KDM-Secure Public-Key Encryption from Constant-Noise LPN

Shuai Han[1,2] and Shengli Liu[1,2,3]

[1] Department of Computer Science and Engineering,
Shanghai Jiao Tong University, Shanghai 200240, China
{dalen17,slliu}@sjtu.edu.cn
[2] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China
[3] Westone Cryptologic Research Center, Beijing 100070, China

Abstract. The Learning Parity with Noise (LPN) problem has found many applications in cryptography due to its conjectured post-quantum hardness and simple algebraic structure. Over the years, constructions of different public-key primitives were proposed from LPN, but most of them are based on the LPN assumption with *low noise* rate rather than *constant noise* rate. A recent breakthrough was made by Yu and Zhang (Crypto'16), who constructed the first Public-Key Encryption (PKE) from constant-noise LPN. However, the problem of designing a PKE with *Key-Dependent Message* (KDM) security from constant-noise LPN is still open.

In this paper, we present the first PKE with KDM-security assuming certain sub-exponential hardness of constant-noise LPN, where the number of users is predefined. The technical tool is two types of *multi-fold LPN on squared-log entropy*, one having *independent secrets* and the other *independent sample subspaces*. We establish the hardness of the multi-fold LPN variants on constant-noise LPN. Two squared-logarithmic entropy sources for multi-fold LPN are carefully chosen, so that our PKE is able to achieve correctness and KDM-security simultaneously.

Keywords: learning parity with noise, key-dependent message security, public-key encryption

1 Introduction

The search Learning Parity with Noise (LPN) problem asks to recover a random secret binary vector $\mathbf{s} \in \mathbb{F}_2^n$ from noisy linear samples of the form $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$, where $\mathbf{a} \in \mathbb{F}_2^n$ is chosen uniformly at random and $e \in \mathbb{F}_2$ follows the Bernoulli distribution $\mathcal{B}_\mu$ with parameter μ (i.e., $\Pr[\mathcal{B}_\mu = 1] = \mu$). The decisional LPN problem simply asks to distinguish the samples $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$ from uniform. The two versions of LPN turn out to be polynomially equivalent [BFKL93, KS06].

From a theoretical point, LPN offers a very strong security guarantee. The LPN problem can be formulated as a well-investigated NP-complete problem, the problem of decoding random linear codes [BMT78]. An efficient algorithm for LPN would imply a major breakthrough in coding theory. LPN also becomes a central hub in learning theory: an efficient algorithm for it could be used to learn several important concept classes such as 2-DNF formulas, juntas and any function with a sparse Fourier spectrum [FGKP06]. Until now, the best known LPN solvers require sub-exponential time. Further, there are no quantum algorithms known to have any advantage over classic ones in solving it. This makes LPN a promising candidate for post-quantum cryptography.

From a practical point, LPN-based schemes are often extremely efficient. The operations of LPN are simply bitwise exclusive OR (XOR) between binary strings, which are more efficient than other quantum-secure candidates like the learning with errors (LWE) assumption [Reg05]. Consequently, LPN-based schemes are very suitable for weak-power devices like RFID tags.

Low-Noise LPN vs. Constant-Noise LPN. Obviously, with the noise rate μ decreasing, the LPN problem can only become easier. Under a *constant noise* rate $0 < \mu < 1/2$, the best known algorithms for solving LPN require $2^{O(n/\log n)}$ time and samples [BKW03, LF06] . The time complexity goes up to $2^{O(n/\log\log n)}$ when given only polynomially many $\mathsf{poly}(n)$ samples [Lyu05], and even $2^{O(n)}$ when given only linearly many $O(n)$ samples [Ste88, MMT11]. Under a *low noise* rate $\mu = O(n^{-c})$ (typically $c = 1/2$), the best LPN solvers need only $2^{O(n^{1-c})}$ time when given $O(n)$ samples [Ste88, CC98, BLP11, Kir11, BJMM12].

The low-noise LPN is mostly believed to be a stronger assumption than constant-noise LPN. Moreover, low-noise LPN results in less efficient schemes than constant-noise LPN. For example, to achieve a same security level, the secret length n of low-noise LPN for noise rate $\mu = O(1/\sqrt{n})$ has to be squared compared with constant-noise LPN [DMN12], according to the time complexity of the attack algorithms.

For public-key primitives, Alekhnovich [Ale03] constructed a chosen-plaintext (IND-CPA) secure public-key encryption (PKE) scheme based on low-noise LPN for noise rate $\mu = O(1/\sqrt{n})$. Recently, Döttling et al. [DMN12] provided a chosen-ciphertext (IND-CCA2) secure PKE scheme from low-noise LPN, and Kiltz et al. [KMP14] improved the efficiency of the PKE scheme significantly. David et al. [DDN14] proposed a universally composable oblivious transfer (OT) protocol from low-noise LPN. All the above schemes are based on LPN for noise rate $\mu = O(1/\sqrt{n})$ or even $\mu = O(n^{-1/2-\epsilon})$ with some $\epsilon > 0$.

Though constant-noise LPN provides more security confidence and efficiency than low-noise LPN, it had been a long-standing open problem to construct public-key primitives based on constant-noise LPN since Alekhnovich's work [Ale03]. This problem was not resolved until the recent work of Yu and Zhang [YZ16], who designed the first IND-CPA secure PKE scheme, the first IND-CCA2 secure PKE scheme and the first OT protocol from constant-noise LPN.

Key-Dependent Message Security. The traditional IND-CPA (or even IND-CCA2) security might be sufficient for some scenarios, but not strong enough for high-level systems like hard disk encryptions [BHHO08] and anonymous credential systems [CL01], where messages are closely dependent on the secret keys. Such an issue was first identified by Goldwasser and Micali [GM84], and appropriate security notion for key-dependent messages was formalized as KDM-security by Black et al. [BRS02]. Over the years, more and more counterexamples were found, suggesting that IND-CPA/IND-CCA2 security does not imply KDM-security (see [ABBC10, CGH12, MO14, BHW15, KRW15, KW16, AP16, GKW17], to name a few).

Roughly speaking, a PKE scheme is called KDM-secure, if for any PPT adversary who is given public keys $(\mathsf{pk}_1, \cdots, \mathsf{pk}_l)$ of l users, it is hard to distinguish encryptions of functions of secret keys $f(\mathsf{sk}_1, \cdots, \mathsf{sk}_l)$ from encryptions of a constant say $\mathbf{0}$, where the functions f are adaptively chosen by the adversary. In this work, we focus on KDM-CPA security, where the adversary has no access to a decryption oracle.

The first KDM-secure PKE scheme in the standard model (i.e., without using random oracles) was proposed by Boneh et al. [BHHO08] and based on the decisional Diffie-Hellman (DDH) assumption. Later, more KDM-secure PKE schemes were constructed from a variety of assumptions, such as the DDH [CCS09, BHHI10, BGK11, GHV12], the quadratic residuosity (QR) [BG10] and the decisional composite residuosity (DCR) [BG10, MTY11, Hof13, LLJ15, HLL16] assumptions. However, these number-theoretic assumptions are succumb to known quantum algorithms. The only exceptions are the KDM-secure PKE designed by Applebaum et al. [ACPS09] from LWE and the one proposed by Döttling [Döt15] from low-noise LPN. Until now, the problem of constructing KDM-secure PKE from constant-noise LPN has remained open.

Applebaum [App11] provided a generic KDM amplification for boosting any KDM-secure PKE for affine functions to a KDM-secure PKE for arbitrary (bounded size) circuits. Thus it suffices to construct KDM-secure PKE schemes for affine functions to obtain schemes with KDM-security against more general class of functions.

Our Contributions. In this paper, we present the first KDM-secure PKE scheme for affine functions from *constant-noise LPN*, where the number l of users is predefined. Our construction is neat and enjoys roughly the same efficiency as the IND-CPA secure PKE scheme proposed by Yu and Zhang [YZ16]. We show a comparison in Table 1.

Table 1. Comparison among known PKE schemes either based on LPN or achieving KDM-security in the standard model under standard assumptions. "KDM?" asks whether the security is proved in the KDM setting. We kindly note that, the operations of LWE (i.e., modular additions and multiplications over a large ring) are less efficient than that of LPN (i.e., bit operations), while low-noise LPN is mostly believed to be a stronger assumption than constant-noise LPN.

Scheme	KDM?	Assumption	Quantum Resistance?
[Ale03, DMN12, KMP14]	✗	Low-noise LPN	✔
[YZ16]	✗	Constant-noise LPN	✔
[BHHO08, CCS09, BHHI10, BGK11, GHV12]	✔	DDH	✗
[BG10]	✔	QR	✗
[BG10, MTY11]	✔	DCR	✗
[Hof13, LLJ15, HLL16]	✔	DDH & DCR	✗
[ACPS09]	✔	LWE	✔
[Döt15]	✔	Low-noise LPN	✔
Ours	✔	Constant-noise LPN	✔

The starting point of our work is a variant of the LPN problem called *LPN on squared-log entropy*, which was developed by Yu and Zhang [YZ16] as a technical tool in their IND-CPA/IND-CCA2 secure PKE construction. Different from standard LPN, the secret $\mathbf{s}$ is not necessarily uniform but only required to have some squared-logarithmic entropy, and the linear samples $\mathbf{a}$ are no longer uniformly chosen but sampled from a random subspace of sublinear-sized dimension.

We introduce two types of *multi-fold version* of LPN on squared-log entropy, one having *independent secrets* and the other *independent sample subspaces*. Informally speaking, it stipulates that the samples $(\mathbf{a}_i, \langle \mathbf{a}_i, \mathbf{s}_i \rangle + e_i)$ are computationally indistinguishable from uniform, even given multiple instances $i = 1, \cdots, k$ for any polynomial k. In the version with independent secrets, $\mathbf{s}_i$ are independently distributed; in the version with independent sample subspaces, $\mathbf{a}_i$ are uniformly chosen from independent subspaces. We establish the hardness of the multi-fold LPN variants on constant-noise LPN.

Then we construct a PKE scheme and reduce the KDM-security to the multi-fold LPN variants, which are in turn implied by constant-noise LPN. In contrast to LPN-based PKE constructions in prior works like [Ale03, DMN12, YZ16], our PKE makes a novel use of two *different* squared-logarithmic entropy distributions for LPN secrets in a delicate combination, one of which is employed in the key generation algorithm and the other is employed in the encryption algorithm. This is crucial to achieving correctness and KDM-security of our PKE scheme simultaneously.

2 Preliminaries

Let $n \in \mathbb{N}$ denote the security parameter. For $i \in \mathbb{N}$, define $[i] := \{1, 2, \cdots, i\}$. Vectors are used in the column form. Denote by $x \leftarrow_\$ X$ the operation of picking an element x according to the distribution X. If X is a set, then this denotes that x is sampled uniformly at random from X. For an algorithm $\mathscr{A}$, denote by $y \leftarrow_\$ \mathscr{A}(x; r)$, or simply $y \leftarrow_\$ \mathscr{A}(x)$, the operation of running $\mathscr{A}$ with input x and randomness r and assigning output to y. Denote by $|\mathbf{s}|$ the Hamming weight of a binary string $\mathbf{s}$. For a random variable X and a distribution D, let $X \sim D$ denote that X is distributed according to D. "PPT" is short for Probabilistic Polynomial-Time. Denote by poly some polynomial function, and negl some negligible function. For random variables X and Y, the min-entropy of X is defined as $\mathbf{H}_\infty(X) := -\log(\max_x \Pr[X = x])$, and the statistical distance between X and Y is defined by $\Delta(X, Y) := \frac{1}{2} \cdot \sum_x \big| \Pr[X = x] - \Pr[Y = x] \big|$. For probability ensembles $X = \{X_n\}_{n \in \mathbb{N}}$ and $Y = \{Y_n\}_{n \in \mathbb{N}}$, X and Y are called statistically indistinguishable, denoted by $X \overset{s}{\sim} Y$, if $\Delta(X_n, Y_n) \le \mathsf{negl}(n)$; X and Y are called computationally indistinguishable, denoted by $X \overset{c}{\sim} Y$, if for any PPT distinguisher $\mathscr{D}$, $\big| \Pr[\mathscr{D}(X_n) = 1] - \Pr[\mathscr{D}(Y_n) = 1] \big| \le \mathsf{negl}(n)$.

2.1 Useful Distributions and Lemmas

For $0 < \mu, \mu_1 < 1$ and integers $n, m, q, \lambda \in \mathbb{N}$, we define some useful distributions as follows.

- Let $\mathcal{B}_\mu$ denote the Bernoulli distribution with parameter μ, i.e., $\Pr[\mathcal{B}_\mu = 1] = \mu$ and $\Pr[\mathcal{B}_\mu = 0] = 1 - \mu$, and $\mathcal{B}_\mu^n$ the concatenation of n independent copies of $\mathcal{B}_\mu$.
- Let $\widetilde{\mathcal{B}}_{\mu_1}^n$ denote the distribution $\mathcal{B}_{\mu_1}^n$ conditioned on $(1 - \frac{\sqrt{6}}{3})\mu_1 n \le |\mathcal{B}_{\mu_1}^n| \le 2\mu_1 n$, and $(\widetilde{\mathcal{B}}_{\mu_1}^n)^q$ an $n \times q$ matrix distribution where each column is an independent copy of $\widetilde{\mathcal{B}}_{\mu_1}^n$.
- Let χ_m^n denote the uniform distribution over the set $\{\mathbf{s} \in \mathbb{F}_2^n \mid |\mathbf{s}| = m\}$.
- Let $\mathcal{U}_n$ (resp., $\mathcal{U}_{q \times n}$) denote the uniform distribution over $\mathbb{F}_2^n$ (resp., $\mathbb{F}_2^{q \times n}$).
- Let $\mathcal{D}_\lambda^{q \times n} := \mathcal{U}_{q \times \lambda} \cdot \mathcal{U}_{\lambda \times n}$.
- Let $\mathcal{P}_n$ denote the uniform distribution over the set of all $n \times n$ permutation matrices, i.e., matrices that have exactly one entry of 1 in each row and each column and 0s elsewhere.

The distribution $\widetilde{\mathcal{B}}_{\mu_1}^n$ was introduced by Yu and Zhang [YZ16] as a very important distribution in the context of constant-noise LPN. $\widetilde{\mathcal{B}}_{\mu_1}^n$ can be efficiently sampleable, e.g., by sampling $\mathbf{s} \leftarrow_\$ \mathcal{B}_{\mu_1}^n$ repeatedly and outputting $\mathbf{s}$ until the condition $(1 - \frac{\sqrt{6}}{3})\mu_1 n \le |\mathbf{s}| \le 2\mu_1 n$ is met.

Remark 1. In this work, we are mostly interested in $\widetilde{\mathcal{B}}_{\mu_1}^n$ and $\chi_{\mu_1 n}^n$ for $\mu_1 = \Theta(\log n / n)$, both of which have *square-logarithmic entropy*, i.e., $\mathbf{H}_\infty(\widetilde{\mathcal{B}}_{\mu_1}^n) = \Theta(\log^2 n)$ and $\mathbf{H}_\infty(\chi_{\mu_1 n}^n) = \Theta(\log^2 n)$, as shown in [YZ16].

Lemma 1 (Chernoff Bound [KMP14, YZ16]). *For any $0 < \mu < 1$ and any $\delta > 0$, we have*

$$\Pr\left[|\mathcal{B}_\mu^n| > (1 + \delta)\mu n\right] < e^{-\frac{\min(\delta, \delta^2)}{3}\mu n}.$$

In particular, for any $0 < \mu \le (\frac{1}{2} - p)$ with $0 < p < 1/2$, we have

$$\Pr\left[|\mathcal{B}_\mu^n| > (\tfrac{1}{2} - \tfrac{p}{2})n\right] < e^{-\frac{p^2 n}{8}}.$$

Lemma 2 (Piling-up Lemma [Mat93]). *For independent random variables $e_i \sim \mathcal{B}_{\mu_i}$, $i \in [q]$, we have $\sum_{i=1}^{q} e_i \sim \mathcal{B}_\sigma$ with $\sigma = \frac{1}{2} - \frac{1}{2} \cdot \prod_{i=1}^{q}(1-2\mu_i)$.*

Lemma 3 ([YZ16, Lemma 4.3 & Lemma 4.4]). *For any $0 < \mu \leq 1/10$, any $\mu_1 = \Theta(\log n/n) \leq 1/8$, any $\mathbf{e} \in \mathbb{F}_2^n$ with $|\mathbf{e}| \leq 1.01\mu n$, and any $\mathbf{s} \in \mathbb{F}_2^n$ with $|\mathbf{s}| \leq 2\mu_1 n$, it holds that*

$$\Pr\left[\hat{\mathbf{s}}^\top \mathbf{e} = 1\right] \leq 1/2 - 2^{-\mu_1 n/2} \qquad \text{and} \qquad \Pr\left[\hat{\mathbf{e}}^\top \mathbf{s} = 1\right] \leq 1/2 - 2^{-\mu_1 n - 1},$$

where $\hat{\mathbf{s}} \sim \widetilde{\mathcal{B}}_{\mu_1}^n$ and $\hat{\mathbf{e}} \sim \mathcal{B}_\mu^n$.

We state a simplified version of the leftover hash lemma, by adopting a specific family of universal hash functions $\mathcal{H} = \{H_\mathbf{U} : \mathbb{F}_2^n \longrightarrow \mathbb{F}_2^l \mid \mathbf{U} \in \mathbb{F}_2^{l \times n}\}$, where $H_\mathbf{U}(\mathbf{x}) := \mathbf{U} \cdot \mathbf{x} \in \mathbb{F}_2^l$ for any $\mathbf{x} \in \mathbb{F}_2^n$.

Lemma 4 (Leftover Hash Lemma [HILL99]). *For any random variable X on $\mathbb{F}_2^n$ with min-entropy $\mathbf{H}_\infty(X) \geq k$, we have $\Delta\big((\mathbf{U}, \mathbf{U} \cdot \mathbf{x}), (\mathbf{U}, \mathcal{U}_l)\big) \leq 2^{-(k-l)/2}$, where $\mathbf{U} \sim \mathcal{U}_{l \times n}$ and $\mathbf{x} \sim X$.*

2.2 Learning Parity with Noise

Definition 1 (Learning Parity with Noise). *Let $0 < \mu < 1/2$. The decisional LPN problem $\mathsf{LPN}_{\mu,n}$ with secret length n and noise rate μ is hard, if for any $q = \mathsf{poly}(n)$, it holds that*

$$(\mathbf{A}, \mathbf{A} \cdot \mathbf{s} + \mathbf{e}) \ \stackrel{c}{\sim}\ (\mathbf{A}, \mathcal{U}_q), \tag{1}$$

where $\mathbf{A} \sim \mathcal{U}_{q \times n}$, $\mathbf{s} \sim \mathcal{U}_n$ and $\mathbf{e} \sim \mathcal{B}_\mu^q$.

