Two Efficient Tag-Based Encryption Schemes on Lattices

Xueqing Wang[1,2(✉)], Biao Wang[1,2], and Rui Xue[1,2(✉)]

[1] State Key Laboratory of Information Security,
Institute of Information Engineering, Chinese Academy of Sciences,
Beijing 100093, China
{wangxueqing,wangbiao,xuerui}@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences,
Beijing 100049, China

Abstract. Tag-based encryption (TBE) is a generalization of public-key encryption (PKE), in which both the encryption and the decryption algorithms take a tag as an extra input, which is potentially useful. However, in contrast to TBE schemes with various types of security and under traditional number-theoretic assumptions, as far as we know, there is only one lattice-based TBE scheme with selective-tag security, which, in fact, is under a variant of DLWE assumption.

In this paper, we propose two efficient TBE schemes, both of which have adaptive-tag security and are under the standard DLWE assumption. For efficiency, we adopt, in both schemes, a particular q-ary lattice equipped with efficient LWE inversion and preimage sampling algorithms, which are efficiently available for solving the related problems on a general q-ary lattice. The probabilistic partition technique is used to achieve the adaptive-tag security. On the other hand, we mainly embed the preimage sampling problem into the first scheme and the LWE inversion problem into the second one, the latter of which has a smaller modulus and a smaller approximation factor.

Our schemes can be applied to construct IND-CCA2 secure PKE schemes and to design protocols that securely realizes the secure message transmission functionality in a hybrid model. Additionally, our first scheme can also be used to construct an adaptively secure identity-based encryption (IBE) scheme with more efficient secret-key extraction algorithm than those in well-known IBE schemes.

Keywords: Tag-based encryption · DLWE · Adaptive security
Probabilistic partitioning technique · G-trapdoor

1 Introduction

The notion of tag-based encryption (TBE) was proposed by MacKenzie et al. [MRY04], while it was originated implicitly from Shoup [Sho01] (where the tag is called label). TBE is a generalization of public-key encryption (PKE), in which

S. Qing et al. (Eds.): ICICS 2017, LNCS 10631, pp. 118–131, 2018.
https://doi.org/10.1007/978-3-319-89500-0_10

the encryption and decryption algorithms both take a tag as an extra input. All the ECIES and RSA-OAEP submissions and Shoup's proposal for an ISO standard of PKE include the notion of a tag (in the first two it is called an encoding parameter), although no indication was given as to the role or function of a tag.

As an independent primitive, in contrast to PKE, TBE has an additional ability to attach a tag to the ciphertext during the encryption process, while the tag is generally not included in the ciphertext and is explicitly given to the decryption algorithm. Such an explicit treatment of a tag has some notational advantages, when we consider an adversary who tries to alter the tag without affecting the ciphertext. The security of TBE can be similarly defined as that of PKE, as well as adding another dimension selective/adaptive-tag indicating whether the adversary submits the target tag before receiving the public key (selective-tag), or in the challenge phase together with a pair of chosen messages (adaptive-tag). And thus its security notions include indistinguishability against selective-tag/adaptive-tag chosen-plaintext/(weak) lunch-time/(weak) chosen-ciphertext attacks, which can be abbreviated, respectively, to IND-sTag-CPA, IND-aTag-CPA, IND-sTag-wCCA1, IND-sTag-CCA1, IND-aTag-CCA1, IND-sTag-wCCA2, IND-sTag-CCA2, IND-aTag-wCCA2 and IND-aTag-CCA2. Note that *w* here is short for *weak*, which means that the adversary is not allowed to query the target tag instead of the pair of the target tag and the challenge ciphertext to the decryption oracle.

As a cryptographic tool, IND-sTag-wCCA2 secure TBE schemes, belongs to a more general class of cryptographic schemes than selectively secure identity-based encryption (IBE) schemes, are sufficient to construct CCA secure PKE schemes, according to [Kil06]. Note that IND-aTag-CCA2 secure TBE schemes are equivalent with IND-CCA2 secure PKE schemes. In addition, IND-aTag-wCCA2 secure TBE schemes can be used, with the technique in [MRY04], to construct protocols that realizes the secure message transmission functionality in the universal composition framework.

TBE is an interesting cryptographic primitive and a useful tool from the above description. As far, except that IND-sTag-wCCA2 (IND-aTag-wCCA2, IND-aTag-CCA2, respectively) secure TBE schemes can be obtained from IND-sID-CPA (IND-aID-CPA, IND-aID-CCA2, respectively) secure IBE schemes by the generic transformation in [Kil06], in the traditional number-theoretic field, there is also an IND-aTag-wCCA2 secure TBE scheme [MRY04] and two IND-sTag-wCCA2 secure TBE schemes [Kil06]. Unfortunately, in the lattice-based field, there is only one IND-sTag-wCCA2 secure lattice-based TBE scheme [SLLF15], which, in fact, is under a variant of DLWE assumption. Our goal here is to construct more efficient lattice-based TBE schemes with stronger security under standard assumptions.

1.1 Our Results

In this paper, we present two IND-aTag-wCCA2 secure TBE schemes TBE1 and TBE2 both under the standard DLWE assumption. We compare the schemes

Table 1. The comparison between our schemes and the one from [SLLF15]

Scheme	Modulus q	Approximation factor α^{-1}	Security
[SLLF15]	$\omega(n^{12})$	$\sqrt{m} \cdot \omega(n^{11})$	IND-sTag-wCCA2
TBE1	$\omega\left(p^2\ell \cdot m^{2.5} \cdot (\log n)^2\right)$	$p^2 \cdot \ell \cdot m^2 \cdot \omega\left(\log n\right)^2$	IND-aTag-wCCA2
TBE2	$\omega\left((\ell + \sqrt{m}) \cdot m \cdot \log n\right)$	$(\ell + \sqrt{m}) \cdot \sqrt{m} \cdot \omega\left(\log n\right)$	IND-aTag-wCCA2

Here n is the security parameter, $\mathbb{Z}_p$ is the message space of TBE1, ℓ is the bit length of the tag, $\ell_q := \lceil \log_2 q \rceil$ and $m = \mathcal{O}(n\ell_q)$.

with the one in [SLLF15] on some aspects in Table 1, which shows that our schemes are more efficient with smaller moduli, are under weaker lattice assumptions with smaller approximation factors, and are with stronger security, than the TBE scheme from [SLLF15].

