Succinct Line-Point Zero-Knowledge Arguments from Homomorphic Secret Sharing

Zhe Li[1], Chaoping Xing[2], Yizhou Yao[2], Chen Yuan[2], Mengmeng Zhou[3]

[1] Institute of Mathematics and Interdisciplinary Sciences,
Xidian University, Xi'an, China
lizh0048@e.ntu.edu.sg
[2] School of Computer Science,
Shanghai Jiao Tong University, Shanghai, China
{xingcp,yaoyizhou0620,chen_yuan}@sjtu.edu.cn
[3] Beijing Academy of Blockchain and Edge Computing, Beijing, China
zhoumm@baec.org.cn

Abstract. Dittmer, Ishai and Ostrovsky (ITC'21) proposed *line-point zero-knowledge proof* (LPZK), a simple "commit-and-prove" system, motivated by practical protocols for compressing correlated pseudorandomness used in secure multiparty computation (MPC). Typically, LPZK admits concretely efficient ZK protocols with a streaming, linear time prover, *but a linear size proof*. A natural question raised in the context is how far can we go in minimizing the proof size, while maintaining the prover efficiency. Though a recent work by Lin, Xing and Yao (ASIACRYPT'24) gives an interactive LPZK with a sublinear proof size $O(n + d^2 \log |\mathcal{C}|)$, it is still far from being *succinct*, where $n, d, |\mathcal{C}|$ are referred to as input size, circuit depth, and circuit size, respectively.

In this work, we beat the proof size barrier and propose *succinct LPZK arguments*, by distilling techniques from orthogonal studies on homomorphic secret sharing and succinct garbling. Specifically, under variants of group/lattice-based assumptions, we show the followings:

i) There exist succinct LPZK arguments with common reference string (CRS) size $O(n^{2/3})$, proof size $O(n^{2/3})$, prover time $O(n^{4/3} + |\mathcal{C}|)$, verification time $O(n + |\mathcal{C}|)$, and negligible soundness error, where both the prover and the verifier executions and be run in a streaming fashion.

ii) The above proof size can be further optimized to $O(1)$, at the cost of a larger CRS size $O(n)$, and prover time increased to $O(n^2 + |\mathcal{C}|)$.

In general, our succinct LPZK arguments pave a new way for building designated-verifier zero-knowledge succinct non-interactive arguments of knowledge (dv-zkSNARKs), and new interesting features (e.g., streaming, constant sized proof with CRS size not proportional to the circuit size) are obtained for the first time along the way.

Table of Contents

1 Introduction . . . 2
1.1 Our Contributions . . . 3
2 Our Techniques . . . 6
3 Preliminaries . . . 11
3.1 Cryptographic Assumptions . . . 11
3.2 Succinct Non-interactive Arguments of Knowledge . . . 14
4 Succinct Line-Point Zero-Knowledge Arguments . . . 16
4.1 Succinct LPZK Arguments from Paillier Groups . . . 16
4.2 Succinct LPZK Arguments from Lattice . . . 21
5 Extractable Succinct Half-Chosen VOLE . . . 23
5.1 Succinct Half-Chosen VOLE from [ARS24] . . . 24
5.2 Succinct Half-Chosen VOLE from [AMR25] . . . 28
A Related Work . . . 35
B More preliminaries . . . 36
B.1 Assumptions in prime-order groups . . . 37
B.2 Damgård-Jurik Cryptosystem [DJ01] . . . 37
C More Constructions . . . 38
C.1 Succinct LPZK Arguments from Prime-order Groups . . . 38
C.2 Succinct LPZK from Damgård-Jurik . . . 40
D Deferred Proof . . . 41
D.1 Security Proof for Π_{zkPai} . . . 41
D.2 Proof of Π_{hcVOLE} . . . 44

1 Introduction

Zero-knowledge (ZK) proofs [GMR89] and secure multiparty computation (MPC) [Yao86], the twin gems on the crown of theoretical computer science, have been under extensive and intersected researches since their births in the 1980s. On the one hand, the well-known GMW compiler [GMW87] indicates that ZK proofs allow to enhance the security of MPC protocols, and recent works [BBC+19, LXYY25] show that such overhead can be almost free. On the other hand, there is a variety of paradigms distilled from the MPC literature, which admit practically efficient ZK protocols. This paper explores in the second direction, where existing, representative examples include:

The MPC-in-the-head framework [IKOS07], where the prover emulates in his head MPC evaluations with imaginary parties and proves to the verifier that the statement is honestly verified by the virtual parties; and the ZK-GC framework [JKO13], where the verifier garbles the verification circuit, and the prover evaluates the garbled circuit (GC) with his witness; and the VOLE-ZK framework [BDSW23], where the two parties jointly evaluate the verification

circuit via a message authenticated code based on *vector oblivious linear evaluation* (VOLE).

Typically, ZK protocols from the above paradigms feature a light-weight, linear time and streaming prover but usually have a linear proof size, inherited from the "gate-by-gate" nature of MPC protocols. Here, streaming means that the proof can be generated "on the fly", so that a memory footprint comparable to that of evaluating the circuit in the clear suffices. While succinct proof sizes are possible to achieve in the context [AHIV17, DH24] — by leveraging powerful tools from probabilistic checkable proofs [BFLS91] and interactive proofs [LFKN92, GKR08] — the prover time might no longer remain linear and moreover the memory footprint would be substantially increased. Hence, it is well-motivated to seek new techniques for succinctness that would not compromise prover efficiency.

A simple intuition is that light-weight, sublinear MPC would inspire a nice solution. In fact, there do exist MPC protocols of sublinear communication without relying on fully homomorphic encryption (FHE) or indistinguishable obfuscation (iO) [BCM23, DIJL23, CK24]. And moreover, a concurrent line of works [MORS24, MORS25, ILL24, ILL25] establishes the framework of *succinct* garbling schemes, which are *practically feasible*. The core technique shared by these works appears to be *homomorphic secret sharing* (HSS), a secret-sharing analogue of fully homomorphic encryption, which allows the parties to locally compute a function f on the secret-sharings of x and obtain the shares of the output $f(x)$. Bearing these in mind, we raise a natural question:

Can we make use of HSS techniques to build efficient succinct zero-knowledge protocols?

1.1 Our Contributions

We answer the above question in the affirmative by putting forth and studying a new framework that we dub *succinct line-point zero-knowledge* (succinct LPZK), which shares many high-level similarities with VOLE-ZK. Essentially, we show a modular approach that compiles VOLE-ZK into succinct arguments via HSS techniques. We demonstrate succinct LPZK constructions from various existing assumptions, and in fact they are *designated-verifier zero-knowledge succinct non-interactive arguments of knowledge*, abbreviated as dv-zkSNARK in the literature[4].

The key features of our succinct LPZK arguments are that i) linear (in the circuit size) prover computation, ii) streaming proving & verifying procedures, iii) a sublinear/constant proof size with a sublinear/linear CRS (in the witness size, *independent* of the circuit size), iv) a "public-key infrastructure"-style trusted setup, v) native to $\mathbb{Z}$-arithmetic. These features are new to the big family of ZK protocols, and we give detailed comparisons to existing frameworks in Table 1 as well as a comparison of related works in Appendix A.

[4] Throughout the paper, succinctness merely means succinct communication complexity, rather than succinct verification time that some SNARKs may additionally offer.

Work	Framework	Prover time	Proof size	Verifier time	**S**	Setup	Assumption
Groth16[Gro16]	Linear PCP	$O(\|\mathcal{C}\| \log \|\mathcal{C}\|)$	$O(1)$	$O(1)$	NO	CD-Trusted	Pairing
Libra[XZZ+19]	IP+PCS	$O(\|\mathcal{C}\|)$	$O(d \log \|\mathcal{C}\|)$	$O(\|\mathcal{C}\|)$	NO	Trusted	RO+Pairing
LaBRADOR[BS23]	IP	$O(\|\mathcal{C}\|)$	$O(\log \|\mathcal{C}\|)$	$O(\|\mathcal{C}\|)$	NO	Transp.	RO+SIS
Brakedown[GLS+23]	IOP+PCS	$O(\|\mathcal{C}\|)$	$O(\sqrt{\|\mathcal{C}\|})$	$O(\sqrt{\|\mathcal{C}\|})$	NO	CD-Transp.	RO
KKW18[KKW18]	MPCitH	$O(\|\mathcal{C}\|)$	$O(\|\mathcal{C}\|)$	$O(\|\mathcal{C}\|)$	YES	Transp.	RO
VOLEitH[BBdSG+23]	VOLE-ZK	$O(\|\mathcal{C}\|)$	$O(\|\mathcal{C}\|)$	$O(\|\mathcal{C}\|)$	YES	Transp.	RO
Rinocchio[GNS23]	Linear PCP	$O(\|\mathcal{C}\| \log \|\mathcal{C}\|)$	$O(1)$	$O(1)$	NO	CD-Trusted	Joye-Libert
ILPZK[LXY24]	VOLE-ZK	$O(\|\mathcal{C}\|)$	$O(n + d \log \|\mathcal{C}\|)$	$O(\|\mathcal{C}\|)$	NO	Trusted	RO+LPN
This work	Succinct LPZK	$O(n^{4/3} + \|\mathcal{C}\|)$	$O(n^{2/3})$	$O(\|\mathcal{C}\|)$	YES	Trusted	RO+DCR/LWE
This work	Succinct LPZK	$O(n^2 + \|\mathcal{C}\|)$	$O(1)$	$O(\|\mathcal{C}\|)$	YES	Trusted	RO+DCR/LWE

Table 1. Comparisons of practically efficient NIZK of knowledge for proving circuit satisfiability. Here, $d, n, |\mathcal{C}|$ are circuit depth, input size, and circuit size, respectively. The column **S** denotes *streamable*, and "CD" refers to circuit-dependent, and "Transp." stands for transparent. The proof sizes of Libra and ILPZK are for layered circuits. The works in the upper half are all publicly verifiable, while the rest are designated-verifier. We omit the input size when it is not a dominated term since $|\mathcal{C}| = \Omega(n)$.

In the setting of reusable dv-zkSNARK, both parties have access to a common reference string (CRS) and the verifier is additionally given a secret verification key. These are obtained through a trusted setup phase, and a key challenge is to make it reusable, i.e., the CRS as well as the verification key can be reused among multiple executions. More precisely, then soundness holds even against a malicious prover that learns the acceptance or rejection of prior proofs by the verifier. Hence, one can view the CRS as a public key and the verification key as a secret key of the verifier.

From VOLE-ZK to succinct LPZK. We start with the information-theoretic LPZK [DIO21] proof system, a simple kind of designated-verifier non-interactive ZK (dv-NIZK) in VOLE-ZK. In an LPZK proof, the prover encodes the witness $\mathbf{w}$ into two vectors $\mathbf{a}, \mathbf{b}$ that specify an affine line $\mathbf{v}(t) = \mathbf{a} \cdot t + \mathbf{b}$, then the verifier queries the line at a random point Δ to learn $\mathbf{v}(\Delta)$ (through a two-round VOLE protocol) and decides whether to accept or reject the proof. However, typically the resulting NIZK has linear proof size, because (i) the dimension of $\mathbf{v}(t)$ is at least linear in the witness size[5], (ii) standard VOLE protocols have inherent linear communication in the VOLE length.

Towards the goal of succinct proof length, we first design a new (computational) LPZK construction with the VOLE length meeting the lower bound $|\mathbf{w}|$, by techniques distilled from recent HSS-based garbling schemes [MORS24, MORS25, ILL24, ILL25]. Then we compile it into a succinct argument by utilizing a succinct form of bilinear HSS [ARS24]. Specifically, we employ the *half-chosen* VOLE protocols in [ARS24] to enable the verifier to learn $\mathbf{v}(\Delta)$ in sublinear communication (in the length of VOLE). It is worth noting that by half-chosen VOLE only $\mathbf{a}$ is determined by the prover, while $\mathbf{b}$ is random. However, this is

[5] This is inherent, since if it is not the case, the existence of an extractor that can extract the entire $\mathbf{w}$ would immediately violate information theory.

allowed in the context of VOLE-ZK, as implicitly claimed by [ARS24], because an honest prover in LPZK [DIO21] originally selects a random $\mathbf{b}$.

At a high-level, the computational LPZK construction in the first step is yielded by a "magic" procedure from HSS that allows to *jointly* encode the witness and compute the query result *without interaction*. That is, given input VOLE correlations $\mathbf{v}_0 = \mathbf{w} \cdot \Delta + \mathbf{b}_0$ and some auxiliary information, the two parties can recover an entire LPZK proof $\mathbf{v} = \mathbf{a} \cdot \Delta + \mathbf{b}$, allowing the verifier to check the validity of the witness. Moreover, our prover inherits the streaming fashion from plain LPZK in [DIO21]. Analogues to most protocols based on HSS techniques, we require the VOLE correlations as well as the verification circuit $\mathcal{C}$ are over $\mathbb{Z}$, and assume the computations in $\mathcal{C}$ are B-bounded[6], i.e., all the wire values are upper bounded by some $B = \mathsf{poly}(\lambda)$. Thanks to the rich study of HSS, the "magic" procedure can be instantiated with various cryptographic assumptions. Our results are summarized as follows.

Theorem 1.1 (Informal, Computational LPZK construction). *Let $\mathcal{C} : \mathbb{Z}^n \to \mathbb{Z}$ be a B-bounded verification circuit. Given some auxiliary information of size $O_\lambda(1)$, there exists an LPZK construction with VOLE length n and prover/verifier computation $O_\lambda(|\mathcal{C}|)$ under either (1) KDM security of Damgård-Jurik cryptosystem (Definition B.4), (2) circular power DDH in Paillier groups (Definition 3.4), (3) circular power DDH in prime-order groups (Definition B.3), (4) circular power (Ring)LWE assumption with a superpolynomial-modulus-to-noise ratio (Definition 3.7).*

Here, $O_\lambda(\cdot)$ hides polynomial factors in the security parameter λ. The above "circular" and "power" assumptions can be substituted by standard assumptions, at the cost of slightly increasing the size of auxiliary information to $O(d)$, where d is the depth of $\mathcal{C}$.

We then compile the computational LPZK construction in Theorem 1.1 into a succinct LPZK argument via half-chosen VOLE [ARS24]. However, there is one missing step to get an *argument of knowledge*: we must prove the knowledge-soundness property, which is not obvious due to succinct half-chosen VOLE protocols. To this end, we show that the construction of [ARS24] offers *partial extraction* and propose a general approach to *full extraction*. Taking assumptions over Paillier groups as concrete instantiations, we obtain the following result:

Theorem 1.2 (Informal, Succinct LPZK). *Let $\mathcal{C} : \mathbb{Z}^n \to \mathbb{Z}$ be a B-bounded verification circuit. There exists a dv-zkSNARK for proving the satisfiability of $\mathcal{C}$ in the CRS model and random oracle model with*

1. *CRS size $O_\lambda(n)$, proof size $O_\lambda(1)$, prover time $O_\lambda(n^2 + |\mathcal{C}|)$, verifier time $O_\lambda(n + |\mathcal{C}|)$, assuming n-ary power DDH (Definition B.1) and circular-power DDH (Definition 3.4) over Paillier groups.*
2. *CRS size $O_\lambda(n^{2/3})$, proof size $O_\lambda(n^{2/3})$, prover time $O_\lambda(n^{4/3} + |\mathcal{C}|)$ and verification time $O_\lambda(n + |\mathcal{C}|)$, assuming DCR (Definition 3.5) and circular-power DDH (Definition 3.4) over Paillier groups.*

[6] For example, boolean circuits can be easily converted into B-bounded circuits over $\mathbb{Z}$.

3. *CRS size* $O_\lambda(\mathsf{polylog}(n))$*, proof size* $O_\lambda(\mathsf{polylog}(n))$*, prover time* $O_\lambda(n \log n + |\mathcal{C}|)$*, verifier time* $O_\lambda(n+|\mathcal{C}|)$*, assuming LWE, circular-power LWE (Definition 3.7) and PCGs for random VOLE [BCG+19a].*

We remark that the above third construction is obtained by employing the LWE-based succinct oblivious tensor evaluation (OTE) protocol in [AMR25], which does not offer sender privacy. To remain zero-knowledge, we first use a pseudorandom correlation generator (PCG) [BCGI18, BCG+19b, SGRR19, BCG+19a] that generates random VOLE as masks, then the prover can invoke the OTE protocol [AMR25] with masked inputs. We leave directly adding sender privacy to [AMR25] without PCGs as future work.

2 Our Techniques

Recap: line-point zero-knowledge (LPZK) proofs and VOLE-ZK. We review the LPZK proof system proposed by Dittmer, Ishai and Ostrovsky [DIO21] and VOLE-ZK in more detail. Recall that in an LPZK proof, the prover first encodes the witness $\mathbf{w}$ into an affine line $\mathbf{v}(t) = \mathbf{a} \cdot t + \mathbf{b}$, then the verifier queries the line at a single random point $t := \Delta$ and decides whether to accept or reject. LPZK proofs are closely related to VOLE-ZK protocols because VOLE matches the query procedure: in VOLE, the sender (as the prover) holds M_x, x and the receiver (the verifier) holds K_x, Δ such that $K_x = x \cdot \Delta + M_x$, where Δ is fixed for all possible x. It is well-known that VOLE is also a linearly homomorphic *message authentication code* (MAC). That is, the sender and the receiver essentially hold an authenticated additive secret sharing of x with respect to the global key Δ, denoted by $[\![x]\!] := [\Delta \cdot x]$. This allows to understand VOLE-ZK protocols in the perspective of secure multiparty computation (MPC):

1. The prover (VOLE sender) and the verifier (VOLE receiver) first obtain authenticated sharings of the witness, i.e., $[\![\mathbf{w}]\!]$.
2. The two parties jointly evaluate the verification circuit $\mathcal{C}$ gate-by-gate on the input sharings $[\![\mathbf{w}]\!]$.
3. Finally, the verifier checks the validity of $[\![\mathcal{C}(\mathbf{w})]\!]$, i.e., $\mathcal{C}(\mathbf{w}) \stackrel{?}{=} 0$.

Conventional VOLE-ZK protocols feature an offline-online design and a gate-by-gate flavor, similar to modern MPC protocols. More specifically, first random VOLE correlations $[\![\mathbf{r}]\!]$ are efficiently prepared in the offline phase, then in the online phase, they are turned into VOLE correlations $[\![\mathbf{w}]\!] := [\![\mathbf{r}]\!] + (\mathbf{w} - \mathbf{r})$ by revealing $(\mathbf{w} - \mathbf{r})$ to the verifier. Moreover, as the prover holds the witness in the clear, he can compute all the intermediate wire values (i.e., the extended witness), which reduces the joint evaluation to step 1 and significantly simplifies the protocol. Due to the linearity of VOLE, now it suffices to check multiplication relations on the authenticated extended witness, which can be verified in $O(1)$ communications through a random linear combination check [YSWW21]. Though random VOLE can be instantiated with sublinear communication by pseudorandom correlation generators (PCG) [BCGI18, BCG+19b, YWL+20], revealing $(\mathbf{w} - \mathbf{r})$ incurs an inherent linear dependence on the number of gates.

Towards succinctness. Achieving succinctness, i.e., making the proof size sublinear in the circuit size as well as the witness size, has been a central goal in the context ever since the birth of VOLE-ZK. Although recent works [WYY+22, DH24, LXY24, BBM+24] have made significant progress to this end, they fail in getting succinctness without introducing additional cryptographic tools beyond random VOLE. To our best knowledge, the state-of-the-art [LXY24] achieves VOLE-ZK with VOLE length $O(|\mathbf{w}| + d^2 \log |\mathcal{C}|)$ for general circuits, from employing the well-known Sumcheck techniques [LFKN92, GKR08, XZZ+19, ZLW+21], where d is the depth of $\mathcal{C}$. However, this yields a ZK argument with proof size $O(|\mathbf{w}| + d^2 \log |\mathcal{C}|)$ via the random VOLE based instantiation, which is apparently not succinct.

Succinct Half-chosen VOLE to the rescue. To make the proof succinct, the idea is very straightforward: just give up the random VOLE in preprocessing model and resort to succinct VOLE protocols to directly encode the witness. Note that standard VOLE with both $\mathbf{x}, M_{\mathbf{x}}$ chosen by the sender has inherent $\Omega(N)$ communication [ARS24], where N is the VOLE length. The main observation here is that a *half-chosen* VOLE protocol with only $\mathbf{x}$ chosen by the sender and uniformly random $M_{\mathbf{x}}$ suffices for VOLE-ZK protocols. Since in typical LPZK proofs, the fact is that a bulk of $\mathbf{b}$ of the generated line $\mathbf{v}(t) = \mathbf{a} \cdot t + \mathbf{b}$ is uniformly random and independent of the witness $\mathbf{w}$[7], and moreover the transformation $[\![x]\!] \leftarrow [\![r]\!] + (x - r)$ from random $[\![r]\!]$,

$$\underbrace{K_r + \Delta \cdot (x - r)}_{K_x} = \Delta \cdot x + \underbrace{M_r}_{M_x},$$

essentially admits a non-succinct half-chosen VOLE protocol, where K_r, Δ, M_r, r are random values given by a random VOLE protocol.

We instead employ succinct half-chosen VOLE protocols [ARS24, AMR25]. We remark that using the construction in [ARS24] with both sender privacy and receiver privacy, the parties can directly obtain $[\![x]\!]$. While for the construction in [AMR25] with only receiver privacy, we instead let the parties obtain $[\![(x - r)]\!]$, which together with random $[\![r]\!]$ produced by a PCG gives the desired $[\![x]\!]$. However, these constructions are only semi-honestly secure, while in the context of dv-zkSNARK, the prover (i.e., the sender) is malicious and we must prove the extraction property, i.e., the existence of a extractor that extracts the sender's input witness. By viewing half-chosen VOLE as a linearly-homomorphic commitment scheme, we complement it with a simple zero-knowledge opening procedure, which allows extraction.