We say that $\mathsf{LPN}_{\mu,n}$ is T-hard, if for any $q \leq T$, any probabilistic distinguisher of running time T, the distinguishing advantage in (1) is upper bounded by $1/T$.

A central tool for constructing IND-CPA/IND-CCA2 secure PKE in [YZ16] is a variant of the LPN problem, called *LPN on squared-log entropy*. There are two main differences: (i) the secret $\mathbf{s}$ is not necessarily uniform, but only required to have some squared-logarithmic entropy; (ii) the rows of $\mathbf{A}$ are no longer uniformly chosen, but sampled from a *random subspace* of squared-logarithmic dimension. It was shown in [YZ16] that under constant-noise LPN with certain sub-exponential hardness, the LPN problem on squared-log entropy is hard even given some log-sized auxiliary input about the secret and noise. Formally, we have the following theorem.

Theorem 1 (LPN on Squared-log Entropy [YZ16, Theorem 4.1]). *Let $0 < \mu < 1/2$ be any constant. Assume that $\mathsf{LPN}_{\mu,n}$ is $2^{\omega(n^{\frac{1}{2}})}$-hard, then for any $\lambda = \Theta(\log^2 n)$, $q = \mathsf{poly}(n)$, any polynomial-time sampleable distribution $\mathcal{S}$ on $\mathbb{F}_2^n$ with $\mathbf{H}_\infty(\mathcal{S}) \geq 2\lambda$, and any polynomial-time computable function $f : (\mathbb{F}_2^n \times \mathbb{F}_2^q) \times \mathcal{Z} \longrightarrow \mathbb{F}_2^{O(\log n)}$ with public coins Z, we have*

$$(\mathbf{A}, \mathbf{A} \cdot \mathbf{s} + \mathbf{e}, Z, f(\mathbf{s}, \mathbf{e}; Z)) \ \stackrel{c}{\sim}\ (\mathbf{A}, \mathcal{U}_q, Z, f(\mathbf{s}, \mathbf{e}; Z)),$$

where $\mathbf{A} \sim \mathcal{D}_\lambda^{q \times n}$, $\mathbf{s} \sim \mathcal{S}$ and $\mathbf{e} \sim \mathcal{B}_\mu^q$.

By Remark 1, $\widetilde{\mathcal{B}}_{\mu_1}^n$ and $\chi_{\mu_1 n}^n$ with $\mu_1 = \Theta(\log n/n)$ are suitable candidate distributions for $\mathcal{S}$, as long as the constant hidden in $\lambda = \Theta(\log^2 n)$ is small enough such that $\mathbf{H}_\infty(\widetilde{\mathcal{B}}_{\mu_1}^n) \geq 2\lambda$ and $\mathbf{H}_\infty(\chi_{\mu_1 n}^n) \geq 2\lambda$ holds.

2.3 Public-Key Encryption and Key-Dependent Message Security

A public-key encryption (PKE) scheme $\mathsf{PKE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ with secret key space $\mathcal{SK}$ and message space $\mathcal{M}$ consists of a tuple of PPT algorithms: (i) the key generation algorithm $\mathsf{KeyGen}(1^n)$ outputs a public key pk and a secret key $\mathsf{sk} \in \mathcal{SK}$; (ii) the encryption algorithm $\mathsf{Enc}(\mathsf{pk}, \mathsf{m})$ takes as input a public key pk and a message $\mathsf{m} \in \mathcal{M}$, and outputs a ciphertext c; (iii) the decryption algorithm $\mathsf{Dec}(\mathsf{sk}, \mathsf{c})$ takes as input a secret key sk and a ciphertext c, and outputs either a message m or a failure symbol $\perp$. Correctness of PKE requires that, for all messages $\mathsf{m} \in \mathcal{M}$, we have

$$\Pr\left[(\mathsf{pk}, \mathsf{sk}) \leftarrow_{\$} \mathsf{KeyGen}(1^n) : \mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, \mathsf{m})) \neq \mathsf{m}\right] \quad \leq \quad \mathsf{negl}(n),$$

where the probability is over the inner coin tosses of KeyGen and Enc.

Definition 2 (KDM-Security for PKE). *Let $l \in \mathbb{N}$ denote the number of users, and let $\mathcal{F}$ be a family of functions from $(\mathcal{SK})^l$ to $\mathcal{M}$. A PKE scheme PKE is called l-KDM[$\mathcal{F}$]-CPA secure, if for any PPT adversary $\mathscr{A}$, in the following l-$\mathsf{kdm}[\mathcal{F}]$-$\mathsf{cpa}$ game played between $\mathscr{A}$ and a challenger $\mathscr{C}$, the advantage of $\mathscr{A}$ is negligible in n.*

<u>KeyGen.</u> *$\mathscr{C}$ picks $b \leftarrow_{\$} \{0,1\}$ as a challenge bit, and proceeds as follows.*
(a) For each user $i \in [l]$, invoke $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow_{\$} \mathsf{KeyGen}(1^n)$.
Finally, $\mathscr{C}$ sends the public keys $(\mathsf{pk}_1, \cdots, \mathsf{pk}_l)$ to $\mathscr{A}$.

<u>Chal($j \in [l], f \in \mathcal{F}$).</u> *$\mathscr{A}$ can query this oracle $\mathsf{poly}(n)$ times. Each time, $\mathscr{A}$ sends a user identity $j \in [l]$ and a function $f \in \mathcal{F}$ to $\mathscr{C}$, and $\mathscr{C}$ proceeds as follows.*
(a) Set $f \leftarrow \mathbf{0}$ (the zero function) if $b = 0$. Then compute a message $\mathsf{m} := f(\mathsf{sk}_1, \cdots, \mathsf{sk}_l) \in \mathcal{M}$.
(b) Compute the encryption of m under the public key pk_j of the j-th user, i.e., $\mathsf{c} \leftarrow_{\$} \mathsf{Enc}(\mathsf{pk}_j, \mathsf{m})$.
Finally, $\mathscr{C}$ returns the challenge ciphertext c to $\mathscr{A}$.

<u>Guess.</u> *$\mathscr{A}$ outputs a guessing bit $b' \in \{0,1\}$. The advantage of $\mathscr{A}$ is defined as $\left|\Pr[b' = b] - \frac{1}{2}\right|$.*

3 Multi-fold LPN on Squared-log Entropy

In this section, we present the technical tools used in our construction of KDM-secure PKE from constant-noise LPN. We develop two types of *multi-fold version* of LPN on squared-log entropy: one has *independent secrets* and the other has *independent sample subspaces.*

3.1 Multi-fold LPN on Squared-log Entropy with Independent Secrets

Firstly, we state a k-fold version of LPN on squared-log entropy with independent secrets and noise vectors, where the auxiliary input per fold is a 2-bit linear leakage of the secret and noise.

Lemma 5. *Let $0 < \mu < 1/2$ be any constant. Assume that $\mathsf{LPN}_{\mu,n}$ is $2^{\omega(n^{\frac{1}{2}})}$-hard, then for any $\mu_1 = \Theta(\log n/n)$ and $\lambda = \Theta(\log^2 n)$ such that $\mathbf{H}_\infty(\widetilde{\mathcal{B}}^n_{\mu_1}) \geq 2\lambda$, and any $k = \mathsf{poly}(n)$, it holds that*

$$(\mathbf{A}, \hat{\mathbf{S}}^\top \mathbf{A} + \hat{\mathbf{E}}^\top, (\mathbf{e}, \mathbf{s}, \mathbf{P}), (\hat{\mathbf{S}}^\top \mathbf{e}, \hat{\mathbf{E}}^\top \mathbf{P}\mathbf{s})) \stackrel{c}{\sim} (\mathbf{A}, \mathcal{U}_{k\times n}, (\mathbf{e}, \mathbf{s}, \mathbf{P}), (\hat{\mathbf{S}}^\top \mathbf{e}, \hat{\mathbf{E}}^\top \mathbf{P}\mathbf{s})), \qquad (2)$$

where $\mathbf{A} \sim \mathcal{D}^{n\times n}_\lambda$, $\hat{\mathbf{S}} \sim (\widetilde{\mathcal{B}}^n_{\mu_1})^k$, $\hat{\mathbf{E}} \sim \mathcal{B}^{n\times k}_\mu$, $\mathbf{e} \sim \mathcal{B}^n_\mu$, $\mathbf{s} \sim \chi^n_{\mu_1 n}$ and $\mathbf{P} \sim \mathcal{P}_n$.

Proof of Lemma 5. By instantiating a transposed version of Theorem 1 with $q = n$, $\mathcal{S} = \widetilde{\mathcal{B}}^n_{\mu_1}$ and $f : (\mathbb{F}_2^n \times \mathbb{F}_2^n) \times (\mathbb{F}_2^n \times \mathbb{F}_2^n \times \mathbb{F}_2^{n\times n}) \longrightarrow \mathbb{F}_2^2$ being $f(\hat{\mathbf{s}}, \hat{\mathbf{e}}; (\mathbf{e}, \mathbf{s}, \mathbf{P})) = (\hat{\mathbf{s}}^\top \mathbf{e}, \hat{\mathbf{e}}^\top \mathbf{P}\mathbf{s})$, we obtain

$$(\mathbf{A}, \hat{\mathbf{s}}^\top \mathbf{A} + \hat{\mathbf{e}}^\top, (\mathbf{e}, \mathbf{s}, \mathbf{P}), (\hat{\mathbf{s}}^\top \mathbf{e}, \hat{\mathbf{e}}^\top \mathbf{P}\mathbf{s})) \overset{c}{\sim} (\mathbf{A}, \mathcal{U}_{1\times n}, (\mathbf{e}, \mathbf{s}, \mathbf{P}), (\hat{\mathbf{s}}^\top \mathbf{e}, \hat{\mathbf{e}}^\top \mathbf{P}\mathbf{s})), \tag{3}$$

where $\mathbf{A} \sim \mathcal{D}_\lambda^{n\times n}$, $\hat{\mathbf{s}} \sim \widetilde{\mathcal{B}}^n_{\mu_1}$, $\hat{\mathbf{e}} \sim \mathcal{B}^n_\mu$, and ($\mathbf{e} \sim \mathcal{B}^n_\mu$, $\mathbf{s} \sim \chi^n_{\mu_1 n}$, $\mathbf{P} \sim \mathcal{P}_n$) are public coins. Observe that (2) is k-fold version of (3), thus a standard hybrid argument leads to Lemma 5. ▌

We also develop a k-fold version of LPN on squared-log entropy with independent secrets and noise vectors, where the auxiliary input per fold is a 1-bit linear leakage of a special form. We show that *the auxiliary input is also computationally indistinguishable from uniform.*

Lemma 6. *Let $0 < \mu < 1/2$ be any constant. Assume that $\mathsf{LPN}_{\mu,n}$ is $2^{\omega(n^{\frac{1}{2}})}$-hard, then for any $\mu_1 = \Theta(\log n/n)$ and $\lambda = \Theta(\log^2 n)$ such that $\mathbf{H}_\infty(\widetilde{\mathcal{B}}^n_{\mu_1}) \geq 2\lambda$, and any $k = \mathsf{poly}(n)$, it holds that*

$$(\mathbf{A}, \hat{\mathbf{S}}^\top \mathbf{A} + \hat{\mathbf{E}}^\top, \mathbf{y}, \hat{\mathbf{S}}^\top \mathbf{y} + \mathbf{e}) \overset{c}{\sim} (\mathbf{A}, \mathcal{U}_{k\times n}, \mathbf{y}, \mathcal{U}_k), \tag{4}$$

where $\mathbf{A} \sim \mathcal{D}_\lambda^{n\times n}$, $\hat{\mathbf{S}} \sim (\widetilde{\mathcal{B}}^n_{\mu_1})^k$, $\hat{\mathbf{E}} \sim \mathcal{B}^{n\times k}_\mu$, $\mathbf{y} \sim \mathcal{U}_n$ and $\mathbf{e} \sim \mathcal{B}^k_\mu$.

Proof of Lemma 6. By instantiating a transposed version of Theorem 1 with $q = n$, $\mathcal{S} = \widetilde{\mathcal{B}}^n_{\mu_1}$ and $f : (\mathbb{F}_2^n \times \mathbb{F}_2^n) \times (\mathbb{F}_2^n \times \mathbb{F}_2) \longrightarrow \mathbb{F}_2$ being $f(\hat{\mathbf{s}}, \hat{\mathbf{e}}; (\mathbf{y}, e)) = \hat{\mathbf{s}}^\top \mathbf{y} + e$, we have

$$\begin{aligned}(\mathbf{A}, \hat{\mathbf{s}}^\top \mathbf{A} + \hat{\mathbf{e}}^\top, (\mathbf{y}, e), \hat{\mathbf{s}}^\top \mathbf{y} + e) &\overset{c}{\sim} (\mathbf{A}, \mathcal{U}_{1\times n}, (\mathbf{y}, e), \hat{\mathbf{s}}^\top \mathbf{y} + e)\\ \Rightarrow (\mathbf{A}, \hat{\mathbf{s}}^\top \mathbf{A} + \hat{\mathbf{e}}^\top, \mathbf{y}, \hat{\mathbf{s}}^\top \mathbf{y} + e) &\overset{c}{\sim} (\mathbf{A}, \mathcal{U}_{1\times n}, \mathbf{y}, \hat{\mathbf{s}}^\top \mathbf{y} + e),\end{aligned} \tag{5}$$

where $\mathbf{A} \sim \mathcal{D}_\lambda^{n\times n}$, $\hat{\mathbf{s}} \sim \widetilde{\mathcal{B}}^n_{\mu_1}$, $\hat{\mathbf{e}} \sim \mathcal{B}^n_\mu$, and ($\mathbf{y} \sim \mathcal{U}_n$, $e \sim \mathcal{B}_\mu$) are public coins. Again, by instantiating a transposed version of Theorem 1 with $q = 1$, $\mathcal{S} = \widetilde{\mathcal{B}}^n_{\mu_1}$ and f that always outputs nothing, we get

$$\begin{aligned}(\mathbf{y}, \hat{\mathbf{s}}^\top \mathbf{y} + e) &\overset{c}{\sim} (\mathbf{y}, \mathcal{U}_1)\\ \Rightarrow (\mathbf{A}, \mathcal{U}_{1\times n}, \mathbf{y}, \hat{\mathbf{s}}^\top \mathbf{y} + e) &\overset{c}{\sim} (\mathbf{A}, \mathcal{U}_{1\times n}, \mathbf{y}, \mathcal{U}_1),\end{aligned} \tag{6}$$

where $\mathbf{A} \sim \mathcal{D}_\lambda^{n\times n}$, $\mathbf{y} \sim \mathcal{D}_\lambda^{n\times 1} = \mathcal{U}_n$, $\hat{\mathbf{s}} \sim \widetilde{\mathcal{B}}^n_{\mu_1}$ and $e \sim \mathcal{B}_\mu$.

By combining (5) with (6), we immediately obtain

$$(\mathbf{A}, \hat{\mathbf{s}}^\top \mathbf{A} + \hat{\mathbf{e}}^\top, \mathbf{y}, \hat{\mathbf{s}}^\top \mathbf{y} + e) \overset{c}{\sim} (\mathbf{A}, \mathcal{U}_{1\times n}, \mathbf{y}, \mathcal{U}_1). \tag{7}$$

Observe that (4) is k-fold version of (7), thus a standard hybrid argument leads to Lemma 6. ▌

3.2 Multi-fold LPN on Squared-log Entropy with Independent Sample Subspaces

We introduce an l-fold version of LPN on squared-log entropy, with independent sample subspaces and noise vectors, but shared a same secret $\mathbf{s}$, i.e.,

$$(\mathbf{A}_i, \mathbf{A}_i \cdot \mathbf{s} + \mathbf{e}_i, Z, f(\mathbf{s}, \mathbf{e}_i; Z))_{i\in[l]} \overset{c}{\sim} (\mathbf{A}_i, \mathcal{U}_q, Z, f(\mathbf{s}, \mathbf{e}_i; Z))_{i\in[l]}.$$

The name of "sample subspaces" originates from the fact that, each $\mathbf{A}_i \sim \mathcal{D}_\lambda^{q\times n}$ is associated with a random *subspace* of dimension λ, from which the rows of $\mathbf{A}_i$ are sampled.

We stress that this cannot be implied by Theorem 1, for two reasons: (i) for l independent $\mathbf{A}_i \sim \mathcal{D}_\lambda^{q\times n}$, the distribution of their concatenation $\begin{pmatrix}\mathbf{A}_1\\ \vdots \\ \mathbf{A}_l\end{pmatrix}$ does not follow the form of $\mathcal{D}_\lambda^{lq\times n}$ any more; (ii) we cannot resort to a hybrid argument since the secret $\mathbf{s}$ is shared by the l folds and unknown to the simulator.

For our KDM-secure PKE, it suffices to consider the case free of auxiliary input.

Theorem 2. *Let $0 < \mu < 1/2$ and $l \in \mathbb{N}$ be any constant. Assume that $\mathsf{LPN}_{\mu,n}$ is $2^{\omega(n^{\frac{1}{2}})}$-hard, then for any $\mu_1 = \Theta(\log n/n)$ and $\lambda = \Theta(\log^2 n)$ such that $\mathbf{H}_\infty(\chi^n_{\mu_1 n}) \geq (l+1)\lambda$, it holds that*

$$(\mathbf{A}_i, \mathbf{A}_i \cdot \mathbf{s} + \mathbf{e}_i)_{i\in[l]} \ \stackrel{c}{\sim}\ (\mathbf{A}_i, \mathbf{u}_i)_{i\in[l]},$$

where $\mathbf{s} \sim \chi^n_{\mu_1 n}$, $\mathbf{A}_i \sim \mathcal{D}_\lambda^{n\times n}$, $\mathbf{e}_i \sim \mathcal{B}^n_\mu$ and $\mathbf{u}_i \sim \mathcal{U}_n$ for $i \in [l]$.

Proof of Theorem 2. Since $\mathbf{H}_\infty(\chi^n_{\mu_1 n}) \geq (l+1)\lambda$, by the leftover hash lemma (i.e., Lemma 4), we have

$$(\mathbf{V}, \mathbf{V}\cdot\mathbf{s}) \ \stackrel{s}{\sim}\ (\mathbf{V}, \mathbf{y}),$$

where $\mathbf{V} \sim \mathcal{U}_{l\lambda\times n}$, $\mathbf{s} \sim \chi^n_{\mu_1 n}$ and $\mathbf{y} \sim \mathcal{U}_{l\lambda}$.

