The Hardness of LPN over Any Integer Ring and Field for PCG Applications

Hanlin Liu[1,4], Xiao Wang[2(✉)], Kang Yang[3(✉)], and Yu Yu[1,4(✉)]

[1] Shanghai Jiao Tong University, Shanghai, China
{hans1024,yyuu}@sjtu.edu.cn
[2] Northwestern University, Evanston, USA
wangxiao@northwestern.edu
[3] State Key Laboratory of Cryptology, Beijing, China
yangk@sklc.org
[4] Shanghai Qi Zhi Institute, Shanghai, China

Abstract. Learning parity with noise (LPN) has been widely studied and used in cryptography. It was recently brought to new prosperity since Boyle et al. (CCS'18), putting LPN to a central role in designing secure multi-party computation, zero-knowledge proofs, private set intersection, and many other protocols. In this paper, we thoroughly studied the security of LPN problems in this particular context. We found that some important aspects have long been ignored and many conclusions from classical LPN cryptanalysis do not apply to this new setting, due to the low noise rates, extremely high dimensions, various types (in addition to $\mathbb{F}_2$) and noise distributions.

- For LPN over a field, we give a parameterized reduction from exact-noise LPN to regular-noise LPN. Compared to the recent result by Feneuil, Joux and Rivain (Crypto'22), we significantly reduce the security loss by paying only a small additive price in dimension and number of samples.
- We analyze the security of LPN over a ring $\mathbb{Z}_{2^\lambda}$. Existing protocols based on LPN over integer rings use parameters as if they are over fields, but we found an attack that effectively reduces the weight of a noise by half compared to LPN over fields. Consequently, prior works that use LPN over $\mathbb{Z}_{2^\lambda}$ overestimate up to 40 bits of security.
- We provide a complete picture of the hardness of LPN over integer rings by showing: 1) the equivalence between its search and decisional versions; 2) an efficient reduction from LPN over $\mathbb{F}_2$ to LPN over $\mathbb{Z}_{2^\lambda}$; and 3) generalization of our results to any integer ring.

Finally, we provide an all-in-one estimator tool for the bit security of LPN parameters in the context of PCG, incorporating the recent advanced attacks.

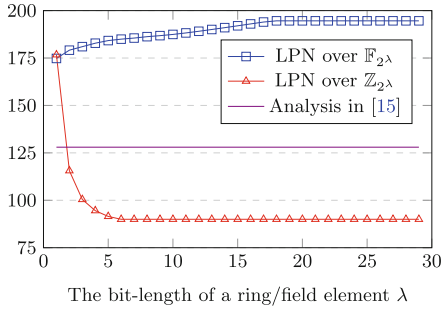
1 Introduction

The learning parity with noise (LPN) assumption states that it is hard to distinguish LPN samples $(\mathbf{A}, \mathbf{A} \cdot \boldsymbol{s} + \boldsymbol{e})$ from random samples, where $\mathbf{A}$ is a public

M. Joye and G. Leander (Eds.): EUROCRYPT 2024, LNCS 14656, pp. 149–179, 2024.
https://doi.org/10.1007/978-3-031-58751-1_6

	Protocol	LPN type
[17, 82]	(C)OT	$\mathbb{F}_2$
[69]	VOLE	$\mathbb{F}_{2^{61}-1}$ and $\mathbb{Z}_{2^{64}}$
[76]	ZK	$\mathbb{F}_2$ and $\mathbb{F}_{2^{61}-1}$
[45]	ZK	$\mathbb{F}_{2^{128}}$
[10]	ZK	$\mathbb{F}_{2^{40}}$ and $\mathbb{F}_{2^{61}-1}$
[8]	ZK	$\mathbb{Z}_{2^{72}}$
[9]	ZK	$\mathbb{Z}_{2^{104}}$
[34, 29]	MPC	$\mathbb{F}_2$, $\mathbb{F}_{2^{40}}$ and $\mathbb{F}_{2^{128}}$
[68, 23, 66]	PSI	$\mathbb{F}_{2^{128}}$

(a) Prior works in the PCG framework and their required LPN variants over different fields and rings.



(b) **The bit-security from our analysis for LPN over $\mathbb{F}_{2^\lambda}$ and $\mathbb{Z}_{2^\lambda}$.** Parameters $N = 2^{10}, k = 652, t = 106$ are used.

Fig. 1. LPN assumptions in prior works, and our analysis on one set of parameters. For a set of parameters (N, k, t), N is the number of samples, k is the dimension and t is the Hamming weight of a noise vector.

matrix, $\boldsymbol{s}$ is a random secret and $\boldsymbol{e}$ is a noise vector sampled from a sparse distribution. The LPN assumption has been applied to build various primitives, e.g., symmetric encryption and authentication (e.g., [49] and follow-up works), public key encryption [4], commitment scheme [53], garbled circuits [5], oblivious transfer [32] and collision-resistant hash functions [21,84]. All these primitives adopt LPN over binary field $\mathbb{F}_2$ with moderate dimensions.

The recent work by Boyle et al. [15] introduced the pseudorandom correlation generator (PCG) paradigm that can produce a large batch of correlated randomness, e.g., (correlated) oblivious transfer ((C)OT) and (vector) oblivious linear evaluation ((V)OLE), at a small communication. The core of the PCG idea is to build a pseudorandom generator (PRG) with a simple internal structure from LPN assumptions and then privately evaluate such a PRG using function secret sharing [20]. The sparsity of a noise $\boldsymbol{e}$ translates to communication efficiency, while the efficiency of LPN encoding translates to computational efficiency. Later, the PCG paradigm was used to build a series of concretely efficient protocols [1,14,16–19,27,67,69,76,82] with sublinear communication for generating random (C)OT or (V)OLE correlations. These PCG-like protocols have gained a lot of interests in designing various concretely efficient protocols, including secure multi-party computation (MPC) (e.g., [28–31,34,48,56,64,74,75,81]), zero-knowledge (ZK) proofs (e.g., [8–10,35,36,76,78,80]), privacy-preserving machine learning [51,69,77], private set intersection (PSI) [23,66,68], etc.

Although widely used in many constructions and some real-world applications, these protocols often use LPN variations that are not much studied in cryptanalysis, especially compared to the classical LPN assumption over $\mathbb{F}_2$ [4,43,46,73]. Furthermore, prior analyses on the classical LPN problems do not directly cover the LPN variants used in the PCG setting because of their unique features:

- **Value type.** Protocols often require an LPN assumption over a ring other than $\mathbb{F}_2$, including a finite field or even an integer ring[1] like $\mathbb{Z}_{2^\lambda}$.
- **Noise distribution.** Most existing analyses focus on a Bernoulli or exact noise distribution. However, most PCG-like protocols, for better performance, adopt a regular noise distribution, where the noise vector is divided into consecutive equal-sized sub-vectors, and each sub-vector has a single noisy coordinate in a random position.

  There are some recent exceptions. [42] showed a generalized reduction in LPN, which can imply a reduction from exact-noise LPN to regular-noise LPN but with a very large security loss; [24] showed an attack specific to regular noises but not for parameters usable in PCG applications; [22] also introduced an algebraic attack which, as we will show in this paper, can be cheaply mitigated without significantly increasing the communication.
- **Dimension and noise rate.** Most applications require an LPN assumption with very high dimension (e.g., millions) and low noise rate (e.g., $1/10^5$), which is out of the typically reported range of parameters considered for coding-theoretic primitives.

At this point, all implementations of PCG-like protocols use the LPN parameters from the original work by Boyle et al. [15], who analyzed the concrete security of LPN over $\mathbb{F}_{2^{128}}$. However, as we summarize in Table 1a, follow-up works used the same analysis to choose parameters for many different variants of LPN over $\mathbb{F}_2$, $\mathbb{F}_p$, and $\mathbb{Z}_{2^\lambda}$, many of which were not covered by the original analysis. It was not clear how large a gap in security when using LPN parameters over a field for LPN over another field or ring.

1.1 Our Contributions

In this paper, we put forth a set of LPN analyses specific to the setting of PCG applications. From the theoretical perspective, we show a tighter reduction from exact-noise LPN to regular-noise LPN and a complete categorization between LPN over integer rings and prime fields. From the concrete side, we summarize and incorporate all existing LPN attacks applicable to the PCG setting into one estimator tool that can be used for researchers to select LPN parameters. In particular, we find that existing PCG applications use parameters more expensive than necessary for fields and less security than needed for integer rings. Below we provide more details of our contributions.

The Hardness of LPN Under Regular Noise Distributions. Recently, Feneuil et al. [42] observed that, as a special case in their main theorem, an exact noise vector (of Hamming weight t) is also regular with some probability (estimated to e^{-t} in Sect. 3), and thus (T, ϵ)-hard[2] LPN under an exact noise

[1] By integer ring we refer to $\mathbb{Z}_N$ for any composite number N, which is used to distinguish from polynomial rings.

[2] We classify a problem as (T, ϵ)-hard when, for any probabilistic algorithm $\mathcal{B}$ with a running time of T, the algorithm's capacity to solve this problem is limited to a success probability of at most ϵ.

Table 1. Comparison between our analysis and [15] for the bit-security of an LPN problem with dimension k, number of samples N and Hamming weight of noises t over different rings. The bit-security considers an exact noise distribution; the values in brackets denote the decrease of bit-security due to the usage of a regular noise distribution. The sets of LPN parameters are adopted from [15].

LPN			This work					[15]
N	k	t	$\mathbb{F}_{2^{128}}$	$\mathbb{F}_{2^8}$	$\mathbb{Z}_{2^{128}}$	$\mathbb{Z}_4$	$\mathbb{F}_2$	Any field
2^{10}	652	57	111 (−0)	104 (−0)	54 (−2)	68 (−2)	94 (−4)	80
2^{12}	1589	98	100 (−0)	92 (−0)	53 (−0)	63 (−1)	83 (−3)	80
2^{14}	3482	198	101 (−0)	97 (−0)	58 (−1)	67 (−1)	86 (−3)	80
2^{16}	7391	389	103 (−0)	101 (−0)	63 (−1)	72 (−2)	91 (−4)	80
2^{18}	15336	760	105 (−0)	105 (−0)	68 (−1)	76 (−1)	95 (−3)	80
2^{20}	32771	1419	107 (−6)	107 (−6)	73 (−1)	81 (−1)	99 (−2)	80
2^{22}	67440	2735	108 (−4)	108 (−4)	75 (−1)	84 (−1)	104 (−5)	80

distribution implies $(T, e^t \cdot \epsilon)$-hard LPN under a regular noise distribution. However, the security loss is sometimes unaffordable as LPN may not have security beyond e^t in many practical settings. To reduce the security loss, we introduce a tunable parameter $\alpha \geq 2$ and divide a noise vector into αt blocks (each denoted by $\boldsymbol{e}_i$). Furthermore, instead of hoping that every $\boldsymbol{e}_i$ has the exact weight 1, we relax the condition to that the weight of $\boldsymbol{e}_i$ is *at most* 1. For each block, we add an extra sample with noise $\tilde{e}_i$ such that vector $(\boldsymbol{e}_i, \tilde{e}_i)$ has the exact weight 1, which allows us to obtain a regular noise vector. As a result, we prove that if the exact-noise LPN problem over an arbitrary field $\mathbb{F}$ with sample number N, dimension k and weight t is (T, ϵ)-hard, then the regular-noise LPN problem over $\mathbb{F}$ with sample number $(N + \alpha t)$, dimension $(k + \alpha t)$ and weight (αt) is $(T - \mathsf{poly}(k, N), 2^{\frac{t}{\alpha}} \cdot \epsilon)$-hard, where the security loss is reduced by at least 2^{α}, while the dimension and number of samples are increased by only αt.

We note that our reduction is not contradictory, but rather complementary, to a very recent work by Briaud and Øygarden [22]. In particular, they proposed a new algebraic attack that can take advantage of regular noise distributions, and demonstrated that the algebraic attack on regular-noise LPN is more efficient than other existing attacks, in the scenarios characterized by small code rates (particularly, some primal-LPN parameter sets). Whereas our reduction establishes an asymptotic connection, suggesting that LPN with regular noise could be as hard as that with exact noise, albeit with some security loss.

The Hardness of LPN over Integer Rings. Although having been used in protocol design [8,9,69], LPN problems over integer rings (e.g., $\mathbb{Z}_{2^\lambda}$) have received relatively limited attention in research. One notable exception is the work of Akavia [2], which explored a generalized LPN assumption over an integer ring within the context of the random samples access model. However, the work does not consider the hardness of LPN problems over integer rings in the PCG setting. As a result, all existing works for PCG-like protocols and applications select the parameters assuming that LPN over an integer ring is as secure as LPN over a finite field.

In this paper, we provide a complete relationship between LPN over fields and that over integer rings, with both asymptotic reduction and concrete analysis. From the theoretic side, we show the equivalence of related problems as shown in Fig. 2. On the concrete side, our analysis (in Fig. 1b and in Tables 1 and the full version of the paper [58, Table 2]) shows that LPN over an integer ring is significantly more vulnerable to attacks than LPN over a finite field of similar size. *What's more, we show that although LPN over a finite field becomes harder to attack as the field size increases, LPN over an integer ring becomes easier to attack as the ring size increases!*

1. Focusing on the most commonly used ring $\mathbb{Z}_{2^\lambda}$, we show a concrete attack that can solve a t-noise LPN over $\mathbb{Z}_{2^\lambda}$ by solving a $\left(\frac{2^{(\lambda-1)}}{2^\lambda-1}\cdot t\right)$-noise (which approximates to $t/2$) LPN over $\mathbb{F}_2$. This means that LPN over an integer ring is concretely weaker than LPN over a finite field and we need to double the weight of noise vectors to cover this attack. The impact to existing cryptographic protocols is significant. It will lead to roughly $2\times$ more communication and computation.
2. On the positive side, we provide an evidence that the LPN problem over an integer ring is generally hard. In particular, we show a reduction between t-noise LPN over $\mathbb{F}_2$ and $(\lambda \cdot t)$-noise LPN over a ring $\mathbb{Z}_{2^\lambda}$, which means that LPN over an integer ring is asymptotically as hard as classical LPN. This "efficient" reduction requires a different noise distribution: instead of sampling t locations and putting a uniform non-zero entry from $\mathbb{Z}_{2^\lambda}$ in each location, we need to independently sample λ weight-t noises $\boldsymbol{e}_0, \ldots, \boldsymbol{e}_{\lambda-1}$ over $\mathbb{F}_2$, and define the final noise vector as $\boldsymbol{e} = \sum_{i\in[\lambda]} 2^i \cdot \boldsymbol{e}_i$ with weight $\leq \lambda \cdot t$. This noise distribution may be interesting, as it can be used in the design of PCG-like protocols by adopting the upper bound $\lambda \cdot t$ to run these protocols. This change of distributions is crucial: without such change, the most favorable reduction we can identify shifts from t-noise LPN over $\mathbb{F}_2$ to $(2^\lambda \cdot t)$-noise LPN over $\mathbb{Z}_{2^\lambda}$, which is exponentially worse than the above. Another interesting fact is that the above reductions only require the code matrix $\mathbf{A}$ to be Boolean, which eliminates the need for integer multiplication during LPN encoding. Prior work [27] observed that using a Boolean code matrix is not vulnerable to existing linear-test attacks for LPN over finite fields; here we show that for LPN over integer rings, using a Boolean matrix is provably secure assuming that classical LPN over $\mathbb{F}_2$ is hard.
3. While the above reductions focus on the decisional version of LPN, we also give a reduction from computational LPN over $\mathbb{Z}_{2^\lambda}$ to that over $\mathbb{F}_2$. Thus, we show the equivalence between computational and decisional versions of LPN over $\mathbb{Z}_{2^\lambda}$ as shown in Fig. 2. We also generalize all the results to any integer ring. In particular, we show a concrete attack that can solve a t-noise LPN over a ring $\mathbb{Z}_{p^{\lambda_1}q^{\lambda_2}}$ by solving either a $\left(\frac{p-1}{p}\cdot t\right)$-noise LPN over $\mathbb{F}_p$ or a $\left(\frac{q-1}{q}\cdot t\right)$-noise LPN over $\mathbb{F}_q$, where p, q are two primes. This attack works for both computational and decisional versions of LPN. We also give a reduction from t-noise LPN over $\mathbb{F}_p$ and t-noise LPN over $\mathbb{F}_q$ to $\left((\lambda_1+\lambda_2)\cdot t\right)$-noise LPN

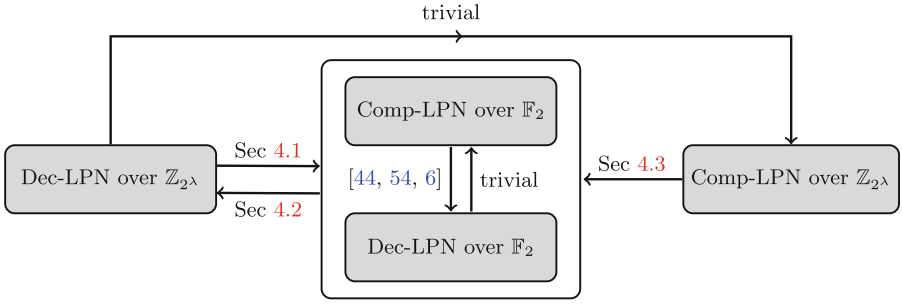


Fig. 2. The reduction relations between computational and decisional versions of LPN over $\mathbb{F}_2$ and $\mathbb{Z}_{2^\lambda}$ in the presence of Bernoulli and exact noise distributions.

over $\mathbb{Z}_{p^{\lambda_1}q^{\lambda_2}}$. Given these reductions over $\mathbb{Z}_{p^{\lambda_1}q^{\lambda_2}}$, one can easily generalize them to any integer ring.