Finally, we briefly discuss the compatibility of existing information-theoretic LPZK constructions and half-chosen VOLE. The succinct half-chosen VOLE protocol in [ARS24] has two simultaneous rounds, communication $O(N^{2/3})$ for

[7] The number of non-uniformly random positions (corresponding to proofs of the correctness of multiplication gates) can be compressed to $O(1)$ via a standard random linear combination technique, hence one can pay constant communications for turning half-chosen VOLE at these positions into standard VOLE.

length N, sender computation $O(N^{4/3})$ and receiver computation $O(N)$, where N is suggested to be a large, cubic integer. Hence, it does not perfectly fits the state-of-the-art interactive LPZK proof [LXY24], where the $O(d^2 \log |\mathcal{C}|)$ term in communication comes from $O(d \log |\mathcal{C}|)$ rounds, as the non-interactive property of half-chosen VOLE cannot be utilized. However, when applied to the round-optimal LPZK proof [DIO21], the prover's computational complexity inevitably becomes a major bottleneck: as the proof requires VOLE length $O(|\mathcal{C}|)$, prover's computation would be super-linear in $|\mathcal{C}|$, losing the typical advantage of fast proving of VOLE-ZK protocols.

Non-interactive circuit evaluation. Our goal is to develop a computational LPZK construction that minimizes *round complexity* and *proof size* simultaneously, so that it can be compiled into efficient succinct arguments via succinct half-chosen VOLE. At the heart of our framework is a method for evaluating multiplication gates *non-interactively*: that is, given $[\![x]\!], [\![y]\!]$ and some auxiliary information (included in a crs), the two parties can locally compute $[\![x \cdot y]\!]$ without any interaction. Recall that addition gates can be locally evaluated due to the linearity of $[\![\cdot]\!]$. Hence, it suffices to invoke half-chosen VOLE protocols to obtain $[\![\mathbf{w}]\!]$, then $[\![\mathcal{C}(\mathbf{w})]\!]$ for any circuit $\mathcal{C}$ can be locally evaluated. Moreover, as long as local evaluation takes linear time, prover's computational complexity remains linear in $|\mathcal{C}|$ (only super-linear in $|\mathbf{w}|$). When the circuit size is substantially larger than the witness size, this significantly reduces the costs compared to directly incorporating [DIO21] with half-chosen VOLE. But how can the parties jointly compute without interaction?

We make use of the powerful distributed discrete logarithm (DDLog) techniques from the homomorphic secret sharing (HSS) literature, greatly inspired by recent breakthroughs in succinct garbling [MORS24, MORS25, ILL24, ILL25]. Roughly speaking, DDLog allows to locally convert multiplicative sharings over the exponent to subtractive sharings. For instance, given g_p held by the prover, and g_v held by the verifier such that $g_v = g^{\Delta \cdot x} \cdot g_p$, the parties can compute

$$K_x \leftarrow \mathsf{DDLog}(g_v), \quad M_x \leftarrow \mathsf{DDLog}(g_p),$$

such that

$$K_x = \Delta \cdot x + M_x \mod N,$$

where g_v, g_p, g are some group elements and N is the order of g. We sketch the high-level idea below. For now, just pretend that g^{Δ} (known to the prover) were a "secure encryption" of Δ, we could see DDLog enabled the parties to obtain

$$[\Delta \cdot x] \leftarrow \mathsf{DDLog}(g^{[\Delta \cdot x]}) = \mathsf{DDLog}\big((g^{\Delta})^{[x]}\big)$$

without revealing Δ to the prover. To non-interactively evaluate multiplication gates, we crucially rely on the following observation: given $[\![x]\!], [\![y]\!]$ with $K_x = \Delta \cdot x + M_x$, and $K_y = \Delta \cdot y + M_y$, we have

$$\begin{aligned} & K_x \cdot K_y = \Delta^2 \cdot xy + \Delta \cdot (x \cdot M_y + y \cdot M_x) + M_x \cdot M_y \\ \iff & \Delta^{-1} \cdot (K_x \cdot K_y - M_x \cdot M_y) = \Delta \cdot xy + (x \cdot M_y + y \cdot M_x). \end{aligned} \tag{1}$$

Hence, given $g^{\Delta^{-1}}$ as a "secure encryption" of Δ^{-1}, the prover can compute

$$M_z \leftarrow \mathsf{DDLog}\big((g^{\Delta^{-1}})^{M_x M_y}\big) + (x \cdot M_y + y \cdot M_x),$$

while the verifier computes

$$K_z \leftarrow \mathsf{DDLog}\big((g^{\Delta^{-1}})^{K_x K_y}\big).$$

By the definition of DDLog and Eq.(1), it holds that $K_z = \Delta \cdot xy + M_z \mod N$. This completes the non-interactive multiplication procedure as Δ is fixed and $g^{\Delta^{-1}}$ can be reused across multiple gates.

Secure instantiations. In the above overview, we omit many details and simply assume g^{Δ} were a secure encryption of Δ, which unfortunately will never be the case. Thanks to the rich HSS literature, we are able to put forward several secure instantiations under various hardness assumptions.

KDM-security of Damgård-Jurik encryption. Let $N = p \cdot q$ be a secure RSA modulus and $\zeta \geq 1$. Damgård-Jurik (DJ) [DJ01] encryption is a generalization of Paillier encryption [Pai99], where a DJ ciphertext of message $x \in \mathbb{Z}_{N^\zeta}$ has the form $r^{N^\zeta}(1+N)^x$ with $r \xleftarrow{\$} \mathbb{Z}_{N^{\zeta+1}}$. By [OSY21, RS21], there exists an efficient DDLog algorithm with perfect correctness for the DJ group with respect to $g = (1+N)$. Let $\Delta := \varphi(N)$, where $\varphi(\cdot)$ denotes the Euler function. The verifier sends the encryption of Δ^{-1} to the prover as the auxiliary information, i.e., $\mathsf{Enc}(\Delta^{-1}) = r^{N^\zeta}(1+N)^{\Delta^{-1} \mod N^\zeta} \mod N^{\zeta+1}$. Let $[\![x]\!], [\![y]\!]$ with $K_x = \Delta \cdot x + M_x$, and $K_y = \Delta \cdot y + M_y$ over $\mathbb{Z}$ be the multiplication inputs. Similarly, the prover computes

$$M_z \leftarrow \mathsf{DDLog}\big((\mathsf{Enc}(\Delta^{-1}))^{M_x M_y}\big) + (x \cdot M_y + y \cdot M_x) \mod N^\zeta,$$

while the verifier computes

$$K_z \leftarrow \mathsf{DDLog}\big((\mathsf{Enc}(\Delta^{-1}))^{K_x K_y}\big) \mod N^\zeta.$$

By the correctness of DDLog, $K_z = \Delta \cdot z + M_z \mod N^\zeta$ holds and the security relies on the KDM-security of DJ encryption (Definition B.4). However, there is a critical issue: we need $K_z = \Delta \cdot z + M_z$ holds over $\mathbb{Z}$ rather than only $\mathbb{Z}_{N^\zeta}$ to support subsequent computations. To this end, we follow the strategy of [OSY21], and make two modifications:

- Require $|\Delta \cdot z| \ll N^\zeta$. This is done by setting $\zeta \geq 2$, so that $N^\zeta \gg \varphi(N) \cdot B$, where z is upper-bounded by $B = \mathsf{poly}(\lambda)$, and $N = \lambda^{\omega(1)}$.
- Lift VOLE over $\mathbb{Z}_{N^\zeta}$ to $\mathbb{Z}$ via a re-randomization technique. Let $r_z \in \mathbb{Z}_{N^\zeta}$ be a pseudorandom shift that the two parties agree on. The parties reset $K_z \leftarrow K_z + r_z \mod N^\zeta$ and $M_z \leftarrow M_z + r_z \mod N^\zeta$.

Now, by the lifting lemma (Lemma 3.2) from [RS21],

$$(K_z + r_z \mod N^\zeta) = (M_z + r_z \mod N^\zeta) + \Delta \cdot z$$

holds over $\mathbb{Z}$ except with probability at most $\frac{|\Delta \cdot z|}{N^\zeta} = \mathsf{negl}(\lambda)$.

Circular-Power DDH in Paillier group. The above instantiation fails in the case that $\zeta = 1$ (e.g., Paillier encryption) and the concrete efficiency might be limited due to the large ciphertext modulus $N^{\zeta+1}$. Here, we give a construction based on the so-called circular-power DDH assumption in Paillier groups [ARS24, ILL24, ILL25]. The high-level idea is very similar, but this time we rewrite Eq.(1) as follows:

$$K_x \cdot K_y = -\Delta \cdot xy + \left((\Delta^2 + \Delta) \cdot xy + \Delta \cdot (x \cdot M_y + y \cdot M_x) + M_x \cdot M_y\right) \quad (2)$$

Instead of fixing Δ to be $\mathsf{sk} = \varphi(N)$ and sending the encryption of sk^{-1} to the prover, we employ a tailored $\mathsf{crs} := (g, h, h_1, h_2)$ that "encodes" Eq.(2), where $g \xleftarrow{\$} \mathbb{Z}^*_{N^2}$, $h = g^r$, $h_1 = h^{\Delta}$ and $h_2 = h^{\Delta^2}(1+N)^{\Delta}$, for $r \xleftarrow{\$} [1, N]$ and $\Delta \xleftarrow{\$} [1, 2^\lambda]$. Now, given crs, a random shift $r_z \in \mathbb{Z}_N$ and subtractive sharings $[\Delta \cdot x], [\Delta \cdot y]$ over $\mathbb{Z}$, the prover can compute

$$M_z \leftarrow -\mathsf{DDLog}\left(h^{M_x M_y} \cdot h_1^{xM_y + yM_x} \cdot h_2^{xy}\right) + r_z \mod N,$$

while the verifier computes

$$K_z \leftarrow -\mathsf{DDLog}\left(h^{K_x K_y}\right) + r_z \mod N.$$

Again, we need $K_z = \Delta \cdot z + M_z$ holds over $\mathbb{Z}$, which is guaranteed by Lemma 3.2 and requiring $N \gg B \cdot 2^\lambda \geq |\Delta \cdot z|$. In particular, we show an alternative method that can relax the restriction on N to be $N = O(2^\lambda) \gg B$. The high-level idea is to decompose Δ into bit strings $\vec{\boldsymbol{\Delta}}$ such that $\Delta = \sum_i \Delta_i \cdot 2^i$ and let the crs include encryptions of digits of Δ. Now, it suffices to maintain $O(\lambda)$ invariants $[\Delta_i \cdot x]$ for every circuit wire value x, which determine $[\Delta \cdot x] := \sum_i [\Delta_i \cdot x] \cdot 2^i$ over $\mathbb{Z}$.

Circular-Power DDH in Prime-order groups. For prime-order groups, there also exist DDLog algorithms [BGI16, BGI17, BGI+18] (but with an inverse polynomial correctness error). We adapt the above construction from Paillier groups by simply replacing the underlying group and the DDLog algorithm. We refer to Appendix C.1 for more details.

Circular-Power RingLWE. Unfortunately, all the above constructions based on group assumptions are not post-quantum secure. Moreover, their concrete computational efficiency seems not promising, since $O(|\mathcal{C}|)$ exponentiation operations in a cyclic group ($\mathbb{Z}^*_{N^{\zeta+1}}$ or $\mathbb{G}$) are required. Towards plausible post-quantum security and concrete efficiency, we extend the above idea and employ the lattice-analogue of DDLog, called distributed decryption algorithm (DDec) [BKS19] to give a construction based on the so-called circular-power RingLWE assumption [ARS24, ILL24, ILL25]. Informally, given a ciphertext ct_x and a subtractive sharing $[\mathsf{sk} \cdot y]$, DDec allows to obtain

$$[x \cdot y] \leftarrow \mathsf{DDec}(\mathsf{ct}_x, [\mathsf{sk} \cdot y]).$$

We refer to Section 4.2 for more details.

Avoid circular-power assumptions via key-switching. For simplicity, we first employ a key-switching technique following [ILL24, ILL25] to avoid circular assumptions. At a high-level, we assign different global keys to every layer of the circuit, so that wire values in the same layer share the same global key. This slightly increases the size of auxiliary materials by a multiplicative factor of the circuit depth. Concretely, assume input subtractive sharings $[\Delta_i \cdot x], [\Delta_i \cdot y]$, the goal is to obtain $[\Delta_{i+1} \cdot xy]$. We slightly modify Eq.(2) into the following form:

$$K_x \cdot K_y = -\Delta_{i+1} \cdot xy + \left((\Delta_i^2 + \Delta_{i+1}) \cdot xy + \Delta_i \cdot (x \cdot M_y + y \cdot M_x) + M_x \cdot M_y\right) \quad (3)$$

Instantiated with Paillier groups, for layer i, the crs includes

$$\left(g, \quad h_i = g^{r_i}, \quad h_{i,1} = h_i^{\Delta_i}, \quad h_{i,2} = h_i^{\Delta_i^2}(1+N)^{\Delta_{i+1}}\right),$$

where $r_i \xleftarrow{\$} [1, N]$. Similarly, given crs, a random shift $r_z \in [1, N]$ and multiplication input $[\Delta_i \cdot x], [\Delta_i \cdot y]$ in layer i, the prover computes

$$M_z \leftarrow -\mathsf{DDLog}\left(h_i^{M_x M_y} \cdot h_{i,1}^{xM_y + yM_x} \cdot h_{i,2}^{xy}\right) + r_z \mod N,$$

while the verifier computes

$$K_z \leftarrow -\mathsf{DDLog}\left(h_i^{K_x K_y}\right) + r_z \mod N.$$

By Lemma 3.2, it holds that $K_z = \Delta_{i+1} \cdot z + M_z$ over $\mathbb{Z}$. And security relies on a "power" assumption that the crs is computationally indistinguishable from random. Inspired from [ILL24, MORS25], we further show constructions based on standard DDH assumptions in Paillier groups by assigning two distinct global keys to one wire value.

3 Preliminaries

Notations. Throughout the paper, we use the symmetric convention for the "mod" operation, i.e., $\mod N : \mathbb{Z} \to (-N/2, N/2]$. A VOLE correlation $K_{\mathbf{x}} = \Delta \cdot \mathbf{x} + M_{\mathbf{x}}$, where the sender holds $\mathbf{x}, M_{\mathbf{x}}$ and the receiver holds $\Delta, K_{\mathbf{x}}$, is also denoted as an additive (subtractive) secret sharing $[\mathbf{x} \cdot \Delta]$, where the sender holds $[\mathbf{x} \cdot \Delta]_0$ and the receiver holds $[\mathbf{x} \cdot \Delta]_1$ such that $[\mathbf{x} \cdot \Delta]_1 = [\mathbf{x} \cdot \Delta]_0 + \mathbf{x} \cdot \Delta$. We also abuse $[\![\mathbf{x}]\!]$ as of $[\mathbf{x} \cdot \Delta]$, when the context is clear.

3.1 Cryptographic Assumptions

In this work, we will use three types of cryptographic assumptions: (i) assumptions in the non-interactive discrete logarithm sharing (NIDLS) framework of [ADOS22], (ii) assumptions with respect to prime-order groups, (iii) lattice assumptions.

Definition 3.1. *The NIDLS framework consists of a triple of* PPT *algorithms* $(\mathsf{Gen}, \mathsf{Sample}, \mathsf{DDLog})$ *with the following syntax:*

– $\mathsf{Gen}(1^\lambda)$ *outputs a tuple* $\mathsf{pp} := (G, F, H, f, q, \ell, \mathsf{aux})$ *where*
  - G *is a finite Abelian group with subgroups* $F, H \leq G$ *such that* $G = F \times H$.
  - F *is a cyclic group of order* $q > 2^\lambda$ *with* f *a generator of* F.
  - ℓ *is an upper-bound on the order of* H, *and* aux *consists of auxiliary information.*
– $\mathsf{Sample}(1^\lambda, \mathsf{pp})$ *outputs an element* $g \in G$ *such that* $f \in \langle g \rangle$, *along with auxiliary information* ρ.
– $\mathsf{DDLog}(\mathsf{pp}, g)$ *is deterministic and outputs an element* $s \in \mathbb{Z}_q$.

The above three algorithms should satisfy at least the following properties:

– *The following two distributions are statistically indistinguishable:*

$$\left\{(\mathsf{pp}, g, \rho, g^r) \mid \mathsf{pp} \leftarrow \mathsf{Gen}(1^\lambda), (g, \rho) \leftarrow \mathsf{Sample}(1^\lambda, \mathsf{pp}), r \xleftarrow{\$} [1, \ell]\right\},$$
$$\left\{(\mathsf{pp}, g, \rho, g') \mid \mathsf{pp} \leftarrow \mathsf{Gen}(1^\lambda), (g, \rho) \leftarrow \mathsf{Sample}(1^\lambda, \mathsf{pp}), g' \xleftarrow{\$} \langle g \rangle\right\}.$$

– *For every* PPT *adversary* $\mathcal{A}$,

$$\Pr\left[s_1 - s_0 \neq m \mod q \mid \begin{array}{c}\mathsf{pp} \leftarrow \mathsf{Gen}(1^\lambda), (g_0, m) \leftarrow \mathcal{A}(1^\lambda, \mathsf{pp}), g_1 := f^m g_0 \\ s_0 \leftarrow \mathsf{DDLog}(\mathsf{pp}, g_0), s_1 \leftarrow \mathsf{DDLog}(\mathsf{pp}, g_1)\end{array}\right] \leq \mathsf{negl}(\lambda).$$

Typically, the DDLog algorithm will accompany with a re-randomization procedure to get subtractive sharings over $\mathbb{Z}$. This technique is yielded by the following lifting lemma.

Lemma 3.2 (Lifting Lemma [RS21]). *For all modulus* $N > 1$ *and subtractive sharings over* $\mathbb{Z}_N$, *i.e.,* $x_0 - x_1 \equiv x \mod N$ *for some* $x \in \mathbb{Z}$, *we have the following*

$$\Pr\left[(x_0 + r \mod N) - (x_1 + r \mod N) = x \mid r \xleftarrow{\$} \mathbb{Z}_N\right] \leq max(1 - \frac{|x|}{N}, 0).$$

Lemma 3.2 directly indicates a philosophy that subtractive sharings of a small value over a large modulus can be efficiently lifting to $\mathbb{Z}$ by the two parties adding a common random shift, on which most HSS constructions rely on. The Power-DDH assumption and Circular-Power DDH assumption in the NIDLS framework are defined as follows.

Definition 3.3 (Power-DDH Assumption). *We say that the Power-DDH assumption holds in the NIDLS framework if the following two distributions are computationally indistinguishable:*

$$\left\{(\mathsf{pp}, \rho, g, g^\alpha, g^{\alpha^2}) \mid \mathsf{pp} \leftarrow \mathsf{Gen}(1^\lambda), (g, \rho) \leftarrow \mathsf{Sample}(1^\lambda, \mathsf{pp}), \alpha \xleftarrow{\$} [1, \ell]\right\},$$
$$\left\{(\mathsf{pp}, \rho, g, g^{\alpha_1}, g^{\alpha_2}) \mid \mathsf{pp} \leftarrow \mathsf{Gen}(1^\lambda), (g, \rho) \leftarrow \mathsf{Sample}(1^\lambda, \mathsf{pp}), (\alpha_1, \alpha_2) \xleftarrow{\$} [1, \ell]^2\right\}.$$

Definition 3.4 (Circular-Power-DDH Assumption). *We say that the Circular-Power-DDH assumption holds in the NIDLS framework if the following two*

distributions are computationally indistinguishable:

$$\left\{(\mathsf{pp}, \rho, g, g^s, g^\alpha, g^{s\alpha}, g^{s^2\alpha} \cdot f^s) \mid \begin{array}{l} \mathsf{pp} \leftarrow \mathsf{Gen}(1^\lambda), (g, \rho) \leftarrow \mathsf{Sample}(1^\lambda, \mathsf{pp}), \\ s, \alpha \xleftarrow{\$} [1, \ell] \end{array}\right\},$$

$$\left\{(\mathsf{pp}, \rho, g, g^s, g^\alpha, g^\beta, g^\gamma) \mid \begin{array}{l} \mathsf{pp} \leftarrow \mathsf{Gen}(1^\lambda), (g, \rho) \leftarrow \mathsf{Sample}(1^\lambda, \mathsf{pp}), \\ s, \alpha, \beta, \gamma \xleftarrow{\$} [1, \ell] \end{array}\right\}.$$

Known instantiations of the NIDLS framework include the Paillier group (and its generalization Damgård-Jurik) [OSY21, RS21], Goldwasser-Micali (and its generalization Joye-Libert) [ADOS22] and class groups [ADOS22]. W.l.o.g., we mainly consider the concrete instantiation from Paillier groups in this work.

Procedure 1: π_{Paillier}

The NIDLS framework instantiated with Paillier groups is given as follows.
$\mathsf{Gen}(1^\lambda)$:

1. Sample random λ-bit safe-primes p_0, p_1, and set $N := p_0 \cdot p_1$.
2. Let $G := \{g^2 \mid g \in \mathbb{Z}^*_{N^2}\}$, $F := \langle N+1 \rangle$, and $H := \{g^N \mid g \in \mathbb{Z}^*_{N^2}\}$.
3. Let $f := N+1$, $q := N > 2^\lambda$, and $\ell := N$.
4. Output $\mathsf{pp} := (G, F, H, f, q, \ell, \mathsf{aux} := N)$.

$\mathsf{Sample}(1^\lambda, \mathsf{pp})$:

1. $\rho \xleftarrow{\$} \mathbb{Z}^*_{N^2}$.
2. Output $(g := \rho^2, \rho)$. //$N+1 \in \langle g^2 \rangle$.

$\mathsf{DDLog}(\mathsf{pp}, g)$:

1. $s := \log_{N+1}(\frac{g}{g \mod N}) \equiv \frac{g/(g \mod N)-1}{N} \mod N^2$.
2. Output $s \in \mathbb{Z}_N$.

