

Cryptanalysis of LDVR: Polynomial-Time Adaptive Attacks against Threshold Schnorr Signatures

Àlex Rodríguez García¹ and Stefano Tessaro²

¹ ETH Zurich

² University of Washington

alex.rodriguezgarcia@epfl.ch, tessaro@cs.washington.edu

Abstract. We present the first polynomial-time attacks against certain parameter choices of the low-dimensional vector representation (LDVR) problem, introduced by Crites, Katz, Komlo, Tessaro, and Zhu (CRYPTO ’25) to prove the adaptive security of FROST in the Algebraic Group Model. Building on work by Crites and Stewart (CRYPTO ’25), they also show that every attack against LDVR directly yields an adaptive attack—i.e., one leveraging adaptive corruptions—against any threshold Schnorr signature that exposes to the attacker the values g^{sk_i} for the secret-key shares sk_i held by the signers, where g is a generator of the underlying group. This class of protocols includes FROST (Komlo and Goldberg, SAC ’20), Sparkle (Crites, Komlo, and Maller, CRYPTO ’23), Lindell’s three-round protocol (CiC ’24), along with other masking-free protocols. Consequently, our results give the first polynomial-time adaptive attacks against a class of practical threshold signatures.

Our attacks require a large, but still polynomial, number of signers: asymptotically, we require $n = \Theta(\log^2 p)$ signers and corrupt $t_c = \Theta(\log p)$ of them. For the prime-order subgroup of Curve25519, concrete and optimized instantiations of our attack require $n = 5020$ signers and corrupt 204 of them. This exceeds the signer counts commonly encountered in practice, but falls within the scenarios considered by NIST’s call for threshold signatures. At the theoretical level, our results also yield a non-artificial separation between static and adaptive security based on a real-world multi-party protocol.

Our results arise from a new reduction from LDVR to structured subset-sum problems determined by the evaluation points of the polynomial used to secret-share the signing key. We further show that allowing these evaluation points to be chosen adversarially can lead to substantially more efficient attacks.

1 Introduction

Threshold signatures [Des93] adopt a split-trust approach to distribute the process of signing a message across n signers, so that a signature can only be issued if a sufficiently large subset of at least $t + 1$ signers participates in signing, whereas no adversarial coalition of $t_c \leq t$ signers should be able to generate a valid signature on its own. Threshold signatures are at the center of standardization efforts by NIST [BD22, BP25] and IRTF [CKGW24], and have been the subject of significant research interest.

THRESHOLD SCHNORR. A Schnorr signature [Sch91] relies on a cyclic group $\mathbb{G} = \langle g \rangle$ of prime order p —the secret key is $\text{sk} \leftarrow \mathbb{F}_p$, and the public verification key is $\text{pk} = g^{\text{sk}}$. A signature for a message $\text{msg} \in \{0, 1\}^*$ is a pair $\sigma = (R, z) \in \mathbb{G} \times \mathbb{F}_p$ such that $R \cdot \text{pk}^c = g^z$, where $c = H(\text{pk}, R, \text{msg})$.³

³ Or, alternatively, a pair $\sigma = (c, z)$ such that $c = H(\text{pk}, g^z \text{pk}^{-c}, \text{msg})$.

Several works, starting with [SS01, GJKR99], have considered the threshold generation of Schnorr signatures, and these protocols play a central role in NIST’s process [BD22]. Currently, FROST [KG20, CKGW24, BCK⁺22] is the most popular scheme used in practice and only requires two communication rounds, but other protocols are also used in production. These protocols leverage linear structures present in Schnorr signatures and use Shamir secret sharing to distribute $\mathbf{sk}$ —each party obtains $p(x_i) = \mathbf{sk}_i$ for a random degree- t polynomial $p(X) \in \mathbb{F}_p[X]$ such that $p(0) = \mathbf{sk}$. Here, $x_1, \dots, x_n \in \mathbb{F}_p^*$ are distinct non-zero evaluation points associated with the n signers.

ADAPTIVE SECURITY. As in the broader context of multi-party computation (MPC), it is necessary to be precise about how adversarial coalitions are formed. A number of threshold signatures are only proved to achieve *static security*, i.e., they are proved secure only against adversaries that commit ahead of time to the set of signers they corrupt. In this work, however, we are interested in *adaptive* security, i.e., the far more realistic notion in which an adversary can corrupt signers adaptively throughout the execution of the signing protocol. NIST [BP25], in particular, states a clear preference for such schemes.

Attaining provable adaptive security without compromising efficiency has turned out to be technically hard. For example, Sparkle [CKM23] uses one additional message-dependent round over FROST to enable an adaptive security proof, in the AGM [FKL18] and ROM [BR93], which was however subsequently shown to be inherently incorrect [BLT⁺24, CS25].

Glacius [BDLR25b] and Gargos [BDLR25a] are five- and three-round threshold Schnorr protocols, respectively, proved to achieve adaptive security in the ROM under the DDH assumption. These protocols rely on a heavier setup than simply sharing $\mathbf{sk}$, but, like Sparkle and FROST, achieve *identifiable abort* (IA), i.e., the property that allows one to identify who supplied malformed signature shares when generating a signature. Katsumata, Reichle, and Takemure [KRT24] and Niot, Reichle, and Takemure [NRT26] gave five- and three-round threshold Schnorr protocols, secure in the ROM under the plain discrete logarithm assumption, that rely on masking output shares and do not natively achieve IA. This masking technique requires each pair of signers to share a dedicated secret key. (IA can be achieved using the additional protocol from [PKN⁺25].) Two recent works [BCL⁺26, BGC⁺25] show that adding such masks to FROST enables a proof of adaptive security in the AGM+ROM.

The bottom line is that all protocols with provable adaptive security are heavier than those only enjoying a proof of static security although there are no obvious adaptive attacks against the latter. Therefore, one could plausibly *conjecture* that a statically secure threshold signature such as FROST is also adaptively secure, and that the lack of a proof of adaptive security is merely an artifact of existing proof techniques. This is because all known separations between static and adaptive security exploit rather recognizable mechanisms, for example by hiding a distinguished subset of parties that a static adversary is unlikely to corrupt but that an adaptive adversary can identify during the execution [CFGN96, CDD⁺04]. A protocol like FROST does not fall into this category.

THRESHOLD SCHNORR AND LDVR. The above informal conjecture that a proof of static security suffices was recently called into question by Crites and Stewart [CS25]. They identified a computational problem (independent of the group $\mathbb{G}$, except for its size p) for which

they prove that solutions exist in certain parameter regimes, and whose efficient solution would compromise the security of all threshold Schnorr signatures that expose $\mathbf{pk}_i = g^{\mathbf{sk}_i}$ for all $i \in [n]$ to an adversary (which can, in particular, choose its corruptions depending on these values). In general, such $\mathbf{pk}_i$ could be public from the outset to enable IA, but protocols such as FROST, Sparkle [CKM23], Lindell’s three-round protocol [Lin24], and a few others [Mak22, BHK⁺24, BLSW24, GJKR07, SS01, KG25, GKMN21] also allow the adversary to reconstruct these values from their view of an execution of the protocol. Crites, Katz, Komlo, Tessaro and Zhu [CKK⁺25] further formalized this problem as what they call the *Low-Dimensional Vector Representation* (LDVR) problem. They prove that, in the AGM+ROM, the hardness of LDVR and the One-More Discrete Logarithm problem imply adaptive security of FROST. They also generalize [CS25] to show that any attack against LDVR translates directly into an equally efficient adaptive attack against any Schnorr signature exposing the $g^{\mathbf{sk}_i}$ ’s. Acharya, Fuchsbauer, O’Neill and Sefranek [AFOS26] developed a corrected analysis of Sparkle, in the ROM, under a new assumption whose generic hardness is implied by LDVR.

Formally, let $\mathbf{v}_i = (1, x_i, x_i^2, \dots, x_i^t) \in \mathbb{F}_p^{t+1}$ be n $(t+1)$ -dimensional vectors mod p associated with the evaluation points used for Shamir secret sharing of the secret key. Also let $\mathbf{v}_0 = (1, 0, \dots, 0)$. Then, the LDVR game allows an adversary to associate with each vector $\alpha \in \mathbb{F}_p^{n+1}$ a challenge $c_\alpha \leftarrow_{\$} \mathbb{F}_p$, and it wins if, for some α , the adversary can produce a set $\text{CS} \subseteq [n]$ with $|\text{CS}| = t_c \leq t$ such that

$$c_\alpha \mathbf{v}_0 + \sum_{i=0}^n \alpha[i] \cdot \mathbf{v}_i \in \text{Span}(\{\mathbf{v}_i\}_{i \in \text{CS}}) .$$

An unconditional proof of hardness for LDVR was given in [CKK⁺25], but outside the parameter regimes covered by that result, the hardness of LDVR remains a conjecture. In particular, the same work conjectures that no attack faster than $O(\sqrt{p})$ exists, and thus that LDVR is exponentially hard.

One should also point out that these issues also extend to other signatures based on linear-identification schemes, such as threshold Okamoto signatures [TZ23].

OUR RESULTS. This work initiates the cryptanalytic study of LDVR and develops attacks invalidating the conjectured asymptotic exponential hardness of the LDVR problem, showing instead the first *polynomial-time* attacks.

Our initial insight is a *general reduction framework* showing, in particular, that to break LDVR with $t_c = t$, it suffices to solve a structured modular subset sum instance in which we need to find a subset $\text{CS} \subseteq [n]$ of size $|\text{CS}| = t_c$ such that

$$\sum_{i \in \text{CS}} x_i^{-1} = c \pmod{p} ,$$

for a random target $c \leftarrow_{\$} \mathbb{F}_p$. Prior work [CKK⁺25] had already noticed different connections to subset sum instances, but the approach required computing discrete logs in finite fields, thus forcing the complexity of the resulting algorithms to be at least subexponential. We

also generalize the statement to the case $t_c < t$. We then leverage this reduction to obtain the following results:

1. **Polynomial attack for $x_i = i$.** This is the most relevant instantiation considered by the RFC [CKGW24], and also corresponds to LDVR as defined in [CKK⁺25]. For this case, we give a polynomial-time algorithm when $n = O(\log^2 p)$ and $t_c = t = O(\log p)$. This directly gives a break of the adaptive security of FROST, Sparkle, and other threshold Schnorr signatures in the aforementioned class for these parameters. We also investigate concrete instantiations of the attack for 256-bit primes, showing that the attack can be mounted with $n \approx 5000$ signers, and also show limitations of the subset-sum approach for smaller numbers of signers.
2. **Worse choices of x_i and backdoors.** We also generalize LDVR and threshold signatures to the use of general evaluation points x_i . Here, we show that malicious choices of the x_i ’s can lead to efficient attacks for a wider range of parameters. We also show how transformations of the evaluation points preserve these attacks when the transformation parameters are known, and discuss the potential use of this observation to plant backdoors.
3. **Separating adaptive and static security.** To our knowledge, our results give the first natural separation between static and adaptive corruption security for practical multi-party protocols that does not rely on an artificially planted or independently selected target committee. This is of independent theoretical interest.

OUTLINE OF THE PAPER. Concretely, Section 2 introduces preliminaries. Section 3 gives our new reductions, and Section 4 uses them to instantiate a concrete polynomial-time attack. We discuss the limitations of this attack template in Section 5, and general evaluation points in Section 6.

AI tools were used in the process of carrying out this research. We give a detailed account of their use in Appendix A.

INTERPRETATION. The presented attacks are not necessarily practical in that they require a larger number of signers than is common in practice. In fact, while $q_{\mathcal{O}}\binom{n}{t} = \Omega(p)$ is necessary by the impossibility result of [CKK⁺25] to break LDVR with constant probability using $q_{\mathcal{O}}$ oracle queries, we give evidence that suggests the attack strategy for evaluation points $x_i = i$ requires even larger parameters. Still, we note that NIST’s call [BP25] explicitly considers an “enormous” regime with $n > 1024$ parties, and our attacks invalidate adaptive security for parameter choices covered by this regime.

However, practical use cases appear to remain covered by the unconditional hardness result from [CKK⁺25], or the number of signers is so low that the classical folklore reduction from static to adaptive security—incurring a multiplicative loss of $\binom{n}{t_c}$ by guessing the corrupted set—suffices to give us ample security margins. As a concrete point of comparison, to our knowledge the largest publicly documented deployment of FROST is Chainflip,⁴ which uses a 100-out-of-150 threshold among its 150 validators for signing vault transactions, and these parameters are well within the regime covered by the unconditional hardness result [CKK⁺25].

