Simpler CCA Secure PKE from LPN Problem Without Double-Trapdoor

Haitao Cheng[1,2], Xiangxue Li[1,3(✉)], Haifeng Qian[1], and Di Yan[4]

[1] Department of Computer Science and Technology, East China Normal University, Shanghai, China
{xxli,hfqian}@cs.ecnu.edu.cn
[2] National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications, Xi'an, China
[3] Westone Cryptologic Research Center, Beijing, China
[4] Department of Computer Science and Engineering, Shanghai Jiaotong University, Shanghai, China

Abstract. The first CCA secure public key encryption (PKE) on the learning parity with noise (LPN) assumption was invented by Döttling et al. (ASIACRYPT 2012). At PKC 2014, Kiltz et al. gave a simpler and more efficient construction, where a double-trapdoor technique was introduced to handle the decryption queries in game simulation. Different from the technique, we build in the standard model the CCA secure PKE on a variant of Extended Knapsack LPN problem (which is provably equivalent to the standard LPN problem). We abstract out an ephemeral key from the LPN assumption, which can then be used to encrypt the underlying plaintext when equipped with several typical classes of cryptographic primitives. Thanks to these techniques, the decryption queries can be correctly answered (yet without relying on a double-trapdoor mechanism) during security reduction from LPN. The resulting simple proposal appears more modular and efficient.

Keywords: Post quantum cryptography · Low-noise LPN
Extended Knapsack LPN

1 Introduction

In cryptography and learning theory, the Learning Parity with Noise (LPN) problem has become a well-known problem. The two versions of LPN have been pointed out to be polynomially equivalent [10]. The decisional one with parameter $0 < \mu < 1/2$ (noise rate), $m = \mathsf{poly}(n)$, $n \in \mathbb{N}$ posulates that $(\mathbf{A}, \langle \mathbf{A}, \mathbf{s} \rangle + \mathbf{e})$ is pseudorandom given $\mathbf{A}$ (i.e., computationally indistinguishable from uniform randomness), where $\mathbf{A} \in \{0,1\}^{m \times n}$, $\mathbf{s} \in \{0,1\}^n$ are chosen uniformly at random, $\mathbf{e} \in \{0,1\}^m$ is distributed to $\mathcal{B}_\mu^m$, (i.e., concatenation of m independent copies of the Bernoulli distribution $\mathcal{B}_\mu$ such that $\Pr[\mathcal{B}_\mu = 1] = \mu$), $\langle \cdot, \cdot \rangle$ denotes the inner product of two vectors and '+' denotes the XOR operation. The computational

D. Naccache et al. (Eds.): ICICS 2018, LNCS 11149, pp. 756–766, 2018.
https://doi.org/10.1007/978-3-030-01950-1_46

version assumes that it is computationally infeasible to find out the random secret binary vector $\mathbf{s} \in \{0,1\}^n$ from those noisy linear samples.

LPN Hardness. The computational LPN problem is deemed as a well-known NP-complete problem "decoding random linear codes" [2], which makes LPN be a promising candidate for post-quantum cryptography. Furthermore, the simplicity of LPN makes it more suitable for weak-power devices (e.g., RFID tags) than other post-quantum candidates such as LWE [17]. The best known algorithms for solving constant noise (noise parameter $0 < \mu < 1/2$) LPN problem require $2^{O(n/\log n)}$ time and samples [4,12]. When given only polynomially many $\mathsf{poly}(n)$ samples, the time complexity goes up to $2^{O(n/\log\log n)}$ [13], and even $2^{O(n)}$ when given only linearly many $O(n)$ samples [14,19]. Under low-noise rate i.e., the 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 [3,19].

1.1 Related Work

PKE with CPA security. Retrospectively, Alekhnovich [1] constructed the first CPA-secure public-key encryption scheme from low-noise LPN (i.e., noise rate $\mu = 1/\sqrt{n}$). Inspired by the schemes of Regev [17] and Gentry et al. [9], Döttling et al. proposed an alternative one [8]. The work of Yu and Zhang [20] in 2016 made a breakthrough in solving the open problem of constructing public-key primitives based on constant-noise LPN problem. In their IND-CPA scheme, they used a variant assumption called LPN on Squared-Log Entropy and gave a tight requirement of secret key's distribution.

PKE with CCA security. IND-CCA security [16] is one of the strongest known notions of security for public-key encryption schemes. Döttling et al. [8] constructed the first CCA-secure PKE scheme from low-noise LPN by using the correlated products approach of [18]. But the complexity of that scheme was hundreds of times worse than Alekhnovich's scheme. Kiltz et al. [11] gave a more efficient CCA-secure construction by means of the techniques from LWE-based encryption in [15] with some technical changes. Specifically, they used a double-trapdoor mechanism, together with a trapdoor switching lemma so that there is always an available trapdoor to answer the decryption queries in game simulation. In [20], Yu and Zhang constructed the first constant-noise LPN problem based CCA-secure scheme which uses a tag-based encryption technique.

1.2 Our Contributions

In this work, we propose a simple and efficient PKE scheme which is IND-CCA secure from low-noise LPN . We build a neat construction with noise rate $\mu \approx O(\sqrt{1/n})$.

With an IND-CPA secure private-key scheme and a collision resistant hash function H we plug the $\mathsf{H}(\mathbf{c}_1, \mathbf{c}_2, \mathbf{s}, \mathbf{H_t})$ into $\mathsf{Enc}'_{\mathbf{k}}(\mathbf{m})$ where $\mathbf{k} = \mathsf{H}(\mathbf{c}_1, \mathbf{c}_2, \mathbf{s}, \mathbf{H_t})$ becomes a secret key of the Enc' algorithm of an IND-CPA-secure private-key

scheme Π'. Intuitively, based on the indistinguishability of LPN samples, it holds that the scheme is IND-sTag-CCA secure (see Definition 4) and can be efficiently transformed into a CCA-secure encryption scheme [5,11,20].