The main idea for constructing our schemes is combining the probabilistic partition technique of [ABB10] for the adaptively secure IBE scheme, originated from the work [Wat05], and the **G**-trapdoor as well as some efficient algorithms from [MP12]. In particular, an ingenious matrix, which comes from [Boy10], in the construction of [ABB10], is $\mathbf{A}_0 \| (\mathbf{B} + \sum_i \mathbf{id}[i]\mathbf{A}_i)$, where each entry of $\mathbf{id}$ is in $\{-1, 1\}$, whose trapdoor can be derived from that of $\mathbf{A}_0$, and is transformed, in the proof, into

$$\mathbf{A}_0 \big\| (\mathbf{A}_0 \sum_i \mathbf{id}[i]\mathbf{R}_i^* + (1 + \sum_i \mathbf{id}[i]h_i)\mathbf{B}), \tag{1}$$

by setting $\mathbf{A}_i := \mathbf{A}_0\mathbf{R}_i^* + h_i\mathbf{B}$, and the trapdoor of $\mathbf{B}$, instead of that of $\mathbf{A}_0$, is generated, the probabilistic partition for adaptively secure IBE is to separate the target identity $\mathbf{id}^*$ from the queried identities $\{\mathbf{id}_j\}$ by the term $1 + \sum_i \mathbf{id}[i]h_i$:

- If $1 + \sum_i \mathbf{id}^*[i]h_i = 0$, the trapdoor of $\mathbf{B}$ is not available, and the simulator's challenge can be embedded into the challenge ciphertext.
- If $1 + \sum_i \mathbf{id}_j[i]h_i \neq 0$, the trapdoor of $\mathbf{B}$ is used to generate that of matrix (1) and hence to generate the secret key for the queried identity $\mathbf{id}_j$.

According to [MP12], $\mathbf{R}_\mathbf{A}$ is a **G**-trapdoor of $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ if $\mathbf{A} = \left(\bar{\mathbf{A}} \| \left(\mathbf{HG} - \bar{\mathbf{A}}\mathbf{R}_\mathbf{A}\right)\right)$, where $\mathbf{H} \in \mathbb{Z}_q^{n\times n}$ is an invertible matrix, $\mathbf{G} \in \mathbb{Z}_q^{n\times w}$ is a primitive matrix, that is, its columns generate all of $\mathbb{Z}_q^n$ and $m \geq w \geq n$.

One of main challenges in the construction is that: If we retain $\mathbf{B}$, it is hard to extend the **G**-trapdoor of $\mathbf{B}$ to obtain that of matrix (1) in use of efficient algorithms in [MP12]. However, we observe that if we replace $\mathbf{B}$ with $\mathbf{G}$ and sample each entry of $\mathbf{R}_i^*$ independently from a proper discrete Gaussian distribution instead of the uniform distribution over $\{-1, 1\}$ as in [ABB10], the ingenious matrix, in the proof of our constructions, will be $\mathbf{A}_\mathbf{t} := \mathbf{A}_0 \big\| (\mathbf{A}_0 \sum_i \mathbf{t}[i]\mathbf{R}_i^* + (1 + \sum_i \mathbf{t}[i]h_i)\mathbf{G})$, where each entry of $\mathbf{t}$ is in $\{-1, 1\}$, and thus $-\sum_i \mathbf{t}[i]\mathbf{R}_i^*$ is just the trapdoor of matrix $\mathbf{A}_\mathbf{t}$ when $(1 + \sum_i \mathbf{t}[i]h_i) \neq 0$ in case that q is a prime, which is key to simulating successfully without the trapdoor of $\mathbf{A}_0$. And $-\sum_i \mathbf{t}^*[i]\mathbf{R}_i^*$ is used to construct the artificial noise in

the challenge ciphertext, which is a solution to make our schemes based on the DLWE assumption.

Additionally, we embed the preimage sampling problem into TBE1[1], which is the reason of adding an extra vector in the public key, and the LWE inversion problem into TBE2. And therefore, for the decryption of TBE1, a preimage sampling algorithm together with a trapdoor extension algorithm are enough.

It's more complex for the decryption of TBE2 since LWE samples are not generated in key generation as Regev encryption [Reg05], the secret key is a $\mathbf{G}$-trapdoor of the first part of LWE samples not the secret vector used to generate LWE samples and the message part will be lost if we execute the inversion algorithm on the second part of LWE samples (as the ciphertext, each entry of which is in $\mathbb{Z}_q$).

To overcome this obstacle, we observe that the inversion algorithm in [MP12] for $\Lambda(\mathbf{A}_\mathbf{t}^T)$ is essentially that for $\Lambda(\mathbf{G}^T)$ by transforming the former into the latter in use of the trapdoor $\mathbf{R}_{\mathbf{A}_\mathbf{t}}$ at first. To solve the above problem for TBE2, we map the message into an element in $\Lambda(\mathbf{G}^T)/2\Lambda(\mathbf{G}^T)$ and use a perturbed vector of $2\Lambda(\mathbf{A}_\mathbf{t})$ to hide the encoded message. And then in decryption, we get the transformed error by executing the first two steps of the inversion algorithm on the perturbed vector of $\Lambda(\mathbf{G}^T)$ and subtract it from the perturbed vector of $\Lambda(\mathbf{G}^T)/2\Lambda(\mathbf{G}^T)$, from which the message can be obtained by the inverse mapping. Note that the mapping is efficient to evaluate and to invert according to [MP12].

1.2 Applications

Adaptively Secure Identity-based Encryption. Although there is not, as far, a generic framework for transforming a TBE scheme into an IBE scheme, our scheme TBE1 can be easily transformed into an adaptively secure IBE scheme. For achieving this, we treat a tag as an identity, take the preimage vector of the vector in public key corresponding to the extended matrix for some identity as a secret key of the identity, and exploit the trapdoor extension algorithm and the preimage sampling algorithm to extract the key. Similar to those IBE schemes in [ZCZ16, Yam17], our derived IBE scheme is also an improved version of the adaptive one in [ABB10] (ABB-IBE). Specifically, Zhang et al. and Yamada used different techniques to make public parameters smaller and to make the DLWE assumption much stronger as well as the key extraction and the encryption more complex, which is a tradeoff. We just exploit $\mathbf{G}$-trapdoor and some related efficient algorithms to get a scheme with the most efficient key extraction algorithm. Their (variant of $\{0,1\}$-message space) comparisons are presented in Table 2.

[1] Although it seems able to construct a TBE scheme by simply, based on the dual-Regev encryption [GPV08], duplicating the same number of the image vectors and the preimage vectors as the bit length of a tag in key generation and just sum the image vectors indexed by the tag during the encryption, such a scheme is only IND-aTag-CPA secure.

Table 2. The comparison of the Series of ABB-IBE

Scheme		Master Public Key	Secret-Key Extraction	Encryption	Modulus q	Approximation Factor α^{-1}
ABB-IBE		$(\mathbb{Z}_q^{n\times m})^{\ell+2} \times \mathbb{Z}_q^n$	SampleLeft	Basic operations	$m^{2.5} \cdot \omega(\sqrt{\log n})$	$\ell^2 \cdot m^2 \cdot \omega(\sqrt{\log n})$
[ZCZ16](Type-II PHF)		$\mathbb{Z}_q^{n\times m} \times (\mathbb{Z}_q^{n\times n\ell_q})^{\mathcal{O}(\log n)+1} \times \mathbb{Z}_q^n$	Eval + TrapExt + SamplePre	Eval + Basic operations	$\widetilde{\mathcal{O}}(n^5m^2)$	$n^5m^{1.5} \cdot \omega(\sqrt{\log n})$
[Yam17]	F_{MAH}	$(\mathbb{Z}_q^{n\times m})^{\omega(\log^2 n)+2} \times \mathbb{Z}_q^n$	PubEval + TrapExt + SamplePre	PubEval + Basic operations	$\ell^2 \cdot n^{7.5}m^6 \cdot \omega(\log n)^{3.5}$	$\ell^2 \cdot n^3m^6 \cdot \omega((\log n)^{3.5})$
	F_{AFF}	$(\mathbb{Z}_q^{n\times m})^{\omega(\log n)+2} \times \mathbb{Z}_q^n$	PubEval + TrapExt + SamplePre	PubEval + Basic operations	$\mathcal{O}(n^{7.5}m^2 \cdot (\log n)^{3.5})$	$\mathcal{O}(n^7m^2 \cdot (\log n)^{3.5})$
Our IBE ← TBE1		$\mathbb{Z}_q^{n\times(m+n\ell_q)} \times (\mathbb{Z}_q^{n\times n\ell_q})^{\ell} \times \mathbb{Z}_q^n$	TrapExt + SamplePre	Basic operations	$\ell \cdot m^{2.5} \cdot (\log n)^2$	$\ell \cdot m^2 \cdot \omega((\log n)^2)$