By expressing $\mathbf{V} = \begin{pmatrix}\mathbf{V}_1\\ \vdots \\ \mathbf{V}_l\end{pmatrix}$ with $\mathbf{V}_i \sim \mathcal{U}_{\lambda\times n}$ and $\mathbf{y} = \begin{pmatrix}\mathbf{y}_1\\ \vdots \\ \mathbf{y}_l\end{pmatrix}$ with $\mathbf{y}_i \sim \mathcal{U}_\lambda$, we get

$$(\mathbf{V}_i, \mathbf{V}_i\cdot\mathbf{s})_{i\in[l]} \ \stackrel{s}{\sim}\ (\mathbf{V}_i, \mathbf{y}_i)_{i\in[l]}$$

$$\Rightarrow ((\mathbf{U}_i, \mathbf{V}_i), \mathbf{U}_i\cdot\mathbf{V}_i\cdot\mathbf{s} + \mathbf{e}_i)_{i\in[l]} \ \stackrel{s}{\sim}\ ((\mathbf{U}_i, \mathbf{V}_i), \mathbf{U}_i\cdot\mathbf{y}_i + \mathbf{e}_i)_{i\in[l]}, \tag{8}$$

where $\mathbf{U}_i \sim \mathcal{U}_{n\times\lambda}$, and $\mathbf{e}_i \sim \mathcal{B}^n_\mu$.

Next, consider the $\mathsf{LPN}_{\mu,\lambda}$ problem on uniform string $\mathbf{y}_i$ of length λ (instead of n), which is assumed to be $2^{\omega(\lambda^{\frac{1}{2}})}$ $(= n^{\omega(1)})$-hard. It implies that

$$(\mathbf{U}_i, \mathbf{U}_i\cdot\mathbf{y}_i + \mathbf{e}_i) \ \stackrel{c}{\sim}\ (\mathbf{U}_i, \mathbf{u}_i),$$

where $\mathbf{u}_i \sim \mathcal{U}_n$, for any $i \in [l]$. Through a standard hybrid argument, we have

$$(\mathbf{U}_i, \mathbf{U}_i\cdot\mathbf{y}_i + \mathbf{e}_i)_{i\in[l]} \ \stackrel{c}{\sim}\ (\mathbf{U}_i, \mathbf{u}_i)_{i\in[l]}$$

$$\Rightarrow ((\mathbf{U}_i, \mathbf{V}_i), \mathbf{U}_i\cdot\mathbf{y}_i + \mathbf{e}_i)_{i\in[l]} \ \stackrel{c}{\sim}\ ((\mathbf{U}_i, \mathbf{V}_i), \mathbf{u}_i)_{i\in[l]}. \tag{9}$$

Finally, by combining (8) with (9) and setting $\mathbf{A}_i := \mathbf{U}_i\cdot\mathbf{V}_i \sim \mathcal{D}_\lambda^{n\times n}$, Theorem 2 follows. ▮

4 Construction of KDM-Secure PKE from Constant-Noise LPN

In this section, we present a PKE scheme with KDM-security for affine functions assuming certain sub-exponential hardness (i.e., $2^{\omega(n^{\frac{1}{2}})}$ for secret size n) of constant-noise LPN.

4.1 The Construction

Our PKE scheme uses the following parameters and building blocks.

- Let $0 < \mu \leq 1/10$, $\alpha > 0$ and $l \in \mathbb{N}$ be any constants, and let $\mu_1 = \alpha \log n / n$.
- Let $\lambda = \beta \log^2 n$ with a constant $\beta > 0$ such that both $\mathbf{H}_\infty(\widetilde{\mathcal{B}}^n_{\mu_1}) \geq 2\lambda$ and $\mathbf{H}_\infty(\chi^n_{\mu_1 n}) \geq (l+1)\lambda$ holds. By Remark 1, such a λ can be easily found by setting β small enough.
- Let $\mathbf{G} \in \mathbb{F}_2^{k \times n}$ be the generator matrix of a binary linear error-correcting code together with an efficient decoding algorithm Decode, which can correct at least $(\frac{1}{2} - \frac{2}{5n^{3\alpha/2}}) \cdot k$ errors. Such a code exists for $k = O(n^{3\alpha+1})$, and explicit constructions of the code can be found in [For66].

We present the construction of $\mathsf{PKE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ with secret key space $\mathbb{F}_2^n$ and message space $\mathbb{F}_2^n$ in Fig. 1.

$(\mathsf{pk}, \mathsf{sk}) \leftarrow_\$ \mathsf{KeyGen}(1^n)$: $\mathbf{A} \leftarrow_\$ \mathcal{D}_\lambda^{n \times n}$. $\mathbf{s} \leftarrow_\$ \chi^n_{\mu_1 n}$. $\mathbf{e} \leftarrow_\$ \mathcal{B}^n_\mu$. $\mathbf{y} := \mathbf{A}\mathbf{s} + \mathbf{e} \in \mathbb{F}_2^n$. Return $\mathsf{pk} := (\mathbf{A}, \mathbf{y})$, $\mathsf{sk} := \mathbf{s} \in \mathbb{F}_2^n$.	$\mathsf{c} \leftarrow_\$ \mathsf{Enc}(\mathsf{pk}, \mathsf{m})$: // $\mathsf{m} \in \mathbb{F}_2^n$ Parse $\mathsf{pk} = (\mathbf{A}, \mathbf{y})$. $\hat{\mathbf{S}} \leftarrow_\$ (\widetilde{\mathcal{B}}^n_{\mu_1})^k$. $\hat{\mathbf{E}} \leftarrow_\$ \mathcal{B}_\mu^{n \times k}$. $\mathbf{C}_1 := \hat{\mathbf{S}}^\top \mathbf{A} + \hat{\mathbf{E}}^\top \in \mathbb{F}_2^{k \times n}$. $\hat{\mathbf{e}} \leftarrow_\$ \mathcal{B}^k_\mu$. $\mathbf{c}_2 := \hat{\mathbf{S}}^\top \mathbf{y} + \hat{\mathbf{e}} + \mathbf{G}\mathsf{m} \in \mathbb{F}_2^k$. Return $\mathsf{c} := (\mathbf{C}_1, \mathbf{c}_2)$.	$\mathsf{m} \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{c})$: Parse $\mathsf{sk} = \mathbf{s}$. Parse $\mathsf{c} = (\mathbf{C}_1, \mathbf{c}_2)$. $\mathbf{z} := \mathbf{c}_2 - \mathbf{C}_1 \mathbf{s} \in \mathbb{F}_2^k$. $\mathsf{m} := \mathsf{Decode}(\mathbf{z}) \in \mathbb{F}_2^n$. Return m.

Fig. 1. Construction of PKE with KDM-security from constant-noise LPN.

Remark 2. In contrast to LPN-based PKE constructions in prior works like [Ale03, DMN12, YZ16], our PKE scheme makes a novel use of two squared-log entropy distributions for LPN secrets in a delicate combination, i.e., $\chi^n_{\mu_1 n}$ in the KeyGen algorithm and $\widetilde{\mathcal{B}}^n_{\mu_1}$ in the Enc algorithm. This is crucial to achieving correctness and KDM-security of our scheme simultaneously. Jumping ahead,

- For KDM-security, the distribution $\chi^n_{\mu_1 n}$ employed in KeyGen allows us to express secret keys of l users, $\mathbf{s}_i \sim \chi^n_{\mu_1 n}$ with $i \in [l]$, as random permutations of a base secret key $\mathbf{s}^* \sim \chi^n_{\mu_1 n}$, i.e., $\mathbf{s}_i := \mathbf{P}_i \cdot \mathbf{s}^*$ for $\mathbf{P}_i \sim \mathcal{P}_n$. Then we are able to reduce KDM-security for l users to that for a single user. This approach makes the KDM-security proof possible. (See Subsect. 4.3 for the formal security proof.)
- For correctness, the distribution $\widetilde{\mathcal{B}}^n_{\mu_1}$ employed in Enc helps us to use Lemma 3 to bound the error term $\hat{\mathbf{S}}^\top \mathbf{e}$ in decryption, where $\hat{\mathbf{S}} \sim (\widetilde{\mathcal{B}}^n_{\mu_1})^k$, and decode the message m successfully. (See Subsect. 4.2 for the formal correctness analysis.)

We stress that $\chi^n_{\mu_1 n}$ and $\widetilde{\mathcal{B}}^n_{\mu_1}$ are carefully selected so that both the correctness and KDM-security can be satisfied. If $\chi^n_{\mu_1 n}$ is adopted in both KeyGen and Enc, it will be hard for us to show the correctness; if $\widetilde{\mathcal{B}}^n_{\mu_1}$ is adopted in both KeyGen and Enc, it will be hard for us to prove the KDM-security.

4.2 Correctness

Theorem 3. *Our PKE scheme* PKE *in Fig. 1 is correct.*

Proof of Theorem 3. For $(\mathsf{pk}, \mathsf{sk}) \leftarrow_{\$} \mathsf{KeyGen}(1^n)$ and $\mathsf{c} \leftarrow_{\$} \mathsf{Enc}(\mathsf{pk}, \mathsf{m})$, we have

$$\mathsf{pk} = (\mathbf{A}, \mathbf{y}) = (\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e}) \quad \text{and} \quad \mathsf{c} = (\mathbf{C}_1, \mathbf{c}_2) = (\hat{\mathbf{S}}^\top \mathbf{A} + \hat{\mathbf{E}}^\top, \hat{\mathbf{S}}^\top \mathbf{y} + \hat{\mathbf{e}} + \mathbf{G}\mathsf{m}),$$

where $\mathbf{s} \sim \chi^n_{\mu_1 n}$, $\mathbf{e} \sim \mathcal{B}^n_\mu$, $\hat{\mathbf{S}} \sim (\widetilde{\mathcal{B}}^n_{\mu_1})^k$, $\hat{\mathbf{E}} \sim \mathcal{B}^{n\times k}_\mu$ and $\hat{\mathbf{e}} \sim \mathcal{B}^k_\mu$. Then in $\mathsf{Dec}(\mathsf{sk}, \mathsf{c})$, it follows that

$$\begin{aligned} \mathbf{z} = \mathbf{c}_2 - \mathbf{C}_1\mathbf{s} &= \hat{\mathbf{S}}^\top \mathbf{y} + \hat{\mathbf{e}} + \mathbf{G}\mathsf{m} - (\hat{\mathbf{S}}^\top \mathbf{A} + \hat{\mathbf{E}}^\top) \cdot \mathbf{s} \\ &= \hat{\mathbf{S}}^\top \cdot (\mathbf{A}\mathbf{s} + \mathbf{e}) + \hat{\mathbf{e}} + \mathbf{G}\mathsf{m} - (\hat{\mathbf{S}}^\top \mathbf{A} + \hat{\mathbf{E}}^\top) \cdot \mathbf{s} \\ &= \mathbf{G}\mathsf{m} + \hat{\mathbf{e}} + \hat{\mathbf{S}}^\top \mathbf{e} - \hat{\mathbf{E}}^\top \mathbf{s}. \end{aligned}$$

We analyze the error term $\hat{\mathbf{e}} + \hat{\mathbf{S}}^\top \mathbf{e} - \hat{\mathbf{E}}^\top \mathbf{s}$. By the Chernoff bound (i.e., Lemma 1), $|\mathbf{e}| \le 1.01\mu n$ holds except with negligible probability $2^{-\Omega(n)}$. Besides, $|\mathbf{s}| = \mu_1 n \le 2\mu_1 n$. Thus, by Lemma 3, we have $\hat{\mathbf{S}}^\top \mathbf{e} \sim \mathcal{B}^k_{\sigma_1}$ for $\sigma_1 \le 1/2 - 2^{-\mu_1 n/2} = 1/2 - n^{-\alpha/2}$, and $\hat{\mathbf{E}}^\top \mathbf{s} \sim \mathcal{B}^k_{\sigma_2}$ for $\sigma_2 \le 1/2 - 2^{-\mu_1 n - 1} = 1/2 - n^{-\alpha}/2$. Then by the Piling-up Lemma (i.e., Lemma 2), $\hat{\mathbf{e}} + \hat{\mathbf{S}}^\top \mathbf{e} - \hat{\mathbf{E}}^\top \mathbf{s} \sim \mathcal{B}^k_\sigma$ for $\sigma \le 1/2 - \frac{4}{5} \cdot n^{-3\alpha/2}$. Finally, by Lemma 1,

$$\Pr\left[\ \left|\hat{\mathbf{e}} + \hat{\mathbf{S}}^\top \mathbf{e} - \hat{\mathbf{E}}^\top \mathbf{s}\right| \le \left(\tfrac{1}{2} - \tfrac{2}{5n^{3\alpha/2}}\right) \cdot k\ \right] \ge 1 - 2^{-\Omega(n^{-3\alpha}k)} = 1 - 2^{-\Omega(n)}.$$

Therefore, with overwhelming probability, it holds that $\left|\hat{\mathbf{e}} + \hat{\mathbf{S}}^\top \mathbf{e} - \hat{\mathbf{E}}^\top \mathbf{s}\right| \le \left(\frac{1}{2} - \frac{2}{5n^{3\alpha/2}}\right) \cdot k$, and in this case, Decode will be able to decode m from $\mathbf{z}$. ▮

4.3 KDM-Security for Affine Functions

Theorem 4. *Let* $\mathcal{F}_{\mathit{aff}} = \left\{ f : (\mathbb{F}^n_2)^l \longrightarrow \mathbb{F}^n_2 \right\}$ *be a family of affine functions. Assume that* $\mathsf{LPN}_{n,\mu}$ *is* $2^{\omega(n^{\frac{1}{2}})}$*-hard, then our PKE scheme* PKE *in Fig. 1 is* l*-KDM*$[\mathcal{F}_{\mathit{aff}}]$*-CPA secure.*

Proof of Theorem 4. Suppose that $\mathscr{A}$ is a PPT adversary against the l-KDM$[\mathcal{F}_{\mathrm{aff}}]$-CPA security of PKE with advantage ϵ. We prove the theorem by defining a sequence of games G_1–G_{12} and showing that ϵ is negligible in n. (We also illustrate the games in Fig. 2-3 in Appendix A.1.) The changes between adjacent games will be highlighted by red underline. In the sequel, by $a \overset{\mathsf{G}_i}{=} b$ we mean that a equals b or is computed as b in game G_i, and by $\Pr_i[\cdot]$ we denote the probability of a particular event occurring in game G_i.

Game G_1. This is the l-$\mathsf{kdm}[\mathcal{F}_{\mathrm{aff}}]$-$\mathsf{cpa}$ security game of PKE, which is played between $\mathscr{A}$ and a challenger $\mathscr{C}$.

KeyGen. $\mathscr{C}$ picks $b \leftarrow_{\$} \{0,1\}$ as the challenge bit, and generates the public keys of l users as follows.

(a) For each user $i \in [l]$, choose $\mathbf{A}_i \leftarrow_{\$} \mathcal{D}^{n\times n}_\lambda$, $\mathbf{s}_i \leftarrow_{\$} \chi^n_{\mu_1 n}$, $\mathbf{e}_i \leftarrow_{\$} \mathcal{B}^n_\mu$, and compute $\mathbf{y}_i := \mathbf{A}_i\mathbf{s}_i + \mathbf{e}_i$. Finally, $\mathscr{C}$ sends the public keys $\mathsf{pk}_i := (\mathbf{A}_i, \mathbf{y}_i)$, $i \in [l]$, to $\mathscr{A}$.

Chal$(j \in [l], f \in \mathcal{F}_{\mathit{aff}})$. $\mathscr{A}$ can query this oracle $Q = \mathsf{poly}(n)$ times. Each time, $\mathscr{A}$ sends a user identity $j \in [l]$ and an affine function $f \in \mathcal{F}_{\mathrm{aff}}$ to $\mathscr{C}$, and $\mathscr{C}$ proceeds as follows.

(a) Set $f \leftarrow \mathbf{0}$ (the zero function) if $b = 0$. Then compute the message $\mathsf{m} := f(\mathsf{sk}_1, \cdots, \mathsf{sk}_l) \in \mathbb{F}_2^n$, which essentially is $\mathsf{m} := \sum_{i \in [l]} \mathbf{T}_i \mathbf{s}_i + \mathbf{t} \in \mathbb{F}_2^n$, where $\mathbf{T}_i \in \mathbb{F}_2^{n \times n}$ and $\mathbf{t} \in \mathbb{F}_2^n$ are $\mathbf{0}$s in the case of $b = 0$ and are specified by $\mathscr{A}$ as the description of the affine function f in the case of $b = 1$.

(b) Compute the encryption of m under the public key $\mathsf{pk}_j = (\mathbf{A}_j, \mathbf{y}_j)$ of the j-th user, i.e., choose $\hat{\mathbf{S}} \leftarrow_\$ (\widetilde{\mathcal{B}}_{\mu_1}^n)^k$, $\hat{\mathbf{E}} \leftarrow_\$ \mathcal{B}_\mu^{n \times k}$, $\hat{\mathbf{e}} \leftarrow_\$ \mathcal{B}_\mu^k$, and compute $\mathbf{C}_1 := \hat{\mathbf{S}}^\top \mathbf{A}_j + \hat{\mathbf{E}}^\top$ and $\mathbf{c}_2 := \hat{\mathbf{S}}^\top \mathbf{y}_j + \hat{\mathbf{e}} + \mathbf{G}\mathsf{m}$.

Finally, $\mathscr{C}$ returns the challenge ciphertext $\mathsf{c} := (\mathbf{C}_1, \mathbf{c}_2)$ to $\mathscr{A}$.

GUESS. $\mathscr{A}$ outputs a guessing bit $b' \in \{0, 1\}$.

Let Win denote the event that $b' = b$. Then by definition, $\epsilon = \left| \Pr_1[\mathsf{Win}] - \frac{1}{2} \right|$.

Game G_2. This game is the same as G_1, except that, the oracle KEYGEN is changed as follows.

KEYGEN. $\mathscr{C}$ picks $b \leftarrow_\$ \{0, 1\}$ uniformly, and proceeds as follows.

(a) Choose a master secret $\mathbf{s}^* \leftarrow_\$ \chi_{\mu_1 n}^n$.

(b) For each user $i \in [l]$, choose $\mathbf{A}_i \leftarrow_\$ \mathcal{D}_\lambda^{n \times n}$, $\mathbf{P}_i \leftarrow_\$ \mathcal{P}_n$, $\mathbf{e}_i \leftarrow_\$ \mathcal{B}_\mu^n$, and compute $\mathbf{s}_i := \mathbf{P}_i \mathbf{s}^* \in \mathbb{F}_2^n$ and $\mathbf{y}_i := \mathbf{A}_i \mathbf{P}_i \mathbf{s}^* + \mathbf{e}_i \in \mathbb{F}_2^n$.

Finally, $\mathscr{C}$ sends the public keys $\mathsf{pk}_i := (\mathbf{A}_i, \mathbf{y}_i)$, $i \in [l]$, to $\mathscr{A}$.

Claim 1. $\Pr_1[\mathsf{Win}] = \Pr_2[\mathsf{Win}]$.