2 Preliminaries

2.1 Notations and Definitions

We use capital letters (e.g., X, Y) for random variables and distributions, standard letters (e.g., x, y) for values. Vectors are used in the column form and denoted by bold lower-case letters (e.g., $\mathbf{a}$). We treat matrices as the sets of its column vectors and denote them by bold capital letters (e.g., $\mathbf{A}$). For a binary string x, $|x|$ refers to the Hamming weight of x. We use $\mathcal{B}_\mu$ to denote the Bernoulli distribution with parameter μ, i.e., $\Pr[\mathcal{B}_\mu = 1] = \mu$, $\Pr[\mathcal{B}_\mu = 0] = 1 - \mu$, while $\mathcal{B}_\mu^n$ denotes the concatenation of n independent copies of $\mathcal{B}_\mu$. For n, $\ell \in \mathbb{N}$, U_n (resp., $U_{\ell \times n}$) denotes the uniform distribution over $\{0,1\}^n$ (resp., $\{0,1\}^{\ell \times n}$) and independent of any other random variables in consideration. $X \sim D$ denotes that random variable X follows distribution D. We use $s \leftarrow S$ to denote sampling an element s according to distribution S. For random variables X and Y, the statistical distance between them is defined by $\Delta(X,Y) = \frac{1}{2} \cdot \sum_x |\Pr[X = x] - \Pr[Y = x]|$. If for probability ensembles $X = \{X_n\}_{n \in \mathbb{N}}$ and $Y = \{Y_n\}_{n \in \mathbb{N}}$, $\Delta(X_n, Y_n) \leq \mathsf{negl}(n)$ holds, then X and Y are called statistically indistinguishable, denoted by $X \overset{s}{\sim} Y$. If for any PPT distinguisher $\mathcal{D}$, $|\Pr[\mathcal{D}(X_n) = 1] - \Pr[\mathcal{D}(Y_n) = 1] \leq \mathsf{negl}(n)|$ holds then X and Y are called computationally indistinguishable, denoted by $X \overset{c}{\sim} Y$.

Collision Resistant Hash Function. A hash function family $\mathcal{H} = \{\mathsf{H} : \mathcal{X} \to \mathcal{Y}\}$ is collision resistant if for any PPT adversary $\mathcal{A}$, it satisfies that $\mathrm{Adv}^{cr}_{\mathcal{H},\mathcal{A}}(n) = \Pr[\mathsf{H} \overset{\$}{\leftarrow} \mathcal{H}, (x, x') \overset{\$}{\leftarrow} \mathcal{A}(\mathsf{H}) : \mathsf{H}(x) = \mathsf{H}(x') \wedge x \neq x'] \leq \mathsf{negl}(n)$.

2.2 Learning Parity with Noise

Definition 1 (Learning Parity with Noise). *The* ***decisional*** $\mathsf{LPN}_{n,m,\mu}$ *problem is hard if for every* $m = \mathsf{poly}(n)$ *we have* $(\mathbf{A},\ \mathbf{A} \cdot \mathbf{s} + \mathbf{e}) \overset{c}{\sim} (\mathbf{A}, \mathbf{b})$ *where* $\mathbf{A} \sim U_{m \times n}$, $\mathbf{s} \sim U_n, \mathbf{e} \sim \mathcal{B}_\mu^m$ *and* $\mathbf{b} \sim U_m$ *while the secret length is* n *and the noise rate is* $0 < \mu < 1/2$. *The* ***computational*** $\mathsf{LPN}_{n,m,\mu}$ *problem is hard if for every* $m = \mathsf{poly}(n)$ *and every PPT algorithm* $\mathcal{D}$ *we have* $\Pr[\ \mathcal{D}(\mathbf{A},\ \mathbf{A} \cdot \mathbf{s} + \mathbf{e}) = \mathbf{s}\ ] = \mathsf{negl}(n)$ *where* $\mathbf{A} \sim U_{m \times n}$, $\mathbf{s} \sim U_n$ *and* $\mathbf{e} \sim \mathcal{B}_\mu^m$.

Definition 2 (Knapsack LPN-KLPN). *The knapsack LPN problem is hard if for* $m > n$ *samples we have* $(\mathbf{A},\ \mathbf{A}^\intercal \mathbf{t}) \overset{c}{\sim} (\mathbf{A},\ \mathbf{b})$ *where* $\mathbf{A} \sim U_{m \times n}$, $\mathbf{t} \sim \mathcal{B}_\mu^m$, $\mathbf{b} \sim U_n$.

With a standard hybrid argument technique, we have results on the ℓ-fold LPN and ℓ-fold KLPN that $(\mathbf{A}, \mathbf{AS} + \mathbf{E}) \overset{c}{\sim} (\mathbf{A}, \mathbf{B_1})$ where $\mathbf{A} \sim U_{m \times n}, \mathbf{S} \sim U_{n \times \ell}, \mathbf{E} \sim \mathcal{B}_\mu^{m \times \ell}$ and $\mathbf{B}_1 \sim U_{m \times \ell}$; $(\mathbf{A}, \mathbf{T}^\intercal \mathbf{A}) \overset{c}{\sim} (\mathbf{A}, \mathbf{B_2})$ where $\mathbf{A} \sim U_{m \times n}, \mathbf{T} \sim \mathcal{B}_\mu^{m \times \ell}$ and $\mathbf{B}_2 \sim U_{\ell \times n}$.

Definition 3 (Extended Knapsack LPN-EKLPN). *The Extended Knapsack LPN problem is hard if for $m > n$ samples we have* $(\mathbf{A}, \mathbf{A}^\intercal\mathbf{t}, \mathbf{e}, \mathbf{t}^\intercal\mathbf{e}) \overset{c}{\sim} (\mathbf{A}, \mathbf{b}, \mathbf{e}, \mathbf{t}^\intercal\mathbf{e})$ *where* $\mathbf{A} \sim U_{m\times n}, \mathbf{b} \sim U_n$, $\mathbf{t}, \mathbf{e} \sim \mathcal{B}_\mu^m$.

Lemma 1. *Assume that the Extended Knapsack LPN problem is hard then we have* $(\mathbf{A}, \mathbf{A}^\intercal\mathbf{t}, \mathbf{e}, \mathbf{t}^\intercal\mathbf{e}) \overset{c}{\sim} (\mathbf{A}, \mathbf{A}^\intercal\mathbf{t}', \mathbf{e}, \mathbf{t}^\intercal\mathbf{e})$.

Proof. From Definition 3 we have $(\mathbf{A}, \mathbf{A}^\intercal\mathbf{t}, \mathbf{e}, \mathbf{t}^\intercal\mathbf{e}) \overset{c}{\sim} (\mathbf{A}, \mathbf{b}, \mathbf{e}, \mathbf{t}^\intercal\mathbf{e})$. From Definition 2 we have $(\mathbf{A},\ \mathbf{A}^\intercal\mathbf{t}') \overset{c}{\sim} (\mathbf{A},\ \mathbf{b})$ where $\mathbf{A} \sim U_{m\times n}$, $\mathbf{t}, \mathbf{t}', \mathbf{e} \sim \mathcal{B}_\mu^m$. By combining these two equations, we immediately obtain $(\mathbf{A}, \mathbf{A}^\intercal\mathbf{t}, \mathbf{e}, \mathbf{t}^\intercal\mathbf{e}) \overset{c}{\sim} (\mathbf{A}, \mathbf{A}^\intercal\mathbf{t}', \mathbf{e}, \mathbf{t}^\intercal\mathbf{e})$.

