Compact-LWE: Enabling Practically Lightweight Public Key Encryption for Leveled IoT Device Authentication

Dongxi Liu, Nan Li, Jongkil Kim, and Surya Nepal

CSIRO, Australia
{dongxi.liu,nan.li,jongkil.kim,surya.nepal}@csiro.au

Abstract. Leveled authentication allows resource-constrained IoT devices to be authenticated at different strength levels according to the particular types of communication. To achieve efficient leveled authentication, we propose a lightweight public key encryption scheme that can produce very short ciphertexts without sacrificing its security.

The security of our scheme is based on the Learning With Secretly Scaled Errors in Dense Lattice (referred to as Compact-LWE) problem. We prove the hardness of Compact-LWE by reducing Learning With Errors (LWE) to Compact-LWE. However, unlike LWE, even if the closest vector problem (CVP) in lattices can be solved, Compact-LWE is still hard, due to the high density of lattices constructed from Compact-LWE samples and the relatively longer error vectors. By using a lattice-based attack tool, we verify that the attacks, which are successful on LWE instantly, cannot succeed on Compact-LWE, even for a small dimension parameter like $n = 13$, hence allowing small dimensions for short ciphertexts.

On the Contiki operating system for IoT, we have implemented our scheme, with which a leveled Needham-Schroeder-Lowe public key authentication protocol is implemented. On a small IoT device with 8MHZ MSP430 16-bit processor and 10KB RAM, our experiment shows that our scheme can complete 50 encryptions and 500 decryptions per second at a security level above 128 bits, with a public key of 2368 bits, generating 176-bit ciphertexts for 16-bit messages. With two small IoT devices communicating over IEEE 802.15.4 and 6LoWPAN, the total time of completing an authentication varies from 640ms (the 1st authentication level) to 8373ms (the 16th authentication level), in which the execution of our encryption scheme takes only a very small faction from 46ms to 445ms.

1 Introduction

In the Internet of Things (IoT), data is usually collected by resource-constrained devices in a variety of environments, and transmitted to powerful platforms to store and process. For sensitive IoT applications, it is desirable that the IoT devices are authenticated to ensure the trust to the data source. On the other hand, the IoT devices should also have a strong method to authenticate the server or other communicating devices, such that the IoT devices cannot be exploited in IoT botnets.

For resource-constrained IoT devices, leveled authentication can be beneficial, since they thus can authenticate each other at different levels, depending on the types of communication. For example, a strong mutual authentication should be given to devices that will exchange messages to update firmware or change configuration, while the requests to sensor readings may just need a relatively lower level of authentication. Leveled authentication allows IoT devices to dynamically allocate scarce resources among different authentication tasks.

Due to constrained resources (i.e., the size of memory, CPU speed, and network bandwidth), small IoT devices are usually authenticated by weak user names and passwords in practice. The shared key authentication protocols might be developed by using lightweight block ciphers, such as PRESENT [8] and and LED [21], with a pool of preshared keys configured in IoT devices before they are deployed, as proposed in [18, 16, 32]. However, the shared key authentication protocols might not be strong enough, because IoT devices deployed in open environments can be physically compromised.

Ideally, the authentication on IoT devices are still established with public key encryption schemes. However, the widely-used public key cryptosystems, such as RSA and ECDSA, are not efficient enough for IoT devices [22]. The new lattice-based cryptographic schemes, which is believed to be secure against attacks using quantum computers, may have acceptable computational efficiency for low-end processors [13, 41]. However, the current lattice-based schemes are still not efficient for IoT devices particularly from the perspective of the sizes of ciperhtexts.

Since IoT devices may not have big RAM to process messages and high wireless communication bandwidth to transfer messages, the sizes of ciperhtexts (and also the sizes of public keys) are critical for IoT applications. Lattice-based schemes usually have much larger ciphertexts than the traditional schemes (e.g., RSA and ECC). For example, the Ring-LWE based lightweight public encryption scheme produces ciphertetxs of $3,584$ bits for only 94-bit security [13], the BLISS signatures are 5K bits for 128-bit security [17], the authenticated key exchange protocol [44] has the sizes of 5.625K bytes for a challenge message and 5.75K bytes for a response message at 80-bit security level, and in the Frodo protocol, the total size of key exchange messages can be 14.22K bytes for 128-bit security [9].

