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

Hanlin Liu
Shanghai Qi Zhi Institute
hans1024@sjtu.edu.cn

Xiao Wang
Northwestern University
wangxiao@northwestern.edu

Kang Yang
State Key Laboratory of Cryptology
yangk@sklc.org

Yu Yu
Shanghai Jiao Tong University
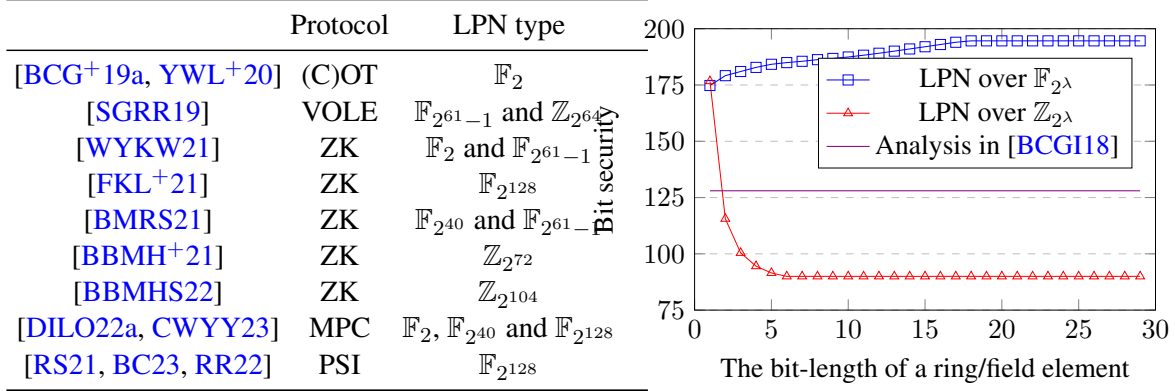Shanghai Qi Zhi Institute
yuyu@yuyu.hk

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 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., [HB01] and follow-up works), public key encryption [Ale03], commitment scheme [JKPT12], garbled circuits [App16], oblivious transfer [DDN14] and collision-resistant hash functions [BLVW19, YZW+19]. All these primitives adopt LPN over binary field $\mathbb{F}_2$ with moderate dimensions.

	Protocol	LPN type
[BCG$^+$19a, YWL$^+$20]	(C)OT	$\mathbb{F}_2$
[SGRR19]	VOLE	$\mathbb{F}_{2^{61}-1}$ and $\mathbb{Z}_{2^{64}}$
[WYKW21]	ZK	$\mathbb{F}_2$ and $\mathbb{F}_{2^{61}-1}$
[FKL$^+$21]	ZK	$\mathbb{F}_{2^{128}}$
[BMRS21]	ZK	$\mathbb{F}_{2^{40}}$ and $\mathbb{F}_{2^{61}-1}$
[BBMH$^+$21]	ZK	$\mathbb{Z}_{2^{72}}$
[BBMHS22]	ZK	$\mathbb{Z}_{2^{104}}$
[DILO22a, CWYY23]	MPC	$\mathbb{F}_2, \mathbb{F}_{2^{40}}$ and $\mathbb{F}_{2^{128}}$
[RS21, BC23, RR22]	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.

Figure 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.

The recent work by Boyle et al. [BCGI18] 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 [BGI15]. The sparsity of a noise 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 [BCG$^+$19b, BCG$^+$19a, SGRR19, YWL$^+$20, BCG$^+$20, WYKW21, CRR21, BCG$^+$22, AS22, BCCD23, RRT23] with sublinear communication for generating random (C)OT or (V)OLE correlations. These PCG-like protocols have gained a lot of interests in designing various concretely efficient protocols, including secure multi-party computation (MPC) (e.g., [DPSZ12, NNOB12, KOS16, WRK17a, WRK17b, HSS20, CDE$^+$18, DEF$^+$19, YWZ20, DILO22a, CWYY23]), zero-knowledge (ZK) proofs (e.g., [WYKW21, BMRS21, DIO21, YSWW21, BBMH$^+$21, DILO22b, BBMHS22, WYY$^+$22]), privacy-preserving machine learning [SGRR19, WYX$^+$21, HjLHD22], private set intersection (PSI) [RS21, RR22, BC23], 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 $\mathbb{F}_2$ [Ale03, FS09, HS13, TS16]. 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. [FJR22] 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; [CCJ23]

---

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

showed an attack specific to regular noises but not for parameters usable in PCG applications; [BØ23] 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. [BCGI18], 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. [FJR22] 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 Section 3), and thus (T, ϵ)-hard [2] LPN under an exact noise 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 [BØ23]. 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 [SGRR19, BBMH+21, BBMHS22], 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 [Aka08], which explored a generalized

---

[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.

LPN			This work					[BCGI18]
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

Table 1: Comparison between our analysis and [BCGI18] 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 [BCGI18].

Dual-LPN			This work					[BCGI18]
n	N	t	$\mathbb{F}_{2^{128}}$	$\mathbb{F}_{2^8}$	$\mathbb{Z}_{2^{128}}$	$\mathbb{Z}_4$	$\mathbb{F}_2$	Any field
2^{10}	2^{12}	44	117 (-0)	109 (-0)	61 (-0)	73 (-0)	97 (-1)	80
2^{12}	2^{14}	39	111 (-0)	103 (-0)	62 (-0)	74 (-0)	95 (-0)	80
2^{14}	2^{16}	34	107 (-0)	99 (-0)	64 (-0)	73 (-0)	93 (-0)	80
2^{16}	2^{18}	32	108 (-0)	100 (-0)	69 (-0)	77 (-0)	95 (-0)	80
2^{18}	2^{20}	31	112 (-0)	104 (-0)	73 (-0)	81 (-0)	99 (-0)	80
2^{20}	2^{22}	30	116 (-0)	108 (-0)	79 (-0)	87 (-0)	103 (-0)	80
2^{22}	2^{24}	29	119 (-0)	112 (-0)	83 (-0)	92 (-0)	107 (-0)	80

Table 2: Comparison between our analysis and [BCGI18] for the bit-security of a dual-LPN (a.k.a., syndrome decoding) problem with dimension $N - n = 3N/4$, number of samples N, Hamming weight of noises t. These parameters are from [BCGI18].

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 Figure 2. On the concrete side, our analysis (in Figure 1b and in Tables 1 and 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 [CRR21] 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 Figure 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 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.
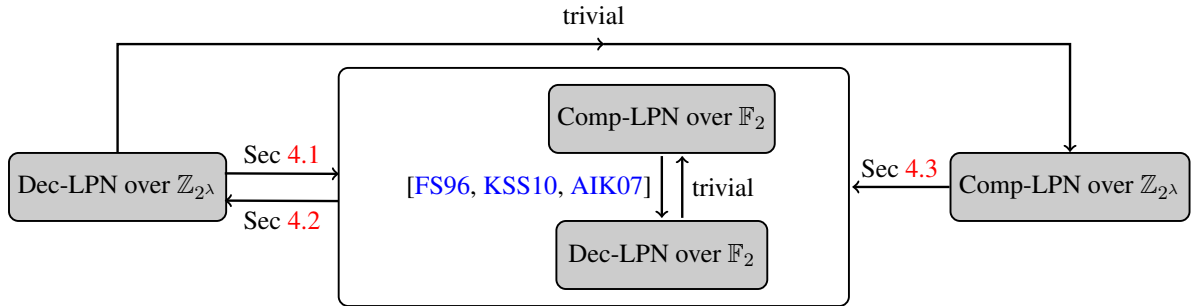

Figure 2: The reduction relations between computational and decisional versions of LPN over $\mathbb{F}_2$ and $\mathbb{Z}_{2^\lambda}$ in the presence of Bernoulli and exact noise distribution.

Concrete security of LPN for PCG. Finally, we provide an estimator tool [3] that incorporates all existing attacks on the concrete security of LPN problems related to the PCG setting. Prior to this work, most PCG-like protocols use the analysis from [BCGI18] for all LPN variants. We refined their analysis and incorporated attacks on integer rings and regular noises. See Table 1 and Table 2 for some representative parameters originally proposed in [BCGI18].

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. [CDHT22]. We formalize this observation by showing that both the optimal SD and SD 2.0 attacks (adapted to the low-noise setting)

---

[3]This tool can be found at `https://gist.github.com/hansliu1024/21c87609e75f6cc52decdc69981e1d5b`.

require more cost, compared to the Prange's original ISD algorithm [Pra62] 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 [CDHT22]. 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., [HLL$^+$23]) to choose LPN parameters for PCG-like protocols. Our attack on integer rings has subsequently been noted by multiple works. Baum et al. [BBMHS22] 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 $\mathbb{Z}_{2^\lambda}$ with the countermeasure. Besides, the updated version by Boyle et al. [BCG$^+$20] and the work by Lin et al. [LXY23] 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 $e = \sum_{i \in [\lambda]} 2^i \cdot e_i$ with independent and random weight-t noises 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 a for column vectors, and bold upper-case letters like $\mathbf{A}$ for matrices. By slightly abusing the notation, for a vector a, we use $|a|$ to denote the Hamming weight of a, and denote by $a[i]$ the i-th component of a. For two vectors x, y, we denote by $\langle x, y \rangle$ the inner product of x and y. For a vector $a \in (\mathbb{Z}_{2^\lambda})^k$, we use $\mathsf{BitDecomp}(a)$ to denote the bit-decomposition of a, and its output is denoted by $(a^0, a^1, \cdots, a^{\lambda-1})$ such that $a^i \in \mathbb{F}_2^k$ for $i \in [\lambda]$ and $(a^0[j], a^1[j], \ldots, a^{\lambda-1}[j])$ is the bit-decomposition of ring element $a[j] \in \mathbb{Z}_{2^\lambda}$ for $j \in [k]$. Let $\mathsf{BitDecomp}^{-1}(a^0, a^1, \cdots, a^{\lambda-1}) = \sum_{i=0}^{\lambda-1} 2^i \cdot a^i \in (\mathbb{Z}_{2^\lambda})^k$ be the inverse of $\mathsf{BitDecomp}(a)$. 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., [YS16]). *For any $\mu \in (0, 1)$, if each coordinate of a vector $v \in \mathbb{F}_2^t$ is independently set to 1 with probability μ, then the probability that $|v| = \lceil \mu t \rceil$ is at least $\Omega(1/\sqrt{t})$.*

2.2 Learning Parity with Noise

Recently, variants of the learning parity with noise (LPN) assumption [BFKL94] 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 [BCG$^+$19b, BCG$^+$19a, YWL$^+$20, CRR21, BCG$^+$22, RRT23], 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 [BCGI18, BCG$^+$19b, BCG$^+$19a, SGRR19, WYKW21, CRR21, BCG$^+$22, RRT23] and the OLE protocols [BCG$^+$19b, BCG$^+$20, AS22, BCCD23].

- Case 3 that $\mathcal{R} = \mathbb{Z}_{2^\lambda}$ (e.g., $\lambda \in \{32, 64, 128\}$) is used to obtain the VOLE protocols [SGRR19, BBMH$^+$21, BBMHS22, LXY23].

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., [BCG$^+$19b, BCG$^+$19a]), 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., [DKL09, BCGI18, CRR21, JLS21]), 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., [AFS05, HOSS18, BCGI18, BCG$^+$19b, BCG$^+$19a, YWL$^+$20, WYKW21, BCG$^+$22, BCCD23, CCJ23], 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 [Ale03], quasi-cyclic codes [AMBD$^+$18], MDPC codes [MTSB13], expand-accumulate codes [BCG$^+$22] etc. We do not analyze the hardness of LPN

problems based on quasi-cyclic codes, which needs to take into account the effect of DOOM attack [Sen11] 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 $(\mathcal{D}, \mathbf{C}, \mathcal{R})$-LPN$(N, k, t)$ problem, and only write $(\mathcal{D}, \mathcal{R})$-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 $(\mathcal{D}, \mathbf{C}^\perp, \mathcal{R})$-dual-LPN$(N, n, t)$ problem is (T, ϵ)-hard if for every probabilistic distinguisher $\mathcal{B}$ running in time T:

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

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

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

where $\mathbf{H}, \mathbf{e}$ 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 $\mathbf{H} \cdot (\mathbf{A} \cdot \mathbf{s} + \mathbf{e}) = (\mathbf{H} \cdot \mathbf{A}) \cdot \mathbf{s} + \mathbf{H} \cdot \mathbf{e} = \mathbf{H} \cdot \mathbf{e}$, 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 [MM11, Lemma 4.9].

3 The Hardness of LPN with Regular Noise Distributions

A series of MPC and ZK protocols (e.g., [AFS05, HOSS18, BCGI18, BCG+19b, BCG+19a, YWL+20, WYKW21, DIO21, BMRS21, YSWW21, BCG+22, BBMH+21, DILO22b, BBMHS22, WYY+22, BCCD23, CCJ23]) rely on the hardness of LPN problems with regular noise distributions. Multiple prior works, e.g., [BCGI18, BCG+19b, BCG+19a, YWL+20, BCG+22, CCJ23], 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 [FJR22]. 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]

---

[4] In particular, [FJR22] 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.

Theorem 1 (Theorem 1 of [FJR22], adapted). *If an exact-LPN problem* $(\mathsf{HW}, \mathbb{F})\text{-}\mathsf{LPN}(N, k, t)$ *is* (T, ϵ)*-hard, the regular-LPN problem* $(\mathsf{RHW}, \mathbb{F})\text{-}\mathsf{LPN}(N, k, t)$ *is*

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

The statement also holds for dual-LPN.

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

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

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 $\boldsymbol{e}_2 = \boldsymbol{0}$ and computes $\boldsymbol{e}_1 = \mathbf{H}_1^{-1} \cdot \boldsymbol{y}$ (without loss of generality, assuming that $\mathbf{H}_1$ is invertible), which succeeds with probability

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

For example, if $N \leq 2n$, the term $\epsilon > e^{-t}$ cannot afford when multiplying it with p_t. For larger time T, it only becomes worse due to the known optimizations and trade-offs between T and ϵ (see, e.g., [TS16]). The case for (primal-)LPN is likewise.

Prior work [FJR22] 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 [FJR22, 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 [FJR22]), 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 $(\mathsf{HW}, \mathbb{F})\text{-LPN}(N, k, t)$ 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., [KOS15, YS16]), pick any full-rank matrix $\mathbf{B} \in \mathbb{F}^{k \times (k + \alpha t)}$ such that $\mathbf{M}$ defined below has full rank

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

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:

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

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

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 $(\mathbf{C}_i, \boldsymbol{b}_i')$ 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, $(\boldsymbol{r}_1^{\mathsf{T}} \cdot \boldsymbol{x}, \ldots, \boldsymbol{r}_{\alpha t}^{\mathsf{T}} \cdot \boldsymbol{x})$ 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, $(\mathbf{C}_i, \boldsymbol{b}_i')$ 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

---

[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.

same weight-1 noise $(\boldsymbol{e}_i^\mathsf{T}, u_i - \mathbf{1}^\mathsf{T} \cdot \boldsymbol{e}_i)$ as $(\mathbf{C}_i, \boldsymbol{b}_i')$.

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

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

$$[\mathbf{A}_i \| \mathbf{P}_i] \cdot \mathbf{M} \cdot \boldsymbol{x} + \boldsymbol{e}_i = (\mathbf{A}_i \cdot \mathbf{B} \cdot \boldsymbol{x} + \boldsymbol{e}_i) + \mathbf{P}_i \cdot \begin{bmatrix} \boldsymbol{r}_1^\mathsf{T} \cdot \boldsymbol{x} \\ \vdots \\ \boldsymbol{r}_{\alpha t}^\mathsf{T} \cdot \boldsymbol{x} \end{bmatrix} ,$$

$$= \boldsymbol{b}_i + \mathbf{P}_i \cdot \begin{bmatrix} \mathbf{1}^\mathsf{T} \cdot \boldsymbol{b}_1 + v_1 - 1 \\ \vdots \\ \mathbf{1}^\mathsf{T} \cdot \boldsymbol{b}_{\alpha t} + v_{\alpha t} - 1 \end{bmatrix} .$$

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 $(\mathsf{RHW}, \mathbb{F})\text{-LPN}(N + \alpha t, k + \alpha t, \alpha t)$. Therefore, we just feed all αt blocks as per (1) to the solver against $(\mathsf{RHW}, \mathbb{F})\text{-LPN}(N + \alpha t, k + \alpha t, \alpha t)$. If it returns $\boldsymbol{x}$, then we recover the secret vector $\boldsymbol{s} := \mathbf{B} \cdot \boldsymbol{x}$ of the exact-LPN instance $(\mathsf{HW}, \mathbb{F})\text{-LPN}(N, k, t)$. Quantitatively, if one breaks $(\mathsf{RHW}, \mathbb{F})\text{-LPN}(N + \alpha t, k + \alpha t, \alpha t)$ with probability p, then it can also break $(\mathsf{HW}, \mathbb{F})\text{-LPN}(N, k, t)$ 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.,

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

which is always possible by letting $\boldsymbol{x}' \overset{\text{def}}{=} \boldsymbol{x} + \mathbf{M}^{-1} \cdot \boldsymbol{h}$ 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 $\mathbb{F}^{m \times n}$.

Proof. Let $\mathcal{E}_{ind}$ be the event that $\boldsymbol{r}_1^\mathsf{T}, \ldots, \boldsymbol{r}_{\alpha t}^\mathsf{T}$ are linearly independent. Following previous work (e.g., [KOS15, YS16]), we have that $\Pr[\mathcal{E}_{ind}] > 1 - |\mathbb{F}|^{-k}$. Conditioned on $\mathcal{E}_{ind}$, the square matrix $\mathbf{M}$ is invertible. Together with the uniformity of $(\mathbf{A}_i, \mathbf{P}_i)$ for $i \in [1, \alpha t]$, we obtain that $[\mathbf{A}_1 \| \mathbf{P}_1] \cdot \mathbf{M}, \ldots, [\mathbf{A}_{\alpha t} \| \mathbf{P}_{\alpha t}] \cdot \mathbf{M}$ are identically distributed to $(U_\mathbb{F}^{m \times (k+\alpha t)})^{\alpha t}$ and are independent of $\boldsymbol{r}_1^\mathsf{T}, \ldots, \boldsymbol{r}_{\alpha t}^\mathsf{T}$ (despite that $\boldsymbol{r}_i^\mathsf{T}$ is a part of $\mathbf{M}$). Furthermore, we have

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

which implies

$$\mathsf{SD}\left(\left(\begin{bmatrix}[\mathbf{A}_i\|\mathbf{P}_i]\cdot\mathbf{M}\\ \boldsymbol{r}_i^\mathsf{T}-\mathbf{1}^\mathsf{T}(\mathbf{A}_i\mathbf{B})\end{bmatrix}\right)_{i\in[1,\alpha t]}\;,\;\left(\begin{bmatrix}[\mathbf{A}_i\|\mathbf{P}_i]\cdot\mathbf{M}\\ U_\mathbb{F}^{1\times(k+\alpha t)}\end{bmatrix}\right)_{i\in[1,\alpha t]}\right)\le|\mathbb{F}|^{-k}\;.$$

Combining with the fact that $\left([\mathbf{A}_i\|\mathbf{P}_i]\cdot\mathbf{M}\right)_{i\in[1,\alpha t]}$ is $|\mathbb{F}|^{-k}$-close to $(U_\mathbb{F}^{m\times(k+\alpha t)})^{\alpha t}$, we complete the proof by a triangle inequality. $\qquad\square$

We are able to obtain a similar result for dual-LPN in the following Corollary 1 via the reductions between LPN and dual-LPN (see Section 2.2).

