(\mathbf{Q}) \cup \{P\})$, however by (126) the extractor knows scalar $d = (\mathfrak{m}^{-1} \cdot \boldsymbol{\beta})_{[i]}$ such that $Q_i = dH$ and, hence, the extractor breaks the DL relation assumption, in the case when $\mathbf{A}$ contains at least two different scalars. The claim is proven.

R RANDOMLY WEIGHTED SUMS IMPLY THE SYSTEM IN MULTRATUG

When moving from the equality (44) to the system (45) in EFLRSLWB we implicitly use Theorem 3. More details about this are proveded in the proof of Theorem 13, particularly in Appendix N, where the equality (44) corresponds to the equality (86).

However, in Multratug, verifier has the equality (61) instead of (44). Transition from (61) to the system (62) 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 (61) to (62) becomes easy, details are in the proof of the following claim.

Claim 3:
If the Multratug protocol in Figure 24 completes successfully, then verifier is convinced that the equality (61) implies the system (62) 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 equations contain the elements from the Multratug scheme in Figure 24, whereas the left-hand sides contain ones from the protocol of Theorem 14 in Figure 23. By the formulas (50) and (51), respectively, the tuples become

$$\mathbf{T} = (\,\hat{U}_k,\ P_{s_k},\ \hat{I}_k,\quad A_k^{\mathbf{tmp}} - A_{s_k},\ U_{s_k} - U_k^{\mathbf{tmp}}\,)\,, \tag{127}$$

$$\mathbf{D} = (\,J_k,\ G,\ \ U_k^{\mathbf{tmp}},\ \ 0,\qquad\qquad 0\,)\,. \tag{128}$$

Also, in accordance to Figure 24, the random scalar vector $\boldsymbol{\xi}$ in the formula (54) becomes

$$\boldsymbol{\xi} = [1, \chi^{-1}, \chi^{-1}\theta, \chi^{-1}\omega, \chi^{-1}\zeta]\,. \tag{129}$$

By Theorem 12, due to the $\texttt{zkLin22Choice}_{l,n,l}$ call in Figure 24, verifier is convinced that prover knows p_k, v_k such that there holds the equality, for each $k \in [0 \ldots l-1]$, to the accuracy of H component

$$G + \theta U_k^{\mathbf{tmp}} + \chi J_k = p_k(P_{s_k} - K + \zeta U_{s_k} - \omega A_{s_k}) + v_k(K + \omega A_k^{\mathbf{tmp}} - \zeta U_k^{\mathbf{tmp}} + \theta \hat{I}_k + \chi \hat{U}_k)\,, \tag{130}$$

which becomes the equality (61) after elimianing the hash to group K. The elimination is performed the same way as for (83) in Appendix N.

As a result, for X, Y calculated by the formulas (52), (53) using (127), (128), (129), the equality (61) rewrites as

$$\chi Y = \chi p_k X\,. \tag{131}$$

Everything to the accuracy of H. Since χ is nonzero and known to both of the prover and verifier prior to applying the Theorem 12 protocol, both sides of (131) can be divided by it, and (131) rewrites as

$$Y = p_k X\,, \tag{132}$$

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, HVZK, and WEE protocol of Theorem 12 (Lin2-2Choice lemma), which proves the relation (30).

Also, according to Figure 24 the following holds. The element $\hat{U}_k$ in the tuple $\mathbf{T}$ (127) is nonzero and is orthogonal to all the other nonzero elements of $\mathbf{T}$ and to the blinding generator H, i.e., $\hat{U}_k \mathrel{!=} \mathrm{lin}(\,\mathrm{nz}(P_{s_k}, \hat{I}_k), \mathrm{nz}(A_k^{\mathbf{tmp}} - A_{s_k}, U_{s_k} - U_k^{\mathbf{tmp}}), H\,)$. The nonzero element H is ortogonal to all nonzero elements of the set $\{P_{s_k}, \hat{I}_k, \hat{U}_k\}$, i.e., $H \mathrel{!=} \mathrm{lin}(\,\mathrm{nz}(P_{s_k}, \hat{I}_k), \hat{U}_k\,)$. The element P_{s_k} is guaranteed nonzero.