⁴ <https://docs.chainflip.io/protocol/protocol-overview>

The question is what to do if we fall outside these parameters, but are not affected by the attack. An optimistic option is to assume the attacks from this paper cannot be improved, and assume that unaffected parameters are safe. The more conservative choice is to simply add masking [BCL⁺26, BGC⁺25] to eliminate the dependence on LDVR.

We also note that since adaptive security accounts for the fact that the corrupted set need not be independent of the public key or the protocol execution, it is the more realistic formal attack model. However, an efficient adaptive attack in the formal model does not necessarily translate into a practical compromise. Our LDVR attacks efficiently identify a set of parties whose corruption enables a forgery; *actually corrupting those parties*, potentially hundreds of them, remains a separate practical obstacle.

OTHER RELATED WORK. LDVR, introduced in [CKK⁺25], is closely related to the problem considered by Crites and Stewart [CS25], and the latter makes a useful connection to *syndrome decoding* explicit. Indeed, if $\mathbf{K} \in \mathbb{F}_p^{(t+1) \times n}$ has as columns the Shamir evaluation vectors $\mathbf{v}_i = (1, x_i, \dots, x_i^t)$, then finding a set CS of size at most $t_c \leq t$ such that a target $\mathbf{v}$ lies in $\text{span}\{\mathbf{v}_i : i \in \text{CS}\}$ is equivalent to finding a vector $\mathbf{e} \in \mathbb{F}_p^n$ of Hamming weight at most t_c such that $\mathbf{K}\mathbf{e} = \mathbf{v}$. Thus, $\mathbf{K}$ can be viewed as a parity-check matrix and $\mathbf{v}$ as a syndrome; for the Vandermonde matrices arising from Shamir sharing, this is a (generalized) Reed–Solomon syndrome-decoding problem. Importantly, Crites and Stewart generate $\mathbf{v}$ uniformly at random to prove existence of a solution, and the resulting problem is an *average-case, random-syndrome* decoding problem. As we are in a parameter range where no efficient decoding algorithm is known, the resulting problem appears hard. In contrast, our attacks crucially exploit the additional algebraic structure of LDVR queries rather than replacing them by truly random targets.

There is also a useful, albeit only moral, analogy with the line of attacks on the so-called ROS problem, which was initially introduced by Schnorr to analyze blind discrete-log signatures [Sch01]. ROS attacks similarly leverage protocol-level interaction to reduce the security of Schnorr-type protocols to a purely algebraic search problem over $\mathbb{F}_p$, rather than to computing discrete logarithms: in particular, the problem amounts to identifying a solvable over-constrained set of linear equations defined by random oracle queries. This is done using generalized-birthday techniques [Wag02] and, in sufficiently high dimension, polynomial-time algorithms [BLL⁺21, JLS25]. Our LDVR attacks have a similar flavor in that security can fail once the adversary can locate an appropriate low-complexity linear representation among many candidates. The source of the combinatorial freedom is different, however, and the resulting algorithmic attacks and insights are different. Thus, the ROS analogy is suggestive rather than directly algorithmic: In both cases, algebraic search assumptions that initially appear hard can become unexpectedly tractable once the number and structure of the available representations are exploited.

Game $\text{LDVR}_{p,\mathcal{X}}^{t_c,t,n}(\mathcal{A}) :$	Oracle $\mathcal{O}(\alpha) :$
$x_0 \leftarrow 0 ; \text{ctr} \leftarrow 0$	$\text{// } \alpha \in \mathbb{F}_p^{n+1}$
$(x_1, \dots, x_n) \leftarrow \mathcal{X}$	$\text{ctr} \leftarrow \text{ctr} + 1$
$(\text{CS}, j) \leftarrow \mathcal{A}^\mathcal{O}(x_1, \dots, x_n)$	$\alpha_{\text{ctr}} \leftarrow \alpha$
$\text{// } \text{CS} \subseteq [n], \text{CS} \leq t_c, j \in [\text{ctr}]$	$c_{\text{ctr}} \leftarrow \mathbb{F}_p$
$\mathbf{w} \leftarrow c_j \mathbf{v}(x_0) + \sum_{i=0}^n \alpha_j[i] \cdot \mathbf{v}(x_i)$	return c_{ctr}
if $\mathbf{w} \in \text{Span}(\{\mathbf{v}(x_i)\}_{i \in \text{CS}})$ then return true	
else return false	

Fig. 1. The LDVR game with parameters $t_c \leq t < n$, for prime modulus p and evaluation point distribution $\mathcal{X}$.

2 Preliminaries

2.1 Notation

Throughout this paper, we write $[N] = \{1, \dots, N\}$ and denote the cardinality of a set S by $|S|$. For a prime p , we let $\mathbb{F}_p$ denote the field of integers modulo p and write $\mathbb{F}_p^* = \mathbb{F}_p \setminus \{0\}$. All linear spans are taken over $\mathbb{F}_p$. For an integer a and a positive integer M , $a \bmod M$ denotes the unique representative in $\{0, 1, \dots, M-1\}$. When comparing a field element with an integer, or reducing it modulo another modulus, we use its least nonnegative integer representative.

We write $x \leftarrow S$ for uniform sampling from a finite set S , and $x \leftarrow \mathcal{X}$ for sampling from a distribution $\mathcal{X}$. We write $\log$ for the base-two logarithm and $\ln$ for the natural logarithm.

For a tuple $\mathbf{x} = (x_1, \dots, x_k) \in \mathbb{F}_p^k$ and scalars $f, d \in \mathbb{F}_p$, we write $\mathbf{x} + f = (x_1 + f, \dots, x_k + f)$ and $d\mathbf{x} = (dx_1, \dots, dx_k)$. When all entries are nonzero, we also write $\mathbf{x}^{-1} = (x_1^{-1}, \dots, x_k^{-1})$. We denote by $\mathbf{e}_0, \dots, \mathbf{e}_t \in \mathbb{F}_p^{t+1}$ the standard basis vectors, with coordinates indexed from 0 to t . For a coefficient vector $\alpha = (\alpha_0, \dots, \alpha_n)$, we also write $\alpha[i] = \alpha_i$.

In all games, the convention is that malformed outputs or outputs violating the indicated constraints cause the game to return **false**.

2.2 The Low-Dimensional Vector Representation Problem

We present here the definition of the *low-dimensional vector-representation (LDVR)* problem from [CKK⁺25], generalized to allow for general evaluation points. We start with a concrete, non-asymptotic, version. The LDVR game, depicted in Figure 1, is parameterized by t_c, t, n with $t_c \leq t < n$. It is also parameterized by a prime p and a *distribution* $\mathcal{X}$ of vectors $\mathbf{x} = (x_1, \dots, x_n) \in (\mathbb{F}_p^*)^n$ of distinct *evaluation points*. The adversary learns a sample $\mathbf{x} \leftarrow \mathcal{X}$ and then gets to query an oracle $\mathcal{O}$ which assigns a random challenge $\mathcal{O}(\alpha) \leftarrow \mathbb{F}_p$ to this vector. Then, the adversary is asked to find one query $\alpha \in \mathbb{F}_p^{n+1}$ such that the vector

$$\mathbf{w} \leftarrow \mathcal{O}(\alpha) \cdot \mathbf{v}(x_0) + \sum_{i=0}^n \alpha[i] \cdot \mathbf{v}(x_i)$$

lies in a subspace of $\mathbb{F}_p^{t+1}$ of dimension at most t_c , where $\mathbf{v}(x) = (1, x, x^2, \dots, x^t)$, with the convention that $x_0 = 0$. More specifically, $\mathcal{A}$ must specify a set $\text{CS} \subseteq [n]$ of size at most t_c , defining the subspace $\text{Span}(\{\mathbf{v}(x_i)\}_{i \in \text{CS}})$ of dimension at most t_c in which $\mathbf{w}$ must lie. We also allow $\mathcal{O}$ to be queried multiple times with the same input; each such query results in a fresh random output.

We denote the advantage of $\mathcal{A}$ in solving the (t_c, t, n) -LDVR problem for modulus p and evaluation point distribution $\mathcal{X}$ as

$$\text{Adv}_{p, \mathcal{X}}^{(t_c, t, n)\text{-ldvr}}(\mathcal{A}) = \Pr[\text{LDVR}_{p, \mathcal{X}}^{t_c, t, n}(\mathcal{A}) \Rightarrow \text{true}].$$

As explained in [CKK⁺25], an efficient (t_c, t, n) -LDVR attack yields an equally efficient adaptive forgery attack against threshold Schnorr protocols that expose the public-key shares g^{sk_i} to the adversary. In this attack, n is the number of signers, $t + 1$ is the reconstruction threshold, and t_c is the number of corrupted parties. Concretely, $\text{sk}_i = p(x_i)$ is the i -th Shamir secret share of the signing key sk , i.e., p is a random degree- t polynomial such that $p(0) = \text{sk}$.

UNCONDITIONAL HARDNESS. Unconditional hardness results were proved in [CKK⁺25], and we briefly state their theorem here, as it is helpful to assess parameter regimes in which every adversary with a given query budget has small success probability. These regimes are complementary to those targeted by our cryptanalysis. Effectively, their result bounds the success probability of any adversary making at most $q_{\mathcal{O}}$ oracle queries, regardless of its computational power. We note that their result was stated for the explicit canonical choice of evaluation points $x_i = i$, but it is very easy to see that their proof does not use any specific property of the evaluation points. Here, we use $(n)_k = n \cdot (n - 1) \cdots (n - k + 1)$ for $k \geq 1$, with $(n)_k = 1$ if $k \leq 0$.

Theorem 2.1 (Unconditional hardness of LDVR problem [CKK⁺25]). *Let $t_c \leq t < n$ be integers. Then, for every $\mathcal{X}$ and prime $p > n$, for any algorithm $\mathcal{A}$ making $q_{\mathcal{O}}$ queries to its oracle,*

$$\text{Adv}_{p, \mathcal{X}}^{(t_c, t, n)\text{-ldvr}}(\mathcal{A}) \leq q_{\mathcal{O}} \cdot \frac{(n)_{2t_c - t}}{(t_c)_{2t_c - t}} \cdot \frac{1}{p}.$$

2.3 Subset Sums

We will consider special structured variants of the modular subset sum problem as part of our attacks against LDVR. Concretely, the variant we consider is defined in Figure 2, and is parametrized by a modulus p , values $t_c < n$, and an instance distribution $\mathcal{X}$ over $(\mathbb{F}_p^*)^n$. The adversary is given an instance $(x_1, \dots, x_n)$ sampled from this distribution, along with a uniformly distributed target c , and needs to now find a subset of size *exactly* t_c of the x_i 's that sums up to c , where all operations are understood modulo p . The advantage of the adversary $\mathcal{A}$ in solving the (t_c, n) -subset sum problem for modulus p and instance distribution $\mathcal{X}$ is defined as

$$\text{Adv}_{p, \mathcal{X}}^{(t_c, n)\text{-ssum}}(\mathcal{A}) = \Pr[\text{SSUM}_{p, \mathcal{X}}^{t_c, n}(\mathcal{A}) \Rightarrow \text{true}].$$

Game $\text{SSUM}_{p, \mathcal{X}}^{t_c, n}(\mathcal{A})$: $(x_1, \dots, x_n) \leftarrow \mathcal{X}$ $c \leftarrow \mathbb{F}_p$ $\text{CS} \leftarrow \mathcal{A}(x_1, \dots, x_n, c)$ $\quad \parallel \text{CS} \subseteq [n], \text{CS} = t_c$ if $c = \sum_{i \in \text{CS}} x_i \pmod{p}$ then $\quad$ return true else return false
--

Fig. 2. The subset sum game with parameters $t_c < n$.

2.4 Polynomial Identities

We introduce an elementary result that relates the roots of any polynomial with its coefficients, as will later be used in Section 3.

Definition 2.2 (Elementary symmetric polynomials). *Let $x_1, \dots, x_n$ be variables. For $k = 0, \dots, n$, the k -th elementary symmetric polynomial is defined by*

$$e_k(x_1, \dots, x_n) = \sum_{1 \leq i_1 < \dots < i_k \leq n} x_{i_1} \cdots x_{i_k},$$

with the convention $e_0(x_1, \dots, x_n) = 1$.