Corollary 1. *Let* $t,N\in\mathbb{N}$ *and* $\alpha\ge2$ *such that* $\alpha t\in\mathbb{N}$ *and* $(\alpha t)|N$. *If the exact-dual-LPN problem* $(\mathsf{HW},\mathbb{F})$-dual-LPN(N,n,t) *is* (T,ϵ)-*hard, then the regular-dual-LPN problem* $(\mathsf{RHW},\mathbb{F})$-dual-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 $|\boldsymbol{e}_i^\mathsf{T}|\le1$) 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 [SGRR19, BBMH+21, BBMHS22, LXY23], where these VOLE protocols could also benefit other works that need VOLE over integer rings like the MPC protocol SPD$\mathbb{Z}_{2^k}$ [CDE+18, DEF+19]. 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 $|\mathbb{F}|\approx2^\lambda$ [BCGI18]. 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 Figure 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 $\mathbb{Z}_{2^\lambda}\to$ 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 $\mathbb{F}_2\to$ 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 $\approx2^\lambda\cdot t$.

- **Computational LPN over $\mathbb{Z}_{2^\lambda}\to$ 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 Appendix A, which can be further generalized to any ring $\mathbb{Z}_N$ for an integer N. All these reductions focus 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). In particular, we have the following lemma.

Lemma 3 (Unique decoding of LPN over any finite field $\mathbb{F}$). *For any $N > k + 4t \log N$, the following probability is bounded by $\frac{N^{2t}}{|\mathbb{F}|^{N-k-2t}}$.*

$$\Pr_{\mathbf{A} \leftarrow \mathbb{F}^{N \times k}} \left[\exists \boldsymbol{s}_1 \neq \boldsymbol{s}_2 \in \mathbb{F}^k, \boldsymbol{e}_1, \boldsymbol{e}_2 \in \mathbb{F}^N : |\boldsymbol{e}_1|, |\boldsymbol{e}_2| \leq t \wedge (\mathbf{A} \cdot \boldsymbol{s}_1 + \boldsymbol{e}_1 = \mathbf{A} \cdot \boldsymbol{s}_2 + \boldsymbol{e}_2) \right].$$

Proof. Let $\boldsymbol{s} \stackrel{\text{def}}{=} \boldsymbol{s}_1 - \boldsymbol{s}_2 \in \mathbb{F}^k$ and $\boldsymbol{e} \stackrel{\text{def}}{=} \boldsymbol{e}_2 - \boldsymbol{e}_1 \in \mathbb{F}^N$. For any $\boldsymbol{s} \neq \boldsymbol{0}$, $\mathbf{A} \cdot \boldsymbol{s}$ is uniform over $\mathbb{F}^N$. Together with $|\boldsymbol{e}| \leq 2t$, the probability that $\mathbf{A} \cdot \boldsymbol{s} = \boldsymbol{e}$ is at most $\sum_{i=0}^{2t} \binom{N}{i}/|\mathbb{F}|^{N-i} \leq \left(\sum_{i=0}^{2t} \binom{N}{i} \right)/|\mathbb{F}|^{N-2t} \leq N^{2t}/|\mathbb{F}|^{N-2t}$. By a union bound on all possible $\boldsymbol{s} \in \mathbb{F}^k$, we obtain the bound claimed in the lemma. $\qquad\square$

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 analyze the bit security of the LPN instance over $\mathbb{F}_2$ (as shown in Section 5). In the *subsequent work*, Baum et al. [BBMHS22] 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 Section 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 $(\mathsf{HW}, \mathbb{Z}_{2^\lambda})\text{-}\mathsf{LPN}(N, k, t)$ is (T, ϵ)-hard, then the decisional exact-LPN problem $(\mathsf{HW}, \mathbb{F}_2)\text{-}\mathsf{LPN}(N, k, \frac{2^{(\lambda-1)}}{2^\lambda-1}t)$ 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 $(\mathsf{Ber}, \mathbb{Z}_{2^\lambda})\text{-}\mathsf{LPN}(N, k, \mu)$ is (T, ϵ)-hard, then the decisional LPN problem $(\mathsf{Ber}, \mathbb{F}_2)\text{-}\mathsf{LPN}(N, k, \mu)$ 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 $(\mathsf{HW}, \mathbb{F}_2)\text{-}\mathsf{LPN}(N, k, t')$ 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. $\qquad\square$

Despite the preserved noise probability μ in the case of Bernoulli distribution, we note that $\mathrm{Ber}_{\mu,N}(\mathbb{Z}_{2^\lambda})$ has expected weight $(1 - 2^{-\lambda})\mu N$, while $\mathrm{Ber}_{\mu,N}(\mathbb{F}_2)$ has expected weight $\mu N/2$ that is roughly $2\times$ smaller than $\mathrm{Ber}_{\mu,N}(\mathbb{Z}_{2^\lambda})$. 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 $(\mathrm{RHW}, \mathbb{Z}_{2^\lambda})$-$\mathrm{LPN}(N, k, t)$ to the decisional exact-LPN problem $(\mathrm{HW}, \mathbb{F}_2)$-$\mathrm{LPN}(N, k, \frac{2^{(\lambda-1)}}{2^\lambda - 1}t)$. 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* $(\mathrm{Ber}, \mathbb{F}_2)$-$\mathrm{LPN}(N, k, \mu/2^\lambda)$ *is* (T, ϵ)-*hard, then decisional* $(\mathrm{Ber}, \mathbb{Z}_{2^\lambda})$-$\mathrm{LPN}(N, k, \mu)$ *is* $(T - \mathrm{poly}(N, k), \lambda \cdot \epsilon)$-*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: $(\mathbf{A}^0, \mathbf{A}^1, \cdots, \mathbf{A}^{\lambda-1}) := \mathrm{BitDecomp}(\mathbf{A})$, $(\boldsymbol{s}^0, \boldsymbol{s}^1, \cdots, \boldsymbol{s}^{\lambda-1}) := \mathrm{BitDecomp}(\boldsymbol{s})$, $(\boldsymbol{e}^0, \boldsymbol{e}^1, \cdots, \boldsymbol{e}^{\lambda-1}) := \mathrm{BitDecomp}(\boldsymbol{e})$ and $(\boldsymbol{b}^0, \boldsymbol{b}^1, \cdots, \boldsymbol{b}^{\lambda-1}) := \mathrm{BitDecomp}(\boldsymbol{b})$. 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 $\mathbf{S}(0, i-1) \overset{\mathrm{def}}{=} (\boldsymbol{s}^{i-1}, \ldots, \boldsymbol{s}^0)$, and $\mathbf{E}(0, i-1) \overset{\mathrm{def}}{=} (\boldsymbol{e}^{i-1}, \ldots, \boldsymbol{e}^0)$, and f_i sums up the other terms not depending on $\boldsymbol{s}^i$ and $\boldsymbol{e}^i$. Define the hybrid distributions

$$H_0 = (\mathbf{A}, \boldsymbol{u}_0, \cdots, \boldsymbol{u}_{i-1}, \boldsymbol{u}_i \cdots, \boldsymbol{u}_{\lambda-1})$$

$$\vdots$$

$$H_i = (\mathbf{A}, \boldsymbol{b}^0, \cdots, \boldsymbol{b}^{i-1}, \boldsymbol{u}_i \cdots, \boldsymbol{u}_{\lambda-1})$$

$$\vdots$$

$$H_\lambda = (\mathbf{A}, \boldsymbol{b}^0, \cdots, \boldsymbol{b}^{i-1}, \boldsymbol{b}^i \cdots, \boldsymbol{b}^{\lambda-1})$$

where $\boldsymbol{u}_j \leftarrow \mathbb{F}_2^N$ 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 $(\mathrm{Ber}, \mathbb{F}_2)$-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 $(e^0[j], e^1[j], \ldots, e^{\lambda-1}[j]) \leftarrow \mathrm{Ber}_{\mu,N}(\mathbb{Z}_{2^\lambda})$, where $e^i[j]$ is the j-th entry of e^i. Conditioned on any non-zero $(e^0[j], \ldots, e^{i-1}[j])$, $e^i[j]$ is uniformly random and thus unconditionally masks the corresponding $b^i[j]$. Otherwise, we have that

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

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 $i = \lambda - 1$. $\qquad\square$

Based on the above theorem, we easily obtain the following corollary.

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

Proof. By Theorem 4, it suffices to show that decisional $(\mathrm{Ber}, \mathbb{Z}_{2^\lambda})$-LPN implies its computational analogue, which in turn implies computational $(\mathrm{HW}, \mathbb{Z}_{2^\lambda})$-LPN. The former is trivial and thus it left out to show the latter. For $e \leftarrow \mathrm{Ber}_{\mu,N}(\mathbb{Z}_{2^\lambda})$, the noise vector e has expected (instead of exact) Hamming weight $(1 - 2^{-\lambda})\mu N$. The difference is not substantial for computational problems as conditioned on $|e| = (1 - 2^{-\lambda})\mu N$, which has probability $\Omega(1/\sqrt{N})$, based on Lemma 1. Therefore, with probability $\Omega(1/\sqrt{N})$, e follows the exact-weight distribution $\mathrm{HW}_{(1-2^{-\lambda})\mu N, N}(\mathbb{Z}_{2^\lambda})$, which completes the proof. $\qquad\square$

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

- $\mathrm{IndBer}_{\mu,N}(\mathbb{Z}_{2^\lambda})$ is bit-wise independent. By $e \leftarrow \mathrm{IndBer}_{\mu,N}(\mathbb{Z}_{2^\lambda})$, we mean that $e := \sum_{i=0}^{\lambda-1} 2^i \cdot e^i \in \mathbb{Z}_{2^\lambda}$ with $e^i \leftarrow \mathrm{Ber}_{\mu,N}(\mathbb{F}_2)$ for $i \in [\lambda]$. The noise rate of $\mathrm{IndBer}_{\mu,N}(\mathbb{Z}_{2^\lambda})$ is the probability that a coordinate of 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 $e \leftarrow \mathrm{IndBer}_{\mu,N}(\mathbb{Z}_{2^\lambda})$ is λt where $t = \mu N/2$.

- $\mathrm{IndHW}_{t,N}(\mathbb{Z}_{2^\lambda})$ decomposes into λ independent vectors from $\mathrm{HW}_{t,N}(\mathbb{F}_2)$. By $e \leftarrow \mathrm{IndHW}_{t,N}(\mathbb{Z}_{2^\lambda})$, we mean that $e := \sum_{i=0}^{\lambda-1} 2^i \cdot e^i$ with $e^i \leftarrow \mathrm{HW}_{t,N}(\mathbb{F}_2)$ for $i \in [\lambda]$. It is easy to see that the Hamming weight of e is at most λt.

Although $\mathrm{IndBer}_{\mu,N}(\mathbb{Z}_{2^\lambda})$ and $\mathrm{IndHW}_{t,N}(\mathbb{Z}_{2^\lambda})$ have not been used in existing protocols, LPN with such distributions can be used to design PCG-like protocols by running these protocols with maximum weight λt. 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.

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

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

- *If decisional* $(\mathcal{D}_2, \mathbb{Z}_{2^\lambda})$-$\mathrm{LPN}(N, k, w)$ *is* (T, ϵ)-*hard, then decisional* $(\mathcal{D}_1, \mathbb{F}_2)$- $\mathrm{LPN}(N, k, w)$ *is* $(T - \mathrm{poly}(N, k), \epsilon)$-*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: $(\mathcal{D}, \mathbb{Z}_{2^\lambda})$-LPN$(N, k, t)$ samples $(\mathbf{A}, \boldsymbol{b} = \mathbf{A} \cdot \boldsymbol{s} + \boldsymbol{e} \mod 2^\lambda)$
Output: $\boldsymbol{s} \in \mathbb{Z}_{2^\lambda}$
1 $(\mathbf{A}^0, \mathbf{A}^1, \cdots, \mathbf{A}^{\lambda-1}) := \mathsf{BitDecomp}(\mathbf{A})$;
2 $(\boldsymbol{b}^0, \boldsymbol{b}^1, \cdots, \boldsymbol{b}^{\lambda-1}) := \mathsf{BitDecomp}(\boldsymbol{b})$;
3 $(\boldsymbol{s}^0, \boldsymbol{e}^0) \leftarrow \mathcal{A}_{\mathrm{LPN}_2}(\mathbf{A}^0, \boldsymbol{b}^0)$;
4 $\boldsymbol{b}' := (\boldsymbol{b} - \mathbf{A} \cdot \boldsymbol{s}^0 - \boldsymbol{e}^0)/2 \mod 2^{(\lambda-1)}$;
5 **Return** $\boldsymbol{s} = \boldsymbol{s}^0 + 2 \cdot \mathcal{A}_{\mathrm{LPN}_{2^{(\lambda-1)}}}\left(\mathbf{A}' := \sum_{i=0}^{\lambda-2} 2^i \cdot \mathbf{A}^i \in \mathbb{Z}_{2^{\lambda-1}}, \boldsymbol{b}'\right)$.

Proof. The proof of the first statement is similar to that of Theorem 4, except that now every $\boldsymbol{e}^i$ is independent of the previous $\boldsymbol{e}^0, \cdots, \boldsymbol{e}^{i-1}$, where $\boldsymbol{e}^i$ follows $\mathsf{Ber}_{\mu,N}(\mathbb{F}_2)$ by the definition of $\boldsymbol{e} \leftarrow \mathsf{IndBer}_{\mu,N}(\mathbb{Z}_{2^\lambda})$. The proof of the second statement can be trivially adapted from that of Theorem 3, i.e., $\mathbf{A}^0 = \mathbf{A} \mod 2$, $\mathbf{A}^0 \cdot \boldsymbol{s}^0 + \boldsymbol{e}^0 = \mathbf{A} \cdot \boldsymbol{s} + \boldsymbol{e} \mod 2$. $\qquad\square$

On the choice of matrix A. As we can see from the proofs of Theorem 4, Theorem 5 and Theorem 6 (shown in Section 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 Section 4.1. Algorithm 1 shows how computational LPN over $\mathbb{Z}_{2^\lambda}$ is reduced to that over $\mathbb{Z}_{2^{\lambda-1}}$, whose correctness is analyzed in Lemma 4. 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 4. *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 $\boldsymbol{e}' = \sum_{i=1}^{\lambda-1} 2^{i-1} \cdot \boldsymbol{e}^i \mod 2^{(\lambda-1)}$.*

Proof. Let $(\mathbf{A}^0, \mathbf{A}^1, \cdots, \mathbf{A}^{\lambda-1})$ and $(\boldsymbol{b}^0, \boldsymbol{b}^1, \cdots, \boldsymbol{b}^{\lambda-1})$ be the matrices and vectors defined in Algorithm 1. Let $(\boldsymbol{s}^0, \boldsymbol{s}^1, \cdots, \boldsymbol{s}^{\lambda-1}) := \mathsf{BitDecomp}(\boldsymbol{s})$ and $(\boldsymbol{e}^0, \boldsymbol{e}^1, \ldots, \boldsymbol{e}^{\lambda-1}) := \mathsf{BitDecomp}(\boldsymbol{e})$. Note that $\mathbf{A}'$ is obtained from $\mathbf{A}$ by truncating the most significant bits (MSBs), and thus follows the distribution in LPN over $\mathbb{Z}_{2^{(\lambda-1)}}$. It suffices to prove $\boldsymbol{b}' = \mathbf{A}' \cdot \boldsymbol{s}' + \boldsymbol{e}' \mod 2^{(\lambda-1)}$, where $\boldsymbol{s}' = \sum_{i=1}^{\lambda-1} 2^{i-1} \cdot \boldsymbol{s}^i$ and $\boldsymbol{e}' = \sum_{i=1}^{\lambda-1} 2^{i-1} \cdot \boldsymbol{e}^i$ are the secret and noise of LPN over $\mathbb{Z}_{2^{(\lambda-1)}}$ respectively. In particular, we have the following:

$$
\begin{aligned}
2 \cdot \boldsymbol{b}' - 2 \cdot (\mathbf{A}' \cdot \boldsymbol{s}' + \boldsymbol{e}') &= \boldsymbol{b} - \mathbf{A} \cdot \boldsymbol{s}^0 - \boldsymbol{e}^0 - 2 \cdot (\mathbf{A}' \cdot \boldsymbol{s}' + \boldsymbol{e}') \\
&= \boldsymbol{b} - \mathbf{A} \cdot \boldsymbol{s}^0 - \boldsymbol{e}^0 - \mathbf{A} \cdot (\boldsymbol{s} - \boldsymbol{s}^0) - \boldsymbol{e} + \boldsymbol{e}^0 \\
&= \boldsymbol{b} - \mathbf{A} \cdot \boldsymbol{s} - \boldsymbol{e} = \mathbf{0} \mod 2^\lambda,
\end{aligned}
$$

where $2 \cdot \mathbf{A} = 2 \cdot \mathbf{A}' \mod 2^\lambda$ and $2 \cdot \mathbf{e}' = \mathbf{e} - \mathbf{e}^0$. Therefore, we have $\mathbf{b}' = \mathbf{A}' \cdot \mathbf{s}' + \mathbf{e}' \mod 2^{(\lambda-1)}$, which completes the proof. $\qquad\square$

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 $\epsilon = 2^{-k^{0.25}}$.

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

Proof. For contradiction assume that there exists an algorithm $\mathcal{A}_{\mathrm{LPN}_2}$ that recovers the secret of LPN over $\mathbb{F}_2$ with probability more than 2ϵ within time T. It suffices to prove the case $(D_1, D_2, w) = (\mathsf{Ber}, \mathsf{Ber}, \mu)$. For proof convenience, consider distribution $\mathsf{Ber}_{\mu,N}(\mathbb{F}_2)$ being sampled in two steps: (1) pick each coordinate with probability μ independently (and let the rest with 0's); and (2) assign the picked coordinates with uniform random bits. [6] Let $\mathbf{A}^0$, $\mathbf{s}_i$ and $\mathbf{e}_i$ for $i \in [\lambda]$ be the vectors defined in Lemma 4. By Markov inequality, there exists at least an ϵ fraction of good $(\mathbf{A}^0, \mathsf{coin}(\mathbf{e}^i))$ for which $\mathcal{A}_{\mathrm{LPN}_2}$ recovers $\mathbf{s}^i$ from $(\mathbf{A}^0, \mathbf{A}^0 \cdot \mathbf{s}^i + \mathbf{e}^i)$ with probability at least ϵ, where $\mathsf{coin}(\mathbf{e}^i)$ denotes the step-1 random coin for sampling $\mathbf{e}^i$, and the probability is taken over the $\mathbf{s}^i$ and the step-2 coin of $\mathbf{e}^i$. Therefore, $\mathcal{A}_{\mathrm{LPN}_{2^\lambda}}$ (see Algorithm 1) invokes $\mathcal{A}_{\mathrm{LPN}_2}$ on $(\mathbf{A}, \mathbf{b} = \mathbf{A} \cdot \mathbf{s} + \mathbf{e} \mod 2^\lambda)$ for λ times and recovers $\mathbf{s}$ with an overall probability of at least $\epsilon^{\lambda+1}$, a contradiction to the assumption. The proofs for the other two cases (i.e., $(D_1, D_2, w) \in \{(\mathsf{IndBer}, \mathsf{Ber}, \mu), (\mathsf{IndHW}, \mathsf{HW}, t)\}$) are slightly simpler because $\mathbf{e}^0$, ..., $\mathbf{e}^{\lambda-1}$ are independent and thus no two-step sampling is needed, i.e., $\mathsf{coin}(\mathbf{e}^i)$ is empty. $\qquad\square$

Below, we further prove the following theorem.