Algorithms for the Paillier encryption scheme are defined as follows.
$\mathsf{KeyGen}(1^\lambda)$:

1. Run $(N, p_0, p_1) \leftarrow \mathsf{RSA.Gen}(1^\lambda)$.
2. Compute $\mathsf{sk} \in \mathbb{Z}$ such that $\mathsf{sk} \equiv 0 \mod \varphi(N)$, and $\mathsf{sk} \equiv 1 \mod N$, i.e. $\mathsf{sk} = \varphi(N) \cdot (\varphi(N)^{-1} \mod N^2)$, where $\varphi(\cdot)$ denotes the Euler function.
3. Output $(\mathsf{pk} := N, \mathsf{sk})$.

$\mathsf{Enc}(x)$:

1. Sample $r \xleftarrow{\$} \mathbb{Z}^*_{N^2}$.
2. Compute $\mathsf{ct} := r^N \cdot (1+N)^x \equiv r^N(1+xN) \mod N^2$.
3. Output ciphertext ct.

$\mathsf{Eval}(\mathsf{ct}_1, \mathsf{ct}_2, \alpha)$:

1. Output $\mathsf{ct} := \mathsf{ct}_1^\alpha \cdot \mathsf{ct}_2$. //Homomorphic evaluate $(x, y) \to (\alpha \cdot x + y)$

$\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$:

1. Output $x := \frac{(\mathsf{ct}^{\mathsf{sk}} \mod N^2)-1}{N} \in \mathbb{Z}_N$.

The security of Paillier encryption relies on the DCR Assumption.

Definition 3.5 (Decisional Composite Residuosity (DCR) Assumption). *We say that the DCR problem is hard, if the following distributions are computationally indistinguishable:*

$$\left\{(N,x) \mid \begin{array}{l}(N,p,q) \leftarrow \mathsf{RSA.Gen} \\ x \xleftarrow{\$} \mathbb{Z}_{N^2}^*\end{array}\right\}, \left\{(N, x^N \mod N^2) \mid \begin{array}{l}(N,p,q) \leftarrow \mathsf{RSA.Gen} \\ x \xleftarrow{\$} \mathbb{Z}_{N^2}^*\end{array}\right\}$$

Lattice Hardness Assumptions. Let $\mathcal{R} := \mathbb{Z}[X]/(X^N+1)$, where N is a power of 2. Let $q(\lambda)$ be a sufficiently large modulus, and $\mathcal{D}_{sk}(\lambda), \mathcal{D}_e(\lambda) \subset \mathcal{R}$ be secret and error distributions. Denote $\mathcal{R}/(q\mathcal{R})$ by $\mathcal{R}_q$. The standard Ring-LWE assumption indicates that for every polynomial $m(\lambda)$, the following two distributions are computationally indistinguishable:

$$\left\{\mathbf{a}, \mathbf{b} \leftarrow \mathcal{R}_q^m\right\}, \left\{\mathbf{a}, s \cdot \mathbf{a} + \mathbf{e} \mid \mathbf{a} \leftarrow \mathcal{R}_q^m, s \leftarrow \mathcal{D}_{sk}, \mathbf{e} \leftarrow \mathcal{D}_e\right\}.$$

We will use the following two variants of Power Ring-LWE assumptions [ARS24].

Definition 3.6 (Power Ring-LWE [ILL25]). *We say that the power ring-LWE assumption holds with respect to the ring $\mathcal{R}$, modulus q, error and secret distribution $\mathcal{D}_e, \mathcal{D}_{sk}$, if the following two distributions are computationally indistinguishable:*

$$\left\{\mathbf{a}, \mathbf{b}, \mathbf{c} \leftarrow \mathcal{R}_q^m\right\}, \left\{\mathbf{a}, s \cdot \mathbf{a} + \mathbf{e}_1, s^2 \cdot \mathbf{a} + \mathbf{e}_2 \mid \mathbf{a} \leftarrow \mathcal{R}_q^m, s \leftarrow \mathcal{D}_{sk}, \mathbf{e}_1, \mathbf{e}_2 \leftarrow \mathcal{D}_e\right\}$$

Definition 3.7 (Circular-Power Ring-LWE [ILL25]). *We say that the circular-power ring-LWE assumption holds with respect to the ring $\mathcal{R}$, modulus p, q with $p \mid q$, error and secret distribution $\mathcal{D}_e, \mathcal{D}_{sk}$, if the following two distributions are computationally indistinguishable:*

$$\left\{\mathbf{a}, \mathbf{b}, \mathbf{c} \leftarrow \mathcal{R}_q^m\right\}, \left\{\mathbf{a}, s \cdot \mathbf{a} + \mathbf{e}_1, s^2 \cdot \mathbf{a} + \mathbf{e}_2 + \frac{q}{p}s \mid \mathbf{a} \leftarrow \mathcal{R}_q^m, s \leftarrow \mathcal{D}_{sk}, \mathbf{e}_1, \mathbf{e}_2 \leftarrow \mathcal{D}_e\right\}$$

3.2 Succinct Non-interactive Arguments of Knowledge

We define the notions for succinct arguments [BCI+22]. Let R be an efficiently computable binary relation consisting of pairs of the form (x, w), where x is a statement and w is a witness. Let $\mathcal{L} = \{x \mid \exists w \ s.t. \ R(x,w) = 1\}$ be the language associated with R, and λ be a computational security parameter. A non-interactive argument system for $\mathcal{L}$ consists of three PPT algorithms $(\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify})$ defined as follows:

- $\mathsf{Setup}(1^\lambda)$ outputs a common (structured) reference string crs together with a private verification key sk.
- $\mathsf{Prove}(\mathsf{crs}, x, w)$ outputs an argument π.

– $\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, x, \pi)$ outputs a symbol accept or reject.

Definition 3.8 (SNARG). *A triple of* PPT *algorithms* $(\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify})$ *is a succinct non-interactive argument (SNARG) for an NP language* $\mathcal{L}$*, if the followings hold:*

1. *Completeness. For all* $(x, w) \in \mathcal{L}$*,*

$$\Pr\left[\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, x, \pi) = \mathsf{accept} \,\middle|\, \begin{matrix}(\mathsf{crs}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda) \\ \pi \leftarrow \mathsf{Prove}(\mathsf{crs}, x, w)\end{matrix}\right] \geq 1 - \mathsf{negl}(\lambda)$$

2. *(Non-adaptive) Soundness. For any* PPT *adversary* $\mathcal{A}$ *and any* $x \notin \mathcal{L}$*,*

$$\Pr\left[\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, x, \pi^*) = \mathsf{accept} \,\middle|\, \begin{matrix}(\mathsf{crs}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda) \\ \pi^* \leftarrow \mathcal{A}^{\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, \cdot, \cdot)}(\mathsf{crs}, x)\end{matrix}\right] \leq \mathsf{negl}(\lambda).$$

3. *Succinctness. For any* x *and* w*, the length of the proof* π *is* $\omega_\lambda(|x| + |w|)$*. In particular, full succinctness requires that the proof size as well as the* crs *length is* $O_\lambda\big(\mathsf{polylog}(|x| + |w|)\big)$*.*

We remark that the above definition essentially gives a multi-theorem SNARG, as the adversary $\mathcal{A}$ is given access to a verification oracle [BCI+22]. A SNARG of knowledge (SNARK) is a SNARG with a stronger knowledge soundness property, where an extractor can extract a witness from interacting with the adversary.

Definition 3.9 (SNARK). *A triple of* PPT *algorithms* $(\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify})$ *is a SNARK for a language* $\mathcal{L}$*, if it is a SNARG where soundness is replaced by the following stronger requirement:*

– *Knowledge soundness. For any* PPT *adversary* $\mathcal{A}$ *and statement* x*, the following holds:*

$$\Pr\left[\begin{matrix}\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, x, \pi^*) = \mathsf{accept} \\ (x, w^*) \notin \mathcal{L}\end{matrix} \,\middle|\, \begin{matrix}(\mathsf{crs}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda) \\ \pi^* \leftarrow \mathcal{A}^{\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, \cdot, \cdot)}(\mathsf{crs}, x) \\ w^* \leftarrow \mathsf{Ext}^{\mathcal{A}}(\mathsf{crs}, x, \pi^*)\end{matrix}\right] \leq \mathsf{negl}(\lambda),$$

where Ext *denotes an extractor, who has oracle access to (the next-message functions of)* $\mathcal{A}$*.*

Definition 3.10 (zkSNARK). *A triple of* PPT *algorithms* $(\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify})$ *is a zero-knowledge SNARK (zkSNARK) for a language* $\mathcal{L}$*, if it is a SNARK additionally satisfying the zero-knowledge property:*

– *There exists a* PPT *simulator* $(\mathcal{S}_1, \mathcal{S}_2)$ *such that* $\mathcal{S}_1$ *outputs a simulated* $(\mathsf{crs}, \mathsf{sk})$ *along with a trapdoor* τ*; and given* crs*,* τ*, and a statement* x*,* $\mathcal{S}_2$ *outputs a simulated proof* π*; and for any* PPT *adversary* $\mathcal{A}$*, the following holds:*

$$\Big|\Pr\Big[\mathcal{A}^{\mathsf{Prove}(\mathsf{crs}, \cdot, \cdot)}(\mathsf{crs}, \mathsf{sk}) = 1 \Big| (\mathsf{crs}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)\Big] - \Pr\big[\mathcal{A}^{\mathcal{S}_2(\tau, \cdot, \cdot)}(\mathsf{crs}, \mathsf{sk}) = 1 \,\big|\, (\mathsf{crs}, \mathsf{sk}, \tau) \leftarrow \mathcal{S}_1(1^\lambda)\big]\Big| \leq \mathsf{negl}(\lambda).$$

Designated vs Public verifiability. If sk is an empty string $\emptyset$, then it is a publicly-verifiable zkSNARK (pv-zkSNARK). Otherwise, it is a designated-verifier zk-SNARK (dv-zkSNARK), where the verifier holds the private verification key sk. Throughout the paper, we will focus on the designated-verifier setting.

Multi-Theorem soundness. It is desirable that the crs can be reused in multiple proofs especially when the trusted setup is costly. This property is automatically guaranteed in the context of pv-zkSNARK. However, in the designated-verifier setting, reusability requires a stronger guarantee: soundness must hold even against a malicious prover with access to a proof verification oracle. Our dv-zkSNARK satisfies this stronger security notion.

4 Succinct Line-Point Zero-Knowledge Arguments

In this section, we propose a new framework of building succinct arguments for circuit satisfiability, and instantiate it with various existing assumptions. In particular, we consider B-bounded computations, that is, one can simply view the verification circuit is defined over $\mathbb{Z}$ and there is no overflow during the evaluations when modulo B. At a high-level, our succinct argument scheme inherits the nature of line-point zero-knowledge proof systems, i.e., the prover and the verifier jointly evaluate the circuit and check the output via vector-OLE correlations. More specifically, the evaluation is operated through non-interactively maintaining the VOLE invariant for each wire value x in the circuit:

$$K_x = \Delta \cdot x + M_x,$$

where the verifier $\mathcal{V}$ holds K_x, Δ, and the prover $\mathcal{P}$ holds M_x, x. In the VOLE-ZK literature, Δ is known as the global key, K_x is the MAC key, while M_x is referred to as the MAC tag. Typically, joint evaluation requires at least linear communication in the number of gates. However, we come up with a new technique that allows non-interactive circuit evaluation, eliminating such dependence on the circuit size. Since no interaction is required as long as the input VOLE correlations are given, it suffices to check the output VOLE via opening the MAC tag to the verifier.

In Section 4.1, we give succinct LPZK arguments based on variants of Paillier group assumptions. And in Section 4.2, we give constructions from lattice assumptions. Finally in Appendix C.1 & Appendix C.2, we give constructions based on assumptions in prime-order groups, and the KDM-security of Damgård-Jurik cryptosystem, respectively.

4.1 Succinct LPZK Arguments from Paillier Groups

Since VOLE correlations have linearity, non-interactively evaluating addition gates is straightforward: given $K_x = \Delta \cdot x + M_x, K_y = \Delta \cdot y + M_y$, denoted by $[\![x]\!], [\![y]\!]$, the parties compute $[\![z]\!] := [\![x]\!] + [\![y]\!]$,

$$\underbrace{K_x + K_y}_{K_z} = \Delta \cdot \underbrace{(x+y)}_{z} + \underbrace{(M_x + M_y)}_{M_z}.$$

Hence, it suffices to consider multiplication gates. To this end, we introduce Paillier groups [Pai99] and utilize the DDLog algorithm [OSY21], inspired by the recent works [MORS24, ILL24, MORS25, ILL25], where the related notions are presented in π_{Paillier}.

Let $N = p \cdot q$ be a secure RSA modulus, and $\mathbb{Z}_{N^2}^* = \mathbb{Z}_N \times \mathbb{Z}_N^*$ be a Paillier group. Recall that the DDLog algorithm [OSY21] allows the two parties to locally convert multiplicative sharings over the exponent (in $\mathbb{Z}_{N^2}^*$) to subtractive sharings (over $\mathbb{Z}_N$), e.g., given $g_v = g_p \cdot (1+N)^{\Delta \cdot x}$, it holds that

$$\mathsf{DDLog}(g_v) = \mathsf{DDLog}(g_p) + \Delta \cdot x \mod N.$$

The main observation is that, given input VOLE variants $[\![x]\!], [\![y]\!]$, we have

$$K_x K_y = -\Delta \cdot xy + \big((\Delta^2 + \Delta) \cdot xy + \Delta \cdot (yM_x + xM_y) + M_x M_y\big).$$

To get $[\![xy]\!]$, we need to let the parties obtain subtractive sharings $\big[(\Delta^2 + \Delta) \cdot xy\big]$, and $[\Delta \cdot (yM_x + xM_y)]$, where $\mathcal{P}$ knows x, y, M_x, M_y, while $\mathcal{V}$ knows Δ. At a high-level, to avoid interactions, $\mathcal{P}$ is given secure "encryptions" of Δ as the auxiliary information, so that they can multiply in the exponent, and get the subtractive sharings via DDLog.

In more detail, let $g \xleftarrow{\$} \mathbb{Z}_{N^2}^*$ be a random N-th residue over the multiplicative group $\mathbb{Z}_{N^2}^*$, which can be obtained via sampling $g_0 \xleftarrow{\$} \mathbb{Z}_{N^2}$ and setting $g := g_0^N$. Let $r \xleftarrow{\$} \mathbb{Z}_N$ and the crs includes (g, h, h_1, h_2), where

$$h := g^r, \quad h_1 := h^{\Delta}, \quad \text{and } h_2 := h^{\Delta^2} \cdot (1+N)^{\Delta}.$$

Given crs as a "encryptions" of Δ and $[\![x]\!], [\![y]\!]$,

1. $\mathcal{P}$ computes $u := h^{M_x \cdot M_y} \cdot h_1^{x \cdot M_y + y \cdot M_x} \cdot h_2^{xy}$ and $M_z := -\mathsf{DDLog}(u)$.
2. $\mathcal{V}$ computes $v := h^{K_x \cdot K_y}$ and $K_z := -\mathsf{DDLog}(v)$.

It holds that $K_z = \Delta \cdot z + M_z \mod N$ with $z := xy$, and the correctness follows from DDLog and:

$$\begin{aligned}
\frac{u}{v} &= \frac{h^{M_x \cdot M_y} \cdot h_1^{x \cdot M_y + y \cdot M_x} \cdot h_2^z}{h^{K_x \cdot K_y}} \\
&= h^{M_x \cdot M_y + (x \cdot M_y + y \cdot M_x) \cdot \Delta + \Delta^2 \cdot z - K_x \cdot K_y} \cdot (1+N)^{\Delta \cdot z} \\
&= (1+N)^{\Delta \cdot z}.
\end{aligned}$$

As we basically require subtractive sharings over $\mathbb{Z}$, we select $\Delta \xleftarrow{\$} [1, 2^\lambda]$ and B, N satisfying $B \cdot 2^\lambda \ll N$, and employ the re-randomization technique from [OSY21]. By the lifting lemma (Lemma 3.2), the following

$$(K_z + r_z \mod N) = \Delta \cdot z + (M_z + r_z \mod N)$$

holds over $\mathbb{Z}$ with an overwhelming probability, where $r_z \in \mathbb{Z}_N$ is a common random shift with respect to z. Security follows from the assumption that

(g, h, h_1, h_2) is computationally indistinguishable from (g, h, h^a, h^b), where $a, b \xleftarrow{\$} \mathbb{Z}_N$, i.e., the so-called circular power DDH assumption in Paillier groups [ILL24, ILL25].

Remark 4.1. Note that in VOLE, $\mathcal{V}$ has Δ and thus the correctness of the crs for $\mathcal{V}$ is verifiable. In addition, both parties' computational costs are dominated by $|\mathcal{C}|$ exponentiations in a Paillier group. Moreover, the proving and verifying procedure are streaming, since the VOLE correlations are evaluated gate-by-gate.

We present a self-contained protocol in Π_{zkPai}, assuming access to an ideal functionality $\mathcal{F}_{\mathsf{hcVOLE}}$ that produces input VOLE correlations $[\![\mathbf{w}]\!]$. By instantiating $\mathcal{F}_{\mathsf{hcVOLE}}$ with extractable succinct half-chosen VOLE protocols (see Section 5) adapted from [ARS24, AMR25], we immediately obtain succinct LPZK arguments, a.k.a. dv-zkSNARKs. The compiled results are referred to Corollary 5.2.

Protocol 2: Π_{zkPai}

Let $\mathcal{C} : \mathbb{Z}^n \to \mathbb{Z}$ be a B-bounded verification circuit, such that $B \cdot 2^\lambda \ll N$. The prover $\mathcal{P}$ wants to convince the verifier $\mathcal{V}$ that he holds $\mathbf{x} \in \mathbb{Z}^n$ such that $\mathcal{C}(\mathbf{x}) = 0$. Assume oracle access to a half-chosen VOLE functionality $\mathcal{F}_{\mathsf{hcVOLE}}$.
$\mathsf{crs}{:=}(N, g, h, h_1, h_2)$ where N is a secure RSA modulus, $g := g_0^N$ with $g_0 \xleftarrow{\$} \mathbb{Z}_{N^2}$, $h := g^r$ with $r \xleftarrow{\$} [1, N]$, $h_1 := h^\Delta$ with $\Delta \xleftarrow{\$} [1, 2^\lambda]$, $h_2 := h^{\Delta^2} \cdot (1+N)^\Delta$. //We implicitly assume crs includes a PRF key to produce common random shifts.

1. Both parties are given crs and $\mathcal{V}$ is additionally given $\Delta \in [1, 2^\lambda]$.
2. $\mathcal{P}$ and $\mathcal{V}$ invoke $\mathcal{F}_{\mathsf{hcVOLE}}$ with their inputs $\mathbf{x}, \Delta$, respectively. Then they receive $M_{\mathbf{x}}, K_{\mathbf{x}}$, where $K_{\mathbf{x}} = \Delta \cdot \mathbf{x} + M_{\mathbf{x}}$.
3. For each gate G in the circuit with respect to input wire values x, y,
   - If G is an addition gate, $\mathcal{P}$ computes $z := x + y, M_z := M_x + M_y$, while $\mathcal{V}$ computes $K_z := K_x + K_y$.
   - If G is a multiplication gate. Let $r_z \in \mathbb{Z}_N$ be a common random shift with respect to output wire z, $\mathcal{P}$ computes
   $$M_z \leftarrow -\mathsf{DDLog}\left(h_2^{xy} \cdot h_1^{yM_x + xM_y} \cdot h^{M_x M_y}\right) + r_z \mod N,$$
   while $\mathcal{V}$ computes $K_z \leftarrow -\mathsf{DDLog}(h^{K_x K_y}) + r_z \mod N$.
   //By the randomness of r_z and Lemma 3.2, $K_z = \Delta \cdot z + M_z$ holds over $\mathbb{Z}$ except with probability at most $\frac{|\Delta \cdot xy|}{N}$.
4. For the single output wire ω, $\mathcal{P}$ opens M_ω to $\mathcal{V}$. $\mathcal{V}$ accepts if $K_\omega = M_\omega$. Otherwise, $\mathcal{V}$ aborts.

Theorem 4.2. *Assuming circular-power DDH in Paillier groups (Definition 3.4) and secure PRFs, Π_{zkPai} is secure in the $(\mathcal{F}_{\mathsf{hcVOLE}}, CRS)$-hybrid model.*

Proof. Correctness directly follows by design and we defer a detailed security proof to Appendix D.1. □

On the choice of Δ. We show an alternative construction that relaxes the requirement $N = 2^\lambda \cdot \lambda^{\omega(1)}$ to $N = \lambda^{\omega(1)}$, by applying a bit-decomposition idea. Let $\vec{\Delta}$ denote the bit-decomposition of $\Delta \in [0, 2^\lambda)$, i.e., $\Delta = \sum_{i=1}^{\lambda} \Delta_i \cdot 2^{i-1}$. More specifically, here we let the crs include $(g, \{h_i, h_{i,1}, h_{i,2}\}_{i\in[1,\lambda]})$, where

$$r_i \xleftarrow{\$} \mathbb{Z}_N, \quad h_i := g^{r_i}, \quad h_{i,1} := h_i^{\Delta}, \quad \text{and } h_{i,2} := h_i^{\Delta^2} \cdot (1+N)^{\Delta_i}.$$

Given crs and subtractive sharings $[\Delta_i \cdot x], [\Delta_i \cdot y]$ and common random shifts $s_{z,i} \xleftarrow{\$} \mathbb{Z}_N$, for $i \in [1, \lambda]$, $K_{x,i} = \Delta_i \cdot x + M_{x,i}$ holds over $\mathbb{Z}$ with overwhelming probability and therefore

$$K_x := \sum_{i=1}^{\lambda} K_{x,i} \cdot 2^{i-1} = \sum_{i=1}^{\lambda} (\Delta_i \cdot x + M_{x,i}) \cdot 2^{i-1} = \underbrace{(\sum_{i=1}^{\lambda} \Delta_i \cdot 2^{i-1})}_{\Delta} \cdot x + \underbrace{\sum_{i=1}^{\lambda} M_{x,i} \cdot 2^{i-1}}_{M_x}$$

from Lemma 3.2. Then

1. $\mathcal{P}$ computes $u_i := h_i^{M_x \cdot M_y} \cdot h_{i,1}^{x \cdot M_y + y \cdot M_x} \cdot h_{i,2}^{xy}$ and $M_{z,i} := -\mathsf{DDLog}(u_i) + s_{z,i} \mod N$.
2. $\mathcal{V}$ computes $v_i := h_i^{K_x \cdot K_y}$ and $K_z := -\mathsf{DDLog}(v_i) + s_{z,i} \mod N$.