Concrete Security of LPN for PCG. Finally, we maintain an easy-to-use tool to estimate the costs of the advanced attacks (Pooled Gauss, SD, ISD and algebraic attacks) on the concrete security of LPN problems related to the PCG setting, and will integrate new attacks found in the future into the estimator tool[3]. Prior to this work, most PCG-like protocols use the analysis from [15] for all LPN variants. We refined their analysis and incorporated attacks on integer rings and regular noises. See Table 1 and the full version of the paper [58, Table 2] for some representative parameters originally proposed in [15].

In the process of summarizing existing attacks, we also made an interesting observation in the context of PCG. Statistical decoding (SD) and information set decoding (ISD) are both important attack techniques for the exact-noise LPN problems. We observe that in the context of PCG, ISD attacks are almost always better than the SD attacks, including the recent work of SD 2.0 by Carrier et al. [25]. We formalize this observation by showing that both the optimal SD and SD 2.0 attacks (adapted to the low-noise setting) require more cost, compared to the Prange's original ISD algorithm [65] for a large set of commonly used parameters. Note that our findings do not diminish the relevance of SD 2.0; rather, they arise from differences in parameter settings between our work and [25]. This also shows the disparity of cryptanalysis between classical LPN problems with high noise rates and low-noise LPN problems used in PCG-like protocols.

Subsequent Works. The estimator tool has been used in subsequent works (e.g., [50]) to choose LPN parameters for PCG-like protocols. Our attack on integer rings has subsequently been noted by multiple works. Baum et al. [9] addressed this attack by a countermeasure: sampling the non-zero values in the noise vector only from invertible elements in $\mathbb{Z}_{2^\lambda}$ (i.e., odd values). This plausibly prevents the attack, and we did not find an efficient attack against LPN over

[3] Available at www.lpnestimator.com.

$\mathbb{Z}_{2^\lambda}$ with the countermeasure. Besides, the updated version by Boyle et al. [19] and the work by Lin et al. [57] adopted the same countermeasure to address our attack. It seems to be hard to prove that LPN over $\mathbb{F}_2$ implies LPN over $\mathbb{Z}_{2^\lambda}$ with random-odd noises, even if a significant security loss is allowed. This is because two noise vectors in two adjacent hybrids have the strong correlation, when a random odd value is sampled for each noisy coordinate. If one is desirable to obtain a tight reduction from LPN over $\mathbb{F}_2$ to that over $\mathbb{Z}_{2^\lambda}$, it may choose the noise distribution in the form of $\boldsymbol{e} = \sum_{i\in[\lambda]} 2^i \cdot \boldsymbol{e}_i$ with independent and random weight-t noises $\boldsymbol{e}_i$ for $i \in [\lambda]$.

2 Preliminary

2.1 Notation

We denote by log the logarithm in base 2. For $a, b \in \mathbb{N}$ with $a \leq b$, we write $[a, b] = \{a, \ldots, b\}$ and use $[n]$ to denote $[0, n-1]$ for simplicity. We use $x \leftarrow S$ to denote sampling x uniformly at random from a set S and $x \leftarrow \mathcal{D}$ to denote sampling x according to a distribution $\mathcal{D}$. For a ring $\mathcal{R}$, we denote by $|\mathcal{R}|$ the size of $\mathcal{R}$. We will use bold lower-case letters like $\boldsymbol{a}$ for column vectors, and bold upper-case letters like $\mathbf{A}$ for matrices. By slightly abusing the notation, for a vector $\boldsymbol{a}$, we use $|\boldsymbol{a}|$ to denote the Hamming weight of $\boldsymbol{a}$, and denote by $\boldsymbol{a}[i]$ the i-th component of $\boldsymbol{a}$. For two vectors $\boldsymbol{x}, \boldsymbol{y}$, we denote by $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ the inner product of $\boldsymbol{x}$ and $\boldsymbol{y}$. For a vector $\boldsymbol{a} \in (\mathbb{Z}_{2^\lambda})^k$, we use $\mathsf{BitDecomp}(\boldsymbol{a})$ to denote the bit-decomposition of $\boldsymbol{a}$, and its output is denoted by $(\boldsymbol{a}^0, \boldsymbol{a}^1, \cdots, \boldsymbol{a}^{\lambda-1})$ such that $\boldsymbol{a}^i \in \mathbb{F}_2^k$ for $i \in [\lambda]$ and $(\boldsymbol{a}^0[j], \boldsymbol{a}^1[j], \ldots, \boldsymbol{a}^{\lambda-1}[j])$ is the bit-decomposition of ring element $\boldsymbol{a}[j] \in \mathbb{Z}_{2^\lambda}$ for $j \in [k]$. Let $\mathsf{BitDecomp}^{-1}(\boldsymbol{a}^0, \boldsymbol{a}^1, \cdots, \boldsymbol{a}^{\lambda-1}) = \sum_{i=0}^{\lambda-1} 2^i \cdot \boldsymbol{a}^i \in (\mathbb{Z}_{2^\lambda})^k$ be the inverse of $\mathsf{BitDecomp}(\boldsymbol{a})$. We use $\mathsf{poly}(\cdot)$ to denote a polynomial function. For two distributions X and Y, we denote by $X \approx_c Y$ that X is computationally indistinguishable from Y. We will use the following lemma:

Lemma 1 (see, e.g., [83]). *For any $\mu \in (0, 1)$, if each coordinate of a vector $\boldsymbol{v} \in \mathbb{F}_2^t$ is independently set to 1 with probability μ, then the probability that $|\boldsymbol{v}| = \lceil \mu t \rceil$ is at least $\Omega(1/\sqrt{t})$.*

2.2 Learning Parity with Noise

Recently, variants of the learning parity with noise (LPN) assumption [13] are used to build PCG-like protocols with sublinear communication for generating (C)OT and (V)OLE correlations. The LPN variants are defined over a general finite ring $\mathcal{R}$. The known LPN-based PCG-like protocols mainly consider three cases for the choices of ring $\mathcal{R}$:

- Case 1 that $\mathcal{R} = \mathbb{F}_2$ is used to design the COT protocols [16–18,27,67,82], which is in turn able to be transformed into standard OT protocols.

- Case 2 that $\mathcal{R}$ is a finite field $\mathbb{F}$ with $|\mathbb{F}| > 2$ is used to construct the VOLE protocols [15–18,27,67,69,76] and the OLE protocols [1,14,18,19].
- Case 3 that $\mathcal{R} = \mathbb{Z}_{2^\lambda}$ (e.g., $\lambda \in \{32, 64, 128\}$) is used to obtain the VOLE protocols [8,9,57,69].

When considering more general rings such as $\mathcal{R} = \mathbb{Z}_{p^\lambda}$ for a prime $p > 2$ and $\mathcal{R} = \mathbb{Z}_{p^{\lambda_1}q^{\lambda_2}}$ for two primes p, q, the LPN problems over such rings may be interesting for future protocols. Following prior works (e.g., [17,18]), we define the (primal-)LPN and dual-LPN assumptions over a general ring $\mathcal{R}$ as follows:

Definition 1 (LPN). *Let $\mathcal{D}(\mathcal{R}) = \{\mathcal{D}_{t,N}(\mathcal{R})\}_{t,N\in\mathbb{N}}$ denote a family of distributions over a ring $\mathcal{R}$ such that for any $t, N \in \mathbb{N}$, $\mathsf{Im}(\mathcal{D}_{t,N}(\mathcal{R})) \subseteq \mathcal{R}^N$. Let $\mathbf{C}$ be a probabilistic code generation algorithm such that $\mathbf{C}(k, N, \mathcal{R})$ outputs a matrix $\mathbf{A} \in \mathcal{R}^{N\times k}$. For dimension $k = k(\kappa)$, number of samples $N = N(\kappa)$, Hamming weight of a noise vector $t = t(\kappa)$, and a ring $\mathcal{R}$, we say that the decisional $(\mathcal{D}, \mathbf{C}, \mathcal{R})$-$\mathsf{LPN}(N, k, t)$ problem is (T, ϵ)-hard if for every probabilistic distinguisher $\mathcal{B}$ running in time T, we have*

$$\left| \Pr_{\mathbf{A},\boldsymbol{s},\boldsymbol{e}} [\mathcal{B}(\mathbf{A}, \boldsymbol{b} = \mathbf{A}\cdot\boldsymbol{s} + \boldsymbol{e}) = 1] - \Pr_{\mathbf{A},\boldsymbol{u}} [\mathcal{B}(\mathbf{A}, \boldsymbol{u}) = 1] \right| \leq \epsilon,$$

where $\mathbf{A} \leftarrow \mathbf{C}(k, N, \mathcal{R})$, $\boldsymbol{s} \leftarrow \mathcal{R}^k$, $\boldsymbol{e} \leftarrow \mathcal{D}_{t,N}(\mathcal{R})$ and $\boldsymbol{u} \leftarrow \mathcal{R}^N$. We say that the computational $(\mathcal{D}, \mathbf{C}, \mathcal{R})$-$\mathsf{LPN}(k, N, t)$ problem is (T, ϵ)-hard if for every probabilistic algorithm $\mathcal{B}$ running in time T, we have

$$\Pr_{\mathbf{A},\boldsymbol{s},\boldsymbol{e}} [\mathcal{B}(\mathbf{A}, \boldsymbol{b} = \mathbf{A}\cdot\boldsymbol{s} + \boldsymbol{e}) = (\boldsymbol{s}, \boldsymbol{e})] \leq \epsilon,$$

where $\mathbf{A}, \boldsymbol{s}, \boldsymbol{e}$ are defined as above.

In the above definition, both T and ϵ are functions of computational security parameter κ. Following the previous work, we consider the following families of noise distributions:

- **Bernoulli.** Let $\mathsf{Ber}(\mathcal{R}) = \{\mathsf{Ber}_{\mu,N}(\mathcal{R})\}_{\mu,N}$ be the family of Bernoulli distributions. In particular, $\mathsf{Ber}_{\mu,N}(\mathcal{R})$ is a Bernoulli distribution with parameters μ, N over a ring $\mathcal{R}$, such that each component in a noise vector sampled from $\mathsf{Ber}_{\mu,N}(\mathcal{R})$ is a uniform element in $\mathcal{R}$ with probability μ and 0 otherwise. Following prior works (e.g., [15,27,37,52]), we adopt such Bernoulli definition which samples a uniform element in $\mathcal{R}$ with probability μ. Note that the definition is equivalent to sampling a uniform *non-zero* element in $\mathcal{R}$ with probability $\mu(|\mathcal{R}| - 1)/|\mathcal{R}|$ for each component. One notational benefit we enjoy with this definition is that if $\boldsymbol{e}$ follows $\mathsf{Ber}_{\mu,N}(\mathcal{R})$ then any bit vector, formed by taking one bit from each corresponding component in $\boldsymbol{e}$, follows $\mathsf{Ber}_{\mu,N}(\mathbb{F}_2)$ for the same parameter μ.
- **Exact.** Let $\mathsf{HW}(\mathcal{R}) = \{\mathsf{HW}_{t,N}(\mathcal{R})\}_{t,N}$ be the family of exact noise distributions. In particular, for $\mathsf{HW}_{t,N}(\mathcal{R})$, each component of a noise vector is a uniform non-zero element in t random positions and zero elsewhere. Informally, we refer to LPN with exact noise distributions as exact-LPN.

– **Regular.** To achieve better efficiency, a series of works, e.g., [7,14–18,24,47,76,82], adopt the family of *regular* noise distributions, denoted by $\mathsf{RHW}(\mathcal{R}) = \{\mathsf{RHW}_{t,N}(\mathcal{R})\}_{t,N}$. In addition to fixed Hamming weight, the noise vector is further divided into t consecutive sub-vectors of size $\lfloor N/t \rfloor$, where each sub-vector has a single noisy coordinate. Sometimes, we refer to LPN with regular noise distributions as regular-LPN.

The existing LPN-based PCG-like protocols adopt the latter two noise distributions, and the standard LPN assumption adopts the Bernoulli distribution. While the standard LPN assumption uses random linear codes to instantiate $\mathbf{C}$ (i.e., sampling $\mathbf{A}$ uniformly at random), multiple LPN-based protocols adopt other kinds of linear codes to obtain faster computation, including local linear codes [4], quasi-cyclic codes [60], MDPC codes [63], expand-accumulate codes [16] etc. We do not analyze the hardness of LPN problems based on quasi-cyclic codes, which needs to take into account the effect of the DOOM attack [70] that allows providing $\sqrt{N}$ computational speedup. We are not aware that other kinds of linear codes listed as above lead to significantly better attacks, compared to random linear codes. The reductions given in this work focus on the case of random linear codes, and we leave that extending them to other linear codes as a future work. To simplify the notation, we often omit $\mathbf{C}$ from the $(\mathcal{D}, \mathbf{C}, \mathcal{R})$-$\mathsf{LPN}(N, k, t)$ problem, and only write $(\mathcal{D}, \mathcal{R})$-$\mathsf{LPN}(N, k, t)$.

Below, we define the dual-LPN assumption over a general finite ring $\mathcal{R}$ with a family $\mathcal{D}$ of noise distributions, where both the decisional version and search version are described. Dual-LPN is also known as syndrome decoding.

Definition 2 (Dual LPN). *Let $\mathcal{D}(\mathcal{R})$ and $\mathbf{C}$ be as in Definition 1. For two integers N, n with $N > n$, we define*

$$\mathbf{C}^{\perp}(N, n, \mathcal{R}) = \left\{\mathbf{H} \in \mathcal{R}^{n \times N} : \mathbf{H} \cdot \mathbf{A} = \mathbf{0},\ \mathbf{A} \in \mathbf{C}(N-n, N, \mathcal{R}), \mathsf{rank}(\mathbf{H}) = n\right\}.$$

For output length $n = n(\kappa)$, number of samples $N = N(\kappa)$, noise-vector Hamming weight $t = t(\kappa)$, we say that the decisional $(\mathcal{D}, \mathbf{C}^{\perp}, \mathcal{R})$-$\mathsf{dual\text{-}LPN}(N, n, t)$ problem is (T, ϵ)-hard if for every probabilistic distinguisher $\mathcal{B}$ running in time T:

$$\left| \Pr_{\mathbf{H}, \boldsymbol{e}} [\mathcal{B}(\mathbf{H}, \mathbf{H} \cdot \boldsymbol{e}) = 1] - \Pr_{\mathbf{H}, \boldsymbol{u}} [\mathcal{B}(\mathbf{H}, \boldsymbol{u}) = 1] \right| \leq \epsilon,$$

where $\mathbf{H} \leftarrow \mathbf{C}^{\perp}(N, n, \mathcal{R})$, $\boldsymbol{e} \leftarrow \mathcal{D}_{t,N}(\mathcal{R})$ and $\boldsymbol{u} \leftarrow \mathcal{R}^N$.
We say that the computational $(\mathcal{D}, \mathbf{C}^{\perp}, \mathcal{R})$-$\mathsf{dual\text{-}LPN}(N, n, t)$ problem is (T, ϵ)-hard if for every probabilistic algorithm $\mathcal{B}$ running in time T, we have

$$\Pr_{\mathbf{H}, \boldsymbol{e}} [\mathcal{B}(\mathbf{H}, \mathbf{H} \cdot \boldsymbol{e}) = \boldsymbol{e}] \leq \epsilon,$$

where $\mathbf{H}, \boldsymbol{e}$ are defined as above.