Theorem 7. *If computational* $(\mathsf{HW}, \mathbb{Z}_{2^\lambda})$-LPN$(N, k, t)$ *is* $(\lambda \cdot T + \mathsf{poly}(N, k), \epsilon^{\lambda+1})$-*hard, then computational* $(\mathsf{HW}, \mathbb{F}_2)$-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* $\delta > 0$.

Proof. Let $\widetilde{\mathsf{HW}}_{t,N}(\mathbb{F}_2)$ be the distribution of $\mathbf{e}^i$ for $i \in [\lambda]$ when $\mathbf{e} \leftarrow \mathsf{HW}_{t,N}(\mathbb{Z}_{2^\lambda})$, where $(\mathbf{e}^0, \dots, \mathbf{e}^{\lambda-1}) := \mathsf{BitDecomp}(\mathbf{e})$. Similar to the proof of Theorem 6, we can show if there exists $\mathcal{A}_{\mathrm{LPN}_2}$ that breaks LPN over $\mathbb{F}_2$ and noise distribution $\widetilde{\mathsf{HW}}_{t,N}(\mathbb{F}_2)$ with probability more than 2ϵ within time T, then it can be used to break LPN over $\mathbb{Z}_{2^\lambda}$ and noise distribution $\mathsf{HW}_{t,N}(\mathbb{Z}_{2^\lambda})$ with probability $\epsilon^{\lambda+1}$. The expected weight of $\mathbf{e}^i$ is $\frac{2^{\lambda-1}}{2^\lambda-1}t$, and thus by a Chernoff bound $\mathbf{e}^i$ is a convex combination of distributions $\mathsf{HW}_{1,N}(\mathbb{F}_2)$, ..., $\mathsf{HW}_{t',N}(\mathbb{F}_2)$ with $t' = \frac{2^{\lambda-1}}{2^\lambda-1}(1+\delta)t$ and $\delta > 0$, except for an error probability bounded by $\exp(-\delta^2 t/6)$. Since $\mathcal{A}_{\mathrm{LPN}_2}$ works on LPN over $\mathbb{F}_2$ with noise distribution $\mathsf{HW}_{t',N}(\mathbb{F}_2)$, it should work on that with noise $\mathsf{HW}_{i,N}(\mathbb{F}_2)$ of weight up to $i = t'$ (and any their convex combination) as well. [7] Therefore, $\mathcal{A}_{\mathrm{LPN}_2}$ that breaks $(\mathsf{HW}, \mathbb{F}_2)$-LPN$(N, k, t')$ with probability $2\epsilon/(1 - \exp(-\delta^2 t/6))$ is the hypothetical algorithm, which completes the proof. $\qquad\square$

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 $(\mathsf{RHW}, \mathbb{Z}_{2^\lambda})$-LPN$(N, k, t)$ to the computational exact-LPN problem $(\mathsf{HW}, \mathbb{F}_2)$-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

---

[6]The two-step sampling is defined to be in line with $\mathsf{Ber}_{\mu,N}(\mathbb{Z}_{2^\lambda})$, and therefore captures the correlations among $\mathbf{e}^0$, ..., $\mathbf{e}^{\lambda-1}$, which share the same step-1 randomness.

[7]Strictly speaking, $\mathcal{A}_{\mathrm{LPN}_2}$ implies such an algorithm with roughly the same complexity and success probability, which can be seen by a simple reduction.

practical key recovery attacks, however, we often expect the success probability to be $(1 - 1/\text{poly}(k))$ or even overwhelming. In this case, we get more efficient reductions as below.

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

Proof. Similar to the proof of Theorem 6, we have by a Markov inequality that for at least a $(1 - \sqrt{\epsilon})$ fraction of $(\mathbf{A}^0, \text{coin}(e^i))$, $\mathcal{A}_{LPN_2}$ recovers s^i from $(\mathbf{A}^0, \mathbf{A}^0 \cdot s^i + e^i)$ with probability at least $1 - \sqrt{\epsilon}$. Overall, $\mathcal{A}_{LPN_{2^\lambda}}$ succeeds with probability $(1 - \sqrt{\epsilon})(1 - \lambda\sqrt{\epsilon}) \geq 1 - (\lambda + 1)\sqrt{\epsilon}$ by a union bound. $\qquad\square$

We also give a proof of the following theorem.

Theorem 9. *If the computational $(\text{HW}, \mathbb{F}_2)$-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 $(\text{HW}, \mathbb{Z}_{2^\lambda})$-LPN$(N, k, t)$ problem can be broken by $\mathcal{A}_{LPN_{2^\lambda}}$ (see Algorithm 1) in time $\lambda \cdot T + \text{poly}(N, k)$ 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 $\delta^2 t \geq 6 \ln(2/\epsilon)$.*

Proof. Similar to the proof of Theorem 8, as long as we succeed in breaking the LPN problem over $\mathbb{F}_2$ and noise e^i with probability at least $(1 - \epsilon)$, then the rest follows from Markov inequality and a union bound. As analyzed in the proof of Theorem 7, e^i is $\exp(-\delta^2 t/6)$-close to a convex combination of distributions $\text{HW}_{1,N}(\mathbb{F}_2), \ldots, \text{HW}_{t',N}(\mathbb{F}_2)$ with $t' = \frac{2^{\lambda-1}}{2^\lambda-1}(1+\delta)t$ and $\delta > 0$. Therefore, we need $\mathcal{A}_{LPN_2}$ to be successful on LPN over $\mathbb{F}_2$ and noise $\text{HW}_{t',N}(\mathbb{F}_2)$ with probability at least

$$\frac{1 - \epsilon}{1 - \exp(-\delta^2 t/6)} \leq 1 - \left(\epsilon - \exp(-\delta^2 t/6)\right) \leq 1 - \frac{\epsilon}{2} \ .$$

$\square$

Optimized attacks on $(\text{Ber}/\text{HW}, \mathbb{Z}_{2^\lambda})$**-LPN.** In practice, we optimize the attacks on $(\text{Ber}/\text{HW}, \mathbb{Z}_{2^\lambda})$-LPN by exploiting the correlations among the noise vectors of the λ instances (i.e., $e^0, \ldots, e^{\lambda-1}$). In particular, Algorithm 1 recovers the corresponding secrets $s^0, s^1, \cdots, s^{\lambda-1}$ sequentially. That means when the attacker works on the $(i+1)$-th LPN instance, it has already seen $e^0, \ldots, e^{i-1}$ from the previous i broken instances. As analyzed in the proof of Theorem 4, for any single noise sample $(e^0[j], e^1[j], \ldots, e^{\lambda-1}[j]) \leftarrow \text{Ber}_{\mu,N}(\mathbb{Z}_{2^\lambda})$, $e^i[j]$ is uniformly random conditioned on any non-zero $(e^0[j], \ldots, e^{i-1}[j])$, and thus sample $b^i[j]$ 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 $e^0, \ldots, e^{i-1}$. Therefore, the success rate of solving the $(\text{Ber}, \mathbb{Z}_{2^\lambda})$-LPN$(N, k, \mu)$ instance is roughly the product of the λ instances of $(\text{Ber}, \mathbb{F}_2)$-LPN with continuously halving noise rates $\mu, \mu/2, \ldots, \mu/2^{\lambda-1}$. For instance, if solving these instances can succeed with probability $\epsilon, \epsilon^{2^{-1}}, \ldots, \epsilon^{2^{-(\lambda-1)}}$ respectively, then it leads to a success probability of approximately ϵ^2 (instead of $\epsilon^{\lambda+1}$). The optimization for reducing $(\text{HW}, \mathbb{Z}_{2^\lambda})$-LPN to $(\text{HW}, \mathbb{F}_2)$-LPN is likewise.

5 Concrete Analysis of Low-Noise LPN over Finite Fields

Recently, a series of works [BCGI18, BCG$^+$19a, SGRR19, BCG$^+$19b, YWL$^+$20, WYKW21, CRR21, BCG$^+$22, BCCD23, RRT23] 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. [BCGI18] to select the concrete parameters for some specified security level. Boyle et al. [BCGI18] obtained the formulas by analyzing three attacks: Pooled Gauss [EKM17], ISD [Pra62] and SD [Al 01]. 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 [Pra62] to solve LPN problems over a large field. This does not cover the advanced ISD variants [Ste88, Dum91, MMT11, BJMM12]. 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 [DT17].

We also give more accurate formulas on the hardness of low-noise (dual-)LPN problems, where the recent SD improvement called SD 2.0 [CDHT22] is also included. Very recently, Meyer-Hilfiger and Tillich [MHT23] 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. [BCGI18] in Tables 6 and 7 in Appendix B, where all the LPN parameters are adopted from [BCGI18]. Under the same LPN parameters, while Boyle et al. [BCGI18] showed that either Pooled Gauss attack or SD attack has the lowest cost, our analysis shows that ISD attack has the lowest cost. 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 [FJR22]. This justifies that it is not accurate to use the same formulas for all field sizes as in [BCGI18].

Under the Gilbert-Varshamov (GV) bound [8], Carrier et al. [CDHT22] 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 [CDHT22] does not behave better when solving the low-noise LPN problems used in the PCG-like protocols. This is because the collision technique [9] used by 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 $0 < c < 1$. This is the case after we incorporate into SD 2.0 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 [BCGI18, Section 2.3], originated from [Zic17]). In Appendix B.2, we prove that the SD 2.0 attack [CDHT22] (that improves the SD attack) adapted to the low-noise setting require more cost than the ISD attack against $(\mathsf{HW}, \mathbb{F})\text{-LPN}(N, k, t)$ with field size $|\mathbb{F}| \geq 4t$.

The previous analysis [BCGI18] 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 $(\mathsf{RHW}, \mathbb{F}_2)\text{-LPN}(N, k, t)$ into an exact-LPN problem $(\mathsf{HW}, \mathbb{F}_2)\text{-LPN}(N - t, k - t, t)$ based on the approach in prior works [EMZ22, BØ23]. Then, we solve a solution of $(\mathsf{HW}, \mathbb{F}_2)\text{-LPN}(N - t, k - t, t)$ using the known attacks without relying on the regular structure. This transformation from regular-LPN to exact-LPN

---

[8]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 $\mathbf{H}$.

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

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: 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 [BØ23].

Regular dual-LPN over a field $\mathbb{F}$			This work ($\log|\mathbb{F}| = 128$)					This work ($\log|\mathbb{F}| = 1$)				
n	N	t	Gauss	SD	SD 2.0	ISD	AGB	Gauss	SD	SD 2.0	ISD	AGB
2^{10}	2^{12}	44	117	189	189	117	127	116	190	167	96	125
2^{12}	2^{14}	39	111	170	170	111	127	111	170	165	95	127
2^{14}	2^{16}	34	107	151	151	107	128	107	151	150	93	128
2^{16}	2^{18}	32	108	145	145	108	132	108	145	145	95	132
2^{18}	2^{20}	31	112	143	143	112	139	112	143	143	99	139
2^{20}	2^{22}	30	116	141	141	116	145	116	141	141	103	145
2^{22}	2^{24}	29	119	139	139	119	152	119	139	139	107	152

Table 4: The bit-security of dual-LPN problems over finite fields with dimension $N - n = 3N/4$, number of samples N and Hamming weight of noises t for a regular noise distribution. "AGB" denotes the recent algebraic attack [BØ23].

works for LPN over $\mathbb{F}_2$, but fails to work for LPN over larger fields (see more details in Section 5.1). On the other hand, our analysis includes the recent algebraic attack by Briaud and Øygarden [BØ23], 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 [CCJ23] 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 [BCGI18], which is shown in Tables 3 and 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 3 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 significantly lower cost than other attacks for some parameter sets (see Table 8 in Appendix B), which has been observed in [BØ23]. 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 5, we need to increase the dimension of LPN problems with a

#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%)

Table 5: Comparison of dimensions between exact-LPN problems and regular-LPN problems over finite fields for 128-bit security level.

regular noise distribution by $0.5\% \sim 48.3\%$ 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 [YWL$^+$20]. For dual-LPN problems, we note that the algebraic attack [BØ23] 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 all existing attacks being applicable to LPN problems in the PCG setting, to automatically evaluate the bit security of a low-noise LPN problem. 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 [EB22] 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 Section 5.1, we first show that $(\mathsf{RHW}, \mathbb{F}_2)\text{-LPN}(N, k, t)$ is not harder than $(\mathsf{HW}, \mathbb{F}_2)\text{-LPN}(N - t, k - t, t)$, 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 Appendix B, we show the imprecisions of the previous analysis [BCGI18] 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 [EMZ22, BØ23], we transform a regular-LPN problem $(\mathsf{RHW}, \mathbb{F}_2)\text{-LPN}(N, k, t)$ into an exact-LPN problem $(\mathsf{HW}, \mathbb{F}_2)\text{-LPN}(N - t, k - t, t)$. 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 $(\mathsf{HW}, \mathbb{F}_2)\text{-LPN}(N - t, k - t, t)$ is always that of $(\mathsf{RHW}, \mathbb{F}_2)\text{-LPN}(N, k, t)$.

Let $m = \lfloor N/t \rfloor$. Given a $(\mathsf{RHW}, \mathbb{F}_2)\text{-LPN}(N, k, t)$ 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, for each $i \in [1,t]$, one of the components of secret vector $\boldsymbol{s}$ is determined by other components. In particular, we can compute $\boldsymbol{s}[0], \ldots, \boldsymbol{s}[t-1]$ with other components in $\boldsymbol{s}$. Then we can remove $\boldsymbol{s}[0], \ldots, \boldsymbol{s}[t-1]$ from $\boldsymbol{s}$, which is equivalent to reduce the dimension of $\boldsymbol{s}$ by t.

For each block (i.e., each $i \in [1,t]$), we pick one sample at random and then remove it. This enables us to eliminate the correlation between the samples within the same block, which is necessary to ensure that the remaining samples still constitute an LPN instance. Finally, we randomly permute all the remaining samples, which results in a new 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 $(\mathsf{HW}, \mathbb{F}_2)\text{-LPN}(N-t, k-t, t)$. Therefore, we can use the bit security of an exact-LPN instance $(\mathsf{HW}, \mathbb{F}_2)\text{-LPN}(N-t, k-t, t)$, based on all known attacks against exact-LPN, to estimate that of a regular-LPN instance $(\mathsf{RHW}, \mathbb{F}_2)\text{-LPN}(N, k, t)$. We can convert a dual-LPN problem into an LPN problem using the approach in [MM11]. 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 [BØ23] 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 [BØ23], 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 $e_i[j_1] \cdot e_i[j_2] = 0$ 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 $(e_i[j])^2 = e_i[j]$ and $\sum_{j=1}^{m} e_i[j] = 1$, which guarantees that every e_i is a unit vector. Standard algorithms such as XL/Gröbner bases [Wie86, Tho02, Cop94, Beu21] 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 e, inspired from the regular version of Prange's algorithm [HOSS18]. 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 Tables 3 and 8). The algebraic attack does not outperform ISD attacks for dual-LPN problems used in PCG-like protocols which have the 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 [BØ23] against LPN problems, while keeping the efficiency essentially unchanged due to the usage of bootstrapping iterations [YWL+20].

Acknowledgements

References

[AFS05] Daniel Augot, Matthieu Finiasz, and Nicolas Sendrier. A family of fast syndrome based cryptographic hash functions. In Ed Dawson and Serge Vaudenay, editors, *Mycrypt 2005*, volume 3715 of *Lecture Notes in Computer Science*, pages 64–83. Springer, 2005.

[AIK07] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography with constant input locality. In Alfred Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 92–110. Springer, Heidelberg, August 2007.

[Aka08] Adi Akavia. *Learning Noisy Characters, Multiplication Codes, and Cryptographic Hardcore Predicates*. PhD thesis, Massachusetts Institute of Technology, 2008.

[Al 01] A. Kh. Al Jabri. A statistical decoding algorithm for general linear block codes. In Bahram Honary, editor, *8th IMA International Conference on Cryptography and Coding*, volume 2260 of *LNCS*, pages 1–8. Springer, Heidelberg, December 2001.

[Ale03] Michael Alekhnovich. More on average case vs approximation complexity. In *44th FOCS*, pages 298–307. IEEE Computer Society Press, October 2003.

[AMBD+18]   Carlos Aguilar-Melchor, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, and Gilles Zémor. Efficient encryption from random quasi-cyclic codes. *IEEE Transactions on Information Theory*, 64, 2018.

[App16]   Benny Applebaum. Garbling XOR gates "for free" in the standard model. *Journal of Cryptology*, 29(3):552–576, July 2016.

[AS22]   Damiano Abram and Peter Scholl. Low-communication multiparty triple generation for SPDZ from ring-LPN. In Goichiro Hanaoka, Junji Shikata, and Yohei Watanabe, editors, *PKC 2022, Part I*, volume 13177 of *LNCS*, pages 221–251. Springer, Heidelberg, March 2022.

[BA21]   Gregory Bard and Martin Albrecht. M4ri(e)- linear algebra over $\mathbf{F}_2$ (and $\mathbf{F}_{2^e}$). In *Free Open Source Software*, 2021.

[BBC+19]   Marco Baldi, Alessandro Barenghi, Franco Chiaraluce, Gerardo Pelosi, and Paolo Santini. A finite regime analysis of information set decoding algorithms. *Algorithms*, 12(10), 2019.

[BBMH+21]   Carsten Baum, Lennart Braun, Alexander Munch-Hansen, Benoît Razet, and Peter Scholl. Appenzeller to brie: Efficient zero-knowledge proofs for mixed-mode arithmetic and Z2k. In Giovanni Vigna and Elaine Shi, editors, *ACM CCS 2021*, pages 192–211. ACM Press, November 2021.

[BBMHS22]   Carsten Baum, Lennart Braun, Alexander Munch-Hansen, and Peter Scholl. Moz$\mathbb{Z}_{2^k}$arella: Efficient vector-OLE and zero-knowledge proofs over $\mathbb{Z}_{2^k}$. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part IV*, volume 13510 of *LNCS*, pages 329–358. Springer, Heidelberg, August 2022.

[BC23]   Dung Bui and Geoffroy Couteau. Improved private set intersection for sets with small entries. In Alexandra Boldyreva and Vladimir Kolesnikov, editors, *PKC 2023, Part II*, volume 13941 of *LNCS*, pages 190–220. Springer, Heidelberg, May 2023.

[BCCD23]   Maxime Bombar, Geoffroy Couteau, Alain Couvreur, and Clément Ducros. Correlated pseudorandomness from the hardness of quasi-abelian decoding. In Helena Handschuh and Anna Lysyanskaya, editors, *Advances in Cryptology – CRYPTO 2023*, pages 567–601. Springer Nature Switzerland, 2023.

[BCDL19]   Rémi Bricout, André Chailloux, Thomas Debris-Alazard, and Matthieu Lequesne. Ternary syndrome decoding with large weight. In Kenneth G. Paterson and Douglas Stebila, editors, *SAC 2019*, volume 11959 of *LNCS*, pages 437–466. Springer, Heidelberg, August 2019.

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

[BCG+19b]   Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, and Peter Scholl. Efficient pseudorandom correlation generators: Silent OT extension and more. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 489–518. Springer, Heidelberg, August 2019.

[BCG+20]   Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, and Peter Scholl. Efficient pseudorandom correlation generators from ring-LPN. In Daniele Micciancio and Thomas

Ristenpart, editors, *CRYPTO 2020, Part II*, volume 12171 of *LNCS*, pages 387–416. Springer, Heidelberg, August 2020.

[BCG+22] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, Nicolas Resch, and Peter Scholl. Correlated pseudorandomness from expand-accumulate codes. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 603–633. Springer, Heidelberg, August 2022.