Proof of Claim 1. Since $\mathbf{s}^* \sim \chi_{\mu_1 n}^n$, we have $|\mathbf{s}^*| = \mu_1 n$. Then as $\mathbf{P}_i \sim \mathcal{P}_n$, $\mathbf{s}_i = \mathbf{P}_i \mathbf{s}^*$ follows the distribution $\chi_{\mu_1 n}^n$ and is independent of $\mathbf{s}^*$, the same as that in game G_1. Besides, $\mathbf{y}_i \overset{\mathsf{G}_1}{=} \mathbf{A}_i \mathbf{s}_i + \mathbf{e}_i \overset{\mathsf{G}_2}{=} \mathbf{A}_i \mathbf{P}_i \mathbf{s}^* + \mathbf{e}_i$. Consequently, the changes are just conceptual, and $\Pr_1[\mathsf{Win}] = \Pr_2[\mathsf{Win}]$. ∎

Game G_3. This game is the same as G_2, except that, the oracle CHAL is changed as follows.

CHAL($j \in [l], f \in \mathcal{F}_{\mathit{aff}}$). $\mathscr{C}$ proceeds as follows.

(a) Set $f \leftarrow \mathbf{0}$ if $b = 0$. Then compute $\mathbf{T}_f := \sum_{i \in [l]} \mathbf{T}_i \mathbf{P}_i \in \mathbb{F}_2^{n \times n}$ and $\mathsf{m} := \mathbf{T}_f \mathbf{s}^* + \mathbf{t} \in \mathbb{F}_2^n$.

(b) Choose $\hat{\mathbf{S}} \leftarrow_\$ (\widetilde{\mathcal{B}}_{\mu_1}^n)^k$, $\hat{\mathbf{E}} \leftarrow_\$ \mathcal{B}_\mu^{n \times k}$, $\hat{\mathbf{e}} \leftarrow_\$ \mathcal{B}_\mu^k$, and compute $\mathbf{C}_1 := \hat{\mathbf{S}}^\top \mathbf{A}_j + \hat{\mathbf{E}}^\top \in \mathbb{F}_2^{k \times n}$ and $\mathbf{c}_2 := (\mathbf{C}_1 \mathbf{P}_j + \mathbf{G}\mathbf{T}_f) \cdot \mathbf{s}^* - \hat{\mathbf{E}}^\top \mathbf{P}_j \mathbf{s}^* + \hat{\mathbf{S}}^\top \mathbf{e}_j + \hat{\mathbf{e}} + \mathbf{G}\mathbf{t} \in \mathbb{F}_2^k$.

Finally, $\mathscr{C}$ returns the challenge ciphertext $\mathsf{c} := (\mathbf{C}_1, \mathbf{c}_2)$ to $\mathscr{A}$.

Claim 2. $\Pr_2[\mathsf{Win}] = \Pr_3[\mathsf{Win}]$.

Proof of Claim 2. Observe that $\mathsf{m} \overset{\mathsf{G}_2}{=} \sum_{i \in [l]} \mathbf{T}_i \mathbf{s}_i + \mathbf{t} = \sum_{i \in [l]} \mathbf{T}_i \cdot (\mathbf{P}_i \mathbf{s}^*) + \mathbf{t} \overset{\mathsf{G}_3}{=} \mathbf{T}_f \mathbf{s}^* + \mathbf{t}$, and

$$
\begin{aligned}
\mathbf{c}_2 &\overset{\mathsf{G}_2}{=} \hat{\mathbf{S}}^\top \mathbf{y}_j + \hat{\mathbf{e}} + \mathbf{G}\mathsf{m} = \hat{\mathbf{S}}^\top \cdot (\mathbf{A}_j \mathbf{P}_j \mathbf{s}^* + \mathbf{e}_j) + \hat{\mathbf{e}} + \mathbf{G} \cdot (\mathbf{T}_f \mathbf{s}^* + \mathbf{t}) \\
&= (\hat{\mathbf{S}}^\top \mathbf{A}_j \mathbf{P}_j + \mathbf{G}\mathbf{T}_f) \cdot \mathbf{s}^* + \hat{\mathbf{S}}^\top \mathbf{e}_j + \hat{\mathbf{e}} + \mathbf{G}\mathbf{t} \\
&= ((\mathbf{C}_1 - \hat{\mathbf{E}}^\top) \mathbf{P}_j + \mathbf{G}\mathbf{T}_f) \cdot \mathbf{s}^* + \hat{\mathbf{S}}^\top \mathbf{e}_j + \hat{\mathbf{e}} + \mathbf{G}\mathbf{t} \\
&\overset{\mathsf{G}_3}{=} (\mathbf{C}_1 \mathbf{P}_j + \mathbf{G}\mathbf{T}_f) \cdot \mathbf{s}^* - \hat{\mathbf{E}}^\top \mathbf{P}_j \mathbf{s}^* + \hat{\mathbf{S}}^\top \mathbf{e}_j + \hat{\mathbf{e}} + \mathbf{G}\mathbf{t},
\end{aligned}
$$

where the penultimate equality is due to $\mathbf{C}_1 = \hat{\mathbf{S}}^\top \mathbf{A}_j + \hat{\mathbf{E}}^\top$. Thus, the changes are just conceptual. ∎

Game G_4. This game is the same as G_3, except that, the oracle CHAL is changed as follows.

$\underline{\textsc{Chal}(j \in [l], f \in \mathcal{F}_{aff}).}$ $\mathscr{C}$ proceeds as follows.

(a) Set $f \leftarrow \mathbf{0}$ if $b = 0$. Then compute $\mathbf{T}_f := \sum_{i\in[l]} \mathbf{T}_i\mathbf{P}_i \in \mathbb{F}_2^{n\times n}$.

(b) Choose $\hat{\mathbf{S}} \leftarrow_\$ (\widetilde{\mathcal{B}}_{\mu_1}^n)^k$, $\hat{\mathbf{E}} \leftarrow_\$ \mathcal{B}_\mu^{n\times k}$, $\hat{\mathbf{e}} \leftarrow_\$ \mathcal{B}_\mu^k$, $\underline{\mathbf{U} \leftarrow_\$ \mathbb{F}_2^{k\times n}}$, and compute $\underline{\mathbf{C}_1 := \mathbf{U} \in \mathbb{F}_2^{k\times n}}$ and $\mathbf{c}_2 := (\mathbf{C}_1\mathbf{P}_j + \mathbf{G}\mathbf{T}_f) \cdot \mathbf{s}^* - \hat{\mathbf{E}}^\top\mathbf{P}_j\mathbf{s}^* + \hat{\mathbf{S}}^\top\mathbf{e}_j + \hat{\mathbf{e}} + \mathbf{G}\mathbf{t} \in \mathbb{F}_2^k$.

Finally, $\mathscr{C}$ returns the challenge ciphertext $\mathsf{c} := (\mathbf{C}_1, \mathbf{c}_2)$ to $\mathscr{A}$.

Claim 3. If $\mathsf{LPN}_{\mu,n}$ is $2^{\omega(n^{\frac{1}{2}})}$-hard, then $\big|\Pr_3[\mathsf{Win}] - \Pr_4[\mathsf{Win}]\big| \leq \mathsf{negl}(n)$.

Proof of Claim 3. Firstly, we introduce a sequence of games $\{\mathsf{G}_{3,\kappa}\}_{\kappa\in[Q+1]}$ between G_3 and G_4.

– **Game $\mathsf{G}_{3,\kappa}$, $\kappa \in [Q+1]$.** This game is a hybrid of game G_3 and game G_4: for the first $\kappa-1$ times of Chal queries, $\mathscr{C}$ computes $\mathbf{C}_1$ as in game G_4; for the remaining Chal queries, $\mathscr{C}$ computes $\mathbf{C}_1$ as in game G_3.

Clearly, game $\mathsf{G}_{3,1}$ is identical to G_3 and game $\mathsf{G}_{3,Q+1}$ is identical to G_4. It suffices to show that $\big|\Pr_{3,\kappa}[\mathsf{Win}] - \Pr_{3,\kappa+1}[\mathsf{Win}]\big| \leq \mathsf{negl}(n)$ for any $\kappa \in [Q]$.

The only difference between game $\mathsf{G}_{3,\kappa}$ and game $\mathsf{G}_{3,\kappa+1}$ is the distribution of $\mathbf{C}_1$ in the κ-th $\textsc{Chal}(j \in [l], f \in \mathcal{F}_{\text{aff}})$ query: in game $\mathsf{G}_{3,\kappa}$, $\mathbf{C}_1$ is computed according to game G_3, i.e., $\mathbf{C}_1 = \hat{\mathbf{S}}^\top\mathbf{A}_j + \hat{\mathbf{E}}^\top$; in game $\mathsf{G}_{3,\kappa+1}$, it is computed according to game G_4, i.e., $\mathbf{C}_1 = \mathbf{U}$.

We construct a PPT distinguisher $\mathscr{D}$ to solve the multi-fold LPN problem described in Lemma 5. Given a challenge $(\mathbf{A}, \mathbf{C}, (\mathbf{e}, \mathbf{s}, \mathbf{P}), (\hat{\mathbf{S}}^\top\mathbf{e}, \hat{\mathbf{E}}^\top\mathbf{P}\mathbf{s}))$, $\mathscr{D}$ wants to distinguish $\mathbf{C} = \hat{\mathbf{S}}^\top\mathbf{A} + \hat{\mathbf{E}}^\top$ from $\mathbf{C} = \mathbf{U}$, where $\mathbf{A} \leftarrow_\$ \mathcal{D}_\lambda^{n\times n}$, $\hat{\mathbf{S}} \leftarrow_\$ (\widetilde{\mathcal{B}}_{\mu_1}^n)^k$, $\hat{\mathbf{E}} \leftarrow_\$ \mathcal{B}_\mu^{n\times k}$, $\mathbf{e} \leftarrow_\$ \mathcal{B}_\mu^n$, $\mathbf{s} \leftarrow_\$ \chi_{\mu_1 n}^n$, $\mathbf{P} \leftarrow_\$ \mathcal{P}_n$ and $\mathbf{U} \leftarrow_\$ \mathbb{F}_2^{k\times n}$. $\mathscr{D}$ is constructed by simulating game $\mathsf{G}_{3,\kappa}$ or game $\mathsf{G}_{3,\kappa+1}$ for $\mathscr{A}$ as follows, where we highlight the challenge received by $\mathscr{D}$.

$\underline{\textsc{KeyGen}.}$ $\mathscr{D}$ picks $b \leftarrow_\$ \{0,1\}$ uniformly, and proceeds as follows.

(a) Set the master secret $\mathbf{s}^* := \mathbf{s}$.

(b) Pick $j^* \leftarrow_\$ [l]$. For each user $i \in [l]$,
  – if $i \neq j^*$, choose $\mathbf{A}_i \leftarrow_\$ \mathcal{D}_\lambda^{n\times n}$, $\mathbf{P}_i \leftarrow_\$ \mathcal{P}_n$, $\mathbf{e}_i \leftarrow_\$ \mathcal{B}_\mu^n$;
  – if $i = j^*$, set $\mathbf{A}_{j^*} := \mathbf{A}$, $\mathbf{P}_{j^*} := \mathbf{P}$, $\mathbf{e}_{j^*} := \mathbf{e}$,
  and compute $\mathbf{s}_i := \mathbf{P}_i\mathbf{s}^* \in \mathbb{F}_2^n$ and $\mathbf{y}_i := \mathbf{A}_i\mathbf{P}_i\mathbf{s}^* + \mathbf{e}_i \in \mathbb{F}_2^n$.

Finally, $\mathscr{D}$ sends the public keys $\mathsf{pk}_i := (\mathbf{A}_i, \mathbf{y}_i)$, $i \in [l]$, to $\mathscr{A}$.

$\underline{\textsc{Chal}(j \in [l], f \in \mathcal{F}_{aff}).}$ $\mathscr{D}$ proceeds as follows.

(a) Set $f \leftarrow \mathbf{0}$ if $b = 0$. Then compute $\mathbf{T}_f := \sum_{i\in[l]} \mathbf{T}_i\mathbf{P}_i \in \mathbb{F}_2^{n\times n}$.

(b) – For the first $\kappa - 1$ queries, $\mathscr{D}$ computes $\mathbf{C}_1$ and $\mathbf{c}_2$ according to game G_4.
  – For the κ-th query, $\mathscr{D}$ aborts immediately if $j \neq j^*$; otherwise $\mathscr{D}$ chooses $\hat{\mathbf{e}} \leftarrow_\$ \mathcal{B}_\mu^k$, and computes $\mathbf{C}_1 := \mathbf{C}$ and $\mathbf{c}_2 := (\mathbf{C}_1\mathbf{P}_{j^*} + \mathbf{G}\mathbf{T}_f) \cdot \mathbf{s}^* - \hat{\mathbf{E}}^\top\mathbf{P}\mathbf{s} + \hat{\mathbf{S}}^\top\mathbf{e} + \hat{\mathbf{e}} + \mathbf{G}\mathbf{t}$.
  – For the remaining queries, $\mathscr{D}$ computes $\mathbf{C}_1$ and $\mathbf{c}_2$ according to game G_3.

Finally, $\mathscr{D}$ returns the challenge ciphertext $\mathsf{c} := (\mathbf{C}_1, \mathbf{c}_2)$ to $\mathscr{A}$.

$\underline{\textsc{Guess}.}$ $\mathscr{A}$ outputs a guessing bit $b' \in \{0,1\}$.

$\mathscr{D}$ finally outputs 1 if and only if $j = j^*$ holds in the κ-th CHAL query (i.e., $\mathscr{D}$ does not abort) and $b' = b$.

We analyze the distinguishing advantage of $\mathscr{D}$.

- In KEYGEN, $\mathbf{s}^*$, $\mathbf{A}_{j^*}$, $\mathbf{P}_{j^*}$ and $\mathbf{e}_{j^*}$ have the same distributions as in both game $\mathsf{G}_{3,\kappa}$ and game $\mathsf{G}_{3,\kappa+1}$. Besides, j^* is completely hidden from $\mathscr{A}$'s view.
- In the κ-th query of CHAL($j \in [l], f \in \mathcal{F}_{\text{aff}}$), $j = j^*$ holds with probability at least $1/l$.
  - If $\mathbf{C} = \hat{\mathbf{S}}^\top \mathbf{A} + \hat{\mathbf{E}}^\top$, then $\mathbf{C}_1 = \hat{\mathbf{S}}^\top \mathbf{A} + \hat{\mathbf{E}}^\top = \hat{\mathbf{S}}^\top \mathbf{A}_{j^*} + \hat{\mathbf{E}}^\top$ and $\mathbf{c}_2 = (\mathbf{C}_1 \mathbf{P}_{j^*} + \mathbf{G}\mathbf{T}_f) \cdot \mathbf{s}^* - \hat{\mathbf{E}}^\top \mathbf{P}_{j^*}\mathbf{s}^* + \hat{\mathbf{S}}^\top \mathbf{e}_{j^*} + \hat{\mathbf{e}} + \mathbf{G}\mathbf{t}$. Thus, $\mathscr{D}$ computes $(\mathbf{C}_1, \mathbf{c}_2)$ for the κ-th CHAL query exactly like game G_3.
  - If $\mathbf{C} = \mathbf{U}$, then $\mathbf{C}_1 = \mathbf{U}$ and $\mathbf{c}_2 = (\mathbf{C}_1 \mathbf{P}_{j^*} + \mathbf{G}\mathbf{T}_f) \cdot \mathbf{s}^* - \hat{\mathbf{E}}^\top \mathbf{P}_{j^*}\mathbf{s}^* + \hat{\mathbf{S}}^\top \mathbf{e}_{j^*} + \hat{\mathbf{e}} + \mathbf{G}\mathbf{t}$. Thus, $\mathscr{D}$ computes $(\mathbf{C}_1, \mathbf{c}_2)$ for the κ-th CHAL query exactly like game G_4.

Therefore, if $\mathscr{D}$ does not abort (which occurs with probability at least $1/l$), $\mathscr{D}$ simulates game $\mathsf{G}_{3,\kappa}$ perfectly for $\mathscr{A}$ in the case of $\mathbf{C} = \hat{\mathbf{S}}^\top \mathbf{A} + \hat{\mathbf{E}}^\top$ and simulates game $\mathsf{G}_{3,\kappa+1}$ perfectly for $\mathscr{A}$ in the case of $\mathbf{C} = \mathbf{U}$. Consequently, $\mathscr{D}$'s distinguishing advantage is at least $\frac{1}{l} \cdot \big| \Pr_{3,\kappa}[\mathsf{Win}] - \Pr_{3,\kappa+1}[\mathsf{Win}] \big|$, which is $\mathsf{negl}(n)$ by Lemma 5.

In conclusion, $\big| \Pr_3[\mathsf{Win}] - \Pr_4[\mathsf{Win}] \big| = \big| \Pr_{3,1}[\mathsf{Win}] - \Pr_{3,Q+1}[\mathsf{Win}] \big| \le \sum_{\kappa \in [Q]} \big| \Pr_{3,\kappa}[\mathsf{Win}] - \Pr_{3,\kappa+1}[\mathsf{Win}] \big| \le Ql \cdot \mathsf{negl}(n)$, which is also negligible in n. This completes the proof of Claim 3. ▌

Game G_5. This game is the same as G_4, except that, the oracle CHAL is changed as follows.

CHAL($j \in [l], f \in \mathcal{F}_{\mathit{aff}}$). $\mathscr{C}$ proceeds as follows.

(a) Set $f \leftarrow \mathbf{0}$ if $b = 0$. Then compute $\mathbf{T}_f := \sum_{i \in [l]} \mathbf{T}_i \mathbf{P}_i \in \mathbb{F}_2^{n \times n}$.

(b) Choose $\hat{\mathbf{S}} \leftarrow_\$ (\widetilde{\mathcal{B}}^n_{\mu_1})^k$, $\hat{\mathbf{E}} \leftarrow_\$ \mathcal{B}^{n \times k}_\mu$, $\hat{\mathbf{e}} \leftarrow_\$ \mathcal{B}^k_\mu$, $\mathbf{U} \leftarrow_\$ \mathbb{F}_2^{k \times n}$, and compute $\mathbf{C}_1 := \mathbf{U} - \mathbf{G}\mathbf{T}_f \mathbf{P}_j^{-1} \in \mathbb{F}_2^{k \times n}$ and $\mathbf{c}_2 := (\mathbf{C}_1 \mathbf{P}_j + \mathbf{G}\mathbf{T}_f) \cdot \mathbf{s}^* - \hat{\mathbf{E}}^\top \mathbf{P}_j \mathbf{s}^* + \hat{\mathbf{S}}^\top \mathbf{e}_j + \hat{\mathbf{e}} + \mathbf{G}\mathbf{t} \in \mathbb{F}_2^k$.

Finally, $\mathscr{C}$ returns the challenge ciphertext $\mathsf{c} := (\mathbf{C}_1, \mathbf{c}_2)$ to $\mathscr{A}$.

Claim 4. $\Pr_4[\mathsf{Win}] = \Pr_5[\mathsf{Win}]$.

Proof of Claim 4. Since $\mathbf{U}$ is uniformly chosen and independent of other parts of the game, $\mathbf{C}_1 = \mathbf{U}$ in game G_4 has the same distribution as $\mathbf{C}_1 = \mathbf{U} - \mathbf{G}\mathbf{T}_f \mathbf{P}_j^{-1}$ in game G_5. Thus, this change is just conceptual, and $\Pr_4[\mathsf{Win}] = \Pr_5[\mathsf{Win}]$. ▌

Game G_6. This game is the same as G_5, except that, the oracle CHAL is changed as follows.