For any fixed code generation algorithm $\mathbf{C}$ and noise distribution $\mathcal{D}$, the dual-LPN problem defined as above is equivalent to the primal-LPN problem from Definition 1 with dimension $k = N - n$ and the number of samples N. The

direction transforming an LPN instance into a dual-LPN instance directly follows the simple fact that $\mathbf{H} \cdot (\mathbf{A} \cdot \boldsymbol{s} + \boldsymbol{e}) = (\mathbf{H} \cdot \mathbf{A}) \cdot \boldsymbol{s} + \mathbf{H} \cdot \boldsymbol{e} = \mathbf{H} \cdot \boldsymbol{e}$, as $\mathbf{H}$ is the parity-check matrix of the code generated by $\mathbf{A}$. The reverse direction can be obtained in a way similar to [62, Lemma 4.9].

3 The Hardness of LPN with Regular Noise Distributions

A series of MPC and ZK protocols (e.g., [7–10,14–18,24,35,36,47,76,78,80,82]) rely on the hardness of LPN problems with regular noise distributions. Multiple prior works, e.g., [15–18,24,82], believe that regular-LPN problems are not significantly easier than exact-LPN problems, or even harder than exact-LPN for a part of parameter sets. However, no reduction from exact-LPN to regular-LPN was provided, until the recent work by Feneuil, Joux and Rivain [42]. They introduced a reduction from a (dual)-LPN problem with a regular noise distribution to that with an exact noise distribution, which is summarized in the following theorem.[4]

Theorem 1 (Theorem 1 of *[42]*, adapted). *If an exact-LPN problem* $(\mathsf{HW}, \mathbb{F})$- $\mathsf{LPN}(N, k, t)$ *is* (T, ϵ)*-hard, the regular-LPN problem* $(\mathsf{RHW}, \mathbb{F})$- $\mathsf{LPN}(N, k, t)$ *is*

$$\left(T, \epsilon \cdot \binom{N}{t} \Big/ \left(\frac{N}{t}\right)^t\right) \text{-hard.}$$

The statement also holds for dual-LPN.

The above reduction suffers from a significant security loss, i.e., the penalty factor

$$p_t = \binom{N}{t} \Big/ \left(\frac{N}{t}\right)^t = \left(\frac{t^t}{t!}\right) \cdot \prod_{i=1}^{t-1} \left(1 - \frac{i}{N}\right) = e^{t - \Theta(\ln t) - \Theta(t^2/N)} = e^{t \cdot (1 - o(1))},$$

where the Stirling's approximation $\ln(t!) = t \cdot \ln t - t + \Theta(\ln t)$ is used, and $4^{-x} \leq 1 - x \leq e^{-x}$ for $0 \leq x \leq 1/2$. Here we focus on the case of $t = o(N)$, which is satisfied by low-noise LPN problems used in the PCG setting. Meanwhile, it is not hard to see that for many non-trivial parameter selections, we have $\epsilon > e^{-t}$. Let us analyze the following dual-LPN problem

$$[\mathbf{H}_1 \ \mathbf{H}_2] \cdot \begin{pmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \end{pmatrix} = \mathbf{H}_1 \cdot \boldsymbol{e}_1 + \mathbf{H}_2 \cdot \boldsymbol{e}_2 = \boldsymbol{y},$$

where $\mathbf{H}_1 \in \mathbb{F}_q^{n \times n}$, $\mathbf{H}_2 \in \mathbb{F}^{n \times (N-n)}$, $\boldsymbol{e}_1 \in \mathbb{F}^n$ and $\boldsymbol{e}_2 \in \mathbb{F}^{N-n}$. A polynomial-time attack simply bets $\boldsymbol{e}_2 = \mathbf{0}$ and computes $\boldsymbol{e}_1 = \mathbf{H}_1^{-1} \cdot \boldsymbol{y}$ (without loss of generality,

[4] In particular, [42] considers a d-split noise, which consists of d blocks of length N/d and each block has weight t/d. For $d = t$, it corresponds to the (most often used) case of regular noise.

assuming that $\mathbf{H}_1$ is invertible), which succeeds with probability

$$\binom{n}{t}\Big/\binom{N}{t} = \prod_{i=1}^{N-n}\left(\frac{n-t+i}{n+i}\right) > \left(1-\frac{t}{n+1}\right)^{N-n} \approx e^{-\frac{t(N-n)}{n+1}}.$$

If $N \leq 2n$, a larger penalty factor p_t only implies that the regular-LPN problem $(\mathsf{RHW},\mathbb{F})\text{-}\mathsf{LPN}(N,k,t)$ becomes $(\mathsf{poly}(\kappa), p_t \cdot \epsilon)$-hard, where $p_t \cdot \epsilon > 1$. Thus, this motivates us to decrease the penalty factor to yield more conservative (yet still meaningful) results.

Prior work [42] incurs a significant security loss, because it simply uses $1/p_t$ to account for the probability that an exact noise vector is regular at the same time. We provide a new reduction with a new parameter α such that [42, Theorem 1] can be seen as a special case of $\alpha = 1$. More importantly, with large α, we are able to reduce the security loss dramatically by dividing the exponent by α, while paying only an additive price αt in dimension and number of samples.

At a high level, we give an overview of the proof idea. Given exact-LPN samples $(\mathbf{A}, \boldsymbol{b} = \mathbf{A}\cdot\boldsymbol{s}+\boldsymbol{e})$ with dimension k and noise weight t, we divide them into αt blocks, i.e., $(\mathbf{A}_i, \boldsymbol{b}_i = \mathbf{A}_i\cdot\boldsymbol{s}+\boldsymbol{e}_i)$ for $i \in [1, \alpha t]$, where α is an additional parameter. Instead of hoping that every $\boldsymbol{e}_i$ has exact weight 1 (as done by Feneuil et al. in [42]), we relax the condition to $|\boldsymbol{e}_i| \leq 1$, which occurs with higher probability (and hence less security loss), especially for large α. For each block, we add an extra random sample $(\boldsymbol{a}_i, v_i = \langle \boldsymbol{a}_i, \boldsymbol{s}\rangle + \tilde{e}_i)$ such that the vector $(\boldsymbol{e}_i^\mathsf{T}, \tilde{e}_i)$ has the exact weight 1 (i.e., the resulting noise vector is regular). This is possible if the dimension of the target regular-LPN problem is $k+\alpha t$. That is, the additional αt values would help to simulate αt values $\{v_i\}$ almost perfectly.

Theorem 2. *Let $t, N \in \mathbb{N}$, and $\alpha \geq 2$ such that $\alpha t \in \mathbb{N}$ and $(\alpha t)|N$. If the exact-LPN problem $(\mathsf{HW},\mathbb{F})\text{-}\mathsf{LPN}(N,k,t)$ is (T,ϵ)-hard, then the regular-LPN problem $(\mathsf{RHW},\mathbb{F})\text{-}\mathsf{LPN}(N+\alpha t, k+\alpha t, \alpha t)$ is $(T-\mathsf{poly}(N,k), 2^{\frac{t}{\alpha}}\cdot\epsilon)$-hard, where $\mathbb{F}$ is any finite field.*

Proof. Let $N = \alpha t m$ for some $m \in \mathbb{N}$. We parse the exact-LPN samples of $(\mathsf{HW},\mathbb{F})\text{-}\mathsf{LPN}(N,k,t)$ as αt blocks:

$$\mathbf{A} \stackrel{\text{def}}{=} \begin{bmatrix} \mathbf{A}_1 \in \mathbb{F}^{m\times k} \\ \vdots \\ \mathbf{A}_{\alpha t} \in \mathbb{F}^{m\times k} \end{bmatrix}, \ \boldsymbol{b} \stackrel{\text{def}}{=} \begin{bmatrix} \boldsymbol{b}_1 = (\mathbf{A}_1\cdot\boldsymbol{s}+\boldsymbol{e}_1) \in \mathbb{F}^m \\ \vdots \\ \boldsymbol{b}_{\alpha t} = (\mathbf{A}_{\alpha t}\cdot\boldsymbol{s}+\boldsymbol{e}_{\alpha t}) \in \mathbb{F}^m \end{bmatrix}, \text{ where } \boldsymbol{s} \leftarrow \mathbb{F}^k.$$

Let $\mathcal{E}$ be the event (not explicitly stated hereafter) that for every $i \in [1, \alpha t]$, the $\boldsymbol{e}_i$'s weight $|\boldsymbol{e}_i| \leq 1$. Then, we have that $\mathcal{E}$ occurs with probability

$$\Pr_{(\boldsymbol{e}_1^\mathsf{T},\ldots,\boldsymbol{e}_{\alpha t}^\mathsf{T})\leftarrow\mathsf{HW}_{t,N}(\mathbb{F})}[\mathcal{E}] = \frac{\binom{\alpha t}{t}\cdot\left(\frac{N}{\alpha t}\right)^t}{\binom{N}{t}} = \prod_{i=1}^{t-1}\frac{(1-\frac{i}{\alpha t})}{(1-\frac{i}{N})} > \frac{4^{\sum_{i=1}^{t-1}-\frac{i}{\alpha t}}}{1} = 2^{\frac{1}{\alpha}-\frac{t}{\alpha}},$$

where the inequality is due to $1 - x \geq 4^{-x}$ for $0 \leq x \leq 1/2$, and $x = \frac{i}{\alpha t} < \frac{1}{\alpha} \leq 1/2$. Our analysis is conditioned on $\mathcal{E}$, and thus incurs a security loss of factor $2^{\frac{1}{\alpha} - \frac{t}{\alpha}}$. Sample row vectors $\boldsymbol{r}_1^\mathsf{T}, \ldots, \boldsymbol{r}_{\alpha t}^\mathsf{T} \leftarrow \mathbb{F}^{k+\alpha t}$. Condition on that they are linearly independent, which has probability more than $1 - |\mathbb{F}|^{-k}$ (see, e.g., [55,83]), pick any full-rank matrix $\mathbf{B} \in \mathbb{F}^{k \times (k+\alpha t)}$ such that $\mathbf{M}$ defined below has full rank

$$\mathbf{M} \stackrel{\text{def}}{=} \begin{bmatrix} \mathbf{B}^\mathsf{T} & \boldsymbol{r}_1 & \ldots & \boldsymbol{r}_{\alpha t} \end{bmatrix}^\mathsf{T} \in \mathbb{F}^{(k+\alpha t) \times (k+\alpha t)}.$$

We denote the secret of a regular LPN instance by $\boldsymbol{x} \leftarrow \mathbb{F}^{k+\alpha t}$, subject to $\mathbf{B} \cdot \boldsymbol{x} = \boldsymbol{s}$. For each $i \in [1, \alpha t]$, we also define a random element $u_i \in \mathbb{F} \backslash \{0\}$ as follows:

$$u_i \stackrel{\text{def}}{=} \begin{cases} \text{the non-zero entry of } \boldsymbol{e}_i, & \text{if } |\boldsymbol{e}_i| = 1 \\ \text{sample a fresh } u_i \leftarrow \mathbb{F} \backslash \{0\}, & \text{if } |\boldsymbol{e}_i| = 0 \end{cases} \text{(recall } |\boldsymbol{e}_i| \leq 1 \text{ conditioned on } \mathcal{E}).$$

$$\text{Let } \mathbf{C}_i \stackrel{\text{def}}{=} \begin{bmatrix} \mathbf{A}_i \cdot \mathbf{B} \\ \boldsymbol{r}_i^\mathsf{T} - \mathbf{1}^\mathsf{T} \cdot (\mathbf{A}_i \cdot \mathbf{B}) \end{bmatrix}, \ \boldsymbol{b}_i' \stackrel{\text{def}}{=} \begin{bmatrix} \boldsymbol{b}_i = \mathbf{A}_i \cdot \mathbf{B} \cdot \boldsymbol{x} + \boldsymbol{e}_i \\ v_i = \boldsymbol{r}_i^\mathsf{T} \cdot \boldsymbol{x} + u_i - \mathbf{1}^\mathsf{T} \cdot \boldsymbol{b}_i \end{bmatrix} \text{ for } i \in [1, \alpha t],$$

where $\mathbf{1}^\mathsf{T}$ is the all-ones row vector (i.e., every component is 1). It is easy to verify that $\boldsymbol{b}_i' = \mathbf{C}_i \cdot \boldsymbol{x} + \begin{bmatrix} \boldsymbol{e}_i \\ u_i - \mathbf{1}^\mathsf{T} \cdot \boldsymbol{e}_i \end{bmatrix}$ and the noise vector $(\boldsymbol{e}_i^\mathsf{T}, u_i - \mathbf{1}^\mathsf{T} \cdot \boldsymbol{e}_i)$ has an exact weight 1.[5] Now we argue $(\mathbf{C}_i, \mathbf{b}_i')$ can be efficiently simulated. Since $\boldsymbol{x}$ is uniform over $\mathbb{F}^{k+\alpha t}$, we have that $\mathbf{M} \cdot \boldsymbol{x}$ is uniformly random over $\mathbb{F}^{k+\alpha t}$ for any full-rank matrix $\mathbf{M}$. Therefore, $\left(\boldsymbol{r}_1^\mathsf{T} \cdot \boldsymbol{x}, \ldots, \boldsymbol{r}_{\alpha t}^\mathsf{T} \cdot \boldsymbol{x}\right)$ is uniformly random over $\mathbb{F}^{\alpha t}$, even conditioned on $\mathbf{M}$, $\mathbf{B} \cdot \boldsymbol{x}$ and other variables (e.g., all the $\mathbf{A}_i$'s, $\boldsymbol{e}_i$'s, u_i's). Thus, even without knowledge of u_i and $\boldsymbol{e}_i$, the reduction can perfectly simulate the additional sample $v_i = \boldsymbol{r}_i^\mathsf{T} \cdot \boldsymbol{x} + u_i - \mathbf{1}^\mathsf{T} \cdot \boldsymbol{b}_i$ by sampling $v_i \in \mathbb{F}$ uniformly at random.

However, $(\mathbf{C}_i, \mathbf{b}_i')$ doesn't constitute the i-th block of the regular-LPN instance, since $\mathbf{A}_i \cdot \mathbf{B}$ (as part of $\mathbf{C}_i$) is *not* uniform over $\mathbb{F}^{m \times (k+\alpha t)}$ (but sampled from a k-dimensional subspace). We first complete the rest proof for the special case $\mathbb{F} = \mathbb{F}_2$ and then proceed to the general case of any finite field $\mathbb{F}$ with $|\mathbb{F}| > 2$.

Case 1: $\mathbb{F} = \mathbb{F}_2$. In this case, we have that u_i is always 1 (i.e., the only non-zero element in $\mathbb{F}_2$). We sample a random matrix $\mathbf{P}_i \leftarrow \mathbb{F}^{m \times \alpha t}$ for each $i \in [1, \alpha t]$. We define the following LPN samples, which have the same weight-1 noise $(\boldsymbol{e}_i^\mathsf{T}, u_i - \mathbf{1}^\mathsf{T} \cdot \boldsymbol{e}_i)$ as $(\mathbf{C}_i, \boldsymbol{b}_i')$.

$$\left(\begin{bmatrix} [\mathbf{A}_i \| \mathbf{P}_i] \cdot \mathbf{M} \\ \boldsymbol{r}_i^\mathsf{T} - \mathbf{1}^\mathsf{T} \cdot (\mathbf{A}_i \cdot \mathbf{B}) \end{bmatrix}, \begin{bmatrix} \boldsymbol{b}_i \\ v_i \end{bmatrix} + \begin{bmatrix} \mathbf{P}_i \cdot \begin{bmatrix} \mathbf{1}^\mathsf{T} \cdot \boldsymbol{b}_1 + v_1 - 1 \\ \vdots \\ \mathbf{1}^\mathsf{T} \cdot \boldsymbol{b}_{\alpha t} + v_{\alpha t} - 1 \end{bmatrix} \\ \mathbf{0} \end{bmatrix} \right), \tag{1}$$

[5] Strictly speaking, the noise vector is ensured to have Hamming weight 1, but its coordinates may not take non-zero values with equal probability. The issue can be easily addressed by shuffling the matrices and samples accordingly.

which can be verified by comparing their difference, i.e.,

$$\begin{aligned}&[\mathbf{A}_i \| \mathbf{P}_i] \cdot \mathbf{M} \cdot \boldsymbol{x} + \boldsymbol{e}_i \\ &= (\mathbf{A}_i \cdot \mathbf{B} \cdot \boldsymbol{x} + \boldsymbol{e}_i) + \mathbf{P}_i \cdot \begin{bmatrix} \boldsymbol{r}_1^\mathsf{T} \cdot \boldsymbol{x} \\ \vdots \\ \boldsymbol{r}_{\alpha t}^\mathsf{T} \cdot \boldsymbol{x} \end{bmatrix} = \boldsymbol{b}_i + \mathbf{P}_i \cdot \begin{bmatrix} \mathbf{1}^\mathsf{T} \cdot \boldsymbol{b}_1 + v_1 - 1 \\ \vdots \\ \mathbf{1}^\mathsf{T} \cdot \boldsymbol{b}_{\alpha t} + v_{\alpha t} - 1 \end{bmatrix} .\end{aligned}$$

Furthermore, the matrices in (1) are $2/|\mathbb{F}|^k$-close to uniform ones, which is proved in the following Lemma 2. Therefore, for each $i \in [1, \alpha t]$, the LPN samples in (1) constitute the i-th block of a regular-LPN instance $(\mathsf{RHW}, \mathbb{F})\text{-}\mathsf{LPN}(N + \alpha t, k + \alpha t, \alpha t)$. Therefore, we just feed all αt blocks as per (1) to the solver against $(\mathsf{RHW}, \mathbb{F})\text{-}\mathsf{LPN}(N+\alpha t, k+\alpha t, \alpha t)$. If it returns $\boldsymbol{x}$, then we recover the secret vector $\boldsymbol{s} := \mathbf{B} \cdot \boldsymbol{x}$ of the exact-LPN instance $(\mathsf{HW}, \mathbb{F})\text{-}\mathsf{LPN}(N, k, t)$. Quantitatively, if one breaks $(\mathsf{RHW}, \mathbb{F})\text{-}\mathsf{LPN}(N+\alpha t, k+\alpha t, \alpha t)$ with probability p, then it can also break $(\mathsf{HW}, \mathbb{F})\text{-}\mathsf{LPN}(N, k, t)$ with probability at least $2^{\frac{1}{\alpha} - \frac{t}{\alpha}} \cdot (p - 2 \cdot |\mathbb{F}|^{-k}) \geq p \cdot 2^{-\frac{t}{\alpha}}$.