The Extended Knapsack LPN to standard LPN problem reduction can be referenced to [7].

3 CCA Secure PKE from Low-Noise LPN

In this section, we construct a CCA-secure PKE from low-noise LPN problem. Technically, we construct a tag-based PKE against selective tag and chosen ciphertext attacks from LPN, which can be transformed into a standard CCA-secure PKE by using known techniques [5,11,20].

3.1 Tag-Based Encryption

A tag-based encryption (TBE) scheme with tag-space $\mathcal{T}$ and message-space $\mathcal{M}$ consists of three PPT algorithms $\mathcal{TBE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$. The randomized key generation algorithm KeyGen takes the security parameter n as input, outputs a public key pk and a secret key sk, denoted as $(pk, sk) \leftarrow \mathsf{KeyGen}(1^n)$. The randomized encryption algorithm Enc takes pk, a tag $\mathbf{t} \in \mathcal{T}$, and a plaintext $\mathbf{m} \in \mathcal{M}$ as input, outputs a ciphertext C, denoted as $C \leftarrow \mathsf{Emc}(pk, \mathbf{t}, \mathbf{m})$. The deterministic algorithm Dec takes sk and C as inputs, outputs a plaintext $\mathbf{m}$, or a special symbol $\perp$, which is denoted as $\mathbf{m} \leftarrow \mathsf{Dec}(sk, \mathbf{t}, C)$. For correctness, we require that for all $(pk, sk) \leftarrow \mathsf{KeyGen}(1^n)$, any tag $\mathbf{t}$, any plaintext $\mathbf{m}$ and any $C \leftarrow \mathsf{Enc}(pk, \mathbf{t}, \mathbf{m})$, the equation $\mathsf{Dec}(sk, \mathbf{t}, C) = \mathbf{m}$ holds with overwhelming probability.

We consider the following game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

Init. The adversary $\mathcal{A}$ takes the security parameter n as input, and outputs a target $\mathbf{t}^*$ to the challenger $\mathcal{C}$.

KeyGen. The challenger $\mathcal{C}$ computes $(pk, sk) \leftarrow \mathsf{KeyGen}(1^n)$, gives the public key pk to the adversary $\mathcal{A}$, and keeps the secret key sk.

Phase 1. The adversary $\mathcal{A}$ can make decryption queries polynomial times for any pair $(\mathbf{t}, C)$, with a restriction that $\mathbf{t} \neq \mathbf{t}^*$, and the challenger $\mathcal{C}$ returns $\mathbf{m} \leftarrow \mathsf{Dec}(sk, \mathbf{t}, C)$ to $\mathcal{A}$ accordingly.

Challenge. The adversary $\mathcal{A}$ outputs two equal length plaintexts $\mathbf{m}_0, \mathbf{m}_1 \in \mathcal{M}$. The challenger $\mathcal{C}$ randomly chooses a bit $b^* \overset{\$}{\leftarrow} \{0, 1\}$, and returns the challenge ciphertext $C^* \leftarrow \mathsf{Enc}(pk, \mathbf{t}^*, \mathbf{m}_{b^*})$ to the adversary $\mathcal{A}$.

Phase 2. The adversary can make more decryption queries as in Phase 1.

Guess. Finally, $\mathcal{A}$ outputs a guess $b \in \{0,1\}$. If $b = b^*$, the challenger $\mathcal{C}$ outputs 1, else outputs 0.

Advantage. $\mathcal{A}$'s advantage is defined as $\mathrm{Adv}^{\mathrm{ind-stag-cca}}_{\mathcal{TBE},\mathcal{A}}(1^n) \stackrel{\mathrm{def}}{=} |\Pr[b = b^*] - \frac{1}{2}|$.

Definition 4 (IND-sTag-CCA.) *We say that a TBE scheme $\mathcal{TBE}$ is IND-sTag-CCA secure if for any PPT adversary $\mathcal{A}$, its advantage is negilible in n.*

3.2 The Construction

Our TBE scheme $\mathcal{TBE}$ is constructed by using the following parameters and building blocks. Let k be the security parameter, $n = \Theta(k^2)$, $m \in \mathbb{Z}$ such that $m \geq 2n$. A constant $0 < c < \frac{1}{6}$ (recall that we set $6c < \alpha < 1$) defining: The Bernoulli parameter $\mu = \sqrt{c/m}$ and the bounding parameter $\beta = 2\sqrt{cm}$ to check consistency during decryption. A generator matrix $\mathbf{G} \in \mathbb{Z}_2^{m \times n}$ of a binary linear error-correcting code $\mathcal{C} = \mathcal{C}(\mathbf{G})$ and has efficient decode algorithm $\mathsf{Decode}_{\mathbf{G}}$ correcting up to αm errors (we refer to [11] for details about error-correcting code). Let the tag-space $\mathcal{T} = \mathbb{F}_{2^n}$. We use a matrix representation $\mathbf{H_t} \in \{0,1\}^{n \times n}$ for finite field elements $\mathbf{t} \in \mathbb{F}_{2^n}$ [5,6,11] such that $\mathbf{H_0} = \mathbf{0}$, $\mathbf{H_t}$ is invertible for any $\mathbf{t} \neq \mathbf{0}$, and $\mathbf{H_{t_1}} + \mathbf{H_{t_2}} = \mathbf{H_{t_1 + t_2}}$. A family of collision resistant hash functions $\mathcal{H} := \{\mathsf{H} : \mathbb{Z}_2^m \times \mathbb{Z}_2^m \times \mathbb{Z}_2^n \times \mathbb{Z}_2^{n \times n} \to \mathbb{Z}_2^\ell\}$. Let $\Pi' = (\mathsf{Enc}', \mathsf{Dec}')$ be a private-key encryption scheme for messages $\mathbf{m} \in \{0,1\}^{\ell'}$ ($\ell' \ll n$, say $\ell' = 128$ typically). We present the construction of $\mathcal{TBE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\{0,1\}^{\ell'}$ in Fig. 1.