A mutual authentication protocol usually needs two authenticating parties to exchange a fixed number of challenge and response messages, which contain encrypted or signed random numbers or nonces, depending on whether an encryption scheme or a signature scheme is used. To achieve leveled authentication, it is not desirable to configure a number of private keys with various lengths on a small IoT device, due to the storage cost. Instead, the IoT device should be able to use one private key to achieve any level of authentication by exchanging a configurable number of challenge and response messages. The current lattice-based cryptographic schemes cannot efficiently support leveled authentication, since only one encrypted or signed message is already very big for small IoT devices.

In this paper, we propose a lightweight public key encryption scheme that is suitable for leveled authentication of small IoT devices. Our scheme allows more flexible configuration on the size of ciphertexts. At a required security level, the length of ciphertexts in our scheme can be adaptable to the size of a message space. For example, our scheme at a security level above 128 bits can be configured to generate 176-bit ciphertexts for a 16-bit message space or 120-bit ciphertexts for a 8-bit message space.

With our scheme, a device can be configured to encrypt challenge and response messages with short nonces, hence generating correspondingly short ciphertexts. Then, the device can achieve leveled authentication by exchanging a specified number of challenge-response messages. At the lowest authentication level, only one pair of nonces needs to be encrypted and exchanged. Since a ciphertext is short, it can be efficiently processed and transmitted in the IoT low bandwidth network.

1.1 Overview of Compact-LWE

The security of our scheme is based on the hardness of the Learning With Secretly Scaled Errors in Dense Lattice (referred to as Compact-LWE) problem. Its hardness is proven by reducing the Learning With Errors (LWE) problem [42] to Compact-LWE.

Let $\mathbf{s}$ and $\mathbf{a}_i$ be n-dimensional vectors drawn uniformly from $\mathbb{Z}_q^n$, and the error terms $e_i \in \mathbb{Z}_q$ be sampled from a discrete Gaussian distribution. The LWE problem involves a set of samples

$$(\mathbf{a}_i, \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i \bmod q),$$

where $\langle \mathbf{a}_i, \mathbf{s} \rangle$ denotes the inner product of $\mathbf{a}_i$ and $\mathbf{s}$.

The Compact-LWE problem also involves a set of samples, but defined differently as

$$(\mathbf{a}_i, \langle \mathbf{a}_i, \mathbf{s} \rangle + k * e_i \bmod q),$$

where $k \in \mathbb{Z}_q$ is a secret value coprime with q, e_i is uniformly sampled from $\mathbb{Z}_r$ for a secret value r, and $\mathbf{a}_i$ is sampled from $\mathbb{Z}_b^n$ with $b < r$. A public key in our scheme consists of a set of Compact-LWE samples, and its private key includes $\mathbf{s}$ and k.

As introduced below, Compact-LWE is resistant to well-known lattice-based attacks to LWE, hence permitting a very small dimension parameter n for short ciphertexts.

1.2 Resistance to Lattice-Based Attacks

Suppose there are either m LWE samples or m Compact-LWE samples. Let $\mathbf{A}$ be a $n * m$ matrix constructed by taking each of the m vectors $\mathbf{a}_i$ as a column of $\mathbf{A}$, and let $\mathbf{e}$ be a m-dimensional error vector obtained by collecting e_i as its entries.

Then, the lattice-based attacks to LWE try to find $\mathbf{s}$ from the m samples by solving the closest vector problem (CVP) in lattices [31, 33, 28]. That is, in the lattice generated from the row vectors of $\mathbf{A}$, the problem is to find a lattice point that is closest to the target $\mathbf{A}^T\mathbf{s} + \mathbf{e}$. In LWE, the vector $\mathbf{e}$ has a small Euclidean norm $\|\mathbf{e}\|$, and thus the lattice point closest to $\mathbf{A}^T\mathbf{s} + \mathbf{e}$ is $\mathbf{A}^T\mathbf{s}$, which can then be used to recover $\mathbf{s}$ by solving a system of noiseless linear equations.