Case 2: $|\mathbb{F}| > 2$. In this case, we have that u_i is uniform over $\mathbb{F} \backslash \{0\}$. The reduction can be oblivious of u_i by letting the secret absorb u_i. We define $\boldsymbol{x}'$ such that $\mathbf{B} \cdot \boldsymbol{x}' \equiv \mathbf{B} \cdot \boldsymbol{x}$ and for all $i \in [1, \alpha t]$, $\boldsymbol{r}_i^\mathsf{T} \cdot \boldsymbol{x}' \equiv \boldsymbol{r}_i^\mathsf{T} \cdot \boldsymbol{x} + u_i - 1$, i.e.,

$$\mathbf{M} \cdot \boldsymbol{x}' \equiv \mathbf{M} \cdot \boldsymbol{x} + \left(\boldsymbol{h} \stackrel{\text{def}}{=} [\underbrace{0, \ldots, 0}_{k}, (u_1 - 1), \ldots, (u_{\alpha t} - 1)]^\mathsf{T} \right) ,$$

which is always possible by letting $\boldsymbol{x}' \stackrel{\text{def}}{=} \boldsymbol{x} + \mathbf{M}^{-1} \cdot \boldsymbol{h}$ for any invertible $\mathbf{M}$. Therefore, the reduction in Case 1 still works in Case 2 by considering $\boldsymbol{x}'$ instead of $\boldsymbol{x}$, where $\mathbf{B} \cdot \boldsymbol{x}' = \boldsymbol{s}$ and $\boldsymbol{r}_i^\mathsf{T} \cdot \boldsymbol{x}' = \mathbf{1}^\mathsf{T} \cdot \boldsymbol{b}_i + v_i - 1$ just like in Case 1. □

Lemma 2. *Let $\mathbf{A}_i$, $\mathbf{P}_i$, $\boldsymbol{r}_i^\mathsf{T}$ for $i \in [1, \alpha t]$, $\mathbf{B}$ and $\mathbf{M}$ be as defined in the proof of Theorem 2. Then,*

$$\mathsf{SD}\left(\left(\begin{bmatrix} [\mathbf{A}_1 \| \mathbf{P}_1] \cdot \mathbf{M} \\ \boldsymbol{r}_1^\mathsf{T} - \mathbf{1}^\mathsf{T}(\mathbf{A}_1 \mathbf{B}) \end{bmatrix}, \ldots, \begin{bmatrix} [\mathbf{A}_{\alpha t} \| \mathbf{P}_{\alpha t}] \cdot \mathbf{M} \\ \boldsymbol{r}_{\alpha t}^\mathsf{T} - \mathbf{1}^\mathsf{T}(\mathbf{A}_{\alpha t} \mathbf{B}) \end{bmatrix} \right), (U_{\mathbb{F}}^{(m+1) \times (k + \alpha t)})^{\alpha t} \right) \leq 2 \cdot |\mathbb{F}|^{-k},$$

where $\mathsf{SD}(\cdot, \cdot)$ denotes the statistical distance between two distributions, and $U_{\mathbb{F}}^{m \times n}$ denotes the uniform distribution over $\mathbb{F}^{m \times n}$.

The proof of Lemma 2 is given in the full version of the paper [58]. We also obtain a similar result for dual-LPN in the following Corollary 1 via the reductions between LPN and dual-LPN (see Sect. 2.2).

Corollary 1. *Let $t, N \in \mathbb{N}$ and $\alpha \geq 2$ such that $\alpha t \in \mathbb{N}$ and $(\alpha t) | N$. If the exact-dual-LPN problem $(\mathsf{HW}, \mathbb{F})\text{-}\mathsf{dual\text{-}LPN}(N, n, t)$ is (T, ϵ)-hard, then the regular-dual-LPN problem $(\mathsf{RHW}, \mathbb{F})\text{-}\mathsf{dual\text{-}LPN}(N + \alpha t, n, \alpha t)$ is $(T - \mathsf{poly}(N, n), 2^{\frac{t}{\alpha}} \cdot \epsilon)$-hard.*

The reduction underlying Theorem 2 can be generalized to that from standard LPN (with Bernoulli or exact noise distributions) to LPN with d-split noise

distributions (refer to Footnote 4). To avoid redundancy, we sketch how to adapt the proof. Similar to the proof of Theorem 2, for each i-th block ($1 \le i \le \alpha d$), introduce t/d additional random samples in the form of

$$\{(\boldsymbol{a}_{i,j}, v_{i,j} = \langle \boldsymbol{a}_{i,j}, \boldsymbol{s} \rangle + \tilde{e}_{i,j})\}_{j \in [1, t/d]}$$

such that the vector $(\boldsymbol{e}_i^\mathsf{T}, \tilde{e}_{i,1}, \cdots \tilde{e}_{i,t/d})$ possesses an exact weight of t/d. This incurs less security loss than Theorem 2 as it only requires $|\boldsymbol{e}_i^\mathsf{T}| \le t/d$ (instead of $|\boldsymbol{e}_i^\mathsf{T}| \le 1$) when the dimension of the target αd-split LPN problem is $k + \alpha t$. Consequently, the additional αt dimensions help to realize the almost-perfect simulation of αt values $\{v_{i,j}\}$.

4 The Hardness of LPN over Integer Rings

LPN over an integer ring (e.g., $\mathbb{Z}_{2^\lambda}$) has been used in VOLE and ZK protocols [8, 9, 57, 69], where these VOLE protocols could also benefit other works that need VOLE over integer rings like the MPC protocol SPD$\mathbb{Z}_{2^k}$ [28, 30]. The current security estimate of LPN over $\mathbb{Z}_{2^\lambda}$ in prior works is directly adapted from that for LPN over a field $\mathbb{F}$ of size $|\mathbb{F}| \approx 2^\lambda$ [15]. As we will show in this section the hardness of LPN over $\mathbb{Z}_{2^\lambda}$ is more related to that over $\mathbb{F}_2$ (rather than that over the λ-bit field). As depicted in Fig. 2, we provide the following reductions between the hardness of LPN over $\mathbb{Z}_{2^\lambda}$ and that over $\mathbb{F}_2$.

- **Decisional LPN over $\mathbb{Z}_{2^\lambda}$ → Decisional LPN over $\mathbb{F}_2$.** We show that distinguishing LPN over $\mathbb{Z}_{2^\lambda}$ with noise weight t is no harder than distinguishing LPN over $\mathbb{F}_2$ with noise weight $\frac{2^{(\lambda-1)}}{2^\lambda - 1} \cdot t \approx t/2$. This reduction directly gives an attack that reduces the noise weight by half for an LPN instance over $\mathbb{Z}_{2^\lambda}$.
- **Decisional LPN over $\mathbb{F}_2$ → Decisional LPN over $\mathbb{Z}_{2^\lambda}$.** We show that distinguishing LPN over $\mathbb{F}_2$ with noise weight t is no harder than the distinguishing attack on LPN over $\mathbb{Z}_{2^\lambda}$ with 1) non-standard Bernoulli-like integer noise of weight at most $\lambda \cdot t$; and 2) standard Bernoulli noise of weight $\approx 2^\lambda \cdot t$.
- **Computational LPN over $\mathbb{Z}_{2^\lambda}$ → Computational LPN over $\mathbb{F}_2$.** We show that a secret recovery attack on LPN over $\mathbb{Z}_{2^\lambda}$ with noise weight t is no harder than that on LPN over $\mathbb{F}_2$ with noise weight roughly $t/2$. While a generic reduction requires $k^{\omega(\lambda)}$-hardness for LPN over $\mathbb{Z}_{2^\lambda}$, we also give more efficient reductions for their weakly one-wayness that is more relevant to practical attacks and security estimates. We also discuss how to optimize the secret recovery attack on LPN over $\mathbb{Z}_{2^\lambda}$ based on that over $\mathbb{F}_2$ in practice.

We give similar reductions for LPN over a ring $\mathbb{Z}_{p^{\lambda_1} q^{\lambda_2}}$ (for any distinct primes p, q) in the full version of the paper [58, Appendix A], which can be further generalized to any ring $\mathbb{Z}_N$ for an integer N. All these reductions focus on the case of (primal)-LPN, and are easy to be generalized to the case of dual-LPN. When we give the reductions between different computational LPN variants, we assume that LPN over a field in consideration has a unique solution in the average case (except for a negligible fraction), which will simplify the analysis. Note that this

is true for most interesting parameter regimes of LPN, which give rise to cryptographic applications (e.g., PCG and public-key encryption), as demonstrated in the full version of the paper [58, Lemma 3]. For the concrete security of an LPN instance $\mathsf{LPN}(N, k, t)$ over $\mathbb{Z}_{2^\lambda}$, we can first reduce it to $\mathsf{LPN}(N, k, \frac{2^{(\lambda-1)}}{2^\lambda-1}t)$ over $\mathbb{F}_2$, and then estimate the bit security of the LPN instance over $\mathbb{F}_2$ as demonstrated in Sect. 5. Thus, we omit the detailed analysis of concrete LPN over $\mathbb{Z}_{2^\lambda}$. In the *subsequent work*, Baum et al. [9] gave a countermeasure by sampling an invertible element in $\mathbb{Z}_{2^\lambda}$ at random for each noisy coordinate to resist our attack. Given the countermeasure, we can reduce an LPN problem over a ring $\mathbb{Z}_{2^\lambda}$ to that over $\mathbb{F}_2$ with the same noise weight, using the same approach shown in Sect. 4.1. In other words, LPN over $\mathbb{Z}_{2^\lambda}$ is no harder than LPN over $\mathbb{F}_2$ under the same parameters. Therefore, when estimating the bit security of LPN over $\mathbb{Z}_{2^\lambda}$, one needs to use the cost attacking LPN over $\mathbb{F}_2$ as an upper bound.

4.1 Reduction from Decisional LPN over $\mathbb{Z}_{2^\lambda}$ to LPN over $\mathbb{F}_2$

We start with a simple observation that the distinguishing attack on LPN over $\mathbb{Z}_{2^\lambda}$ can be based on that over $\mathbb{F}_2$ with roughly halved noise weight. Specifically, we have the following theorem.

Theorem 3. *If the decisional exact-LPN problem* $(\mathsf{HW}, \mathbb{Z}_{2^\lambda})$-$\mathsf{LPN}(N, k, t)$ *is* (T, ϵ)*-hard, then the decisional exact-LPN problem* $(\mathsf{HW}, \mathbb{F}_2)$-$\mathsf{LPN}(N, k, \frac{2^{(\lambda-1)}}{2^\lambda-1}t)$ *is* $(T - \mathsf{poly}(N, k), O(\sqrt{t} \cdot \epsilon))$*-hard.*

The above statement can be generalized to the case of Bernoulli distributions. If the decisional LPN problem $(\mathsf{Ber}, \mathbb{Z}_{2^\lambda})$-$\mathsf{LPN}(N, k, \mu)$ *is* (T, ϵ)*-hard, then the decisional LPN problem* $(\mathsf{Ber}, \mathbb{F}_2)$-$\mathsf{LPN}(N, k, \mu)$ *is* $(T - \mathsf{poly}(N, k), O(\epsilon))$*-hard.*

Proof. Given LPN samples over a ring $\mathbb{Z}_{2^\lambda}$ $(\mathbf{A}, \boldsymbol{b} = \mathbf{A} \cdot \boldsymbol{s} + \boldsymbol{e})$, we observe that least significant bits (LSBs) of these samples $(\mathbf{A}^0 := \mathbf{A} \mod 2, \boldsymbol{b}^0 := \boldsymbol{b} \mod 2)$ constitute exactly the LPN samples over $\mathbb{F}_2$ for noise $\boldsymbol{e}^0 = \boldsymbol{e} \mod 2$. In case that $\boldsymbol{e} \leftarrow \mathsf{HW}_{t,N}(\mathbb{Z}_{2^\lambda})$, the noise vector $\boldsymbol{e}^0$ follows a Bernoulli-like distribution over $\mathbb{F}_2^N$, which is sampled by first picking t out of N coordinates at random and then filling in these t coordinates with random non-zero elements over $\mathbb{Z}_{2^\lambda}$ (and the rest with zeros). Thus, overall $\boldsymbol{e}^0$ has expected weight $t' = \frac{2^{(\lambda-1)}}{2^\lambda-1} \cdot t$, where $\frac{2^{(\lambda-1)}}{2^\lambda-1}$ is the probability that a random non-zero element of $\mathbb{Z}_{2^\lambda}$ is odd. By Lemma 1, this implies that with probability $\Omega(1/\sqrt{t})$, the noise vector $\boldsymbol{e}^0$ follows the exact noise distribution $\mathsf{HW}_{t',N}(\mathbb{F}_2)$. On the other hand, the LSBs of $(\mathbf{A}, \boldsymbol{u})$ with a uniform $\boldsymbol{u} \in \mathbb{Z}_{2^\lambda}$ are uniform as well. Therefore, one can use the solver of $(\mathsf{HW}, \mathbb{F}_2)$-$\mathsf{LPN}(N, k, t')$ to distinguish $(\mathbf{A}^0, \boldsymbol{b}^0)$ from uniform samples. The proof for the second statement is likewise, except when taking the LSBs of $\boldsymbol{e} \leftarrow \mathsf{Ber}_{\mu,N}(\mathbb{Z}_{2^\lambda})$ we immediately get $\boldsymbol{e}^0 \sim \mathsf{Ber}_{\mu,N}(\mathbb{F}_2)$ as desired. □

Despite the preserved noise probability μ in the case of Bernoulli distribution, we note that $\mathsf{Ber}_{\mu,N}(\mathbb{Z}_{2^\lambda})$ has expected weight $(1 - 2^{-\lambda})\mu N$, while $\mathsf{Ber}_{\mu,N}(\mathbb{F}_2)$ has expected weight $\mu N/2$ that is roughly $2\times$ smaller than $\mathsf{Ber}_{\mu,N}(\mathbb{Z}_{2^\lambda})$. We

can transform regular-LPN samples into exact-LPN samples by randomly shuffling these samples, and thus obtain a reduction from the decisional regular-LPN problem $(\mathsf{RHW}, \mathbb{Z}_{2^\lambda})$-$\mathsf{LPN}(N, k, t)$ to the decisional exact-LPN problem $(\mathsf{HW}, \mathbb{F}_2)$-$\mathsf{LPN}(N, k, \frac{2^{(\lambda-1)}}{2^\lambda - 1}t)$. The reductions directly give an efficient attack to reduce the noise weight of an exact-LPN or regular-LPN instance over a ring $\mathbb{Z}_{2^\lambda}$ by half.

4.2 Reduction from LPN over $\mathbb{F}_2$ to Decisional LPN over $\mathbb{Z}_{2^\lambda}$

We first show that the LPN assumption over $\mathbb{F}_2$ implies that over $\mathbb{Z}_{2^\lambda}$ under the standard Bernoulli noise distribution. However, we achieve the goal by paying a price in the security loss due to the dependence among different noise vectors. As a result, we get the very conservative statement that decisional LPN over $\mathbb{F}_2$ with noise weight t is no harder than decisional LPN over $\mathbb{Z}_{2^\lambda}$ with noise weight roughly $2^\lambda t$. We then introduce more useful Bernoulli-like noise distributions to enable more efficient reductions. In particular, we can reduce to an LPN over $\mathbb{Z}_{2^\lambda}$ with noise weight λt.