$(pk, sk) \stackrel{\$}{\leftarrow} \mathsf{KeyGen}(1^k)$:	$c \stackrel{\$}{\leftarrow} \mathsf{Enc}(pk, \mathbf{t}, \mathbf{m})$: //$\mathbf{m} \in \{0,1\}^{\ell'}, \mathbf{t} \in \mathbb{F}_{2^n}$	$\mathbf{m} \leftarrow \mathsf{Dec}(sk, \mathbf{t}, c)$:
$\mathbf{A} \stackrel{\$}{\leftarrow} U_{m \times n}$.	Parse $pk = (\mathbf{A}, \mathbf{B})$.	Parse $sk = \mathbf{T}$.
$\mathbf{T} \stackrel{\$}{\leftarrow} \mathcal{B}_\mu^{m \times m}$.	$\mathbf{s} \stackrel{\$}{\leftarrow} U_n$.	Parse $c := (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$.
$\mathbf{B} := \mathbf{TA}$.	$\mathbf{e}_1 \stackrel{\$}{\leftarrow} \mathcal{B}_\mu^m$.	$\mathbf{y} := \mathbf{c}_2 - \mathbf{T}\mathbf{c}_1$.
	$\mathbf{T}' \stackrel{\$}{\leftarrow} \mathcal{B}_\mu^{m \times m}$.	$\mathbf{H_t s} = \mathbf{b} := \mathsf{Decode}_{\mathbf{G}}(\mathbf{y})$.
Return $pk := (\mathbf{A}, \mathbf{B})$,	$\mathbf{c}_1 := \mathbf{As} + \mathbf{e}_1$.	Compute $\mathbf{s} = \mathbf{H_t^{-1}b}$, and check whether
$sk := \mathbf{T}$.	$\mathbf{c}_2 := (\mathbf{GH_t} + \mathbf{B})\mathbf{s} + \mathbf{T}'\mathbf{e}_1$.	$\lvert \underbrace{\mathbf{c}_1 - \mathbf{As}}_{\mathbf{e}_1} \rvert \leq \beta \wedge \lvert \underbrace{\mathbf{c}_2 - (\mathbf{GH_t} + \mathbf{B})\mathbf{s}}_{\mathbf{T}'\mathbf{e}_1} \rvert \leq \frac{\alpha m}{3}$.
	$\mathbf{k} = \mathsf{H}(\mathbf{c}_1, \mathbf{c}_2, \mathbf{s}, \mathbf{H_t})$.	If yes, compute $\mathbf{k} = \mathsf{H}(\mathbf{c}_1, \mathbf{c}_2, \mathbf{s}, \mathbf{H_t})$,
	$\mathbf{c}_3 := \mathsf{Enc}'_{\mathbf{k}}(\mathbf{m})$.	$\mathbf{m} = \mathsf{Dec}'_{\mathbf{k}}(\mathbf{c}_3)$, otherwise let $\mathbf{m} = \perp$.
	Return $c := (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$.	Return $\mathbf{m}$.

Fig. 1. IND-sTag-CCA secure $\mathcal{TBE}$ from low-noise LPN

3.3 Correctness

Lemma 2 (Chernoff Bound [11,20]). *For any* $0 < \mu < 1$ *and any* $\delta > 0$, *we have* $\Pr[|\mathcal{B}_\mu^m| > (1+\delta)\mu m] < e^{\frac{-\min(\delta,\delta^2)}{3}\mu m}$, *in particular, for* $\delta = 1$ $\Pr[|\mathcal{B}_\mu^m| > 2\mu m] < e^{-\mu m/3}$.

Obviously, for the chosen $\mathbf{e}_1 \xleftarrow{\$} \mathcal{B}_\mu^m$, the Chernoff Bound yields: $\Pr[|\mathbf{e}_1| > \underbrace{\beta}_{=2\mu m}] < e^{-\mu m/3} = 2^{-\Theta(\sqrt{m})}$.

Theorem 1 (Correctness). *Let parameters be chosen as in our construction then with overwhelming probability over the choice of the public and secret keys and for all* $\mathbf{m} \in \{0,1\}^{\ell'}$, $\mathsf{Dec}(sk, c)$ *outputs* $\mathbf{m}$ *correctly over* $c \leftarrow \mathsf{Enc}(pk, \mathbf{m})$.

Proof. The scheme's correctness requires the following:

1. $|(\mathbf{T}' - \mathbf{T})\mathbf{e}_1| \leq \alpha m$ (to let $\mathsf{Decode}_{\mathbf{G}}$ reconstruct $\mathbf{s}$ from $\mathbf{y} = \mathbf{c}_2 - \mathbf{T}\mathbf{c}_1$).
2. $|\mathbf{c}_1 - \mathbf{A}\mathbf{s}| \leq \beta \wedge |\mathbf{c}_2 - (\mathbf{G}\mathbf{H}_\mathbf{t} + \mathbf{B})\mathbf{s}| \leq \frac{\alpha m}{3}$.

For the decryption algorithm we require that the Hamming weight of the inner-product of a matrix $\mathbf{T} \xleftarrow{\$} \mathcal{B}_\mu^{m \times m}$ and a vector $\mathbf{e}_1 \xleftarrow{\$} \mathcal{B}_\mu^m$ is upper bounded by $\frac{1}{3}\alpha m$ with overwhelming probability. We firstly analyze the inner-product of a vector $\mathbf{t} \xleftarrow{\$} \mathcal{B}_\mu^m$ and the vector $\mathbf{e}_1 \xleftarrow{\$} \mathcal{B}_\mu^m$ whose Hamming weight is at most β described as above. Since $|\mathbf{e}_1| \leq \beta$, a necessary condition for $\mathbf{t}^\intercal\mathbf{e}_1 = 1$ is that $\mathbf{t}[i] = 1$ for at least one of the i's where $\mathbf{e}_1[i] = 1$. By a simple XOR-Lemma, it holds that $\mu' = \Pr[\mathbf{t}^\intercal\mathbf{e}_1 = 1] \leq \beta\mu = 2c$.

By the Chernoff Bound (1) and with $\delta = \alpha/(3\mu') - 1$ (where $\mu' \leq 2c < \alpha/3$) $\Pr\left[|\mathbf{T}\mathbf{e}_1| > \frac{1}{3}\alpha m\right] = \Pr\left[|\mathbf{T}\mathbf{e}_1| > (1+\delta)\mu' m\right] < e^{\frac{-\min(\delta,\delta^2)}{3}\mu' m}$.