By Lemma 3.2, $K_{z,i} = \Delta_i \cdot z + M_{z,i}$ holds over $\mathbb{Z}$ with a high probability (i.e., $1 - \frac{|B|}{N}$) and therefore

$$K_z := \sum_{i=1}^{\lambda} K_{z,i} \cdot 2^{i-1} = \sum_{i=1}^{\lambda} (\Delta_i \cdot z + M_{z,i}) \cdot 2^{i-1} = \underbrace{(\sum_{i=1}^{\lambda} \Delta_i \cdot 2^{i-1})}_{\Delta} \cdot z + \underbrace{\sum_{i=1}^{\lambda} M_{z,i} \cdot 2^{i-1}}_{M_z}.$$

In this case, it suffices to require $N = \lambda^{\omega(1)}$ for $B = \mathsf{poly}(\lambda)$ and the security follows from the circular-power DDH assumption (Definition 3.4) in Paillier groups that $(g, \{h_i, h_{i,1}, h_{i,2}\}_{i\in[1,\lambda]})$ is computationally indistinguishable from $(g, \{h_i, h_i^{a_i}, h_i^{b_i}\}_{i\in[1,\lambda]})$, where $a_i, b_i \xleftarrow{\$} \mathbb{Z}_N$.

Key-Switching. In addition, we show key-switching techniques to avoid assumptions of the "circular-power" flavor. We first show how to remove "circular", followed by a construction solely based on DDH assumptions in Paillier groups.

Step One. At a high-level, we assign different global keys to every layer of the circuit, so that wire values in the same layer i share the same global key Δ_i. Concretely, given input subtractive sharings $[\Delta_i \cdot x], [\Delta_i \cdot y]$, the goal is to obtain $[\Delta_{i+1} \cdot xy]$. We slightly modify Eq.(2) into the following form:

$$K_x \cdot K_y = -\Delta_{i+1} \cdot xy + \big((\Delta_i^2 + \Delta_{i+1}) \cdot xy + \Delta_i \cdot (x \cdot M_y + y \cdot M_x) + M_x \cdot M_y\big).$$

Based on that, we instead let the crs include for each layer i

$$\big(g, \quad h_i = g^{r_i}, \quad h_{i,1} = h_i^{\Delta_i}, \quad h_{i,2} = h_i^{\Delta_i^2} (1+N)^{\Delta_{i+1}}\big),$$

where $r_i \xleftarrow{\$} [1, N]$. Similarly, given crs, a random shift $r_z \in [1, N]$ and multiplication inputs $[\Delta_i \cdot x], [\Delta_i \cdot y]$ in layer i,

1. $\mathcal{P}$ computes $u := h_i^{M_x \cdot M_y} \cdot h_{i,1}^{x \cdot M_y + y \cdot M_x} \cdot h_{i,2}^{xy}$ and $M_z := -\mathsf{DDLog}(u) + r_z \mod N$.
2. $\mathcal{V}$ computes $v := h_i^{K_x \cdot K_y}$ and $K_z := -\mathsf{DDLog}(v) + r_z \mod N$.

By Lemma 3.2, $K_z = \Delta_{i+1} \cdot z + M_z$ holds over $\mathbb{Z}$ with a high probability. And security relies on a "power" assumption (Definition 3.3) that the crs is computationally indistinguishable from random.

Step Two. Inspired by [ILL24, MORS25], we further eliminate the need for "power" assumptions. The high-level idea is to assign two global keys Δ_i^L, Δ_i^R to each value x in layer i, and instead maintain the two invariants $([\Delta_i^L \cdot x], [\Delta_i^R \cdot x])$ simultaneously. Now we let the crs include

$$\big(g, \quad h_{i,L} = g^{r_{i,L}}, h_{i,R}, \quad h_{i,1}^{LR} = h_{i,R}^{\Delta_i^L}, h_{i,1}^{RL}, h_{i,1}^{LL}, h_{i,1}^{RR}, \quad h_{i,2}^{L} = h_{i,L}^{\Delta_i^L \cdot \Delta_i^R} (1+N)^{\Delta_{i+1}^L}, h_{i,2}^{R}\big),$$

where $h_{i,R} = g^{r_{i,R}}$, $h_{i,1}^{RL} = h_{i,L}^{\Delta_i^R}$, $h_{i,1}^{LL} = h_{i,L}^{\Delta_i^L}$, $h_{i,1}^{RR} = h_{i,R}^{\Delta_i^R}$, and $h_{i,2}^{R} = h_{i,R}^{\Delta_i^L \cdot \Delta_i^R} (1+N)^{\Delta_{i+1}^R}$ are defined analogously, for random $r_{i,L}, r_{i,R}$.

Given crs, random shifts $r_z^L, r_z^R \in [1, N]$ and multiplication inputs $([\Delta_i^L \cdot x], [\Delta_i^R \cdot x])$, $([\Delta_i^L \cdot y], [\Delta_i^R \cdot y])$ in layer i,

1. $\mathcal{P}$ computes $M_z^L := -\mathsf{DDLog}(u^L) + r_z^L \mod N$, and $M_z^R := -\mathsf{DDLog}(u^R) + r_z^R \mod N$, where

$$u^L = h_{i,L}^{M_x^L \cdot M_y^R} \cdot (h_{i,1}^{LL})^{x \cdot M_y^R} \cdot (h_{i,1}^{RL})^{y \cdot M_x^L} \cdot (h_{i,2}^{L})^{xy}, \ \textit{and}$$

$$u^R = h_{i,R}^{M_y^L \cdot M_x^R} \cdot (h_{i,1}^{RR})^{x \cdot M_y^L} \cdot (h_{i,1}^{LR})^{y \cdot M_x^R} \cdot (h_{i,2}^{R})^{xy}.$$

2. $\mathcal{V}$ first computes $v^L := h_{i,L}^{K_x^L \cdot K_y^R}$, $v^R := h_{i,R}^{K_x^R \cdot K_y^L}$, and then $K_z^L := -\mathsf{DDLog}(v^L) + r_z^L \mod N$, and $K_z^R := -\mathsf{DDLog}(v^R) + r_z^R \mod N$.

Correctness of $K_z^L = \Delta_{i+1}^L \cdot z + M_z^L$, and $K_z^R = \Delta_{i+1}^R \cdot z + M_z^R$ over $\mathbb{Z}$ follows from Lemma 3.2 and the equations:

$$K_x^L \cdot K_y^R = -\Delta_{i+1}^L \cdot xy + \big((\Delta_i^L \Delta_i^R + \Delta_{i+1}^L) \cdot xy + \Delta_i^L \cdot x \cdot M_y^R + \Delta_i^R \cdot y \cdot M_x^L) + M_x^L \cdot M_y^R\big),$$

$$K_y^L \cdot K_x^R = -\Delta_{i+1}^R \cdot xy + \big((\Delta_i^L \Delta_i^R + \Delta_{i+1}^R) \cdot xy + \Delta_i^R \cdot x \cdot M_y^L + \Delta_i^L \cdot y \cdot M_x^R) + M_x^R \cdot M_y^L\big).$$

While security relies on standard DDH assumptions in Paillier groups.

4.2 Succinct LPZK Arguments from Lattice

The constructions of succinct LPZK arguments in Section 4.1 rely on variants of group assumptions, which are not post-quantum secure. In this section, we add plausible post-quantum security by extending the construction to build upon lattice assumptions.

To begin with, we introduce the distributed decryption technique (denoted as DDec) in [BKS19], which is a lattice-analogue of the DDLog algorithms over groups. Informally, given a ciphertext ct_y and subtractive sharings $[\Delta \cdot x]$, DDec allows the two parties to locally compute

$$[xy] \leftarrow \mathsf{DDec}(\mathsf{ct}_y, [\Delta \cdot x]).$$

We leverage this technique and again the equality

$$K_z := K_x K_y = \Delta \cdot xy + \underbrace{\left((\Delta^2 - \Delta) \cdot xy + \Delta \cdot (yM_x + xM_y) + M_x M_y\right)}_{M_z}$$

to compute $[\Delta \cdot xy]$ from a tailored ciphertext ct_Δ and $[\Delta \cdot x]$, $[\Delta \cdot y]$ non-interactively. Let $\mathcal{R} = \mathbb{Z}[\mathtt{X}]/(\mathtt{X}^N + 1)$, where N is a power-of-two. The ciphertext space and plaintext space are $\mathcal{R}_q := \mathcal{R} \mod q$ and $\mathcal{R}_p := \mathcal{R} \mod p$, respectively. Specifically, the ciphertext of Δ (included in crs) has the following form:

$$\mathsf{ct}_\Delta := (a, b, c) = (a,\ a \cdot \Delta + e_1,\ a \cdot \Delta^2 + e_2 - \frac{q}{p}\Delta),$$

where $a \xleftarrow{\$} \mathcal{R}_q$, and e_1, e_2 are small errors sampled from an error distribution $\mathcal{D}_e(\mathcal{R}_q)$. Let $r_{\mathsf{DDec}} \in \mathcal{R}_q$, $r_z \in \mathcal{R}_p$ be (fresh) common random shifts, given crs and $[\Delta \cdot x]$, $[\Delta \cdot y]$, the parties do the following:

1. $\mathcal{P}$ computes $u := aM_xM_y + b(xM_y + yM_x) + cxy + r_{\mathsf{DDec}} \mod q$, and

$$M_z := \mathsf{DDec}(u) + r_z \mod p = \left\lfloor \frac{p}{q} \cdot u \right\rceil + r_z \mod p.$$

2. $\mathcal{V}$ computes $v := aK_xK_y + r_{\mathsf{DDec}} \mod q$, and

$$K_z := \mathsf{DDec}(v) + r_z \mod p = \left\lfloor \frac{p}{q} \cdot v \right\rceil + r_z \mod p.$$

In the above, $\lfloor \cdot \rceil$ denotes the standard rounding operation [DHRW16]. To establish correctness, we require the following conditions to be satisfied:

(i) $p > B \cdot B_\Delta \cdot \lambda^{\omega(1)}$, where B_Δ is the upper bound of the secret Δ.
(ii) $p \mid q$ and $\frac{q}{p} > (B_e \cdot B^2 + 2B_eBp) \cdot \lambda^{\omega(1)}$, where B_e is the upper bound of the errors e_1, e_2.
(iii) subtractive sharings $[\Delta \cdot x], [\Delta \cdot y]$ are over $\mathcal{R}$ with $K_x, K_y, M_x, M_y \in \mathcal{R}_p$.

Then it follows that

$$
\begin{aligned}
u &:= aM_xM_y + b(xM_y + yM_x) + cxy + r_{\mathsf{DDec}} \\
&\equiv \underbrace{\left(aM_xM_y + \Delta(xM_y + yM_x) + \Delta^2 xy + r_{\mathsf{DDec}}\right)}_{v := K_xK_y + r_{\mathsf{DDec}}} - \frac{q}{p}\Delta xy + \underbrace{e_2 xy + e_1(xM_y + yM_x)}_{\|error\|_\infty \le Bp \cdot \mathsf{poly}(\lambda) \ll \frac{q}{p}} \mod q
\end{aligned}
$$

Hence, by [BKS19, Lemma 1], the following holds except with a negligible probability:

$$
\begin{aligned}
K_z &\equiv \left\lfloor \frac{p}{q}(aK_xK_y + r_{\mathsf{DDec}} \mod q) \right\rceil + r_z \\
&= \left\lfloor \frac{p}{q}\left(aM_xM_y + b(xM_y + yM_x) + cxy + r_{\mathsf{DDec}} \mod q\right) \right\rceil + r_z + \Delta \cdot xy \\
&\equiv M_z + \Delta \cdot xy \mod p
\end{aligned}
$$

Finally, by the lifting lemma over $\mathcal{R}$ [BKS19, Lemma 2], $K_z = \Delta \cdot z + M_z$ holds over $\mathcal{R}$ except with a negligible probability, since by condition (i), $p > B \cdot B_\Delta \cdot \lambda^{\omega(1)}$ such that $\frac{B \cdot B_\Delta}{p} < \mathsf{negl}(\lambda)$.

Remark 4.3. To guarantee every input subtractive sharings $[\Delta \cdot x]$ satisfy condition (iii), the parties need to re-randomize their sharings via consuming a common random shift $r_x \xleftarrow{\$} \mathcal{R}_p$. In more detail, provided that $p > B \cdot B_\Delta \cdot \lambda^{\omega(1)}$, the following holds with a high probability

$$(K_x + r_x \mod p) = \Delta \cdot x + (M_x + r_x \mod p).$$

Equivalently, it suffices to perform the re-randomization procedure on the output wire of every input/addition/multiplication gate.

We present a self-contained construction Π_{zkLWE} based on circular-power RingLWE assumptions (Definition 3.7), assuming oracle access to a half-chosen VOLE functionality $\mathcal{F}_{\mathsf{hcVOLE}}$. Similarly by instantiating $\mathcal{F}_{\mathsf{hcVOLE}}$ with any (extractable) succinct half-chosen VOLE protocol, one immediately get a dv-zkSNARK. In particular, we compile with the LWE-based half-chosen VOLE protocol in Section 5.2, and obtain fully succinct dv-zkSNARKs (see Corollary 5.5).

Protocol 3: Π_{zkLWE}

Let $\mathcal{C} : \mathcal{R}^n \to \mathcal{R}$ be a B-bounded verification circuit. The prover $\mathcal{P}$ wants to convince the verifier $\mathcal{V}$ that he holds $\mathbf{x} \in \mathcal{R}^n$ such that $\mathcal{C}(\mathbf{x}) = 0$. Assume oracle access to a half-chosen VOLE functionality $\mathcal{F}_{\mathsf{hcVOLE}}$.

crs: (N, p, q, a, b, c) where N is a power of two, p, q satisfy conditions (i) & (ii), $a \xleftarrow{\$} \mathcal{R}_q$, $b := a \cdot \Delta + e_1$, $c := a \cdot \Delta^2 + e_2 - \frac{q}{p}\Delta$ with $\Delta \xleftarrow{\$} \mathcal{D}_{sk}$, $e_1, e_2 \xleftarrow{\$} \mathcal{D}_e$.
//We implicitly assume crs includes a PRF key to produce common random shifts.

1. Both parties are given crs and $\mathcal{V}$ is additionally given Δ.
2. $\mathcal{P}$ (as sender) and $\mathcal{V}$ (as receiver) invoke $\mathcal{F}_{\mathsf{hcVOLE}}$ with their inputs $\mathbf{x}, \Delta$, respectively. Then they receive $M_{\mathbf{x}}, K_{\mathbf{x}}$, where $K_{\mathbf{x}} = \Delta \cdot \mathbf{x} + M_{\mathbf{x}}$ over $\mathcal{R}$.
3. Let $r_{\mathbf{x}}$ be pseudorandom shift factors over $\mathcal{R}_p$. The two parties first re-randomize $[\Delta \cdot \mathbf{x}]$ by setting

$$K_{\mathbf{x}} \leftarrow (K_{\mathbf{x}} + r_{\mathbf{x}} \mod p), \quad M_{\mathbf{x}} \leftarrow (M_{\mathbf{x}} + r_{\mathbf{x}} \mod p),$$

   respectively. By the lifting lemma, $K_{\mathbf{x}} = \Delta \cdot \mathbf{x} + M_{\mathbf{x}}$ still holds over $\mathcal{R}$.
4. For each gate G in the circuit with respect to input wire values x, y,
   - If G is an addition gate, let $r_z \in \mathcal{R}_p$ be a pseudorandom "shift factor". Then $\mathcal{P}$ computes $z := x + y$, $M_z := M_x + M_y + r_z \mod p$, while $\mathcal{V}$ computes $K_z := K_x + K_y + r_z \mod p$.
   - If G is a multiplication gate, let $r_z \in \mathcal{R}_p$ and $r_{\mathsf{DDec}} \in \mathcal{R}_q$ be pseudorandom "shift factors". Then $\mathcal{P}$ computes

$$M_z \leftarrow \mathsf{DDec}\big(aM_xM_y + b(xM_y + yM_x) + cxy + r_{\mathsf{DDec}} \mod q\big) + r_z \mod p,$$

     while $\mathcal{V}$ computes $K_z \leftarrow \mathsf{DDec}(aK_xK_y + r_{\mathsf{DDec}} \mod q) + r_z \mod p$.

   In both cases, by the lifting lemma, it holds (with high probability) that $K_z = \Delta \cdot z + M_z$ over $\mathcal{R}$, where $K_z, M_z \in \mathcal{R}_p$.
5. For the single output wire ω, $\mathcal{P}$ opens M_ω to $\mathcal{V}$. $\mathcal{V}$ accepts if $K_\omega = M_\omega$. Otherwise, $\mathcal{V}$ aborts.

Theorem 4.4. *Assuming circular-power RingLWE (Definition 3.7) and secure PRFs, Π_{zkLWE} is secure in the $(\mathcal{F}_{\mathsf{hcVOLE}}, CRS)$-hybrid model.*

Proof. *The proof follows analogously to Theorem 4.2.* □

Remark 4.5. The above construction also works for the case $\mathcal{R} = \mathbb{Z}$. We consider a general ring analogue of LWE, and assume computations are over $\mathcal{R}$. Because $\mathcal{R} = \mathbb{Z}[X]/(X^N + 1)$ has the advantages of faster evaluation. Namely, one can apply Fast Fourier Transforms (FFTs) to compute multiplications over $\mathcal{R}$ and moreover if $\mathcal{C}$ is a data-parallel circuit over $\mathbb{Z}$, it can be converted into a smaller circuit $\mathcal{C}'$ over $\mathcal{R}$. Similarly, one can employ bit-decomposition and key-switching in Section 4.1, admitting a trade-off of the parameters and assumptions.

5 Extractable Succinct Half-Chosen VOLE

In this section, we give succinct half-chosen VOLE protocols in the CRS model, by adapting the semi-honest secure constructions of [ARS24, AMR25]. In particular, as we would plug them into our computational LPZK constructions in Section 4 to get dv-zkSNARKs, we pay a special attention to the extractability of succinct half-chosen VOLE. The half-chosen VOLE functionality is formalized in $\mathcal{F}_{\mathsf{hcVOLE}}$.

Functionality 4: $\mathcal{F}_{\mathsf{hcVOLE}}$

Let λ be the security parameter, $N > 2^\lambda$ be a modulus.
Setup: Upon receiving ($\mathtt{Setup}$) from the receiver $\mathrm{P_R}$,

1. Sample and record $\Delta \xleftarrow{\$} [1, 2^\lambda]$.
2. Send Δ to $\mathrm{P_R}$.

Gen: Upon receiving $(\mathsf{Gen}, \mathbf{x} \in \mathbb{Z}_N^L, \mathsf{sid})$ from the sender $\mathrm{P_S}$ with sid being a session identifier, proceeds as follows:

1. If $\mathrm{P_S}$ is honest, sample $M_{\mathbf{x}} \xleftarrow{\$} \mathbb{Z}_N^L$. Otherwise, receive $M_{\mathbf{x}} \in \mathbb{Z}_N^L$ from $\mathcal{A}$.
2. If $\mathrm{P_R}$ is honest, compute $K_{\mathbf{x}} := \Delta \cdot \mathbf{x} + M_{\mathbf{x}} \mod N$. Otherwise, receive $K_{\mathbf{x}} \in \mathbb{Z}_N^L$ from $\mathcal{A}$ and recompute $M_{\mathbf{x}} := -\Delta \cdot \mathbf{x} + K_{\mathbf{x}} \mod N$.
3. Store $(\mathsf{sid}, M_{\mathbf{x}}, K_{\mathbf{x}})$ and ignore future message $(\mathsf{Gen}, \mathbf{x}', \mathsf{sid})$.

Send: Upon receiving $(\mathsf{send}, \mathsf{sid})$ from $\mathcal{A}$, verify that there exists $(\mathsf{sid}, \mathbf{x}, M_{\mathbf{x}}, K_{\mathbf{x}})$. Next, send $(\mathsf{sid}, M_{\mathbf{x}})$ to $\mathrm{P_S}$ and $(\mathsf{sid}, K_{\mathbf{x}})$ to $\mathrm{P_R}$, and ignore future message $(\mathsf{send}, \mathsf{sid})$.
Query: If $\mathrm{P_S}$ is corrupted, upon receiving $(\mathtt{Query}, \Delta')$, return $\mathtt{true}$ if $\Delta' = \Delta$, otherwise return $\mathtt{false}$ to $\mathcal{A}$ and send $\mathtt{abort}$ to $\mathrm{P_R}$.

5.1 Succinct Half-Chosen VOLE from [ARS24]

We start with the Paillier-based succinct half-chosen VOLE protocol in [ARS24], which has one simultaneous round, i.e., the sender and the receiver only exchanges two short messages. To fit the context of zkSNARKs, we explicitly include the receiver's message in a common reference string crs, and show that it original offers a "partial extraction" property, which is not the case for most polynomial commitment schemes (PCS)[8]. By viewing half-chosen VOLE as a commitment scheme and adding a opening procedure, we prove the desired extraction property.

Recap of [ARS24]. At a high-level, the idea is to first translate a scalar multiplication of Δ (held by $\mathrm{P_R}$) and $\mathbf{x}$ (held by $\mathrm{P_S}$) into a (near-square) matrix multiplication $Y \cdot X$, where $Y = \Delta \cdot I_n$, and entries of X agree with $\mathbf{x}$'s. Then they exchange two messages: a row-wise encryption of Y, and a column-wise hashing of X, where $\mathrm{P_R}$ can operate on the column hashes of X to get $[YX]_1$, and $\mathrm{P_S}$ can operate on the row encryptions of Y to get $[YX]_0$, such that $[YX]_1 = [YX]_0 + YX$.