Theorem 4. *If decisional* $(\mathsf{Ber}, \mathbb{F}_2)$-$\mathsf{LPN}(N, k, \mu/2^\lambda)$ *is* (T, ϵ)*-hard, then decisional* $(\mathsf{Ber}, \mathbb{Z}_{2^\lambda})$-$\mathsf{LPN}(N, k, \mu)$ *is* $(T - \mathsf{poly}(N, k), \lambda \cdot \epsilon)$*-hard.*

Proof. Let $(\mathbf{A},\ \boldsymbol{b} = \mathbf{A} \cdot \boldsymbol{s} + \boldsymbol{e})$ be LPN samples over $\mathbb{Z}_{2^\lambda}$. Decompose the matrix and vectors into λ ones over $\mathbb{F}_2$ as follows: $(\mathbf{A}^0, \mathbf{A}^1, \cdots, \mathbf{A}^{\lambda-1}) := \mathsf{BitDecomp}(\mathbf{A})$, $(\boldsymbol{s}^0, \boldsymbol{s}^1, \cdots, \boldsymbol{s}^{\lambda-1}) := \mathsf{BitDecomp}(\boldsymbol{s})$, $(\boldsymbol{e}^0, \boldsymbol{e}^1, \cdots, \boldsymbol{e}^{\lambda-1}) := \mathsf{BitDecomp}(\boldsymbol{e})$ and $(\boldsymbol{b}^0, \boldsymbol{b}^1, \cdots, \boldsymbol{b}^{\lambda-1}) := \mathsf{BitDecomp}(\boldsymbol{b})$. Therefore, for $i \in [\lambda]$, $\boldsymbol{b}^i$ depends only on $\mathbf{A}$, $(\boldsymbol{s}^i, \ldots, \boldsymbol{s}^0)$, $(\boldsymbol{e}^i, \ldots, \boldsymbol{e}^0)$, and we write it as $\boldsymbol{b}^i = \mathbf{A}^0 \cdot \boldsymbol{s}^i + \boldsymbol{e}^i + f_i\big(\mathbf{A}, \mathbf{S}(0, i-1), \mathbf{E}(0, i-1)\big) \mod 2$, where $\mathbf{S}(0, i-1) \stackrel{\text{def}}{=} (\boldsymbol{s}^{i-1}, \ldots, \boldsymbol{s}^0)$, and $\mathbf{E}(0, i-1) \stackrel{\text{def}}{=} (\boldsymbol{e}^{i-1}, \ldots, \boldsymbol{e}^0)$, and f_i sums up the other terms not depending on $\boldsymbol{s}^i$ and $\boldsymbol{e}^i$. Define the hybrid distributions $H_0, H_1, \cdots, H_\lambda$, where each $H_i = (\mathbf{A}, \boldsymbol{b}^0, \cdots, \boldsymbol{b}^{i-1}, \boldsymbol{u}_i\ \cdots\ \boldsymbol{u}_{\lambda-1})$ and $\boldsymbol{u}_j \leftarrow \mathbb{F}_2^N$ for $j \in [\lambda]$ is sampled independently at random. Note that all the $\boldsymbol{s}^i$'s are independent and uniformly random. Therefore, for $i \in [\lambda]$, by the decisional $(\mathsf{Ber}, \mathbb{F}_2)$-$\mathsf{LPN}$ assumption,

$$\big(\mathbf{A}^0, \boldsymbol{u}_i, \mathbf{S}(0, i-1), \mathbf{E}(0, i-1)\big) \approx_c \big(\mathbf{A}^0, \mathbf{A}^0 \cdot \boldsymbol{s}^i + \boldsymbol{e}^i \mod 2, \mathbf{S}(0, i-1), \mathbf{E}(0, i-1)\big)$$

where $\mathbf{S}(0, i-1)$ is independent of any other variables, and the actual noise rate of LPN is that of $\boldsymbol{e}^i$ conditioned on $\mathbf{E}(0, i-1)$ (see analysis blow). This implies

$$\big(\mathbf{A}, \boldsymbol{b}^0, \cdots, \boldsymbol{b}^{i-1}, \boldsymbol{u}_i + f_i(\mathbf{A}, \mathbf{S}(0, i-1), \mathbf{E}(0, i-1)) \mod 2\big) \approx_c \big(\mathbf{A}, \boldsymbol{b}^0, \cdots, \boldsymbol{b}^{i-1}, \boldsymbol{b}^i\big)$$

which in turn implies $H_i \approx_c H_{i+1}$, where $\boldsymbol{b}^0, \ldots, \boldsymbol{b}^{i-1}$, $f_i(\mathbf{A}, \mathbf{S}(0, i-1), \mathbf{E}(0, i-1))$ can be efficiently computed from $\mathbf{A}, \mathbf{S}(0, i-1), \mathbf{E}(0, i-1)$.

Therefore, if all the adjacent H_i and H_{i+1} are computationally indistinguishable except with probability ϵ, then H_0 and H_λ are computationally indistinguishable by a hybrid argument except with probability $\lambda \cdot \epsilon$. It thus remains to estimate the noise rate needed by the LPN assumption. Consider a single noise

sample $(\boldsymbol{e}^0[j], \boldsymbol{e}^1[j], \ldots, \boldsymbol{e}^{\lambda-1}[j]) \leftarrow \mathsf{Ber}_{\mu,N}(\mathbb{Z}_{2^\lambda})$, where $\boldsymbol{e}^i[j]$ is the j-th entry of $\boldsymbol{e}^i$. Conditioned on any non-zero $(\boldsymbol{e}^0[j], \ldots, \boldsymbol{e}^{i-1}[j])$, $\boldsymbol{e}^i[j]$ is uniformly random and thus unconditionally masks the corresponding $\boldsymbol{b}^i[j]$. Otherwise, we have that

$$\Pr\left[\boldsymbol{e}^i[j] = 1 \,\middle|\, (\boldsymbol{e}^0[j], \ldots, \boldsymbol{e}^{i-1}[j]) = 0^i\right] = \frac{\mu \cdot 2^{-(i+1)}}{1 - \mu + \mu \cdot 2^{-i}} \geq \mu \cdot 2^{-(i+1)}$$

is the noise rate needed to keep the computational indistinguishability between H_i and H_{i+1}, which reaches its minimum $\mu \cdot 2^{-\lambda}$ when $i = \lambda - 1$. □

Based on the above theorem, we easily obtain the following corollary, with its proof given in the full version of the paper [58].

Corollary 2. *If decisional* $(\mathsf{Ber}, \mathbb{F}_2)$-$\mathsf{LPN}(N, k, \mu/2^\lambda)$ *is hard, then computational* $(\mathsf{HW}, \mathbb{Z}_{2^\lambda})$-$\mathsf{LPN}(N, k, t = (1 - 2^{-\lambda})\mu N)$ *is hard.*

The dependency among the noise vectors $\{\boldsymbol{e}^i\}$ incurs a significant loss during the reduction. This motivates us to introduce two specific noise distributions, i.e., $\mathsf{IndBer}_{\mu,N}(\mathbb{Z}_{2^\lambda})$ and $\mathsf{IndHW}_{t,N}(\mathbb{Z}_{2^\lambda})$, where Ind refers that the noise's bit-decomposition $\boldsymbol{e}^0$, …, $\boldsymbol{e}^{\lambda-1}$ are independent and identically distributed, and parameter μ (resp., t) is noise rate (resp., weight) of each $\boldsymbol{e}^i$.

- $\mathsf{IndBer}_{\mu,N}(\mathbb{Z}_{2^\lambda})$ is bit-wise independent. By $\boldsymbol{e} \leftarrow \mathsf{IndBer}_{\mu,N}(\mathbb{Z}_{2^\lambda})$, we mean that $\boldsymbol{e} := \sum_{i=0}^{\lambda-1} 2^i \cdot \boldsymbol{e}^i \in \mathbb{Z}_{2^\lambda}$ with $\boldsymbol{e}^i \leftarrow \mathsf{Ber}_{\mu,N}(\mathbb{F}_2)$ for $i \in [\lambda]$. The noise rate of $\mathsf{IndBer}_{\mu,N}(\mathbb{Z}_{2^\lambda})$ is the probability that a coordinate of $\boldsymbol{e}$ is non-zero, i.e., $1 - (1 - \mu/2)^\lambda \leq \lambda\mu/2$ by Bernoulli's inequality. Therefore, the expected Hamming weight of $\boldsymbol{e} \leftarrow \mathsf{IndBer}_{\mu,N}(\mathbb{Z}_{2^\lambda})$ is λt where $t = \mu N/2$.
- $\mathsf{IndHW}_{t,N}(\mathbb{Z}_{2^\lambda})$ decomposes into λ independent vectors from $\mathsf{HW}_{t,N}(\mathbb{F}_2)$. By $\boldsymbol{e} \leftarrow \mathsf{IndHW}_{t,N}(\mathbb{Z}_{2^\lambda})$, we mean that $\boldsymbol{e} := \sum_{i=0}^{\lambda-1} 2^i \cdot \boldsymbol{e}^i$ with $\boldsymbol{e}^i \leftarrow \mathsf{HW}_{t,N}(\mathbb{F}_2)$ for $i \in [\lambda]$. It is easy to see that the Hamming weight of $\boldsymbol{e}$ is at most λt.

Although $\mathsf{IndBer}_{\mu,N}(\mathbb{Z}_{2^\lambda})$ and $\mathsf{IndHW}_{t,N}(\mathbb{Z}_{2^\lambda})$ have not been used in existing protocols, LPN with such noise distributions can be used to design PCG-like VOLE protocols by running these protocols with maximum weight λt. The PCG-like VOLE protocols employing the non-standard noise distributions are approximately $\lambda/2$ times less efficient than the state-of-the-art protocol [9] using LPN with regular noise distributions over $\mathbb{Z}_{2^\lambda}$. Despite their lower efficiency, these PCG-like VOLE protocols enjoy (1) that the underlying LPN problem over $\mathbb{Z}_{2^\lambda}$ is tightly equivalent to LPN over $\mathbb{F}_2$; (2) a simpler approach to detect malicious behaviors. Below, we show that decisional LPN over $\mathbb{F}_2$ with noise weight t is *tightly* equivalent to decisional LPN over $\mathbb{Z}_{2^\lambda}$ with noise weight roughly λt under the new noise distributions. The proof of Theorem 5 is detailed in the full version of the paper [58].

Theorem 5. *Let* $(\mathcal{D}_1, \mathcal{D}_2, w) \in \{(\mathsf{Ber}, \mathsf{IndBer}, \mu), (\mathsf{HW}, \mathsf{IndHW}, t)\}$ *and we have:*

- *If decisional* $(\mathcal{D}_1, \mathbb{F}_2)$-$\mathsf{LPN}(N, k, w)$ *is* (T, ϵ)*-hard, then decisional* $(\mathcal{D}_2, \mathbb{Z}_{2^\lambda})$-$\mathsf{LPN}(N, k, w)$ *is* $(T - \mathsf{poly}(N, k), \lambda \cdot \epsilon)$*-hard.*

Algorithm 1: $\mathcal{A}_{\mathrm{LPN}_{2^\lambda}}$, the secret recovery algorithm on LPN over $\mathbb{Z}_{2^\lambda}$ ($\lambda \geq 2$) with oracle access to $\mathcal{A}_{\mathrm{LPN}_2}$ (the solver for LPN over $\mathbb{F}_2$).

Input: $(\mathcal{D}, \mathbb{Z}_{2^\lambda})$-$\mathsf{LPN}(N, k, t)$ samples $(\mathbf{A}, \boldsymbol{b} = \mathbf{A} \cdot \boldsymbol{s} + \boldsymbol{e} \mod 2^\lambda)$
Output: $\boldsymbol{s} \in \mathbb{Z}_{2^\lambda}$

$(\mathbf{A}^0, \mathbf{A}^1, \cdots, \mathbf{A}^{\lambda-1}) := \mathsf{BitDecomp}(\mathbf{A})$;
$(\boldsymbol{b}^0, \boldsymbol{b}^1, \cdots, \boldsymbol{b}^{\lambda-1}) := \mathsf{BitDecomp}(\boldsymbol{b})$;
$(\boldsymbol{s}^0, \boldsymbol{e}^0) \leftarrow \mathcal{A}_{\mathrm{LPN}_2}(\mathbf{A}^0, \boldsymbol{b}^0)$;
$\boldsymbol{b}' := (\boldsymbol{b} - \mathbf{A} \cdot \boldsymbol{s}^0 - \boldsymbol{e}^0)/2 \mod 2^{(\lambda-1)}$;
Return $\boldsymbol{s} = \boldsymbol{s}^0 + 2 \cdot \mathcal{A}_{\mathrm{LPN}_{2^{(\lambda-1)}}}\left(\mathbf{A}' := \sum_{i=0}^{\lambda-2} 2^i \cdot \mathbf{A}^i \in \mathbb{Z}_{2^{\lambda-1}}, \boldsymbol{b}'\right)$.

– *If decisional* $(\mathcal{D}_2, \mathbb{Z}_{2^\lambda})$-$\mathsf{LPN}(N, k, w)$ *is* (T, ϵ)*-hard, then decisional* $(\mathcal{D}_1, \mathbb{F}_2)$-$\mathsf{LPN}(N, k, w)$ *is* $(T - \mathsf{poly}(N, k), \epsilon)$*-hard.*

On the Choice of Matrix A. As we can see from the proofs of Theorem 4, Theorem 5 and Theorem 6 (shown in Sect. 4.3), all the reductions only rely on that $\mathbf{A}^0$ is uniformly distributed over $\mathbb{F}_2^{N \times k}$ while $\mathbf{A}^1, \cdots, \mathbf{A}^{\lambda-1}$ can be arbitrary (or even zero matrix), where $(\mathbf{A}^0, \mathbf{A}^1, \ldots, \mathbf{A}^{\lambda-1}) := \mathsf{BitDecomp}(\mathbf{A})$. In other words, it suffices to use a Boolean matrix $\mathbf{A} = \mathbf{A}^0$, and the choices of $\mathbf{A}^1, \ldots, \mathbf{A}^{\lambda-1}$ do not introduce any further hardness to the LPN problem over $\mathbb{Z}_{2^\lambda}$. Overall, we give a positive result that LPN over a ring $\mathbb{Z}_{2^\lambda}$ with Boolean matrices is secure if the corresponding LPN over binary field $\mathbb{F}_2$ is secure.

4.3 Reduction from Computational LPN over $\mathbb{Z}_{2^\lambda}$ to LPN over $\mathbb{F}_2$

In the computational setting, we show that an LPN instance over $\mathbb{Z}_{2^\lambda}$ can be efficiently translated to λ instances of LPN over $\mathbb{F}_2$, which are independent except that they share the same random matrix $\mathbf{A}^0$ over $\mathbb{F}_2$ and that the noise vectors of the λ instances are somehow correlated. We refer to the proof of Theorem 6 on how to address the correlation issue. Here we give a reduction from computational LPN over a ring $\mathbb{Z}_{2^\lambda}$ to that over $\mathbb{F}_2$ by extending the corresponding reduction between their decisional versions shown in Sect. 4.1. Algorithm 1 shows how computational LPN over $\mathbb{Z}_{2^\lambda}$ is reduced to that over $\mathbb{Z}_{2^{\lambda-1}}$. The correctness of this reduction is analyzed in Lemma 3, and its proof is available in the complete version of the paper [58]. Note that by recursion, $\mathcal{A}_{\mathrm{LPN}_{2^\lambda}}$ degenerates to secret recovery algorithm for LPN over $\mathbb{F}_2$ when $\lambda = 1$. Without loss of generality, we assume that $\mathcal{A}_{\mathrm{LPN}_2}$ returns the noise vector in addition to the recovered secret.

Lemma 3. *Let* $(\mathbf{A}, \boldsymbol{b} = \mathbf{A} \cdot \boldsymbol{s} + \boldsymbol{e} \mod 2^\lambda)$ *be the LPN samples over* $\mathbb{Z}_{2^\lambda}$*, then* $(\mathbf{A}', \boldsymbol{b}')$ *as defined in Algorithm 1 constitute the LPN samples over* $\mathbb{Z}_{2^{(\lambda-1)}}$*, where* $\mathbf{A}' = \sum_{i=0}^{\lambda-2} 2^i \cdot \mathbf{A}^i \mod 2^{(\lambda-1)}$, $\boldsymbol{b}' = \mathbf{A}' \cdot \boldsymbol{s}' + \boldsymbol{e}' \mod 2^{(\lambda-1)}$, $\boldsymbol{s}' = \sum_{i=1}^{\lambda-1} 2^{i-1} \cdot \boldsymbol{s}^i \mod 2^{(\lambda-1)}$ *and* $\boldsymbol{e}' = \sum_{i=1}^{\lambda-1} 2^{i-1} \cdot \boldsymbol{e}^i \mod 2^{(\lambda-1)}$.