Lemma 2.3 (Vieta's formulas). *Let*

$$f(x) = x^n + a_{n-1}x^{n-1} + \dots + a_1x + a_0$$

be a monic polynomial of degree n with roots $r_1, \dots, r_n$. Then

$$f(x) = \prod_{i=1}^n (x - r_i),$$

and, for $k = 1, \dots, n$,

$$e_k(r_1, \dots, r_n) = (-1)^k a_{n-k}.$$

3 Reductions

Throughout this section we consider an instance of the (t_c, t, n) -LDVR problem as defined in Section 2.2 over a prime p . In this section we look for a better understanding of this problem. Our most important result is a reduction to an instance of a subset sum problem for the case $t = t_c$, as stated in Lemma 3.3. We also show in Lemma 3.4 how known transformations of the evaluation points preserve the ability to solve LDVR.

The rest of the paper will leverage these results in various forms to develop attacks.

TECHNICAL LEMMAS. Our reductions rely on the following two lemmas.

Lemma 3.1. *Let $\text{CS} \subseteq [n]$ have size t_c , and let $x_1, \dots, x_n \in \mathbb{F}_p$ be distinct. Define the polynomial $Q_{\text{CS}}(X)$ as follows:*

$$Q_{\text{CS}}(X) := \prod_{i \in \text{CS}} (X - x_i) = \sum_{i=0}^{t_c} q_i X^i.$$

Then, for any $\mathbf{w} = (\omega_0, \dots, \omega_t) \in \mathbb{F}_p^{t+1}$

$$\mathbf{w} \in \text{Span}(\{\mathbf{v}(x_i)\}_{i \in \text{CS}}) \iff \sum_{i=0}^{t_c} q_i \omega_{i+k} = 0, \quad \forall k = 0, \dots, t - t_c.$$

Proof. ($\implies$) Assume $\mathbf{w} \in \text{Span}(\{\mathbf{v}(x_i)\}_{i \in \text{CS}})$. Then, writing the equations component-wise:

$$\omega_\ell = \sum_{j \in \text{CS}} a_j x_j^\ell, \quad \forall \ell = 0, \dots, t.$$

Then,

$$\begin{aligned} \sum_{i=0}^{t_c} q_i \omega_{i+k} &= \sum_{i=0}^{t_c} q_i \sum_{j \in \text{CS}} a_j x_j^{i+k} \\ &= \sum_{j \in \text{CS}} a_j x_j^k \sum_{i=0}^{t_c} q_i x_j^i = \sum_{j \in \text{CS}} a_j x_j^k \underbrace{Q_{\text{CS}}(x_j)}_{=0} = 0, \quad \forall k = 0, \dots, t - t_c. \end{aligned}$$

($\impliedby$) Assume $\sum_{i=0}^{t_c} q_i \omega_{i+k} = 0, \quad \forall k = 0, \dots, t - t_c$. Since Vandermonde vectors are linearly independent, the $t_c \times t_c$ matrix $(x_j^\ell)_{0 \leq \ell < t_c, j \in \text{CS}}$ is invertible. We can thus uniquely determine $a_j \in \mathbb{F}_p$ satisfying

$$\omega_\ell = \sum_{j \in \text{CS}} a_j x_j^\ell, \quad \forall \ell = 0, \dots, t_c - 1.$$

Define

$$\hat{\omega}_\ell := \sum_{j \in \text{CS}} a_j x_j^\ell, \quad \forall \ell = 0, \dots, t.$$

Since $\hat{\mathbf{w}} \in \text{Span}(\{\mathbf{v}(x_i)\}_{i \in \text{CS}})$, then

$$\sum_{i=0}^{t_c} q_i \hat{\omega}_{i+k} = 0, \quad \forall k = 0, \dots, t - t_c.$$

By definition, $\hat{\omega}_\ell = \omega_\ell \forall \ell = 0, \dots, t_c - 1$. By observing the $t - t_c + 1$ equations they both satisfy, and noting $q_{t_c} = 1$, we can extend the equality $\forall \ell = 0, \dots, t$ inductively, thus proving the lemma. $\square$

Lemma 3.2. Let $\mathbf{w} = (\omega_0, \omega_1, \dots, \omega_t) \in \mathbb{F}_p^{t+1}$, $f \in F_p$, and $d \in \mathbb{F}_p^*$, and let $x_1, \dots, x_n \in \mathbb{F}_p^*$ be distinct. Also let $\text{CS} \subseteq [n]$, $|\text{CS}| = t_c$. Define $\hat{\mathbf{w}} := (\hat{\omega}_0, \dots, \hat{\omega}_t)$, where $\hat{\omega}_\ell := \sum_{j=0}^\ell \binom{\ell}{j} f^j \omega_{\ell-j}$. The following are equivalent:

- (i) $\mathbf{w} \in \text{Span}(\{\mathbf{v}(x_i)\}_{i \in \text{CS}})$,
- (ii) $(\omega_t, \omega_{t-1}, \dots, \omega_0) \in \text{Span}(\{\mathbf{v}(x_i^{-1})\}_{i \in \text{CS}})$,
- (iii) $\hat{\mathbf{w}} \in \text{Span}(\{\mathbf{v}(x_i + f)\}_{i \in \text{CS}})$,
- (iv) $(\omega_0, d\omega_1, \dots, d^t \omega_t) \in \text{Span}(\{\mathbf{v}(dx_i)\}_{i \in \text{CS}})$.

Proof. We prove equivalences with (i).

(i) $\iff$ (ii): Let $\hat{Q}_{\text{CS}}(X) := \prod_{i \in \text{CS}} (X - x_i^{-1}) = \sum_{i=0}^{t_c} \hat{q}_i X^i$. Using Lemma 3.1,

$$\mathbf{w} \in \text{Span}(\{\mathbf{v}(x_i)\}_{i \in \text{CS}}) \iff \sum_{i=0}^{t_c} q_i \omega_{i+k} = 0, \quad \forall k = 0, \dots, t - t_c.$$

$$(\omega_t, \omega_{t-1}, \dots, \omega_0) \in \text{Span}(\{\mathbf{v}(x_i^{-1})\}_{i \in \text{CS}}) \iff \sum_{i=0}^{t_c} \hat{q}_i \omega_{t-(i+k)} = 0, \quad \forall k = 0, \dots, t - t_c.$$

By Lemma 2.3, one can show that $q_{t_c-i} = q_0 \cdot \hat{q}_i$. Using this equality:

$$q_0 \sum_{i=0}^{t_c} \hat{q}_i \omega_{t-(i+k)} = \sum_{i=0}^{t_c} q_{t_c-i} \omega_{t-i-k} = \sum_{i=0}^{t_c} q_i \omega_{t-(t_c-i)-k} = \sum_{i=0}^{t_c} q_i \omega_{i+(t-t_c-k)}.$$

Since $q_0 = (-1)^{t_c} \prod_{i \in \text{CS}} x_i$ includes the product of elements indexed by CS (up to sign), in particular, it is non-zero. Therefore, since $k \mapsto t - t_c - k$ permutes $\{0, 1, \dots, t - t_c\}$ the two systems of equations are equivalent.

(i) $\iff$ (iii): This follows from:

$$\omega_\ell = \sum_{i \in \text{CS}} a_i x_i^\ell \quad \forall \ell = 0, \dots, t \iff \hat{\omega}_\ell = \sum_{i \in \text{CS}} a_i (x_i + f)^\ell \quad \forall \ell = 0, \dots, t.$$

(i) $\iff$ (iv): This follows from:

$$\omega_\ell = \sum_{i \in \text{CS}} a_i x_i^\ell \quad \forall \ell = 0, \dots, t \iff d^\ell \omega_\ell = \sum_{i \in \text{CS}} a_i (d \cdot x_i)^\ell \quad \forall \ell = 0, \dots, t. \square$$

REDUCTION TO SUBSET SUM. Let $\mathcal{X}$ be an evaluation point distribution over $(\mathbb{F}_p^*)^n$ and define $\mathcal{X}^{-1}$ to be the distribution that samples $\mathbf{x} \leftarrow \mathcal{X}$ and then returns $\mathbf{x}^{-1}$. The following lemma shows that any adversary against the (t_c, n) -SSUM problem with instance distribution $\mathcal{X}^{-1}$ can be turned into an adversary against (t_c, t, n) -LDVR with evaluation point distribution $\mathcal{X}$, for the same prime modulus p , whenever $t = t_c$.

Lemma 3.3. Suppose $1 < t = t_c < n$, and let $p > n$ be prime. Further, let $\mathcal{X}$ be a distribution over $(\mathbb{F}_p^*)^n$ sampling a tuple of distinct non-zero elements from $\mathbb{F}_p$. Then, for every adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$, nearly as efficient as $\mathcal{A}$, such that

$$\text{Adv}_{p, \mathcal{X}^{-1}}^{(t_c, n)\text{-ssum}}(\mathcal{A}) = \text{Adv}_{p, \mathcal{X}}^{(t_c, t, n)\text{-ldvr}}(\mathcal{B}).$$

Proof. Suppose $\mathbf{w} = (\omega_0, \omega_1, 0, \dots, 0)$ with $\omega_1 \neq 0$ and $\text{CS} \subseteq [n]$ with $|\text{CS}| = t_c = t$, then Lemma 3.1 and Lemma 3.2 yield

$$\begin{aligned} \mathbf{w} \in \text{Span}(\{\mathbf{v}(x_i)\}_{i \in \text{CS}}) &\iff (0, \dots, 0, \omega_1, \omega_0) \in \text{Span}(\{\mathbf{v}(x_i^{-1})\}_{i \in \text{CS}}) \\ &\iff \omega_1 q_{t_c-1} + \omega_0 q_{t_c} = 0 \end{aligned}$$

where $q_{t_c-1} = -\sum_{i \in \text{CS}} \frac{1}{x_i}$ and $q_{t_c} = 1$ are the coefficients from the polynomial $\prod_{i \in \text{CS}} (X - x_i^{-1}) = \sum_{a=0}^{t_c} q_a X^a$. This means that

$$\mathbf{w} \in \text{Span}(\{\mathbf{v}(x_i)\}_{i \in \text{CS}}) \iff \sum_{i \in \text{CS}} \frac{1}{x_i} = \frac{\omega_0}{\omega_1}. \quad (1)$$

Thus, the adversary $\mathcal{B}$ operates as follows, given $\mathbf{x} = (x_1, \dots, x_n)$. It makes a single query $\mathcal{O}(\alpha)$ for any $\alpha \in \mathbb{F}_p^{n+1}$ such that $\sum_{j=0}^n \alpha[j] \cdot \mathbf{v}(x_j) = (0, 1, 0, \dots, 0)$. Therefore, if this query returns $c \leftarrow \mathbb{F}_p$, we have

$$\mathbf{w} \leftarrow c\mathbf{v}(x_0) + \sum_{j=0}^n \alpha[j] \cdot \mathbf{v}(x_j) = (\omega_0 = c, \omega_1 = 1, 0, \dots, 0).$$

We can thus now compute $\text{CS} \leftarrow \mathcal{A}(\mathbf{x}^{-1}, c = \omega_0/\omega_1)$, and return $(\text{CS}, 1)$ as $\mathcal{B}$'s output. Then, $\mathcal{B}$ wins if and only if $\mathcal{A}$ succeeds by Equation (1). $\square$

RELATING DISTRIBUTIONS. We next show that certain transformations of the evaluation points preserve the ability to solve LDVR when the transformation is known. We discuss their potential use in disguising bad choices of evaluation points in Section 6.2.

Fix a prime $p > n$ and parameters $t_c \leq t < n$. Let $\mathcal{X}$ be a distribution over vectors of n distinct elements of $\mathbb{F}_p^*$. Define $\mathcal{X}_y$ by sampling $\mathbf{x}$ from $\mathcal{X}$, then sampling (f, d) according to a conditional distribution $\mathcal{Y}(\cdot \mid \mathbf{x})$ supported on pairs in $\mathbb{F}_p \times \mathbb{F}_p^*$ satisfying $x_i^{-1} + f \neq 0$ for all $i \in [n]$, and outputting

$$\hat{\mathbf{x}} = d \cdot (\mathbf{x}^{-1} + f)^{-1}.$$

The operations are component-wise, and the resulting evaluation points are again distinct and non-zero. For the following lemma to be formally correct, we extend the LDVR experiment to allow the evaluation-point sampler to provide auxiliary information to the adversary. In the experiment associated with $\mathcal{X}_y$, this information consists of the sampled pair (f, d) . (We leave this extension implicit.)