In Compact-LWE, the lattice point $\mathbf{A}^T\mathbf{s}$ is no longer the closest one to $\mathbf{A}^T\mathbf{s} + k * \mathbf{e}$, because their distance is $k * \|\mathbf{e}\|$, which can be as big as q; hence, the lattice-based attacks [31, 33, 28] are not directly applicable any more. More importantly, even if k happens to be guessed correctly, the lattice point $(k^{-1}\mathbf{A}^T)\mathbf{s}$ is still not the closest one to $(k^{-1}\mathbf{A}^T)\mathbf{s} + \mathbf{e}$, because the error vector $\mathbf{e}$ can be much longer than each row vector of $\mathbf{A}$ in Compact-LWE (i.e., due to $b < r$). Note that $k^{-1}\mathbf{A}^T$ and $\mathbf{A}^T$ form the same lattice modulo q since k is coprime with q.

In other words, since $b < r$, the lattice generated from the row vectors of $\mathbf{A}$ in Compact-LWE is more dense in the sense that a large number of lattice points could be within a distance less than $\|\mathbf{e}\|$ centered around the target $(k^{-1}\mathbf{A}^T)\mathbf{s} + \mathbf{e}$. Thus, Compact-lWE is still hard even if CVP, which itself is a hard problem in lattices [23], can be efficiently solved; this feature makes a big deviation of Compact-LWE from LWE. If CVP could be efficiently solved, LWE is no longer hard, and all LWE-based public key encryption schemes or key exchange schemes [42, 31, 9] are thus not secure any more. As discussed later, Compact-LWE can be regarded as a new hard problem, lying in between Learning Parity with Noise (LPN) and LWE and taking the advantages of both LPN (resistance to lattice-nbased attacks) and LWE (big moduli).

The following simple example illustrates the hardness of Compact-LWE by showing the lattice point $(k^{-1}\mathbf{A}^T)\mathbf{s}$ is not the closest one to the target $(k^{-1}\mathbf{A}^T)\mathbf{s} + \mathbf{e}$ in Compact-LWE.

Let $q = 8$, $k = 3$, $\mathbf{A} = \begin{bmatrix} 0 & 3 \\ 2 & 1 \end{bmatrix}$, $\mathbf{s} = \begin{bmatrix} 7 \\ 2 \end{bmatrix}$, and $\mathbf{e} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$. Then, we have $k^{-1} = 3$, $(k^{-1}\mathbf{A}^T)\mathbf{s} = \begin{bmatrix} 4 \\ 5 \end{bmatrix} \mod 8$, and $(k^{-1}\mathbf{A}^T)\mathbf{s} + \mathbf{e} = \begin{bmatrix} 7 \\ 7 \end{bmatrix} \mod 8$, as shown in Fig. 1. Moreover, if the error vector becomes $\mathbf{e} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$, then the target point becomes $\begin{bmatrix} 6 \\ 0 \end{bmatrix}$, which itself is also a valid lattice point. In that case, the lattice-based attacks do not make any sense to Compact-LWE, since the bounded-distance decoding algorithm used in such attacks simply returns the target point as its solution.

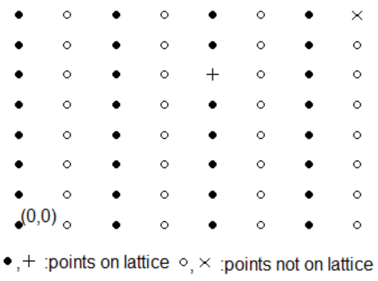

Fig. 1. Lattice with the basis $\{(0, 3), (2, 1)\}$ or the basis $\{(0 * 3, 3 * 3), (2 * 3, 1 * 3)\}$ modulo 8; $+$ and $\times$ are not the closest to each other.

For practical applications of LWE-based encryption schemes, the values of LWE parameters, in particular n, need to be determined by considering the current lattice-based attacks[1] [31, 33, 28]. For example, $\mathbf{s}$ can be recovered with the lattice-based attacks in 2.7 hours, when $n = 100$, $q = 4093$, and the standard deviation of the discrete Gaussian distribution is 4 [28].