Below, we show that $(\epsilon^{\lambda+1})$-hard computational LPN over $\mathbb{Z}_{2^\lambda}$ implies (2ϵ)-hard LPN over $\mathbb{F}_2$. Here $\lambda = O(1)$ needs to be small in general for polynomial hardness, and it can be up to $\lambda = k^{\Theta(1)}$ for sub-exponential hardness, e.g., $\lambda = k^{0.25}$ and $\epsilon = 2^{-k^{0.25}}$. The proofs of Theorem 6 and Theorem 7 are detailed in the full version of the paper [58].

Theorem 6. *If computational* $(D_1, \mathbb{Z}_{2^\lambda})$-$\mathsf{LPN}(N, k, w)$ *is* $(\lambda \cdot T + \mathsf{poly}(N, k), \epsilon^{\lambda+1})$*-hard, then computational* $(D_2, \mathbb{F}_2)$-$\mathsf{LPN}(N, k, w)$ *is* $(T, 2\epsilon)$*-hard, where* $(D_1, D_2, w) \in \{(\mathsf{Ber}, \mathsf{Ber}, \mu), (\mathsf{IndBer}, \mathsf{Ber}, \mu), (\mathsf{IndHW}, \mathsf{HW}, t)\}$.

Theorem 7. *If computational* $(\mathsf{HW}, \mathbb{Z}_{2^\lambda})$-$\mathsf{LPN}(N, k, t)$ *is* $(\lambda \cdot T + \mathsf{poly}(N, k), \epsilon^{\lambda+1})$*-hard, then computational* $(\mathsf{HW}, \mathbb{F}_2)$-$\mathsf{LPN}(N, k, t')$ *is* $(T, \frac{2\epsilon}{1-\exp(-\delta^2 t/6)})$*-hard, where* $t' = \frac{2^{(\lambda-1)}}{2^\lambda - 1}(1+\delta)t$ *for any constant* $\delta > 0$.

Recall that we can transform regular-LPN samples into exact-LPN samples by randomly shuffling these samples. Therefore, we are able to obtain a reduction from the computational regular-LPN problem $(\mathsf{RHW}, \mathbb{Z}_{2^\lambda})$-$\mathsf{LPN}(N, k, t)$ to the computational exact-LPN problem $(\mathsf{HW}, \mathbb{F}_2)$-$\mathsf{LPN}(N, k, \frac{2^{(\lambda-1)}}{2^\lambda-1}(1+\delta)t)$. The above reduction suffers a significant security loss by exponent factor $1/(\lambda+1)$ since computationally intractable problems typically require a small success probability for efficient adversaries. In the setting of practical key recovery attacks, however, we often expect the success probability to be $(1 - 1/\mathsf{poly}(k))$ or even overwhelming. In this case, we get more efficient reductions as below. The proofs of Theorem 8 and Theorem 9 are provided in the full version of the paper [58].

Theorem 8. *If the computational* $(D_1, \mathbb{F}_2)$-$\mathsf{LPN}(N, k, w)$ *problem can be broken by* $\mathcal{A}_{LPN_2}$ *in time* T *with success probability at least* $(1-\epsilon)$*, then the computational* $(D_2, \mathbb{Z}_{2^\lambda})$-$\mathsf{LPN}(N, k, w)$ *problem can be broken by* $\mathcal{A}_{LPN_{2^\lambda}}$ *(see Algorithm 1) in time* $\lambda \cdot T + \mathsf{poly}(N, k)$ *with success probability at least* $1 - (\lambda+1)\sqrt{\epsilon}$*, where* $(D_1, D_2, w) \in \{(\mathsf{Ber}, \mathsf{Ber}, \mu), (\mathsf{Ber}, \mathsf{IndBer}, \mu), (\mathsf{HW}, \mathsf{IndHW}, t)\}$.

Theorem 9. *If the computational* $(\mathsf{HW}, \mathbb{F}_2)$-$\mathsf{LPN}(N, k, t')$ *problem can be broken by* $\mathcal{A}_{LPN_2}$ *in time* T *with success probability at least* $(1-\epsilon/2)$*, then the computational* $(\mathsf{HW}, \mathbb{Z}_{2^\lambda})$-$\mathsf{LPN}(N, k, t)$ *problem can be broken by* $\mathcal{A}_{LPN_{2^\lambda}}$ *(see Algorithm 1) in time* $\lambda \cdot T + \mathsf{poly}(N, k)$ *with success probability at least* $1 - (\lambda+1)\sqrt{\epsilon}$*, where* $t' = \frac{2^{\lambda-1}}{2^\lambda-1}(1+\delta)t$ *for any* δ *and* ϵ *satisfying* $\delta^2 t \geq 6\ln(2/\epsilon)$.

Optimized Attacks on $(\mathsf{Ber}/\mathsf{HW}, \mathbb{Z}_{2^\lambda})$-$\mathsf{LPN}$**.** In practice, we optimize the attacks on $(\mathsf{Ber}/\mathsf{HW}, \mathbb{Z}_{2^\lambda})$-$\mathsf{LPN}$ by exploiting the correlations among the noise vectors of the λ instances (i.e., $\boldsymbol{e}^0, \ldots, \boldsymbol{e}^{\lambda-1}$). In particular, Algorithm 1 recovers the corresponding secrets $\boldsymbol{s}^0, \boldsymbol{s}^1, \cdots, \boldsymbol{s}^{\lambda-1}$ sequentially. That means when the attacker works on the $(i+1)$-th LPN instance, it has already seen $\boldsymbol{e}^0, \ldots, \boldsymbol{e}^{i-1}$ from the previous i broken instances. As analyzed in the proof of Theorem 4, for any single noise sample $(\boldsymbol{e}^0[j], \boldsymbol{e}^1[j], \ldots, \boldsymbol{e}^{\lambda-1}[j]) \leftarrow \mathsf{Ber}_{\mu,N}(\mathbb{Z}_{2^\lambda})$, $\boldsymbol{e}^i[j]$ is uniformly random conditioned on any non-zero $(\boldsymbol{e}^0[j], \ldots, \boldsymbol{e}^{i-1}[j])$, and thus sample $\boldsymbol{b}^i[j]$ is useless (encrypted by one-time padding) and should be discarded. In

other words, the effective noise rate of the i-th LPN instance is roughly $\mu \cdot 2^{-(i+1)}$ given the attacker's knowledge about $\boldsymbol{e}^0, \ldots, \boldsymbol{e}^{i-1}$. Therefore, the success rate of solving the $(\mathsf{Ber}, \mathbb{Z}_{2^\lambda})$-$\mathsf{LPN}(N, k, \mu)$ instance is roughly the product of the λ instances of $(\mathsf{Ber}, \mathbb{F}_2)$-$\mathsf{LPN}$ with continuously halving noise rates μ, $\mu/2$, ..., $\mu/2^{\lambda-1}$. For instance, if solving these instances can succeed with probability ϵ, $\epsilon^{2^{-1}}$, ..., $\epsilon^{2^{-(\lambda-1)}}$ respectively, then it leads to a success probability of approximately ϵ^2 (instead of $\epsilon^{\lambda+1}$). The optimization for reducing $(\mathsf{HW}, \mathbb{Z}_{2^\lambda})$-$\mathsf{LPN}$ to $(\mathsf{HW}, \mathbb{F}_2)$-$\mathsf{LPN}$ is likewise.

5 Concrete Analysis of Low-Noise LPN over Finite Fields

Recently, a series of works [14–18,27,67,69,76,82] use the (dual-)LPN problem with very low noise rate over finite fields to construct concretely efficient PCG-like protocols, which extend a small number of correlations (e.g., COT, VOLE and OLE) to a large number of correlations with sublinear communication. These protocols can be used as building blocks to design a variety of MPC and ZK protocols. Therefore, the hardness of (dual-)LPN problems is crucial to guarantee the security of all the protocols.

Before our work, almost all of the known PCG-like protocols based on (dual-)LPN adopt the formulas by Boyle et al. [15] to select the concrete parameters for some specified security level. Boyle et al. [15] obtained the formulas by analyzing three attacks: Pooled Gauss [40], ISD [65] and SD [3]. However, we found some imprecisions for their analysis, which are outlined as follows:

- When analyzing the hardness of LPN with exact noise distribution $\mathsf{HW}_{t,N}(\mathbb{F})$, the formula against Pooled Gauss attack is obtained by viewing $\mathsf{HW}_{t,N}(\mathbb{F})$ as a Bernoulli distribution $\mathsf{Ber}_{t/N,N}(\mathbb{F})$, which makes the formula not accurate.
- When analyzing the hardness of LPN against ISD attacks, the formula is obtained by an upper bound of the complexity of the Prange's ISD algorithm [65] to solve LPN problems over a large field. This does not cover the advanced ISD variants [11,38,59,71]. Additionally, their analysis does not capture the impact of field sizes when calculating the ISD cost.
- When analyzing the hardness of LPN against SD attacks, each parity-check vector is assumed to be independently in compliance with a Bernoulli distribution, which is inaccurate [33].

We also give more accurate formulas on the hardness of low-noise (dual-)LPN problems, where the recent SD improvement called SD 2.0 [25] is also included. Very recently, Meyer-Hilfiger and Tillich [61] shown that the SD 2.0 algorithm can be modified to obtain the same complexity under a weaker assumption. For LPN with exact noise distributions, we compare our more accurate costs of Pooled Gauss, SD and ISD attacks with that by Boyle et al. [15] in the full version of the paper [58, Tables 6 and 7], where all the LPN parameters are adopted from [15]. Under the same LPN parameters, while Boyle et al. [15] showed that either Pooled Gauss attack or SD attack has the lowest cost, our analysis shows that ISD attack has the lowest cost. [58, Tables 6 and 7] also show

that the ISD attack has lower cost for smaller field size, which is also observed in prior works such as [42]. This justifies that it is not accurate to use the same formulas for all field sizes as in [15].

Under the Gilbert-Varshamov (GV) bound[6], Carrier et al. [25] shown that SD 2.0 outperforms all ISD algorithms for the case that the code rate $k/N < 0.3$. However, we observe that the SD 2.0 algorithm [25] does not behave better when solving the low-noise LPN problems used in the PCG-like protocols. This is because the collision technique[7] (a subroutine of SD 2.0) takes exponential time $2^{\theta(k)}$ that is much larger than the subexponential time $2^{O(k\mu)}$ to solve the low-noise LPN problem with ISD, where $\mu = 1/k^c$ is the noise rate (i.e., t/N) for constant $0 < c < 1$. Thus, in SD 2.0, we incorporate other collision techniques that are known to perform better for low-noise LPN (e.g., the one used in low-weight parity-check attack shown in [15, Sect. 2.3], originated from [85]). In the full version of the paper [58, Appendix B.2], we prove that the SD 2.0 attack [25] (that improves the SD attack) adapted to the low-noise setting require more cost than the ISD attack against $(\mathsf{HW}, \mathbb{F})$-$\mathsf{LPN}(N, k, t)$ with field size $|\mathbb{F}| \geq 4t$.

The previous analysis [15] focuses on exact noise distributions, but the recent PCG-like protocols mainly adopt regular noise distributions to achieve better efficiency. To close the gap, our analysis includes two aspects to capture the regular structure of noises. On the one hand, we transform a regular-LPN problem $(\mathsf{RHW}, \mathbb{F}_2)$-$\mathsf{LPN}(N, k, t)$ into an exact-LPN problem $(\mathsf{HW}, \mathbb{F}_2)$-$\mathsf{LPN}(N - t, k - t, t)$ based on the approach in prior works [22,41]. Then, we solve the $(\mathsf{HW}, \mathbb{F}_2)$-$\mathsf{LPN}(N - t, k - t, t)$ problem by applying established attacks, independent of the regular structure. This transformation from regular-LPN to exact-LPN works for LPN over $\mathbb{F}_2$, but fails to work for LPN over larger fields (see more details in Sect. 5.1). On the other hand, our analysis includes the recent algebraic attack by Briaud and Øygarden [22], which exploits the regular structure of noises. This attack is able to obtain lower cost for regular-LPN problems with small code rate k/N for some parameter sets. Recently, Carozza, Couteau and Joux [24] also proposed new attacks tailored to LPN with regular noises, but focus on the parameter selection satisfies the condition $(N/t)^t \leq 2^{N-k} \leq \binom{N}{t}$, which notably differs from the parameter selection used in the PCG setting. Thus, we do not cover their attacks.

For regular noise distributions, we give the costs of different attacks against LPN problems with the parameters given in [15], which is shown in Tables 2 and the full version of the paper [58, Table 4]. For the case of $\log|\mathbb{F}| = 128$ and $(N, k, t) = (2^{20}, 32771, 1419)$ or $(N, k, t) = (2^{22}, 67440, 2735)$, the algebraic attack achieves the lowest cost among these attacks. When the LPN parameters listed in Table 2 achieve the bit security at most 111, we have two choices to achieve 128-bit security: (a) increasing the dimension k; (b) increasing the noise weight t. When only increasing weight t, the algebraic attack would have a

[6] The GV bound decoding over $\mathbb{F}_2$ is to solve LPN instances that achieve the GV relative distance $t/N = \mathbf{H}^{-1}(1 - k/N)$, where $\mathbf{H}(\mu) = \mu \cdot \log(1/\mu) + (1 - \mu) \cdot \log(1/(1 - \mu))$ is the binary entropy function and $\mathbf{H}^{-1}$ is the inverse of $\mathbf{H}$.

[7] The collision technique refers to the process of finding parity check vectors.

Table 2. The bit-security of LPN problems over finite fields with number of samples N, dimension k and Hamming weight of noises t for a regular noise distribution. The abbreviation "AGB" denotes the recent algebraic attack [22].

Regular LPN over a field $\mathbb{F}$			This work ($\log\|\mathbb{F}\| = 128$)					This work ($\log\|\mathbb{F}\| = 1$)				
N	k	t	Gauss	SD	SD 2.0	ISD	AGB	Gauss	SD	SD 2.0	ISD	AGB
2^{10}	652	57	111	184	184	111	111	106	183	108	90	101
2^{12}	1589	98	100	151	151	100	107	96	146	130	80	103
2^{14}	3482	198	101	149	149	101	110	97	143	136	83	106
2^{16}	7391	389	103	147	147	103	111	99	141	138	87	108
2^{18}	15336	760	105	146	146	105	107	101	140	138	92	104
2^{20}	32771	1419	107	145	145	107	102	104	139	139	97	98
2^{22}	67440	2735	108	138	138	108	104	103	133	133	99	103

Table 3. Comparison of dimensions between exact-LPN problems and regular-LPN problems over finite fields for 128-bit security level.

#Samples	Weight	Dimension for $\log\|\mathbb{F}\| = 128$		Dimension for $\log\|\mathbb{F}\| = 1$	
N	t	Exact-LPN	Regular-LPN	Exact-LPN	Regular-LPN
2^{12}	172	1321	1377 (+4.2%)	1549	1657 (+7.0%)
2^{14}	338	2895	2909 (+0.5%)	3373	3655 (+8.3%)
2^{16}	667	6005	6091 (+1.4%)	6956	7560 (+8.7%)
2^{18}	1312	12160	14796 (+21.7%)	13898	15996 (+15.1%)
2^{20}	2467	25346	30978 (+22.2%)	28289	33354 (+17.9%)
2^{22}	4788	50854	75396 (+48.3%)	55408	80074 (+44.5%)

significantly lower cost than other attacks for some parameter sets (see the full version of the paper [58, Table 8]), which has been observed in [22]. To resist the algebraic attack and the attack strategy based on the above regular-to-exact transformation, a better choice is to increase dimension k. For example, as shown in Table 3, we need to increase the dimension of LPN problems with a regular noise distribution by 0.5%–48.3% to achieve the same 128-bit security as LPN problems with an exact noise distribution. The increase of dimension k has a negligible impact on the efficiency of PCG-like protocols, due to the usage of the Bootstrapping-iteration technique [82]. For dual-LPN problems, we note that the algebraic attack [22] has significantly more cost than Pooled Gauss and ISD attacks for all the listed parameters, as the code rate is constant (typically 1/2 or 3/4).

In this section, we aim to give more accurate formulas by adjusting the known attacks to analyze the cost of low-noise LPN problems in the PCG setting. In particular, we provide an estimator tool (see Footnote 3), which incorporates the advanced attacks being applicable to LPN problems in the PCG setting,

to automatically evaluate the bit security of low-noise LPN problems. This will help future works to select LPN parameters when designing or applying PCG-like protocols. While the recent estimator tool by Esser and Bellini [39] focuses on ISD attacks to analyze the hardness of classical LPN problems over $\mathbb{F}_2$ with an exact noise distribution in the traditional public-key setting, our estimator tool covers Pooled Gauss, SD, SD 2.0, ISD and algebraic attacks to evaluate the hardness of low-noise LPN problems over an arbitrary finite field (or integer ring) with a regular or exact noise distribution in the PCG setting.