Lemma 3.4. *Under the above conventions, every adversary $\mathcal{A}$ for (t_c, t, n) -LDVR with evaluation-point distribution $\mathcal{X}$ yields an adversary $\mathcal{B}$ for the same problem with distribution $\mathcal{X}_y$ and auxiliary information (f, d) , with polynomial overhead and the same number of oracle queries, such that*

$$\text{Adv}_{p, \mathcal{X}}^{(t_c, t, n)\text{-ldvr}}(\mathcal{A}) = \text{Adv}_{p, \mathcal{X}_y}^{(t_c, t, n)\text{-ldvr}}(\mathcal{B}).$$

Proof. The adversary $\mathcal{B}$, on input $\hat{\mathbf{x}}$, f , and d , runs $\mathcal{A}$ on input $\mathbf{x} = ((d^{-1}\hat{\mathbf{x}})^{-1} - f)^{-1}$, and then:

1. Whenever $\mathcal{A}$ makes its j -th query α_j , $\mathcal{B}$ proceeds as follows:
 - (a) $\mathcal{B}$ defines its corresponding $\nu_j = (\nu_0^{(j)}, \dots, \nu_t^{(j)}) := \sum_{i=0}^n \alpha_j[i] \cdot \mathbf{v}(x_i)$.
 - (b) For every ν_j , $\mathcal{B}$ defines $\hat{\nu}_j := (\hat{\nu}_0^{(j)}, \dots, \hat{\nu}_t^{(j)})$, such that $\hat{\nu}_\ell^{(j)} := d^\ell \cdot \sum_{r=0}^{t-\ell} \binom{t-\ell}{r} f^r \nu_{\ell+r}^{(j)}$, and picks $\hat{\alpha}_j$ such that $\hat{\nu}_j = \sum_{i=0}^n \hat{\alpha}_j[i] \cdot \mathbf{v}(\hat{x}_i)$, where $\hat{x}_0 = 0$.
 - (c) $\mathcal{B}$ makes its j -th oracle call $\hat{\alpha}_j$, which returns c_j , and gives c_j back to $\mathcal{A}$.
2. When $\mathcal{A}$ returns a pair (CS, j) , $\mathcal{B}$ returns the same pair (CS, j) .

The simulation is correct, since the reconstructed $\mathbf{x}$ follows the distribution of $\mathcal{X}$, and each forwarded query supplies a fresh uniform response in $\mathbb{F}_p$. We claim that $\mathcal{B}$ succeeds if and only if $\mathcal{A}$ succeeds. Indeed, using Lemma 3.2 with $|\text{CS}|$ instead of t_c (we assume this is larger than 0, as otherwise the proof is immediate), for the final j returned by the adversary, we have

$$\begin{aligned}
(\nu_0^{(j)} + c_j, \nu_1^{(j)}, \dots, \nu_t^{(j)}) &\in \text{Span}(\{\mathbf{v}(x_i)\}_{i \in \text{CS}}) && \iff \\
(\nu_t^{(j)}, \dots, \nu_1^{(j)}, \nu_0^{(j)} + c_j) &\in \text{Span}(\{\mathbf{v}(x_i^{-1})\}_{i \in \text{CS}}) && \iff \\
(d^{-t} \cdot \hat{\nu}_t^{(j)}, \dots, d^{-1} \cdot \hat{\nu}_1^{(j)}, \hat{\nu}_0^{(j)} + c_j) &\in \text{Span}(\{\mathbf{v}(x_i^{-1} + f)\}_{i \in \text{CS}}) && \iff \\
(\hat{\nu}_0^{(j)} + c_j, d^{-1} \cdot \hat{\nu}_1^{(j)}, \dots, d^{-t} \cdot \hat{\nu}_t^{(j)}) &\in \text{Span}(\{\mathbf{v}((x_i^{-1} + f)^{-1})\}_{i \in \text{CS}}) && \iff \\
(\hat{\nu}_0^{(j)} + c_j, \hat{\nu}_1^{(j)}, \dots, \hat{\nu}_t^{(j)}) &\in \text{Span}(\{\mathbf{v}(d(x_i^{-1} + f)^{-1})\}_{i \in \text{CS}})
\end{aligned}$$

Note that $\mathcal{A}$ wins iff $\omega_j = \nu_j + c_j e_0 \in \text{Span}(\{\mathbf{v}(x_i)\}_{i \in \text{CS}})$, which happens iff $\hat{\omega}_j = \hat{\nu}_j + c_j e_0 \in \text{Span}(\{\mathbf{v}(d(x_i^{-1} + f)^{-1})\}_{i \in \text{CS}})$, which in turn happens iff $\mathcal{B}$ wins the game. $\square$

4 A Polynomial-Time Attack Against LDVR

The main result of this section is the following theorem, which attacks the canonical version of LDVR as defined in [CKK⁺25], and as needed for standard instantiations of threshold signatures.

Theorem 4.1 (LDVR attack). *Let p be a sufficiently large prime. Then, there exist $t_c = t = \Theta(\log p)$ and $n = \Theta(\log^2 p)$, and an adversary $\mathcal{A}$ running in polynomial time in $\log p$ making a single query to its oracle such that*

$$\text{Adv}_{p, \mathcal{X}}^{(t_c, t, n)\text{-ldvr}}(\mathcal{A}) = \Omega\left(\frac{1}{\log p}\right),$$

where $\mathcal{X}$ is the canonical distribution that outputs $(1, 2, \dots, n)$ with probability one.

By Lemma 3.3, it is sufficient to give an adversary $\mathcal{B}$ for the modular subset-sum problem for $\mathcal{X}^{-1}$ which outputs $(1, 2^{-1}, \dots, n^{-1})$, where the inverses are computed modulo p . For this reason, Section 4.2 will develop an efficient attacker $\mathcal{B}$ for this subset-sum variant. Moreover, Section 4.1 develops the high level idea first. Furthermore, we discuss below in Section 4.3 some concrete-parameter aspects of the attack, along with optimizations.

Remark 4.2. We note that the LDVR adversary we build for Theorem 4.1 only makes a single oracle query. We can in fact boost its probability to be $\Omega(1)$ easily by simply running the adversary $\Omega(\log p)$ times.

4.1 Starting Point: A Toy Attack Idea

It is helpful to first give an attack idea that does not directly yield a valid attack. To this end, let $p > 3$, set $m = \lceil \log p \rceil$, and let $L = 3 \cdot 2^{m-1}$. For $z \in \mathbb{F}_p$, write its least nonnegative integer representative as $z = \sum_{j=0}^{m-1} z_j 2^j$, where $z_j \in \{0, 1\}$. Now, pick

$$\text{CS}_z = \{3^{1-z_j} 2^{m-1-j} \mid j \in \{0, 1, \dots, m-1\}\},$$

assuming $n \geq 3 \cdot 2^{m-1}$. Over the integers, we have

$$\begin{aligned} \sum_{i \in \text{CS}_z} \frac{L}{i} &= \sum_{j=0}^{m-1} 3^{z_j} 2^j = \sum_{j=0}^{m-1} (1 + 2z_j) 2^j \\ &= 2z + \sum_{j=0}^{m-1} 2^j = 2z + 2^m - 1. \end{aligned}$$

Therefore, given a target $c \in \mathbb{F}_p$, we let $z = (Lc - 2^m + 1)2^{-1} \pmod{p}$, where the inverse is taken modulo p . With $\text{CS} = \text{CS}_z$, this gives $\sum_{i \in \text{CS}} L/i = Lc \pmod{p}$. Since $p > 3$, both L and all selected indices are nonzero modulo p . Multiplying by L^{-1} therefore yields

$$\sum_{i \in \text{CS}} i^{-1} = c \pmod{p}.$$

This does not quite work, however, as an attack on the canonical instance. To ensure that $\text{CS} \subseteq [n]$ for every target, we require $n \geq 3 \cdot 2^{m-1}$, a bound between $3p/2$ and $3p$. However, the canonical evaluation points must satisfy $n < p$ to be nonzero and distinct. Reducing the elements of CS_z modulo p does not resolve the size issue: some possible indices are already of order p , so accommodating them still requires a large n .

The attack below overcomes these issues by applying a similar idea modulo several small primes, whose product P is close to p . This leads directly to an algorithm that always succeeds modulo P . A few additional steps then turn this into an algorithm that succeeds modulo p with probability P/p .

4.2 Full Attack (Proof of Theorem 4.1)

We will resort to Lemma 3.3, and thus here just give an adversary $\mathcal{B}$ such that $\text{Adv}_{p, \mathcal{X}^{-1}}^{(t_c, n)\text{-ssum}}(\mathcal{B}) = \Omega(1/\log p)$, where $\mathcal{X}^{-1}$ outputs $(1, 2^{-1}, 3^{-1}, \dots, n^{-1})$ with probability 1, for a suitable n and $t_c < n$ to be specified below.

We let $\mathcal{Q}$ be a set of primes larger than three. We let $P = \prod_{q \in \mathcal{Q}} q$, and set $D = LP$, with $L = 3 \cdot 2^{m-1}$ as above, for m such that $\lceil \log q \rceil \leq m$ for all $q \in \mathcal{Q}$. We further assume that $LP < p$. Then, we let $B = \ln p$ and further restrict the choice so that $q > B/2$ for every $q \in \mathcal{Q}$. Further, assume that $B \geq 2e^4$. For now, we simply assume that we can pick all of these values—below, we explain that this is indeed possible.

THE BASIC IDEA. Assume that we are given $(z_q)_{q \in \mathcal{Q}}$ such that $z_q \in \mathbb{F}_q$. Let $z_q = \sum_{j=0}^{m-1} z_{q,j} 2^j$ be the binary decomposition of z_q , interpreted as an integer, which exists since $z_q < q \leq 2^m$. Now, we define the set

$$\text{CS}_q = \{q \cdot 3^{1-z_{q,j}} 2^{m-1-j} \mid j \in \{0, 1, \dots, m-1\}\}.$$

Also define $\text{CS} = \bigcup_{q \in \mathcal{Q}} \text{CS}_q$. Then,

$$\begin{aligned} \sum_{i \in \text{CS}_q} \frac{LP}{i} &= \frac{P}{q} \sum_{j=0}^{m-1} 3^{z_{q,j}} 2^j \\ &= \frac{P}{q} \sum_{j=0}^{m-1} (1 + 2z_{q,j}) 2^j \\ &= \frac{P}{q} (2z_q + \sum_{j=0}^{m-1} 2^j) = \frac{P}{q} (2z_q + 2^m - 1) \pmod{q}. \end{aligned} \tag{2}$$

Also note that this value is 0 mod $q' \in \mathcal{Q} \setminus \{q\}$. Now, assume that for some $A \in \mathbb{Z}$, we set $z_q = (A(P/q)^{-1} - 2^m + 1)2^{-1} \pmod{q}$, for all $q \in \mathcal{Q}$, where both inverses are computed mod q . Then, for every $q \in \mathcal{Q}$, we have

$$\sum_{i \in \text{CS}} \frac{LP}{i} = \sum_{i \in \text{CS}_q} \frac{LP}{i} = A \pmod{q}.$$

By the Chinese Remainder Theorem,

$$\sum_{i \in \text{CS}} \frac{LP}{i} = A \pmod{P}.$$

This is *almost* what we want, in that it ensures the subset-sum constraint holds modulo P , whereas our goal is to satisfy it modulo p .

ATTACK DESCRIPTION. The following algorithm gives the full attack against SSUM. Here, $\text{BinDecomp}_m(z)$ returns the binary decomposition $(z_j)_{j \in \{0, 1, \dots, m-1\}}$ of an integer $0 \leq z < 2^m$, i.e., $z = \sum_{j=0}^{m-1} z_j 2^j$.