With the lattice-based attack tool provided in [28], we confirm the attack-resistance effectiveness of Compact-LWE. In our experiment, by assuming that k is correctly

[1] The algebraic attacks [4] and combinatorial attacks [2, 27] are not effective when the number of LWE samples is limited [36].

guessed, the lattice-based attacks cannot succeed on Compact-LWE for a small dimension parameter like $n = 13$, for which LWE is attacked successfully in seconds.

1.3 Leveled Authentication Protocol and Performance Evaluation

We use the Needham-Schroeder-Lowe (NSL) public key authentication protocol [37, 34] and our lightweight encryption scheme to implement leveled authentication for IoT devices. Let A and B be two devices. The NSL authentication protocol works by requiring A and B to exchange two nonces encrypted with each other's public key. If A and B really own the corresponding private key to decrypt the encrypted nonces, then they can successfully authenticate each other. For leveled authentication, we revise the NSL protocol, such that the number of nonces exchanged depends on the expected authentication level.

We have implemented our lightweight public encryption scheme on the Contiki platform and evaluated on the resource-constrained MTM-CM5000-MSP[2] wireless sensor node, which is compliant with TelosB/Tmote Sky specification. This device uses Texas Instruments (TI) MSP430 16-bit processor running at 8MHz, with 48KB ROM and 10KB RAM. This device supports the IEEE 802.15.4 protocol for wireless communication, which provides 250Kbps physical layer transfer rate.

As an overview of the performance of our scheme, the device performs about 50 encryptions and 500 decryptions per second for plaintexts of 16 bits, generating ciphertexts of 176 bits, at a security level above 128 bits, with the public key of 2368 bits. This performance is higher than the performance of the lightweight scheme proposed in [13], where 33 encryptions and 79.4 decryptions performed per second on a 32MHz ARM Cortex-M0 processor for 84-bit security. In addition, unlike [13] and other lattice-based schemes, our scheme does not incur any possibility of decryption failures.

On two MTM-CM5000-MSP devices, the leveled NSL authentication protocol is evaluated. In the evaluation, we let a nonce be an 1-byte random integer and evaluate 16 levels of authentication. At the 1st authentication level, only one nonce is generated respectively by each device, and thus a device can be cheated online with the probability $\frac{1}{2^8}$, which is reduced to $\frac{1}{2^{128}}$ at the 16th authentication level, where each device generates 16 nonces. Accordingly, the device which initiates an authentication takes 640ms to complete the authentication at the first level, with three 176-bit ciphertexts exchanged, while taking 8273ms for the 16th authentication level by exchanging thirty-three 176-bit ciphertexts. In the total authentication time, our encryption scheme only takes a small fraction from 46ms to 445ms.

Moreover, a higher level of authentication can be achieved gradually and adjusted dynamically. For example, two IoT devices have authenticated at the 16th authentication level, and later on they can increase the level by exchanging several extra nonces for protecting a more sensitive communication.

1.4 Contributions and Paper Organization

The contributions of this paper are summarized as follows.

[2] http://www.advanticsys.com/shop/mtmcm5000msp-p-14.html

- We propose the hard problem Compact-LWE, based on which a lightweight public key encryption scheme is constructed. Unlike other LWE schemes [42, 31, 9], even if the closest vector problem in lattices can be efficiently solved, our scheme is still secure. This security feature permits very small dimension parameters for Compact-LWE, thus leading to much shorter ciphertexts.
- We prove the hardness of Compact-LWE by giving the reduction from LWE to Compact-LWE. In addition to the formal proof, we analyze the resistance of Compact-LWE against various attacks. A lattice-based attack tool has been used to verify the security of our scheme with concrete parameters, which can be taken directly in practical applications.
- We implemented our encryption scheme on the Contiki platform and the leveled authentication protocol based on the NSL public key authentication protocol. The IoT devices MTM-CM5000-MSP have been used to evaluate the performance of our scheme and authentication protocol. In the implementation, we describe how our encryption scheme can be adapted to Contiki and MTM-CM5000-MSP, so that modulo operations can be done by the underlying hardware.

The rest of this paper is organized as follows. We introduce the construction of our scheme and prove its correctness in Section 2. The security of our scheme is analyzed in Section 3, where the hardness of Compact-LWE is defined and proved. In Section 4, we analyze the resistance of Compact-lWE to three types of attacks, including the lattice-based attacks. Based on the attack analysis, the concrete security level is defined in Section 5. Our implementation and evaluation is described in Section 6. The last two sections include the discussion of related work and the summary of this paper.