[1] n is the security parameter, ℓ is the bit length of the identity in ABB-IBE, F_{MAH}-case in [Yam17] and our scheme, and $m = \mathcal{O}(n\ell_q)$. SampleLeft is based on the preimage sampling algorithm from [GPV08], which, according to [MP12], is rather complex and not suitable for practice, in either runtime or the quality of its outputs and these drawbacks are overcome by SamplePre from [MP12]. Additionally, TrapExt from [MP12] is also efficient. Both Eval and PubEval are quite complex, whose details are referred to [ZCZ16] and [Yam17], respectively.

Chosen-Ciphertext Secure Public-Key Encryption. Kiltz [Kil06] proposed a transformation that turns an IND-sTag-wCCA2 secure TBE scheme into a IND-CCA2 secure PKE scheme, together with a strongly one-time secure signature or a strongly one-time secure message authentication code. So we can construct two IND-CCA2 PKE schemes from our schemes TBE1 and TBE2, respectively. In case of strongly one-time signature, our resulted IND-CCA2 secure PKE schemes also have smaller moduli and are under weaker lattice assumptions, than the one in [SLLF15], according to Table 1.

Secure Message Transmission Functionality. Intuitively, the secure message transmission functionality allows multiple parties to send messages to a single receiver with preserving the secrecy and the integrity of the message. MacKenzie et al. [MRY04] adopted their IND-aTag-wCCA2 secure TBE scheme to design a protocol that securely realizes this functionality in a hybrid model. With the same technique, we can design two protocols, realizing the functionality, by using our two schemes, respectively.

2 Preliminaries

2.1 Basic Notation

In this paper, we use bold lower case letters (e.g. $\mathbf{a}, \mathbf{b}$) to denote column vectors and bold upper case letters (e.g. $\mathbf{A}, \mathbf{B}$) to denote matrices. For a matrix $\mathbf{A}$, $\mathbf{A}^{-1}$, $\mathbf{A}^T$ denote its inversion and transposition, respectively, $\mathbf{A}[i,j]$ denotes the entry in the i-th row and the j-th column, $\|\mathbf{A}\| := \max_{\mathbf{u}} \|\mathbf{Au}\|$ for all unit vectors $\mathbf{u}$ and the norm of a vector $\mathbf{x}$ is defined as $\|\mathbf{x}\| := \sqrt{\sum_i \mathbf{x}[i]^2}$, where

$\mathbf{x}[i]$ denotes the i-th entry of $\mathbf{x}$. For a positive integer n, let $\mathbf{I}_n$ denote the n-dimensional identity matrix. For an integer $q \geq 2$, the notation ℓ_q is $\lceil \log_2 q \rceil$. For a set S, then $s \xleftarrow{\$} S$ represents the operation of picking an element s from S uniformly at random. For $k \in \mathbb{N}$, then $[k]$ denotes the set $\{1, \ldots, k\}$. Let PPT short for probabilistic polynomial-time.

2.2 Lattices

In general, an m-dimensional lattice Λ is a discrete additive subgroup of $\mathbb{R}^m$. If Λ is generated as the set of all integer linear combinations of some k linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_k$, then k is called the rank and $\mathbf{B} := (\mathbf{b}_1, \ldots, \mathbf{b}_k)$ is called a basis of Λ, i.e., $\Lambda = \{\mathbf{Bz} : \mathbf{z} \in \mathbb{Z}^k\}$. In case of $k = m$, $\Lambda \subseteq \mathbb{Z}^m$ is called a full-rank integer lattice.

In this paper, we focus on a particular family of so-called q-ary integer lattices, which contain $q\mathbb{Z}^m$ as a sublattice for any positive integer q. For positive integers n, m, q and a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, define the following full-rank m-dimensional q-ary lattices: $\Lambda^{\perp}(\mathbf{A}) := \{\mathbf{z} \in \mathbb{Z}^m : \mathbf{Az} = \mathbf{0} \mod q\}$, $\Lambda(\mathbf{A}^T) := \{\mathbf{z} \in \mathbb{Z}^m : \exists\ \mathbf{s} \in \mathbb{Z}_q^n \text{ s.t. } \mathbf{z} = \mathbf{A}^T\mathbf{s} \mod q\}$. For any $\mathbf{u} \in \mathbb{Z}_q^n$ admitting an integer solution $\mathbf{x}$ to $\mathbf{Ax} = \mathbf{u} \mod q$, define the coset (or "shifted" lattice): $\Lambda_{\mathbf{u}}^{\perp}(\mathbf{A}) := \{\mathbf{z} \in \mathbb{Z}^m : \mathbf{Az} = \mathbf{u} \mod q\} = \Lambda^{\perp}(\mathbf{A}) + \mathbf{x}$.

2.3 Discrete Gaussians

For any $\mathbf{c} \in \mathbb{R}^m$ and a positive parameter $s \in \mathbb{R}$, the m-dimensional Gaussian function $\rho_{\mathbf{c},s} : \mathbb{R}^m \to (0, 1]$ is defined as: $\rho_{\mathbf{c},s}(\mathbf{x}) := \exp(-\frac{\pi\|\mathbf{x}-\mathbf{c}\|^2}{s^2})$. For a lattice $\Lambda \subset \mathbb{R}^m$, the discrete Gaussian distribution over Λ with center $\mathbf{c}$ and parameter s is defined as $D_{\Lambda+\mathbf{c},s}(\mathbf{x}) := \frac{\rho_{\mathbf{c},s}(\mathbf{x})}{\rho_{\mathbf{c},s}(\Lambda)}$, where $\rho_{\mathbf{c},s}(\Lambda) = \sum_{\mathbf{x}\in\Lambda} \rho_{\mathbf{c},s}(\mathbf{x})$.

Combining the result of Lemma 3.1 in [GPV08] with Lemma 4.4 in [MR07], we have the following tail bound on discrete Gaussians.