Adversary $\mathcal{B}(1, 2^{-1}, \dots, n^{-1}, c)$:

$A \leftarrow D \cdot c \pmod{p}$

foreach $q \in \mathcal{Q}$ **do**

$z_q = (A(P/q)^{-1} - 2^m + 1)2^{-1} \pmod{q}$

// Inverses above are to be interpreted modulo q

$(z_{q,j})_{j \in \{0, 1, \dots, m-1\}} \leftarrow \text{BinDecomp}_m(z_q)$ // $z_q = \sum_{j=0}^{m-1} z_{q,j} 2^j$

$\text{CS}_q \leftarrow \{q \cdot 3^{1-z_{q,j}} 2^{m-1-j} \mid j \in \{0, 1, \dots, m-1\}\}$

$\text{CS} \leftarrow \bigcup_{q \in \mathcal{Q}} \text{CS}_q$

$A' \leftarrow \sum_{i \in \text{CS}} \frac{LP}{i}$ // evaluated in $\mathbb{Z}$

if $A = A'$ **then return** CS **else return** fail.

For this algorithm, we consider in particular the case of $t_c = t = m|\mathcal{Q}|$. Indeed, we are going to pick n large enough below so that each CS_q has m distinct non-zero elements at most n , and different CS_q are disjoint, since an index in CS_q has q as its unique prime factor greater than 3.

CORRECTNESS. Assume $A = A'$, i.e., the adversary does not abort. Then we also have,

$$\sum_{i \in \text{CS}} \frac{LP}{i} = A ,$$

over $\mathbb{Z}$, and thus also over $\mathbb{F}_p$, after taking both sides modulo p . Because the inverse D^{-1} modulo p of $D = LP$ is well-defined (since p does not divide D), and $\frac{LP}{i} = LPi^{-1} \pmod{p}$, then

$$\sum_{i \in \text{CS}} i^{-1} = c \pmod{p} .$$

SUCCESS PROBABILITY. Let us denote by $\alpha(A)$ the value $A' \in \mathbb{Z}$ produced by the above algorithm on input $A \in \mathbb{F}_p$, i.e., $\alpha: \{0, 1, \dots, p-1\} \rightarrow \mathbb{Z}$. We now want to show that $\alpha(A) = A$ holds for exactly P different values of A , thus leading to the algorithm succeeding with probability P/p .

Lemma 4.3. *The mapping α satisfies the following properties:*

- (i) $0 \leq \alpha(A) < p$ for all $A \in \mathbb{F}_p$,
- (ii) $\alpha(A) = A \pmod{P}$,
- (iii) $A = \hat{A} \pmod{P}$ implies $\alpha(A) = \alpha(\hat{A})$.

Proof. To prove (i), first define $\Phi_q = \sum_{i \in \text{CS}_q} \frac{LP}{i}$. Note that every $i \in \text{CS}_q$ satisfies $i \in \{3q2^{m-1-j}, q2^{m-1-j}\}$ for some $j = 0, \dots, m-1$, and thus

$$\frac{LP}{i} \leq \frac{32^{m-1}P}{q2^{m-1-j}} = \frac{3P}{q}2^j ,$$

and thus

$$\Phi_q \leq \frac{3P}{q} \sum_{j=0}^{m-1} 2^j \leq \frac{3P2^m}{q} = \frac{2LP}{q} .$$

Now, because $(B/2)^{|\mathcal{Q}|} \leq \prod_{q \in \mathcal{Q}} q < p = e^B$ and thus $|\mathcal{Q}| \leq B/\ln(B/2)$, and thus

$$\alpha(A) = \sum_{i \in \text{CS}} \frac{LP}{i} = \sum_{q \in \mathcal{Q}} \Phi_q \leq 2LP \sum_{q \in \mathcal{Q}} \frac{1}{q} \leq |\mathcal{Q}| \frac{4LP}{B} \leq \frac{4LP}{\ln(B/2)} \leq LP < p ,$$

for $B \geq 2e^4$.

As for (ii), it was essentially argued above, and follows from Equation 2. Finally, (iii) follows from the simple observation that the value of $\alpha(A)$ only depends on the reduction modulo q , for every $q \in \mathcal{Q}$, of the input. $\square$

The first statement shows that α can be interpreted as a map $\alpha: \mathbb{F}_p \rightarrow \mathbb{F}_p$. The third statement in the lemma means that α is constant over equivalence classes induced by equivalence modulo P . Then, let $\mathcal{R} = \text{range}(\alpha) \subseteq \mathbb{F}_p$ be the range of α . Then, (ii) and (iii) imply that $|\mathcal{R}| = P$. Also, if $A \in \mathcal{R}$, then $\alpha(A) = A$. Therefore, because A is uniform over $\mathbb{F}_p$, the algorithm succeeds with probability exactly $|\mathcal{R}|/p = P/p$.

PICKING PARAMETERS. Our choice of parameters relies on repeated use of the prime number theorem, which we re-state for convenience, including a convenient corollary.

Theorem 4.4 (Prime Number Theorem [Apo76]). *Let $\mathcal{Q}_x$ be the set of primes at most x . Let $\pi(x) = |\mathcal{Q}_x|$ and $\vartheta(x) = \sum_{p \in \mathcal{Q}_x} \ln p$. Then, $\pi(x) = (1 + o(1)) \frac{x}{\ln x}$ and $\vartheta(x) = (1 + o(1))x$ for $x \rightarrow \infty$.*

Corollary 4.5. *For every $0 < \alpha < \beta$, we have $\pi(\beta x) - \pi(\alpha x) = (\beta - \alpha + o(1)) \frac{x}{\ln x}$ and $\vartheta(\beta x) - \vartheta(\alpha x) = (\beta - \alpha + o(1))x$.*

Let $B = \ln p$. Then we let $R = 2^m$ be the smallest power of two such that $R \geq 2B$, i.e., $m = \lceil \log(2B) \rceil$. Then, we use the following lemma to pick $\mathcal{Q}$. Below, we briefly argue that this set can be found in polynomial-time and explain the consequence of the lemma for the success probability of the algorithm.

Lemma 4.6. *For all sufficiently large p , we can give a set of primes $\mathcal{Q} \subseteq (B/2, 2B] \cap \mathbb{Z}$ such that $\lceil \log q \rceil \leq m$ for all $q \in \mathcal{Q}$ and*

$$\frac{p}{2L} < P = \prod_{q \in \mathcal{Q}} q < \frac{p}{L}.$$

Proof. Let $q_1 < q_2 < \dots < q_N$ be all primes in $(B/2, 2B]$. Let $\mathcal{I} \subseteq [N]$, and define

$$P(\mathcal{I}) = \prod_{i \in \mathcal{I}} q_i.$$

Let $k \in [N]$ be largest such that $P([k]) < p/L$ (below, we show that k is well defined). We are also going to show that $P(\{N - k + 1, \dots, N\}) > p/L$. Then, we can build a sequence $\mathcal{I}_0, \mathcal{I}_1, \dots, \mathcal{I}_\ell \subseteq [N]$ of sets of size k where $\mathcal{I}_0 = \{N - k + 1, \dots, N\}$ and $\mathcal{I}_\ell = [k]$, where in each step i we obtain $\mathcal{I}_{i+1}$ by replacing some $j \in \mathcal{I}_i$ with its predecessor $j - 1 \notin \mathcal{I}_i$ (such j must exist, unless we reached $[k]$). Let then i^* be such that $P(\mathcal{I}_{i^*}) \geq p/L$, but $P(\mathcal{I}_{i^*+1}) < p/L$. Let j be such that it was replaced by $j - 1$ in this transition. Note that if $q_{j-1} > B$, then $q_{j-1}/q_j > 1/2$, since $q_j \leq 2B$. If $q_{j-1} \leq B$, then Bertrand's postulate tells us that there exists a prime in the interval $(q_{j-1}, 2q_{j-1})$, and thus q_j must be in this interval. Therefore,

$$P(\mathcal{I}_{i^*+1}) = \frac{q_{j-1}}{q_j} P(\mathcal{I}_{i^*}) > \frac{1}{2} P(\mathcal{I}_{i^*}) \geq \frac{p}{2L}.$$

Thus, we can set $\mathcal{Q} = \{q_j \mid j \in \mathcal{I}_{i^*+1}\}$.

To show the existence of a suitable k , note that because $3B \leq L < 6B$,

$$\ln(p/L) = B - \ln L = B - \ln B + O(1) = B - O(\ln B)$$

and thus, by Corollary 4.5,

$$\sum_{i=1}^N \ln q_i = \vartheta(2B) - \vartheta(B/2) = \left(\frac{3}{2} + o(1)\right) B > \ln(p/L)$$

for sufficiently large p , or, equivalently, $\prod_{i=1}^N q_i > p/L$. Further, $q_1 < p/L$ follows easily by the existence of a prime in $(B/2, 0.75B)$ for sufficiently large p . Thus, we can pick the largest $1 \leq k < N$ such that $P([k]) < p/L$.

Now, observe that $P([k])q_{k+1} \geq p/L$, by maximality, which means that $P([k]) \geq p/(Lq_{k+1}) \geq p/(2LB)$, because $q_{k+1} \leq 2B$. We can use this, in conjunction with $P([k]) < p/L$, to conclude that

$$B - \ln(L) - \ln(B) - \ln(2) \leq \sum_{i=1}^k \ln q_i < B - \ln(L) ,$$

which means that $\sum_{i=1}^k \ln q_i = (1 + o(1))B$, and because $\ln(q_i) = (1 + o(1))\ln(B)$,

$$k = (1 + o(1)) \frac{B}{\ln(B)} . \quad (3)$$

By Corollary 4.5, we note that

$$N = \pi(2B) - \pi(B/2) = \left(\frac{3}{2} + o(1)\right) \frac{B}{\ln B} . \quad (4)$$

and thus $k > N/2$ for sufficiently large p (which we assume). In turn, this means that

$$\frac{P(\{N - k + 1, \dots, N\})}{P([k])} = \frac{P(\{k + 1, \dots, N\})}{P([N - k])} ,$$

since all primes q_i for $i = N - k + 1, \dots, k$ are present in both the numerator and the denominator. Note that $N - k = (1/2 + o(1))B/\ln(B)$. By Corollary 4.5, there are $(0.6 + o(1))B/\ln(B)$ primes in $(1.4B, 2B]$ and as many in $(B/2, 1.1B]$. Therefore, all primes in the numerator are larger than $1.4B$, whereas those in the denominator are at most $1.1B$. Thus,

$$\frac{P(\{N - k + 1, \dots, N\})}{P([k])} \geq \left(\frac{1.4}{1.1}\right)^{N-k} > 2B ,$$

because $c^{\Theta(B/\ln B)} > 2B$ for $c > 1$ and sufficiently large p . Therefore,

$$P(\{N - k + 1, \dots, N\}) > 2B \cdot P([k]) \geq 2B \cdot \frac{p}{2LB} = \frac{p}{L} .$$

This concludes the proof. □

Note that this leads to success probability of $\frac{P}{p} \geq \frac{1}{2L} = \frac{1}{3 \cdot 2^m} = \Omega\left(\frac{1}{\ln p}\right)$. Furthermore, the running time to find $\mathcal{Q}$ is polynomial in $B = \ln p$. This is because of the following:

1. Finding all primes $q_1 < q_2 < \dots < q_N$ in $(B/2, 2B]$ can be done in time polynomial in B .
2. Finding the largest k such that $\prod_{i=1}^k q_i < p/L$ can easily be done by expanding the product for up to N steps, where N is $O(B/\ln B)$.
3. Iterating from $\mathcal{I}_0 = \{N - k + 1, \dots, N\}$ to $\mathcal{I}_\ell = [k]$ requires decreasing the index of every selected prime in $\mathcal{I}_0$ exactly $N - k$ times, and thus this takes $k(N - k)$ steps, which is $O(B^2/\ln(B)^2)$.

We also note that in practice this computation only depends on the order p , and in the context of attacking the LDVR problem as a means to break threshold signatures, p is determined by the elliptic curve. Therefore, finding $\mathcal{Q}$ only needs to be done once per curve.

We also note that our attack needs n to accommodate all possible values picked in the set CS, and for this it is enough to pick

$$n = \lceil 2BL \rceil = \Theta(B^2)$$

whereas the size of the output set is

$$t_c = |\mathcal{Q}| \cdot m = \Theta(B) .$$

4.3 Concrete Parameters and Optimizations

Inferring concrete choices of parameters from Theorem 4.1 leads to very large parameters, in particular a very large n . However, this is mostly because its proof only aims for an asymptotic statement. For a concrete prime p , one can obtain a substantially better instantiation. For each $q \in \mathcal{Q}$, let U_q be an upper bound on the local integer sum $\sum_{i \in \text{CS}_q} \frac{qL}{i}$, valid for every target residue. We require the following properties:

1. $q > 3$ for every $q \in \mathcal{Q}$.
2. $P = \prod_{q \in \mathcal{Q}} q < p$.
3. $P \sum_{q \in \mathcal{Q}} \frac{U_q}{q} < p$.