Since $\delta\mu' = \alpha/3 - \mu' \geq \alpha/3 - 2c > 0$ and $\delta = \alpha/(3\mu') - 1 \geq \alpha/(6c) - 1 > 0$ are lower bounded by constants and therefore $\Pr\left[|\mathbf{T}\mathbf{e}_1| > \frac{1}{3}\alpha m\right] < e^{\frac{-\min(\delta,\delta^2)}{3}\mu' m} = 2^{-\Theta(m)}$.

Finally, in the ciphertext of our construction we have $|\mathbf{c}_1 - \mathbf{A}\mathbf{s}| = |\mathbf{e}_1| \leq \beta \wedge |\mathbf{c}_2 - (\mathbf{G}\mathbf{H}_\mathbf{t} + \mathbf{B})\mathbf{s}| = |\mathbf{T}'\mathbf{e}_1| \leq \frac{1}{3}\alpha m$ holds with overwhelming probability $1 - 2^{-\Theta(\sqrt{m})}$. In the decrption operation, $\mathbf{y} = \mathbf{c}_2 - \mathbf{T} \cdot \mathbf{c}_1 = (\mathbf{G}\mathbf{H}_\mathbf{t} + \mathbf{B}) \cdot \mathbf{s} + \mathbf{T}'\mathbf{e}_1 - \mathbf{T}(\mathbf{A} \cdot \mathbf{s} + \mathbf{e}_1) = \mathbf{G}\mathbf{H}_\mathbf{t} \cdot \mathbf{s} + (\mathbf{T}' - \mathbf{T}) \cdot \mathbf{e}_1$ it is sufficient to bound the error item $|(\mathbf{T}' - \mathbf{T})\mathbf{e}_1|$. It holds that $|(\mathbf{T}' - \mathbf{T})\mathbf{e}_1| \leq |\mathbf{T}'\mathbf{e}_1| + |\mathbf{T}\mathbf{e}_1| \leq \frac{2}{3}\alpha m < \alpha m$. Therefore, the decoding-procedure $\mathsf{Decode}_{\mathbf{G}}$ will successfully recover $\mathbf{s}$.

In all, the message $\mathbf{m}$ can be decrypted with overwhelming probability. □

3.4 Security

Theorem 2. *Assume that the LPN problem is hard,* H *is a collision resistant hash function and* Π' *is an IND-CPA-secure private-key encryption scheme then our TBE scheme* $\mathcal{TBE}$ *in Fig. 1. is IND-sTag-CCA secure.*

Proof. Let $\mathcal{A}$ be any PPT adversary that can attack our scheme $\mathcal{TBE}$ with advantage ε. We show that ε must be negligible in n. We continue the proof by using a sequence of games, where the first game is the real game, while the last is a random game in which the challenge ciphertext contains one component from an IND-CPA secure private-key encryption. Thus if $\mathcal{A}$ can win in the last game he breaks the IND-CPA secure private-key encryption as well which violates the assumption. The security of $\mathcal{TBE}$ can be established by showing that $\mathcal{A}$'s advantage in any two consecutive games are negligibly close.

Game 1. This is the IND-sTag-CCA experiment. The challenger $\mathcal{C}$ honestly runs the adversary $\mathcal{A}$ with the security parameter k and obtains a target tag $\mathbf{t}^*$ from $\mathcal{A}$. Then, it simulates the IND-sTag-CCA security game for $\mathcal{A}$ as follows:

KeyGen. First uniformly choose a collision resistant hash function $\mathsf{H} \xleftarrow{\$} \mathcal{H}$ and matrices $\mathbf{A} \xleftarrow{\$} U_{m\times n}$, $\mathbf{T} \xleftarrow{\$} \mathcal{B}_{\mu}^{m\times m}$. Then, compute $\mathbf{B} = \mathbf{TA} \in \{0,1\}^{m\times n}$. Finally, $\mathcal{C}$ sends $pk = (\mathbf{A}, \mathbf{B})$ to the adversary $\mathcal{A}$, and keeps $sk = \mathbf{T}$ to itself.

Phase 1. While receiving a decryption query $c = (\mathbf{t}, (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3))$ from adversary $\mathcal{A}$, the challenger $\mathcal{C}$ directly returns $\perp$ if $\mathbf{t} = \mathbf{t}^*$. Otherwise it first computes $\mathbf{y} = \mathbf{c}_2 - \mathbf{T}\cdot\mathbf{c}_1 = (\mathbf{GH_t} + \mathbf{B})\cdot\mathbf{s} + \mathbf{T}'\mathbf{e}_1 - \mathbf{T}(\mathbf{A}\cdot\mathbf{s} + \mathbf{e_1}) = \mathbf{GH_t}\cdot\mathbf{s} + (\mathbf{T}' - \mathbf{T})\mathbf{e}_1$. Then the challenger reconstructs $\mathbf{b} = \mathbf{H_t s}$ from the error $(\mathbf{T}' - \mathbf{T})\mathbf{e}_1$ by using the error correction peoperty of $\mathbf{G}$ and computes $\mathbf{s} = \mathbf{H_t^{-1}b}$. Then the challenger $\mathcal{C}$ checks that whether it satisfies that $|\mathbf{c}_1 - \mathbf{As}| \leq \beta \wedge |\mathbf{c}_2 - (\mathbf{GH_t} + \mathbf{B})\mathbf{s}| \leq \frac{1}{3}\alpha m$. If yes it computes $\mathbf{k} = \mathsf{H}(\mathbf{c}_1, \mathbf{c}_2, \mathbf{s}, \mathbf{H_t}), \mathbf{m} = \mathsf{Dec}'_{\mathbf{k}}(\mathbf{c}_3)$ otherwise lets $\mathbf{m} = \perp$. Finally it returns $\mathbf{m}$ to $\mathcal{A}$.