Lemma 1. *Let $\Lambda \subset \mathbb{R}^m$ be a lattice with basis $\mathbf{B}$, $\mathbf{c} \in span(\Lambda) := \{\mathbf{Br} : \mathbf{r} \in \mathbb{R}^k\}$ and $s \geq \|\widetilde{\mathbf{B}}\| \cdot \omega(\sqrt{\log m})$, where $\widetilde{\mathbf{B}}$ is the Gram-Schmidt orthogonalization of $\mathbf{B}$, we have $\Pr_{\mathbf{x}\leftarrow D_{\Lambda+\mathbf{c},s}}[\|\mathbf{x}\| \geq s\sqrt{m}] = \mathrm{negl}(m)$.*

For positive $\alpha \in \mathbb{R}$, Ψ_α is defined to be the distribution on $\mathbb{T} := \mathbb{R}/\mathbb{Z}$ of a normal variable with mean 0 and standard deviation $\alpha/\sqrt{2\pi}$, reduced modulo 1. And its discretization $\bar{\Psi}_\alpha$ is the discrete distribution over $\mathbb{Z}_q$, for integer $q \geq 2$, of the random variable $\lfloor q \cdot X_{\Psi_\alpha} \rceil \mod q$, where X_{Ψ_α} has distribution Ψ_α.

2.4 Learning with Errors (LWE)

The LWE problem was introduced by Regev [Reg05]. Decisional LWE (DLWE) is defined as follows.

Definition 1 (DLWE). *For security parameter λ, let $n = n(\lambda)$ be an integer dimension, let an integer $q = q(\lambda) \geq 2$ be a modulus, and $\bar{\Psi}_\alpha$ be a noise distribution. The decisional learning with errors problem, denoted by $\mathsf{DLWE}_{n,q,\bar{\Psi}_\alpha}$, is to distinguish the following two distributions: In the first distribution, denoted by $U(\mathbb{Z}_q^n \times \mathbb{Z}_q)$, one samples $(\mathbf{a}, b)$ uniformly from $\mathbb{Z}_q^n \times \mathbb{Z}_q$. In the second distribution, denoted by $A_{\mathbf{s},\bar{\Psi}_\alpha}$ for uniformly random $\mathbf{s} \in \mathbb{Z}_q^n$, one samples $(\mathbf{a}, b) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ by sampling $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_q^n$ uniformly at random, $e \leftarrow \bar{\Psi}_\alpha$, and setting $b = \langle \mathbf{a}, \mathbf{s} \rangle + e$. The $\mathsf{DLWE}_{n,q,\bar{\Psi}_\alpha}$ assumption is that the $\mathsf{DLWE}_{n,q,\bar{\Psi}_\alpha}$ problem is infeasible.*

There are known quantum [Reg05] and classical [Pei09] reductions between $\mathsf{DLWE}_{n,q,\bar{\Psi}_\alpha}$ and approximating short vector problems on lattices. In particular, for $\alpha q \geq 2\sqrt{n}$, solving the $\mathsf{DLWE}_{n,q,\bar{\Psi}_\alpha}$ problem is at least as hard as solving worst-case lattice problems with approximation factors of $\widetilde{\mathcal{O}}(n/\alpha)$.

2.5 Trapdoors for Lattices

It is much required to generate a (nearly) uniform parity-check matrix $\mathbf{A}$ together with some strong trapdoor for advanced lattice-based cryptographic schemes, including chosen-ciphertext secure encryption, "hash-and-sign" digital signature, identity-based encryption, et al.

In 1999, Ajtai [Ajt99] showed how to sample an essentially uniform $\mathbf{A}$, along with a relatively short trapdoor $\mathbf{S} \subset \Lambda^\perp(\mathbf{A})$. And later Micciancio and Goldwasser [MG02] and Gentry et al. [GPV08] successively improved the result slightly.

Lemma 2 (Trapdoor Generation I [Ajt99,GPV08]). *For any prime $q = \mathrm{poly}(n)$ and any integer $m \geq 5n\ell_q$, there exists a PPT algorithm that, on input 1^n, outputs a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a full-rank set $\mathbf{S} \subset \Lambda^\perp(\mathbf{A})$, where the distribution of $\mathbf{A}$ is statistically close to uniform over $\mathbb{Z}_q^{n \times m}$ and the length $\|\mathbf{S}\| \leq L = m^{1+\varepsilon}$ for any $\varepsilon > 0$.*

In particular, by Lemma 7.1 in [MG02], given an arbitrary basis of $\Lambda^\perp(\mathbf{A})$, the full-rank set $\mathbf{S}$ can be converted efficiently to a good basis $\mathbf{T}$ such that $\|\widetilde{\mathbf{T}}\| \leq \|\widetilde{\mathbf{S}}\|$.

In 2011, Alwen and Peikert [AP11] elucidated and generalized Ajtai's algorithm to provide a basis of essentially optimal length.

Lemma 3 (Trapdoor Generation II [AP11]). *For any integer $q \geq 2$ and $m \geq 2n\ell_q^2$, there exists a PPT algorithm that, on inputs n, q and m, outputs a nearly uniform matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a basis $\mathbf{S}$ of $\Lambda^\perp(\mathbf{A})$ with $\|\widetilde{\mathbf{S}}\| \leq 5\sqrt{n\ell_q}$.*

In 2012, Micciancio and Peikert [MP12] proposed a significantly more efficient algorithm, which essentially amounts to just one multiplication of two random matrices. For any positive integers n and $q \geq 2$, let $\mathbf{G} := \mathbf{I}_n \otimes \mathbf{g}^T \in \mathbb{Z}_q^{n \times n\ell_q}$, where $\mathbf{g}^T = (2^0, 2^1, \ldots, 2^{\ell_q - 1})$.[2]

[2] This $\mathbf{g}$ can be generalized into ones using other bases.

Lemma 4 (Trapdoor Generation III [MP12]**).** *For any positive integers* $n, q \geq 2, m = \mathcal{O}(n\ell_q)$*, there exists a PPT algorithm* TrapGen *that, on inputs a uniform matrix* $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n\times m}$ *and an invertible matrix* $\mathbf{H} \in \mathbb{Z}_q^{n\times n}$,[3] *outputs a statistically near-uniform matrix* $\mathbf{A} \in \mathbb{Z}_q^{n\times(m+n\ell_q)}$ *and its* $\mathbf{G}$*-trapdoor* $\mathbf{R} \in \mathbb{Z}^{m\times n\ell_q}$ *w.r.t.* $\mathbf{H}$*, by firstly choosing each element of* $\mathbf{R}$ *independently from a proper discrete Gaussian distribution* D *over* $\mathbb{Z}$*, and then setting* $\mathbf{A} := (\bar{\mathbf{A}}\|(\mathbf{HG} - \bar{\mathbf{A}}\mathbf{R}))$*. Note that* $\|\mathbf{R}\| \leq \mathcal{O}(\sqrt{m} + \sqrt{n\ell_q}) \cdot \omega(\sqrt{\log n}) = \mathcal{O}(\sqrt{n\ell_q}) \cdot \omega(\sqrt{\log n})$ *except with probability* $2^{-\Omega(m+n\ell_q)}$*.*

Based on the above $\mathbf{G}$-trapdoor, Micciancio and Peikert constructed efficient parallel algorithms for preimage sampling over a shifted lattice, LWE inversion and trapdoor extension, respectively.