Condition 3 ensures that the total integer sum $\sum_{i \in \text{CS}} LP/i$ is smaller than p , as needed for the success-probability argument. For the binary construction above, $L = 3 \cdot 2^{m-1}$ and we may take $U_q = 2L$, recovering the condition $2LP \sum_{q \in \mathcal{Q}} 1/q < p$. For each m , we search over the primes $3 < q < 2^m$ and choose $\mathcal{Q}$ so as to maximize P , subject to conditions 2 and 3. We then choose the smallest m for which the resulting success probability P/p is sufficiently large.

For the prime-order subgroup of Curve25519, $m = 7$ is not sufficient, since even the product of all primes in $\{5, 7, \dots, 127\}$ is smaller than 2^{160} . For $m = 8$, we can take

$$\mathcal{Q} = \{q \text{ prime} : 61 \leq q \leq 251\} \setminus \{71, 73, 173\}.$$

This gives $P/p \approx 1/194$, while $2LP \sum_{q \in \mathcal{Q}} 1/q \approx 0.9993p < p$. The largest value appearing in CS is qL for some $q \in \mathcal{Q}$. Since 251 is the largest prime in $\mathcal{Q}$, it therefore suffices to set $n = 3 \cdot 251 \cdot 2^7 = 96384$.

CONCRETE-PARAMETER OPTIMIZATION. In fact, we can reduce the number of necessary parties to $n = 5020$ by choosing L and the local representations more carefully. This task is easier if we fix the prime we are targeting, and we do not want to provide an asymptotic instantiation. For each $q \in \mathcal{Q}$ and target residue $z \in \mathbb{F}_q$, it suffices to find a set $\mathcal{D}_q$ of divisors of L such that

$$\sum_{i \in \text{CS}_q} \frac{LP}{i} = \frac{P}{q} \sum_{d \in \mathcal{D}_q} d = z \pmod{q}, \quad \text{CS}_q = \left\{ \frac{qL}{d} : d \in \mathcal{D}_q \right\}.$$

As before, this contribution is zero modulo every $q' \in \mathcal{Q} \setminus \{q\}$.

A concrete choice is to set $L = 180$. A direct exhaustive check shows that every integer in $\{117, \dots, 405\}$ can be written as the sum of six distinct elements of

$$\mathcal{D} = \{9, 10, 12, 15, 18, 20, 30, 36, 45, 60, 90, 180\}.$$

Indeed, for a given $z \in \mathbb{F}_q$, let r_q be the unique integer in $\{117, \dots, q + 116\}$ satisfying $r_q = z(P/q)^{-1} \pmod{q}$. Since every $q \in \mathcal{Q}$ is at most 251, we have $r_q \leq 367$, and hence there exists a six-element set $\mathcal{D}_q \subseteq \mathcal{D}$ whose elements sum to r_q . Consequently,

$$\text{CS}_q \subseteq \{q, 2q, 3q, 4q, 5q, 6q, 9q, 10q, 12q, 15q, 18q, 20q\}$$

has size six and gives the required residue modulo q .

We use the same set $\mathcal{Q}$ as above, so conditions 1 and 2 remain satisfied. For the new representations, we have

$$\sum_{i \in \text{CS}_q} \frac{qL}{i} = \sum_{d \in \mathcal{D}_q} d = r_q \leq q + 116.$$

Thus we may take $U_q = q + 116$, and condition 3 follows from

$$P \sum_{q \in \mathcal{Q}} \frac{U_q}{q} = P \left(|\mathcal{Q}| + 116 \sum_{q \in \mathcal{Q}} \frac{1}{q} \right) \approx 0.3261p < p.$$

The success probability therefore remains $P/p \approx 1/194$. Finally, the largest possible element of CS is $20 \cdot 251$, and thus $n = 5020$ suffices and $t = t_c = 6 \cdot |\mathcal{Q}| = 6 \cdot 34 = 204$, as it is easy to verify that the CS_q 's are disjoint, and consist of elements at most 5020.

5 Limits of Subset-sum Attacks

This section investigates the limits of the attack template presented in Section 4. As an illustrative example, consider the concrete case where p is any 256-bit prime number, $n = 256 \approx \log_2(p)$, $t = t_c = 128$ and $\mathcal{X}$ is the canonical distribution that outputs $(1, 2, \dots, n)$ with probability one. [CS25, Th. 1] shows that a solution exists with probability > 0.047 , but that is not the case for the attack described in Section 4. The reason is that the existence bound in [CS25] applies to the case where an adversary queries a uniformly random ambient

vector to the LDVR oracle, while the reduction in Lemma 3.3 enforces the resulting vector ω to have zeroes in all but two components. As a result, for those parameters, we will show that no adversary can win the subset sum game on $(1, 1/2, \dots, 1/n)$ modulo p with probability higher than $2^{-106.5}$. In fact, this is a consequence of a more general derivation from this section.

Concretely, we count the number of distinct values one can obtain when choosing any possible t -subset of such a subset-sum instance. Roughly, the idea is to consider the range of values one can obtain when the sum is taken over the rationals, which can be easily upper and lower bounded. Then, we clear denominators and compute the possible residues modulo $q_j^{v_j}$ for many q_j to obtain a tighter bound. In Table 1 we can see that only a very small portion of values can be attained. Throughout we assume $p > n$ and fix $t_c = t = n/2$ for simplicity.

NOTATION. We write $\mathbb{Z}$ and $\mathbb{Q}$ for the integers and rationals, respectively, and $\mathbb{Z}/M\mathbb{Z}$ for the integers modulo a positive integer M , represented by $\{0, \dots, M-1\}$. Write

$$L_n := \text{lcm}(1, \dots, n), \quad H_n := \sum_{i=1}^n \frac{1}{i},$$

where lcm denotes the least common multiple, let c denote a target, and call c *achievable* if $c = \sum_{i \in \text{CS}} 1/i$ for some t -subset $\text{CS} \subseteq [n]$, with the sum interpreted in $\mathbb{Q}$ or $\mathbb{F}_p$ as appropriate. We then define both

$$\Delta := |\{c \in \mathbb{Q} : c \text{ achievable}\}|,$$

as well as

$$\delta := \frac{|\{c \in \mathbb{F}_p : c \text{ achievable}\}|}{p},$$

so that δ is exactly the probability that a uniformly distributed target admits a solution. If the target is not achievable, then no adversary can win. Clearly, $\delta \leq \Delta/p$, and from now on, we aim to bound Δ .

Two constraints do that. The first is a bound on the size of an achievable rational target, the second a set of congruences. For the first, note that $\sum_{i \in \text{CS}} 1/i$ is largest at $\text{CS} = \{1, \dots, t\}$ and smallest at its complement, so every achievable rational c lies in $[H_n - H_t, H_t]$, an interval of length $2H_t - H_n$. For the second, clear denominators: if $c \in \mathbb{Q}$ is achievable then $L_n c$ is an integer, and its residue modulo any prime power is constrained.

MODULAR CONSTRAINTS. Let $q_1, \dots, q_s$ be the primes at most n . For each q_j , denote by $v_j := v_{q_j}(L_n)$ the largest integer such that $q_j^{v_j} \mid L_n$, i.e., the q_j -adic valuation of L_n , and define

$$A_j := \left\{ \sum_{i \in \text{CS}} L_n/i \bmod q_j^{v_j} : \text{CS} \subseteq \{1, \dots, n\}, |\text{CS}| = t \right\} \subseteq \mathbb{Z}/q_j^{v_j}\mathbb{Z},$$

the set of residues the scaled target $L_n c$ can have. In Lemma 5.1 we claim that the size of those sets can be efficiently computed. Then, we run the computation for $n = 256$ and $t = 128$ and report the results in Table 1. The cardinality of these sets will be used at the end of the section to upper bound the success probability.

primes q_j	Count	$ A_j $	$\log_2 \prod_j A_j $
$131 \leq q_j \leq 251$	23	2	23.000
$89 \leq q_j \leq 127$	8	4	16.000
$67 \leq q_j \leq 83$	5	8	15.000
$53 \leq q_j \leq 61$	3	16	12.000
13, 11, 5	3	26, 44, 100	16.804
3	1	162	7.340
2, 7, 17, 19, 23, 29, 31	7	$q_j^{v_j}$	36.286
37, 41, 43, 47	4	31, 39, 29, 25	19.741
$\log_2 \prod_j A_j $			146.171

Table 1. The residue sets A_j of Lemma 5.1 at $n = 256$, grouped by the value of $|A_j|$, with $\log_2 \prod_j |A_j|$ taken over each group.

Lemma 5.1. *Let $q_j \leq n$ be prime and let $M_j := \{i \in [n] : q_j \mid i\}$, of size $\lfloor n/q_j \rfloor$. Then*

$$A_j = \left\{ \sum_{i \in T} L_n/i \bmod q_j^{v_j} : T \subseteq M_j \right\},$$

and this set is obtained in $O(\lfloor n/q_j \rfloor q_j^{v_j})$ steps by starting from $\{0\}$ and processing each weight $L_n/i \bmod q_j^{v_j}$, $i \in M_j$, once, adding its sum with every previously reachable residue while retaining the existing residues.

Proof. Since $v_{q_j}(L_n/i) = v_j - v_{q_j}(i)$, we have $q_j^{v_j} \mid L_n/i$ exactly when $q_j \nmid i$, so only the terms with $i \in M_j$ survive modulo $q_j^{v_j}$ and the residue depends on CS only through $T := \text{CS} \cap M_j$. Conversely every $T \subseteq M_j$ occurs: $|T| \leq \lfloor n/q_j \rfloor \leq n/2 = t$, and there are $n - |M_j| \geq t$ indices outside M_j . Thus T can be completed to a t -subset by adding $t - |T|$ indices from $[n] \setminus M_j$. The two sets therefore coincide. The computation processes each weight once and maintains a reachable set in $\mathbb{Z}/q_j^{v_j}\mathbb{Z}$, of size at most $q_j^{v_j}$, giving the claimed running time. $\square$

Both the modulus $q_j^{v_j} \leq n$ and the number of weights $\lfloor n/q_j \rfloor \leq t$ are small, so Table 1 can be computed efficiently. Reading it, the saving comes from the prime powers that are large relative to n . Only indices divisible by $q_j^{v_j}$ contribute a non-zero term modulo q_j . If $q_j^{v_j} > n/2$ then $\{1, \dots, n\}$ contains a single such index, the least significant base- q_j digit of the sum takes only two values, and $|A_j| \leq 2q_j^{v_j-1}$. More generally, $|A_j| \leq 2^m q_j^{v_j-1}$ with $m := \lfloor n/q_j^{v_j} \rfloor$, since at most 2^m residues modulo q_j can occur. At $n = 256$, the computed sets attain this bound for every $q_j^{v_j} \geq 53$ — that is, for 44 of the 54 primes. Moreover, A_j is all of $\mathbb{Z}/q_j^{v_j}\mathbb{Z}$ for $q_j \in \{2, 7, 17, 19, 23, 29, 31\}$, while for 37, 41, 43, 47 it is a proper subset, of size 31, 39, 29, 25 respectively.

THE DENSITY. For the possible residues modulo $q_j^{v_j}$ discussed above, we extend the argument to a density argument over an interval of $\mathbb{Z}$ defined by the maximum and minimum value the scaled target can possibly attain.