[BCGI18] Elette Boyle, Geoffroy Couteau, Niv Gilboa, and Yuval Ishai. Compressing vector OLE. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *ACM CCS 2018*, pages 896–912. ACM Press, October 2018.

[Beu21] Ward Beullens. Improved cryptanalysis of UOV and Rainbow. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 348–373. Springer, Heidelberg, October 2021.

[BFKL94] Avrim Blum, Merrick L. Furst, Michael J. Kearns, and Richard J. Lipton. Cryptographic primitives based on hard learning problems. In Douglas R. Stinson, editor, *CRYPTO'93*, volume 773 of *LNCS*, pages 278–291. Springer, Heidelberg, August 1994.

[BGI15] Elette Boyle, Niv Gilboa, and Yuval Ishai. Function secret sharing. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 337–367. Springer, Heidelberg, April 2015.

[BJMM12] Anja Becker, Antoine Joux, Alexander May, and Alexander Meurer. Decoding random binary linear codes in $2^{n/20}$: How $1 + 1 = 0$ improves information set decoding. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 520–536. Springer, Heidelberg, April 2012.

[BKW00] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. In *32nd ACM STOC*, pages 435–440. ACM Press, May 2000.

[BLP11] Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Smaller decoding exponents: Ball-collision decoding. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 743–760. Springer, Heidelberg, August 2011.

[BLVW19] Zvika Brakerski, Vadim Lyubashevsky, Vinod Vaikuntanathan, and Daniel Wichs. Worst-case hardness for LPN and cryptographic hashing via code smoothing. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part III*, volume 11478 of *LNCS*, pages 619–635. Springer, Heidelberg, May 2019.

[BM18] Leif Both and Alexander May. Decoding linear codes with high error rate and its impact for LPN security. In Tanja Lange and Rainer Steinwandt, editors, *Post-Quantum Cryptography - 9th International Conference, PQCrypto 2018*, pages 25–46. Springer, Heidelberg, 2018.

[BMRS21] Carsten Baum, Alex J. Malozemoff, Marc B. Rosen, and Peter Scholl. Mac'n'cheese: Zero-knowledge proofs for boolean and arithmetic circuits with nested disjunctions. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part IV*, volume 12828 of *LNCS*, pages 92–122, Virtual Event, August 2021. Springer, Heidelberg.

[BØ23] Pierre Briaud and Morten Øygarden. A new algebraic approach to the regular syndrome decoding problem and implications for PCG constructions. In Carmit Hazay and Martijn

Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 391–422. Springer, Heidelberg, April 2023.

[CCJ23]   Eliana Carozza, Geoffroy Couteau, and Antoine Joux. Short signatures from regular syndrome decoding in the head. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part V*, volume 14008 of *LNCS*, pages 532–563. Springer, Heidelberg, April 2023.

[CDE$^+$18]   Ronald Cramer, Ivan Damgård, Daniel Escudero, Peter Scholl, and Chaoping Xing. SPD $\mathbb{Z}_{2^k}$: Efficient MPC mod 2^k for dishonest majority. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 769–798. Springer, Heidelberg, August 2018.

[CDHT22]   Kevin Carrier, Thomas Debris Alazard, Charles Meyer Hilfiger, and Jean-Pierre Tillich. Statistical decoding 2.0: Reducing decoding to LPN. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part IV*, volume 13794 of *LNCS*, pages 477–507. Springer, Heidelberg, December 2022.

[CG90]   John T. Coffey and Rodney M. Goodman. The complexity of information set decoding. *IEEE Transactions on Information Theory*, 36(5), 1990.

[Cop94]   Don Coppersmith. Solving homogeneous linear equations over gf(2) via block wiedemann algorithm. *Mathematics of Computation*, 62(205):333–350, 1994.

[CRR21]   Geoffroy Couteau, Peter Rindal, and Srinivasan Raghuraman. Silver: Silent VOLE and oblivious transfer from hardness of decoding structured LDPC codes. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part III*, volume 12827 of *LNCS*, pages 502–534, Virtual Event, August 2021. Springer, Heidelberg.

[CWYY23]   Hongrui Cui, Xiao Wang, Kang Yang, and Yu Yu. Actively secure half-gates with minimum overhead under duplex networks. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part II*, volume 14005 of *LNCS*, pages 35–67. Springer, Heidelberg, April 2023.

[DDN14]   Bernardo David, Rafael Dowsley, and Anderson C. A. Nascimento. Universally composable oblivious transfer based on a variant of LPN. In Dimitris Gritzalis, Aggelos Kiayias, and Ioannis G. Askoxylakis, editors, *CANS 14*, volume 8813 of *LNCS*, pages 143–158. Springer, Heidelberg, October 2014.

[DEF$^+$19]   Ivan Damgård, Daniel Escudero, Tore Kasper Frederiksen, Marcel Keller, Peter Scholl, and Nikolaj Volgushev. New primitives for actively-secure MPC over rings with applications to private machine learning. In *2019 IEEE Symposium on Security and Privacy*, pages 1102–1120. IEEE Computer Society Press, May 2019.

[DEM19]   Claire Delaplace, Andre Esser, and Alexander May. Improved low-memory subset sum and LPN algorithms via multiple collisions. In Martin Albrecht, editor, *17th IMA International Conference on Cryptography and Coding*, volume 11929 of *LNCS*, pages 178–199. Springer, Heidelberg, December 2019.

[DILO22a]   Samuel Dittmer, Yuval Ishai, Steve Lu, and Rafail Ostrovsky. Authenticated garbling from simple correlations. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part IV*, volume 13510 of *LNCS*, pages 57–87. Springer, Heidelberg, August 2022.

[DILO22b] Samuel Dittmer, Yuval Ishai, Steve Lu, and Rafail Ostrovsky. Improving line-point zero knowledge: Two multiplications for the price of one. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *ACM CCS 2022*, pages 829–841. ACM Press, November 2022.

[DIO21] Samuel Dittmer, Yuval Ishai, and Rafail Ostrovsky. Line-point zero knowledge and its applications. In *2nd Conference on Information-Theoretic Cryptography*, 2021.

[DKL09] Yevgeniy Dodis, Yael Tauman Kalai, and Shachar Lovett. On cryptography with auxiliary input. In Michael Mitzenmacher, editor, *41st ACM STOC*, pages 621–630. ACM Press, May / June 2009.

[DPSZ12] Ivan Damgård, Valerio Pastro, Nigel P. Smart, and Sarah Zakarias. Multiparty computation from somewhat homomorphic encryption. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 643–662. Springer, Heidelberg, August 2012.

[DT17] Thomas Debris-Alazard and Jean-Pierre Tillich. Statistical decoding. In *ISIT 2017*, 2017.

[Dum91] Ilya Dumer. On minimum distance decoding of linear codes. In *Proc. 5th Joint Soviet-Swedish Int. Workshop Inform. Theory*, 1991.

[EB22] Andre Esser and Emanuele Bellini. Syndrome decoding estimator. In Goichiro Hanaoka, Junji Shikata, and Yohei Watanabe, editors, *PKC 2022, Part I*, volume 13177 of *LNCS*, pages 112–141. Springer, Heidelberg, March 2022.

[EHK+18] Andre Esser, Felix Heuer, Robert Kübler, Alexander May, and Christian Sohler. Dissection-BKW. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 638–666. Springer, Heidelberg, August 2018.

[EKM17] Andre Esser, Robert Kübler, and Alexander May. LPN decoded. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 486–514. Springer, Heidelberg, August 2017.

[EKZ21] Andre Esser, Robert Kübler, and Floyd Zweydinger. A faster algorithm for finding closest pairs in hamming metric. In *FSTTCS 2021*, volume 213, 2021.

[EMZ22] Andre Esser, Alexander May, and Floyd Zweydinger. McEliece needs a break - solving McEliece-1284 and quasi-cyclic-2918 with modern ISD. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part III*, volume 13277 of *LNCS*, pages 433–457. Springer, Heidelberg, May / June 2022.

[FJR22] Thibauld Feneuil, Antoine Joux, and Matthieu Rivain. Syndrome decoding in the head: Shorter signatures from zero-knowledge proofs. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 541–572. Springer, Heidelberg, August 2022.

[FKI07] Marc P. C. Fossorier, Kazukuni Kobara, and Hideki Imai. Modeling bit flipping decoding based on nonorthogonal check sums with application to iterative decoding attack of mceliece cryptosystem. *IEEE Trans. Inf. Theory*, 53(1), 2007.

[FKL+21] Nicholas Franzese, Jonathan Katz, Steve Lu, Rafail Ostrovsky, Xiao Wang, and Chenkai Weng. Constant-overhead zero-knowledge for RAM programs. In Giovanni Vigna and Elaine Shi, editors, *ACM CCS 2021*, pages 178–191. ACM Press, November 2021.

[FS96]   Jean-Bernard Fischer and Jacques Stern. An efficient pseudo-random generator provably as secure as syndrome decoding. In Ueli M. Maurer, editor, *EUROCRYPT'96*, volume 1070 of *LNCS*, pages 245–255. Springer, Heidelberg, May 1996.

[FS09]   Matthieu Finiasz and Nicolas Sendrier. Security bounds for the design of code-based cryptosystems. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 88–105. Springer, Heidelberg, December 2009.

[GKH17]   Cheikh Thiécoumba Gueye, Jean Belo Klamti, and Shoichi Hirose. Generalization of BJMM-ISD using may-ozerov nearest neighbor algorithm over an arbitrary finite field $\mathbb{F}_q$. In *International Conference on Codes, Cryptology, and Information Security*, volume 10194, 2017.

[HB01]   Nicholas J. Hopper and Manuel Blum. Secure human identification protocols. In Colin Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 52–66. Springer, Heidelberg, December 2001.

[Hir16]   Shoichi Hirose. May-ozerov algorithm for nearest-neighbor problem over $\mathbb{F}(q)$ and its application to information set decoding. In *SECITC 2016*, volume 10006, 2016.

[HJ10]   Nick Howgrave-Graham and Antoine Joux. New generic algorithms for hard knapsacks. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 235–256. Springer, Heidelberg, May / June 2010.

[HjLHD22] Zhicong Huang, Wen jie Lu, Cheng Hong, and Jiansheng Ding. Cheetah: Lean and fast secure two-party deep neural network inference. In Kevin R. B. Butler and Kurt Thomas, editors, *USENIX Security 2022*, pages 809–826. USENIX Association, August 2022.

[HLL+23] Xiaoyang Hou, Jian Liu, Jingyu Li, Yuhan Li, Wen jie Lu, Cheng Hong, and Kui Ren. Ciphergpt: Secure two-party gpt inference. Cryptology ePrint Archive, Paper 2023/1147, 2023. https://eprint.iacr.org/2023/1147.

[HOSS18] Carmit Hazay, Emmanuela Orsini, Peter Scholl, and Eduardo Soria-Vazquez. TinyKeys: A new approach to efficient multi-party computation. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part III*, volume 10993 of *LNCS*, pages 3–33. Springer, Heidelberg, August 2018.

[HS13]   Yann Hamdaoui and Nicolas Sendrier. A non asymptotic analysis of information set decoding. Cryptology ePrint Archive, Report 2013/162, 2013. https://eprint.iacr.org/2013/162.

[HSS20]   Carmit Hazay, Peter Scholl, and Eduardo Soria-Vazquez. Low cost constant round MPC combining BMR and oblivious transfer. *Journal of Cryptology*, 33(4):1732–1786, October 2020.

[JKPT12] Abhishek Jain, Stephan Krenn, Krzysztof Pietrzak, and Aris Tentes. Commitments and efficient zero-knowledge proofs from learning parity with noise. In Xiaoyun Wang and Kazue Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 663–680. Springer, Heidelberg, December 2012.

[JLS21]   Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. In Samir Khuller and Virginia Vassilevska Williams, editors, *53rd ACM STOC*, pages 60–73. ACM Press, June 2021.

[KOS15] Marcel Keller, Emmanuela Orsini, and Peter Scholl. Actively secure OT extension with optimal overhead. In Rosario Gennaro and Matthew J. B. Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 724–741. Springer, Heidelberg, August 2015.

[KOS16] Marcel Keller, Emmanuela Orsini, and Peter Scholl. MASCOT: Faster malicious arithmetic secure computation with oblivious transfer. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 2016*, pages 830–842. ACM Press, October 2016.

[KSS10] Jonathan Katz, Ji Sun Shin, and Adam Smith. Parallel and concurrent security of the HB and HB+ protocols. *Journal of Cryptology*, 23(3):402–421, July 2010.

[LB88] Pil Joong Lee and Ernest F. Brickell. An observation on the security of McEliece's public-key cryptosystem. In C. G. Günther, editor, *EUROCRYPT'88*, volume 330 of *LNCS*, pages 275–280. Springer, Heidelberg, May 1988.

[Leo88] Jeffrey S. Leon. A probabilistic algorithm for computing minimum weights of large error-correcting codes. *IEEE Trans. Inf. Theory*, 34, 1988.

[LXY23] Fuchun Lin, Chaoping Xing, and Yizhou Yao. 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.

[LY22] Hanlin Liu and Yu Yu. A non-heuristic approach to time-space tradeoffs and optimizations for BKW. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022, Part III*, volume 13793 of *LNCS*, pages 741–770. Springer, Heidelberg, December 2022.

[Lyu05] Vadim Lyubashevsky. The parity problem in the presence of noise, decoding random linear codes, and the subset sum problem. In *RANDOM 2005*, volume 3624, 2005.

[Meu12] Alexander Meurer. A coding-theoretic approach to cryptanalysis. Universität Bochum Ruhr, November 2012. Dissertation thesis.

[MHT23] Charles Meyer-Hilfiger and Jean-Pierre Tillich. Rigorous foundations for dual attacks in coding theory. Cryptology ePrint Archive, Paper 2023/1460, to appear at TCC 2023, 2023. https://eprint.iacr.org/2023/1460.

[MM11] Daniele Micciancio and Petros Mol. Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 465–484. Springer, Heidelberg, August 2011.

[MMT11] Alexander May, Alexander Meurer, and Enrico Thomae. Decoding random linear codes in $\tilde{\mathcal{O}}(2^{0.054n})$. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 107–124. Springer, Heidelberg, December 2011.

[MO15] Alexander May and Ilya Ozerov. On computing nearest neighbors with applications to decoding of binary linear codes. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 203–228. Springer, Heidelberg, April 2015.

[MTSB13] Rafael Misoczki, Jean-Pierre Tillich, Nicolas Sendrier, and Paulo S. L. M. Barreto. Mdpc-mceliece: New mceliece variants from moderate density parity-check codes. In *Proceedings of the 2013 IEEE International Symposium on Information Theory, 2013*, pages 2069–2073. IEEE, 2013.

[NNOB12]   Jesper Buus Nielsen, Peter Sebastian Nordholt, Claudio Orlandi, and Sai Sheshank Burra. A new approach to practical active-secure two-party computation. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 681–700. Springer, Heidelberg, August 2012.

[Ove06]   Raphael Overbeck. Statistical decoding revisited. In Lynn Margaret Batten and Reihaneh Safavi-Naini, editors, *ACISP 06*, volume 4058 of *LNCS*, pages 283–294. Springer, Heidelberg, July 2006.

[Pet10]   Christiane Peters. Information-set decoding for linear codes over F_q. In Nicolas Sendrier, editor, *The Third International Workshop on Post-Quantum Cryptography, PQCRYPTO 2010*, pages 81–94. Springer, Heidelberg, May 2010.

[Pra62]   Eugene Prange. The use of information sets in decoding cyclic codes. *IRE Trans. Inf. Theory*, 8, 1962.

[RR22]   Srinivasan Raghuraman and Peter Rindal. Blazing fast PSI from improved OKVS and subfield VOLE. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *ACM CCS 2022*, pages 2505–2517. ACM Press, November 2022.

[RRT23]   Srinivasan Raghuraman, Peter Rindal, and Titouan Tanguy. Expand-convolute codes for pseudorandom correlation generators from LPN. In *Advances in Cryptology – CRYPTO 2023*, pages 602–632. Springer Nature Switzerland, 2023.

[RS21]   Peter Rindal and Phillipp Schoppmann. VOLE-PSI: Fast OPRF and circuit-PSI from vector-OLE. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part II*, volume 12697 of *LNCS*, pages 901–930. Springer, Heidelberg, October 2021.

[Sen11]   Nicolas Sendrier. Decoding one out of many. In Bo-Yin Yang, editor, *Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011*, pages 51–67. Springer, Heidelberg, November / December 2011.

[SGRR19]   Phillipp Schoppmann, Adrià Gascón, Leonie Reichert, and Mariana Raykova. Distributed vector-OLE: Improved constructions and implementation. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *ACM CCS 2019*, pages 1055–1072. ACM Press, November 2019.

[Ste88]   Jacques Stern. A method for finding codewords of small weight. In *Coding Theory and Applications*, volume 388, 1988.

[Tho02]   Emmanuel Thomé. Subquadratic computation of vector generating polynomials and improvement of the block wiedemann algorithm. *Journal of symbolic computation*, 33(5):757–775, 2002.

[Tor17]   Rodolfo Canto Torres. Asymptotic analysis of isd algorithms for the q-ary case. In *Proceedings of the Tenth International Workshop on Coding and Cryptography WCC 2017*, Sept. 2017.

[TS16]   Rodolfo Canto Torres and Nicolas Sendrier. Analysis of information set decoding for a sublinear error weight. In Tsuyoshi Takagi, editor, *Post-Quantum Cryptography - 7th International Workshop, PQCrypto 2016*, pages 144–161. Springer, Heidelberg, 2016.

[Wag02]   David Wagner. A generalized birthday problem. In Moti Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 288–303. Springer, Heidelberg, August 2002.

[Wie86] Douglas H. Wiedemann. Solving sparse linear equations over finite fields. *IEEE Trans. Inf. Theory*, 32(1):54–62, 1986.

[WRK17a] Xiao Wang, Samuel Ranellucci, and Jonathan Katz. Authenticated garbling and efficient maliciously secure two-party computation. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 21–37. ACM Press, October / November 2017.

[WRK17b] Xiao Wang, Samuel Ranellucci, and Jonathan Katz. Global-scale secure multiparty computation. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 39–56. ACM Press, October / November 2017.

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

[WYX+21] Chenkai Weng, Kang Yang, Xiang Xie, Jonathan Katz, and Xiao Wang. Mystique: Efficient conversions for zero-knowledge proofs with applications to machine learning. In Michael Bailey and Rachel Greenstadt, editors, *USENIX Security 2021*, pages 501–518. USENIX Association, August 2021.

[WYY+22] Chenkai Weng, Kang Yang, Zhaomin Yang, Xiang Xie, and Xiao Wang. AntMan: Interactive zero-knowledge proofs with sublinear communication. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *ACM CCS 2022*, pages 2901–2914. ACM Press, November 2022.

[YS16] Yu Yu and John P. Steinberger. Pseudorandom functions in almost constant depth from low-noise LPN. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 154–183. Springer, Heidelberg, May 2016.

[YSWW21] Kang Yang, Pratik Sarkar, Chenkai Weng, and Xiao Wang. QuickSilver: Efficient and affordable zero-knowledge proofs for circuits and polynomials over any field. In Giovanni Vigna and Elaine Shi, editors, *ACM CCS 2021*, pages 2986–3001. ACM Press, November 2021.

[YWL+20] Kang Yang, Chenkai Weng, Xiao Lan, Jiang Zhang, and Xiao Wang. Ferret: Fast extension for correlated OT with small communication. In Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna, editors, *ACM CCS 2020*, pages 1607–1626. ACM Press, November 2020.