CHAL($j \in [l], f \in \mathcal{F}_{\mathit{aff}}$). $\mathscr{C}$ proceeds as follows.

(a) Set $f \leftarrow \mathbf{0}$ if $b = 0$. Then compute $\mathbf{T}_f := \sum_{i \in [l]} \mathbf{T}_i \mathbf{P}_i \in \mathbb{F}_2^{n \times n}$.

(b) Choose $\hat{\mathbf{S}} \leftarrow_\$ (\widetilde{\mathcal{B}}^n_{\mu_1})^k$, $\hat{\mathbf{E}} \leftarrow_\$ \mathcal{B}^{n \times k}_\mu$, $\hat{\mathbf{e}} \leftarrow_\$ \mathcal{B}^k_\mu$, and compute $\mathbf{C}_1 := \hat{\mathbf{S}}^\top \mathbf{A}_j + \hat{\mathbf{E}}^\top - \mathbf{G}\mathbf{T}_f \mathbf{P}_j^{-1} \in \mathbb{F}_2^{k \times n}$ and $\mathbf{c}_2 := (\mathbf{C}_1 \mathbf{P}_j + \mathbf{G}\mathbf{T}_f) \cdot \mathbf{s}^* - \hat{\mathbf{E}}^\top \mathbf{P}_j \mathbf{s}^* + \hat{\mathbf{S}}^\top \mathbf{e}_j + \hat{\mathbf{e}} + \mathbf{G}\mathbf{t} \in \mathbb{F}_2^k$.

Finally, $\mathscr{C}$ returns the challenge ciphertext $\mathsf{c} := (\mathbf{C}_1, \mathbf{c}_2)$ to $\mathscr{A}$.

Claim 5. If $\mathsf{LPN}_{\mu,n}$ is $2^{\omega(n^{\frac{1}{2}})}$-hard, then $\big| \Pr_5[\mathsf{Win}] - \Pr_6[\mathsf{Win}] \big| \le \mathsf{negl}(n)$.

The proof of Claim 5 is essentially the same as that for Claim 3, since the change from game G_5 to game G_6 is symmetric to the change from game G_3 to game G_4. For completeness, we put the proof in Appendix A.2.

Game G_7. This game is the same as G_6, except that, the oracle CHAL is changed as follows.

CHAL$(j \in [l], f \in \mathcal{F}_{\mathit{aff}})$. $\mathscr{C}$ proceeds as follows.

(a) Set $f \leftarrow \mathbf{0}$ if $b = 0$. Then compute $\mathbf{T}_f := \sum_{i\in[l]} \mathbf{T}_i\mathbf{P}_i \in \mathbb{F}_2^{n\times n}$.

(b) Choose $\hat{\mathbf{S}} \leftarrow_{\$} (\widetilde{\mathcal{B}}_{\mu_1}^n)^k$, $\hat{\mathbf{E}} \leftarrow_{\$} \mathcal{B}_\mu^{n\times k}$, $\hat{\mathbf{e}} \leftarrow_{\$} \mathcal{B}_\mu^k$, and compute $\mathbf{C}_1 := \hat{\mathbf{S}}^\top\mathbf{A}_j + \hat{\mathbf{E}}^\top - \mathbf{G}\mathbf{T}_f\mathbf{P}_j^{-1} \in \mathbb{F}_2^{k\times n}$ and $\mathbf{c}_2 := \hat{\mathbf{S}}^\top\mathbf{y}_j + \hat{\mathbf{e}} + \mathbf{G}\mathbf{t} \in \mathbb{F}_2^k$.

Finally, $\mathscr{C}$ returns the challenge ciphertext $\mathsf{c} := (\mathbf{C}_1, \mathbf{c}_2)$ to $\mathscr{A}$.

Claim 6. $\Pr_6[\mathsf{Win}] = \Pr_7[\mathsf{Win}]$.

Proof of Claim 6. Observe that

$$\begin{aligned}\mathbf{c}_2 &\overset{\mathsf{G}_6}{=} (\mathbf{C}_1\mathbf{P}_j + \mathbf{G}\mathbf{T}_f)\cdot\mathbf{s}^* - \hat{\mathbf{E}}^\top\mathbf{P}_j\mathbf{s}^* + \hat{\mathbf{S}}^\top\mathbf{e}_j + \hat{\mathbf{e}} + \mathbf{G}\mathbf{t}\\ &= ((\hat{\mathbf{S}}^\top\mathbf{A}_j + \hat{\mathbf{E}}^\top)\mathbf{P}_j)\cdot\mathbf{s}^* - \hat{\mathbf{E}}^\top\mathbf{P}_j\mathbf{s}^* + \hat{\mathbf{S}}^\top\mathbf{e}_j + \hat{\mathbf{e}} + \mathbf{G}\mathbf{t}\\ &= \hat{\mathbf{S}}^\top\cdot(\mathbf{A}_j\mathbf{P}_j\mathbf{s}^* + \mathbf{e}_j) + \hat{\mathbf{e}} + \mathbf{G}\mathbf{t} \overset{\mathsf{G}_7}{=} \hat{\mathbf{S}}^\top\mathbf{y}_j + \hat{\mathbf{e}} + \mathbf{G}\mathbf{t},\end{aligned}$$

where the second equality follows from the fact that $\mathbf{C}_1 = \hat{\mathbf{S}}^\top\mathbf{A}_j + \hat{\mathbf{E}}^\top - \mathbf{G}\mathbf{T}_f\mathbf{P}_j^{-1}$. Consequently, this change is just conceptual, and $\Pr_6[\mathsf{Win}] = \Pr_7[\mathsf{Win}]$. ▌

Game G_8. This game is the same as G_7, except that, the oracle KEYGEN is changed as follows.

KEYGEN. $\mathscr{C}$ picks $b \leftarrow_{\$} \{0,1\}$ uniformly, and proceeds as follows.

(a) Choose a master secret $\mathbf{s}^* \leftarrow_{\$} \chi_{\mu_1 n}^n$.

(b) For each user $i \in [l]$, choose $\mathbf{B}_i \leftarrow_{\$} \mathcal{D}_\lambda^{n\times n}$, $\mathbf{P}_i \leftarrow_{\$} \mathcal{P}_n$, $\mathbf{e}_i \leftarrow_{\$} \mathcal{B}_\mu^n$, and compute $\mathbf{A}_i := \mathbf{B}_i\mathbf{P}_i^{-1} \in \mathbb{F}_2^{n\times n}$ and $\mathbf{y}_i := \mathbf{B}_i\mathbf{s}^* + \mathbf{e}_i \in \mathbb{F}_2^n$.

Finally, $\mathscr{C}$ sends the public keys $\mathsf{pk}_i := (\mathbf{A}_i, \mathbf{y}_i)$, $i \in [l]$, to $\mathscr{A}$.

Claim 7. $\Pr_7[\mathsf{Win}] = \Pr_8[\mathsf{Win}]$.

Proof of Claim 7. For each $i \in [l]$, the permutation $\mathbf{P}_i \sim \mathcal{P}_n$ is invertible. Then as $\mathbf{B}_i \sim \mathcal{D}_\lambda^{n\times n}$, $\mathbf{A}_i = \mathbf{B}_i\mathbf{P}_i^{-1}$ also follows the distribution $\mathcal{D}_\lambda^{n\times n}$ and independent of $\mathbf{P}_i$. The reason is as follows. $\mathbf{B}_i \sim \mathcal{D}_\lambda^{n\times n}$ basically means that $\mathbf{B}_i = \mathbf{U}_i\mathbf{V}_i$ for $\mathbf{U}_i \sim \mathcal{U}_{n\times\lambda}$ and $\mathbf{V}_i \sim \mathcal{U}_{\lambda\times n}$. Then $\mathbf{A}_i = \mathbf{B}_i\mathbf{P}_i^{-1} = \mathbf{U}_i(\mathbf{V}_i\mathbf{P}_i^{-1})$, where $\mathbf{V}_i\mathbf{P}_i^{-1}$ follows the distribution $\mathcal{U}_{\lambda\times n}$ since $\mathbf{V}_i$ is. Consequently, $\mathbf{A}_i$ is distributed according to $\mathcal{D}_\lambda^{n\times n}$, the same as that in game G_7.

Besides, $\mathbf{y}_i \overset{\mathsf{G}_7}{=} \mathbf{A}_i\mathbf{P}_i\mathbf{s}^* + \mathbf{e}_i = (\mathbf{B}_i\mathbf{P}_i^{-1})\cdot\mathbf{P}_i\mathbf{s}^* + \mathbf{e}_i \overset{\mathsf{G}_8}{=} \mathbf{B}_i\mathbf{s}^* + \mathbf{e}_i$. Thus, the changes are just conceptual, and $\Pr_7[\mathsf{Win}] = \Pr_8[\mathsf{Win}]$. ▌

Game G_9. This game is the same as G_8, except that, the oracle KEYGEN is changed as follows.

KEYGEN. $\mathscr{C}$ picks $b \leftarrow_{\$} \{0,1\}$ uniformly, and proceeds as follows.

(a) For each user $i \in [l]$, choose $\mathbf{B}_i \leftarrow_{\$} \mathcal{D}_\lambda^{n\times n}$, $\mathbf{P}_i \leftarrow_{\$} \mathcal{P}_n$, and compute $\mathbf{A}_i := \mathbf{B}_i\mathbf{P}_i^{-1} \in \mathbb{F}_2^{n\times n}$ and $\mathbf{y}_i \leftarrow_{\$} \mathbb{F}_2^n$.

Finally, $\mathscr{C}$ sends the public keys $\mathsf{pk}_i := (\mathbf{A}_i, \mathbf{y}_i)$, $i \in [l]$, to $\mathscr{A}$.

Claim 8. If $\mathsf{LPN}_{\mu,n}$ is $2^{\omega(n^{\frac{1}{2}})}$-hard, then $\big|\Pr_8[\mathsf{Win}] - \Pr_9[\mathsf{Win}]\big| \le \mathsf{negl}(n)$.

Proof of Claim 8. The only difference between game G_8 and game G_9 is that $\mathbf{y}_i = \mathbf{B}_i\mathbf{s}^* + \mathbf{e}_i$ in G_8 is replaced by $\mathbf{y}_i \leftarrow_{\$} \mathbb{F}_2^n$ in G_9. Observe that the master secret key $\mathbf{s}^*$ and the noise vectors $\mathbf{e}_i$, $i \in [l]$, are never used in the CHAL oracle in both G_8 and G_9. Therefore, we can directly bound the difference by constructing a PPT distinguisher $\mathscr{D}$ to solve the multi-fold LPN problem described in Theorem 2.

Given a challenge $(\mathbf{B}_i, \mathbf{y}_i)_{i\in[l]}$, $\mathscr{D}$ wants to distinguish $\mathbf{y}_i = \mathbf{B}_i\mathbf{s} + \mathbf{e}_i$ from $\mathbf{y}_i \leftarrow_{\$} \mathbb{F}_2^n$, where $\mathbf{s} \leftarrow_{\$} \chi_{\mu_1 n}^n$, $\mathbf{B}_i \leftarrow_{\$} \mathcal{D}_\lambda^{n\times n}$ and $\mathbf{e}_i \leftarrow_{\$} \mathcal{B}_\mu^n$. $\mathscr{D}$ is constructed by simulating game G_8 or game G_9 for $\mathscr{A}$ as follows, where we highlight the challenge received by $\mathscr{D}$.

KEYGEN. $\mathscr{D}$ picks $b \leftarrow_{\$} \{0,1\}$ uniformly, and proceeds as follows.

(a) For each user $i \in [l]$, set $\mathbf{B}_i := \boxed{\mathbf{B}_i} \in \mathbb{F}_2^{n\times n}$, choose $\mathbf{P}_i \leftarrow_{\$} \mathcal{P}_n$, and compute $\mathbf{A}_i := \mathbf{B}_i\mathbf{P}_i^{-1} \in \mathbb{F}_2^{n\times n}$ and $\mathbf{y}_i := \boxed{\mathbf{y}_i} \in \mathbb{F}_2^n$.

Finally, $\mathscr{D}$ sends the public keys $\mathsf{pk}_i := (\mathbf{A}_i, \mathbf{y}_i)$, $i \in [l]$, to $\mathscr{A}$.

CHAL($j \in [l], f \in \mathcal{F}_{\mathit{aff}}$). $\mathscr{D}$ computes $\mathbf{C}_1$ and $\mathbf{c}_2$ in the same way as both G_8 and G_9. That is,

(a) Set $f \leftarrow \mathbf{0}$ if $b = 0$. Then compute $\mathbf{T}_f := \sum_{i\in[l]} \mathbf{T}_i\mathbf{P}_i \in \mathbb{F}_2^{n\times n}$.

(b) Choose $\hat{\mathbf{S}} \leftarrow_{\$} (\widetilde{\mathcal{B}}_{\mu_1}^n)^k$, $\hat{\mathbf{E}} \leftarrow_{\$} \mathcal{B}_\mu^{n\times k}$, $\hat{\mathbf{e}} \leftarrow_{\$} \mathcal{B}_\mu^k$, and compute $\mathbf{C}_1 := \hat{\mathbf{S}}^\top\mathbf{A}_j + \hat{\mathbf{E}}^\top - \mathbf{G}\mathbf{T}_f\mathbf{P}_j^{-1} \in \mathbb{F}_2^{k\times n}$ and $\mathbf{c}_2 := \hat{\mathbf{S}}^\top\mathbf{y}_j + \hat{\mathbf{e}} + \mathbf{G}\mathbf{t} \in \mathbb{F}_2^k$.

Finally, $\mathscr{D}$ returns the challenge ciphertext $\mathsf{c} := (\mathbf{C}_1, \mathbf{c}_2)$ to $\mathscr{A}$.

GUESS. $\mathscr{A}$ outputs a guessing bit $b' \in \{0,1\}$.

$\mathscr{D}$ finally outputs 1 if and only if $b' = b$ holds (i.e., $\mathscr{A}$ wins).

Clearly, if $\mathbf{y}_i = \mathbf{B}_i\mathbf{s} + \mathbf{e}_i$, $\mathscr{D}$ simulates game G_8 perfectly for $\mathscr{A}$; if $\mathbf{y}_i \leftarrow_{\$} \mathbb{F}_2^n$, $\mathscr{D}$ simulates game G_9 perfectly for $\mathscr{A}$. Consequently, $\mathscr{D}$'s distinguishing advantage is at least $\big|\Pr_8[\mathsf{Win}] - \Pr_9[\mathsf{Win}]\big|$, which is negligible in n by Theorem 2. This completes the proof of Claim 8. ∎

Game G_{10}. This game is the same as G_9, except that, the oracle KEYGEN is changed as follows.

KEYGEN. $\mathscr{C}$ picks $b \leftarrow_{\$} \{0,1\}$ uniformly, and proceeds as follows.

(a) For each user $i \in [l]$, choose $\mathbf{A}_i \leftarrow_{\$} \mathcal{D}_\lambda^{n\times n}$, $\mathbf{P}_i \leftarrow_{\$} \mathcal{P}_n$, and $\mathbf{y}_i \leftarrow_{\$} \mathbb{F}_2^n$.

Finally, $\mathscr{C}$ sends the public keys $\mathsf{pk}_i := (\mathbf{A}_i, \mathbf{y}_i)$, $i \in [l]$, to $\mathscr{A}$.

Claim 9. $\Pr_9[\mathsf{Win}] = \Pr_{10}[\mathsf{Win}]$.

Proof of Claim 9. The proof is essentially the same as that for Claim 7. The key observation is that $\mathbf{A}_i = \mathbf{B}_i\mathbf{P}_i^{-1}$ in game G_9 is distributed according to $\mathcal{D}_\lambda^{n\times n}$ and independent of $\mathbf{P}_i$, the same as that in game G_{10}. Thus, this change is just conceptual, and $\Pr_9[\mathsf{Win}] = \Pr_{10}[\mathsf{Win}]$. ∎

Game G_{11}. This game is the same as G_{10}, except that, the oracle CHAL is changed as follows.

CHAL($j \in [l], f \in \mathcal{F}_{\mathit{aff}}$). $\mathscr{C}$ proceeds as follows.

(a) Set $f \leftarrow \mathbf{0}$ if $b = 0$. Then compute $\mathbf{T}_f := \sum_{i\in[l]} \mathbf{T}_i\mathbf{P}_i \in \mathbb{F}_2^{n\times n}$.

(b) Choose $\mathbf{U} \leftarrow_{\$} \mathbb{F}_2^{k\times n}$, $\mathbf{u} \leftarrow_{\$} \mathbb{F}_2^{k}$, and compute $\mathbf{C}_1 := \mathbf{U} - \mathbf{G}\mathbf{T}_f\mathbf{P}_j^{-1} \in \mathbb{F}_2^{k\times n}$ and $\mathbf{c}_2 := \mathbf{u} + \mathbf{G}\mathbf{t} \in \mathbb{F}_2^{k}$.

Finally, $\mathscr{C}$ returns the challenge ciphertext $\mathsf{c} := (\mathbf{C}_1, \mathbf{c}_2)$ to $\mathscr{A}$.

Claim 10. If $\mathsf{LPN}_{\mu,n}$ is $2^{\omega(n^{\frac{1}{2}})}$-hard, then $\left|\Pr_{10}[\mathsf{Win}] - \Pr_{11}[\mathsf{Win}]\right| \leq \mathsf{negl}(n)$.

Proof of Claim 10. Firstly, we introduce a sequence of intermediate games $\{\mathsf{G}_{10,\kappa}\}_{\kappa\in[Q+1]}$ between G_{10} and G_{11}.

- **Game $\mathsf{G}_{10,\kappa}$, $\kappa \in [Q+1]$.** This game is a hybrid of games G_{10} and G_{11}: for the first $\kappa - 1$ times of CHAL queries, $\mathscr{C}$ computes $\mathbf{C}_1$ and $\mathbf{c}_2$ as in game G_{11}; for the remaining CHAL queries, $\mathscr{C}$ computes $\mathbf{C}_1$ and $\mathbf{c}_2$ as in game G_{10}.

Clearly, game $\mathsf{G}_{10,1}$ is identical to G_{10} and game $\mathsf{G}_{10,Q+1}$ is identical to G_{11}. It suffices to show that $\left|\Pr_{10,\kappa}[\mathsf{Win}] - \Pr_{10,\kappa+1}[\mathsf{Win}]\right| \leq \mathsf{negl}(n)$ for any $\kappa \in [Q]$.