Theorem 5.2. *With the notation above,*

$$\Delta \leq (\lfloor 2H_t - H_n \rfloor + 1) \prod_j |A_j|$$

and, consequently,

$$\delta \leq \frac{\Delta}{p} \leq \frac{\lfloor 2H_t - H_n \rfloor + 1}{p} \prod_j |A_j|.$$

Proof. Let $c \in \mathbb{Q}$ be achievable and put $y := L_n c$. Then y is an integer, it lies in an interval of length $(2H_t - H_n)L_n$, and $y \bmod q_j^{v_j} \in A_j$ for every j ; distinct achievable c give distinct y . Since $L_n = \prod_j q_j^{v_j}$, the Chinese remainder theorem says that the residues modulo L_n meeting all of those conditions number exactly $\prod_j |A_j|$, and an interval of length $(2H_t - H_n)L_n$ contains at most $\lfloor 2H_t - H_n \rfloor + 1$ integers of any one residue class modulo L_n . Hence $\Delta \leq (\lfloor 2H_t - H_n \rfloor + 1) \prod_j |A_j|$. The bound on δ is this together with $\delta \leq \Delta/p$, noted above. $\square$

For $n = 256$ and $t_c = t = 128$ we have $2H_t - H_n \approx 4.742$ and $\log_2 \prod_j |A_j| \approx 146.171$, so for any 256-bit prime p , using $p \geq 2^{255}$,

$$\Delta \leq 2^{148.5}, \quad \delta \leq 2^{-106.5}.$$

6 LDVR Attacks over General Evaluation Points

The attack from Section 4 leverages the canonical evaluation points $x_i = i$. We have also shown in Section 5 that when $\binom{n}{t}$ approaches p , that particular set of evaluation points is potentially no longer amenable to efficient attacks following the subset-sum approach. Here, we briefly observe that it can be substantially easier to find attacks if one uses different choices of evaluation points. It is important to appreciate that the original formulation from [CKK⁺25] does use the canonical choice, and that the FROST RFC [CKGW24] (and consequently its implementations) similarly adopts it. However, the full treatment of FROST (this includes its provable security treatment [BCK⁺22, CKK⁺25]) extends to any choice of evaluation points, and there are several implementation choices that may lead one to want to do so. Below, we also discuss how certain choices (in particular, random evaluation points) seem to mitigate the effects of LDVR attacks (with respect to our current cryptanalytic understanding).

6.1 Bad Evaluation Points

The subset-sum viewpoint introduced by Lemma 3.3 immediately surfaces very poor choices of evaluation points that make attacks more straightforward. In particular, if $\mathcal{X}$ outputs $\mathbf{x}$ such that $\mathbf{x}^{-1}$ allows for an easy representation of each target, an efficient algorithm is immediate.

A SIMPLE ATTACK. A simple attack emerges already from Section 4.1. The reason the attack idea as illustrated was inefficient in terms of necessary parties when using $x_i = i$ is that we need to be able to accommodate the constructed CS as a subset of $[n]$, and this requires a large n . However, we only need very few of these evaluation points. If we choose $x_1, \dots, x_n$ such that the first $2m \leq n$ evaluation points are $x_{2j+1} = 2^{m-1-j} \pmod{p}$ and $x_{2j+2} = 3 \cdot 2^{m-1-j} \pmod{p}$ for $j = 0, 1, \dots, m-1$, where $m = \lceil \log p \rceil$, we immediately get to select a

CS of size exactly m whose reciprocals can represent every target in $\mathbb{F}_p$. Here, only $n = 2m = O(\log p)$ and $t = t_c = m = O(\log p)$ suffice (in concrete terms, we would get $n = 512$ and $t = 256$ as a feasible regime). One small caveat here is that the points may not be distinct, depending on the choice of p , and thus this condition needs to be checked.

When we cast this strategy as an LDVR attack, we note that the resulting attack is a single-query attack, and the computational costs are minimal.

GENERALIZATION. The above is a special case of a larger class of attacks. We let $t_c = t < n < p$. We let $m := \lceil \log p \rceil$, and assume $m \leq t_c \leq n - m$. Then, we assume that the first $2m$ evaluation points are so that there exists $\gamma \in \mathbb{F}_p^*$ such that for all $j \in \{0, 1, \dots, m-1\}$, we have

$$\frac{1}{x_{2j+2}} - \frac{1}{x_{2j+1}} = \gamma 2^j.$$

The remaining $n - 2m$ evaluation points are arbitrary. Then, given a target $c \in \mathbb{F}_p$, we proceed as follows. We pick an arbitrary subset $T \subseteq \{2m+1, \dots, n\}$ with $|T| = t_c - m$, and let

$$K = \sum_{i \in T} \frac{1}{x_i} + \sum_{j=0}^{m-1} \frac{1}{x_{2j+1}}.$$

Then, compute $b_0, \dots, b_{m-1} \in \{0, 1\}$ such that $z = \gamma^{-1}(c - K) = \sum_{i=0}^{m-1} b_i 2^i$ and now we are going to pick the set

$$\text{CS} = T \cup \{2j+1 + b_j \mid j \in \{0, 1, \dots, m-1\}\},$$

and note that $|\text{CS}| = t_c$. Now observe that

$$\begin{aligned} \sum_{i \in \text{CS}} \frac{1}{x_i} &= \sum_{i \in T} \frac{1}{x_i} + \sum_{j=0}^{m-1} \left(\frac{1}{x_{2j+1+b_j}} - \frac{1}{x_{2j+1}} \right) + \sum_{j=0}^{m-1} \frac{1}{x_{2j+1}} \\ &= K + \gamma \sum_{j=0}^{m-1} b_j 2^j = K + \gamma \cdot \gamma^{-1}(c - K) = c. \end{aligned}$$

SMALLER CORRUPTION SETS. The above attack template requires $t_c \geq m$ and $n \geq 2m$, where $m = \lceil \log p \rceil$. What happens if we have fewer parties? To this end, assume instead that $m' \leq t_c = t \leq n - m'$, where $m' < m$. Then, we can carry out exactly the same attack as above, replacing m with m' . That however means that if $z = \gamma^{-1}(c - K) \geq 2^{m'}$, we need to abort. Since z is uniform, our attack succeeds with probability $2^{m'}/p$. It can be repeated, leveraging multiple queries from the LDVR game, to win with an expected number of $q = p/2^{m'}$ queries. For example, for $n = 400$, $m' = 200$, and $p \approx 2^{256}$, we only need roughly 2^{56} queries.

ON PARAMETERS. Theorem 2.1 implies that $q_{\mathcal{O}} \binom{n}{t_c} \geq \varepsilon p$ for us to be able to carry out an attack with advantage ε . The above attack is not yet optimal in this respect, although it is much closer than the attack against canonical evaluation point choices. An open question is to close this gap.

6.2 Backdoors for Evaluation Points

Section 6.1 presented structured choices of evaluation points that lead to efficient LDVR attacks. Although the structure is apparent in those examples, an adversarial generator need not expose it in its original form. In particular, the transformations underlying Lemma 3.4 allow the generator to change the evaluation points while retaining the ability to transfer an attack, using the transformation parameters as auxiliary information. In particular, for every $\mathcal{X}$ admitting an efficient LDVR attack, an efficient attack against $\mathcal{X}_y$ follows when the sampled transformation parameters (f, d) are known. Keeping these parameters secret may make the original structure less immediately apparent to someone inspecting the resulting points.

This suggests a potential backdoor risk when evaluation points are chosen adversarially. The transformation alone does not guarantee that the structure is difficult to recognize: for example, the paired relations from Section 6.1 survive it up to a common scaling factor. Nevertheless, other structured families may be less readily recognizable after transformation. We view this as a heuristic concern, rather than a claim of computationally hidden structure, and as further motivation for a publicly specified procedure for generating evaluation points.

6.3 Random Evaluation Points

If the x_i 's are uniformly random and independent over $\mathbb{F}_p^*$, Lemma 3.3 gives a random, fixed-weight subset-sum instance, and we can already resort to the large body of works in this space. We show scenarios in which attacks outperforming the conjectured $O(\sqrt{p})$ hardness are possible.

BACKGROUND. First off, Wagner's generalized birthday algorithm [Wag02] solves the k -list sum problem over $\mathbb{F}_p$ in time and space roughly $\tilde{O}(p^{1/(1+\log_2 k)})$; in particular, allowing the number of lists to grow yields subexponential complexity in $\lambda = \log_2 p$. The application of this technique to random⁵ dense subset-sum instances was developed by Lyubashevsky [Lyu05] and subsequently analyzed rigorously by Shallue [Sha08]. Importantly for our setting, Shallue [Sha12] extended these techniques to solutions of prescribed weight. This relies on k -set splitting systems: the coordinates are partitioned into k blocks so that a solution of weight t contributes exactly t/k elements from each block, after which Wagner's algorithm is applied to the corresponding lists of subset sums.

CONCRETE CALCULATION. Let $k \geq 2$ be a power of two dividing both n and t . For a direct application of Shallue's analysis, one requires

$$\binom{n/k}{t/k} > p .$$

For fixed k , the resulting attack has expected time and space

$$\tilde{O}(t^{(k-1)/2} p^{1/(1+\log_2 k)}) \quad \text{and} \quad \tilde{O}(p^{1/(1+\log_2 k)}) ,$$

⁵ We use "random" loosely, in that the definition of the problem changes slightly depending on whether the instance requires individual random summands to be distinct or not, or non-zero. This is because we are in a large p regime where different constraints only negligibly change the distribution of the instance, and thus do not affect algorithmic performance.

respectively, where the implicit constants may depend on k . We ignore hidden constants, as determining exact complexities is fairly involved—we merely want to convey a somewhat high-level message here. When choosing $p \approx 2^{256}$, the displayed time and space factors have binary logarithms

$$B_T = \frac{256}{1 + \log_2 k} + \frac{k-1}{2} \log_2 t, \quad B_S = \frac{256}{1 + \log_2 k}.$$

Overall this paints the following picture: Non-trivial attacks (i.e., beating the $\sqrt{p}$ -barrier corresponding to the hardness of breaking the discrete logarithm problem) are possible, but are significantly costlier than those considered for canonical and ad-hoc evaluation points given above.

- For a 256-bit prime, one concrete choice is $t_c = t = 524$ and $n = 1048$, where we can take $k = 4$. Indeed, $\log_2 \binom{262}{131} \approx 257.66 > 256$. The displayed time factors then contribute approximately $B_T = 256/3 + (3/2) \log_2(524) \approx 98.9$ bits of work, compared with 128 bits for generic discrete logarithm.
- If we were to take $n = 512$ and $t = 256$, for which we have given a simple attack above leveraging ad-hoc evaluation points, then Shallue’s analysis does not suggest speed-up even for $k = 2$, since $\log_2 \binom{256}{128} \approx 251.67 < 256$ does not satisfy the necessary condition for the analysis to apply.
- If we consider the setting $n = 5020$ and $t = 204$, for which we gave an efficient attack against the canonical evaluation points in Section 4.3, we can pick $k = 4$. Then, $\log \binom{1255}{51} \approx 303.6 > 256$; here $B_T \approx 96.8$. One can also verify that no larger k satisfies the condition.

We also note that asymptotically, one can also find attacks running in a concrete subexponential regime, but this requires a more careful analysis of the hidden terms dependent on k affecting the complexity of the subset-sum algorithms.

6.4 Countermeasures

With respect to the above discussion, we have seen that two concerns are: (1) random evaluation points appear to offer the strongest resilience to cryptanalytic attacks, at least based on our current understanding of LDVR cryptanalysis, and (2) there are several ways in which adversarially chosen evaluation points can lead to fast attacks, and they can be obfuscated as discussed in Section 6.2. A simple countermeasure here is to use $x_i = H(i)$, for a hash function H producing outputs in $\mathbb{F}_p$, of course rejecting in the (unlikely) case of a collision and/or a point being zero. For instance, assuming H behaves as a random oracle will ensure nothing-up-my-sleeve generation of the x_i ’s with distribution consistent with random subset-sum instances for attacks leveraging Lemma 3.3.

Acknowledgments

We thank Chelsea Komlo for discussions about the impact of these attacks on FROST deployments. Tessaro was partially supported by NSF grants CNS-2154174, CNS-2426905

and a Stellar Development Foundation Academic Research Award. Tessaro’s work was carried out while visiting ETH Zurich.