[YWZ20] Kang Yang, Xiao Wang, and Jiang Zhang. More efficient MPC from improved triple generation and authenticated garbling. In Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna, editors, *ACM CCS 2020*, pages 1627–1646. ACM Press, November 2020.

[YZW+19] Yu Yu, Jiang Zhang, Jian Weng, Chun Guo, and Xiangxue Li. Collision resistant hashing from sub-exponential learning parity with noise. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part II*, volume 11922 of *LNCS*, pages 3–24. Springer, Heidelberg, December 2019.

[Zic17] Lior Zichron. Locally computable arithmetic pseudorandom generators. Master's thesis, School of Electrical Engineering, Tel Aviv University, 2017.

A   The Hardness of LPN over More General Rings

We generalize the reductions between LPN over $\mathbb{Z}_{p^{\lambda_1}q^{\lambda_2}}$ and LPN over $\mathbb{F}_p/\mathbb{F}_q$ for distinct primes p, q. Our techniques can also be generalized to an arbitrary integer ring with more than two prime factors. We recall that every number $\boldsymbol{a} \in [p^{\lambda_1}q^{\lambda_2}]$ can be uniquely represented using the multi-base $(1, p, \ldots, p^{\lambda_1}, p^{\lambda_1}q, \ldots, p^{\lambda_1}q^{\lambda_2-1})$ for distinct primes p, q. We define function DigitDecomp that decomposes a number/vector/matrix $\boldsymbol{a} \in [p^{\lambda_1}q^{\lambda_2}]^{dim}$ (where $dim = 1, n, n_1 \times n_2$ for number, vector, or matrix respectively by applying the operation component-wise) into the above multi-base representation, i.e.,

$$\mathsf{DigitDecomp}(\boldsymbol{a}) = (\boldsymbol{a}^0, \boldsymbol{a}^1, \cdots, \boldsymbol{a}^{\lambda_1+\lambda_2-1})$$

such that $\boldsymbol{a}^i \in [p]^{dim}$ and $\boldsymbol{a}^j \in [q]^{dim}$ for every $i \in [\lambda_1]$ and $j \in [\lambda_1, \lambda_1 + \lambda_2]$, and $\boldsymbol{a} = \sum_{i=0}^{\lambda_1-1} p^i \boldsymbol{a}^i + \sum_{j=\lambda_1}^{\lambda_1+\lambda_2-1} p^{\lambda_1}q^{j-\lambda_1}\boldsymbol{a}^j$, and its inverse $\mathsf{DigitDecomp}^{-1}$ such that $\boldsymbol{a} = \mathsf{DigitDecomp}^{-1}(\boldsymbol{a}^0, \boldsymbol{a}^1, \cdots, \boldsymbol{a}^{\lambda_1+\lambda_2-1})$.

We show how to reduce the hardness of both decisional LPN over $\mathbb{F}_p$ and decisional LPN over $\mathbb{F}_q$ to that of decisional LPN over $\mathbb{Z}_{p^{\lambda_1}q^{\lambda_2}}$ with the noise distributions, $\mathsf{IndBer}_{\mu,N}(\mathbb{Z}_{p^{\lambda_1}q^{\lambda_2}})$ and $\mathsf{IndHW}_{t,N}(\mathbb{Z}_{p^{\lambda_1}q^{\lambda_2}})$, extending the noise distributions $\mathsf{IndBer}_{\mu,N}$ and $\mathsf{IndHW}_{t,N}$ from over $\mathbb{Z}_{2^\lambda}$ to $\mathbb{Z}_{p^{\lambda_1}q^{\lambda_2}}$.

- $\boldsymbol{e} \leftarrow \mathsf{IndBer}_{\mu,N}(\mathbb{Z}_{p^{\lambda_1}q^{\lambda_2}})$ refers to $\boldsymbol{e} := \mathsf{DigitDecomp}^{-1}(\boldsymbol{e}^0, \boldsymbol{e}^1, \cdots, \boldsymbol{e}^{\lambda_1+\lambda_2-1})$ with $\boldsymbol{e}^i \leftarrow \mathsf{Ber}_{\mu,N}(\mathbb{F}_p)$ and $\boldsymbol{e}^j \leftarrow \mathsf{Ber}_{\mu,N}(\mathbb{F}_q)$ for $i \in [\lambda_1]$ and $j \in [\lambda_1, \lambda_1 + \lambda_2]$.

- $\boldsymbol{e} \leftarrow \mathsf{IndHW}_{t,N}(\mathbb{Z}_{p^{\lambda_1}q^{\lambda_2}})$ means $\boldsymbol{e} := \mathsf{DigitDecomp}^{-1}(\boldsymbol{e}^0, \boldsymbol{e}^1, \cdots, \boldsymbol{e}^{\lambda_1+\lambda_2-1})$ with $\boldsymbol{e}^i \leftarrow \mathsf{HW}_{t,N}(\mathbb{F}_p)$ and $\boldsymbol{e}^j \leftarrow \mathsf{HW}_{t,N}(\mathbb{F}_q)$ for $i \in [\lambda_1]$ and $j \in [\lambda_1, \lambda_1 + \lambda_2]$.

Theorem 10 (Equivalence of Decisional LPN over $\mathbb{Z}_{p^{\lambda_1}q^{\lambda_2}}$ and $\mathbb{F}_p/\mathbb{F}_q$)**.**

1. *Both decisional* $(\mathcal{D}_1, \mathbb{F}_p)\text{-}\mathsf{LPN}(N, k, w)$ *and decisional* $(\mathcal{D}_1, \mathbb{F}_q)\text{-}\mathsf{LPN}(N, k, w)$ *are hard if and only if decisional* $(\mathcal{D}_2, \mathbb{Z}_{p^{\lambda_1}q^{\lambda_2}})\text{-}\mathsf{LPN}(N, k, w)$ *is hard, where* $(\mathcal{D}_1, \mathcal{D}_2, w) \in \{(\mathsf{Ber}, \mathsf{IndBer}, \mu), (\mathsf{HW}, \mathsf{IndHW}, t)\}$.

2. *If the decisional* $(\mathsf{Ber}, \mathbb{Z}_{p^{\lambda_1}q^{\lambda_2}})\text{-}\mathsf{LPN}(N, k, \mu)$ *problem is hard, then both decisional* $(\mathsf{Ber}, \mathbb{F}_p)\text{-}\mathsf{LPN}(N, k, \mu)$ *and decisional* $(\mathsf{Ber}, \mathbb{F}_q)\text{-}\mathsf{LPN}(N, k, \mu)$ *are hard.*

3. *If the decisional* $(\mathsf{HW}, \mathbb{Z}_{p^{\lambda_1}q^{\lambda_2}})\text{-}\mathsf{LPN}(N, k, t)$ *problem is hard, then both decisional* $(\mathsf{HW}, \mathbb{F}_p)\text{-}\mathsf{LPN}(N, k, \frac{(p-1)p^{\lambda_1-1}q^{\lambda_2}}{p^{\lambda_1}q^{\lambda_2}-1} \cdot t)$ *and decisional* $(\mathsf{HW}, \mathbb{F}_q)\text{-}\mathsf{LPN}(N, k, \frac{(q-1)p^{\lambda_1}q^{\lambda_2-1}}{p^{\lambda_1}q^{\lambda_2}-1} \cdot t)$ *are hard.*

Proof. The proof of the first statement is essentially similar to that of Theorem 4 and Theorem 5. Let $(\mathbf{A}, \boldsymbol{b} = \mathbf{A} \cdot \boldsymbol{s} + \boldsymbol{e} \mod p^{\lambda_1}q^{\lambda_2})$ be the LPN over $\mathbb{Z}_{p^{\lambda_1}q^{\lambda_2}}$. Decompose the matrices and vectors into the corresponding size-$(\lambda_1 + \lambda_2)$ lists, $(\boldsymbol{s}^0, \boldsymbol{s}^1, \cdots, \boldsymbol{s}^{\lambda_1+\lambda_2-1}) := \mathsf{DigitDecomp}(\boldsymbol{s})$, $(\boldsymbol{e}^0, \boldsymbol{e}^1, \cdots, \boldsymbol{e}^{\lambda_1+\lambda_2-1}) := \mathsf{DigitDecomp}(\boldsymbol{e})$ and $(\boldsymbol{b}^0, \boldsymbol{b}^1, \cdots, \boldsymbol{b}^{\lambda_1+\lambda_2-1}) := \mathsf{DigitDecomp}(\boldsymbol{b})$. Therefore, for $i \in [\lambda_1]$ and $j \in [\lambda_1, \lambda_1 + \lambda_2]$, we can write

$$\boldsymbol{b}^i = \mathbf{A}' \cdot \boldsymbol{s}^i + \boldsymbol{e}^i + f_i(\mathbf{A}, \boldsymbol{s}^0, \cdots, \boldsymbol{s}^{i-1}, \boldsymbol{e}^0, \cdots, \boldsymbol{e}^{i-1}) \mod p \ ,$$

$$\boldsymbol{b}^j = \mathbf{A}'' \cdot \boldsymbol{s}^j + \boldsymbol{e}^j + f_j(\mathbf{A}, \boldsymbol{s}^0, \cdots, \boldsymbol{s}^{j-1}, \boldsymbol{e}^0, \cdots, \boldsymbol{e}^{j-1}) \mod q \ ,$$

where $\mathbf{A}' = \mathbf{A} \mod p$, $\mathbf{A}'' = \mathbf{A} \mod q$ and f_i (resp., f_j) is the sum of all other terms involving the individual components of $\boldsymbol{s}$ and $\boldsymbol{e}$ with index up to $i - 1$ (resp., $j - 1$). Define the hybrid distributions $H_0, \cdots, H_{\lambda_1+\lambda_2}$ as

$$H_0 = (\mathbf{A}, \boldsymbol{u}_0, \cdots, \boldsymbol{u}_{k-1}, \boldsymbol{u}_k \cdots, \boldsymbol{u}_{\lambda_1+\lambda_2-1})$$

$$\vdots$$

$$H_k = (\mathbf{A}, \boldsymbol{b}^0, \cdots, \boldsymbol{b}^{k-1}, \boldsymbol{u}_k \cdots, \boldsymbol{u}_{\lambda_1+\lambda_2-1})$$

$$\vdots$$

$$H_{\lambda_1+\lambda_2} = (\mathbf{A}, \boldsymbol{b}^0, \cdots, \boldsymbol{b}^{k-1}, \boldsymbol{b}^k \cdots, \boldsymbol{b}^{\lambda_1+\lambda_2-1})$$

Algorithm 2: $\mathcal{A}_{\mathrm{LPN}_{p^{\lambda_1}q^{\lambda_2}}}$, the secret recovery algorithm on LPN over $\mathbb{Z}_{p^{\lambda_1}q^{\lambda_2}}$ ($\lambda_1 + \lambda_2 \geq 2$) with oracle access to $\mathcal{A}_{\mathrm{LPN}_r}$ (the solver for LPN over $\mathbb{F}_r$) and $r \in \{p, q\}$.

Input: $(\mathcal{D}, \mathbb{Z}_{p^{\lambda_1}q^{\lambda_2}})$-LPN$(k, N, t)$ samples $(\mathbf{A}, \boldsymbol{b} = \mathbf{A} \cdot \boldsymbol{s} + \boldsymbol{e} \mod p^{\lambda_1}q^{\lambda_2})$
Output: $\boldsymbol{s} \in \mathbb{Z}_{p^{\lambda_1}q^{\lambda_2}}$

1 $(\mathbf{A}^0, \mathbf{A}^1, \cdots, \mathbf{A}^{\lambda_1+\lambda_2-1}) := \mathsf{DigitDecomp}(\mathbf{A})$;
2 $(\boldsymbol{b}^0, \boldsymbol{b}^1, \cdots, \boldsymbol{b}^{\lambda_1+\lambda_2-1}) := \mathsf{DigitDecomp}(\boldsymbol{b})$;
3 **if** $\lambda_1 >= 1$ **then**
4     $(\boldsymbol{s}^0, \boldsymbol{e}^0) \leftarrow \mathcal{A}_{\mathrm{LPN}_p}(\mathbf{A}^0, \boldsymbol{b}^0)$;
5     $\boldsymbol{b}' := (\boldsymbol{b} - \mathbf{A} \cdot \boldsymbol{s}^0 - \boldsymbol{e}^0)/p$;
6   **Return** $\boldsymbol{s} = \boldsymbol{s}^0 + p \cdot \mathcal{A}_{\mathrm{LPN}_{p^{\lambda_1-1}q^{\lambda_2}}}\left(\mathbf{A}' := \mathbf{A} \mod p^{\lambda_1-1}q^{\lambda_2}, \boldsymbol{b}'\right)$;
7 $(\boldsymbol{s}^0, \boldsymbol{e}^0) \leftarrow \mathcal{A}_{\mathrm{LPN}_q}(\mathbf{A}^0, \boldsymbol{b}^0)$;
8 $\boldsymbol{b}' := (\boldsymbol{b} - \mathbf{A} \cdot \boldsymbol{s}^0 - \boldsymbol{e}^0)/q$;
9 **Return** $\boldsymbol{s} = \boldsymbol{s}^0 + q \cdot \mathcal{A}_{\mathrm{LPN}_{q^{\lambda_2-1}}}\left(\mathbf{A}' := \mathbf{A} \mod q^{\lambda_2-1}, \boldsymbol{b}'\right)$;

---

where every $\boldsymbol{u}_i \leftarrow \mathbb{F}_p^N$ and $\boldsymbol{u}_j \leftarrow \mathbb{F}_q^N$ is sampled independently for $i \in [\lambda_1]$ and $j \in [\lambda_1, \lambda_1+\lambda_2]$. Note that all the $\boldsymbol{s}^k$'s are independent, and by the definition of $\boldsymbol{e} \leftarrow \mathsf{IndBer}_{\mu,N}(\mathbb{Z}_{p^{\lambda_1}q^{\lambda_2}})$ (resp., $\boldsymbol{e} \leftarrow \mathsf{IndHW}_{t,N}(\mathbb{Z}_{p^{\lambda_1}q^{\lambda_2}})$) we have that $\boldsymbol{e}^i$ follows $\mathsf{Ber}_{\mu,N}(\mathbb{F}_p)$ (resp., $\mathsf{HW}_{t,N}(\mathbb{F}_p)$) and $\boldsymbol{e}^j$ follows $\mathsf{Ber}_{\mu,N}(\mathbb{F}_q)$ (resp., $\mathsf{HW}_{t,N}(\mathbb{F}_q)$) given its (independent) prefix $\boldsymbol{e}^0, \cdots, \boldsymbol{e}^{i-1}$ and prefix $\boldsymbol{e}^0, \cdots, \boldsymbol{e}^{j-1}$ for $i \in [\lambda_1]$ and $j \in [\lambda_1, \lambda_1 + \lambda_2]$. Therefore, all the adjacent H_{i-1} and H_i are computationally indistinguishable and so are H_0 and H_λ by a hybrid argument.

The proof of the second statement is essentially similar to that of Theorem 3. Given the LPN over ring $\mathbb{Z}_{p^{\lambda_1}q^{\lambda_2}}$ samples $(\mathbf{A}, \boldsymbol{b} = \mathbf{A} \cdot \boldsymbol{s} + \boldsymbol{e} \mod p^{\lambda_1}q^{\lambda_2})$, we also observe that the samples $(\mathbf{A}^0 := \mathbf{A} \mod r, \boldsymbol{b}^0 := \boldsymbol{b} \mod r)$ constitute exactly the LPN over $\mathbb{F}_r$ samples for noise $\boldsymbol{e}^0 = \boldsymbol{e} \mod r$, where $r \in \{p, q\}$. In case that $\boldsymbol{e} \leftarrow \mathsf{HW}_{t,N}(\mathbb{Z}_{p^{\lambda_1}q^{\lambda_2}})$, the noise vector $\boldsymbol{e} \mod p$ has expected weight $t_p = \frac{(p-1)p^{\lambda_1-1}q^{\lambda_2}}{p^{\lambda_1}q^{\lambda_2}-1} \cdot t$ and the noise vector $\boldsymbol{e} \mod q$ has expected weight $t_q = \frac{(q-1)p^{\lambda_1}q^{\lambda_2-1}}{p^{\lambda_1}q^{\lambda_2}-1} \cdot t$. This implies that with probability $\Omega(1/\sqrt{t})$ the noise vector $\boldsymbol{e} \mod p$ follows the exact-weight distribution $\mathsf{HW}_{t_p,N}(\mathbb{F}_p)$ and the noise vector $\boldsymbol{e} \mod q$ follows the exact-weight distribution $\mathsf{HW}_{t_q,N}(\mathbb{F}_q)$. The proof for the second statement is likewise, except when taking $\boldsymbol{e} \leftarrow \mathsf{Ber}_{\mu,N}(\mathbb{Z}_{p^{\lambda_1}q^{\lambda_2}})$ we get $\boldsymbol{e} \mod r \sim \mathsf{Ber}_{\mu,N}(\mathbb{F}_r)$, where $r \in \{p, q\}$. $\qquad\square$

Similar to the proof of Theorem 8, we give a reduction from the computational LPN problem over a ring $\mathbb{Z}_{p^{\lambda_1}q^{\lambda_2}}$ to that over $\mathbb{F}_p$ and $\mathbb{F}_q$ (see Theorem 11), extending the corresponding reduction between the computational LPN problem over $\mathbb{Z}_{2^\lambda}$ to that over $\mathbb{F}_2$. Algorithm 2 shows how the computational LPN problem over $\mathbb{Z}_{p^{\lambda_1}q^{\lambda_2}}$ is reduced to that over $\mathbb{F}_p$ or $\mathbb{F}_q$ by recursion, whose correctness is analyzed in Lemma 5.

Lemma 5. *Let* $(\mathbf{A}, \boldsymbol{b} = \mathbf{A} \cdot \boldsymbol{s} + \boldsymbol{e} \mod p^{\lambda_1}q^{\lambda_2})$ *be the LPN samples over* $\mathbb{Z}_{p^{\lambda_1}q^{\lambda_2}}$, *then* $(\mathbf{A}', \boldsymbol{b}')$ *as defined in step 6 (resp., step 9) of Algorithm 2 constitute the LPN samples over* $\mathbb{Z}_{p^{\lambda_1-1}q^{\lambda_2}}$ *(resp.,* $\mathbb{Z}_{q^{\lambda_2-1}}$).