Lemma 5 (Preimage Sampling [MP12]**).** *For parameters given in Lemma 4, some* $s \in \mathbb{R}$ *and a uniformly random vector* $\mathbf{u} \in \mathbb{Z}_q^n$*, there exists a PPT algorithm* SamplePre *that, on inputs* $\mathbf{R}, \bar{\mathbf{A}}, \mathbf{H}, \mathbf{u}, s$*, outputs a vector* $\mathbf{e}$*, whose distribution is statistically close to* $D_{\mathbb{Z},s\cdot\omega(\sqrt{\log n})}^{m+n\ell_q}$*, satisfying* $(\bar{\mathbf{A}}\|(\mathbf{HG} - \bar{\mathbf{A}}\mathbf{R})) \cdot \mathbf{e} = \mathbf{u} \mod q$*.*

Lemma 6 (LWE Inversion [MP12]**).** *For parameters given in Lemma 4, a vector* $\mathbf{b} = \mathbf{A}^T\mathbf{s} + \mathbf{e}$ *for any uniform* $\mathbf{s} \in \mathbb{Z}_q^n$ *and suitably small* $\mathbf{e} \in \mathbb{Z}^{m+n\ell_q}$*, there exists a PPT algorithm* Invert *that, on inputs* $\mathbf{R}, \bar{\mathbf{A}}, \mathbf{H}, \mathbf{b}$*, outputs* $\mathbf{s}$ *and* $\mathbf{e}$*, by first transforming the perturbed vector* $\mathbf{b}$ *w.r.t.* $\Lambda(\mathbf{A}^T)$ *into* $\mathbf{b}' := (\mathbf{R}^T\|\mathbf{I}_{n\ell_q}) \cdot \mathbf{b}$ *w.r.t.* $\Lambda(\mathbf{G}^T)$*, then obtaining the solution* $(\mathbf{s}', \mathbf{e}')$ *in use of the inversion algorithm for* $\Lambda(\mathbf{G}^T)$*, and finally computing* $\mathbf{s} := (\mathbf{H}^{-1})^T\mathbf{s}'$*,* $\mathbf{e} := \mathbf{b} - \mathbf{A}^T\mathbf{s}$*.*

Lemma 7 (Trapdoor Extension [MP12]**).** *For parameters given in Lemma 4, a uniform matrix* $\mathbf{B} \in \mathbb{Z}_q^{n\times n\ell_q}$*, there exists a PPT algorithm* TrapExt *that, on inputs a* $\mathbf{G}$*-trapdoor* $\mathbf{R}$ *for* $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ *w.r.t. some invertible* $\mathbf{H}$*, an extension* $\mathbf{A}' = (\mathbf{A}\|\mathbf{B})$ *of* $\mathbf{A}$*, an invertible* $\mathbf{H}' \in \mathbb{Z}_q^{n\times n}$ *and* $s \in \mathbb{R}$*, outputs a* $\mathbf{G}$*-trapdoor* $\mathbf{R}'$ *for* $\mathbf{A}'$ *w.r.t.* $\mathbf{H}'$*. Particularly, the* i*-th column of* $\mathbf{R}'$ *is sampling independently from a discrete Gaussian with parameter* s *over* $\Lambda^{\perp}_{(\mathbf{H}'\mathbf{G}-\mathbf{B})[i]}(\mathbf{A})$ *in use of* $\mathbf{R}$*, where* $(\mathbf{H}'\mathbf{G} - \mathbf{B})[i]$ *is the* i*-th column of* $(\mathbf{H}'\mathbf{G} - \mathbf{B})$*. Note that* $\|\mathbf{R}'\| \leq s \cdot \mathcal{O}(\sqrt{m} + \sqrt{n\ell_q})$ *except with negligible probability.*

2.6 Tag-Based Encryption

Informally, in a tag-based encryption scheme, both the encryption and decryption algorithms take an additional tag as input. A tag may be a binary string of appropriate length or has any particular internal structure. We recall its definition from [Kil06].

A tag-based encryption scheme, with message space $\mathcal{M}$ and tag space $\mathcal{T}$, consists of three polynomial-time algorithms (Gen, Enc, Dec) described as follows:

- $\mathsf{Gen}(1^\lambda) \to (pk, sk)$**:** A probabilistic algorithm that takes the security parameter 1^λ as input, generates and outputs a pair of public key and private key (pk, sk).

[3] $\mathbf{H}$ here and below can be $\mathbf{I}_n$.

- $\mathsf{Enc}(pk, t \in \mathcal{T}, \mu \in \mathcal{M}) \rightarrow c$**:** A probabilistic algorithm that takes the public key pk, a tag t and a message μ as input, generates and outputs a ciphertext c. Note that the tag is not explicitly contained in the ciphertext.
- $\mathsf{Dec}(sk, t, c) \rightarrow \mu$**:** A deterministic or probabilistic algorithm that takes the secret key sk, a tag t and a ciphertext c as input, generates and outputs a message μ if t is valid and is just the tag used to generate c, and outputs $\perp$ meaning decryption failure otherwise.

The correctness and security are defined as follows:

- **Correctness.** For all $\lambda \in \mathbb{N}$, all tags $t \in \mathcal{T}$ and all messages $\mu \in \mathcal{M}$, we have $\Pr\left[\mathsf{Dec}\left(sk, t, \mathsf{Enc}\left(pk, t, \mu\right)\right) = \mu\right] = 1$, where the probability is taken over the choice of $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$, and the coins of the algorithms in the expression.
- **Security.** Due to known applications of TBE, there is only some security definitions, IND-sTag-wCCA2, IND-aTag-wCCA2 and IND-aTag-CCA2, for TBE. In fact, other standard security definitions for TBE can be easily defined corresponding to those for PKE. In this paper, we focus on IND-aTag-wCCA2 security, which is defined as a game executed by a challenger $\mathcal{C}$ and a PPT adversary $\mathcal{A}$ interactively.
  - **Key Generation Phase.** The challenger $\mathcal{C}$ calls $\mathsf{Gen}(1^\lambda)$ to generate (pk, sk) and sends pk to the adversary $\mathcal{A}$.
  - **Decryption Query Phase I.** On the query (t, c) from $\mathcal{A}$, $\mathcal{C}$ acts as the decryption oracle: Calls $\mathsf{Dec}(sk, t, c)$ to generate a message μ or the failure symbol $\perp$ as the answer to $\mathcal{A}$.
  - **Challenge Phase.** Once $\mathcal{A}$ submits t^* and a pair of messages (μ_0, μ_1) with the same length, $\mathcal{C}$ samples a random bit b, calls $\mathsf{Enc}(pk, t^*, \mu_b)$ to generate the challenge ciphertext c^* and sends it to $\mathcal{A}$.
  - **Decryption Query Phase II.** On the query (t, c) from $\mathcal{A}$, $\mathcal{C}$ checks whether $t = t^*$: if yes, $\mathcal{C}$ aborts the game and outputs a random bit; else, $\mathcal{C}$ answers to $\mathcal{A}$ as in the decryption query phase I.
  - **Guess Phase.** Once $\mathcal{A}$ submits its guess b', $\mathcal{C}$ checks whether $b' = b$: if yes, it outputs 1, and outputs 0 otherwise.

  The advantage of $\mathcal{A}$ is defined as

$$\mathbf{Adv}_{\mathcal{A}}^{\text{aTag-wCCA2}}(\lambda) := |\Pr[\mathcal{C} \text{ outputs } 1] - 1/2|, \tag{2}$$

  a TBE scheme is said to be IND-aTag-wCCA2 secure if the advantage function (2) is negligible for all PPT adversaries $\mathcal{A}$.