References

- AFOS26. Ojaswi Acharya, Georg Fuchsbauer, Adam O’Neill, and Marek Sefranek. Revisiting the security of sparkle. Cryptology ePrint Archive, Paper 2026/431, 2026.
- Apo76. Tom M. Apostol. *Introduction to analytic number theory*. Undergraduate Texts in Mathematics. Springer-Verlag, New York, 1976.
- BCK⁺22. Mihir Bellare, Elizabeth C. Crites, Chelsea Komlo, Mary Maller, Stefano Tessaro, and Chenzhi Zhu. Better than advertised security for non-interactive threshold signatures. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part IV*, volume 13510 of *LNCS*, pages 517–550. Springer, Cham, August 2022.
- BCL⁺26. Renas Bacho, Yanbo Chen, Julian Loss, Stefano Tessaro, and Chenzhi Zhu. Adaptively secure partially non-interactive threshold schnorr signatures in the AGM. In Joan Daemen and Emmanuel Thomé, editors, *EUROCRYPT 2026, Part I*, volume 16541 of *LNCS*, pages 33–62. Springer, Cham, May 2026.
- BD22. Luís Brandão and Michael Davidson. Notes on threshold eddsa/schnorr signatures. Technical report, 2022.
- BDLR25a. Renas Bacho, Sourav Das, Julian Loss, and Ling Ren. Adaptively secure three-round threshold Schnorr signatures from DDH. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part VI*, volume 16005 of *LNCS*, pages 390–422. Springer, Cham, August 2025.
- BDLR25b. Renas Bacho, Sourav Das, Julian Loss, and Ling Ren. Glacius: Threshold schnorr signatures from DDH with full adaptive security. In Serge Fehr and Pierre-Alain Fouque, editors, *EUROCRYPT 2025, Part II*, volume 15602 of *LNCS*, pages 304–334. Springer, Cham, May 2025.
- BGC⁺25. Ruben Baecker, Paul Gerhart, Davide Li Calsi, Luigi Russo, Dominique Schröder, and Arkady Yerukhimovich. Fully adaptive FROST in the algebraic group model from falsifiable assumptions. Cryptology ePrint Archive, Report 2025/1950, 2025.
- BHK⁺24. Fabrice Benhamouda, Shai Halevi, Hugo Krawczyk, Yiping Ma, and Tal Rabin. SPRINT: High-throughput robust distributed Schnorr signatures. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part V*, volume 14655 of *LNCS*, pages 62–91. Springer, Cham, May 2024.
- BLL⁺21. Fabrice Benhamouda, Tancrede Lepoint, Julian Loss, Michele Orrù, and Mariana Raykova. On the (in)security of ROS. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 33–53. Springer, Cham, October 2021.
- BLSW24. Renas Bacho, Julian Loss, Gilad Stern, and Benedikt Wagner. HARTS: High-threshold, adaptively secure, and robust threshold Schnorr signatures. In Kai-Min Chung and Yu Sasaki, editors, *ASIACRYPT 2024, Part III*, volume 15486 of *LNCS*, pages 104–140. Springer, Singapore, December 2024.
- BLT⁺24. Renas Bacho, Julian Loss, Stefano Tessaro, Benedikt Wagner, and Chenzhi Zhu. Twinkle: Threshold signatures from DDH with full adaptive security. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part I*, volume 14651 of *LNCS*, pages 429–459. Springer, Cham, May 2024.
- BP25. Luís Brandão and Rene Peralta. Nist first call for multi-party threshold schemes. Technical report, 2025.
- BR93. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby, editors, *ACM CCS 93*, pages 62–73. ACM Press, November 1993.
- CDD⁺04. Ran Canetti, Ivan Damgård, Stefan Dziembowski, Yuval Ishai, and Tal Malkin. Adaptive versus non-adaptive security of multi-party protocols. *Journal of Cryptology*, 17(3):153–207, June 2004.
- CFGN96. Ran Canetti, Uriel Feige, Oded Goldreich, and Moni Naor. Adaptively secure multi-party computation. In *28th ACM STOC*, pages 639–648. ACM Press, May 1996.
- CKGW24. D. Connolly, C. Komlo, I. Goldberg, and C. A. Wood. The Flexible Round-Optimized Schnorr Threshold (FROST) Protocol for Two-Round Schnorr Signatures. RFC 9591, June 2024.
- CKK⁺25. Elizabeth C. Crites, Jonathan Katz, Chelsea Komlo, Stefano Tessaro, and Chenzhi Zhu. On the adaptive security of FROST. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part VI*, volume 16005 of *LNCS*, pages 480–511. Springer, Cham, August 2025.
- CKM23. Elizabeth C. Crites, Chelsea Komlo, and Mary Maller. Fully adaptive Schnorr threshold signatures. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part I*, volume 14081 of *LNCS*, pages 678–709. Springer, Cham, August 2023.

- CS25. Elizabeth C. Crites and Alistair Stewart. A plausible attack on the adaptive security of threshold Schnorr signatures. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part VI*, volume 16005 of *LNCS*, pages 457–479. Springer, Cham, August 2025.
- Des93. Yvo Desmedt. Threshold cryptosystems (invited talk). In Jennifer Seberry and Yuliang Zheng, editors, *AUSCRYPT'92*, volume 718 of *LNCS*, pages 3–14. Springer, Berlin, Heidelberg, December 1993.
- FKL18. Georg Fuchsbauer, Eike Kiltz, and Julian Loss. The algebraic group model and its applications. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 33–62. Springer, Cham, August 2018.
- GJKR99. Rosario Gennaro, Stanislaw Jarecki, Hugo Krawczyk, and Tal Rabin. Secure distributed key generation for discrete-log based cryptosystems. In Jacques Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 295–310. Springer, Berlin, Heidelberg, May 1999.
- GJKR07. Rosario Gennaro, Stanislaw Jarecki, Hugo Krawczyk, and Tal Rabin. Secure distributed key generation for discrete-log based cryptosystems. *Journal of Cryptology*, 20(1):51–83, January 2007.
- GKMN21. François Garillot, Yashvanth Kondi, Payman Mohassel, and Valeria Nikolaenko. Threshold Schnorr with stateless deterministic signing from standard assumptions. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 127–156, Virtual Event, August 2021. Springer, Cham.
- JLS25. Antoine Joux, Julian Loss, and Giacomo Santato. Dimensional eROSion: Improving the ROS attack with decomposition in higher bases. In Benny Applebaum and Huijia (Rachel) Lin, editors, *TCC 2025, Part III*, volume 16270 of *LNCS*, pages 323–352. Springer, Cham, December 2025.
- KG20. Chelsea Komlo and Ian Goldberg. FROST: Flexible round-optimized Schnorr threshold signatures. In Orr Dunkelman, Michael J. Jacobson, Jr., and Colin O’Flynn, editors, *SAC 2020*, volume 12804 of *LNCS*, pages 34–65. Springer, Cham, October 2020.
- KG25. Chelsea Komlo and Ian Goldberg. Arctic: Lightweight and stateless threshold Schnorr signatures. In Tibor Jager and Jiaxin Pan, editors, *PKC 2025, Part V*, volume 15678 of *LNCS*, pages 234–267. Springer, Cham, May 2025.
- KRT24. Shuichi Katsumata, Michael Reichle, and Kaoru Takemure. Adaptively secure 5 round threshold signatures from MLWE/MSIS and DL with rewinding. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part VII*, volume 14926 of *LNCS*, pages 459–491. Springer, Cham, August 2024.
- Lin24. Yehuda Lindell. Simple three-round multiparty Schnorr signing with full simulatability. *CiC*, 1(1):25, 2024.
- Lyu05. Vadim Lyubashevsky. The parity problem in the presence of noise, decoding random linear codes, and the subset sum problem. In Chandra Chekuri, Klaus Jansen, José D. P. Rolim, and Luca Trevisan, editors, *Approximation, Randomization and Combinatorial Optimization, Algorithms and Techniques, 8th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2005 and 9th International Workshop on Randomization and Computation, RANDOM 2005, Berkeley, CA, USA, August 22–24, 2005, Proceedings*, volume 3624 of *Lecture Notes in Computer Science*, pages 378–389. Springer, 2005.
- Mak22. Nikolaos Makriyannis. On the classic protocol for MPC Schnorr signatures. Cryptology ePrint Archive, Report 2022/1332, 2022.
- NRT26. Guilhem Niot, Michael Reichle, and Kaoru Takemure. Adaptively-secure three-round threshold schnorr from DL. In Joan Daemen and Emmanuel Thomé, editors, *EUROCRYPT 2026, Part I*, volume 16541 of *LNCS*, pages 3–32. Springer, Cham, May 2026.
- PKN⁺25. Rafaël Del Pino, Shuichi Katsumata, Guilhem Niot, Michael Reichle, and Kaoru Takemure. Unmasking TRaccoon: A lattice-based threshold signature with an efficient identifiable abort protocol. In Yael Tauman Kalai and Seny F. Kamara, editors, *CRYPTO 2025, Part VI*, volume 16005 of *LNCS*, pages 423–456. Springer, Cham, August 2025.
- Sch91. Claus-Peter Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, January 1991.
- Sch01. Claus-Peter Schnorr. Security of blind discrete log signatures against interactive attacks. In Sihan Qing, Tatsuaki Okamoto, and Jianying Zhou, editors, *ICICS 01*, volume 2229 of *LNCS*, pages 1–12. Springer, Berlin, Heidelberg, November 2001.
- Sha08. Andrew Shallue. An improved multi-set algorithm for the dense subset sum problem. In Alfred J. van der Poorten and Andreas Stein, editors, *Algorithmic Number Theory, 8th International Symposium, ANTS-VIII, Banff, Canada, May 17–22, 2008, Proceedings*, volume 5011 of *Lecture Notes in Computer Science*, pages 416–429. Springer, 2008.

- Sha12. Andrew Shallue. Division algorithms for the fixed weight subset sum problem. *CoRR*, abs/1201.2739, 2012.
- SS01. Douglas R. Stinson and Reto Stroh. Provably secure distributed Schnorr signatures and a (t, n) threshold scheme for implicit certificates. In Vijay Varadharajan and Yi Mu, editors, *ACISP 01*, volume 2119 of *LNCS*, pages 417–434. Springer, Berlin, Heidelberg, July 2001.
- TZ23. Stefano Tessaro and Chenzhi Zhu. Threshold and multi-signature schemes from linear hash functions. In Carmi Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 628–658. Springer, Cham, April 2023.
- Wag02. David Wagner. A generalized birthday problem. In Moti Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 288–303. Springer, Berlin, Heidelberg, August 2002.

A Statement on AI Use

The results of this paper, and in particular the framework and reductions of Section 3, were developed entirely by the authors, with the exception of the attack in Section 4 and parts of the argument in Section 5, as we explain below. The paper was also written by the authors, with minor writing assistance from LLMs.

The central step in the development of the paper was the authors’ reduction of LDVR to a structured subset-sum problem presented in Section 3. This reduction is the key technical enabler for all attacks studied subsequently: it exposes the arithmetic structure of the evaluation points in a form that makes it possible to reason about when efficient subset-sum strategies can succeed. Starting from this framework, the authors also observed that poorly chosen evaluation points can lead to efficient attacks and that random evaluation points admit attacks improving upon the conjectured hardness from prior work.

The core technical idea leading to the polynomial-time LDVR attack for $x_i = i$ was first identified by OpenAI’s GPT-5.6 Sol during an extended interaction with the authors. The model initially returned repeatedly to variants of the reductions and attacks underlying Section 3, as well as to heuristic subexponential approaches close to directions already investigated by the authors. Only later in this interaction did it propose the key idea leading to the polynomial-time attack, initially in a substantially more complicated form than the attack eventually presented in Section 4. The authors subsequently verified the argument, isolated its essential ingredients, simplified and formalized the attack, and developed the proof and presentation appearing in the paper.

There was substantial overlap between this AI-assisted discovery and attacks that the authors had already previously developed independently. In particular, using the same subset-sum viewpoint, the authors had already obtained closely related polynomial-time attacks in parameter regimes in which LDVR instances are unlikely to admit solutions; the analysis of these attacks led to the results of Section 5. The new contribution arising from the interaction with GPT-5.6 Sol was the identification of choices of n and t_c for which the subset-sum strategy succeeds with high probability on the relevant LDVR instances, thereby turning this line of attack into the polynomial-time attack of Section 4.

The subsequent optimization of the attack for the concrete prime moduli considered in the paper was again carried out predominantly by the authors. This included adapting the asymptotic strategy to the specific arithmetic constraints of the concrete instances and deriving the optimized attacks reported in the paper. Additional numerical refinements arose during interaction with GPT-5.6 Sol, and the authors verified these.

Claude Opus 5 was used separately in connection with an earlier, heuristic version of Section 5. The authors initially intended to use the model to generate the code underlying Table 1 and to explore an attack on smaller instances. After producing the requested data, the model independently suggested that, in the relevant regime, such an attack should in fact fail with non-negligible probability and proposed a proof of this statement without having been explicitly asked to do so. The authors subsequently verified, simplified, and formalized this argument.

The authors take full responsibility for every statement, proof, experiment, and conclusion in this paper.