In more detail, let $g_0, g_1, \ldots, g_n$ be pre-defined random elements in a Paillier group $\mathbb{Z}_{N^2}^*$. Each column of X, say $(x_1, \ldots, x_n)$, is hashed via a Pedersen commitment [Ped91], i.e.,

$$d = g_0^v \cdot \prod_{i=1}^{n} g_i^{x_i},$$

[8] Informally, a PCS consists of two phase, commitment phase and evaluation phase, where typically the extractor is constructed by utilizing the evaluation phase. By "partial extraction", one can think of that the extractor can extract part of the coefficients of the committed polynomial from the commitment phase.

where v is a random mask. Meanwhile, the encryption of a row $(y_1, \ldots, y_n)$ of Y, consists of ElGamal-like ciphertexts

$$\left(g_0^w,\, g_1^w \cdot (1+N)^{y_1},\, \ldots,\, g_n^w \cdot (1+N)^{y_n}\right),$$

where w is a random private key held by the receiver. To compute the inner-product of $(x_1, \ldots, x_n)$ and $(y_1, \ldots, y_n)$, the key observation is the following:

$$\frac{(g_0^w)^v \cdot \prod_{i=1}^n (g_i^w \cdot (1+N)^{y_i})^{x_i}}{d^w} = \frac{g_0^{wv} \cdot \prod_{i=1}^n (g_i^{x_i w} \cdot (1+N)^{x_i y_i})}{(g_0^v \cdot \prod_{i=1}^n g_i^{x_i})^w} = (1+N)^{\sum_{i=1}^n x_i y_i}.$$

Namely, the two parties can compute a multiplicative secret sharing of $(1+N)^{\sum_{i=1}^n x_i y_i}$, which by DDLog in Paillier groups, can be converted into subtractive sharings $[\sum_{i=1}^n x_i y_i]$. By reusing the encryptions and hashes for each row-column pair, the parties obtain the subtractive sharing of the entire matrix $YX = \Delta \cdot X$, i.e., half-chosen VOLE, as desired. Suppose $\mathbf{x}$ has length L, by choosing $n = L^{1/3}$, both the hashes and the encryptions are of size $O(L^{2/3})$.

Linear Size crs *with One Hash.* Since we will include the receiver's message in the crs, it has not to be sublinear in L. That is, a *half-succinct* half-chosen VOLE suffices for our case. By setting $n = L$ in the above, it is readily to see that the sender only needs to send one hash value, however the crs would have size $O(L^2)$.

We show that the size of crs can be optimized to $O(L)$, by introducing an n-ary power DDH assumption Definition B.1 as in [ARS24]. The high-level idea is to make the private keys as well as pre-sampled elements g_i correlated, so that the ciphertexts can be compressed. Specifically, there are two modifications:

1. For each $i \in [0, n]$, $g_i = g^{\alpha^i}$, where α is a random secret exponent held by $\mathrm{P_R}$ and g is a pre-sampled random element in $\mathbb{Z}_{N^2}^*$.
2. For each row $j \in [1, n]$, $w_j = w \cdot \alpha^{n-j}$, where w is a random private key held by $\mathrm{P_R}$.

Now the ciphertexts of the matrix Y can be represented as

$$\begin{pmatrix} g_0^{w_1} & g_1^{w_1}(1+N)^{y_{1,1}} & \cdots & g_n^{w_1}(1+N)^{y_{1,n}} \\ g_0^{w_2} & g_1^{w_2}(1+N)^{y_{2,1}} & \cdots & g_n^{w_2}(1+N)^{y_{2,n}} \\ \vdots & \vdots & \vdots & \vdots \\ g_0^{w_n} & g_1^{w_n}(1+N)^{y_{n,1}} & \cdots & g_n^{w_n}(1+N)^{y_{n,n}} \end{pmatrix} = \begin{pmatrix} g^{w\cdot\alpha^{n-1}} & g^{w\cdot\alpha^n}(1+N)^{\Delta} & \cdots & g^{w\cdot\alpha^{2n-1}} \\ g^{w\cdot\alpha^{n-2}} & g^{w\cdot\alpha^{n-1}} & \cdots & g^{w\cdot\alpha^{2n-2}} \\ \vdots & \vdots & \vdots & \vdots \\ g^{w} & g^{w\cdot\alpha} & \cdots & g^{w\cdot\alpha^{n}}(1+N)^{\Delta} \end{pmatrix},$$

which is a Toeplitz matrix with only $2n$ distinct values in the entries. Hence, it suffices to let crs include

$$\left(g^w, \ldots, g^{w\cdot\alpha^{n-1}}, g^{w\cdot\alpha^n} \cdot (1+N)^{\Delta}, g^{w\cdot\alpha^{n+1}}, \ldots, g^{w\cdot\alpha^{2n-1}}\right),$$

leading to a linear size crs for $n = L$.

On the extractability. The privacy of the above construction crucially relies on that neither the sender nor the receiver holds $\varphi(N)$. Though, a receiver holding $\varphi(N)$ still cannot recover entire $(x_{1,j}, \ldots, x_{n,j})$ from its hash, recovering a linear relation on $\mathbf{x}$ is possible. We observe that an extractor can utilize it to extract part of the inputs.

Suppose $L = n \times \ell$, we construct an extractor that extracts one row (w.l.o.g., the first row) of X in the following way:

- First run a trapdoor setup algorithm $\mathsf{TrapSetup}(1^\lambda)$:
  1. Sample $(N, p, q) \leftarrow \mathsf{GenPQ}(1^\lambda)$ and compute d such that $d = 0 \pmod{\varphi(N)}$ and $d = 1 \pmod{N}$. //d is the decryption key of Paillier scheme.
  2. Sample $(h_0, h_1, \ldots, h_n) \leftarrow (\mathbb{Z}^*_{N^2})^{n+1}$ and compute $(g_0, g_1^*, g_2, \ldots, g_n) \leftarrow (h_0^N, h_1^N \cdot (1+N), h_2^N, \ldots, h_n^N)$.
  3. Output the common reference string $\mathsf{crs}^* := (N, g_0, g_1^*, g_2, \ldots, g_n)$ and the trapdoor $\mathsf{td} := d$. //here for simplicity we omit the receiver's message.
- Second, on input hashes $d_j = g_0^{v_j} \cdot \prod_{i=1}^n g_i^{x_{i,j}}$ for each $j \in [1, \ell]$, run $\mathsf{Extract}(\mathsf{td}, d_j)$:
  1. outputs $x^*_{1,j} = \frac{(d_j^{\mathsf{td}} \bmod N^2) - 1}{N} \in \mathbb{Z}_N$.

However, the extractor cannot extract all rows of X by inputting the malicious sender with different crs^*, because X were not consistent in each invocation. To fully extract the inputs, we come up with a simple technique that works for any semi-honestly secure half-chosen VOLE protocol.

The idea is to view half-chosen VOLE as a linearly-homomorphic commitment to the sender's input X. Then, the extractability can be obtained by adding a zero-knowledge opening procedure, which essentially proves the knowledge of the committed X. Since VOLE has linear homomorphism, a standard random-linear combination check suffices for the opening. To make the opening zero-knowledge, we let the sender pad a random value to his input, and therefore $L + 1 = n \times \ell$ VOLE correlations will be generated.

Concretely, the two parties proceed like a Σ-protocol:

1. $\mathrm{P_S}$ and $\mathrm{P_R}$ run a half-chosen VOLE protocol to obtain $[(r \parallel \mathbf{x}) \cdot \Delta]$, where r is a random mask sampled by $\mathrm{P_S}$.
2. $\mathrm{P_R}$ samples random coefficients $\chi_1, \ldots, \chi_L$, and send them to $\mathrm{P_S}$.
3. $\mathrm{P_S}$ computes $x' = r + \sum_{i=1}^L \chi_i x_i$, and $M' = [r \cdot \Delta]_0 + \sum_{i=1}^L \chi_i [x_i \cdot \Delta]_0$, and sends (x', M') to $\mathrm{P_R}$.
4. $\mathrm{P_R}$ computes $K' = [r \cdot \Delta]_1 + \sum_{i=1}^L \chi_i [x_i \cdot \Delta]_1$, and checks $K' \stackrel{?}{=} x' \cdot \Delta + M'$.

Now, $\mathbf{x}$ can be extracted by asking the malicious prover to repeatedly open for L times. Moreover, the extraction property will be preserved if we perform Fiat-Shamir transform to eliminate the additional round.

For completeness, we give the extractable half-chosen VOLE protocol in Π_{hcVOLE} based on standard DCR assumptions. One can easily extend it into a non-interactive extractable protocol with a linear size CRS and $O(1)$ communication by the above discussions. Moreover, extractable half-chosen VOLE protocols with a linear size CRS and $O(1)$ communications can be also realized with

other assumptions, e.g., L-ary power Ring-LWE, by extending the corresponding constructions in [ARS24]. We omit re-creating such constructions here, and refer readers to [ARS24] for more details.

Protocol 5: Π_{hcVOLE}

Let $\mathbb{Z}_{N^2}^*$ be a Paillier group. Let n, l, L be positive integers such that $L+1 = n \cdot \ell$. Here $L = n \cdot l - 1$ is referred to as the input length.
$\mathsf{crs} := \big(\{g_\kappa \in \mathbb{Z}_{N^2}^* \mid \kappa \in [0,n]\}, \{e_{i,\kappa} \in \mathbb{Z}_{N^2}^* \mid i \in [1,n], \kappa \in [0,n]\}\big)$, such that $e_{i,0} := g_0^{w_i}$ and $e_{i,\kappa} := g_\kappa^{w_i} \cdot (1+N)^{y_{i,\kappa}}$ for $i, \kappa \in [1,n]$, where $w_i \xleftarrow{\$} \mathbb{Z}_{N^2}$ and $(y_{i,j})_{i,j\in[1,n]} := Y = \Delta \cdot I_n \in \mathbb{Z}_N^{n\times n}$.
Setup: The receiver $\mathrm{P_R}$ is additionally given $(w_1, \ldots, w_n)$ and Δ, i.e., Y.
Commit: The sender $\mathrm{P_S}$ inputs a length-L vector over $\mathbb{Z}_N$.

1. The sender $\mathrm{P_S}$ appends a random value in $\mathbb{Z}_N$ to his input and re-arranges it as $X := (x_{\kappa,j})_{\kappa\in[1,n],j\in[1,l]} \in \mathbb{Z}_N^{n\times l}$. Then for each $j \in [1,l]$, $\mathrm{P_S}$ samples $v_j \xleftarrow{\$} \mathbb{Z}_{N^2}$ and computes

$$d_j := g_0^{v_j} \cdot \prod_{\kappa=1}^{n} g_\kappa^{x_{\kappa,j}} \in \mathbb{Z}_{N^2}^*.$$

   $\mathrm{P_S}$ retrieves $\{(e_{i,0}, \ldots, e_{i,n})\}_{i\in[1,n]}$ and computes

$$[z_{i,j}]_0 \leftarrow -\mathsf{DDLog}(e_{i,0}^{v_j} \cdot \prod_{\kappa=1}^{n} e_{i,\kappa}^{x_{\kappa,j}}),$$

   for $i \in [1,n]$, $j \in [1,l]$. $\mathrm{P_S}$ sends $\{d_j\}_{j\in[1,l]}$ to $\mathrm{P_R}$.
2. Upon receiving $\{d_j\}_{j\in[1,l]}$ from $\mathrm{P_S}$, the receiver $\mathrm{P_R}$ computes

$$[z_{i,j}]_1 \leftarrow -\mathsf{DDLog}(d_j^{w_i}),$$

   for $i \in [1,n]$, $j \in [1,l]$.
3. $\mathrm{P_S}$ and $\mathrm{P_R}$ output $[Z] = [\Delta \cdot X]$, where $Z := (z_{i,j})_{i\in[1,n],j\in[1,l]} = YX \in \mathbb{Z}_N^{n\times\ell}$. This is essentially a commitment of X, denoted by $[\![X]\!]$.

Open:

1. $\mathrm{P_R}$ sends random coefficients $\chi_{i,j} \xleftarrow{\$} \mathbb{Z}_N$ for $i \in [1,n], j \in [1,l]$ to $\mathrm{P_S}$. Then, $\mathrm{P_R}$ computes $K' := \sum_{i\in[1,n]} \sum_{j\in[1,l]} \chi_{i,j} \cdot [z_{i,j}]_1$. //Actually, $n\ell - 1$ coefficients are sent, where we let $\chi_{1,1}$ (corresponding to the random mask) be always 1.
2. Upon receiving $\chi_{i,j}$, $\mathrm{P_S}$ computes $M' := \sum_{i\in[1,n]} \sum_{j\in[1,l]} \cdot \chi_{i,j}[z_{i,j}]_0$ and $x' := \sum_{i\in[1,n]} \sum_{j\in[1,l]} \chi_{i,j} \cdot X_{i,j}$. Then $\mathrm{P_S}$ sends (M', x') to $\mathrm{P_R}$.
3. Upon receiving (M', x') from $\mathrm{P_S}$, $\mathrm{P_R}$ checks $K' \stackrel{?}{=} \Delta \cdot x' + M'$. $\mathrm{P_R}$ accepts if the equality holds, otherwise rejects.

Theorem 5.1. *Π_{hcVOLE} securely realizes $\mathcal{F}_{\mathsf{hcVOLE}}$ in the CRS model, assuming DCR.*

Proof. Correctness directly follows by design and we defer a detailed proof to Appendix D.2. □

By Fiat-Shamir transformation, Π_{hcVOLE} can be made non-interactive in the random oracle model. Then plugged into Theorem 4.2 (Π_{zkPai}), we obtain the following results:

Corollary 5.2. *For proving the satisfiability of a B-bounded circuit $\mathcal{C} : \mathbb{Z}^n \to \mathbb{Z}$,*

1. *Assuming DCR over Paillier groups and circular-power DDH in Paillier groups, there exists a dv-zkSNARK with* crs *size $O_\lambda(n^{2/3})$, proof size $O_\lambda(n^{2/3})$, prover time $O_\lambda(n^{4/3} + |\mathcal{C}|)$, verifier time $O_\lambda(n + |\mathcal{C}|)$.*
2. *Assuming n-ary power DDH and circular-power DDH in Paillier groups, there exists a dv-zkSNARK with* crs *size $O_\lambda(n)$, proof size $O_\lambda(1)$, prover time $O_\lambda(n^2 + |\mathcal{C}|)$, verifier time $O_\lambda(n + |\mathcal{C}|)$.*

5.2 Succinct Half-Chosen VOLE from [AMR25]

In Section 5.1, we add extractability to the half-chosen VOLE protocol based on Paillier groups [ARS24], via a zero-knowledge opening procedure. As the approach only makes a black-box use of the underlying half-chosen VOLE protocols, we can apply to the state-of-the-art construction from [AMR25], where both the communications and crs size are made logarithmic. We briefly summarize the results of [AMR25] in the following:

Theorem 5.3. *[AMR25, Theorem 1.1] Assuming the hardness of LWE, there exists a protocol (satisfying receiver privacy) with one simultaneous round for computing the tensor product $\mathbf{x} \otimes \mathbf{y}$, where $\mathbf{x} \in \mathbb{Z}_q^L$ (held by the sender) and $\mathbf{y} \in \mathbb{Z}_q^\ell$ (held by the receiver), with communication complexity $\ell \cdot \mathsf{poly}(\lambda) + \mathsf{poly}(\lambda, \log L)$ and a* crs *of size* $\mathsf{poly}(\lambda, \log L)$.

By setting $\ell = 1$, one immediately gets a succinct half-chosen VOLE protocol. Moreover, we can view the receiver's message as a part of the crs, because the protocol has only one simultaneous round. However, the protocol does not offer sender privacy, we bypass this issue by using a PCG for random VOLE [SGRR19], which is non-interactive and has logarithmic communication complexity. In more detail, to generate $[\mathbf{x} \cdot \Delta]$, random VOLE correlations $[\mathbf{a} \cdot \Delta]$ are first produced by PCG, then it suffices to use the half-chosen VOLE protocol to obtain $[(\mathbf{a} + \mathbf{x}) \cdot \Delta]$. Finally, by adding a zero-knowledge opening procedure (and Fiat-Shamir transform) for extraction, we end up with the following result:

Corollary 5.4. *Assuming the hardness of LWE and a secure PCG for VOLE, there exists a protocol securely realizes $\mathcal{F}_{\mathsf{hcVOLE}}$ in the CRS model and the random oracle, whose* crs *size and communication complexity are* $\mathsf{poly}(\lambda, \log L)$*, where L is the VOLE length.*

Instantiating $\mathcal{F}_{\mathsf{hcVOLE}}$ with the construction of Corollary 5.4 for the computational LPZK construction in Theorem 4.4 (Π_{zkLWE}), we obtain fully succinct dv-zkSNARKs. The results are summarized as follows.

Corollary 5.5. *For proving the satisfiability of a B-bounded circuit $\mathcal{C} : \mathbb{Z}^n \to \mathbb{Z}$,*

1. *Assuming LWE, circular-power LWE and secure PCGs for VOLE, there exists a dv-zkSNARK with* crs *size $O_\lambda(\log n)$, proof size $O_\lambda(\log n)$, prover time $O_\lambda(n \log n + |\mathcal{C}|)$, verifier time $O_\lambda(n + |\mathcal{C}|)$.*

Acknowledgment. The authors would like to thank anonymous reviewers at ASIACRYPT 2025. The work was supported in part by the National Natural Science Foundation of China under Grants 12361141818, 12426302, 12031011 and 12271084, the National Key Research and Development Program of China under Grants 2023YFE0123900 and 2022YFA1004900, and the Natural Science Foundation of Shanghai under the 2024 Shanghai Action Plan for Science, Technology, and Innovation Grant 24BC3200700.

References

ADI25. Gal Arnon, Jesko Dujmovic, and Yuval Ishai. Designated-verifier snargs with one group element. In *Advances in Cryptology - CRYPTO 2025*, volume 16006 of *LNCS*, pages 192–224. Springer, 2025. 36

ADOS22. Damiano Abram, Ivan Damgård, Claudio Orlandi, and Peter Scholl. An algebraic framework for silent preprocessing with trustless setup and active security. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology - CRYPTO 2022*, volume 13510 of *LNCS*, pages 421–452. Springer, 2022. 11, 13, 37

AFLN24. Martin R. Albrecht, Giacomo Fenzi, Oleksandra Lapiha, and Ngoc Khanh Nguyen. SLAP: succinct lattice-based polynomial commitments from standard assumptions. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology - EUROCRYPT 2024*, volume 14656 of *LNCS*, pages 90–119. Springer, 2024. 36

AHIV17. Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. Ligero: Lightweight sublinear arguments without a trusted setup. In Bhavani Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *CCS 2017*, pages 2087–2104. ACM, 2017. 3

AMR25. Damiano Abram, Giulio Malavolta, and Lawrence Roy. Succinct oblivious tensor evaluation and applications: Adaptively-secure laconic function evaluation and trapdoor hashing for all circuits. In Michal Koucký and Nikhil Bansal, editors, *STOC 2025*, pages 1875–1886. ACM, 2025. 2, 6, 7, 18, 23, 28

ARS24. Damiano Abram, Lawrence Roy, and Peter Scholl. Succinct homomorphic secret sharing. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology - EUROCRYPT 2024*, volume 14656 of *LNCS*, pages 301–330. Springer, 2024. 2, 4, 5, 7, 10, 14, 18, 23, 24, 25, 27, 28

BBB+18. Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Gregory Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE Symposium on Security and Privacy, SP 2018*, pages 315–334. IEEE Computer Society, 2018. 36

BBC+19. Dan Boneh, Elette Boyle, Henry Corrigan-Gibbs, Niv Gilboa, and Yuval Ishai. Zero-knowledge proofs on secret-shared data via fully linear pcps. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology - CRYPTO 2019*, volume 11694 of *LNCS*, pages 67–97. Springer, 2019. 2

BBdSG+23. Carsten Baum, Lennart Braun, Cyprien Delpech de Saint Guilhem, Michael Klooß, Emmanuela Orsini, Lawrence Roy, and Peter Scholl. Publicly verifiable zero-knowledge and post-quantum signatures from vole-in-the-head. In *CRYPTO 2023*, volume 14085 of *LNCS*, pages 581–615. Springer, 2023. 4, 35

BBHR18. Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast reed-solomon interactive oracle proofs of proximity. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018*, volume 107 of *LIPIcs*, pages 14:1–14:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. 35

BBM+24. Carsten Baum, Ward Beullens, Shibam Mukherjee, Emmanuela Orsini, Sebastian Ramacher, Christian Rechberger, Lawrence Roy, and Peter Scholl. One tree to rule them all: Optimizing GGM trees and owfs for post-quantum signatures. In Kai-Min Chung and Yu Sasaki, editors, *Advances in Cryptology - ASIACRYPT 2024*, volume 15484 of *LNCS*, pages 463–493. Springer, 2024. 7

BC24. Dan Boneh and Binyi Chen. Latticefold: A lattice-based folding scheme and its applications to succinct proof systems. *IACR Cryptol. ePrint Arch.*, page 257, 2024. 36

BCC+16. Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Jens Groth, and Christophe Petit. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016*, volume 9666 of *LNCS*, pages 327–357. Springer, 2016. 36

BCG+19a. Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, Peter Rindal, and Peter Scholl. Efficient two-round OT extension and silent non-interactive secure computation. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *CCS 2019*, pages 291–308. ACM, 2019. 6

BCG+19b. Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, and Peter Scholl. Efficient pseudorandom correlation generators: Silent OT extension and more. In *CRYPTO 2019*, volume 11694 of *LNCS*, pages 489–518. Springer, 2019. 6, 35

BCGI18. Elette Boyle, Geoffroy Couteau, Niv Gilboa, and Yuval Ishai. Compressing vector OLE. In *CCS 2018*, pages 896–912. ACM, 2018. 6, 35

BCI+22. Nir Bitansky, Alessandro Chiesa, Yuval Ishai, Rafail Ostrovsky, and Omer Paneth. Succinct non-interactive arguments via linear interactive proofs. *J. Cryptol.*, 35(3):15, 2022. 14, 15, 36