Proof. Let $(\boldsymbol{s}^0, \boldsymbol{s}^1, \cdots, \boldsymbol{s}^{\lambda_1+\lambda_2-1}) := \mathsf{DigitDecomp}(\boldsymbol{s})$ and $(\boldsymbol{e}^0, \boldsymbol{e}^1, \cdots, \boldsymbol{e}^{\lambda_1+\lambda_2-1}) := \mathsf{DigitDecomp}(\boldsymbol{e})$. The proof of the statement about $(\mathbf{A}', \boldsymbol{b}')$ as defined in step 6 and step 9 are similar to that of Lemma 4. For the statement about $(\mathbf{A}', \boldsymbol{b}')$ as defined in step 6, it suffices to prove $\boldsymbol{b}' = \mathbf{A}' \cdot \boldsymbol{s}' + \boldsymbol{e}' \mod p^{\lambda_1-1}q^{\lambda_2}$, where $\boldsymbol{s}' = (\boldsymbol{s} - \boldsymbol{s}^0)/p$ and $\boldsymbol{e}' = (\boldsymbol{e} - \boldsymbol{e}^0)/p$ are the secret and noise of the LPN problem over $\mathbb{Z}_{p^{\lambda_1-1}q^{\lambda_2}}$

respectively. That is,

$$
\begin{aligned}
p \cdot \boldsymbol{b}' - p \cdot (\mathbf{A}' \cdot \boldsymbol{s}' + \boldsymbol{e}') &= \boldsymbol{b} - \mathbf{A} \cdot \boldsymbol{s}^0 - \boldsymbol{e}^0 - p \cdot (\mathbf{A}' \cdot \boldsymbol{s}' + \boldsymbol{e}') \\
&= \boldsymbol{b} - \mathbf{A} \cdot \boldsymbol{s}^0 - \boldsymbol{e}^0 - \mathbf{A} \cdot (\boldsymbol{s} - \boldsymbol{s}^0) - \boldsymbol{e} + \boldsymbol{e}^0 \\
&= \boldsymbol{b} - \mathbf{A} \cdot \boldsymbol{s} - \boldsymbol{e} = \boldsymbol{0} \mod p^{\lambda_1} q^{\lambda_2} \ ,
\end{aligned}
$$

where the first equality follows from $\boldsymbol{b}' = (\boldsymbol{b} - \mathbf{A} \cdot \boldsymbol{s}^0 - \boldsymbol{e}^0)/p$, the second is due to $p \cdot \mathbf{A}' \cdot \boldsymbol{s}' = p \cdot \mathbf{A} \cdot \boldsymbol{s}'$ $\mod p^{\lambda_1} q^{\lambda_2}$ and $p \cdot \boldsymbol{e}' = \boldsymbol{e} - \boldsymbol{e}^0$, and the last is by the LPN assumption over $\mathbb{Z}_{p^{\lambda_1 - 1} q^{\lambda_2}}$. For the statement about $(\mathbf{A}', \boldsymbol{b}')$ as defined in step 9, it suffices to prove $\boldsymbol{b}' = \mathbf{A}' \cdot \boldsymbol{s}' + \boldsymbol{e}' \mod q^{\lambda_2 - 1}$, where $\boldsymbol{s}' = (\boldsymbol{s} - \boldsymbol{s}^0)/q$ and $\boldsymbol{e}' = (\boldsymbol{e} - \boldsymbol{e}^0)/q$ are the secret and noise of the LPN problem over $\mathbb{Z}_{q^{\lambda_2 - 1}}$ respectively. That is,

$$
\begin{aligned}
q \cdot \boldsymbol{b}' - q \cdot (\mathbf{A}' \cdot \boldsymbol{s}' + \boldsymbol{e}') &= \boldsymbol{b} - \mathbf{A} \cdot \boldsymbol{s}^0 - \boldsymbol{e}^0 - q \cdot (\mathbf{A}' \cdot \boldsymbol{s}' + \boldsymbol{e}') \\
&= \boldsymbol{b} - \mathbf{A} \cdot \boldsymbol{s}^0 - \boldsymbol{e}^0 - \mathbf{A} \cdot (\boldsymbol{s} - \boldsymbol{s}^0) - \boldsymbol{e} + \boldsymbol{e}^0 \\
&= \boldsymbol{b} - \mathbf{A} \cdot \boldsymbol{s} - \boldsymbol{e} = \boldsymbol{0} \mod q^{\lambda_2} \ ,
\end{aligned}
$$

where the first equality follows from $\boldsymbol{b}' = (\boldsymbol{b} - \mathbf{A} \cdot \boldsymbol{s}^0 - \boldsymbol{e}^0)/q$, the second is due to $q \cdot \mathbf{A}' \cdot \boldsymbol{s}' = q \cdot \mathbf{A} \cdot \boldsymbol{s}'$ $\mod q^{\lambda_2}$ and $q \cdot \boldsymbol{e}' = \boldsymbol{e} - \boldsymbol{e}^0$, and the last is by the LPN assumption over $\mathbb{Z}_{q^{\lambda_2 - 1}}$. □

Theorem 11. *If computational* $(\mathsf{Ber}, \mathbb{F}_p)\text{-}\mathsf{LPN}(N, k, \mu)$ *and* $(\mathsf{Ber}, \mathbb{F}_q)\text{-}\mathsf{LPN}(N, k, \mu)$ *problems can be broken with probability at least* $(1 - \epsilon)$ *in time* T *respectively, then the computational* $(\mathsf{Ber}, \mathbb{Z}_{p^{\lambda_1} q^{\lambda_2}})\text{-}\mathsf{LPN}(N, k, \mu)$ *problem can be broken in time* $(\lambda_1 + \lambda_2)T + \mathsf{poly}(k, N)$ *with success probability at least* $1 - (\lambda_1 + \lambda_2 + 2)\sqrt{\epsilon}$.

Proof. Algorithm 2 translates an $(\mathsf{Ber}, \mathbb{Z}_{p^{\lambda_1} q^{\lambda_2}})\text{-}\mathsf{LPN}(N, k, \mu)$ instance into λ_1 $(\mathsf{Ber}, \mathbb{F}_p)\text{-}\mathsf{LPN}(N, k, \mu)$ instances and λ_2 $(\mathsf{Ber}, \mathbb{F}_q)\text{-}\mathsf{LPN}(N, k, \mu)$ instances, which are independent except that all $(\mathsf{Ber}, \mathbb{F}_p)\text{-}\mathsf{LPN}(N, k, \mu)$ instances share the same random matrix $\mathbf{A} \mod p$, all $(\mathsf{Ber}, \mathbb{F}_q)\text{-}\mathsf{LPN}(N, k, \mu)$ instances share the same random matrix $\mathbf{A} \mod q$ and that the noise vectors of the $\lambda_1 + \lambda_2$ instances are somehow correlated.

For contradiction assume that there exists an algorithm $\mathcal{A}_{\mathsf{LPN}_p}$ (resp., an algorithm $\mathcal{A}_{\mathsf{LPN}_q}$) that recovers the secret of LPN over $\mathbb{F}_p$ (resp., the secret of LPN over $\mathbb{F}_q$) with probability more than 2ϵ within time T. Similar to the proof of Theorem 6, we consider distribution $\mathsf{Ber}_{\mu, N}(\mathbb{F}_r)$, for $r \in \{p, q\}$, being sampled in two steps: first pick each coordinate with probability μ independently (and let the rest with 0's), and second assign the picked coordinates with uniform random field element.

We have by a Markov inequality that for at least a $(1 - \sqrt{\epsilon})$ fraction of $(\mathbf{A} \mod p, \mathsf{coin}(\boldsymbol{e}^i))$, $\mathcal{A}_{\mathsf{LPN}_p}$ recovers $\boldsymbol{s}^i$ (for $i \in [\lambda_1]$) from $(\mathbf{A} \mod p, (\mathbf{A} \mod p) \cdot \boldsymbol{s}^i + \boldsymbol{e}^i)$ with probability at least $1 - \sqrt{\epsilon}$. Thus, the secret of all $(\mathsf{Ber}, \mathbb{F}_p)\text{-}\mathsf{LPN}(N, k, \mu)$ instances can be recovered with the following probability by a union bound

$$
(1 - \sqrt{\epsilon})(1 - \lambda_1 \sqrt{\epsilon}) \geq 1 - (\lambda_1 + 1)\sqrt{\epsilon} \ .
$$

The proof about the $(\mathsf{Ber}, \mathbb{F}_q)\text{-}\mathsf{LPN}(N, k, \mu)$ instances is likewise, and we have that the secrets of all $(\mathsf{Ber}, \mathbb{F}_q)\text{-}\mathsf{LPN}(N, k, \mu)$ instances can be recovered with the following probability by a union bound

$$
(1 - \sqrt{\epsilon})(1 - \lambda_2 \sqrt{\epsilon}) \geq 1 - (\lambda_2 + 1)\sqrt{\epsilon} \ .
$$

Overall, $\mathcal{A}_{\mathsf{LPN}_{p^{\lambda_1} q^{\lambda_2}}}$ succeeds with probability at least $1 - (\lambda_1 + \lambda_2 + 2)\sqrt{\epsilon}$ by a union bound. □

B Concrete Analysis for Low-Noise Exact-LPN Problems

In Tables 6 and 7, we compare our analysis with the previous work by Boyle et al. [BCGI18] for the costs of Pooled Gauss, SD and ISD attacks to solve exact-LPN problems. From these tables, we can see that the

Exact LPN over a field $\mathbb{F}$			This work $(\log|\mathbb{F}| = 128)$				This work $(\log|\mathbb{F}| = 1)$				[BCGI18] (Any field size)		
N	k	t	Gauss	SD	SD 2.0	ISD	Gauss	SD	SD 2.0	ISD	Gauss	SD	ISD
2^{10}	652	57	111	184	184	111	111	194	116	94	80	93	115
2^{12}	1589	98	100	151	151	100	100	154	137	83	85	80	104
2^{14}	3482	198	101	149	149	101	101	150	144	86	94	80	108
2^{16}	7391	389	103	147	147	103	103	148	146	91	99	80	112
2^{18}	15336	760	105	146	146	105	105	146	146	95	103	80	117
2^{20}	32771	1419	107	145	145	107	107	145	145	99	106	80	121
2^{22}	67440	2735	108	144	144	108	108	144	144	104	108	80	126

Table 6: The comparison between our analysis and [BCGI18] for the costs of different attacks to solve an LPN problem with dimension k, number of samples N, Hamming weight of noises t for an exact noise distribution.

Exact dual-LPN over a field $\mathbb{F}$			This work $(\log|\mathbb{F}| = 128)$				This work $(\log|\mathbb{F}| = 1)$				[BCGI18] (Any field size)		
n	N	t	Gauss	SD	SD 2.0	ISD	Gauss	SD	SD 2.0	ISD	Gauss	SD	ISD
2^{10}	2^{12}	44	117	189	189	117	117	191	170	97	80	100	117
2^{12}	2^{14}	39	111	170	169	111	111	170	166	95	80	92	112
2^{14}	2^{16}	34	107	151	151	107	107	151	151	93	80	84	107
2^{16}	2^{18}	32	108	145	145	108	108	145	145	95	84	82	109
2^{18}	2^{20}	31	112	143	143	112	112	143	143	99	88	82	112
2^{20}	2^{22}	30	116	141	141	116	116	141	141	103	93	82	116
2^{22}	2^{24}	29	119	139	139	119	119	139	139	107	97	82	120

Table 7: The comparison of our analysis and [BCGI18] about the costs of different attacks to solve a dual-LPN problem with dimension $N - n = 3N/4$, number of samples N, Hamming weight of noises t for an exact noise distribution.

LPN parameters given in [BCGI18] actually guarantee higher security, based on our more accurate analysis. Besides, in Table 8, we show the costs of different attacks against regular-LPN problems that have the same parameters as [BCGI18], except that larger noise weights are used. This table provides us with a hint, i.e., increasing the weight t may let the algebraic attack have better attack advantage, and it is not a good choice to strengthen the security of regular-LPN problems via increasing the weight t. As described in Section 5.1, it is better to increase the dimension k and keep t unchanged, so as to strengthen the regular-LPN instances to the 128-bit security.

For analyzing the cost of low-noise LPN problems with exact noise distributions, we focus on Pooled Gauss, SD and ISD attacks. Our analysis does not cover the BKW attack [BKW00] and its subsequent improvements (e.g., [Lyu05, EHK+18, DEM19, LY22]) as well as the combinations of Pooled Gauss, ISD and BKW attacks [EKM17], since all these attacks either require a sub-exponential number of samples that is *not* satisfied in the LPN-based protocols under the PCG framework, or take significantly more time than the Pooled Gauss, SD and ISD attacks for solving LPN with low noise rate. In the following subsections, for low-noise (dual)-LPN problems with exact noise distributions, we will show the imprecisions of the previous analysis [BCGI18] and provide the MPC and ZK communities with more accurate formulas and an estimator tool.

Regular LPN with larger noise weight			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	106	194	351	350	194	179	176	350	194	159	145
2^{12}	1589	172	155	261	261	155	150	143	245	216	121	135
2^{14}	3482	338	150	247	247	150	150	140	229	218	122	138
2^{16}	7391	667	151	244	244	151	150	141	225	220	125	139
2^{18}	15336	1312	153	242	242	153	133	143	224	221	129	122
2^{20}	32771	2467	155	241	241	155	131	146	224	223	135	125
2^{22}	67440	4788	152	231	231	156	110	144	215	214	135	103

Table 8: The bit-security of LPN problems over finite fields with number of samples N, dimension k and larger weight t for regular noise distributions. The abbreviation "AGB" denotes the recent algebraic attack [BØ23].

B.1 The Hardness of LPN against Pooled Gauss Attack

Pooled Gauss. Gauss attack is a natural extension of Gaussian elimination to recover the secret vector from an LPN instance with a Bernoulli distribution. This attack guesses a fresh batch of k non-noisy LPN samples by picking them at random in each iteration, inverts the corresponding submatrix, computes a candidate secret s', and then verifies whether s' is correct or not. However, when considering the concrete LPN parameters, the number of samples required by this attack is *not* achieved. To reduce the number of samples, Esser, Kübler and May [EKM17] introduced *Pooled Gauss* attack, which guesses k non-noisy samples by picking them at random from a pool of the *fixed* $N = k^2 \log^2 k$ LPN samples in each iteration, and then inverts the corresponding subsystem to get a candidate vector s' and verifies if s' is correct. For LPN with noise rate r, this attack recovers the secret in time $\frac{k^3 \log^2 k}{(1-r)^k}$ using $k^2 \log^2 k$ samples. As pointed out in [EKM17], Pooled Gauss attack solves the LPN problem via finding a non-noisy index set, which is also called an information set. Therefore, we can view Pooled Gauss as a special case of the information set decoding (ISD) algorithm, particularly Pooled Gauss resembles the well-known algorithm of Prange [Pra62].

Previous analysis. For solving the problems of LPN and dual-LPN with exact noise distributions, Boyle et al. [BCGI18] gave a formula to compute the cost of Pooled Gauss attack by simplifying the underlying noise distribution $\mathsf{HW}_{t,N}(\mathbb{F})$ to $\mathsf{Ber}_{t/N,N}(\mathbb{F})$. Below, we discuss why the concrete security under $\mathsf{Ber}_{t/N,N}(\mathbb{F})$-noise is significantly lower than that under $\mathsf{HW}_{t,N}(\mathbb{F})$-noise. Furthermore, we give a more accurate formula to estimate the cost under Pooled Gauss attack, and discuss the difference between the previous analysis [BCGI18] and our analysis.

One may consider that the concrete security of $\mathsf{HW}_{t,N}(\mathbb{F})$-LPN is roughly the same as $\mathsf{Ber}_{t/N,N}(\mathbb{F})$-LPN, since a noise $\mathbf{e} \leftarrow \mathsf{Ber}_{t/N,N}(\mathbb{F})$ follows the exact noise distribution $\mathsf{HW}_{t,N}(\mathbb{F})$ conditioned on $|\mathbf{e}| = t$, which occurs with noticeable probability (see Lemma 1) if $|\mathbb{F}| > N$. However, this reduction gives only an upper bound on the security of $\mathsf{Ber}_{t/N,N}(\mathbb{F})$-LPN, which is not tight for concrete costs. To see this, take the dual-LPN parameter $(n = 2^{10}, N = 2^{12}, t = 44)$ from [BCGI18] as an example. We denote by $T(\mathsf{Ber}, \mu, p)$ and $T(\mathsf{HW}, t, p)$ the bit-security of dual-LPN with Bernoulli noise of rate μ and exact noise of weight-t respectively, with successful probability $\geq p$. The above reduction translates to $T(\mathsf{Ber}, t/N, 0.06) \leq T(\mathsf{HW}, t = 44, 0.99)$, where $\Pr[|\mathbf{e}| = 44] \approx 6\%$ for $\mathbf{e} \leftarrow \mathsf{Ber}_{\mu,N}$ and $\mu = t/N = 11/1024$. For a tighter bound on $T(\mathsf{Ber}, t/N, p)$, we consider collectively the weights up to a smaller threshold, e.g, $\Pr[|\mathbf{e}| \leq 35] \approx 10\%$, $\Pr[|\mathbf{e}| \leq 33] \approx 5\%$, and thus

$$T(\mathsf{Ber}, t/N, 0.1) \leq T(\mathsf{HW}, 35, 0.99) \text{ and } T(\mathsf{Ber}, t/N, 0.05) \leq T(\mathsf{HW}, 33, 0.99).$$

Therefore, when evaluating the bit-security for dual-LPN with exact noise weight $t = 44$ (i.e., $T(\mathsf{HW}, 44, 0.99)$), [BCGI18] simply used $e \leftarrow \mathsf{Ber}_{t/N,N}$ instead of $e \leftarrow \mathsf{HW}_{t,N}$, and hence what they obtained is more close to $T(\mathsf{HW}, 35, 0.99)$ or even $T(\mathsf{HW}, 33, 0.99)$. According to our estimator, the bit security w.r.t. $T(\mathsf{HW}, w, 0.99)$ for $w = 44$, $w = 35$ and $w = 33$ is 117, 98 and 94 respectively. This explains the concrete security gap between using Bernoulli and exact noise distributions.

Our analysis. To give a precise formula, we extend Pooled Gauss attack [EKM17] from solving a (dual-)LPN problem from a Bernoulli distribution to solving that with an exact noise distribution. Specifically, the Pooled Gauss attack performs as follows:

- Given a $(\mathsf{HW}, \mathbb{F})\text{-LPN}(N, k, t)$ instance $\boldsymbol{b} = \mathbf{A} \cdot \boldsymbol{s} + \boldsymbol{e}$, for each iteration, guess k non-noisy coordinates of vector $\boldsymbol{b}$ by sampling them at random, and obtain a length-k vector $\boldsymbol{b}'$ and $k \times k$ matrix $\mathbf{A}'$. Then, compute $\boldsymbol{s}' := (\mathbf{A}')^{-1} \cdot \boldsymbol{b}'$ and verify whether $\boldsymbol{s}'$ is correct or not by running the test algorithm in [EKM17].

- Given a $(\mathsf{HW}, \mathbb{F})\text{-dual-LPN}(N, n, t)$ instance $\boldsymbol{b} = \mathbf{H} \cdot \boldsymbol{e}$, for each iteration, guess n coordinates of vector $\boldsymbol{e}$ by sampling them uniformly at random such that these coordinates contain t noisy coordinates of $\boldsymbol{e}$, choose the corresponding $n \times n$ sub-matrix $\mathbf{H}'$ from $\mathbf{H}$ according to the n coordinates, compute $\boldsymbol{e}' := (\mathbf{H}')^{-1} \cdot \boldsymbol{b}$, and then checks if $|\boldsymbol{e}'| = t$.