In Sect. 5.1, we first show that $(\mathsf{RHW}, \mathbb{F}_2)$-$\mathsf{LPN}(N, k, t)$ is not harder than $(\mathsf{HW}, \mathbb{F}_2)$-$\mathsf{LPN}(N-t, k-t, t)$, and also give an overview of the algebraic attack. For LPN over larger fields, we do not find such an efficient transformation from regular-LPN to exact-LPN. Therefore, we are able to analyze the costs of Pooled Gauss, SD and ISD attacks against LPN problems in a similar way for both exact and regular noise distributions. Then, in the full version of the paper [58, Appendix B], we show the imprecisions of the previous analysis [15] and give more accurate formulas against Pooled Gauss, SD and ISD attacks for the hardness of low-noise LPN problems.

5.1 The Hardness of LPN with Regular Noise Distributions

Transformation from Regular-LPN to Exact-LPN over $\mathbb{F}_2$. Building upon prior works [22,41], we transform a regular-LPN problem $(\mathsf{RHW}, \mathbb{F}_2)$-$\mathsf{LPN}(N, k, t)$ into an exact-LPN problem $(\mathsf{HW}, \mathbb{F}_2)$-$\mathsf{LPN}(N-t, k-t, t)$. The reduction is useful for the case of $2^{N-k} > \binom{N}{t}$ which is satisfied by the LPN parameters in the PCG setting. In this case, both regular-LPN and exact-LPN problems have unique solutions for these parameters, and thus the solution of $(\mathsf{HW}, \mathbb{F}_2)$-$\mathsf{LPN}(N-t, k-t, t)$ is always that of $(\mathsf{RHW}, \mathbb{F}_2)$-$\mathsf{LPN}(N, k, t)$.

Let $m = \lfloor N/t \rfloor$. Given a $(\mathsf{RHW}, \mathbb{F}_2)$-$\mathsf{LPN}(N, k, t)$ instance $(\mathbf{A}, \boldsymbol{b})$ with $\boldsymbol{b} = \mathbf{A} \cdot \boldsymbol{s} + \boldsymbol{e} \in \mathbb{F}_2^N$ and $\boldsymbol{s} \in \mathbb{F}_2^k$, we define

$$\mathbf{A} \stackrel{\text{def}}{=} \begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_t \end{bmatrix}, \ \boldsymbol{e} \stackrel{\text{def}}{=} \begin{bmatrix} \boldsymbol{e}_1 \\ \vdots \\ \boldsymbol{e}_t \end{bmatrix} \text{ and } \boldsymbol{b} \stackrel{\text{def}}{=} \begin{bmatrix} \boldsymbol{b}_1 = \mathbf{A}_1 \cdot \boldsymbol{s} + \boldsymbol{e}_1 \\ \vdots \\ \boldsymbol{b}_t = \mathbf{A}_t \cdot \boldsymbol{s} + \boldsymbol{e}_t \end{bmatrix},$$

where $\mathbf{A}_i \in \mathbb{F}_2^{m \times k}$, $\boldsymbol{e}_i \in \mathbb{F}_2^m$ and $\boldsymbol{b}_i \in \mathbb{F}_2^m$ for $i \in [1, t]$. Note that the Hamming weight of each sub-vector $\boldsymbol{e}_i$ is exactly 1. We use $\mathbf{A}_i[j]$ to denote the j-th row vector of $\mathbf{A}_i$, and recall that $\boldsymbol{b}_i[j]$ and $\boldsymbol{e}_i[j]$ is the j-th component of vectors $\boldsymbol{b}_i$ and $\boldsymbol{e}_i$ respectively. Then, for each $i \in [1, t]$, we can obtain the following equation:

$$\sum_{j=1}^{m} \boldsymbol{b}_i[j] = \sum_{j=1}^{m} \mathbf{A}_i[j] \cdot \boldsymbol{s} + \sum_{j=1}^{m} \boldsymbol{e}_i[j] = \left(\sum_{j=1}^{m} \mathbf{A}_i[j] \right) \cdot \boldsymbol{s} + 1 \ .$$

Therefore, we extract t linear relations about the secret and reduce the dimension of $\boldsymbol{s}$ by t. Specifically, we replace $\boldsymbol{s}[0], \ldots, \boldsymbol{s}[t-1]$ with a linear function of other components in $\boldsymbol{s}$, allowing us to eliminate $\boldsymbol{s}[0], \ldots, \boldsymbol{s}[t-1]$ from $\boldsymbol{s}$.

We eliminate the correlation by removing one sample within each block, where correlation indicates that the noise bit of the removed sample is fully determined by the remaining $m-1$ samples in the same block. After removing the t samples, we show that the remaining samples, permuted randomly, still constitute an LPN instance. For the remaining samples in each block $i \in [1,t]$, we denote by w_i the Hamming weight of the noise sub-vector. Then we have that w_i follows a Bernoulli distribution, i.e., $\Pr[w_i = 1] = 1-1/m$ and $\Pr[w_i = 0] = 1/m$. By a union bound, we have that the resulting noise vector follows the exact noise distribution $\mathsf{HW}_{t,N-t}(\mathbb{F}_2)$, with probability at least $(1-1/m)^t \geq 1-t/m$, which is close to 1 as $m = \lfloor N/t \rfloor$ is sufficiently large for the LPN parameters used in the PCG setting. Thus, the resulting LPN instance is an exact-LPN instance $(\mathsf{HW}, \mathbb{F}_2)$-$\mathsf{LPN}(N-t, k-t, t)$. Therefore, we can use the bit security of an exact-LPN instance $(\mathsf{HW}, \mathbb{F}_2)$-$\mathsf{LPN}(N-t, k-t, t)$, based on all known attacks against exact-LPN, to estimate that of a regular-LPN instance $(\mathsf{RHW}, \mathbb{F}_2)$-$\mathsf{LPN}(N, k, t)$. We can convert a dual-LPN problem into an LPN problem using the approach in [62]. Thus, we are also able to perform the above transformation for dual-LPN problems over $\mathbb{F}_2$.

For LPN problems over a field $\mathbb{F}$ with $|\mathbb{F}| > 2$, the above transformation fails to work. For each noisy coordinate, a regular-LPN instance now samples a random element in $\mathbb{F}\backslash\{0\}$ rather than only 1. In this case, for each block $i \in [1,t]$, we have that $\sum_{j=1}^{m} \boldsymbol{b}_i[j] = (\sum_{j=1}^{m} \mathbf{A}_i[j]) \cdot \boldsymbol{s} + r$ where $r \in \mathbb{F}\backslash\{0\}$ is random and unknown. Now, we have to guess the random element r, which succeeds with probability at most $\frac{1}{|\mathbb{F}|-1}$. For all t blocks, we can succeed in guessing all random elements in t noisy coordinates with probability at most $\frac{1}{(|\mathbb{F}|-1)^t} \leq \frac{1}{2^t}$. Besides, we are able to perform the above transformation for a part of blocks. However, it does not allow us to decrease the cost of solving a regular-LPN problem by guessing the random elements located in noisy coordinates and performing the above transformation. In conclusion, we choose to use the known attacks of Pooled Gauss, SD and ISD against exact-LPN to estimate the cost of regular-LPN against these attacks for the case of larger fields.

The Recent Algebraic Attack Against Regular-LPN. Recently, Briaud and Øygarden [22] introduced a new algebraic attack that is tailored to LPN problems with regular noise distributions. Specifically, their attack solves a polynomial system involving the coordinates of a regular noise vector $\boldsymbol{e}$, leveraging the quadratic system that captures the regular structure. This algebraic attack, as described in [22], converts solving a dual-LPN problem over a field $\mathbb{F}$ into solving a polynomial system of degree 2 involving the coordinates of an error vector. In particular, the polynomial system consists of n parity-check equations (represented as $\mathbf{H} \cdot \boldsymbol{e} = \boldsymbol{y}$) along with another quadratic system that encodes the regular structure of a noise vector $\boldsymbol{e} = (\boldsymbol{e}_1, \ldots, \boldsymbol{e}_t)$ where $\boldsymbol{e}_i$ is defined as above. In more detail, for each sub-vector $\boldsymbol{e}_i \in \mathbb{F}^m$ with $m = \lfloor N/t \rfloor$, all quadratic equations of the form $\boldsymbol{e}_i[j_1] \cdot \boldsymbol{e}_i[j_2] = 0$ for $j_1 < j_2$ are involved. For the case of $\mathbb{F}_2$, a variation of the quadratic system is employed by introducing *additional* structural equations of the form $(\boldsymbol{e}_i[j])^2 = \boldsymbol{e}_i[j]$ and $\sum_{j=1}^{m} \boldsymbol{e}_i[j] = 1$, which guarantees that every $\boldsymbol{e}_i$ is a unit vector. Standard algorithms such as XL/Gröbner

bases [12,26,72,79] are then applied to solve the degree-2 polynomial system. Furthermore, a hybrid approach is proposed to reduce the computation complexity. This approach involves guessing some error-free positions of the noise error $\boldsymbol{e}$, inspired from the regular version of Prange's algorithm [47]. It is not easy to give a succinct formula to compute the cost of their algebraic attack. Instead, we choose to provide an estimator tool (see Footnote 3), which allows us to automatically estimate the cost of the algebraic attack.

Compared to linear attacks such as Pooled Gauss, SD and ISD attacks, their algebraic attack achieves lower cost when solving regular-LPN problems with small code rate for some parameter sets (see Table 2 and the full version of the paper [58, Table 8]). The algebraic attack does not outperform ISD attacks for dual-LPN problems used in PCG-like protocols that have constant code rate (i.e., 1/2 or 3/4). Given the number of samples (corresponding to the number of PCG correlations), we are able to increase the dimension k and keep the noise weight t unchanged to resist the algebraic attack [22] against LPN problems, while keeping the efficiency essentially unchanged due to the usage of bootstrapping iterations [82].

Acknowledgements. Work of Yu Yu is supported by the National Key Research and Development Program of China (Grant No. 2020YFA0309705) and the National Natural Science Foundation of China (Grant Nos. 62125204 and 61872236). Yu Yu's work has also been supported by the New Cornerstone Science Foundation through the XPLORER PRIZE. Work of Kang Yang is supported by the National Natural Science Foundation of China (Grant Nos. 62102037 and 61932019). Work of Xiao Wang is supported in part by DARPA under Contract No. HR001120C0087, NSF awards #2016240 and #2236819. The views, opinions, and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

References

1. Abram, D., Scholl, P.: Low-communication multiparty triple generation for SPDZ from ring-LPN. In: Hanaoka, G., Shikata, J., Watanabe, Y. (eds.) PKC 2022, Part I. LNCS, vol. 13177, pp. 221–251. Springer, Heidelberg (2022). https://doi.org/10.1007/978-3-030-97121-2_9
2. Akavia, A.: Learning noisy characters, multiplication codes, and cryptographic hardcore predicates. Ph.D. thesis, Massachusetts Institute of Technology (2008). https://people.csail.mit.edu/akavia/AkaviaPhDThesis.pdf
3. Al Jabri, A.: A statistical decoding algorithm for general linear block codes. In: Honary, B. (ed.) 8th IMA International Conference on Cryptography and Coding. LNCS, vol. 2260, pp. 1–8. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45325-3_1
4. Alekhnovich, M.: More on average case vs approximation complexity. In: 44th FOCS, pp. 298–307. IEEE Computer Society Press (2003). https://doi.org/10.1109/SFCS.2003.1238204
5. Applebaum, B.: Garbling XOR gates "for free" in the standard model. J. Cryptol. **29**(3), 552–576 (2016). https://doi.org/10.1007/s00145-015-9201-9

6. Applebaum, B., Ishai, Y., Kushilevitz, E.: Cryptography with constant input locality. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 92–110. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74143-5_6
7. Augot, D., Finiasz, M., Sendrier, N.: A family of fast syndrome based cryptographic hash functions. In: Dawson, E., Vaudenay, S. (eds.) Mycrypt 2005. LNCS, vol. 3715, pp. 64–83. Springer, Heidelberg (2005). https://doi.org/10.1007/11554868_6
8. Baum, C., Braun, L., Munch-Hansen, A., Razet, B., Scholl, P.: Appenzeller to Brie: efficient zero-knowledge proofs for mixed-mode arithmetic and Z2k. In: Vigna, G., Shi, E. (eds.) ACM CCS 2021, pp. 192–211. ACM Press, November 2021. https://doi.org/10.1145/3460120.3484812
9. Baum, C., Braun, L., Munch-Hansen, A., Scholl, P.: Moz$\mathbb{Z}_2^k$arella: efficient vector-OLE and zero-knowledge proofs over $\mathbb{Z}_2^k$. In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022, Part IV. LNCS, vol. 13510, pp. 329–358. Springer, Heidelberg (2022). https://doi.org/10.1007/978-3-031-15985-5_12
10. Baum, C., Malozemoff, A.J., Rosen, M.B., Scholl, P.: Mac'n'Cheese: zero-knowledge proofs for boolean and arithmetic circuits with nested disjunctions. In: Malkin, T., Peikert, C. (eds.) CRYPTO 2021, Part IV. LNCS, vol. 12828, pp. 92–122. Springer, Heidelberg, Virtual Event (2021). https://doi.org/10.1007/978-3-030-84259-8_4
11. Becker, A., Joux, A., May, A., Meurer, A.: Decoding random binary linear codes in $2^{n/20}$: how $1 + 1 = 0$ improves information set decoding. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 520–536. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_31
12. Beullens, W.: Improved cryptanalysis of UOV and Rainbow. In: Canteaut, A., Standaert, F.X. (eds.) EUROCRYPT 2021, Part I. LNCS, vol. 12696, pp. 348–373. Springer, Heidelberg (2021). https://doi.org/10.1007/978-3-030-77870-5_13
13. Blum, A., Furst, M.L., Kearns, M.J., Lipton, R.J.: Cryptographic primitives based on hard learning problems. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 278–291. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-48329-2_24
14. Bombar, M., Couteau, G., Couvreur, A., Ducros, C.: Correlated pseudorandomness from the hardness of quasi-abelian decoding. In: Handschuh, H., Lysyanskaya, A. (eds.) CRYPTO 2023, Part IV, pp. 567–601. LNCS, Springer, Heidelberg (2023). https://doi.org/10.1007/978-3-031-38551-3_18
15. Boyle, E., Couteau, G., Gilboa, N., Ishai, Y.: Compressing vector OLE. In: Lie, D., Mannan, M., Backes, M., Wang, X. (eds.) ACM CCS 2018, pp. 896–912. ACM Press, October 2018. https://doi.org/10.1145/3243734.3243868
16. Boyle, E., et al.: Correlated pseudorandomness from expand-accumulate codes. In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022, Part II. LNCS, vol. 13508, pp. 603–633. Springer, Heidelberg (2022). https://doi.org/10.1007/978-3-031-15979-4_21
17. Boyle, E., et al.: Efficient two-round OT extension and silent non-interactive secure computation. In: Cavallaro, L., Kinder, J., Wang, X., Katz, J. (eds.) ACM CCS 2019, pp. 291–308. ACM Press, November 2019. https://doi.org/10.1145/3319535.3354255
18. Boyle, E., Couteau, G., Gilboa, N., Ishai, Y., Kohl, L., Scholl, P.: Efficient pseudorandom correlation generators: silent OT extension and more. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part III. LNCS, vol. 11694, pp. 489–518. Springer, Heidelberg (2019). https://doi.org/10.1007/978-3-030-26954-8_16