3 Tag-Based Encryption Scheme TBE1

In this section, we construct the first scheme TBE1, in which the preimage sampling problem is mainly embedded. Specifically, this scheme has an encryption similar to the dual-Regev encryption, where the relation $\mathbf{Ae} = \mathbf{u} \mod q$ is the core. And hence there is an image vector $\mathbf{u}$ in the public key, its corresponding

LWE value is used to hide the message, the tag is bound to the part of $\mathbf{A}$ and the preimage vector $\mathbf{e}$ will first be sampled for decryption.

Let n be the security parameter, a prime $q = \text{poly}(n)$, $\alpha \in (0,1)$ such that $\alpha q \geq 2\sqrt{n}$, $m = \mathcal{O}(n\ell_q)$ and D is the distribution used in Lemma 4. The tag space is $\mathcal{T} = \{0,1\}^\ell$ and the message space is $\mathbb{Z}_p$ for some $2 \leq p < q$.

- $\mathsf{TBE1.Gen}(1^n)$**:** Sample $\bar{\mathbf{A}}_0 \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$ and run $(\mathbf{A}_0, \mathbf{T}_{\mathbf{A}_0}) \leftarrow \mathsf{TrapGen}(\bar{\mathbf{A}}_0, \mathbf{I}_n)$, where $\mathbf{A}_0 := \left(\bar{\mathbf{A}}_0 \middle\| \left(\mathbf{G} - \bar{\mathbf{A}}_0\mathbf{T}_{\mathbf{A}_0}\right)\right)$. Choose $\mathbf{A}_1, \ldots, \mathbf{A}_\ell \xleftarrow{\$} \mathbb{Z}_q^{n\times n\ell_q}$, $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^n$, output $pk := (\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_\ell, \mathbf{u})$ and $sk := \mathbf{T}_{\mathbf{A}_0}$.
- $\mathsf{TBE1.Enc}(pk, \mathbf{t} \in \mathcal{T}, \mu \in \mathbb{Z}_p)$**:** Sample $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{R}_i \leftarrow D^{(m+n\ell_q)\times n\ell_q}$ for $i \in [\ell]$, $x \leftarrow \bar{\Psi}_\alpha$, $\mathbf{y} \leftarrow \bar{\Psi}_\alpha^{m+n\ell_q}$, let $\mathbf{A_t} := (\mathbf{A}_0 \| (\mathbf{G} + \sum_{i=1}^{\ell}(-1)^{\mathbf{t}[i]}\mathbf{A}_i), \mathbf{R_t} := \sum_{i=1}^{\ell}(-1)^{\mathbf{t}[i]}\mathbf{R}_i, \mathbf{z} := -\mathbf{R_t}^T\mathbf{y}$, compute and output $\mathbf{c} := (\mathbf{u}\|\mathbf{A_t})^T\mathbf{s} + (x, \mathbf{y}^T, \mathbf{z}^T)^T + (\mu \cdot \lfloor q/p \rfloor, \mathbf{0}_{1\times(m+2n\ell_q)})^T \mod q$.
- $\mathsf{TBE1.Dec}(sk, \mathbf{t}, \mathbf{c})$**:** First derive a trapdoor $\mathbf{T}_{\mathbf{A_t}} \leftarrow \mathsf{TrapExt}(\mathbf{T}_{\mathbf{A}_0}, \mathbf{A_t}, \mathbf{I}_n, \|\mathbf{T}_{\mathbf{A}_0}\|)$. And then sample $\mathbf{e_t} \leftarrow \mathsf{SamplePre}(\mathbf{T}_{\mathbf{A_t}}, \mathbf{A_t}, \mathbf{u}, \|\mathbf{T}_{\mathbf{A_t}}\|)$, such that $\mathbf{A_t}\mathbf{e_t} = \mathbf{u} \mod q$, and compute $\delta := (1, -\mathbf{e_t}^T) \cdot \mathbf{c}/q$. Finally find and output $\mu \in \mathbb{Z}_p$ such that $\delta - \mu/p$ is closest to 0 modulo 1.

Lemma 8 (Correctness). *Let a prime $q = \omega(p^2\ell \cdot (n\ell_q)^{2.5} \cdot (\log n)^2)$ and $\alpha < (p^2\ell \cdot (n\ell_q)^2 \cdot \omega((\log n)^2))^{-1}$. Then TBE1.Dec works with overwhelming probability.*

Proof. In the decryption algorithm, $\mathsf{TrapExt}$ and $\mathsf{SamplePre}$ are firstly called and their correctness are guaranteed by Lemmas 7 and 5 respectively. Subsequently, $(1, -\mathbf{e_t}^T) \cdot \mathbf{c} = \mu \cdot (q/p) - \mu \cdot (q/p - \lfloor q/p \rfloor) + x - \mathbf{e_t}^T(\mathbf{y}^T, \mathbf{z}^T)^T \mod q$, in which the error term is $-\mu \cdot (q/p - \lfloor q/p \rfloor) + x - (\mathbf{e}_{\mathbf{t},1} - \mathbf{R}\mathbf{e}_{\mathbf{t},2})^T\mathbf{y}$, if we parse $\mathbf{e_t}$ as $(\mathbf{e}_{\mathbf{t},1}^T, \mathbf{e}_{\mathbf{t},2}^T)^T$.
According to Lemmas 1, 4 and 5, we have

$$\|\mathbf{e_t}\| \leq \|\mathbf{T}_{\mathbf{A_t}}\| \cdot \omega(\sqrt{\log n}) \cdot \sqrt{m + 2n\ell_q} \leq \mathcal{O}((n\ell_q)^{1.5}) \cdot \omega(\log n),$$

and since $\|\mathbf{R}\| \leq \ell \cdot \mathcal{O}(\sqrt{m + n\ell_q} + \sqrt{n\ell_q}) \cdot \omega(\sqrt{\log n}) = \ell \cdot \mathcal{O}(\sqrt{n\ell_q}) \cdot \omega(\sqrt{\log n})$ by Lemma 4, $\|\mathbf{e}_{\mathbf{t},1} - \mathbf{R}\mathbf{e}_{\mathbf{t},2}\| \leq \|\mathbf{e}_{\mathbf{t},1}\| + \|\mathbf{R}\mathbf{e}_{\mathbf{t},2}\| \leq \ell \cdot \mathcal{O}((n\ell_q)^2) \cdot \omega((\log n)^{1.5})$, and hence by Lemma 12 in [ABB10],

$$\begin{aligned}|(\mathbf{e}_{\mathbf{t},1} - \mathbf{R}\mathbf{e}_{\mathbf{t},2})^T\mathbf{y}| \leq \|\mathbf{e}_{\mathbf{t},1} - \mathbf{R}\mathbf{e}_{\mathbf{t},2}\| \cdot \alpha q \cdot \omega(\sqrt{\log(m + n\ell_q)}) \\ + \|\mathbf{e}_{\mathbf{t},1} - \mathbf{R}\mathbf{e}_{\mathbf{t},2}\| \cdot \sqrt{m + n\ell_q}/2,\end{aligned}$$