2 Construction of Our Lightweight Public Key Encryption Scheme

Let $pp = (q, n, m, t, w, b)$ be a tuple of positive integers, which are the public parameters of our scheme. We require $n + 1 < m < n^2$, $n < b$, $(2\log_2 b * b + 2) * b < q$, and $2\log_2 b < n$. Let $\mathbb{Z}_q = \{0, \ldots, q - 1\}$ and other notations like $\mathbb{Z}_b$ and $\mathbb{Z}_t$ be similarly defined. Our scheme consists of three algorithms: key generation, encryption, and decryption.

2.1 Key Generation

Given the public parameter pp, the key generation algorithm generates a random key pair $(\mathbf{K}, \mathbf{PK})$, where $\mathbf{K}$ is the private key and $\mathbf{PK}$ is the public key. Let $\mathbf{Gen}(pp) = (\mathbf{K}, \mathbf{PK})$ denote the key generation algorithm.

The private key $\mathbf{K}$ is defined as $\mathbf{K} = (\mathbf{s}, sk, r, p)$, where $\mathbf{s}$ is a n-dimensional vector, uniformly sampled from $\mathbb{Z}_q^n$, and the components sk, r and p are positive integers in $\mathbb{Z}_q$. A private key needs to satisfy the following conditions.

- $t \leq p$
- sk, p and q are mutually co-prime
- $sk * (t - 1) + w * r * p < q$

- $b < r$

The above conditions are used later when proving both correctness and security of our scheme. After the private key $\mathbf{K}$ is generated, the algorithm $\mathbf{Gen}(pp)$ then generates the corresponding public key $\mathbf{PK}$. The public key $\mathbf{PK}$ consists of m random Compact-LWE samples, as defined below.

Let $\mathbf{a}_i \in \mathbb{Z}_b^n$ be a vector uniformly sampled for the ith public key sample. Then, the ith public key sample is the pair

$$(\mathbf{a}_i, pk_i = \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i * sk_q^{-1} * p \bmod q),$$

where e_i is uniformly sampled from $\mathbb{Z}_r$ and sk_q^{-1} satisfies $sk * sk_q^{-1} = -1 \bmod q$. Recall that we used k to refer to $sk_q^{-1} * p$ in the previous introduction for brevity.

To reduce the size of public keys, all IoT devices (e.g., manufactured by the same company) can share the common $\mathbf{a}_i$ $(1 \leq i \leq m)$. As such, an IoT device only needs to publish pk_i in its public key $\mathbf{PK}$. This idea of reducing the size of a public key is described in other LWE-based public key encryption schemes [42, 31].

2.2 Encryption

A plaintext value v comes from $\mathbb{Z}_t$. It is encrypted into a ciphertext $\mathbf{c}$ with the public key $\mathbf{PK}$, denoted $\mathbf{c} = \mathbf{Enc}(\mathbf{PK}, v)$. The ciphertext $\mathbf{c}$ is a $(n+1)$-dimensional vector. In the encryption algorithm, w public key samples are randomly selected and added. Given two samples $(\mathbf{a}_i, pk_i)$ and $(\mathbf{a}_j, pk_j)$, their addition is represented as $(\mathbf{a}_i, pk_i) + (\mathbf{a}_j, pk_j) = (\mathbf{a}_i + \mathbf{a}_j, pk_i + pk_j) \bmod q$, where $\mathbf{a}_i + \mathbf{a}_j$ is a vector addition. The encryption algorithm $\mathbf{c} = \mathbf{Enc}(\mathbf{PK}, v)$ works as follows.

- Sample an integer i uniformly from the set $\{1, \ldots, m\}$, and let $\mathbf{c}' = (\mathbf{a}_i, pk_i)$ be the corresponding sample in $\mathbf{PK}$.
- Sample $w - 1$ integers uniformly from the set $\{1, \ldots, m\}$, and for each sampled integer i, do the update $\mathbf{c}' = \mathbf{c}' + (\mathbf{a}_i, pk_i)$.
- Suppose $\mathbf{c}' = (\mathbf{a}, pk)$, and generate $\mathbf{c} = (\mathbf{a}, v - pk \bmod q)$.