19. Boyle, E., Couteau, G., Gilboa, N., Ishai, Y., Kohl, L., Scholl, P.: Efficient pseudorandom correlation generators from ring-LPN. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part II. LNCS, vol. 12171, pp. 387–416. Springer, Heidelberg (2020). https://doi.org/10.1007/978-3-030-56880-1_14
20. Boyle, E., Gilboa, N., Ishai, Y.: Function secret sharing. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part II. LNCS, vol. 9057, pp. 337–367. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46803-6_12
21. Brakerski, Z., Lyubashevsky, V., Vaikuntanathan, V., Wichs, D.: Worst-case hardness for LPN and cryptographic hashing via code smoothing. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019, Part III. LNCS, vol. 11478, pp. 619–635. Springer, Heidelberg (2019). https://doi.org/10.1007/978-3-030-17659-4_21
22. Briaud, P., Øygarden, M.: A new algebraic approach to the regular syndrome decoding problem and implications for PCG constructions. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part V. LNCS, vol. 14008, pp. 391–422. Springer, Heidelberg (2023). https://doi.org/10.1007/978-3-031-30589-4_14
23. Bui, D., Couteau, G.: Improved private set intersection for sets with small entries. In: Boldyreva, A., Kolesnikov, V. (eds.) PKC 2023, Part II. LNCS, vol. 13941, pp. 190–220. Springer, Heidelberg (2023). https://doi.org/10.1007/978-3-031-31371-4_7
24. Carozza, E., Couteau, G., Joux, A.: Short signatures from regular syndrome decoding in the head. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part V. LNCS, vol. 14008, pp. 532–563. Springer, Heidelberg (2023). https://doi.org/10.1007/978-3-031-30589-4_19
25. Carrier, K., Debris-Alazard, T., Meyer-Hilfiger, C., Tillich, J.P.: Statistical decoding 2.0: reducing decoding to LPN. In: Agrawal, S., Lin, D. (eds.) ASIACRYPT 2022, Part IV. LNCS, vol. 13794, pp. 477–507. Springer, Heidelberg (2022). https://doi.org/10.1007/978-3-031-22972-5_17
26. Coppersmith, D.: Solving homogeneous linear equations over GF(2) via block Wiedemann algorithm. Math. Comput. **62**(205), 333–350 (1994)
27. Couteau, G., Rindal, P., Raghuraman, S.: Silver: silent VOLE and oblivious transfer from hardness of decoding structured LDPC codes. In: Malkin, T., Peikert, C. (eds.) CRYPTO 2021, Part III. LNCS, vol. 12827, pp. 502–534. Springer, Heidelberg, Virtual Event (2021). https://doi.org/10.1007/978-3-030-84252-9_17
28. Cramer, R., Damgård, I., Escudero, D., Scholl, P., Xing, C.: SPD $\mathbb{Z}_2^k$: efficient MPC mod 2^k for dishonest majority. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part II. LNCS, vol. 10992, pp. 769–798. Springer, Heidelberg (2018). https://doi.org/10.1007/978-3-319-96881-0_26
29. Cui, H., Wang, X., Yang, K., Yu, Y.: Actively secure half-gates with minimum overhead under duplex networks. In: Hazay, C., Stam, M. (eds.) EUROCRYPT 2023, Part II. LNCS, vol. 14005, pp. 35–67. Springer, Heidelberg (2023). https://doi.org/10.1007/978-3-031-30617-4_2
30. Damgård, I., Escudero, D., Frederiksen, T.K., Keller, M., Scholl, P., Volgushev, N.: New primitives for actively-secure MPC over rings with applications to private machine learning. In: 2019 IEEE Symposium on Security and Privacy, pp. 1102–1120. IEEE Computer Society Press, May 2019. https://doi.org/10.1109/SP.2019.00078
31. Damgård, I., Pastro, V., Smart, N.P., Zakarias, S.: Multiparty computation from somewhat homomorphic encryption. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 643–662. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32009-5_38

32. David, B., Dowsley, R., Nascimento, A.C.A.: Universally composable oblivious transfer based on a variant of LPN. In: Gritzalis, D., Kiayias, A., Askoxylakis, I.G. (eds.) CANS 2014. LNCS, vol. 8813, pp. 143–158. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-319-12280-9_10
33. Debris-Alazard, T., Tillich, J.: Statistical decoding. In: ISIT 2017 (2017). https://doi.org/10.1109/ISIT.2017.8006839
34. Dittmer, S., Ishai, Y., Lu, S., Ostrovsky, R.: Authenticated garbling from simple correlations. In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022, Part IV. LNCS, vol. 13510, pp. 57–87. Springer, Heidelberg (2022). https://doi.org/10.1007/978-3-031-15985-5_3
35. Dittmer, S., Ishai, Y., Lu, S., Ostrovsky, R.: Improving line-point zero knowledge: two multiplications for the price of one. In: Yin, H., Stavrou, A., Cremers, C., Shi, E. (eds.) ACM CCS 2022, pp. 829–841. ACM Press, November 2022. https://doi.org/10.1145/3548606.3559385
36. Dittmer, S., Ishai, Y., Ostrovsky, R.: Line-point zero knowledge and its applications. In: 2nd Conference on Information-Theoretic Cryptography (2021). https://doi.org/10.4230/LIPICS.ITC.2021.5
37. Dodis, Y., Kalai, Y.T., Lovett, S.: On cryptography with auxiliary input. In: Mitzenmacher, M. (ed.) 41st ACM STOC, pp. 621–630. ACM Press, May/June 2009. https://doi.org/10.1145/1536414.1536498
38. Dumer, I.: On minimum distance decoding of linear codes. In: Proceedings of 5th Joint Soviet-Swedish International Workshop Information Theory (1991)
39. Esser, A., Bellini, E.: Syndrome decoding estimator. In: Hanaoka, G., Shikata, J., Watanabe, Y. (eds.) PKC 2022, Part I. LNCS, vol. 13177, pp. 112–141. Springer, Heidelberg (2022). https://doi.org/10.1007/978-3-030-97121-2_5
40. Esser, A., Kübler, R., May, A.: LPN decoded. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part II. LNCS, vol. 10402, pp. 486–514. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-319-63715-0_17
41. Esser, A., May, A., Zweydinger, F.: McEliece needs a break - solving McEliece-1284 and quasi-cyclic-2918 with modern ISD. In: Dunkelman, O., Dziembowski, S. (eds.) EUROCRYPT 2022, Part III. LNCS, vol. 13277, pp. 433–457. Springer, Heidelberg, May/June 2022. https://doi.org/10.1007/978-3-031-07082-2_16
42. Feneuil, T., Joux, A., Rivain, M.: Syndrome decoding in the head: shorter signatures from zero-knowledge proofs. In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022, Part II. LNCS, vol. 13508, pp. 541–572. Springer, Heidelberg (2022). https://doi.org/10.1007/978-3-031-15979-4_19
43. Finiasz, M., Sendrier, N.: Security bounds for the design of code-based cryptosystems. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 88–105. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-10366-7_6
44. Fischer, J.B., Stern, J.: An efficient pseudo-random generator provably as secure as syndrome decoding. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 245–255. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-68339-9_22
45. Franzese, N., Katz, J., Lu, S., Ostrovsky, R., Wang, X., Weng, C.: Constant-overhead zero-knowledge for RAM programs. In: Vigna, G., Shi, E. (eds.) ACM CCS 2021, pp. 178–191. ACM Press, November 2021. https://doi.org/10.1145/3460120.3484800
46. Hamdaoui, Y., Sendrier, N.: A non asymptotic analysis of information set decoding. Cryptology ePrint Archive, Report 2013/162 (2013). https://eprint.iacr.org/2013/162

47. Hazay, C., Orsini, E., Scholl, P., Soria-Vazquez, E.: TinyKeys: a new approach to efficient multi-party computation. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part III. LNCS, vol. 10993, pp. 3–33. Springer, Heidelberg (2018). https://doi.org/10.1007/978-3-319-96878-0_1
48. Hazay, C., Scholl, P., Soria-Vazquez, E.: Low cost constant round MPC combining BMR and oblivious transfer. J. Cryptol. **33**(4), 1732–1786 (2020). https://doi.org/10.1007/s00145-020-09355-y
49. Hopper, N.J., Blum, M.: Secure human identification protocols. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 52–66. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45682-1_4
50. Hou, X., et al.: CipherGPT: secure two-party GPT inference. Cryptology ePrint Archive, Paper 2023/1147 (2023). https://eprint.iacr.org/2023/1147
51. Huang, Z., Lu, W.J., Hong, C., Ding, J.: Cheetah: lean and fast secure two-party deep neural network inference. In: Butler, K.R.B., Thomas, K. (eds.) USENIX Security 2022, pp. 809–826. USENIX Association, August 2022
52. Jain, A., Lin, H., Sahai, A.: Indistinguishability obfuscation from well-founded assumptions. In: Khuller, S., Williams, V.V. (eds.) 53rd ACM STOC, pp. 60–73. ACM Press, June 2021. https://doi.org/10.1145/3406325.3451093
53. Jain, A., Krenn, S., Pietrzak, K., Tentes, A.: Commitments and efficient zero-knowledge proofs from learning parity with noise. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 663–680. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34961-4_40
54. Katz, J., Shin, J.S., Smith, A.: Parallel and concurrent security of the HB and HB+ protocols. J. Cryptol. **23**(3), 402–421 (2010). https://doi.org/10.1007/s00145-010-9061-2
55. Keller, M., Orsini, E., Scholl, P.: Actively secure OT extension with optimal overhead. In: Gennaro, R., Robshaw, M.J.B. (eds.) CRYPTO 2015, Part I. LNCS, vol. 9215, pp. 724–741. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-47989-6_35
56. Keller, M., Orsini, E., Scholl, P.: MASCOT: faster malicious arithmetic secure computation with oblivious transfer. In: Weippl, E.R., Katzenbeisser, S., Kruegel, C., Myers, A.C., Halevi, S. (eds.) ACM CCS 2016, pp. 830–842. ACM Press, October 2016. https://doi.org/10.1145/2976749.2978357
57. Lin, F., Xing, C., Yao, Y.: More efficient zero-knowledge protocols over $\mathbb{Z}\_2^k$ via galois rings. Cryptology ePrint Archive, Report 2023/150 (2023). https://eprint.iacr.org/2023/150
58. Liu, H., Wang, X., Yang, K., Yu, Y.: The hardness of LPN over any integer ring and field for PCG applications. Cryptology ePrint Archive, Report 2022/712 (2022). https://eprint.iacr.org/2022/712
59. May, A., Meurer, A., Thomae, E.: Decoding random linear codes in $\tilde{\mathcal{O}}(2^{0.054n})$. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 107–124. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25385-0_6
60. Melchor, C.A., Blazy, O., Deneuville, J., Gaborit, P., Zémor, G.: Efficient encryption from random quasi-cyclic codes. IEEE Trans. Inf. Theory **64**(5), 3927–3943 (2018). https://doi.org/10.1109/TIT.2018.2804444
61. Meyer-Hilfiger, C., Tillich, J.: Rigorous foundations for dual attacks in coding theory. In: Rothblum, G.N., Wee, H. (eds.) TCC 2023. LNCS, vol. 14372, pp. 3–32. Springer, Heidelberg (2023). https://doi.org/10.1007/978-3-031-48624-1_1

62. Micciancio, D., Mol, P.: Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 465–484. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22792-9_26
63. Misoczki, R., Tillich, J., Sendrier, N., Barreto, P.S.L.M.: MDPC-McEliece: new McEliece variants from moderate density parity-check codes. In: Proceedings of the 2013 IEEE International Symposium on Information Theory, 2013. pp. 2069–2073. IEEE (2013). https://doi.org/10.1109/ISIT.2013.6620590
64. Nielsen, J.B., Nordholt, P.S., Orlandi, C., Burra, S.S.: A new approach to practical active-secure two-party computation. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 681–700. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32009-5_40
65. Prange, E.: The use of information sets in decoding cyclic codes. IRE Trans. Inf. Theory **8**, 5–9 (1962). https://doi.org/10.1109/TIT.1962.1057777
66. Raghuraman, S., Rindal, P.: Blazing fast PSI from improved OKVS and subfield VOLE. In: Yin, H., Stavrou, A., Cremers, C., Shi, E. (eds.) ACM CCS 2022, pp. 2505–2517. ACM Press, November 2022. https://doi.org/10.1145/3548606.3560658
67. Raghuraman, S., Rindal, P., Tanguy, T.: Expand-convolute codes for pseudorandom correlation generators from LPN. In: CRYPTO 2023, Part IV, pp. 602–632. LNCS, Springer, Heidelberg (2023). https://doi.org/10.1007/978-3-031-38551-3_19
68. Rindal, P., Schoppmann, P.: VOLE-PSI: fast OPRF and circuit-PSI from vector-OLE. In: Canteaut, A., Standaert, F.X. (eds.) EUROCRYPT 2021, Part II. LNCS, vol. 12697, pp. 901–930. Springer, Heidelberg (2021). https://doi.org/10.1007/978-3-030-77886-6_31
69. Schoppmann, P., Gascón, A., Reichert, L., Raykova, M.: Distributed vector-OLE: improved constructions and implementation. In: Cavallaro, L., Kinder, J., Wang, X., Katz, J. (eds.) ACM CCS 2019, pp. 1055–1072. ACM Press, November 2019. https://doi.org/10.1145/3319535.3363228
70. Sendrier, N.: Decoding one out of many. In: Yang, B.Y. (ed.) Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011, pp. 51–67. Springer, Heidelberg, November/December 2011. https://doi.org/10.1007/978-3-642-25405-5_4
71. Stern, J.: A method for finding codewords of small weight. In: Coding Theory and Applications, vol. 388 (1988). https://doi.org/10.1007/BFB0019850
72. Thomé, E.: Subquadratic computation of vector generating polynomials and improvement of the block Wiedemann algorithm. J. Symb. Comput. **33**(5), 757–775 (2002). https://doi.org/10.1006/JSCO.2002.0533
73. Torres, R.C., Sendrier, N.: Analysis of information set decoding for a sub-linear error weight. In: Takagi, T. (ed.) Post-Quantum Cryptography - 7th International Workshop, PQCrypto 2016, pp. 144–161. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-319-29360-8_10
74. Wang, X., Ranellucci, S., Katz, J.: Authenticated garbling and efficient maliciously secure two-party computation. In: Thuraisingham, B.M., Evans, D., Malkin, T., Xu, D. (eds.) ACM CCS 2017, pp. 21–37. ACM Press, October/November 2017. https://doi.org/10.1145/3133956.3134053
75. Wang, X., Ranellucci, S., Katz, J.: Global-scale secure multiparty computation. In: Thuraisingham, B.M., Evans, D., Malkin, T., Xu, D. (eds.) ACM CCS 2017, pp. 39–56. ACM Press, October/November 2017. https://doi.org/10.1145/3133956.3133979

76. Weng, C., Yang, K., Katz, J., Wang, X.: Wolverine: fast, scalable, and communication-efficient zero-knowledge proofs for boolean and arithmetic circuits. In: 2021 IEEE Symposium on Security and Privacy, pp. 1074–1091. IEEE Computer Society Press, May 2021. https://doi.org/10.1109/SP40001.2021.00056
77. Weng, C., Yang, K., Xie, X., Katz, J., Wang, X.: Mystique: efficient conversions for zero-knowledge proofs with applications to machine learning. In: Bailey, M., Greenstadt, R. (eds.) USENIX Security 2021, pp. 501–518. USENIX Association, August 2021
78. Weng, C., Yang, K., Yang, Z., Xie, X., Wang, X.: AntMan: interactive zero-knowledge proofs with sublinear communication. In: Yin, H., Stavrou, A., Cremers, C., Shi, E. (eds.) ACM CCS 2022, pp. 2901–2914. ACM Press, November 2022. https://doi.org/10.1145/3548606.3560667
79. Wiedemann, D.H.: Solving sparse linear equations over finite fields. IEEE Trans. Inf. Theory **32**(1), 54–62 (1986)
80. Yang, K., Sarkar, P., Weng, C., Wang, X.: QuickSilver: efficient and affordable zero-knowledge proofs for circuits and polynomials over any field. In: Vigna, G., Shi, E. (eds.) ACM CCS 2021, pp. 2986–3001. ACM Press, November 2021. https://doi.org/10.1145/3460120.3484556
81. Yang, K., Wang, X., Zhang, J.: More efficient MPC from improved triple generation and authenticated garbling. In: Ligatti, J., Ou, X., Katz, J., Vigna, G. (eds.) ACM CCS 2020, pp. 1627–1646. ACM Press, November 2020. https://doi.org/10.1145/3372297.3417285
82. Yang, K., Weng, C., Lan, X., Zhang, J., Wang, X.: Ferret: fast extension for correlated OT with small communication. In: Ligatti, J., Ou, X., Katz, J., Vigna, G. (eds.) ACM CCS 2020, pp. 1607–1626. ACM Press, November 2020. https://doi.org/10.1145/3372297.3417276
83. Yu, Y., Steinberger, J.P.: Pseudorandom functions in almost constant depth from low-noise LPN. In: Fischlin, M., Coron, J.S. (eds.) EUROCRYPT 2016, Part II. LNCS, vol. 9666, pp. 154–183. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49896-5_6
84. Yu, Y., Zhang, J., Weng, J., Guo, C., Li, X.: Collision resistant hashing from sub-exponential learning parity with noise. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019, Part II. LNCS, vol. 11922, pp. 3–24. Springer, Heidelberg (2019). https://doi.org/10.1007/978-3-030-34621-8_1
85. Zichron, L.: Locally computable arithmetic pseudorandom generators. Master's thesis, School of Electrical Engineering, Tel Aviv University (2017)