The only difference between game $\mathsf{G}_{10,\kappa}$ and game $\mathsf{G}_{10,\kappa+1}$ is the distribution of $\mathbf{C}_1$ and $\mathbf{c}_2$ in the κ-th CHAL($j \in [l], f \in \mathcal{F}_{\mathrm{aff}}$) query: in game $\mathsf{G}_{10,\kappa}$, $\mathbf{C}_1$ and $\mathbf{c}_2$ are computed according to game G_{10}, i.e., $\mathbf{C}_1 := \hat{\mathbf{S}}^\top\mathbf{A}_j + \hat{\mathbf{E}}^\top - \mathbf{G}\mathbf{T}_f\mathbf{P}_j^{-1}$ and $\mathbf{c}_2 = \hat{\mathbf{S}}^\top\mathbf{y}_j + \hat{\mathbf{e}} + \mathbf{G}\mathbf{t}$; in game $\mathsf{G}_{10,\kappa+1}$, they are computed according to game G_{11}, i.e., $\mathbf{C}_1 = \mathbf{U} - \mathbf{G}\mathbf{T}_f\mathbf{P}_j^{-1}$ and $\mathbf{c}_2 = \mathbf{u} + \mathbf{G}\mathbf{t}$.

We construct a PPT distinguisher $\mathscr{D}$ to solve the multi-fold LPN problem described in Lemma 6. Given a challenge $(\mathbf{A}, \mathbf{C}, \mathbf{y}, \mathbf{c})$, $\mathscr{D}$ wants to distinguish $\mathbf{C} = \hat{\mathbf{S}}^\top\mathbf{A} + \hat{\mathbf{E}}^\top$ and $\mathbf{c} = \hat{\mathbf{S}}^\top\mathbf{y} + \hat{\mathbf{e}}$ from $\mathbf{C} = \mathbf{U}$ and $\mathbf{c} = \mathbf{u}$, where $\mathbf{A} \leftarrow_{\$} \mathcal{D}_\lambda^{n\times n}$, $\hat{\mathbf{S}} \leftarrow_{\$} (\widetilde{\mathcal{B}}_{\mu_1}^n)^k$, $\hat{\mathbf{E}} \leftarrow_{\$} \mathcal{B}_\mu^{n\times k}$, $\mathbf{y} \leftarrow_{\$} \mathbb{F}_2^n$, $\hat{\mathbf{e}} \leftarrow_{\$} \mathcal{B}_\mu^k$, $\mathbf{U} \leftarrow_{\$} \mathbb{F}_2^{k\times n}$ and $\mathbf{u} \leftarrow_{\$} \mathbb{F}_2^k$. $\mathscr{D}$ is constructed by simulating game $\mathsf{G}_{10,\kappa}$ or $\mathsf{G}_{10,\kappa+1}$ for $\mathscr{A}$ as follows, where we highlight the challenge received by $\mathscr{D}$.

KEYGEN. $\mathscr{D}$ picks $b \leftarrow_{\$} \{0,1\}$ uniformly, and proceeds as follows.

(a) Pick $j^* \leftarrow_{\$} [l]$. For each user $i \in [l]$,
  - if $i \neq j^*$, choose $\mathbf{A}_i \leftarrow_{\$} \mathcal{D}_\lambda^{n\times n}$, $\mathbf{P}_i \leftarrow_{\$} \mathcal{P}_n$, and $\mathbf{y}_i \leftarrow_{\$} \mathbb{F}_2^n$;
  - if $i = j^*$, set $\mathbf{A}_{j^*} := \mathbf{A}$, $\mathbf{y}_{j^*} := \mathbf{y}$, and choose $\mathbf{P}_{j^*} \leftarrow_{\$} \mathcal{P}_n$.

Finally, $\mathscr{D}$ sends the public keys $\mathsf{pk}_i := (\mathbf{A}_i, \mathbf{y}_i)$, $i \in [l]$, to $\mathscr{A}$.

CHAL($j \in [l], f \in \mathcal{F}_{\mathit{aff}}$). $\mathscr{D}$ proceeds as follows.

(a) Set $f \leftarrow \mathbf{0}$ if $b = 0$. Then compute $\mathbf{T}_f := \sum_{i\in[l]} \mathbf{T}_i\mathbf{P}_i \in \mathbb{F}_2^{n\times n}$.

(b)
  - For the first $\kappa - 1$ queries, $\mathscr{D}$ computes $\mathbf{C}_1$ and $\mathbf{c}_2$ according to game G_{11}.
  - For the κ-th query, $\mathscr{D}$ aborts immediately if $j \neq j^*$; otherwise $\mathscr{D}$ computes $\mathbf{C}_1 := \mathbf{C} - \mathbf{G}\mathbf{T}_f\mathbf{P}_{j^*}^{-1}$ and $\mathbf{c}_2 := \mathbf{c} + \mathbf{G}\mathbf{t}$.
  - For the remaining queries, $\mathscr{D}$ computes $\mathbf{C}_1$ and $\mathbf{c}_2$ according to game G_{10}.

Finally, $\mathscr{D}$ returns the challenge ciphertext $\mathsf{c} := (\mathbf{C}_1, \mathbf{c}_2)$ to $\mathscr{A}$.

GUESS. $\mathscr{A}$ outputs a guessing bit $b' \in \{0,1\}$.

$\mathscr{D}$ finally outputs 1 if and only if $j = j^*$ holds in the i-th CHAL query (i.e., $\mathscr{D}$ does not abort) and $b' = b$.

Next, we analyze the distinguishing advantage of $\mathscr{D}$.

- In KeyGen, $\mathbf{A}_{j^*}$ and $\mathbf{y}_{j^*}$ have the same distributions as in both game $\mathsf{G}_{10,\kappa}$ and game $\mathsf{G}_{10,\kappa+1}$. Besides, j^* is completely hidden from $\mathscr{A}$'s view.
- In the κ-th query of Chal$(j \in [l], f \in \mathcal{F}_{\text{aff}})$, $j = j^*$ holds with probability at least $1/l$.
  - If $\mathbf{C} = \hat{\mathbf{S}}^\top \mathbf{A} + \hat{\mathbf{E}}^\top$ and $\mathbf{c} = \hat{\mathbf{S}}^\top \mathbf{y} + \hat{\mathbf{e}}$, then $\mathbf{C}_1 = \hat{\mathbf{S}}^\top \mathbf{A} + \hat{\mathbf{E}}^\top - \mathbf{G}\mathbf{T}_f \mathbf{P}_{j^*}^{-1} = \hat{\mathbf{S}}^\top \mathbf{A}_{j^*} + \hat{\mathbf{E}}^\top - \mathbf{G}\mathbf{T}_f \mathbf{P}_{j^*}^{-1}$ and $\mathbf{c}_2 = \hat{\mathbf{S}}^\top \mathbf{y} + \hat{\mathbf{e}} + \mathbf{G}\mathbf{t} = \hat{\mathbf{S}}^\top \mathbf{y}_{j^*} + \hat{\mathbf{e}} + \mathbf{G}\mathbf{t}$. Thus, $\mathscr{D}$ computes $(\mathbf{C}_1, \mathbf{c}_2)$ for the κ-th Chal query exactly like game G_{10}.
  - If $\mathbf{C} = \mathbf{U}$ and $\mathbf{c} = \mathbf{u}$, then $\mathbf{C}_1 = \mathbf{U} - \mathbf{G}\mathbf{T}_f \mathbf{P}_{j^*}^{-1}$ and $\mathbf{c}_2 = \mathbf{u} + \mathbf{G}\mathbf{t}$. Thus, $\mathscr{D}$ computes $(\mathbf{C}_1, \mathbf{c}_2)$ for the κ-th Chal query exactly like game G_{11}.

Therefore, if $\mathscr{D}$ does not abort (which occurs with probability at least $1/l$), $\mathscr{D}$ simulates game $\mathsf{G}_{10,\kappa}$ perfectly for $\mathscr{A}$ in the case of $\mathbf{C} = \hat{\mathbf{S}}^\top \mathbf{A} + \hat{\mathbf{E}}^\top$ and $\mathbf{c} = \hat{\mathbf{S}}^\top \mathbf{y} + \hat{\mathbf{e}}$, and simulates game $\mathsf{G}_{10,\kappa+1}$ perfectly for $\mathscr{A}$ in the case of $\mathbf{C} = \mathbf{U}$ and $\mathbf{c} = \mathbf{u}$. Consequently, $\mathscr{D}$'s distinguishing advantage is at least $\frac{1}{l} \cdot \big| \Pr_{10,\kappa}[\mathsf{Win}] - \Pr_{10,\kappa+1}[\mathsf{Win}] \big|$, which is $\mathsf{negl}(n)$ by Lemma 6.

In conclusion, $\big| \Pr_{10}[\mathsf{Win}] - \Pr_{11}[\mathsf{Win}] \big| = \big| \Pr_{10,1}[\mathsf{Win}] - \Pr_{10,Q+1}[\mathsf{Win}] \big| \leq \sum_{\kappa \in [Q]} \big| \Pr_{10,\kappa}[\mathsf{Win}] - \Pr_{10,\kappa+1}[\mathsf{Win}] \big| \leq Ql \cdot \mathsf{negl}(n)$, which is also negligible in n. This completes the proof of Claim 10. ▌

Game G_{12}. This game is the same as G_{11}, except that, the oracle Chal is changed as follows.

Chal$(j \in [l], f \in \mathcal{F}_{aff})$. $\mathscr{C}$ proceeds as follows.

(a) Choose $\mathbf{U} \leftarrow_\$ \mathbb{F}_2^{k \times n}$, $\mathbf{u} \leftarrow_\$ \mathbb{F}_2^k$, and compute $\mathbf{C}_1 := \mathbf{U} \in \mathbb{F}_2^{k \times n}$ and $\mathbf{c}_2 := \mathbf{u} \in \mathbb{F}_2^k$.

Finally, $\mathscr{C}$ returns the challenge ciphertext $\mathsf{c} := (\mathbf{C}_1, \mathbf{c}_2)$ to $\mathscr{A}$.

Claim 11. $\Pr_{11}[\mathsf{Win}] = \Pr_{12}[\mathsf{Win}] = \frac{1}{2}$.

Proof of Claim 11. Since $\mathbf{U}$ and $\mathbf{u}$ are uniformly chosen and independent of other parts of the game, $\mathbf{C}_1 = \mathbf{U} - \mathbf{G}\mathbf{T}_f \mathbf{P}_j^{-1}$ and $\mathbf{C}_2 = \mathbf{u} + \mathbf{G}\mathbf{t}$ in game G_{11} have the same distributions as $\mathbf{C}_1 = \mathbf{U}$ and $\mathbf{C}_2 = \mathbf{u}$ in game G_{12}, respectively. Therefore, the changes are just conceptual, and $\Pr_{11}[\mathsf{Win}] = \Pr_{12}[\mathsf{Win}]$.

Moreover, the challenge bit b is never used in game G_{12}, thus completely hidden from $\mathscr{A}$'s view. Consequently, we have $\Pr_{12}[\mathsf{Win}] = \frac{1}{2}$. ▌

Taking all things together, by Claim 1-11, it follows that $\epsilon = \big| \Pr_1[\mathsf{Win}] - \frac{1}{2} \big| \leq \mathsf{negl}(n)$. This completes the proof of Theorem 4. ▌

Acknowledgments. We thank the anonymous reviewers for their comments and suggestions. The authors are supported by the National Natural Science Foundation of China Grant (Nos. 61672346, 61373153).

References

[ABBC10] Acar, T., Belenkiy, M., Bellare, M., Cash, D.: Cryptographic agility and its relation to circular encryption. In: Gilbert, H. (ed.) EUROCRYPT 2010, LNCS, vol. 6110, pp. 403–422. Springer (2010)

[ACPS09] Applebaum, B., Cash, D., Peikert, C., Sahai, A.: Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In: Halevi, S. (ed.) CRYPTO 2009, LNCS, vol. 5677, pp. 595–618. Springer (2009)

[Ale03] Alekhnovich, M.: More on average case vs approximation complexity. In: FOCS 2003, pp. 298–307. IEEE Computer Society (2003)

[AP16] Alamati, N., Peikert, C.: Three's compromised too: Circular insecurity for any cycle length from (Ring-)LWE. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part II, LNCS, vol. 9815, pp. 659–680. Springer (2016)

[App11] Applebaum, B.: Key-dependent message security: Generic amplification and completeness. In: Paterson, K.G. (ed.) EUROCRYPT 2011, LNCS, vol. 6632, pp. 527–546. Springer (2011)

[BFKL93] Blum, A., Furst, M.L., Kearns, M.J., Lipton, R.J.: Cryptographic primitives based on hard learning problems. In: Stinson, D.R. (ed.) CRYPTO 1993, LNCS, vol. 773, pp. 278–291. Springer (1993)

[BG10] Brakerski, Z., Goldwasser, S.: Circular and leakage resilient public-key encryption under subgroup indistinguishability - (or: Quadratic residuosity strikes back). In: Rabin, T. (ed.) CRYPTO 2010, LNCS, vol. 6223, pp. 1–20. Springer (2010)

[BGK11] Brakerski, Z., Goldwasser, S., Kalai, Y.T.: Black-box circular-secure encryption beyond affine functions. In: Ishai, Y. (ed.) TCC 2011, LNCS, vol. 6597, pp. 201–218. Springer (2011)

[BHHI10] Barak, B., Haitner, I., Hofheinz, D., Ishai, Y.: Bounded key-dependent message security. In: Gilbert, H. (ed.) EUROCRYPT 2010, LNCS, vol. 6110, pp. 423–444. Springer (2010)

[BHHO08] Boneh, D., Halevi, S., Hamburg, M., Ostrovsky, R.: Circular-secure encryption from decision Diffie-Hellman. In: Wagner, D. (ed.) CRYPTO 2008, LNCS, vol. 5157, pp. 108–125. Springer (2008)

[BHW15] Bishop, A., Hohenberger, S., Waters, B.: New circular security counterexamples from decision linear and learning with errors. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015, Part II, LNCS, vol. 9453, pp. 776–800. Springer (2015)

[BJMM12] Becker, A., Joux, A., May, A., Meurer, A.: Decoding random binary linear codes in $2^{n/20}$: How $1+1=0$ improves information set decoding. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012, LNCS, vol. 7237, pp. 520–536. Springer (2012)

[BKW03] Blum, A., Kalai, A., Wasserman, H.: Noise-tolerant learning, the parity problem, and the statistical query model. Journal of the ACM vol. 50(4), pp. 506–519 (2003)

[BLP11] Bernstein, D.J., Lange, T., Peters, C.: Smaller decoding exponents: Ball-collision decoding. In: Rogaway, P. (ed.) CRYPTO 2011, LNCS, vol. 6841, pp. 743–760. Springer (2011)

[BMT78] Berlekamp, E.R., McEliece, R.J., van Tilborg, H.C.A.: On the inherent intractability of certain coding problems. IEEE Transactions on Information Theory vol. 24(3), pp. 384–386 (1978)

[BRS02] Black, J., Rogaway, P., Shrimpton, T.: Encryption-scheme security in the presence of key-dependent messages. In: Nyberg, K., Heys, H.M. (eds.) SAC 2002, LNCS, vol. 2595, pp. 62–75. Springer (2002)

[CC98] Canteaut, A., Chabaud, F.: A new algorithm for finding minimum-weight words in a linear code: Application to mceliece's cryptosystem and to narrow-sense BCH codes of length 511. IEEE Transactions on Information Theory vol. 44(1), pp. 367–378 (1998)

[CCS09] Camenisch, J., Chandran, N., Shoup, V.: A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. In: Joux, A. (ed.) EUROCRYPT 2009, LNCS, vol. 5479, pp. 351–368. Springer (2009)

[CGH12] Cash, D., Green, M., Hohenberger, S.: New definitions and separations for circular security. In: Fischlin, M., Buchmann, J.A., Manulis, M. (eds.) PKC 2012, LNCS, vol. 7293, pp. 540–557. Springer (2012)

[CL01] Camenisch, J., Lysyanskaya, A.: An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In: Pfitzmann, B. (ed.) EUROCRYPT 2001, LNCS, vol. 2045, pp. 93–118. Springer (2001)

[DDN14] David, B., Dowsley, R., Nascimento, A.C.A.: Universally composable oblivious transfer based on a variant of LPN. In: Gritzalis, D., Kiayias, A., Askoxylakis, I.G. (eds.) CANS 2014, LNCS, vol. 8813, pp. 143–158. Springer (2014)

[DMN12] Döttling, N., Müller-Quade, J., Nascimento, A.C.A.: IND-CCA secure cryptography based on a variant of the LPN problem. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012, LNCS, vol. 7658, pp. 485–503. Springer (2012)

[Döt15] Döttling, N.: Low noise LPN: KDM secure public key encryption and sample amplification. In: Katz, J. (ed.) PKC 2015, LNCS, vol. 9020, pp. 604–626. Springer (2015)

[FGKP06] Feldman, V., Gopalan, P., Khot, S., Ponnuswami, A.K.: New results for learning noisy parities and halfspaces. In: FOCS 2006, pp. 563–574. IEEE Computer Society (2006)

[For66] Forney, G.D.: Concatenated codes. MIT Press (1966)

[GHV12] Galindo, D., Herranz, J., Villar, J.L.: Identity-based encryption with master key-dependent message security and leakage-resilience. In: Foresti, S., Yung, M., Martinelli, F. (eds.) ESORICS 2012, LNCS, vol. 7459, pp. 627–642. Springer (2012)