Challenge. After receiving two equal length plaintexts $\mathbf{m}_0$, $\mathbf{m}_1 \in \{0,1\}^{\ell'}$ from the adversary $\mathcal{A}$, the challenger $\mathcal{C}$ first randomly chooses a bit $b^* \xleftarrow{\$} \{0,1\}$, and $\mathbf{s} \xleftarrow{\$} U_n, \mathbf{e}_1 \xleftarrow{\$} \mathcal{B}_{\mu}^{m}, \mathbf{T}' \xleftarrow{\$} \mathcal{B}_{\mu}^{m\times m}$. Then, it calculates $\mathbf{c}_1^* := \mathbf{As} + \mathbf{e}_1 \in \{0,1\}^m, \mathbf{c}_2^* := (\mathbf{GH_{t^*}} + \mathbf{B})\mathbf{s} + \mathbf{T}'\mathbf{e}_1 \in \{0,1\}^m, \mathbf{k} = \mathsf{H}(\mathbf{c}_1^*, \mathbf{c}_2^*, \mathbf{s}, \mathbf{H_{t^*}}) \in \{0,1\}^{\ell}, \mathbf{c}_3^* := \mathsf{Enc}'_{\mathbf{k}}(\mathbf{m}_{b^*}) \in \{0,1\}^{\ell'}$, and returns the challenge ciphertext $(\mathbf{c}_1^*, \mathbf{c}_2^*, \mathbf{c}_3^*)$ to the adversary $\mathcal{A}$.

Phase 2. The adversary can make more decryption queries and the challenger $\mathcal{C}$ responds to $\mathcal{A}$ as in Phase 1.

Guess. Finally, $\mathcal{A}$ outputs a guess $b \in \{0,1\}$. If $b = b^*$, the challenger $\mathcal{C}$ outputs 1, else outputs 0.

Let W_i be the event that $\mathcal{C}$ outputs 1 in Game i for i in $\{1,2,3\}$.

Game 2. This Game is identical to Game 1 except that the challenge phase is changed as follows:

Challenge. After receiving two equal length plaintexts $\mathbf{m}_0$, $\mathbf{m}_1 \in \{0,1\}^{\ell'}$ from the adversary $\mathcal{A}$, the challenger $\mathcal{C}$ first randomly chooses a bit $b^* \xleftarrow{\$} \{0,1\}$, and $\mathbf{s} \xleftarrow{\$} U_n, \mathbf{e}_1 \xleftarrow{\$} \mathcal{B}_{\mu}^{m}$. Then, it calculates $\mathbf{c}_1^* := \mathbf{As} + \mathbf{e}_1 \in \{0,1\}^m, \mathbf{c}_2^* := (\mathbf{GH_{t^*}} + \mathbf{B})\mathbf{s} + \mathbf{Te}_1 \in \{0,1\}^m, \mathbf{k} = \mathsf{H}(\mathbf{c}_1^*, \mathbf{c}_2^*, \mathbf{s}, \mathbf{H_{t^*}}) \in \{0,1\}^{\ell}, \mathbf{c}_3^* := \mathsf{Enc}'_{\mathbf{k}}(\mathbf{m}_{b^*}) \in \{0,1\}^{\ell'}$, and returns the challenge ciphertext $(\mathbf{c}_1^*, \mathbf{c}_2^*, \mathbf{c}_3^*)$ to the adversary $\mathcal{A}$.

Lemma 3. $|\Pr[W_1] - \Pr[W_2]| \leq \mathsf{negl}(n)$

Proof. The only difference between Game 1 and Game 2 is that $\mathcal{C}$ replaces $\mathbf{c}_2^* := (\mathbf{GH}_{\mathbf{t}^*} + \mathbf{B})\mathbf{s} + \mathbf{T}'\mathbf{e}_1$ in Game 1 with $\mathbf{c}_2^* := (\mathbf{GH}_{\mathbf{t}^*} + \mathbf{B})\mathbf{s} + \mathbf{T}\mathbf{e}_1$ in Game 2. Next, we introduce a sequence of games $\{\text{Game}_{1,i}\}_{i\in[0,m]}$ between Game 1 and Game 2 to replace $\mathbf{T}'$ in the $\mathbf{c}_2^*$ row by row. Firstly, we define $\mathbf{T} = (\mathbf{t}_1, \cdots, \mathbf{t}_m)^\intercal, \mathbf{T}' = (\mathbf{t}'_1, \cdots, \mathbf{t}'_m)^\intercal$.

- $\text{Game}_{1,i}$, $i \in [m]$. This game is a hybrid of Game 1 and Game 2: the challenger $\mathcal{C}$ replaces $\mathbf{t}_i'^\intercal$ with $\mathbf{t}_i^\intercal$ in $\mathbf{c}_2^*$ during the challenge phase and keeps the remaining rows as in $\text{Game}_{1,i-1}$. Let $\text{Game}_{1,0}$ be Game 1. Obviously, $\text{Game}_{1,m}$ is identical to Game 2.

It suffices to show that $|\Pr[W_{1,i}] - \Pr[W_{1,i-1}]| \leq \mathsf{negl}(n)$ for any $i \in [m]$. The hardness of the EKLPN problem ensures that the probability for adversary $\mathcal{A}$ to distinguish $\text{Game}_{1,i}$ from $\text{Game}_{1,i-1}$ is negligible. Otherwise we can construct an algorithm $\mathcal{B}$ to solve EKLPN problem. Precisely, $\mathcal{B}$ is constructed by simulating $\text{Game}_{1,i}$ or Game $_{1,i-1}$ for $\mathcal{A}$. $\mathcal{B}$ is given a quadruple $(\mathbf{A}, (\bar{\mathbf{t}}_i^\intercal \mathbf{A})^\intercal, \mathbf{e}_1, \bar{z}_i)$, where $\bar{z}_i$ is either $\bar{\mathbf{t}}_i^\intercal \mathbf{e}_1$ or $\bar{\mathbf{t}}_i'^\intercal \mathbf{e}_1$. $\mathcal{B}$'s behavior is as follows.

KeyGen. $\mathcal{B}$ picks $\mathsf{H} \xleftarrow{\$} \mathcal{H}$, $\mathbf{T}_i = (\mathbf{t}_1, \cdots, \mathbf{r}_i, \cdots, \mathbf{t}_m)^\intercal$ and then $\mathcal{B}$ sets $\mathbf{B} = \left(\mathbf{A}^\intercal\mathbf{t}_1, \cdots, \boxed{\mathbf{A}^\intercal\bar{\mathbf{t}}_i}, \cdots, \mathbf{A}^\intercal\mathbf{t}_m\right)^\intercal$. Finally, $\mathcal{B}$ sends $pk = (\mathbf{A}, \mathbf{B})$ to the adversary $\mathcal{A}$, and keeps $sk = \mathbf{T}_i$ to itself. Note that the i^{th} row in $\mathbf{T}_i$ is chosen randomly and the i^{th} row in $\mathbf{B}$ is independent of it.