Note that from the set $\{1, \ldots, m\}$, a public key sample might be selected repeatedly, at most w times. The above encryption algorithm is similar to the encryption algorithm proposed in [42], where the pubic key is a set of LWE samples and the encryption algorithm adds a random subset of public key samples (i.e., a public key sample is selected only once) to generate the ciphertext for a message, which can only be 0 or 1. However, our decryption algorithm is totally different from the one in [42].

2.3 Decryption

Let $\mathbf{K} = (\mathbf{s}, sk, r, p)$ and sk_p^{-1} be the multiplicative inverse of sk modulo p (i.e., $sk_p^{-1} * sk = 1 \bmod p$). Given the ciphertext $\mathbf{c} = (\mathbf{a}, d)$, the decryption algorithm $\mathbf{Dec}(\mathbf{K}, \mathbf{c}) = v$ recovers the plaintext value v with the following steps.

- Calculate $c' = \langle \mathbf{a}, \mathbf{s} \rangle + d \bmod q$.

- Calculate $skv = sk * c' \bmod q$.
- Calculate $v = sk_p^{-1} * skv \bmod p$.

Our decryption algorithm is deterministic, and there is no any decryption failure for all ciphertexts generated from the encryption algorithm. On the contrary, the existing LWE-based encryption schemes like [13, 42] may incur decryption failures.

2.4 Correctness

The correctness of our scheme is stated in the theorem below.

Theorem 1. *Let $pp = (q, n, m, t, w, b)$ be the public parameters and $\mathbf{Gen}(pp) = (\mathbf{K}, \mathbf{PK})$. Then, for any $v \in \mathbb{Z}_t$, we have*

$$\mathbf{Dec}(\mathbf{K}, \mathbf{Enc}(\mathbf{PK}, v)) = v.$$

Proof. At the first two steps of the encryption algorithm, w public key samples are randomly selected and added. Let $l[1]$,..., $l[w]$ be the indexes of the selected public key samples. Then, based on the key generation algorithm $\mathbf{Gen}(pp)$, the selected samples satisfy the following w equations.

$$pk_{l[1]} = \langle \mathbf{a}_{l[1]}, \mathbf{s} \rangle + e_{l[1]} * sk_q^{-1} * p \bmod q$$
$$\cdots$$
$$pk_{l[w]} = \langle \mathbf{a}_{l[w]}, \mathbf{s} \rangle + e_{l[w]} * sk_q^{-1} * p \bmod q$$

By adding the two sides of the above equations, we get

$$\sum_{i=1}^{w} pk_{l[i]} = \langle \sum_{i=1}^{w} \mathbf{a}_{l[i]}, \mathbf{s} \rangle + \sum_{i=1}^{w} e_{l[i]} * sk_q^{-1} * p \bmod q$$

That is, $\mathbf{c}' = (\sum_{i=1}^{w} \mathbf{a}_{l[i]}, \sum_{i=1}^{w} pk_{l[i]})$ after the second step of encryption. The ciphertext of v generated at the third step of encryption is $(\sum_{i=1}^{w} \mathbf{a}_{l[i]}, d = v - \sum_{i=1}^{w} pk_{l[i]} \bmod q)$.

Given the above ciphertext of v, with the secret vector $\mathbf{s}$ in the private key $\mathbf{K}$, the first step of decryption is written as

$$\mathbf{c}' = \langle \sum_{i=1}^{w} \mathbf{a}_{l[i]}, \mathbf{s} \rangle + d = \langle \sum_{i=1}^{w} \mathbf{a}_{l[i]}, \mathbf{s} \rangle + (v - \sum_{i=1}^{w} pk_{l[i]})$$

That is, $\mathbf{c}' = v - \sum_{i=1}^{w} e_{l[i]} * sk_q^{-1} * p$ after the first decryption step. Since $sk * sk_q^{-1} = -1 \pmod{q}$, the second step of decryption generates

$$skv = sk * (v - \sum_{i=1}^{w} e_{l[i]} * sk_q^{-1} * p) = sk * v + \sum_{i=1}^{w} e_{l[i]} * p \bmod q.$$