BCM23. Elette Boyle, Geoffroy Couteau, and Pierre Meyer. Sublinear-communication secure multiparty computation does not require FHE. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology - EUROCRYPT 2023*, volume 14005 of *LNCS*, pages 159–189. Springer, 2023. 3

BDSW23. Carsten Baum, Samuel Dittmer, Peter Scholl, and Xiao Wang. Sok: vector ole-based zero-knowledge protocols. *Des. Codes Cryptogr.*, 91(11):3527–3561, 2023. 2, 35

BFLS91. László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In Cris Koutsougeras and Jeffrey Scott Vitter, editors, *STOC 1991*, pages 21–31. ACM, 1991. 3

BGI16. Elette Boyle, Niv Gilboa, and Yuval Ishai. Breaking the circuit size barrier for secure computation under DDH. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016*, volume 9814 of *LNCS*, pages 509–539. Springer, 2016. 10, 35, 38

BGI17. Elette Boyle, Niv Gilboa, and Yuval Ishai. Group-based secure computation: Optimizing rounds, communication, and computation. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology - EUROCRYPT 2017*, volume 10211 of *LNCS*, pages 163–193, 2017. 10, 38

BGI+18. Elette Boyle, Niv Gilboa, Yuval Ishai, Huijia Lin, and Stefano Tessaro. Foundations of homomorphic secret sharing. In Anna R. Karlin, editor, *9th Innovations in Theoretical Computer Science Conference, ITCS 2018*, volume 94 of *LIPIcs*, pages 21:1–21:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. 10, 38

BHI+24. Nir Bitansky, Prahladh Harsha, Yuval Ishai, Ron D. Rothblum, and David J. Wu. Dot-product proofs and their applications. In *FOCS 2024*, pages 806–825. IEEE, 2024. 36

BIOW20. Ohad Barta, Yuval Ishai, Rafail Ostrovsky, and David J. Wu. On succinct arguments and witness encryption from groups. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology - CRYPTO 2020*, volume 12170 of *LNCS*, pages 776–806. Springer, 2020. 36

BKS19. Elette Boyle, Lisa Kohl, and Peter Scholl. Homomorphic secret sharing from lattices without FHE. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2019*, volume 11477 of *LNCS*, pages 3–33. Springer, 2019. 10, 21, 22

BMRS21. Carsten Baum, Alex J. Malozemoff, Marc B. Rosen, and Peter Scholl. Mac'n'cheese: Zero-knowledge proofs for boolean and arithmetic circuits with nested disjunctions. In *CRYPTO 2021*, volume 12828 of *LNCS*, pages 92–122. Springer, 2021. 35

BS23. Ward Beullens and Gregor Seiler. Labrador: Compact proofs for R1CS from module-sis. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology - CRYPTO 2023*, volume 14085 of *LNCS*, pages 518–548. Springer, 2023. 4, 36

CK24. Geoffroy Couteau and Naman Kumar. 10-party sublinear secure computation from standard assumptions. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology - CRYPTO 2024*, volume 14928 of *LNCS*, pages 39–73. Springer, 2024. 3

CMNW24. Valerio Cini, Giulio Malavolta, Ngoc Khanh Nguyen, and Hoeteck Wee. Polynomial commitments from lattices: Post-quantum security, fast verification and transparent setup. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology - CRYPTO 2024*, volume 14929 of *LNCS*, pages 207–242. Springer, 2024. 36

DFGK14. George Danezis, Cédric Fournet, Jens Groth, and Markulf Kohlweiss. Square span programs with applications to succinct NIZK arguments.

In Palash Sarkar and Tetsu Iwata, editors, *Advances in Cryptology - ASIACRYPT 2014*, volume 8873 of *LNCS*, pages 532–550. Springer, 2014. 36

DH24. Changchang Ding and Yan Huang. Phecda: Post-quantum transparent zksnarks from improved polynomial commitment and vole-in-the-head with application in publicly verifiable aes. In *2025 IEEE Symposium on Security and Privacy (SP)*, pages 55–55. IEEE Computer Society, 2024. 3, 7, 35

DHRW16. Yevgeniy Dodis, Shai Halevi, Ron D. Rothblum, and Daniel Wichs. Spooky encryption and its applications. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016*, volume 9816 of *LNCS*, pages 93–122. Springer, 2016. 21

DIJL23. Quang Dao, Yuval Ishai, Aayush Jain, and Huijia Lin. Multi-party homomorphic secret sharing and sublinear MPC from sparse LPN. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology - CRYPTO 2023*, volume 14082 of *LNCS*, pages 315–348. Springer, 2023. 3

DILO22. Samuel Dittmer, Yuval Ishai, Steve Lu, and Rafail Ostrovsky. Improving line-point zero knowledge: Two multiplications for the price of one. In *CCS 2022*, pages 829–841. ACM, 2022. 35

DIO21. Samuel Dittmer, Yuval Ishai, and Rafail Ostrovsky. Line-point zero knowledge and its applications. In *ITC 2021*, volume 199 of *LIPIcs*, pages 5:1–5:24. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. 4, 5, 6, 8, 35

DJ01. Ivan Damgård and Mads Jurik. A generalisation, a simplification and some applications of paillier's probabilistic public-key system. In Kwangjo Kim, editor, *PKC 2001*, volume 1992 of *LNCS*, pages 119–136. Springer, 2001. 2, 9, 37

DMS24. Michel Dellepere, Pratyush Mishra, and Alireza Shirzad. Garuda and pari: Faster and smaller snarks via equifficient polynomial commitments. *IACR Cryptol. ePrint Arch.*, page 1245, 2024. 36

GGPR13. Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic span programs and succinct nizks without pcps. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology - EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 626–645. Springer, 2013. 36

GKR08. Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. Delegating computation: interactive proofs for muggles. In Cynthia Dwork, editor, *STOC 2008*, pages 113–122. ACM, 2008. 3, 7, 35

GLS+23. Alexander Golovnev, Jonathan Lee, Srinath T. V. Setty, Justin Thaler, and Riad S. Wahby. Brakedown: Linear-time and field-agnostic snarks for R1CS. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023*, volume 14082 of *LNCS*, pages 193–226. Springer, 2023. 4

GMR89. Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1):186–208, 1989. 2

GMW87. Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In Alfred V. Aho, editor, *STOC 1987*, pages 218–229. ACM, 1987. 2

GNS23. Chaya Ganesh, Anca Nitulescu, and Eduardo Soria-Vazquez. Rinocchio: Snarks for ring arithmetic. *J. Cryptol.*, 36(4):41, 2023. 4

Gro10. Jens Groth. Short pairing-based non-interactive zero-knowledge arguments. In Masayuki Abe, editor, *Advances in Cryptology - ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 321–340. Springer, 2010. 36

Gro16. Jens Groth. On the size of pairing-based non-interactive arguments. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016*, volume 9666 of *LNCS*, pages 305–326. Springer, 2016. 4, 36

HSS24. Intak Hwang, Jinyeong Seo, and Yongsoo Song. Concretely efficient lattice-based polynomial commitment from standard assumptions. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology - CRYPTO 2024*, volume 14929 of *LNCS*, pages 414–448. Springer, 2024. 36

IKOS07. Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Zero-knowledge from secure multiparty computation. In *STOC 2007*, pages 21–30. ACM, 2007. 2

ILL24. Yuval Ishai, Hanjun Li, and Huijia Lin. Succinct homomorphic macs from groups and applications. *IACR Cryptol. ePrint Arch.*, page 2073, 2024. 3, 4, 8, 10, 11, 17, 18, 20, 35, 36

ILL25. Yuval Ishai, Hanjun Li, and Huijia Lin. A unified framework for succinct garbling from homomorphic secret sharing. In *Advances in Cryptology - CRYPTO 2025*, volume 16003 of *LNCS*, pages 390–425. Springer, 2025. 3, 4, 8, 10, 11, 14, 17, 18, 35

JKO13. Marek Jawurek, Florian Kerschbaum, and Claudio Orlandi. Zero-knowledge using garbled circuits: how to prove non-algebraic statements efficiently. In *CCS 2013*, pages 955–966. ACM, 2013. 2

KKW18. Jonathan Katz, Vladimir Kolesnikov, and Xiao Wang. Improved non-interactive zero knowledge with applications to post-quantum signatures. In *CCS 2018*, pages 525–537. ACM, 2018. 4

LFKN92. Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *J. ACM*, 39(4):859–868, 1992. 3, 7, 35

LHLA25. Mengling Liu, Yang Heng, Xingye Lu, and Man Ho Au. Jesseq: Efficient zero-knowledge proofs for circuits over any field. In Marina Blanton, William Enck, and Cristina Nita-Rotaru, editors, *IEEE Symposium on Security and Privacy, SP 2025*, pages 3402–3420. IEEE, 2025. 35

Lip13. Helger Lipmaa. Succinct non-interactive zero knowledge arguments from span programs and linear error-correcting codes. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology - ASIACRYPT 2013*, volume 8269 of *LNCS*, pages 41–60. Springer, 2013. 36

Lip24. Helger Lipmaa. Polymath: Groth16 is not the limit. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology - CRYPTO 2024*, volume 14929 of *LNCS*, pages 170–206. Springer, 2024. 36

LXY24. Fuchun Lin, Chaoping Xing, and Yizhou Yao. Interactive line-point zero-knowledge with sublinear communication and linear computation. In Kai-Min Chung and Yu Sasaki, editors, *Advances in Cryptology - ASIACRYPT 2024*, volume 15488 of *LNCS*, pages 337–366. Springer, 2024. 4, 7, 8, 35

LXYY25. Zhe Li, Chaoping Xing, Yizhou Yao, and Chen Yuan. On the power of sumcheck in secure multiparty computation. *IACR Cryptol. ePrint Arch.*, 2025. 2

MORS24. Pierre Meyer, Claudio Orlandi, Lawrence Roy, and Peter Scholl. Rate-1 arithmetic garbling from homomorphic secret sharing. In Elette Boyle and Mohammad Mahmoody, editors, *Theory of Cryptography - 22nd*

International Conference, TCC 2024, volume 15367 of *LNCS*, pages 71–97. Springer, 2024. 3, 4, 8, 17, 35

MORS25. Pierre Meyer, Claudio Orlandi, Lawrence Roy, and Peter Scholl. Silent circuit relinearisation: Sublinear-size (boolean and arithmetic) garbled circuits from dcr. In *Advances in Cryptology - CRYPTO 2025*, volume 16003 of *LNCS*, pages 426–458. Springer, 2025. 3, 4, 8, 11, 17, 20, 35

NS24. Ngoc Khanh Nguyen and Gregor Seiler. Greyhound: Fast polynomial commitments from lattices. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology - CRYPTO 2024*, volume 14929 of *LNCS*, pages 243–275. Springer, 2024. 36

OSY21. Claudio Orlandi, Peter Scholl, and Sophia Yakoubov. The rise of paillier: Homomorphic secret sharing and public-key silent OT. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology - EUROCRYPT 2021*, volume 12696 of *LNCS*, pages 678–708. Springer, 2021. 9, 13, 17, 35, 40

Pai99. Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Jacques Stern, editor, *Advances in Cryptology - EUROCRYPT '99*, volume 1592 of *LNCS*, pages 223–238. Springer, 1999. 9, 17

Ped91. Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Advances in Cryptology - CRYPTO 1991*, volume 576 of *LNCS*, pages 129–140. Springer, 1991. 24

RS21. Lawrence Roy and Jaspal Singh. Large message homomorphic secret sharing from DCR and applications. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology - CRYPTO 2021*, volume 12827 of *LNCS*, pages 687–717. Springer, 2021. 9, 12, 13, 35

SGRR19. Phillipp Schoppmann, Adrià Gascón, Leonie Reichert, and Mariana Raykova. Distributed vector-ole: Improved constructions and implementation. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 1055–1072, 2019. 6, 28

WYKW21. Chenkai Weng, Kang Yang, Jonathan Katz, and Xiao Wang. Wolverine: Fast, scalable, and communication-efficient zero-knowledge proofs for boolean and arithmetic circuits. In *IEEE Symposium on Security and Privacy 2021*, pages 1074–1091. IEEE, 2021. 35

WYY+22. Chenkai Weng, Kang Yang, Zhaomin Yang, Xiang Xie, and Xiao Wang. Antman: Interactive zero-knowledge proofs with sublinear communication. In *CCS 2022*, pages 2901–2914. ACM, 2022. 7, 35

XZZ+19. Tiancheng Xie, Jiaheng Zhang, Yupeng Zhang, Charalampos Papamanthou, and Dawn Song. Libra: Succinct zero-knowledge proofs with optimal prover computation. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019*, volume 11694 of *LNCS*, pages 733–764. Springer, 2019. 4, 7

Yao86. Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *FOCS 1986*, pages 162–167. IEEE Computer Society, 1986. 2

YSWW21. Kang Yang, Pratik Sarkar, Chenkai Weng, and Xiao Wang. Quicksilver: Efficient and affordable zero-knowledge proofs for circuits and polynomials over any field. In *CCS 2021*, pages 2986–3001. ACM, 2021. 6, 35

YWL+20. Kang Yang, Chenkai Weng, Xiao Lan, Jiang Zhang, and Xiao Wang. Ferret: Fast extension for correlated OT with small communication. In

Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna, editors, *CCS 2020*, pages 1607–1626. ACM, 2020. 6

ZLW+21. Jiaheng Zhang, Tianyi Liu, Weijie Wang, Yinuo Zhang, Dawn Song, Xiang Xie, and Yupeng Zhang. Doubly efficient interactive proofs for general arithmetic circuits with linear prover time. In Yongdae Kim, Jong Kim, Giovanni Vigna, and Elaine Shi, editors, *CCS '21*, pages 159–177. ACM, 2021. 7

Supplementary Material

A Related Work

VOLE-ZK. The introduction of pseudorandom correlation generators (PCG) [BCGI18, BCG+19b] has emerged the study on VOLE-based ZK protocols, featuring a super fast proving procedure with low memory consumption. Typically, most VOLE-ZK protocols [WYKW21, BMRS21, DIO21, YSWW21, DILO22, LHLA25] have linear communication complexity, inherited from a "gate-by-gate" nature of MPC protocols. We refer the readers to [BDSW23] for more details. Compared to these works, we offer succinct communication complexity, though the concrete computational complexity is considerably higher.

Notably, several works [WYY+22, LXY24, DH24] have developed techniques to compress the communication complexity. Specifically, [WYY+22] utilizes linearly-homomorphic encryption and develops a polynomial authentication code (PAC) that groups a batch of VOLE, yielding communication $O(n + |\mathcal{C}|^{3/4})$. Besides, [LXY24] and [DH24] introduce the well-known Sumcheck techniques [LFKN92, GKR08] into the context and realize communication complexity $O(n + d \log |\mathcal{C}|)$, $O(\log^2 n + d \log |\mathcal{C}|)$ for layered circuits, respectively. In particular, the $O(\log^2 n)$ term communication of [DH24] is achieved by a polynomial commitment scheme [BBHR18], instead of a VOLE-based commitment. Hence [DH24]'s prover is not streaming, and have computation complexity $O(n \log n + |\mathcal{C}|)$. Compared to [DH24], our LWE-based construction achieves $O(\log n)$ communication with $O(n \log n + |\mathcal{C}|)$ computation, and maintains a streaming prover property.

Finally, it is worth to mention that by using VOLE-in-the-head (VOLEitH) compiler [BBdSG+23] and applying Fiat-Shamir transformation, public-coin, interactive VOLE-ZK protocols can be made publicly verifiable and non-interactive. While most VOLE-ZK protocols enjoy this property, our constructions are incompatible with VOLEitH due to the heavy use of HSS techniques.

Succinct Garbling Schemes & Algebraic Homomorphic MAC. Recent work [MORS24] presents a rate-1 arithmetic garbling scheme using techniques from homomorphic secret sharing (HSS) [BGI16, OSY21, RS21]. Follow-up works [ILL24, MORS25, ILL25] further achieve 1 bit communication per gate for general Boolean circuits and $O(1/\log\log|\mathcal{C}|)$ for layered Boolean circuits. In particular, [ILL24] proposes the notion of algebraic homomorphic MAC (aHMAC), a stepping stone to succinct garbling schemes, which is essentially a VOLE-based MAC that

allows performing any number of multiplications and additions. This allows us to distill techniques for non-interactive circuit evaluation in VOLE-ZK. At a high-level, our constructions differ from aHMACs mainly in the way generating VOLE correlations for the inputs, where [ILL24] simply assumes the existence of a trusted party and make the receiver's MAC key independent of the committed values to fit the scenario of garbling. Instead, we use extractable half-chosen VOLE protocols for succinct arguments.

SNARG with One Group Element. In generic bilinear groups, [Gro16] achieves a proof with two $\mathbb{G}_1$ elements and one $\mathbb{G}_2$ element in an asymmetric group setting, following earlier works [Gro10, GGPR13, BCI+22, Lip13, DFGK14]. [Lip24] improves this to three $\mathbb{G}_1$ elements and one field element, while [DMS24] further reduces the proof to two $\mathbb{G}_1$ elements and two field elements.

Several works [BCI+22, BCC+16, BBB+18, BIOW20, BHI+24] focus on optimizing proof length in pairing-free groups. The most succinct pairing-free SNARGs work only in the designated-verifier model. In the generic group model (GGM), [BIOW20] constructs a dv-SNARG with two group elements and inverse-polynomial soundness error. To achieve negligible soundness error, the proof requires a super-constant number of group elements. [ADI25] presents the first dv-SNARG with a constant number of group elements, a linear (in the circuit size) size CRS and negligible soundness error. Specifically, the proof consists of a single group element and an additive term relying on the soundness. In comparison, our construction (e.g., Π_{zkPai}, instantiated with Π_{hcVOLE} without opening) admits a dv-SNARG with two Paillier group elements (one for generating input VOLE correlations, and another for verifying the proof), a CRS size linear only in the witness size, and a negligible soundness error.

Lattice-based SNARKs. Among lattice-based proof systems, LaBRADOR [BS23] has extremely small proof size for R1CS with varying number of constraints and a linear time verifier. Very recently, there is a line of works [CMNW24, NS24, AFLN24, BC24, HSS24] building lattice-based SNARKs by first designing a lattice-based polynomial commitment scheme and compiling it with polynomial interactive oracle proofs (IOP). Most of these works rely on variants of SIS assumptions, which typically yield slack extraction. In contrast, for our LWE-based dv-zkSNARKs, the exact witness can be extracted.

B More preliminaries

Definition B.1 (n-ary Power-DDH Assumption in the NIDLS framework). *We say that the n-ary Power-DDH assumption holds in the NIDLS framework if the following two distributions are computationally indistinguishable:*

$$\left\{(\mathsf{pp}, \rho, g, g^{\alpha}, \ldots, g^{\alpha^n}) \mid \mathsf{pp} \leftarrow \mathsf{Gen}(1^\lambda), (g, \rho) \leftarrow \mathsf{Sample}(1^\lambda, \mathsf{pp}), \alpha \xleftarrow{\$} [1, \ell]\right\},$$

$$\left\{(\mathsf{pp}, \rho, g, g^{\alpha_1}, \ldots, g^{\alpha_n}) \mid \mathsf{pp} \leftarrow \mathsf{Gen}(1^\lambda), (g, \rho) \leftarrow \mathsf{Sample}(1^\lambda, \mathsf{pp}), (\alpha_1 \ldots \alpha_n) \xleftarrow{\$} [1, \ell]^n\right\}.$$

B.1 Assumptions in prime-order groups

Definition B.2 (Power-DDH Assumption). *We say that the Power-DDH assumption holds in prime-order groups if the following two distributions are computationally indistinguishable:*

$$\left\{\left((\mathbb{G},p,g),g^{\alpha},g^{\alpha^2}\right) \mid (\mathbb{G},p,g)\leftarrow \mathsf{Gen}(1^{\lambda}),\alpha \xleftarrow{\$}[1,p]\right\},$$
$$\left\{\left((\mathbb{G},p,g),g^{\alpha_1},g^{\alpha_2}\right) \mid (\mathbb{G},p,g)\leftarrow \mathsf{Gen}(1^{\lambda}),\alpha_1,\alpha_2 \xleftarrow{\$}[1,p]\right\}.$$

Definition B.3 (Circular-Power-DDH Assumption). *We say that the Circular-Power-DDH assumption holds in the NIDLS framework if the following two distributions are computationally indistinguishable:*

$$\left\{\left((\mathbb{G},p,g),g^{s},g^{\alpha},g^{s\alpha},g^{s^2\alpha+s}\right) \mid \begin{array}{l}\mathsf{pp}\leftarrow \mathsf{Gen}(1^{\lambda}),\\ s,\alpha \xleftarrow{\$}[1,p]\end{array}\right\},$$
$$\left\{\left((\mathbb{G},p,g),g^{s},g^{\alpha},g^{\beta},g^{\gamma}\right) \mid \begin{array}{l}\mathsf{pp}\leftarrow \mathsf{Gen}(1^{\lambda}),\\ s,\alpha,\beta,\gamma \xleftarrow{\$}[1,p]\end{array}\right\}.$$

B.2 Damgård-Jurik Cryptosystem [DJ01]

We present notions for the Damgård-Jurik cryptosystem, which is a generalization of the Paillier scheme and fits the NIDLS framework [ADOS22] as well.