[GKW17] Goyal, R., Koppula, V., Waters, B.: Separating IND-CPA and circular security for unbounded length key cycles. In: Fehr, S. (ed.) PKC 2017, Part I, LNCS, vol. 10174, pp. 232–246. Springer (2017)
[GM84] Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of Computer and System Sciences vol. 28(2), pp. 270–299 (1984)
[HILL99] Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM J. Comput. vol. 28(4), pp. 1364–1396 (1999)
[HLL16] Han, S., Liu, S., Lyu, L.: Efficient KDM-CCA secure public-key encryption for polynomial functions. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, Part II, LNCS, vol. 10032, pp. 307–338 (2016)
[Hof13] Hofheinz, D.: Circular chosen-ciphertext security with compact ciphertexts. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013, LNCS, vol. 7881, pp. 520–536. Springer (2013)
[Kir11] Kirchner, P.: Improved generalized birthday attack. IACR Cryptology ePrint Archive, Report 2011/377 (2011)
[KMP14] Kiltz, E., Masny, D., Pietrzak, K.: Simple chosen-ciphertext security from low-noise LPN. In: Krawczyk, H. (ed.) PKC 2014, LNCS, vol. 8383, pp. 1–18. Springer (2014)
[KRW15] Koppula, V., Ramchen, K., Waters, B.: Separations in circular security for arbitrary length key cycles. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015, Part II, LNCS, vol. 9015, pp. 378–400. Springer (2015)
[KS06] Katz, J., Shin, J.S.: Parallel and concurrent security of the HB and HB^+ protocols. In: Vaudenay, S. (ed.) EUROCRYPT 2006, LNCS, vol. 4004, pp. 73–87. Springer (2006)
[KW16] Koppula, V., Waters, B.: Circular security separations for arbitrary length cycles from LWE. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part II, LNCS, vol. 9815, pp. 681–700. Springer (2016)
[LF06] Levieil, É., Fouque, P.: An improved LPN algorithm. In: Prisco, R.D., Yung, M. (eds.) SCN 2006, LNCS, vol. 4116, pp. 348–359. Springer (2006)
[LLJ15] Lu, X., Li, B., Jia, D.: KDM-CCA security from RKA secure authenticated encryption. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part I, LNCS, vol. 9056, pp. 559–583. Springer (2015)
[Lyu05] Lyubashevsky, V.: The parity problem in the presence of noise, decoding random linear codes, and the subset sum problem. In: Chekuri, C., Jansen, K., Rolim, J.D.P., Trevisan, L. (eds.) APPROX 2005 and RANDOM 2005, LNCS, vol. 3624, pp. 378–389. Springer (2005)
[Mat93] Matsui, M.: Linear cryptanalysis method for DES cipher. In: Helleseth, T. (ed.) EUROCRYPT 1993, LNCS, vol. 765, pp. 386–397. Springer (1993)
[MMT11] May, A., Meurer, A., Thomae, E.: Decoding random linear codes in $\tilde{O}(2^{0.054n})$. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011, LNCS, vol. 7073, pp. 107–124. Springer (2011)
[MO14] Marcedone, A., Orlandi, C.: Obfuscation ⇒ (IND-CPA security !⇒ circular security). In: Abdalla, M., Prisco, R.D. (eds.) SCN 2014, LNCS, vol. 8642, pp. 77–90. Springer (2014)
[MTY11] Malkin, T., Teranishi, I., Yung, M.: Efficient circuit-size independent public key encryption with KDM security. In: Paterson, K.G. (ed.) EUROCRYPT 2011, LNCS, vol. 6632, pp. 507–526. Springer (2011)
[Reg05] Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Gabow, H.N., Fagin, R. (eds.) STOC 2005, pp. 84–93. ACM (2005)
[Ste88] Stern, J.: A method for finding codewords of small weight. In: Cohen, G.D., Wolfmann, J. (eds.) Coding Theory and Applications 1988, LNCS, vol. 388, pp. 106–113. Springer (1988)
[YZ16] Yu, Y., Zhang, J.: Cryptography with auxiliary input and trapdoor from constant-noise LPN. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part I, LNCS, vol. 9814, pp. 214–243. Springer (2016)

A Omitted Figures and Proofs in the Proof of Theorem 4

A.1 Figures for Proof of Theorem 4

Game G_1	$\underline{\textsc{KeyGen:}}$ $b \leftarrow_\$ \{0,1\}$. // challenge bit For $i \in [l]$, $\mathbf{A}_i \leftarrow_\$ \mathcal{D}_\lambda^{n\times n}$. $\mathbf{s}_i \leftarrow_\$ \chi^n_{\mu_1 n}$. $\mathbf{e}_i \leftarrow_\$ \mathcal{B}^n_\mu$. $\mathbf{y}_i := \mathbf{A}_i\mathbf{s}_i + \mathbf{e}_i \in \mathbb{F}_2^n$. $\mathsf{pk}_i := (\mathbf{A}_i, \mathbf{y}_i)$, $\mathsf{sk}_i := \mathbf{s}_i \in \mathbb{F}_2^n$. Return $(\mathsf{pk}_1, \cdots, \mathsf{pk}_l)$.	$\underline{\textsc{Chal}(j \in [l], f \in \mathcal{F}_{\text{aff}}):}$ If $b = 0$, $f \leftarrow \mathbf{0}$. $\mathsf{m} := f(\mathsf{sk}_1, \cdots, \mathsf{sk}_l) = \sum_{i\in[l]} \mathbf{T}_i\mathbf{s}_i + \mathbf{t} \in \mathbb{F}_2^n$. $\hat{\mathbf{S}} \leftarrow_\$ (\widetilde{\mathcal{B}}^n_{\mu_1})^k$. $\hat{\mathbf{E}} \leftarrow_\$ \mathcal{B}^{n\times k}_\mu$. $\hat{\mathbf{e}} \leftarrow_\$ \mathcal{B}^k_\mu$. $\mathbf{C}_1 := \hat{\mathbf{S}}^\top\mathbf{A}_j + \hat{\mathbf{E}}^\top \in \mathbb{F}_2^{k\times n}$. $\mathbf{c}_2 := \hat{\mathbf{S}}^\top\mathbf{y}_j + \hat{\mathbf{e}} + \mathbf{G}\mathsf{m} \in \mathbb{F}_2^k$. Return $\mathsf{c} := (\mathbf{C}_1, \mathbf{c}_2)$.
Game G_2	$\underline{\textsc{KeyGen:}}$ $b \leftarrow_\$ \{0,1\}$. // challenge bit $\underline{\mathbf{s}^* \leftarrow_\$ \chi^n_{\mu_1 n}.}$ For $i \in [l]$, $\mathbf{A}_i \leftarrow_\$ \mathcal{D}_\lambda^{n\times n}$. $\underline{\mathbf{P}_i \leftarrow_\$ \mathcal{P}_n.}$ $\mathbf{e}_i \leftarrow_\$ \mathcal{B}^n_\mu$. $\underline{\mathbf{s}_i := \mathbf{P}_i\mathbf{s}^* \in \mathbb{F}_2.}$ $\underline{\mathbf{y}_i := \mathbf{A}_i\mathbf{P}_i\mathbf{s}^* + \mathbf{e}_i \in \mathbb{F}_2^n.}$ $\mathsf{pk}_i := (\mathbf{A}_i, \mathbf{y}_i)$, $\mathsf{sk}_i := \mathbf{s}_i \in \mathbb{F}_2^n$. Return $(\mathsf{pk}_1, \cdots, \mathsf{pk}_l)$.	$\underline{\textsc{Chal}(j \in [l], f \in \mathcal{F}_{\text{aff}}):}$ If $b = 0$, $f \leftarrow \mathbf{0}$. $\mathsf{m} := f(\mathsf{sk}_1, \cdots, \mathsf{sk}_l) = \sum_{i\in[l]} \mathbf{T}_i\mathbf{s}_i + \mathbf{t} \in \mathbb{F}_2^n$. $\hat{\mathbf{S}} \leftarrow_\$ (\widetilde{\mathcal{B}}^n_{\mu_1})^k$. $\hat{\mathbf{E}} \leftarrow_\$ \mathcal{B}^{n\times k}_\mu$. $\hat{\mathbf{e}} \leftarrow_\$ \mathcal{B}^k_\mu$. $\mathbf{C}_1 := \hat{\mathbf{S}}^\top\mathbf{A}_j + \hat{\mathbf{E}}^\top \in \mathbb{F}_2^{k\times n}$. $\mathbf{c}_2 := \hat{\mathbf{S}}^\top\mathbf{y}_j + \hat{\mathbf{e}} + \mathbf{G}\mathsf{m} \in \mathbb{F}_2^k$. Return $\mathsf{c} := (\mathbf{C}_1, \mathbf{c}_2)$.
Game G_3	$\underline{\textsc{KeyGen:}}$ $b \leftarrow_\$ \{0,1\}$. // challenge bit $\mathbf{s}^* \leftarrow_\$ \chi^n_{\mu_1 n}$. For $i \in [l]$, $\mathbf{A}_i \leftarrow_\$ \mathcal{D}_\lambda^{n\times n}$. $\mathbf{P}_i \leftarrow_\$ \mathcal{P}_n$. $\mathbf{e}_i \leftarrow_\$ \mathcal{B}^n_\mu$. $\mathbf{y}_i := \mathbf{A}_i\mathbf{P}_i\mathbf{s}^* + \mathbf{e}_i \in \mathbb{F}_2^n$. $\mathsf{pk}_i := (\mathbf{A}_i, \mathbf{y}_i)$. Return $(\mathsf{pk}_1, \cdots, \mathsf{pk}_l)$.	$\underline{\textsc{Chal}(j \in [l], f \in \mathcal{F}_{\text{aff}}):}$ If $b = 0$, $f \leftarrow \mathbf{0}$. $\underline{\mathbf{T}_f := \sum_{i\in[l]} \mathbf{T}_i\mathbf{P}_i \in \mathbb{F}_2^{n\times n}.}$ $\underline{\mathsf{m} := \mathbf{T}_f\mathbf{s}^* + \mathbf{t} \in \mathbb{F}_2^n.}$ $\hat{\mathbf{S}} \leftarrow_\$ (\widetilde{\mathcal{B}}^n_{\mu_1})^k$. $\hat{\mathbf{E}} \leftarrow_\$ \mathcal{B}^{n\times k}_\mu$. $\hat{\mathbf{e}} \leftarrow_\$ \mathcal{B}^k_\mu$. $\mathbf{C}_1 := \hat{\mathbf{S}}^\top\mathbf{A}_j + \hat{\mathbf{E}}^\top \in \mathbb{F}_2^{k\times n}$. $\underline{\mathbf{c}_2 := (\mathbf{C}_1\mathbf{P}_j + \mathbf{G}\mathbf{T}_f)\cdot\mathbf{s}^* - \hat{\mathbf{E}}^\top\mathbf{P}_j\mathbf{s}^* + \hat{\mathbf{S}}^\top\mathbf{e}_j + \hat{\mathbf{e}} + \mathbf{G}\mathbf{t} \in \mathbb{F}_2^k.}$ Return $\mathsf{c} := (\mathbf{C}_1, \mathbf{c}_2)$.
Game G_4	$\underline{\textsc{KeyGen:}}$ $b \leftarrow_\$ \{0,1\}$. // challenge bit $\mathbf{s}^* \leftarrow_\$ \chi^n_{\mu_1 n}$. For $i \in [l]$, $\mathbf{A}_i \leftarrow_\$ \mathcal{D}_\lambda^{n\times n}$. $\mathbf{P}_i \leftarrow_\$ \mathcal{P}_n$. $\mathbf{e}_i \leftarrow_\$ \mathcal{B}^n_\mu$. $\mathbf{y}_i := \mathbf{A}_i\mathbf{P}_i\mathbf{s}^* + \mathbf{e}_i \in \mathbb{F}_2^n$. $\mathsf{pk}_i := (\mathbf{A}_i, \mathbf{y}_i)$. Return $(\mathsf{pk}_1, \cdots, \mathsf{pk}_l)$.	$\underline{\textsc{Chal}(j \in [l], f \in \mathcal{F}_{\text{aff}}):}$ If $b = 0$, $f \leftarrow \mathbf{0}$. $\mathbf{T}_f := \sum_{i\in[l]} \mathbf{T}_i\mathbf{P}_i \in \mathbb{F}_2^{n\times n}$. $\hat{\mathbf{S}} \leftarrow_\$ (\widetilde{\mathcal{B}}^n_{\mu_1})^k$. $\hat{\mathbf{E}} \leftarrow_\$ \mathcal{B}^{n\times k}_\mu$. $\hat{\mathbf{e}} \leftarrow_\$ \mathcal{B}^k_\mu$. $\underline{\mathbf{U} \leftarrow_\$ \mathbb{F}_2^{k\times n}.}$ $\underline{\mathbf{C}_1 := \mathbf{U} \in \mathbb{F}_2^{k\times n}.}$ $\mathbf{c}_2 := (\mathbf{C}_1\mathbf{P}_j + \mathbf{G}\mathbf{T}_f)\cdot\mathbf{s}^* - \hat{\mathbf{E}}^\top\mathbf{P}_j\mathbf{s}^* + \hat{\mathbf{S}}^\top\mathbf{e}_j + \hat{\mathbf{e}} + \mathbf{G}\mathbf{t} \in \mathbb{F}_2^k$. Return $\mathsf{c} := (\mathbf{C}_1, \mathbf{c}_2)$.
Game G_5	$\underline{\textsc{KeyGen:}}$ $b \leftarrow_\$ \{0,1\}$. // challenge bit $\mathbf{s}^* \leftarrow_\$ \chi^n_{\mu_1 n}$. For $i \in [l]$, $\mathbf{A}_i \leftarrow_\$ \mathcal{D}_\lambda^{n\times n}$. $\mathbf{P}_i \leftarrow_\$ \mathcal{P}_n$. $\mathbf{e}_i \leftarrow_\$ \mathcal{B}^n_\mu$. $\mathbf{y}_i := \mathbf{A}_i\mathbf{P}_i\mathbf{s}^* + \mathbf{e}_i \in \mathbb{F}_2^n$. $\mathsf{pk}_i := (\mathbf{A}_i, \mathbf{y}_i)$. Return $(\mathsf{pk}_1, \cdots, \mathsf{pk}_l)$.	$\underline{\textsc{Chal}(j \in [l], f \in \mathcal{F}_{\text{aff}}):}$ If $b = 0$, $f \leftarrow \mathbf{0}$. $\mathbf{T}_f := \sum_{i\in[l]} \mathbf{T}_i\mathbf{P}_i \in \mathbb{F}_2^{n\times n}$. $\hat{\mathbf{S}} \leftarrow_\$ (\widetilde{\mathcal{B}}^n_{\mu_1})^k$. $\hat{\mathbf{E}} \leftarrow_\$ \mathcal{B}^{n\times k}_\mu$. $\hat{\mathbf{e}} \leftarrow_\$ \mathcal{B}^k_\mu$. $\mathbf{U} \leftarrow_\$ \mathbb{F}_2^{k\times n}$. $\underline{\mathbf{C}_1 := \mathbf{U} - \mathbf{G}\mathbf{T}_f\mathbf{P}_j^{-1} \in \mathbb{F}_2^{k\times n}.}$ $\mathbf{c}_2 := (\mathbf{C}_1\mathbf{P}_j + \mathbf{G}\mathbf{T}_f)\cdot\mathbf{s}^* - \hat{\mathbf{E}}^\top\mathbf{P}_j\mathbf{s}^* + \hat{\mathbf{S}}^\top\mathbf{e}_j + \hat{\mathbf{e}} + \mathbf{G}\mathbf{t} \in \mathbb{F}_2^k$. Return $\mathsf{c} := (\mathbf{C}_1, \mathbf{c}_2)$.
Game G_6	$\underline{\textsc{KeyGen:}}$ $b \leftarrow_\$ \{0,1\}$. // challenge bit $\mathbf{s}^* \leftarrow_\$ \chi^n_{\mu_1 n}$. For $i \in [l]$, $\mathbf{A}_i \leftarrow_\$ \mathcal{D}_\lambda^{n\times n}$. $\mathbf{P}_i \leftarrow_\$ \mathcal{P}_n$. $\mathbf{e}_i \leftarrow_\$ \mathcal{B}^n_\mu$. $\mathbf{y}_i := \mathbf{A}_i\mathbf{P}_i\mathbf{s}^* + \mathbf{e}_i \in \mathbb{F}_2^n$. $\mathsf{pk}_i := (\mathbf{A}_i, \mathbf{y}_i)$. Return $(\mathsf{pk}_1, \cdots, \mathsf{pk}_l)$.	$\underline{\textsc{Chal}(j \in [l], f \in \mathcal{F}_{\text{aff}}):}$ If $b = 0$, $f \leftarrow \mathbf{0}$. $\mathbf{T}_f := \sum_{i\in[l]} \mathbf{T}_i\mathbf{P}_i \in \mathbb{F}_2^{n\times n}$. $\hat{\mathbf{S}} \leftarrow_\$ (\widetilde{\mathcal{B}}^n_{\mu_1})^k$. $\hat{\mathbf{E}} \leftarrow_\$ \mathcal{B}^{n\times k}_\mu$. $\hat{\mathbf{e}} \leftarrow_\$ \mathcal{B}^k_\mu$. $\underline{\mathbf{C}_1 := \hat{\mathbf{S}}^\top\mathbf{A}_j + \hat{\mathbf{E}}^\top - \mathbf{G}\mathbf{T}_f\mathbf{P}_j^{-1} \in \mathbb{F}_2^{k\times n}.}$ $\mathbf{c}_2 := (\mathbf{C}_1\mathbf{P}_j + \mathbf{G}\mathbf{T}_f)\cdot\mathbf{s}^* - \hat{\mathbf{E}}^\top\mathbf{P}_j\mathbf{s}^* + \hat{\mathbf{S}}^\top\mathbf{e}_j + \hat{\mathbf{e}} + \mathbf{G}\mathbf{t} \in \mathbb{F}_2^k$. Return $\mathsf{c} := (\mathbf{C}_1, \mathbf{c}_2)$.

Fig. 2. Games G_1–G_6 for l-KDM[$\mathcal{F}_{\text{aff}}$]-CPA security of PKE (see also Fig. 3).