The above attack uses the fixed N samples for both LPN and dual-LPN. For solving LPN, Pooled Gauss attack runs in time $\frac{\binom{N}{t}}{\binom{N-k}{t}} \cdot k^{2.8}$, where $\frac{\binom{N-k}{t}}{\binom{N}{t}}$ is the probability of guessing successfully in one iteration, and $k^{2.8}$ is the cost of inverting matrix $\mathbf{A}'$ via Strassen's algorithm. For solving dual-LPN, Pooled Gauss attack runs in time $\frac{\binom{N}{t}}{\binom{N-k}{t}} \cdot (N-k)^{2.8}$, where $k = N - n$ is the dimension and $(N-k)^{2.8}$ is the cost of inverting matrix $\mathbf{H}'$ via Strassen's algorithm. As LPN can be efficiently transformed into dual-LPN and vice versa, Pooled Gauss attack solves a (dual-)LPN problem with number of samples N, dimension k and weight of noises t in time $\frac{\binom{N}{t}}{\binom{N-k}{t}} \cdot \min(k^{2.8}, (N-k)^{2.8})$. Therefore, the bit-security of a (dual-)LPN instance with respect to Pooled Gauss attack is computed as

$$\log\left(\min(k^{2.8}, (N-k)^{2.8})\right) + \log\binom{N}{t} - \log\binom{N-k}{t} .$$

For the LPN problem with an exact noise distribution, [BCGI18] simplified the cost of Pooled Gauss attack as $(\frac{1}{1-t/N})^k \cdot k^{2.8}$. In the following, we compare the attack cost in [BCGI18] and our estimate by computing their ratio:

$$\frac{T_{\text{ours}}}{T_{[\text{BCGI18}]}} \approx \left(1 - \frac{t}{N}\right)^k \cdot \frac{\binom{N}{t}}{\binom{N-k}{t}} \geq \left(1 - \frac{t}{N}\right)^k \cdot \left(\frac{N}{N-k}\right)^t \approx e^{tk \cdot \left(\frac{1}{N-k} - \frac{1}{N}\right)} = e^{\frac{t \cdot k^2}{N \cdot (N-k)}},$$

where $\approx$ denotes an approximate relation that omits a polynomial factor. For the (dual-)LPN parameters N, k used in known PCG-like protocols (e.g., $k = 0.75N$ considered in [BCGI18]), the cost of Pooled Gauss attack estimated by Boyle et al. [BCGI18] is about $2^{O(t)}$ times larger than our estimate.

B.2 The Hardness of LPN against SD Attack

Statistical decoding (SD). While Gauss and ISD attacks recover the secret vector $\boldsymbol{s}$, the SD attack [Al 01, Ove06, FKI07, DT17] (a.k.a., low-weight parity-check attack in [BCGI18, BCG+19a, BCG+19b]) distinguishes LPN samples $(\mathbf{A}, \boldsymbol{b} = \mathbf{A} \cdot \boldsymbol{s} + \boldsymbol{e})$ over $\mathbb{F}$ from random samples $(\mathbf{A}, \boldsymbol{u})$. The core of this attack is to find a set of parity-check vectors

$$\mathcal{V} \subset \{\boldsymbol{v} \mid \boldsymbol{v} \cdot \mathbf{A} = \mathbf{0} \text{ and } |\boldsymbol{v}| = w \text{ with a sufficiently small } w > 0\} .$$

While $\Pr[\langle \boldsymbol{v}, \boldsymbol{u} \rangle = 0] = 1/|\mathbb{F}|$ for random samples, $\Pr[\langle \boldsymbol{v}, \boldsymbol{b} \rangle = \langle \boldsymbol{v}, \boldsymbol{e} \rangle = 0] = 1/|\mathbb{F}| + \epsilon$ with some $\epsilon > 0$ for LPN samples, since both Hamming weights of vectors $\boldsymbol{e}$ and $\boldsymbol{v}$ are small. For each $\boldsymbol{v} \in \mathcal{V}$, this attack can compute a vote $\langle \boldsymbol{v}, \boldsymbol{y} \rangle$ with either $\boldsymbol{y} = \boldsymbol{b}$ or $\boldsymbol{y} = \boldsymbol{u}$. By repeating the process $|\mathcal{V}| = 1/\epsilon^2$ times, this attack outputs a majority of votes indicating whether $\boldsymbol{y} = \boldsymbol{b}$ or $\boldsymbol{y} = \boldsymbol{u}$.

Recently, by introducing fast Fourier transform (FFT), Carrier, Debris-Alazard, Meyer-Hilfiger and Tillich [CDHT22] proposed the SD 2.0 algorithm, which improves the SD algorithm considerably and even outperforms the ISD algorithm for the case that the noise rate achieves the GV bound (defined in Footnote 8) and code rate $k/N < 0.3$. The SD 2.0 algorithm [CDHT22] invokes the following two techniques to reduce the time complexity.

1. Use FFT to perform the majority voting, achieving smaller ϵ.

2. Use the collision technique used in the ISD algorithms [Ste88, Dum91, MMT11, BJMM12] to find a set $\mathcal{V}$ with smaller w.

The computation complexity of the second step is $|\mathbb{F}|^{\theta(k)}$, which has no advantage in analyzing low-noise LPN problems used in the PCG setting (see Tables 6 and 7 for examples). Below, we will prove that the SD 2.0 attack (adapted to the low-noise setting) requires more cost than the Prange's ISD algorithm [Pra62] for analyzing low-noise LPN over large fields.

Previous analysis. Boyle et al. [BCGI18] analyzed the cost of solving LPN problems against SD attack, and shown that this attack performs the best among three kinds of attacks for their parameters selection. In the following, we show two imprecisons for their analysis [BCGI18]. First, Boyle et al. [BCGI18] assumes that each parity-check vector $\boldsymbol{v} \in \mathcal{V}$ independently follows a Bernoulli distribution $\mathsf{Ber}_{w/N,N}(\mathbb{F})$ where $w \in \mathbb{N}$ is the Hamming weight of $\boldsymbol{v}$, which is inaccurate [10] [DT17]. To obtain more accurate complexity of this attack, a weaker assumption was proposed by [DT17] where each parity-check vector $\boldsymbol{v} \in \mathcal{V}$ is assumed to be independently in compliance with an exact noise distribution $\mathsf{HW}_{w,N}(\mathbb{F})$. Second, Boyle et al. [BCGI18] underestimated the cost of this attack as $T = T_1/\epsilon$, where T_1 is the time of finding one parity check vector $\boldsymbol{v} \in \mathcal{V}$ and ϵ is the distinguishing advantage for one vote. The SD attack solves the decisional LPN problem with negligible advantage in time T, while other attacks solve the LPN problem with constant advantage. Following the previous works [Al01, Ove06, FKI07, DT17], this attack takes time $T = T_1/\epsilon^2$ to distinguish LPN samples from random samples with constant advantage. [11]

Our analysis. We can view the traditional SD attack as a special case of the SD 2.0 attack. Thus, we focus on analyzing the cost of SD 2.0 and giving its formula for low-noise LPN. We adapt SD 2.0 to analyze low-noise LPN by using the collision technique [Zic17, BCGI18], and also generalize it from $\mathbb{F}_2$ to any finite field $\mathbb{F}$. Given either a $(\mathsf{HW}, \mathbb{F})$-LPN$(N, k, t)$ instance $\boldsymbol{b} = \mathbf{A} \cdot \boldsymbol{s} + \boldsymbol{e}$ or a random instance $\boldsymbol{u} \xleftarrow{\$} \mathbb{F}^N$, the adapted SD 2.0 algorithm introduces a new parameter s, and proceeds in the following two stages.

1. Using the collision technique [Zic17, BCGI18], we aim to identify a set of parity-check vectors

$$\mathcal{V} \subset \left\{ \boldsymbol{v} \stackrel{\text{def}}{=} (\boldsymbol{v}_1, \boldsymbol{v}_2) \,\big|\, \boldsymbol{v}_1 \in \mathbb{F}^s, \, \boldsymbol{v} \cdot \mathbf{A} = \mathbf{0} \text{ and } |\boldsymbol{v}_2| = w \right\},$$

where $w > 0$ is sufficiently small. It's essential to note that for each $\boldsymbol{v}$, we can express $\langle \boldsymbol{v}, \boldsymbol{b} \rangle$ as $\langle \boldsymbol{v}, \mathbf{A} \cdot \boldsymbol{s} + \boldsymbol{e} \rangle = \langle \boldsymbol{v}, \boldsymbol{e} \rangle = \langle \boldsymbol{v}_1, \boldsymbol{e}_1 \rangle + \langle \boldsymbol{v}_2, \boldsymbol{e}_2 \rangle$, where $\boldsymbol{e} \stackrel{\text{def}}{=} (\boldsymbol{e}_1, \boldsymbol{e}_2)$, $\boldsymbol{e}_1 \in \mathbb{F}^s$ and $\boldsymbol{e}_2 \in \mathbb{F}^{N-s}$. The SD 2.0

---

[10]The Bernoulli distribution admits a slackness event that the weight of a parity-check vector $\boldsymbol{v}$ goes much below the expected w leading to underestimate the attack cost. However, for an optimal weight w, such low-weight vectors $\boldsymbol{v}$'s violating the Gilbert–Varshamov bound may not exist at all.

[11]We shall distinguish the differences between $1/\epsilon$ and $1/\epsilon^2$. If a single *key-recovery* attack succeeds with probability ϵ, then repeating roughly $1/\epsilon$ independent instances achieves constant (or even overwhelming) success probability. In contrast, if a single *distinguishing attack* gains advantage ϵ, then the number of independent votes needed to amplify the advantage to constant is about $1/\epsilon^2$ by a Chernoff bound.

algorithm [CDHT22] makes the assumption that $\langle v_2, e_2 \rangle$ independently follows a Bernoulli distribution $\mathsf{Ber}_{\epsilon,1}(\mathbb{F})$ with some $\epsilon > 0$ for each $v \in \mathcal{V}$. Very recently, Meyer-Hilfiger and Tillich [MHT23] shown that the strong independence assumption can be replaced with a very mild assumption, and modified the SD 2.0 algorithm to obtain the same complexity under the mild assumption. In this paper, we focus on the cost of SD 2.0 instead of the underlying assumption, and thus follow the original SD 2.0 algorithm to give a simpler description.

2. Use FFT, if an LPN instance is given, then the ratio on the number of zero entries in the following set

$$\left\{ \langle v, b \rangle - \langle v_1, e_1 \rangle \,\middle|\, (v_1, v_2) \in \mathcal{V} \text{ and } v_1 \in \mathbb{F}^s \right\}$$

is greater than $\epsilon/2 + 1/|\mathbb{F}|$. Otherwise, the ratio for the following set

$$\left\{ \langle v, u \rangle - \langle v_1, e_1 \rangle \,\middle|\, (v_1, v_2) \in \mathcal{V} \text{ and } v_1 \in \mathbb{F}^s \right\}$$

is at most $\epsilon/2 + 1/|\mathbb{F}|$. Note that v and u are independent.

An appropriate value for parameter s is chosen so that the two stages take almost the same cost. In the following, we analyze the SD 2.0 cost of solving low-noise LPN by considering two cases: binary field $\mathbb{F}_2$ and larger fields.

Binary field $\mathbb{F}_2$. Given a $(\mathsf{HW}, \mathbb{F})\text{-}\mathsf{LPN}(N, k, t)$ instance, the adapted SD 2.0 algorithm finds a parity-check vector set with Hamming weight $w = w(s)$. According to [Lyu05, Lemma 3], we have that

$$\epsilon_s = \Pr[\langle v_2, e_2 \rangle = 0] - 1/|\mathbb{F}| = \frac{1}{2} \left(\frac{N - 2w - t + 1}{N - t + 1} \right)^t ,$$

and the time complexity is $T = \min_s \left(T_1 \cdot (1/\epsilon_s)^2 + s \cdot 2^s \right)$, where $w = (k + 1 - s)/2$ and $T_1 = k + 1$ following the work [BCGI18].

Larger fields. For fields of size $|\mathbb{F}| \geq 4t$, we adapt the SD 2.0 algorithm [CDHT22] to analyze the cost of solving the $(\mathsf{HW}, \mathbb{F})\text{-}\mathsf{LPN}(N, k, t)$ problem. In particular, we have the following theorem.

Theorem 12. *For $w = w(s) \in \mathbb{N}$ and a finite field $\mathbb{F}$ with $|\mathbb{F}| \geq 4t$, the adapted SD 2.0 algorithm solves the decisional $(\mathsf{HW}, \mathbb{F})\text{-}\mathsf{LPN}(N, k, t)$ problem in time*

$$T = \min_s \left(T_1 \cdot \left(\frac{\binom{N}{t}}{\binom{N-w}{t}} \cdot \frac{2|\mathbb{F}|}{|\mathbb{F}| - 1} \right)^2 + s \cdot |\mathbb{F}|^s \cdot \log |\mathbb{F}| \right),$$

with constant advantage, where T_1 is the time of finding one parity-check vector.

Proof. We first analyze the time complexity with a slightly different noise distribution $\mathsf{HW}'_{t,N}(\mathbb{F})$ (to be defined below). Then, we show when switching from $\mathsf{HW}'_{t,N}(\mathbb{F})$ back to $\mathsf{HW}_{t,N}(\mathbb{F})$, the distinguishing advantage only diminishes slightly, but it remains constant, all while maintaining the same time complexity. Specifically, we define $\mathsf{HW}'_{t,N}(\mathbb{F})$ as follows:

1. Sample t out of N positions uniformly at random, and set the rest $N - t$ positions as 0.

2. For each noisy coordinate picked in the previous step, sample a random element in $\mathbb{F}$.

Let $\boldsymbol{b} = \mathbf{A} \cdot \boldsymbol{s} + \boldsymbol{e}'$ with $\boldsymbol{e}' \leftarrow \mathsf{HW}'_{t,N}(\mathbb{F})$. Let $\boldsymbol{v}'$ be a parity-check vector in the set $\mathcal{V}$, which can be found in time T_1 using the techniques [Zic17, BCGI18]. We denote by E the event that there is no intersection between the set of non-zero coordinates in vector $\boldsymbol{v}'$ and that of the noisy coordinates of $\boldsymbol{e}'$. Then, we have the following:

$$
\begin{aligned}
\Pr[\langle \boldsymbol{v}', \boldsymbol{b} \rangle = 0] &= \Pr[\langle \boldsymbol{v}', \boldsymbol{e}' \rangle = 0] \\
&= \Pr[\langle \boldsymbol{v}', \boldsymbol{e}' \rangle = 0 \,|\, E] \cdot \Pr[E] + \Pr[\langle \boldsymbol{v}', \boldsymbol{e}' \rangle = 0 \,|\, \neg E] \cdot \Pr[\neg E] \\
&= \Pr[E] + \frac{1}{|\mathbb{F}|} \cdot \Pr[\neg E] = \frac{\binom{N-w}{t}}{\binom{N}{t}} + \frac{1}{|\mathbb{F}|} \cdot \left(1 - \frac{\binom{N-w}{t}}{\binom{N}{t}} \right).
\end{aligned}
$$

The SD 2.0 algorithm performs maximum likelihood decoding for $1/\epsilon^2$ votes, which takes time $O(s|\mathbb{F}|^s \log |\mathbb{F}|)$ using FFT. Together with $\epsilon = (\Pr[\langle \boldsymbol{v}', \boldsymbol{b} \rangle = 0] - 1/|\mathbb{F}|)/2 = \frac{\binom{N-w}{t}}{\binom{N}{t}} \cdot \frac{|\mathbb{F}|-1}{2|\mathbb{F}|}$, the adapted SD 2.0 algorithm solves the decisional $(\mathsf{HW}', \mathbb{F})$-$\mathsf{LPN}(N, k, t)$ problem with advantage $\geq \frac{1}{2}$ in time

$$
T = \min_s \left(T_1 \cdot \left(\frac{\binom{N}{t}}{\binom{N-w}{t}} \cdot \frac{2|\mathbb{F}|}{|\mathbb{F}|-1} \right)^2 + s \cdot |\mathbb{F}|^s \cdot \log |\mathbb{F}| \right).
$$

The statistical distance between $\mathsf{HW}_{t,N}(\mathbb{F})$ and $\mathsf{HW}'_{t,N}(\mathbb{F})$ is bounded by

$$
\mathsf{SD}(\mathsf{HW}_{t,N}, \mathsf{HW}'_{t,N}) \leq \mathsf{SD}(\mathcal{U}_{(\mathbb{F} \setminus \{0\})^t}, \mathcal{U}_{\mathbb{F}^t}) = 1 - (1 - \frac{1}{|\mathbb{F}|})^t \leq \frac{t}{|\mathbb{F}|} \leq \frac{1}{4},
$$

where $\mathcal{U}_{\mathcal{R}}$ denotes the uniform distribution over $\mathcal{R}$. Overall, the adapted SD 2.0 algorithm solves the decisional $(\mathsf{HW}, \mathbb{F})$-$\mathsf{LPN}(N, k, t)$ problem with advantage at least $1/4$. $\qquad\square$

We set $w = k - s + 1$ and $T_1 = k + 1$ following the work [BCGI18]. For the case of $2 \leq |\mathbb{F}| < 4t$, we can still use the above formula to estimate the cost of the adapted SD 2.0 algorithm, which is smaller than the actual cost and makes our analysis more conservative.

Debris-Alazard and Tillich [DT17] shown that an optimal SD algorithm requires more cost than the Prange's ISD algorithm [Pra62] for solving LPN over $\mathbb{F}_2$ with noise rate achieving the GV bound. Below, we show that for solving low-noise LPN problems over large fields, the SD 2.0 algorithm always takes more cost than the Prange's ISD algorithm [Pra62]. Note that the traditional SD attack always has more cost than the (adapted) SD 2.0 attack. In the following theorem, we assume that LPN problems adopt random linear codes as in [DT17].

Theorem 13. *For any* $(\mathsf{HW}, \mathbb{F})$-$\mathsf{LPN}(N, k, t)$ *problem with* $|\mathbb{F}| = k^{\omega(1)}$, $(1 + \beta)k \leq N = \mathsf{poly}(k)$ *for a constant* $\beta > 0$ *and* $t = o(N)$, *the adapted SD* 2.0 *attack requires more cost than the Prange's ISD algorithm.*

Proof. For the case of $s = \Omega(k)$, it trivially holds, since the SD 2.0 algorithm takes at least $|\mathbb{F}|^s$, which is much greater than the time complexity $2^{o(k)}$ of the Prange's ISD algorithm. Hence, we only need to consider $s = o(k)$. The SD 2.0 algorithm needs to find a set of parity-check vectors

$$
\mathcal{V} \subset \left\{ \boldsymbol{v} \stackrel{\text{def}}{=} (\boldsymbol{v}_1, \boldsymbol{v}_2) \,\middle|\, \boldsymbol{v}_1 \in \mathbb{F}^s, \boldsymbol{v} \cdot \mathbf{A} = \boldsymbol{0} \text{ and } |\boldsymbol{v}_2| = w \right\},
$$

with a sufficiently small $w > 0$. The expected number of such parity-check vectors is $\mathbb{E}[|\mathcal{V}|] \leq \binom{N-s}{w} \cdot |\mathbb{F}|^{w-k+s}$. We need $\mathbb{E}[|\mathcal{V}|] \geq 1$ to guarantee existence. In particular, we have the following:

$$
2^{w \cdot \log N + (w-k+s) \cdot \log |\mathbb{F}|} = N^w \cdot |\mathbb{F}|^{w-k+s} \geq \binom{N-s}{w} \cdot |\mathbb{F}|^{w-k+s} \geq 1.
$$

Together with $|\mathbb{F}| = k^{\omega(1)}$ and $N = \mathsf{poly}(k)$, we have that $w \geq (1 - o(1))k$. Since the cost of SD 2.0 increases as w increases, we set $w = (1 - o(1))k$ for an optimal attack in the SD 2.0 framework and we conservatively assume that the time of finding a parity-check vector with minimal weight w is at least the time using the Gaussian elimination method (denoted by T_{Gauss}). [12] According to Theorem 12, we have that the adapted SD 2.0 algorithm solves the LPN problem with constant algorithm in time

$$T = \min_s \left(T_{\mathrm{Gauss}} \cdot \left(\frac{\binom{N}{t}}{\binom{N-w}{t}} \cdot \frac{2|\mathbb{F}|}{|\mathbb{F}| - 1} \right)^2 + s \cdot |\mathbb{F}|^s \cdot \log |\mathbb{F}| \right) .$$

The cost of the Prange's ISD algorithm is upper bounded by $T_{\mathrm{Gauss}} \cdot \frac{\binom{N}{t}}{\binom{N-k}{t}}$ (see, e.g., [BCGI18, HOSS18]). The cost-ratio between the optimal attack in the SD 2.0 framework and the Prange's ISD attack is greater than

$$\frac{\left(\frac{\binom{N}{t}}{\binom{N-w}{t}} \right)^2}{\frac{\binom{N}{t}}{\binom{N-k}{t}}} = \frac{\binom{N}{t} \cdot \binom{N-k}{t}}{\binom{N-w}{t}^2} = \prod_{i=1}^t \frac{(N - t + i) \cdot (N - k - t + i)}{(N - w - t + i)^2} . \tag{2}$$

For $N \geq (1 + \beta)k$, $t = o(N)$ and $w = (1 - o(1))k$, we have that $w(2N - 2t - w) \geq k(N - t)$. It follows that the above value in (2) is greater than 1. $\qquad\square$

B.3 The Hardness of LPN against ISD Attack

Information set decoding (ISD). Solving LPN is equivalent to solving its dual variant, which is able to be interpreted as the task of decoding a linear code from its syndrome. To address this challenge, Prange's ISD algorithm [Pra62] seeks to identify a subset of size t from the rows of the parity-check matrix $\mathbf{H}$ that spans $\mathbf{H} \cdot \boldsymbol{e}$, where recall that t is the Hamming weight of noise vector $\boldsymbol{e}$. For solving dual-LPN over binary field $\mathbb{F}_2$, the Prange's ISD algorithm [Pra62] was gradually improved, e.g., [LB88, Leo88, Ste88, Dum91, FS09, BLP11]. The improved ISD algorithm [Ste88, Dum91] is called the Stern-Dumer variant (SD-ISD) in [HS13]. Later, the May-Meurer-Thomae variant (MMT-ISD) [MMT11] and the Becker-Joux-May-Meurer variant (BJMM-ISD) [BJMM12] improved the SD-ISD by using the generalized birthday algorithm [Wag02]. We give an overview of the three ISD variants in Appendix C. Recently, several works [BBC+19, EB22, EMZ22] reduced the significant space consumption of the MMT-ISD and BJMM-ISD algorithms. Our analysis does not cover the ISD algorithms [BBC+19, EB22, EMZ22] with more efficient space consumption, and always assume that sufficient memory is available which makes the LPN parameters more conservative.