Procedure 6: π_{DJ}

Let $\zeta \geq 1$ be a constant defined for the plaintext size. Define two maps:

$$\exp : \mathbb{Z}_{N^{\zeta}} \to 1+N\cdot\mathbb{Z}_{N^{\zeta+1}},\quad \exp(x):=\sum_{k=0}^{\zeta}\frac{(Nx)^k}{k!}.$$

$$\log : 1+N\cdot\mathbb{Z}_{N^{\zeta+1}} \to \mathbb{Z}_{N^{\zeta}},\quad \log(1+Nx):=\sum_{k=1}^{\zeta}\frac{(-N)^{k-1}x^k}{k}.$$

$\mathsf{KeyGen}(1^{\lambda})$:

1. Run $(N,p,q)\leftarrow \mathsf{RSA.Gen}$.
2. Compute $\varphi := (p-1)(q-1)$.
3. Output (N,φ).

$\mathsf{Enc}(x)$:

1. Sample $r \xleftarrow{\$} \mathbb{Z}^*_{N^{\zeta+1}}$.
2. Compute $c := r^{N^{\zeta}}\cdot\exp(x)$.
3. Output c.

$\mathsf{Eval}(c_1,c_2,\alpha)$:

1. Compute and output $c := c_1^{\alpha} \cdot c_2$. //Homomorphic evaluate $(x, y) \to (\alpha \cdot x + y)$

$\mathsf{Dec}(c)$:

1. Compute $\psi := \varphi^{-1} \in \mathbb{Z}_{N^\zeta}$
2. Compute and output $x := \psi \cdot \log(c^{\varphi})$.

In particular, the DDLog algorithm in a DJ group $\mathbb{Z}^*_{N^{\zeta+1}}$ is given by

$$\mathsf{DDLog}(h) = \log(\frac{h}{h \mod N}).$$

We define the key-dependent message (KDM) in the following.

Definition B.4 (KDM Security). *A public key encryption scheme is KDM secure w.r.t. a class of functions $\mathcal{F}$ if for every PPT adversary $\mathcal{A}$ there exists a negligible function $\mathsf{negl}(\lambda)$ such that for all $\lambda \in \mathbb{N}$:*

$$\left|\Pr[\mathrm{Exp}^{\mathcal{A},\mathcal{F},0}_{\mathrm{KDM}}(\lambda) = 1] - \Pr[\mathrm{Exp}^{\mathcal{A},\mathcal{F},1}_{\mathrm{KDM}}(\lambda) = 1]\right| \leq \mathsf{negl}(\lambda),$$

where the experiment $\mathrm{Exp}^{\mathcal{A},\mathcal{F},b}_{\mathrm{KDM}}(\lambda)$ is defined as follows:

1. *Sample public and secret keys $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$ and launch $\mathcal{A}(1^\lambda, \mathsf{pk})$.*
2. *Answer arbitrary number of queries $f \in \mathcal{F}$ from $\mathcal{A}$ with*

$$c_f = \mathsf{Enc}(\mathsf{pk}, f(\mathsf{sk})), \text{ if } b = 0, \text{ or } c_f = \mathsf{Enc}(\mathsf{pk}, 0^{|f(\mathsf{sk})|}), \text{ if } b = 1.$$

3. *In the end, $\mathcal{A}$ outputs a bit b' as the experiment result.*

C More Constructions

C.1 Succinct LPZK Arguments from Prime-order Groups

In addition to Paillier groups, we also construct succinct LPZK arguments from prime-order groups. The construction basically shares the same back-bone structure as Π_{zkPai}, where the parties non-interactively evaluate the circuit via VOLE and DDLog techniques. However, the major difference is that it has a non-negligible correctness error, since the DDLog algorithm over prime-order groups [BGI16, BGI17, BGI+18] has $\delta = 1/\mathsf{poly}(\lambda)$ correctness error and takes $O(\sqrt{1/\delta})$ computations.

Let $\mathbb{G}$ be a group with a sufficiently large prime order N. The security relies on the circular-power DDH assumption over $\mathbb{G}$ (Definition B.3). That is, $\mathsf{crs} := (g, h, h_1, h_2) \in \mathbb{G}^4$ is computationally indistinguishable from uniform, where $g \xleftarrow{\$} \mathbb{G}$, $h = g^r$, $h_1 = h^{\Delta}$, $h_2 = h^{\Delta^2} \cdot g^{\Delta}$, for $r \xleftarrow{\$} [1, N]$ and $\Delta \xleftarrow{\$} [1, 2^\lambda]$. Similarly, by the following equality:

$$\begin{aligned} K_z :&= K_x K_y = (\Delta \cdot x + M_x)(\Delta \cdot y + M_y) \\ &= -\Delta \cdot z + \underbrace{\big((\Delta^2 + \Delta) \cdot xy + \Delta \cdot (yM_x + xM_y) + M_x M_y\big)}_{M_z}, \end{aligned}$$

the prover and the verifier can locally evaluate multiplication gates on inputs $[\![x]\!], [\![y]\!]$ and crs in the following way,

1. $\mathcal{P}$ computes $u := h^{M_x M_y} \cdot h_1^{xM_y + yM_x} \cdot h_2^{xy}$ and $M_z := r_z - \mathsf{DDLog}_g(u) \mod N$.
2. $\mathcal{V}$ computes $v := h^{K_x K_y}$ and $K_z := r_z - \mathsf{DDLog}_g(v) \mod N$.

Note that r_z is a common random shift in $\mathbb{Z}_N$, and can be generating by evaluating a PRF with its key included in crs. Suppose $N \gg B \cdot \lambda^{\omega(1)}$. By the correctness of $\mathsf{DDLog}_g(\cdot)$ and Lemma 3.2, $K_z = \Delta \cdot z + M_z$ holds over $\mathbb{Z}$ with probability at least $1 - \delta(\lambda) - \mathsf{negl}(\lambda)$. Then by a union bound, the VOLE invariant for the output wire w holds with probability at least $1 - |\mathcal{C}|\delta(\lambda) - \mathsf{negl}(\lambda)$.

We remark that the bit-decomposition method and the key-switching in Section 4.1 are also applicable in this case. For completeness and simplicity, we present a self-contained construction Π_{zkPri} from circular-power DDH assumptions over prime-order groups.

Protocol 7: Π_{zkPri}

Let $\mathcal{C} : \mathbb{Z}^n \to \mathbb{Z}$ be a B-bounded verification circuit, such that $B \cdot 2^\lambda \ll N$. The prover $\mathcal{P}$ wants to convince the verifier $\mathcal{V}$ that he holds $\mathbf{x} \in \mathbb{Z}^n$ such that $\mathcal{C}(\mathbf{x}) = 0$. Assume oracle access to a half-chosen VOLE functionality $\mathcal{F}_{\mathsf{hcVOLE}}$.

crs: $(\mathbb{G}, N, g, h, h_1, h_2)$ where $\mathbb{G}$ is a group with prime order N, and $g \xleftarrow{\$} \mathbb{G}$, $h := g^r$ with $r \xleftarrow{\$} [1, N]$, $h_1 := h^\Delta$ with $\Delta \xleftarrow{\$} [1, 2^\lambda]$, $h_2 := h^{\Delta^2} \cdot g^\Delta$. //We implicitly assume crs includes a PRF key to produce common random shifts.

1. Both parties are given crs and $\mathcal{V}$ is additionally given $\Delta \in [1, 2^\lambda]$.
2. $\mathcal{P}$ and $\mathcal{V}$ invoke $\mathcal{F}_{\mathsf{hcVOLE}}$ with their inputs $\mathbf{x}, \Delta$, respectively. Then they receive $M_{\mathbf{x}}, K_{\mathbf{x}}$, where $K_{\mathbf{x}} = \Delta \cdot \mathbf{x} + M_{\mathbf{x}}$.
3. For each gate G in the circuit with respect to input wire values x, y,
   - If G is an addition gate, $\mathcal{P}$ computes $z := x + y, M_z := M_x + M_y$, while $\mathcal{V}$ computes $K_z := K_x + K_y$.
   - If G is a multiplication gate. Let $r_z \in \mathbb{Z}_N$ be a common random shift with respect to output wire z, $\mathcal{P}$ computes
   $$M_z \leftarrow -\mathsf{DDLog}_g\left(h_2^{xy} \cdot h_1^{yM_x + xM_y} \cdot h^{M_x M_y}\right) + r_z \mod N,$$
   while $\mathcal{V}$ computes $K_z \leftarrow -\mathsf{DDLog}_g(h^{K_x K_y}) + r_z \mod N$.
   By the randomness of r_z and correctness of DDLog_g, $K_z = \Delta \cdot z + M_z$ holds over $\mathbb{Z}$ except with probability at most $\delta(\lambda) + \frac{|\Delta \cdot xy|}{N}$.
4. For the single output wire ω, $\mathcal{P}$ opens M_ω to $\mathcal{V}$. $\mathcal{V}$ accepts if $K_\omega = M_\omega$. Otherwise, $\mathcal{V}$ aborts.

Theorem C.1. *Assume circular-power DDH in prime-order groups (Definition B.3) and secure PRFs, Π_{zkPri} is secure in the $(\mathcal{F}_{\mathsf{hcVOLE}}, CRS)$-hybrid model.*

Proof. *The proof is analogues to Theorem 4.2, except for an inverse-polynomial correctness error introduced by* DDLog *over prime-order groups.* □

C.2 Succinct LPZK from Damgård-Jurik

In this section, we show a succinct LPZK construction based on the KDM security of Damgård-Jurik encryption scheme. We refer notions for DJ encryption to Section B.2. Assume $N = p \cdot q$ is a secure RSA modulus and $\zeta \geq 1$. Recall that the goal is to non-interactively evaluate multiplications on VOLE. Here the main observation is that, given input VOLE variants $[\![x]\!], [\![y]\!]$, we have

$$\Delta^{-1} \cdot \left(K_x K_y - M_x M_y\right) = \Delta \cdot xy + (yM_x + xM_y).$$

To get $[\![xy]\!]$, it remains to let the parties obtain subtractive sharings $\left[\Delta^{-1} \cdot M_x M_y\right]$, where no party holds all Δ^{-1}, M_x, M_y in the clear. The high-level idea is to multiply in the exponent and then compute its discrete logarithm distributively. In more detail, we let $\Delta := \mathsf{sk} \in \mathbb{Z}_N$, and reveal $h := \mathsf{DJ.Enc}(\Delta^{-1})$ to $\mathcal{P}$, which admits that

$$\begin{aligned}
h^{K_x K_y} &= h^{M_x M_y} \cdot h^{\Delta \cdot \left(\Delta \cdot xy + (yM_x + xM_y)\right)} \\
&= h^{M_x M_y} \cdot r^{N^\zeta \Delta \cdot \left(\Delta \cdot xy + (yM_x + xM_y)\right)} \mod N^\zeta \varphi(N) \cdot (1+N)^{\Delta^{-1} \Delta \cdot \left(\Delta \cdot xy + (yM_x + xM_y)\right)} \mod N^\zeta \\
&\equiv h^{M_x M_y} \cdot (1+N)^{\Delta \cdot xy + (yM_x + xM_y)} \mod N^\zeta \mod N^{\zeta+1}.
\end{aligned}$$

Hence, by applying DDLog on the both side, we immediately get

$$\mathsf{DDLog}(h^{K_x K_y}) = \mathsf{DDLog}(h^{M_x M_y}) + \Delta \cdot xy + (yM_x + xM_y) \mod N^\zeta.$$

We actually want subtractive sharings hold over $\mathbb{Z}$ to support subsequent computations. This can be guaranteed with high probability by the lifting technique of [OSY21], if $\Delta \cdot xy$ is much smaller than N^ζ. Hence, we choose $\zeta \geq 2$ and assume the computations are bounded by $B = \mathsf{poly}(\lambda) \ll N$.

Protocol 8: Π_{zkDJ}

Let $\mathcal{C} : \mathbb{Z}^n \to \mathbb{Z}$ be a (B-bounded) verification circuit, such that $B \ll N$. The prover $\mathcal{P}$ wants to convince the verifier $\mathcal{V}$ that he holds $\mathbf{x} \in \mathbb{Z}^n$ such that $\mathcal{C}(\mathbf{x}) = 0$. Given oracle access to a half-chosen VOLE functionality $\mathcal{F}_{\mathsf{hcVOLE}}$.
crs: (N, ζ, h) where N is a secure RSA modulus, $\zeta \geq 2$, h is a DJ ciphertext of sk^{-1}, i.e., $h := r^{N^\zeta} \cdot \exp(\mathsf{sk}^{-1})$ for $r \xleftarrow{\$} \mathbb{Z}^*_{N^{\zeta+1}}$, where $\mathsf{sk} = \varphi(N)$. //We implicitly assume crs includes a PRF key to produce common random shifts.

1. Both parties are given crs and $\mathcal{V}$ is additionally given $\Delta := \mathsf{sk} \in \mathbb{Z}_N$.
2. $\mathcal{P}$ and $\mathcal{V}$ invoke $\mathcal{F}_{\mathsf{hcVOLE}}$ with their inputs $\mathbf{x}, \Delta$, respectively. Then they receive $M_{\mathbf{x}}, K_{\mathbf{x}}$, where $K_{\mathbf{x}} = \Delta \cdot \mathbf{x} + M_{\mathbf{x}}$.
3. For each gate G in the circuit with respect to input wire values x, y,
   - If G is an addition gate, $\mathcal{P}$ computes $z := x + y, M_z := M_x + M_y$, while $\mathcal{V}$ computes $K_z := K_x + K_y$.
   - If G is a multiplication gate, let $r_z \in \mathbb{Z}_{N^\zeta}$ be the common pseudorandom shift related to output wire z, $\mathcal{P}$ computes

$$M_z \leftarrow \mathsf{DDLog}_g\left(h^{M_x M_y}\right) + yM_x + xM_y + r_z \mod N^\zeta,$$

while $\mathcal{V}$ computes $K_z \leftarrow \mathsf{DDLog}_g(h^{K_x K_y}) + r_z \mod N^\zeta$.
By the Lifting lemma, $K_z = \Delta \cdot z + M_z$ holds over $\mathbb{Z}$ with probability at least $1 - \frac{\varphi(N)B}{N^\zeta}$.

4. For the single output wire ω, $\mathcal{P}$ opens M_ω to $\mathcal{V}$. $\mathcal{V}$ accepts if $K_\omega = M_\omega$. Otherwise, $\mathcal{V}$ aborts.

Theorem C.2. *Assume KDM-security of Damgård-Jurik cryptosystem and secure PRFs, Π_{zkDJ} is secure in the $(\mathcal{F}_{\mathsf{hcVOLE}}, CRS)$-hybrid model.*

Proof. The proof follows analogously to Theorem 4.2. □

D Deferred Proof

D.1 Security Proof for Π_{zkPai}

We prove knowledge soundness and zero-knowledge separately.

Theorem D.1 (Proof of Knowledge Soundness). *Assuming circular-power DDH in Paillier groups (Definition 3.4) and secure PRFs, Π_{zkPai} achieves computational knowledge soundness in the $(\mathcal{F}_{\mathsf{hcVOLE}}, CRS)$-hybrid model.*

Note that Π_{zkPai} achieves *reusable* knowledge soundness. The soundness proof proceeds by gradually eliminating the verification query's dependence on Δ and $(K_{\mathbf{x}}, K_\omega)$ through a sequence of games: Game_2 eliminates reliance on K_ω, and Game_3 removes dependence on verifier's secret key sk. We show that if a PPT adversary $\mathcal{A}$ produces a valid proof for $\mathcal{C}(\mathbf{x}) \neq 0$, then $\mathcal{A}$ can be used to break the circular-power DDH assumption in Paillier groups.

Proof. Knowledge soundness is established via a sequence of games.

- Game_0: It is the real proof of knowledge soundness definition. Specifically, the game is defined as:
  1. Sample $(\mathsf{crs}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$ and send crs to $\mathcal{A}$.
  2. $\mathcal{A}$ makes a polynomial number of verification queries. Given a query $(\mathcal{C}, \pi)$, the challenger runs as an honest verifier. Specifically, for each the query they invoke $\mathcal{F}_{\mathsf{hcVOLE}}$ which returns input VOLE correlations $[\mathbf{x} \cdot \Delta]$ and the proof message is $\pi =: M_\omega$. The challenger computes K_ω and outputs 1 if $K_\omega = M_\omega$ and 0 otherwise.
  3. After the queries, $\mathcal{A}$ invokes $\mathcal{F}_{\mathsf{hcVOLE}}$ with $\mathbf{x}^*$ and outputs a π^*. $\mathcal{A}$ wins if $\mathcal{C}(\mathbf{x}^*) \neq 0$ and an honest verifier would accept the proof.
- Game_1: Same as Game_0 except that the challenger emulates the half-chosen VOLE functionality $\mathcal{F}_{\mathsf{hcVOLE}}$. Hence, the challenger obtains $\mathcal{A}$'s inputs $\mathbf{x}$ for each invocation. Note that this holds as long as $\mathcal{F}_{\mathsf{hcVOLE}}$ is implemented with an extractable half-chosen VOLE protocol.

– Game_2: Same as Game_1 except that the challenger responses the verification queries in the following way: Given a query $(\mathcal{C}, \pi)$ with $\pi := M_\omega$ (with $\mathbf{x}$ sent to $\mathcal{F}_{\mathsf{hcVOLE}}$), the challenger obtains $(\mathbf{x}, M'_{\mathbf{x}})$, then computes M'_z gate by gate and then obtains M'_ω for output as an honest prover. Finally, the challenger outputs 1 if $M'_\omega + \mathcal{C}(\mathbf{x}) \cdot \Delta = M_\omega$ and 0 otherwise. Note that K_ω is not used in this equation.
– Game_3: Same as Game_2 except that the challenger responses the verification queries in the following way: Given a query $(\mathcal{C}, \pi)$ with $\pi := M_\omega$ (with $\mathbf{x}$ sent to $\mathcal{F}_{\mathsf{hcVOLE}}$), the challenger outputs 1 if both $\mathcal{C}(\mathbf{x}) = 0$ and $M'_\omega = M_\omega$ hold and 0 otherwise. Note that Δ is not used in this game.

Now, we prove the above games are computationally indistinguishable for a PPT $\mathcal{A}$. Define G_i as the event that $\mathcal{A}$ wins in Game_i.

Lemma D.2. $|\Pr[G_0] - \Pr[G_1]| \leq \mathsf{negl}(\lambda)$.

Proof. Assuming $\mathcal{F}_{\mathsf{hcVOLE}}$ is realized using a half-chosen VOLE protocol with correctness probability $1 - \mathsf{negl}(\lambda)$, it follows that $|\Pr[G_0] - \Pr[G_1]| \leq \mathsf{negl}(\lambda)$. □

Lemma D.3. $|\Pr[G_1] - \Pr[G_2]| \leq \mathsf{negl}(\lambda)$.

Proof. By the gate-by-gate correctness of DDLog, $M'_\omega + \mathcal{C}(\mathbf{x}) \cdot \Delta = K_\omega$ holds except for a negligible probability. Hence, the two checks $M'_\omega + \mathcal{C}(\mathbf{x}) \cdot \Delta = M_\omega$ and $K_\omega = M_\omega$ are equivalent except for a negligible probability. □

Lemma D.4. *Assume the circular-power DDH assumption in Paillier groups. Then* $|\Pr[G_2] - \Pr[G_3]| \leq \mathsf{negl}(\lambda)$.

Proof. Assume $\mathcal{A}$ makes $q + 1 = \mathsf{poly}(\lambda)$ verification queries. Without loss of generality, let the final output $(\mathbf{x}^*, \pi^*)$ be the last query. For $i \in [0, q+1]$, let H_i be the hybrid game the first i queries are answered by the strategy in Game_3 and the remaining queries are answered by the strategy in Game_2. Then $H_0 = G_2$ and $H_{q+1} = G_3$. Note that the $(i+1)$-th query in H_i is answered by the strategy in Game_3 and in H_{i+1} answered by by the strategy in Game_2. Let E_j be the event that $\mathcal{A}$ wins in H_j. Let F_{i+1} be the event that $(i+1)$-th query of $\mathcal{A}$ causes a difference between H_i and H_{i+1}. Then H_i and H_{i+1} are identical if F_{i+1} does not happen. Thus, $|\Pr[E_i] - \Pr[E_{i+1}]| \leq \Pr[F_{i+1}]$. We claim that $\Pr[F_{i+1}] \leq \mathsf{negl}(\lambda)$. Then, since $q = \mathsf{poly}(\lambda)$, we have

$$|\Pr[G_2] - \Pr[G_3]| = |\Pr[E_0] - \Pr[E_{q+1}]| \leq \sum_{i \in [1,q]} \Pr[F_{i+1}] \leq \mathsf{negl}(\lambda)$$

Claim. It holds that $\Pr[F_{i+1}] \leq \mathsf{negl}(\lambda)$ assuming the circular-power DDH assumption in Paillier groups.

Proof (Proof of the Claim). Note that the verification condition

$$\mathcal{C}(\mathbf{x}) = 0 \text{ and } M'_\omega = M_\omega$$

of Game_3 is a sufficient condition of the verification condition

$$M'_\omega + \mathcal{C}(\mathbf{x}) \cdot \Delta = M_\omega$$

of Game_2. The complement condition of Game_3 in Game_2 is that

$$\mathcal{C}(\mathbf{x}) \neq 0 \iff M'_\omega \neq M_\omega.$$

From the complement condition, we have $\Delta = (M_\omega - M'_\omega)/\mathcal{C}(\mathbf{x})$ and thus $\Pr[F_{i+1}] \leq \Pr[\Delta = (M_\omega - M'_\omega)/\mathcal{C}(\mathbf{x})]$. Note that both in H_i and H_{i+1}, Δ is *never* used before the $(i+1)$-th query as the first i queries are answered by the strategy in Game_3.

Next we argue that for the $(i+1)$-th query, the probability that $\mathcal{A}$ outputs M_ω such that $\Delta = (M_\omega - M'_\omega)/\mathcal{C}(\mathbf{x})$ is negligible, i.e.,

$$\Pr[\Delta = (M_\omega - M'_\omega)/\mathcal{C}(\mathbf{x})] \leq \mathsf{negl}(\lambda).$$

Since Δ is not used prior to the $(i+1)$-th query, a PPT adversary $\mathcal{A}$ that succeeds in this query can be used to break the circular-power DDH assumption. □

This completes the proof. □

Remark D.5. Note that exactly in this proof, the number of verification queries are taken into account for the sake of reusability.

Lemma D.6. $\Pr[G_3] = 0$.