Game G_7	<u>KeyGen:</u> $b \leftarrow\$ \{0,1\}$. // challenge bit $\mathbf{s}^* \leftarrow\$ \chi^n_{\mu_1 n}$. For $i \in [l]$, $\quad \mathbf{A}_i \leftarrow\$ \mathcal{D}_\lambda^{n\times n}$. $\mathbf{P}_i \leftarrow\$ \mathcal{P}_n$. $\mathbf{e}_i \leftarrow\$ \mathcal{B}_\mu^n$. $\quad \mathbf{y}_i := \mathbf{A}_i\mathbf{P}_i\mathbf{s}^* + \mathbf{e}_i \in \mathbb{F}_2^n$. $\quad \mathsf{pk}_i := (\mathbf{A}_i, \mathbf{y}_i)$. Return $(\mathsf{pk}_1, \cdots, \mathsf{pk}_l)$.	<u>Chal($j \in [l], f \in \mathcal{F}_{\text{aff}}$):</u> If $b = 0$, $\quad f \leftarrow \mathbf{0}$. $\mathbf{T}_f := \sum_{i\in[l]} \mathbf{T}_i\mathbf{P}_i \in \mathbb{F}_2^{n\times n}$. $\hat{\mathbf{S}} \leftarrow\$ (\widetilde{\mathcal{B}}^n_{\mu_1})^k$. $\hat{\mathbf{E}} \leftarrow\$ \mathcal{B}_\mu^{n\times k}$. $\hat{\mathbf{e}} \leftarrow\$ \mathcal{B}_\mu^k$. $\mathbf{C}_1 := \hat{\mathbf{S}}^\top\mathbf{A}_j + \hat{\mathbf{E}}^\top - \mathbf{G}\mathbf{T}_f\mathbf{P}_j^{-1} \in \mathbb{F}_2^{k\times n}$. $\mathbf{c}_2 := \hat{\mathbf{S}}^\top\mathbf{y}_j + \hat{\mathbf{e}} + \mathbf{G}\mathbf{t} \in \mathbb{F}_2^k$. Return $\mathsf{c} := (\mathbf{C}_1, \mathbf{c}_2)$.
Game G_8	<u>KeyGen:</u> $b \leftarrow\$ \{0,1\}$. // challenge bit $\mathbf{s}^* \leftarrow\$ \chi^n_{\mu_1 n}$. For $i \in [l]$, $\quad \mathbf{B}_i \leftarrow\$ \mathcal{D}_\lambda^{n\times n}$. $\mathbf{P}_i \leftarrow\$ \mathcal{P}_n$. $\mathbf{e}_i \leftarrow\$ \mathcal{B}_\mu^n$. $\quad \mathbf{A}_i := \mathbf{B}_i\mathbf{P}_i^{-1} \in \mathbb{F}_2^{n\times n}$. $\mathbf{y}_i := \mathbf{B}_i\mathbf{s}^* + \mathbf{e}_i \in \mathbb{F}_2^n$. $\quad \mathsf{pk}_i := (\mathbf{A}_i, \mathbf{y}_i)$. Return $(\mathsf{pk}_1, \cdots, \mathsf{pk}_l)$.	<u>Chal($j \in [l], f \in \mathcal{F}_{\text{aff}}$):</u> If $b = 0$, $\quad f \leftarrow \mathbf{0}$. $\mathbf{T}_f := \sum_{i\in[l]} \mathbf{T}_i\mathbf{P}_i \in \mathbb{F}_2^{n\times n}$. $\hat{\mathbf{S}} \leftarrow\$ (\widetilde{\mathcal{B}}^n_{\mu_1})^k$. $\hat{\mathbf{E}} \leftarrow\$ \mathcal{B}_\mu^{n\times k}$. $\hat{\mathbf{e}} \leftarrow\$ \mathcal{B}_\mu^k$. $\mathbf{C}_1 := \hat{\mathbf{S}}^\top\mathbf{A}_j + \hat{\mathbf{E}}^\top - \mathbf{G}\mathbf{T}_f\mathbf{P}_j^{-1} \in \mathbb{F}_2^{k\times n}$. $\mathbf{c}_2 := \hat{\mathbf{S}}^\top\mathbf{y}_j + \hat{\mathbf{e}} + \mathbf{G}\mathbf{t} \in \mathbb{F}_2^k$. Return $\mathsf{c} := (\mathbf{C}_1, \mathbf{c}_2)$.
Game G_9	<u>KeyGen:</u> $b \leftarrow\$ \{0,1\}$. // challenge bit For $i \in [l]$, $\quad \mathbf{B}_i \leftarrow\$ \mathcal{D}_\lambda^{n\times n}$. $\mathbf{P}_i \leftarrow\$ \mathcal{P}_n$. $\quad \mathbf{A}_i := \mathbf{B}_i\mathbf{P}_i^{-1} \in \mathbb{F}_2^{n\times n}$. $\mathbf{y}_i \leftarrow\$ \mathbb{F}_2^n$. $\quad \mathsf{pk}_i := (\mathbf{A}_i, \mathbf{y}_i)$. Return $(\mathsf{pk}_1, \cdots, \mathsf{pk}_l)$.	<u>Chal($j \in [l], f \in \mathcal{F}_{\text{aff}}$):</u> If $b = 0$, $\quad f \leftarrow \mathbf{0}$. $\mathbf{T}_f := \sum_{i\in[l]} \mathbf{T}_i\mathbf{P}_i \in \mathbb{F}_2^{n\times n}$. $\hat{\mathbf{S}} \leftarrow\$ (\widetilde{\mathcal{B}}^n_{\mu_1})^k$. $\hat{\mathbf{E}} \leftarrow\$ \mathcal{B}_\mu^{n\times k}$. $\hat{\mathbf{e}} \leftarrow\$ \mathcal{B}_\mu^k$. $\mathbf{C}_1 := \hat{\mathbf{S}}^\top\mathbf{A}_j + \hat{\mathbf{E}}^\top - \mathbf{G}\mathbf{T}_f\mathbf{P}_j^{-1} \in \mathbb{F}_2^{k\times n}$. $\mathbf{c}_2 := \hat{\mathbf{S}}^\top\mathbf{y}_j + \hat{\mathbf{e}} + \mathbf{G}\mathbf{t} \in \mathbb{F}_2^k$. Return $\mathsf{c} := (\mathbf{C}_1, \mathbf{c}_2)$.
Game G_{10}	<u>KeyGen:</u> $b \leftarrow\$ \{0,1\}$. // challenge bit For $i \in [l]$, $\quad \mathbf{A}_i \leftarrow\$ \mathcal{D}_\lambda^{n\times n}$. $\mathbf{P}_i \leftarrow\$ \mathcal{P}_n$. $\mathbf{y}_i \leftarrow\$ \mathbb{F}_2^n$. $\quad \mathsf{pk}_i := (\mathbf{A}_i, \mathbf{y}_i)$. Return $(\mathsf{pk}_1, \cdots, \mathsf{pk}_l)$.	<u>Chal($j \in [l], f \in \mathcal{F}_{\text{aff}}$):</u> If $b = 0$, $\quad f \leftarrow \mathbf{0}$. $\mathbf{T}_f := \sum_{i\in[l]} \mathbf{T}_i\mathbf{P}_i \in \mathbb{F}_2^{n\times n}$. $\hat{\mathbf{S}} \leftarrow\$ (\widetilde{\mathcal{B}}^n_{\mu_1})^k$. $\hat{\mathbf{E}} \leftarrow\$ \mathcal{B}_\mu^{n\times k}$. $\hat{\mathbf{e}} \leftarrow\$ \mathcal{B}_\mu^k$. $\mathbf{C}_1 := \hat{\mathbf{S}}^\top\mathbf{A}_j + \hat{\mathbf{E}}^\top - \mathbf{G}\mathbf{T}_f\mathbf{P}_j^{-1} \in \mathbb{F}_2^{k\times n}$. $\mathbf{c}_2 := \hat{\mathbf{S}}^\top\mathbf{y}_j + \hat{\mathbf{e}} + \mathbf{G}\mathbf{t} \in \mathbb{F}_2^k$. Return $\mathsf{c} := (\mathbf{C}_1, \mathbf{c}_2)$.
Game G_{11}	<u>KeyGen:</u> $b \leftarrow\$ \{0,1\}$. // challenge bit For $i \in [l]$, $\quad \mathbf{A}_i \leftarrow\$ \mathcal{D}_\lambda^{n\times n}$. $\mathbf{P}_i \leftarrow\$ \mathcal{P}_n$. $\mathbf{y}_i \leftarrow\$ \mathbb{F}_2^n$. $\quad \mathsf{pk}_i := (\mathbf{A}_i, \mathbf{y}_i)$. Return $(\mathsf{pk}_1, \cdots, \mathsf{pk}_l)$.	<u>Chal($j \in [l], f \in \mathcal{F}_{\text{aff}}$):</u> If $b = 0$, $\quad f \leftarrow \mathbf{0}$. $\mathbf{T}_f := \sum_{i\in[l]} \mathbf{T}_i\mathbf{P}_i \in \mathbb{F}_2^{n\times n}$. $\mathbf{U} \leftarrow\$ \mathbb{F}_2^{k\times n}$. $\mathbf{u} \leftarrow\$ \mathbb{F}_2^k$. $\mathbf{C}_1 := \mathbf{U} - \mathbf{G}\mathbf{T}_f\mathbf{P}_j^{-1} \in \mathbb{F}_2^{k\times n}$. $\mathbf{c}_2 := \mathbf{u} + \mathbf{G}\mathbf{t} \in \mathbb{F}_2^k$. Return $\mathsf{c} := (\mathbf{C}_1, \mathbf{c}_2)$.
Game G_{12}	<u>KeyGen:</u> $b \leftarrow\$ \{0,1\}$. // challenge bit For $i \in [l]$, $\quad \mathbf{A}_i \leftarrow\$ \mathcal{D}_\lambda^{n\times n}$. $\mathbf{y}_i \leftarrow\$ \mathbb{F}_2^n$. $\quad \mathsf{pk}_i := (\mathbf{A}_i, \mathbf{y}_i)$. Return $(\mathsf{pk}_1, \cdots, \mathsf{pk}_l)$.	<u>Chal($j \in [l], f \in \mathcal{F}_{\text{aff}}$):</u> $\mathbf{U} \leftarrow\$ \mathbb{F}_2^{k\times n}$. $\mathbf{u} \leftarrow\$ \mathbb{F}_2^k$. $\mathbf{C}_1 := \mathbf{U} \in \mathbb{F}_2^{k\times n}$. $\mathbf{c}_2 := \mathbf{u} \in \mathbb{F}_2^k$. Return $\mathsf{c} := (\mathbf{C}_1, \mathbf{c}_2)$.

Fig. 3. Games G_7–G_{12} for l-KDM[$\mathcal{F}_{\text{aff}}$]-CPA security of PKE (see also Fig. 2).

A.2 Proof of Claim 5

Firstly, we introduce a sequence of intermediate games $\{\mathsf{G}_{5,\kappa}\}_{\kappa\in[Q+1]}$ between G_5 and G_6.

– **Game $\mathsf{G}_{5,\kappa}$, $\kappa \in [Q+1]$.** This game is a hybrid of game G_5 and game G_6: for the first $\kappa-1$ times of CHAL queries, $\mathscr{C}$ computes $\mathbf{C}_1$ as in game G_6; for the remaining CHAL queries, $\mathscr{C}$ computes $\mathbf{C}_1$ as in game G_5.

Clearly, game $\mathsf{G}_{5,1}$ is identical to G_5 and game $\mathsf{G}_{5,Q+1}$ is identical to G_6. It suffices to show that $\big|\Pr_{5,\kappa}[\mathsf{Win}] - \Pr_{5,\kappa+1}[\mathsf{Win}]\big| \le \mathsf{negl}(n)$ for any $\kappa \in [Q]$.

The only difference between game $\mathsf{G}_{5,\kappa}$ and game $\mathsf{G}_{5,\kappa+1}$ is the distribution of $\mathbf{C}_1$ in the κ-th CHAL$(j \in [l], f \in \mathcal{F}_{\mathrm{aff}})$ query: in game $\mathsf{G}_{5,\kappa}$, $\mathbf{C}_1$ is computed according to game G_5, i.e., $\mathbf{C}_1 = \mathbf{U} - \mathbf{G}\mathbf{T}_f\mathbf{P}_j^{-1}$; in game $\mathsf{G}_{5,\kappa+1}$, it is computed according to game G_6, i.e., $\mathbf{C}_1 = \hat{\mathbf{S}}^\top\mathbf{A}_j + \hat{\mathbf{E}}^\top - \mathbf{G}\mathbf{T}_f\mathbf{P}_j^{-1}$.

We construct a PPT distinguisher $\mathscr{D}$ to solve the multi-fold LPN problem described in Lemma 5. Given a challenge $(\mathbf{A}, \mathbf{C}, (\mathbf{e}, \mathbf{s}, \mathbf{P}), (\hat{\mathbf{S}}^\top\mathbf{e}, \hat{\mathbf{E}}^\top\mathbf{P}\mathbf{s}))$, $\mathscr{D}$ wants to distinguish $\mathbf{C} = \hat{\mathbf{S}}^\top\mathbf{A} + \hat{\mathbf{E}}^\top$ from $\mathbf{C} = \mathbf{U}$, where $\mathbf{A} \leftarrow_\$ \mathcal{D}_\lambda^{n\times n}$, $\hat{\mathbf{S}} \leftarrow_\$ (\widetilde{\mathcal{B}}_{\mu_1}^n)^k$, $\hat{\mathbf{E}} \leftarrow_\$ \mathcal{B}_\mu^{n\times k}$, $\mathbf{e} \leftarrow_\$ \mathcal{B}_\mu^n$, $\mathbf{s} \leftarrow_\$ \chi_{\mu_1 n}^n$, $\mathbf{P} \leftarrow_\$ \mathcal{P}_n$ and $\mathbf{U} \leftarrow_\$ \mathbb{F}_2^{k\times n}$. $\mathscr{D}$ is constructed by simulating game $\mathsf{G}_{5,\kappa}$ or game $\mathsf{G}_{5,\kappa+1}$ for $\mathscr{A}$ as follows, where we highlight the challenge received by $\mathscr{D}$.

<u>KEYGEN.</u> $\mathscr{D}$ picks $b \leftarrow_\$ \{0,1\}$ uniformly, and proceeds as follows.

(a) Set the master secret $\mathbf{s}^* := \mathbf{s}$.

(b) Pick $j^* \leftarrow_\$ [l]$. For each user $i \in [l]$,
  - if $i \ne j^*$, choose $\mathbf{A}_i \leftarrow_\$ \mathcal{D}_\lambda^{n\times n}$, $\mathbf{P}_i \leftarrow_\$ \mathcal{P}_n$, $\mathbf{e}_i \leftarrow_\$ \mathcal{B}_\mu^n$;
  - if $i = j^*$, set $\mathbf{A}_{j^*} := \mathbf{A}$, $\mathbf{P}_{j^*} := \mathbf{P}$, $\mathbf{e}_{j^*} := \mathbf{e}$,

  and compute $\mathbf{s}_i := \mathbf{P}_i\mathbf{s}^* \in \mathbb{F}_2^n$ and $\mathbf{y}_i := \mathbf{A}_i\mathbf{P}_i\mathbf{s}^* + \mathbf{e}_i \in \mathbb{F}_2^n$.

Finally, $\mathscr{D}$ sends the public keys $\mathsf{pk}_i := (\mathbf{A}_i, \mathbf{y}_i)$, $i \in [l]$, to $\mathscr{A}$.

<u>CHAL$(j \in [l], f \in \mathcal{F}_{\mathit{aff}})$.</u> $\mathscr{D}$ proceeds as follows.

(a) Set $f \leftarrow \mathbf{0}$ if $b = 0$. Then compute $\mathbf{T}_f := \sum_{i\in[l]} \mathbf{T}_i\mathbf{P}_i \in \mathbb{F}_2^{n\times n}$.

(b)
  - For the first $\kappa - 1$ queries, $\mathscr{D}$ computes $\mathbf{C}_1$ and $\mathbf{c}_2$ according to game G_6.
  - For the κ-th query, $\mathscr{D}$ aborts immediately if $j \ne j^*$; otherwise $\mathscr{D}$ chooses $\hat{\mathbf{e}} \leftarrow_\$ \mathcal{B}_\mu^k$, and computes $\mathbf{C}_1 := \mathbf{C} - \mathbf{G}\mathbf{T}_f\mathbf{P}_{j^*}^{-1}$ and $\mathbf{c}_2 := (\mathbf{C}_1\mathbf{P}_{j^*} + \mathbf{G}\mathbf{T}_f)\cdot\mathbf{s}^* - \hat{\mathbf{E}}^\top\mathbf{P}\mathbf{s} + \hat{\mathbf{S}}^\top\mathbf{e} + \hat{\mathbf{e}} + \mathbf{G}\mathbf{t}$.
  - For the remaining queries, $\mathscr{D}$ computes $\mathbf{C}_1$ and $\mathbf{c}_2$ according to game G_5.

Finally, $\mathscr{D}$ returns the challenge ciphertext $\mathsf{c} := (\mathbf{C}_1, \mathbf{c}_2)$ to $\mathscr{A}$.

<u>GUESS.</u> $\mathscr{A}$ outputs a guessing bit $b' \in \{0,1\}$.

$\mathscr{D}$ finally outputs 1 if and only if $j = j^*$ holds in the κ-th CHAL query (i.e., $\mathscr{D}$ does not abort) and $b' = b$.

We analyze the distinguishing advantage of $\mathscr{D}$.

- In KEYGEN, $\mathbf{s}^*$, $\mathbf{A}_{j^*}$, $\mathbf{P}_{j^*}$ and $\mathbf{e}_{j^*}$ have the same distributions as in both game $\mathsf{G}_{5,\kappa}$ and game $\mathsf{G}_{5,\kappa+1}$. Besides, j^* is completely hidden from $\mathscr{A}$'s view.
- In the κ-th query of CHAL$(j \in [l], f \in \mathcal{F}_{\mathrm{aff}})$, $j = j^*$ holds with probability at least $1/l$.

– If $\mathbf{C} = \hat{\mathbf{S}}^\top \mathbf{A} + \hat{\mathbf{E}}^\top$, then $\mathbf{C}_1 = \hat{\mathbf{S}}^\top \mathbf{A} + \hat{\mathbf{E}}^\top - \mathbf{G}\mathbf{T}_f \mathbf{P}_{j^*}^{-1} = \hat{\mathbf{S}}^\top \mathbf{A}_{j^*} + \hat{\mathbf{E}}^\top - \mathbf{G}\mathbf{T}_f \mathbf{P}_{j^*}^{-1}$ and $\mathbf{c}_2 = (\mathbf{C}_1 \mathbf{P}_{j^*} + \mathbf{G}\mathbf{T}_f) \cdot \mathbf{s}^* - \hat{\mathbf{E}}^\top \mathbf{P}_{j^*} \mathbf{s}^* + \hat{\mathbf{S}}^\top \mathbf{e}_{j^*} + \hat{\mathbf{e}} + \mathbf{G}\mathbf{t}$. Thus, $\mathscr{D}$ computes $(\mathbf{C}_1, \mathbf{c}_2)$ for the κ-th CHAL query exactly like game G_6.
– If $\mathbf{C} = \mathbf{U}$, then $\mathbf{C}_1 = \mathbf{U} - \mathbf{G}\mathbf{T}_f \mathbf{P}_{j^*}^{-1}$ and $\mathbf{c}_2 = (\mathbf{C}_1 \mathbf{P}_{j^*} + \mathbf{G}\mathbf{T}_f) \cdot \mathbf{s}^* - \hat{\mathbf{E}}^\top \mathbf{P}_{j^*} \mathbf{s}^* + \hat{\mathbf{S}}^\top \mathbf{e}_{j^*} + \hat{\mathbf{e}} + \mathbf{G}\mathbf{t}$. Thus, $\mathscr{D}$ computes $(\mathbf{C}_1, \mathbf{c}_2)$ for the κ-th CHAL query exactly like game G_5.

Therefore, if $\mathscr{D}$ does not abort (which occurs with probability at least $1/l$), $\mathscr{D}$ simulates game $\mathsf{G}_{5,\kappa+1}$ perfectly for $\mathscr{A}$ in the case of $\mathbf{C} = \hat{\mathbf{S}}^\top \mathbf{A} + \hat{\mathbf{E}}^\top$ and simulates game $\mathsf{G}_{5,\kappa}$ perfectly for $\mathscr{A}$ in the case of $\mathbf{C} = \mathbf{U}$. Consequently, $\mathscr{D}$'s distinguishing advantage is at least $\frac{1}{l} \cdot \big| \Pr_{5,\kappa}[\mathsf{Win}] - \Pr_{5,\kappa+1}[\mathsf{Win}] \big|$, which is $\mathsf{negl}(n)$ by Lemma 5.

In conclusion, $\big| \Pr_5[\mathsf{Win}] - \Pr_6[\mathsf{Win}] \big| = \big| \Pr_{5,1}[\mathsf{Win}] - \Pr_{5,Q+1}[\mathsf{Win}] \big| \leq \sum_{\kappa \in [Q]} \big| \Pr_{5,\kappa}[\mathsf{Win}] - \Pr_{5,\kappa+1}[\mathsf{Win}] \big| \leq Ql \cdot \mathsf{negl}(n)$, which is also negligible in n. This completes the proof of Claim 5.

Contents