and thus $|-\mu \cdot (q/p - \lfloor q/p \rfloor) + x - (\mathbf{e}_{\mathbf{t},1} - \mathbf{R}\mathbf{e}_{\mathbf{t},2})^T\mathbf{y}|$ is less than

$$p + \frac{2p^2\sqrt{n\ell_q}}{2p} + \frac{2p^2\ell \cdot (n\ell_q)^{2.5} \cdot \omega((\log n)^2)}{2p} + \ell \cdot \mathcal{O}((n\ell_q)^{2.5}) \cdot \omega((\log n)^{1.5}) < \frac{q}{2p},$$

therefore $|\delta - \mu/p| = |(1, -\mathbf{e_t}^T) \cdot \mathbf{c}/q - \mu/p| = |\,\mu/p + 1/q \cdot (-\mu \cdot (q/p - \lfloor q/p \rfloor) + x - \mathbf{e_t}^T(\mathbf{y}^T, \mathbf{z}^T)^T) - \mu/p| < 1/(2p)$,
so $\mathsf{TBE1.Dec}$ outputs μ as desired. □

Theorem 1 (Security). *The above scheme TBE1 is IND-aTag-wCCA2 secure in the standard model if the* $\mathsf{DLWE}_{n,q,\bar{\Psi}_\alpha}$ *assumption holds.*

Proof. To prove the theorem, we suppose that an arbitrary PPT adversary $\mathcal{A}$ against IND-aTag-wCCA2 security of TBE1. And we consider the following games, each of which is described as the modification from its previous one.

Game 0. This is the original IND-aTag-wCCA2 security experiment between $\mathcal{A}$ and a challenger $\mathcal{C}$. In addition, $\mathcal{C}$ maintains a list T for storing queried tags.

Game 1. In preparation for use of probabilistic partition technique, we slightly change the way that $\mathcal{C}$ generates the matrices $\mathbf{A}_i$ for all $i \in [\ell]$: At the key generation phase, $\mathcal{C}$ chooses $\mathbf{R}_i^*$ as in Game 0 and also chooses random scalar $h_i \in \mathbb{Z}_q$ uniformly, and then it uses $\mathbf{A}_0$ generated as in Game 0 to construct $\mathbf{A}_i$ as $\mathbf{A}_i := -\mathbf{A}_0\mathbf{R}_i^* + h_i\mathbf{G}$.

Game 2. After receiving $\mathcal{A}$'s guess, we start to partition the challenge tag $\mathbf{t}^*$ from $\mathcal{A}$'s queried tags T by introducing the abort check with the abort-resistant function $H_{\mathbf{h}}(\mathbf{t}) := 1 + \sum_{i=1}^{\ell}(-1)^{\mathbf{t}[i]}h_i$, and then use an artificial abort to force the probability of aborting to be independent of $\mathcal{A}$'s particular queries, as in [Wat05, ABB10, ZCZ16].

Game 3. $\mathcal{C}$ chooses $\mathbf{A}_0$ uniformly at random from $\mathbb{Z}_q^{n\times(m+n\ell_q)}$. For answering the decryption query on $(\mathbf{t}_i, \mathbf{c}_i)$, since

$$\mathbf{A}_{\mathbf{t}} = (\mathbf{A}_0 \| (-\mathbf{A}_0 \sum_{i=1}^{\ell}(-1)^{\mathbf{t}[i]}\mathbf{R}_i^* + H_{\mathbf{h}}(\mathbf{t}) \cdot \mathbf{G})), \tag{3}$$

$\mathcal{C}$ first computes $H_{\mathbf{h}}(\mathbf{t}_i)$ and checks if $H_{\mathbf{h}}(\mathbf{t}_i) = 0$: If yes, it aborts the game and outputs a random bit; else, it computes $\mathbf{A}_{\mathbf{t}_i}$ as in (3), uses its trapdoor $\sum_{j=1}^{\ell}(-1)^{\mathbf{t}_i[j]}\mathbf{R}_j^*$ to generate $\mathbf{e}_{\mathbf{t}_i}$, and finally it exploits $\mathbf{e}_{\mathbf{t}_i}$ to decrypt $\mathbf{c}_i$ and sends the result to $\mathcal{A}$.

At the challenge phase, once receiving $\mathbf{t}^*$ from $\mathcal{A}$, $\mathcal{C}$ first computes $H_{\mathbf{h}}(\mathbf{t}^*)$ and checks whether it equals to 0: If not, it aborts the game and outputs a random bit; else, it generates a challenge ciphertext as in Game 2.

At the guess phase, $\mathcal{C}$ just performs the artificial abort as in Game 2.

Game 4. This game is identical to Game 3 except that the challenge ciphertext is chosen as a random element in $\mathbb{Z}_q^{m+2n\ell_q+1}$.

4 Tag-Based Encryption Scheme TBE2 with Smaller Modulus and Approximation Factor

In this section, our second scheme TBE2 is presented. TBE2 has a smaller modulus, which is a key factor of efficiency, and a smaller approximation factor,

which means a weaker lattice assumption, than TBE1. In particular, we mainly embed the LWE inversion problem without the image vector in the public key and hence the preimage sampling algorithm is not available. LWE samples are also generated freshly for hiding the message in the encryption, which has to be decrypted by recovering the secret vector or the noise.

The tag space is similar to that of TBE1. The message space is $\{0,1\}^{n\ell_q}$, according to [MP12], which can be mapped bijectively to the coset of $\Lambda(\mathbf{G}^T)/2\Lambda(\mathbf{G}^T)$ via a function $f : \{0,1\}^{n\ell_q} \to \mathbb{Z}^{n\ell_q}$. Note that f is efficient to evaluate and to invert and its inversion is denoted as f^{-1}.