Proof. $\mathcal{A}$ cannot win in Game_3 with input $\mathbf{x}$ such that $\mathcal{C}(\mathbf{x}) \neq 0$. □

Combing the above lemmas, we obtain $\Pr[G_0] \leq \mathsf{negl}(\lambda)$. This completes the proof of knowledge soundness. □

Theorem D.7 (Zero-Knowledge). *Assume a secure PRF, Π_{zkPai} is zero-knowledge in the $(\mathcal{F}_{\mathsf{hcVOLE}}, CRS)$-hybrid model.*

Proof. Let $(\mathcal{S}_1, \mathcal{S}_2)$ be the simulator of Π_{zkPai}. The simulator works as follows.

- $\mathcal{S}_1(1^\lambda)$: Generate a random valid $(\widetilde{\mathsf{crs}}, \widetilde{\mathsf{sk}})$ as in Π_{zkPai} and give $(\widetilde{\mathsf{crs}}, \widetilde{\mathsf{sk}})$ to $\mathcal{A}$.
- $\mathcal{S}_2(\widetilde{\mathsf{crs}}, \widetilde{\mathsf{sk}}, \mathcal{C})$: Emulate $\mathcal{F}_{\mathsf{hcVOLE}}$ by sending $\widetilde{K}_{\mathbf{x}} \xleftarrow{\$} \mathbb{Z}_N^n$ to $\mathcal{A}$. Next, compute the $\widetilde{K}_z$ gate by gate as an honest verifier in Π_{zkPai} to obtain $\widetilde{K}_\omega$ for the output wire and let $\widetilde{M}_\omega := \widetilde{K}_\omega$. Output the simulated proof $\widetilde{\pi} := \widetilde{M}_\omega$.

We now prove the simulated proof is computationally indistinguishable from the real proof via a series of games.

- Game_0: This is the real proof generation game as follows.
  1. $\mathcal{A}$ is given crs as well as a verification key sk.
  2. $\mathcal{A}$ makes a polynomial number of queries to a proving oracle. Specifically, on each proving query $(\mathcal{C}, \mathbf{x})$, if $\mathcal{C}(\mathbf{x}) \neq 0$, return $\perp$. Otherwise, $\mathcal{F}_{\mathsf{hcVOLE}}$ is invoked (to obtain $[\![\mathbf{x}]\!]$), and then M_ω is computed from $(\mathbf{x}, M_{\mathbf{x}})$ following the honest prover's algorithm. Finally, the proving oracle returns M_ω to $\mathcal{A}$.

3. $\mathcal{A}$ outputs a bit b.

– Game_1: For each oracle query, same as Game_0 except that $\mathcal{S}$ generates the $(\mathsf{crs}, \mathsf{sk})$ and emulates $\mathcal{F}_{\mathsf{hcVOLE}}$ by sending uniformly random $\widetilde{K}_{\mathbf{x}}$ to $\mathcal{A}$ and setting $\widetilde{M}_{\mathbf{x}} = \widetilde{K}_{\mathbf{x}} - \Delta \cdot \mathbf{x} \mod N$. If $\mathcal{F}_{\mathsf{hcVOLE}}$ is implemented by a protocol Π_{hcVOLE}, $\mathcal{S}$ runs the simulator of Π_{hcVOLE} and obtains $\widetilde{K}_{\mathbf{x}}$.
– Game_2: Same as Game_1 except that $\mathcal{S}$ computes $\widetilde{K}_\omega$ as an honest verifier and returns $\widetilde{M}_\omega := \widetilde{K}_\omega$ to $\mathcal{A}$.

Note that Game_2 is exactly the simulator. Let G_i be the event that $\mathcal{A}$ outputs 1 in Game_i.

Lemma D.8. $|\Pr[G_0] - \Pr[G_1]| \leq \mathsf{negl}(\lambda)$.

Proof. If $\mathcal{F}_{\mathsf{hcVOLE}}$ is implemented by a protocol with correctness holding with probability $1 - \mathsf{negl}(\lambda)$, $K_{\mathbf{x}}$ and $\widetilde{K}_{\mathbf{x}}$ are indistinguishable in the two games. □

Lemma D.9. $|\Pr[G_1] - \Pr[G_2]| \leq \mathsf{negl}(\lambda)$.

Proof. By the correctness of DDLog (this requires Lemma 3.2 and a secure PRF), $M_\omega = \widetilde{K}_\omega = \widetilde{M}_\omega$ holds except with a negligible probability, where M_ω is the output computed in Game_1 from $(\widetilde{M}_{\mathbf{x}}, \mathbf{x})$. Thus, $\mathcal{A}$ cannot distinguish Game_1 and Game_2 with a non-trivial advantage. □

Combining the above lemmas, we have $|\Pr[G_0] - \Pr[G_2]| \leq \mathsf{negl}(\lambda)$. This completes the proof of zero-knowledge. □

D.2 Proof of Π_{hcVOLE}

Recall that in Π_{hcVOLE}, $\mathsf{crs} := ((g_\kappa)_{\kappa\in[0,n]}, (e_{i,\kappa})_{i\in[1,n],\kappa\in[0,n]})$. Denote $\mathsf{sk} := (w_1 \dots w_n, \Delta)$. For simplicity, assume the sender message $(d_j)_{j\in[1,\ell]}$ in Π_{hcVOLE} is a commitment of the input $X \in \mathbb{Z}_N^{n\times\ell}$. Denote it as $\mathsf{com} := \mathsf{Commit}(\mathsf{crs}, X)$. Define $(M_{\mathbf{x}}, K_{\mathbf{x}})$ as $M_{\mathbf{x}} := ([z_{ij}]_0)_{i,j}$ and $K_{\mathbf{x}} := ([z_{ij}]_1)_{i,j}$. Assume the dimension of X with appended mask is m.

Security against a malicious receiver. We prove that for a receiver $\mathrm{P_R}$, the sender message is simulatable. Before the proof, a formal definition of simulatable sender is explicitly given.

Definition D.10 (Simulatable Sender). *There exists a* PPT *simulator* $(\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$ *such that* $\mathcal{S}_1$ *outputs a simulated* $(\mathsf{crs}, \mathsf{sk})$ *along with a trapdoor* τ*; given* (crs, τ), $\mathcal{S}_2$ *outputs a simulated commitment* com*; and given* $(\mathsf{crs}, (\chi_1 \dots \chi_m), \mathsf{st})$, $\mathcal{S}_3$ *outputs* (x', M')*; and for any* PPT *adversary* $\mathcal{A}$*, the following holds:*

$$\Big| \Pr\left[\mathcal{A}^{\mathsf{Sender}(\mathsf{crs},\cdot)}(\mathsf{crs}, \mathsf{sk}) = 1 \middle| (\mathsf{crs}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)\right] - \Pr\left[\mathcal{A}^{\mathcal{S}_2(\mathsf{crs},\tau,\cdot),\mathcal{S}_3(\mathsf{crs},\mathsf{st},\cdot)}(\mathsf{crs}, \mathsf{sk}) = 1 \middle| (\mathsf{crs}, \mathsf{sk}, \tau) \leftarrow \mathcal{S}_1(1^\lambda)\right] \Big| \leq \mathsf{negl}(\lambda).$$

Theorem D.11. *The sender of* Π_{hcVOLE} *is simulatable.*

Proof. Let $(\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3)$ be the simulator of Π_{hcVOLE}. The simulator works as follows.

- $\mathcal{S}_1(1^\lambda)$: Generate a valid $(\mathsf{crs}, \mathsf{sk})$ as in Π_{hcVOLE}. Give $(\mathsf{crs}, \mathsf{sk})$ to $\mathcal{A}$ and let $\tau := \mathsf{sk}$.
- $\mathcal{S}_2(\mathsf{crs}, \tau)$: Generate a commitment as $\widetilde{\mathsf{com}} := \mathsf{Commit}(\mathsf{crs}, \mathbf{0})$ and run the honest receiver computation on $(\mathsf{crs}, \mathsf{com}, \tau)$ to obtain $K_{\mathbf{x}}$. Finally, output $\widetilde{\mathsf{com}}$.
- $\mathcal{S}_3(\mathsf{crs}, (\chi_1 \dots \chi_m), \mathsf{sk})$: Compute $K' := \sum_i \chi_i \cdot K_{\mathbf{x},i}$ and sample $\widetilde{x}'$ uniformly. Next, compute $\widetilde{M'} := K' - \widetilde{x}' \cdot \Delta$. Output $(\widetilde{x}', \widetilde{M'})$.

We now prove the simulated sender message is computationally indistinguishable from the real proof via a series of games.

- Game_0: It is the real sender message generation game as follows.
  1. Sample $(\mathsf{crs}, \mathsf{sk})$ and give $(\mathsf{crs}, \mathsf{sk})$ to $\mathcal{A}$.
  2. Given $(1^\lambda, \mathsf{crs}, \mathsf{sk})$, $\mathcal{A}$ makes a polynomial number of queries to a sender oracle. Given input input $\mathbf{x}$, the oracle $\mathsf{Sender}(\mathsf{crs}, \cdot)$ works as follows. Sample $\mathsf{com} \leftarrow \mathsf{Commit}(\mathsf{crs}, \mathbf{x})$ and compute $M_{\mathbf{x}}$ from $(\mathbf{x}, \mathsf{crs})$ and the private randomness of Commit. Output com to $\mathcal{A}$. Given a linear combination coefficient $(\chi_1 \dots \chi_m)$, compute $x' := \sum_i \chi_i \cdot \mathbf{x}_i$ and $M'_{\mathbf{x}} := \sum_i \chi_i \cdot M_{\mathbf{x},i}$ and output (x', M') to $\mathcal{A}$.
  3. $\mathcal{A}$ outputs a bit b.
- Game_1: Same as Game_0 except $(\widetilde{x}', \widetilde{M'})$. Compute $K_{\mathbf{x}}$ as an honest verifier from $(\mathsf{sk}, \mathsf{com})$. Given a linear combination coefficient $(\chi_1 \dots \chi_m)$, compute $K' := \sum_i \chi_i \cdot K_{\mathbf{x},i}$ and sample $\widetilde{x}'$ uniformly. Next, compute $\widetilde{M'} := K' - \widetilde{x}' \cdot \Delta$. Finally, output $(\widetilde{x}', \widetilde{M'})$.
- Game_2: Same as Game_1 except $\widetilde{\mathsf{com}}$. Compute $\widetilde{\mathsf{com}} := \mathsf{Commit}(\mathsf{crs}, \mathbf{0})$.

Note that Game_2 is indeed the simulator. Let G_i be the event that $\mathcal{A}$ outputs 1 in Game_i.

Lemma D.12. $|\Pr[G_0] - \Pr[G_1]| \leq \mathsf{negl}(\lambda)$.

Proof. From the correctness of distributed algorithms, it holds that $M' + \Delta \cdot x' = K' = \widetilde{M'} + \Delta \cdot \widetilde{x}'$ except with negligible probability. Because $\widetilde{x}'$ is uniform sampled, $\widetilde{M'}$ admits the uniform distribution. Thus Game_0 and Game_1 admit the same distribution. □

Claim. The commitment is statistically hiding.

Proof (Proof of the Claim). For each commitment, it is re-randomized by g_0^v and thus it is statistically hiding. □

Lemma D.13. $|\Pr[G_1] - \Pr[G_2]| \leq \mathsf{negl}(\lambda)$.

Proof. For a PPT $\mathcal{A}$, it follows from the hiding property of the commitment that the encodings of $\mathbf{x}$ and $\mathbf{0}$ are indistinguishable. □

Combining the above lemmas, we have $|\Pr[G_0] - \Pr[G_2]| \leq \mathsf{negl}(\lambda)$. This completes the simulatable proof. □

Security against a malicious sender. We prove reusable soundness and extraction. The soundness means that if an adversary $\mathcal{A}$ outputs a commitment com and given a random challenge coefficient $(\chi_1 \dots \chi_m)$, $\mathcal{A}$ replies with inconsistent (M'^*, x'^*) such that the receiver accepts with negligible probability. The reusability means $\mathcal{A}$ can repeat the above procedure polynomially many times and learn whether the receiver accepts or rejects an opening. For simplicity, we explicitly define the function $\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, \mathsf{com}, M', x', (\chi_1 \dots \chi_m))$ from Π_{hcVOLE} for the verification in opening.

– $\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, \mathsf{com}, M', x', (\chi_1 \dots \chi_m))$: Compute the receiver result as $K_{\mathbf{x}}$ from $(\mathsf{crs}, \mathsf{sk}, \mathsf{com})$ and $K' := \sum_i \chi_i \cdot K_{\mathbf{x},i}$. Output 1 if $x' \cdot \Delta + M' = K'$ and 0 otherwise.

Definition D.14 (Reusable Extractable Half-Chosen VOLE). *For any* PPT *adversary* $\mathcal{A}$*, it holds that:*

$$\Pr\left[\begin{array}{l|l} \mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, \mathsf{com}^*, M'^*, & (\mathsf{crs}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda) \\ x'^*, (\chi_1 \dots \chi_m)) = 1 & (\mathsf{com}^*, \mathsf{st}) \leftarrow \mathcal{A}^{\mathsf{Verify}(\mathsf{crs}, \mathsf{sk}, \cdot, \cdot, \cdot)}(\mathsf{crs}) \\ x'^* \neq x' & (\chi_1 \dots \chi_m) \xleftarrow{\$} \mathbb{Z}_N^m \\ & (M'^*, x'^*) \leftarrow \mathcal{A}(\mathsf{st}, \chi_1 \dots \chi_m) \\ & \mathbf{x}^* \leftarrow \mathsf{Ext}^{\mathcal{A}}(\mathsf{crs}, \mathsf{com}^*) \\ & x' \leftarrow \sum_i \chi_i \cdot \mathbf{x}_i^* \end{array}\right] \leq \mathsf{negl}(\lambda).$$

Theorem D.15 (Reusability). *Assuming DCR,* Π_{hcVOLE} *is reusable and extractable.*

We prove that if a PPT adversary $\mathcal{A}$ outputs an acceptable response for inconsistent commitments and responses, then $\mathcal{A}$ breaks the DCR assumption. To achieve this, the proof proceeds by gradually removing the reliance on $(K_{\mathbf{x}}, \Delta)$.

Proof. We establish reusability through a sequence of games.

– Game_0: It is the real reusability game. Specifically, the game is defined as:
  1. Sample $(\mathsf{crs}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$ and send crs to $\mathcal{A}$.
  2. $\mathcal{A}$ makes a polynomial number of query experiments. The query experiment is defined as:
     (a) $\mathcal{A}$ sends a commitment com to the challenger.
     (b) The challenger responses with the coefficients $\{\chi_i\}_i$. Additionally, the challenger computes $K_{\mathbf{x}}$.
     (c) $\mathcal{A}$ replies with (x', M').
     (d) The challenger computes $K' = \sum_i \chi_i \cdot K_{\mathbf{x},i}$. The experiment outputs 1 if $x' \cdot \Delta + M' = K'$ and 0 otherwise.
  3. After the queries, $\mathcal{A}$ outputs com. The challenger computes $K'^*_{\mathbf{x}}$ from $(\mathsf{com}, \mathsf{sk})$. Given $(\chi_1 \dots \chi_m)$, $\mathcal{A}$ replies with (M'^*, x'^*). Let $\mathbf{x}^*$ be the extracted input and $x' := \sum \chi_i \cdot \mathbf{x}_i^*$. Let $K'^* := \sum_i \chi_i \cdot K'^*_{\mathbf{x},i}$. $\mathcal{A}$ wins if $M'^* + x'^* \cdot \Delta = K'^*$ and $x' \neq x'^*$.
– Game_1: It is the real reusability game with *extractor* and it is as same as Game_0 except Step (2b) is rewound m times. Specifically,

(a) $\mathcal{A}$ sends a commitment com to the challenger.
(b) The challenger responses m blocks of coefficients χ_j for $j \in [m]$ with each block $\chi_j \in \mathbb{Z}_N^m$. Additionally, the challenger computes $K_{\mathbf{x}}$.
(c) $\mathcal{A}$ replies with $(x'_j, M'_j)_{j\in[m]}$.
(d) The challenger computes $K'_j = \sum_i \chi_{ji} \cdot K_{\mathbf{x},i}$. The experiment outputs 1 if $\forall j \in [m], x'_j \cdot \Delta + M'_j = K'_j$ and 0 otherwise.

– Game_2: It is as same as Game_1 except Step (d) additionally extracts the input $\mathbf{x}$ from the linear combination answers and and computes $M_{\mathbf{x}} := K_{\mathbf{x}} - \Delta \cdot \mathbf{x}$. In particular, the challenger is changed to
(d) The challenger computes $\mathbf{x}$ from $(x'_j, \chi_j)_j$ and $M_{\mathbf{x}} := K_{\mathbf{x}} - \Delta \cdot \mathbf{x}$.
(e) The challenger computes $K'_j = \sum_i \chi_{ji} \cdot K_{\mathbf{x},i}$. The experiment outputs 1 if $x'_j \cdot \Delta + M'_j = K'_j$ for every $j \in [m]$ and 0 otherwise.

– Game_3: It is as same as Game_2 except Step (e) is changed to
(e) The challenger computes (x'^*_j, M'^*_j) via linear combinations for $j \in [m]$ from $(\mathbf{x}, M_{\mathbf{x}})$.
(f) The challenger verifies $x'_j \cdot \Delta + M'_j = x'^*_j \cdot \Delta + M'^*_j$.
Note that in Game_3 the secret information of receiver except Δ is not involved.

– Game_4: It is as same as Game_3 except Step (f) is changed to
(f) The challenger verifies $(x'_j, M'_j) = (x'^*_j, M'^*_j)$ for each $j \in [m]$.
Note that in Game_4 the secret information of receiver is entirely not involved.

Define G_i as the event that $\mathcal{A}$ wins in Game_i. Next, we prove the above games are computationally indistinguishable for a PPT adversary $\mathcal{A}$.

Lemma D.16. $\Pr[G_0] \leq \Pr[G_1]/m$.

Note that Game_1 is the rewinding of Game_0 with m times.

Lemma D.17. $|\Pr[G_1] - \Pr[G_2]| = 0$.

From $\mathcal{A}$'s view, there is no difference between Game_1 and Game_2 because the extractor computation is oblivious to $\mathcal{A}$.

Lemma D.18. $|\Pr[G_2] - \Pr[G_3]| \leq \mathsf{negl}(\lambda)$.

Proof. We prove the verification condition

$$x'_j \cdot \Delta + M'_j = x'^*_j \cdot \Delta + M'^*_j \tag{4}$$

in Game_3 is equivalent to

$$x'_j \cdot \Delta + M'_j = K'_j \tag{5}$$

in Game_2. From the correctness of extractor and DDLog, the equality $x'^*_j \cdot \Delta + M'^*_j = K'_j$ holds except for a negligible probability. Therefore, the two conditions Eq.(4) and Eq.(5) are computationally indistinguishable. □

Lemma D.19. *Assuming the DCR assumptions, it holds that* $|\Pr[G_3] - \Pr[G_4]| \leq \mathsf{negl}(\lambda)$.

Proof. Assume $\mathcal{A}$ launches $q+1 = \mathsf{poly}(\lambda)$ verification queries. Without loss of generality, let the final output (x^*, π^*) be the last query. For $i \in [1, q+1]$, let H_i be the hybrid that the first i queries are answered by Verify_4 and the remaining queries are answered by Verify_3. Denote E_j as the event that $\mathcal{A}$ wins in hybrid H_j. Denote F_{i+1} as the event that $(i+1)$-th query of $\mathcal{A}$ causes a difference between H_i and H_{i+1}. Then H_i and H_{i+1} are identical if F_{i+1} does not happen. Thus, $|\Pr[E_i] - \Pr[E_{i+1}]| \leq \Pr[F_{i+1}]$. We claim that $\Pr[F_{i+1}] \leq \mathsf{negl}(\lambda)$. Then,

$$|\Pr[G_3] - \Pr[G_4]| = |\Pr[E_1] - \Pr[E_{q+1}]| \leq \sum_{i \in [1,q]} \Pr[F_{i+1}] \leq \mathsf{negl}(\lambda)$$

because $q = \mathsf{poly}(\lambda)$.

Claim. It holds that $\Pr[F_{i+1}] \leq \mathsf{negl}(\lambda)$ assuming the DCR assumption.

Proof (Proof of the Claim). Recall that the verification condition

$$\forall j \in [m], (x'_j, M'_j) = (x'^*_j, M'^*_j)$$

in Game_4 is a proper subset of the verification condition

$$\forall j \in [m], x'_j \cdot \Delta + M'_j = x'^*_j \cdot \Delta + M'^*_j$$

in Game_3. The complement condition of Game_4 in Game_3 is that

$$\exists j \in [m], (x'_j, M'_j) \neq (x'^*_j, M'^*_j).$$

Note that $x'_j \neq x'^*_j$ is equivalent to $M'_j \neq M'^*_j$ conditioned on $x'_j \cdot \Delta + M'_j = x'^*_j \cdot \Delta + M'^*_j$. From the complement condition, we have $\Delta = (M'^*_j - M'_j)/(x'_j - x'^*_j)$ and thus

$$\Pr[F_{i+1}] \leq \Pr[\Delta = \frac{M'^*_j - M'_j}{x'_j - x'^*_j}].$$

Note that both in H_i and H_{i+1}, Δ is never used before the $(i+1)$-th query because the first i queries are answered by Verify_4.

Next we argue that for the $(i+1)$-th query, the probability that the event F_{i+1} i.e., $\Delta = (M'^*_j - M'_j)/(x'_j - x'^*_j)$ happens with negligible probability. Before $(i+1)$-th query, Δ is not involved. Therefore, $\mathcal{A}$ can be used to break the DCR assumption. □

□

Lemma D.20. $\Pr[G_4] = 0$.

Proof. Note that winning condition is that $x' \neq x'^*$ and the verifiable condition in Game_4 is that $(x'_j, M'_j) = (x'^*_j, M'^*_j)$ for $j \in [m]$. Hence, $\mathcal{A}$ never wins Game_4. □

Combing the above lemmas, we obtain $\Pr[G_1] \leq \mathsf{negl}(\lambda)$ and thus $\Pr[G_0] \leq \mathsf{negl}(\lambda)$ as $m = \mathsf{poly}(\lambda)$. This completes the proof of reusability. □