Thus, all steps of the $\texttt{zkTElemRW}_{2,2}$ protocol in Figure 23 have been performed and the premise of Theorem 14 is met. Therefore, by Theorem 14 verifier is convinced that the relation (55) holds, and, hence, the tuples (127), (128) are elementwise proportional to each other, to the acccuracy of H, which is equivalent to the system (62).

S SIGNATURE MULTRATUG

Proof: [Theorem 15] According to Figure 24, as the new vectors $\mathbf{U}^{\mathbf{tmp}}, \hat{\mathbf{I}}$ are defined by the formulas (57), (56), all proofs of Theorem 13 for the EFLRSLWB scheme in Figure 21 transfer to the Multratug scheme in Figure 24.

In fact, $\mathbf{U}^{\mathbf{tmp}}$ is indistinguishable from the independent uniform randomness due to the blinding components $\hat{\mu}H$ in it (57), hence $\mathbf{U}^{\mathbf{tmp}}$ does not change anything. The same is for $\hat{\mathbf{I}}$ (56), which is indistinguishable from the independent uniform randomness and from the former $\mathbf{I}$ (34). This is proved in [20], and also can be proved using the method of [8]. Also, the vector of hash images $\hat{\mathbf{I}}$ gets $\mathbf{U}^{\mathbf{tmp}}$ in its pre-image, however this does not change anything in it, only depricates linear dependency of the vectors. The same is for the blinding generator H, which gets the new vectors into its pre-image.

With the former $\mathbf{I}$, EFLRSLWB has (44) and gets (45) from it. With the new $\mathbf{U}^{\mathbf{tmp}}, \hat{\mathbf{I}}$, Multratug has (61) instead of (44), and gets (62) from it by Claim 3 in Appendix R, instead of (45). As (45) is a subset of (62), 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.

Thus, 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 21 transfer to the linkable threshold ring signature Multratug in Figure 24.

T VECTOR SCHNORR ARGUMENT

Proof: [Theorem 16] Design of the protocol in Figure 25 is clearly Schnorr-like. Hence, its completeness, HVZK, and WEE 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.

Also, additional details can be found in [2], where the HVZK and WEE properties are proved for a similar protocol.

U NON-ZK LOG-SIZE VECTOR COMMITMENT ARGUMENT

Proof: [Theorem 17] For $n > 4$, the protocol in Figure 26 comprises the reductions used in the inner product argument [5] with $\mathbf{b} = \{0\}^n$ and, hence, it is complete and has WEE 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 WEE.

Also, in [2] the HVZK and WEE properties are proved for a similar protocol.

V OPTIMIZED ZK LOG-SIZE VECTOR COMMITMENT ARGUMENT

Proof: [Theorem 18] Completeness is by-design. The $\texttt{argVC}_{n+1}$ call in the last step of $\texttt{zkVC}_n^{\mathbf{opt}}$ has WEE by Theorem 17. Having extracted the witness $\boldsymbol{\tau}$ from it, the protocol turns out to be $\texttt{zkNElemComm}_{n+1}$, which has WEE by Theorem 16. Thus, $\texttt{zkVC}_n^{\mathbf{opt}}$ has WEE. Even with the opened $\boldsymbol{\tau}$ the protocol remains HVZK by Theorem 16, so partially hiding it inside $\texttt{argVC}_{n+1}$ doesn't make $\texttt{zkVC}_n^{\mathbf{opt}}$ less zero-knowledge. Thus, $\texttt{zkVC}_n^{\mathbf{opt}}$ is HVZK.

Also, in [2] such a composition is proved to be having HVZK and WEE properties.

W NOTES ABOUT DUALRING-EC

The DualRing-EC signature, according to its security model in [24], 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 doesn't depricate malformed keys 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 [24] 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.

X LOW ANONYMITY OF U/X

Let's determine 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. It 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 simple and therefore very possible case of non-uniform distribution of x's. Let the distribution have a high probability for pairs of private keys (x_1, x_2) such that $x_2 = 2x_1$. Consequently, two signatures signed with keys from the same pair will be linked together by 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 tightly coupled keys. This is, for example, the case in [22], where private keys behind the public keys in the rings have the form $x = b + r$ with hidden b, and 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 [13, 23, 17], 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.