- $\mathsf{TBE2.Gen}(1^n)$**:** Similar to $\mathsf{TBE1.Gen}(1^n)$ without sampling $\mathbf{u}$, output $pk := (\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_\ell)$ and $sk := \mathbf{T}_{\mathbf{A}_0}$.
- $\mathsf{TBE2.Enc}(pk, \mathbf{t} \in \mathcal{T}, \boldsymbol{\mu} \in \{0,1\}^{n\ell_q})$**:** Sample $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e}_1 \leftarrow \bar{\Psi}_\alpha^{m+n\ell_q}$, $\mathbf{R}_i \leftarrow D^{(m+n\ell_q)\times n\ell_q}$ for $i \in [\ell]$, where D is the distribution used in Lemma 4. Let $\mathbf{A}_\mathbf{t} := (\mathbf{A}_0 \| (\mathbf{G} + \sum_{i=1}^{\ell}(-1)^{\mathbf{t}[i]}\mathbf{A}_i)), \mathbf{R}_\mathbf{t} := \sum_{i=1}^{\ell}(-1)^{\mathbf{t}[i]}\mathbf{R}_i, \mathbf{e}_2 := -\mathbf{R}_\mathbf{t}^T\mathbf{e}_1$, compute and output $\mathbf{c} := 2(\mathbf{A}_\mathbf{t}^T\mathbf{s} \mod q) + (\mathbf{e}_1^T, \mathbf{e}_2^T)^T + (\mathbf{0}_{1\times(m+n\ell_q)}, f(\boldsymbol{\mu})^T)^T \mod 2q$.
- $\mathsf{TBE2.Dec}(sk, \mathbf{t}, \mathbf{c})$**:** Let $\mathbf{T}_{\mathbf{A}_\mathbf{t}} \leftarrow \mathsf{TrapExt}(\mathbf{T}_{\mathbf{A}_0}, \mathbf{A}_\mathbf{t}, \mathbf{I}_n, \|\mathbf{T}_{\mathbf{A}_0}\|)$, where $\mathbf{A}_0\mathbf{T}_{\mathbf{A}_\mathbf{t}} = \mathbf{G} - (\mathbf{G} + \sum_{i=1}^{\ell}(-1)^{\mathbf{t}[i]}\mathbf{A}_i) = -\sum_{i=1}^{\ell}(-1)^{\mathbf{t}[i]}\mathbf{A}_i$. And to compute $(\mathbf{z} \in \mathbb{Z}_q^n, \mathbf{e} \in \mathbb{Z}^{n\ell_q})$ as in Lemma 6 on inputs $(\mathbf{T}_{\mathbf{A}_\mathbf{t}}, \mathbf{A}_\mathbf{t}, \mathbf{c} \mod q)$ as follows:
  1. $\mathbf{b} := (\mathbf{T}_{\mathbf{A}_\mathbf{t}}^T \| \mathbf{I}_{n\ell_q}) \cdot (\mathbf{c} \mod q)$ as a perturbed vector w.r.t. $\Lambda(\mathbf{G}^T)$;
  2. run the inversion algorithm for $\Lambda(\mathbf{G}^T)$, to get the inversion $(\mathbf{z}, \mathbf{e})$ for $\mathbf{b}$.

  If $\|\mathbf{e}\| \geq (\ell + \sqrt{n\ell_q}) \cdot \alpha q \cdot \mathcal{O}(\sqrt{n\ell_q}) \cdot \tilde{\omega}(\log n) + (\ell + \sqrt{n\ell_q}) \cdot \mathcal{O}(n\ell_q) \cdot \omega(\sqrt{\log n})$, output $\perp$. Compute $\mathbf{u} := (\mathbf{T}_{\mathbf{A}_\mathbf{t}}^T \| \mathbf{I}_{n\ell_q}) \cdot (\mathbf{c} \mod 2q) - \mathbf{e}$, and output $f^{-1}(\mathbf{u})$.

Lemma 9 (Correctness). *Let* $q = \omega((\ell + \sqrt{n\ell_q}) \cdot n\ell_q \cdot \log n)$ *and* $\alpha < ((\ell + \sqrt{n\ell_q}) \cdot \sqrt{n\ell_q} \cdot \omega(\log n))^{-1}$. *Then* TBE2.Dec *works with overwhelming probability.*

The above lemma can be proved similarly as that for Lemma 8.

Theorem 2 (Security). *The above scheme* TBE2 *is IND-aTag-wCCA2 secure in the standard model if the* $\mathsf{DLWE}_{n,q,\bar{\Psi}_{\alpha'}}$ *assumption holds for* $\alpha' = \alpha/3 \geq 2\sqrt{n}/q$.

Proof. The proof is identical to that for Theorem 1 except that the difficulty of distinguishing from the latter two games is based on a particular form of discretized DLWE assumption: It is infeasible to distinguish the following two distributions for any uniform $\mathbf{s} \in \mathbb{Z}_q^n$, $U(\mathbb{Z}_q^n \times \mathbb{T}) := \{(\mathbf{a}, b)\}_{\mathbf{a} \xleftarrow{\$} \mathbb{Z}_q^n,\, b \xleftarrow{\$} \mathbb{T}}$ and $A_{\mathbf{s},\alpha'} := \{(\mathbf{a}, b := \langle \mathbf{a}, \mathbf{s} \rangle / q + e \mod 1)\}_{\mathbf{a} \xleftarrow{\$} \mathbb{Z}_q^n,\, e \leftarrow \bar{\Psi}'_\alpha}$, which can be transformed into the distributions over $\mathbb{Z}_q^n \times \mathbb{Z}_{2q}$ by the mapping $b \mapsto 2qb + D_{\mathbb{Z}-2qb, \sqrt{(\alpha q)^2 - (2\alpha' q)^2}}$ by Theorem 6.3 in [MP12]. □

Acknowledgment. This work is supported by National Natural Science Foundation of China (No. 61402471, 61472414, 61602061, 61772514), and IIE's Cryptography Research Project.

References

[ABB10] Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (H)IBE in the standard model. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 553–572. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_28

[Ajt99] Ajtai, M.: Generating hard instances of the short basis problem. In: Wiedermann, J., van Emde Boas, P., Nielsen, M. (eds.) ICALP 1999. LNCS, vol. 1644, pp. 1–9. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48523-6_1

[AP11] Alwen, J., Peikert, C.: Generating shorter bases for hard random lattices. Theory Comput. Syst. **48**(3), 535–553 (2011)

[Boy10] Boyen, X.: Lattice mixing and vanishing trapdoors: a framework for fully secure short signatures and more. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010. LNCS, vol. 6056, pp. 499–517. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13013-7_29

[GPV08] Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: STOC 2008, pp. 197–206. ACM (2008)

[Kil06] Kiltz, E.: Chosen-ciphertext security from tag-based encryption. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 581–600. Springer, Heidelberg (2006). https://doi.org/10.1007/11681878_30

[MG02] Micciancio, D., Goldwasser, S.: Complexity of Lattice Problems, vol. 671, p. x,220. Springer, New York (2002). https://doi.org/10.1007/978-1-4615-0897-7

[MP12] 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

[MR07] Micciancio, D., Regev, O.: Worst-case to average-case reductions based on gaussian measures. SIAM J. Comput. **37**(1), 267–302 (2007)

[MRY04] MacKenzie, P., Reiter, M.K., Yang, K.: Alternatives to non-malleability: definitions, constructions, and applications. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 171–190. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24638-1_10

[Pei09] Peikert. C.: Public-key cryptosystems from the worst-case shortest vector problem. In: STOC 2009, pp. 333–342. ACM (2009)

[Reg05] Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: STOC 2005, pp. 84–93 (2005)

[Sho01] Shoup, V.: A proposal for the ISO standard for public-key encryption (version 2.1). IACR E (2001)

[SLLF15] Sun, X., Li, B., Lu, X., Fang, F.: CCA secure public key encryption scheme based on LWE without gaussian sampling. In: Lin, D., Wang, X.F., Yung, M. (eds.) Inscrypt 2015. LNCS, vol. 9589, pp. 361–378. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-38898-4_21

[Wat05] Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005). https://doi.org/10.1007/11426639_7

[Yam17] Yamada, S.: Asymptotically compact adaptively secure lattice IBEs and verifiable random functions via generalized partitioning techniques. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017. LNCS, vol. 10403, pp. 161–193. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63697-9_6

[ZCZ16] Zhang, J., Chen, Y., Zhang, Z.: Programmable hash functions from lattices: short signatures and IBEs with small key sizes. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9816, pp. 303–332. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53015-3_11