Compared to the case of $\mathbb{F}_2$, the ISD algorithms to solve dual-LPN over larger fields are less studied. An initial study of ISD over a field $\mathbb{F}$ with $|\mathbb{F}| > 2$ was given by Coffey and Goodman [CG90], who provided an asymptotic analysis of the Prange's ISD algorithm over $\mathbb{F}$. Peters [Pet10] generalized the more efficient SD-ISD algorithm from binary field $\mathbb{F}_2$ to any finite field $\mathbb{F}$. Later, Meurer [Meu12] proposed a new generalization of the SD-ISD algorithm over any finite field.

The NN-ISD algorithms [MO15, Hir16, GKH17, BM18, EKZ21] applied "nearest neighbors" search to the SD-ISD or BJMM-ISD algorithms, and obtain better asymptotic complexities. However, these NN-ISD algorithms introduce a quite large polynomial overhead, which makes them less efficient when analyzing the concrete costs of low-noise LPN problems.

Previous analysis. Based on the concrete analysis of ISD attack in [HOSS18], Boyle et al. [BCGI18] used an upper bound of the cost of the Prange's ISD algorithm [Pra62] to evaluate the hardness of LPN problems

---
[12]If there exists more efficient algorithms to find such a parity-check vector, then they can also be used to improve the ISD algorithms.

over any field against ISD attacks. As shown in Tables 6 and 7, the upper-bound formula cannot capture accurately the cost of more advanced ISD variants [Ste88, Dum91, MMT11, BJMM12]. In the following, we first summarize the known ISD variants, and then use the state-of-the-art ISD algorithm to evaluate the hardness of LPN problems over finite fields [13].

Our analysis. For estimating the ISD cost of low-noise LPN problems, we distinguish two cases: binary field $\mathbb{F}_2$ and other larger fields.

Binary field $\mathbb{F}_2$. For binary field $\mathbb{F}_2$, we adopt the state-of-the-art BJMM-ISD algorithm [BJMM12] to analyze the concrete hardness of low-noise (dual-)LPN problems. From the expected time of BJMM-ISD shown in Theorem 16 of Appendix C, we can see that it is hard to give a succinct formula to compute the cost of BJMM-ISD. Thus, we choose to provide an estimator script (see Footnote 3), which allows to automatically estimate the cost of the BJMM-ISD attack.

Larger fields. For the case of non-binary fields, we focus on the hardness of (dual-)LPN problems over any finite field $\mathbb{F}$ with $|\mathbb{F}| \geq 256$ (especially for large fields with $|\mathbb{F}| \geq 2^\rho$). For other field sizes (i.e., $|\mathbb{F}| \in \{4, 8, 16, 32, 64, 128\}$), low-noise (dual-)LPN problems seem to be less interesting for PCG, MPC and ZK applications. We adopt the generalized SD-ISD algorithm [Pet10] to analyze the cost of solving low-noise (dual-)LPN problems over a field $\mathbb{F}$. As such, we provide an estimator tool to automatically estimate the cost of the SD-ISD attack.

Compared to the SD-ISD algorithm by Peters [Pet10], the SD-ISD variant by Meurer [Meu12] has the performance advantage when the field size $|\mathbb{F}| < 128$, and the advantage becomes vanishingly small when $|\mathbb{F}| \geq 128$, which is also observed in prior works such as [Tor17, BCDL19]. Thus, for the case that $|\mathbb{F}| \geq 256$, it is enough to adopt the Peters's SD-ISD algorithm to evaluate the hardness of low-noise (dual-)LPN problems. It is unclear how to extend the generalization approaches [Pet10, Meu12] to more efficient MMT-ISD and BJMM-ISD algorithms for solving low-noise LPN over a field $\mathbb{F}$ with $|\mathbb{F}| > 2$ and make the resulting ISD algorithm be significantly more efficient than the SD-ISD variant [Pet10]. Even if these generalization approaches can be efficiently applied in MMT-ISD and BJMM-ISD, the performance advantage of MMT-ISD and BJMM-ISD (compared to SD-ISD) decreases when the field size increases, and will diminish for a sufficiently large field.

For any finite field $\mathbb{F}$, the generalized SD-ISD variant [Pet10] generates two sets $\mathcal{S}_0$ and $\mathcal{S}_1$ with size $\binom{(k+\ell)/2}{p/2} \cdot (|\mathbb{F}| - 1)^{p/2}$, where p and ℓ are two additional parameters for the SD-ISD variant. Note that the size of $|\mathcal{S}_0|$ and $|\mathcal{S}_1|$ increases exponentially with p. Following the work [Pet10], p should be quite small to minimize the cost of going though sets $\mathcal{S}_0$ and $\mathcal{S}_1$, and ℓ is set as $\ell = \log_{|\mathbb{F}|} \binom{k/2}{p} + p \log_{|\mathbb{F}|}(|\mathbb{F}| - 1)$. If the even integer $p \neq 0$, then the cost of the generalized SD-ISD algorithm is at least $O((k + \ell)|\mathbb{F}|)$, which is the cost to just find identical elements in two sets $\mathcal{S}_0$ and $\mathcal{S}_1$. When the field size is large enough, we will have to choose $p = 0$ and $\ell = 0$ to minimize the cost of the generalized SD-ISD attack, according to the above equation. In this case, the generalized SD-ISD attack actually becomes Pooled Gauss attack.

C Variants of Information Set Decoding

Following the analysis [FS09, Sen11, HS13, TS16], we summarize the ISD variants by Stern-Dumer [Ste88, Dum91], May et al. [MMT11] and Becker et al. [BJMM12]. We refer the reader to the corresponding papers and the surveys [FS09, Sen11, HS13, TS16] for a more detailed description.

---

[13] Torres and Sendrier [TS16] show that known variants of ISD have the same asymptotic complexity in the sub-linear error weight regime, and the difference between the non-asymptotic exponent of some ISD variants is relatively small (but non-zero). We pick more advanced ISD for a more accurate security estimate.

C.1 Stern-Dumer Variant

The SD-ISD attack introduces two additional parameters p and ℓ, and adjusts both parameters to minimize the whole running time. Specifically, given an instance of the $(\mathsf{HW}, \mathbb{F}_2)$-dual-LPN$(N, N - k, t)$ problem $(\mathbf{H} \in \mathbb{F}_2^{(N-k) \times N}, \mathbf{y} = \mathbf{H} \cdot \mathbf{e} \in \mathbb{F}_2^{(N-k)})$, the SD-ISD attack first transforms the instance to the following equation (3) via partial Gaussian elimination, where $\mathbf{U}$ is a non-singular $(N - k) \times (N - k)$ matrix and $\mathbf{P}$ is a random $N \times N$ permutation matrix.

$$
\mathbf{U} \cdot \mathbf{H} \cdot \mathbf{P} =
\begin{array}{c}
\\
\\
\end{array}
\left.
\begin{array}{c}
1 \\
\\
\ell
\end{array}
\right\vert
\overset{\overset{\displaystyle N - k - \ell \qquad\qquad k + \ell}{}}{
\begin{bmatrix}
\begin{smallmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{smallmatrix} & \mathbf{R}_0 \\
\mathbf{0} & \mathbf{R}_1
\end{bmatrix}
}
\quad , \quad
\mathbf{U} \cdot \mathbf{y} =
\begin{bmatrix}
\mathbf{y}_0 \\
\mathbf{y}_1
\end{bmatrix}
\tag{3}
$$

Then, this attack finds $\mathbf{e}_1 \in \mathbb{F}_2^{(k+\ell)}$ such that $\mathbf{R}_1 \cdot \mathbf{e}_1 = \mathbf{y}_1$ and $|\mathbf{e}_1| \leq p$ via the following meet-in-the-middle attack:

1. For each $\mathbf{e}_{1,0} \in \mathbb{F}_2^{(k+\ell)/2}$ with $|\mathbf{e}_{1,0}| \leq p/2$, add the vector $\mathbf{R}_1 \cdot \begin{bmatrix} \mathbf{e}_{1,0} \mid \mathbf{0} \end{bmatrix}$ into a sorted set $\mathcal{S}_0$.

2. For each $\mathbf{e}_{1,1} \in \mathbb{F}_2^{(k+\ell)/2}$ with $|\mathbf{e}_{1,1}| \leq p/2$, add the vector $\mathbf{y}_1 - \mathbf{R}_1 \cdot \begin{bmatrix} \mathbf{0} \mid \mathbf{e}_{1,1} \end{bmatrix}$ into a sorted set $\mathcal{S}_1$.

3. Search for identical elements in sets $\mathcal{S}_0$ and $\mathcal{S}_1$, and then add the corresponding vectors $(\mathbf{e}_{1,0} + \mathbf{e}_{1,1})$ into a set $\mathcal{S} \subseteq \{\mathbf{e}_1 \mid \mathbf{R}_1 \cdot \mathbf{e}_1 = \mathbf{y}_1\}$.

This attack repeats the above steps until $|\mathbf{e}_0| + |\mathbf{e}_1| \leq t$ where $\mathbf{e}_0 = \mathbf{R}_0 \cdot \mathbf{e}_1 + \mathbf{y}_0$, and then outputs a noise vector $\mathbf{e} = \mathbf{P} \cdot (\mathbf{e}_0, \mathbf{e}_1)^\mathsf{T}$.

    Many variants [MMT11, BJMM12, MO15, BM18] improved the above step 3 of SD-ISD attack (finding candidate $\mathbf{e}_1$) via the generalized birthday algorithm [Wag02], the representation technique [HJ10] and the "Nearest Neighbours" search.

C.2 May-Meurer-Thomae Variant

The May-Meurer-Thomae variant (MMT-ISD) [MMT11] replaced the birthday algorithm of Stern-Dumer variant [Ste88, Dum91] with an order-2 generalized birthday algorithm [Wag02]. This variant applied this and the representation technique [HJ10] to improve ISD asymptotically and does as following,

1. For all $\mathbf{e}_{1,0} \in \mathbb{F}_2^{(k+\ell)/2}$ with $|\mathbf{e}_{1,0}| \leq p/4$, store $\mathbf{R}_1 \cdot \begin{bmatrix} \mathbf{0} \mid \mathbf{e}_{1,0} \end{bmatrix}$ in sorted $\mathcal{S}_1$ and $\mathcal{S}_3$ with $\mathcal{S}_1 = \mathcal{S}_3$.

2. For all $\mathbf{e}_{1,1} \in \mathbb{F}_2^{(k+\ell)/2}$ with $|\mathbf{e}_{1,1}| \leq p/4$, store $\mathbf{R}_1 \cdot \begin{bmatrix} \mathbf{e}_{1,1} \mid \mathbf{0} \end{bmatrix}$ in sorted $\mathcal{S}_2$ and store $\mathbf{y}_1 - \mathbf{R}_1 \cdot \begin{bmatrix} \mathbf{e}_{1,1} \mid \mathbf{0} \end{bmatrix}$ in sorted $\mathcal{S}_4$.

3. Search from $\mathcal{S}_1$ to $\mathcal{S}_4$ and generate $\mathcal{S} \subseteq \{(\mathbf{e}_1 \mid \mathbf{R}_1 \cdot \mathbf{e}_1 = \mathbf{y}_1\}$ by an order-2 generalized birthday algorithm [Wag02].

    Note that the set $\mathcal{C}$ is a singleton in the original algorithm. By allowing several $\mathbf{c} \in \mathcal{C}$ we allow larger values of r_1 and give more flexibility in the search for optimal parameters. A larger r_1 also allows smaller memory requirements with the same algorithmic complexity.

C.3 Becker-Joux-May-Meurer Variant

The Becker-Joux-May-Meurer variant (BJMM-ISD) [BJMM12] further applied an order 3 generalized birthday algorithm [Wag02] and does as following,

1. For all $e_{1,0} \in \mathbb{F}_2^{(k+\ell)/2}$ with $|e_{1,0}| \le p_2/2$, store $\mathbf{R}_1 \cdot \left[\ \mathbf{0}\ |\ e_{1,0}\ \right]$ in sorted $\mathcal{S}_1$, $\mathcal{S}_3$, $\mathcal{S}_5$ and $\mathcal{S}_7$ with $\mathcal{S}_1 = \mathcal{S}_3 = \mathcal{S}_5 = \mathcal{S}_7$.

2. For all $e_{1,1} \in \mathbb{F}_2^{(k+\ell)/2}$ with $|e_{1,1}| \le p_2/2$, store $\mathbf{R}_1 \cdot \left[\ e_{1,1}\ |\ \mathbf{0}\ \right]$ in sorted $\mathcal{S}_2$, $\mathcal{S}_4$ and $\mathcal{S}_6$ with $\mathcal{S}_2 = \mathcal{S}_4 = \mathcal{S}_6$ and store $y_1 - \mathbf{R}_1 \cdot \left[\ e_{1,1}\ |\ \mathbf{0}\ \right]$ in sorted $\mathcal{S}_8$.

3. Search from $\mathcal{S}_1$ to $\mathcal{S}_8$ and generate generate $\mathcal{S} \subseteq \{(e_1\ |\ \mathbf{R}_1 \cdot e_1 = \mathbf{0}\}$ by an order-3 generalized birthday algorithm [Wag02], where $p_2 = O(p)$ is positive additional parameters.

C.4 Cost of ISD variants

We essentially follow and simplify the analysis done in [FS09, Sen11, HS13, TS16] and count complexity by the number of field operations. The expected run-time of ISD attack consists of the below parts.

1. We denote T_{Gauss} as the cost of the partial Gaussian elimination. A naive implementation leads to $T_{Gauss} = (N - k - \ell)N^2$ field operation. Fast linear algebra [BA21] leads to $T_{Gauss} = (N - k - \ell)N^2/\log(N - k - \ell)$.

2. We estimate the success probability of one iteration. It is common in existing literature [Sen11] that each individual e_1 leads independently to success with the probability

$$\varepsilon(p, \ell)2^\ell \approx \binom{N - k - \ell}{t - p}2^\ell \Big/ \binom{N}{t}.$$

   It follows that the probability of success of one iteration is equal to

$$\mathcal{P}(p, \ell) \approx \varepsilon(p, \ell)2^\ell|\mathcal{S}|$$

   The expected value of the set $\mathcal{S}$ will depend on various birthday_decoding.

3. Complexity of various birthday_decoding.

4. The final test cost $2|\mathcal{S}|N$ field operation.

Theorem 14 (SD-ISD [HS13, Ste88, Dum91]). *The* $(\mathsf{HW}, \mathbb{F}_2)$-$\mathsf{LPN}(k, N, t)$ *problem can be solved by the SD-ISD variant in expected time*

$$T_{SD}(N, k, t) = \min_{p, \ell} \frac{1}{\mathcal{P}(p, \ell)}\Big(T_{Gauss} + 2L_0 \cdot N + 2\mathbb{E}\big[|\mathcal{S}|\big] \cdot N\Big),$$

where $L_0 = |\mathcal{S}_1| = |\mathcal{S}_2| = \binom{(k+\ell)/2}{p/2}$ *and* $\mathbb{E}\big[|\mathcal{S}|\big] = \frac{L_0^2}{2^\ell}$.

Theorem 15 (MMT-ISD variant [HS13, MMT11]). *the* $(\mathsf{HW}, \mathbb{F}_2)$-$\mathsf{LPN}(k, N, t)$ *problem can be solved in expected time* $T_{MMT}(N, k, t)$ *by the MMT-ISD variant as below*

$$T_{MMT}(N, k, t) = \min_{\ell, r_1, p, |\mathcal{C}|} \frac{1}{\mathcal{P}(p, \ell)}\Big(T_{Gauss} + |\mathcal{C}| \cdot N \cdot \Big(4L_0 + \frac{2L_0^2}{2^{r_1}} + \frac{2L_0^4}{2^{\ell+r_1}}\Big)\Big),$$

where $L_0 = \binom{(k+\ell)/2}{p/4}$ *and* $|\mathcal{S}| = \frac{|\mathcal{C}|L_0^4}{2^{\ell+r_1}}$.

Theorem 16 (BJMM-ISD variant [HS13, BJMM12]). *The* $(\mathsf{HW}, \mathbb{F}_2)$-$\mathsf{LPN}(k, N, t)$ *problem can be solved in expected time* $T_{BJMM}(N, k, t)$ *by the BJMM-ISD variant as below*

$$T_{BJMM}(N, k, t) = \min_{p, \ell, r_1, r_2, e_1, e_2} \frac{1}{\mathcal{P}(p, \ell)} \Big(T_{Gauss} + (8S_3 + 4C_3 + 2C_2 + 2C_1) \cdot N \Big),$$

where $S_3 = \binom{(k+\ell)/2}{p_2}$, $C_3 = \frac{S_3^2}{2^{r_2}}$, $C_2 = \frac{C_3^2}{2^{r_1}}$, $C_1 = \frac{S_1^2}{2^{\ell-r_1-r_2}}$, $S_1 = \min\{\mu_2 C_2, \frac{\binom{k+\ell}{p_1}}{2^{r_1+r_2}}\}$, $|\mathcal{S}| = \min\{\mu_1 C_1, \frac{\binom{k+\ell}{p}}{2^{\ell}}\}$, $\mu_1 = \frac{\binom{p_1}{e_1}\binom{k+\ell-p_1}{p_1-e_1}}{\binom{k+\ell}{p_1}}$, $\mu_2 = \frac{\binom{p_2}{e_2}\binom{k+\ell-p_2}{p_2-e_2}}{\binom{k+\ell}{p_2}}$, $p_2 = p_1/2 + e_2$ *and* $p_1 = p/2 + e_1$.