Phase 1. While receiving a decryption query $c = (\mathbf{t}, (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3))$ from adversary $\mathcal{A}$, $\mathcal{B}$ directly returns $\perp$ if $\mathbf{t} = \mathbf{t}^*$. Otherwise it first computes $\mathbf{y} = \mathbf{c}_2 - \mathbf{T}_i \cdot \mathbf{c}_1 = (\mathbf{GH}_\mathbf{t} + \mathbf{B}) \cdot \mathbf{s} + \mathbf{T}'\mathbf{e}_1 - \mathbf{T}_i(\mathbf{A} \cdot \mathbf{s} + \mathbf{e}_1) = \mathbf{GH}_\mathbf{t} \cdot \mathbf{s} + \underbrace{\begin{pmatrix} 0 \\ \vdots \\ (\bar{\mathbf{t}}_i^\intercal - \mathbf{r}_i^\intercal)\mathbf{As} \\ \vdots \\ 0 \end{pmatrix} + \begin{pmatrix} (\mathbf{t}_1'^\intercal - \mathbf{t}_1^\intercal)\mathbf{e}_1 \\ \vdots \\ (\mathbf{t}_i'^\intercal - \mathbf{r}_i^\intercal)\mathbf{e}_1 \\ \vdots \\ (\mathbf{t}_m'^\intercal - \mathbf{t}_m^\intercal)\mathbf{e}_1 \end{pmatrix}}_{\Delta_i}$, $\mathbf{H}_\mathbf{t}\mathbf{s} = \mathsf{Decode}(\mathbf{y})$.

Let $\mathbf{y} = \mathbf{GH}_\mathbf{t}\mathbf{s} + \Delta_i$, where $|\Delta_i| \leq \frac{2}{3}\alpha m + 1 < \alpha m$, $\mathsf{Decode}_\mathbf{G}$ also can handle correct $\mathbf{s}$ from $\mathbf{y}$. Then $\mathcal{B}$ checks that whether it satisfies that $|\mathbf{c}_1 - \mathbf{As}| \leq \beta \wedge |\mathbf{c}_2 - (\mathbf{GH}_\mathbf{t} + \mathbf{B})\mathbf{s}| \leq \frac{1}{3}\alpha m$. If yes it computes $\mathbf{k} = \mathsf{H}(\mathbf{c}_1, \mathbf{c}_2, \mathbf{s}, \mathbf{H}_\mathbf{t}), \mathbf{m} = \mathsf{Dec}'_\mathbf{k}(\mathbf{c}_3)$ otherwise lets $\mathbf{m} = \perp$. Finally it returns $\mathbf{m}$ to $\mathcal{A}$. Therefore, the decryption oracle can behave correctly.

Challenge. After receiving two equal length plaintexts $\mathbf{m}_0$, $\mathbf{m}_1 \in \{0,1\}^{\ell'}$ from the adversary $\mathcal{A}$, $\mathcal{B}$ first randomly chooses a bit $b^* \xleftarrow{\$} \{0,1\}$, and $\mathbf{s} \xleftarrow{\$} U_n, \mathbf{e}_1 \xleftarrow{\$} \mathcal{B}_\mu^m$. Then, it calculates $\mathbf{c}_1^* := \mathbf{As} + \mathbf{e}_1 \in \{0,1\}^m, \mathbf{c}_2^* = (\mathbf{GH}_{\mathbf{t}^*} + \mathbf{B})\mathbf{s} + \left(\mathbf{e}_1^\intercal\mathbf{t}_1, \cdots, \mathbf{e}_1^\intercal\mathbf{t}_{i-1}\boxed{\bar{z}_i}, \mathbf{e}_1^\intercal\mathbf{t}'_{i+1} \cdots, \mathbf{e}_1^\intercal\mathbf{t}'_m\right)^\intercal \in \{0,1\}^m, \mathbf{k} = \mathsf{H}(\mathbf{c}_1^*, \mathbf{c}_2^*, \mathbf{s}, \mathbf{H}_{\mathbf{t}^*}) \in \{0,1\}^\ell, \mathbf{c}_3^* := \mathsf{Enc}'_\mathbf{k}(\mathbf{m}_{b^*}) \in \{0,1\}^{\ell'}$, and returns the challenge ciphertext $(\mathbf{c}_1^*, \mathbf{c}_2^*, \mathbf{c}_3^*)$ to the adversary $\mathcal{A}$.

PHASE 2. The adversary can make more decryption queries and $\mathcal{B}$ responds to $\mathcal{A}$ as in Phase 1.

GUESS. Finally, $\mathcal{A}$ outputs a guess $b \in \{0,1\}$. If $b = b^*$, $\mathcal{B}$ outputs 1, else outputs 0.

If $\bar{z}_i = \bar{\mathbf{t}}_i'^{\intercal}\mathbf{e}_1$, then $\mathcal{B}$ simulates the behavior of the challenger in $\text{Game}_{1,i-1}$ exactly. Hence, $\Pr[W_{1,i-1}] = \Pr\left[\mathcal{B}(\mathbf{A}, (\bar{\mathbf{t}}_i^{\intercal}\mathbf{A})^{\intercal}, \mathbf{e}_1, \bar{\mathbf{t}}_i'^{\intercal}\mathbf{e}_1) = 1\right]$.

If $\bar{z}_i = \bar{\mathbf{t}}_i^{\intercal}\mathbf{e}_1$, then $\mathcal{B}$ simulates the behavior of the challenger in $\text{Game}_{1,i}$ exactly. Hence, $\Pr[W_{1,i-1}] = \Pr\left[\mathcal{B}(\mathbf{A}, (\bar{\mathbf{t}}_i^{\intercal}\mathbf{A})^{\intercal}, \mathbf{e}_1, \bar{\mathbf{t}}_i^{\intercal}\mathbf{e}_1) = 1\right]$.

Therefore, for $i \in [m]$, we have $|\Pr[W_{1,i-1}] - \Pr[W_{1,i}]| \leq \mathsf{negl}(n)$.

Game 3. This Game is identical to Game 2 except that the challenger $\mathcal{C}$ replaces $\mathbf{B} = \mathbf{TA}$ with $\mathbf{B}' = \mathbf{B} - \mathbf{GH}_{\mathbf{t}^*} \in \{0,1\}^{m \times n}$ in the key generation phase.

Lemma 4. $\Pr[W_3] = \Pr[W_2]$.

Proof. The only difference between Game 2 and Game 3 is that $\mathcal{C}$ replaces $\mathbf{B} = \mathbf{TA}$ in Game 2 with $\mathbf{B}' = \mathbf{B} - \mathbf{GH}_{\mathbf{t}^*}$ in Game 3. This means that the public key in Game 3 has the same distribution in Game 2. Thus we have $\Pr[W_3] = \Pr[W_2]$.

Game 4. This Game is identical to Game 3 except that the challenger $\mathcal{C}$ replaces $\mathbf{c}_1^* = \mathbf{As} + \mathbf{e}_1 \in \{0,1\}^m$ with $\mathbf{c}_1^* = \mathbf{u} \in \{0,1\}^m$ in the challenge phase. Note that in Game 2, $\mathbf{c}_2^* = (\mathbf{GH}_{\mathbf{t}^*} + \mathbf{B})\mathbf{s} + \mathbf{Te}_1 = \mathbf{GH}_{\mathbf{t}^*}\mathbf{s} + \mathbf{Tc}_1^*$. Therefore, in Game 3 we have $\mathbf{c}_2^* = (\mathbf{GH}_{\mathbf{t}^*} + \mathbf{B}')\mathbf{s} + \mathbf{Te}_1 = \mathbf{Tc}_1^*$.

Lemma 5. $|\Pr[W_4] - \Pr[W_3]| \leq \mathsf{negl}(n)$.

Proof. Since the only difference between Game 3 and Game 4 is that $\mathcal{C}$ replaces $\mathbf{c}_1^* = \mathbf{As} + \mathbf{e}_1 \in \{0,1\}^m$ in Game 3 with $\mathbf{c}_1^* = \mathbf{u} \in \{0,1\}^m$ in Game 4, we can construct a distinguisher $\mathcal{D}$ that distinguishes the distributions $(\mathbf{A}, \mathbf{A} \cdot \mathbf{s} + \mathbf{e})$ and $(\mathbf{A}, \mathbf{u})$ (where $\mathbf{u} \xleftarrow{\$} U_m$) with advantage $\mathsf{adv}(n)$ (assuming that $\mathcal{A}$ distinguishes 3 and Game 4 with non-negligible $\mathsf{adv}(n)$), contradicting the assumption. Thus we have $|\Pr[\mathcal{D}(\mathbf{A}, \mathbf{A} \cdot \mathbf{s} + \mathbf{e})] | - |\Pr[\mathcal{D}(\mathbf{A}, \mathbf{u})]| = |\Pr[W_3]| - |\Pr[W_4]| = \mathsf{adv}(n)$, which contradicts the assumption. This means that we have $|\Pr[W_3]| - |\Pr[W_4]| \leq \mathsf{negl}(n)$.

Lemma 6. $\Pr[W_4] = \frac{1}{2} + \mathsf{negl}(n)$.

Proof. This lemma follows from that the challenge ciphertext $(\mathbf{c}_1^*, \mathbf{c}_2^*)$ in game 4 is uniformly distributed. From $\mathcal{A}$'s view, $\mathbf{s}$ is perfectly hidden since $\mathbf{c}_1^*$ is uniformly distributed. The collision resistant hash function implies that it's nearly impossible for $\mathcal{A}$ to guess $\mathbf{k}$ correctly. Combining with the IND-CPA secure private-key encryption scheme it ensures that the advantage of the adversary $\mathcal{A}$ is negligible.

Note that the security requirement of private-key encryption scheme Π' is IND-CPA secure, for example an one-time pad scheme, since the replacement of

the pseudorandomness with randomness makes the challenge ciphertext perfectly random thus it is impossible for adversary to guess correctly with probability more than 1/2. Meanwhile it answers the decryption queries correctly. In all, we have $\Pr[W_1] = \frac{1}{2} + \mathsf{negl}(n)$, such that $\varepsilon = \mathsf{negl}(n)$. Thus we complete the proof.

Acknowledgement. The work was supported by the National Cryptography Development Fund (Grant No. MMJJ20180106).

References

1. Alekhnovich, M.: More on average case vs approximation complexity. In: 44th Annual Symposium on Foundations of Computer Science, pp. 298–307. IEEE, Cambridge, October 2003
2. Berlekamp, E., McEliece, R.J., van Tilborg, H.: On the inherent intractability of certain coding problems. IEEE Trans. Inf. Theor. **24**(3), 384–386 (1978)
3. 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, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22792-9_42
4. Blum, A., Kalai, A., Wasserman, H.: Noise-tolerant learning, the parity problem, and the statistical query model. J. ACM **50**(4), 506–519 (2003)
5. Boneh, D., Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. SIAM J. Comput. **36**(5), 1301–1328 (2006). https://doi.org/10.1137/S009753970544713X
6. Cramer, R., Damgård, I.: On the amortized complexity of zero-knowledge protocols. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 177–191. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03356-8_11
7. 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, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46447-2_27
8. 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, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34961-4_30
9. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Ladner, R.E., Dwork, C. (eds.) Proceedings of the 40th Annual ACM Symposium on Theory of Computing, pp. 197–206. ACM, Victoria, 17–20 May 2008
10. 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, Heidelberg (2006). https://doi.org/10.1007/11761679_6
11. 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, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54631-0_1
12. Levieil, É., Fouque, P.-A.: An improved LPN algorithm. In: De Prisco, R., Yung, M. (eds.) SCN 2006. LNCS, vol. 4116, pp. 348–359. Springer, Heidelberg (2006). https://doi.org/10.1007/11832072_24

13. 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/RANDOM -2005. LNCS, vol. 3624, pp. 378–389. Springer, Heidelberg (2005). https://doi.org/10.1007/11538462_32
14. May, A., Meurer, A., Thomae, E.: Decoding random linear codes in $\tilde{\mathcal{O}}(2^{0.054n})$. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 107–124. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25385-0_6
15. Micciancio, D., Peikert, C.: Trapdoors for lattices: simpler, tighter, faster, smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_41
16. Rackoff, C., Simon, D.R.: Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 433–444. Springer, Heidelberg (1992). https://doi.org/10.1007/3-540-46766-1_35
17. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Gabow, H.N., Fagin, R. (eds.) STOC, pp. 84–93. ACM (2005)
18. Rosen, A., Segev, G.: Chosen-ciphertext security via correlated products. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 419–436. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00457-5_25
19. Stern, J.: A method for finding codewords of small weight. In: Cohen, G., Wolfmann, J. (eds.) Coding Theory 1988. LNCS, vol. 388, pp. 106–113. Springer, Heidelberg (1989). https://doi.org/10.1007/BFb0019850
20. Yu, Y., Zhang, J.: Cryptography with auxiliary input and trapdoor from constant-noise LPN. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9814, pp. 214–243. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53018-4_9