Since $v \in \mathbb{Z}_t$, $e_{l[i]} < r$ in each public key sample, and $sk * (t-1) + w * r * p < q$, we have

$$0 \leq sk * v + \sum_{i=1}^{w} e_{l[i]} * p < q,$$

and thus

$$skv = sk * v + \sum_{i=1}^{w} e_{l[i]} * p \pmod q = sk * v + \sum_{i=1}^{w} e_{l[i]} * p.$$

At last, since $sk * sk_p^{-1} = 1 \texttt{ mod } p$, the last operation $(sk_p^{-1} * sk * v + \sum_{i=1}^{w} e_{l[i]} * sk_p^{-1} * p) \texttt{ mod } p$ in the decryption algorithm returns exactly v.

3 Security Analysis

In this section, we prove the hardness of Compact-LWE by giving reduction from LWE to Compact-LWE. Thus, if an adversary can solve the Compact-LWE problem efficiently, then it can solve the LWE problem efficiently, too. Then, we prove that our lightweight public key encryption scheme is semantically secure [20], meaning that the adversary cannot get any information about a message from its ciphertext and the public key.

3.1 The Hardness of LWE

Let n and q be positive integers, $0 < \alpha < 1$, and let $\mathcal{X}_{\alpha q}$ be a discrete Gaussian distribution with the standard deviation αq. As defined in [39], the LWE distribution, from which LWE samples are drawn, is given below.

Definition 1. *For* $\mathbf{s}$ *uniformly sampled from* $\mathbb{Z}_q^n$*, the LWE distribution* $A_{n,\mathbf{s},q,\alpha}$ *over* $\mathbb{Z}_q^n \times \mathbb{Z}_q$ *is obtained by sampling* $\mathbf{a}$ *from* $\mathbb{Z}_q^n$ *uniformly,* e *from* $\mathcal{X}_{\alpha q}$*, outputting* $(\mathbf{a}, y = \langle \mathbf{a}, \mathbf{s} \rangle + e \texttt{ mod } q)$.

There are two hardness problems related to LWE: the search LWE problem and the decision LWE problem.

Definition 2. *Given a set of independent samples* $(\mathbf{a}_i, y_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ *drawn from* $A_{n,\mathbf{s},q,\alpha}$*, the search LWE problem* $\texttt{Search-LWE}_{n,q,\alpha}$ *is to find* $\mathbf{s}$.

The hardness of $\texttt{Search-LWE}_{n,q,\alpha}$ is proved in [42] and [38] by giving the quantum or classical reductions from the worst-case hardness of the GapSVP problem to the search LWE problem. The hardness of $\texttt{Search-LWE}_{n,q,\alpha}$ is summarized in the following theorem from [11].

Theorem 2 (Theorem 2.16 of [11]). *Let* $n, q \geq 1$ *be integers and let* $\alpha \in (0, 1)$ *be such that* $\alpha q \geq 2\sqrt{n}$*. Then there exits a quantum reduction from worst case n-dimensional* $\mathrm{GapSVP}_{\tilde{O}(n/\alpha)}$ *to* $\texttt{Search-LWE}_{n,q,\alpha}$*. If in addition* $q \geq 2^{n/2}$ *then there is also a classical reduction between these two problems.*

Definition 3. *Given a set of independent samples* $(\mathbf{a}_i, y_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$*, the decision LWE problem* $\texttt{Decision-LWE}_{n,q,\alpha}$ *is to distinguish whether a sample is drawn from* $A_{n,\mathbf{s},q,\alpha}$ *or drawn from the uniform distribution (i.e.,* y_i *is also drawn uniformly).*

The hardness of $\texttt{Decision-LWE}_{n,q,\alpha}$ is proved by showing search-to-decision reductions [42, 38]. That is, if $\texttt{Decision-LWE}_{n,q,\alpha}$ can be solved efficiently, then $\texttt{Search-LWE}_{n,q,\alpha}$ is solved efficiently, too. There are several such reductions, each of which is for a different type of modulus q. For example, the modulus is supposed to be polynomial and prime in [42], the modulus must be smooth (i.e., the product of multiple primes) [38], and it can also be a power of 2 in [11]. The theorem for the hardness of $\texttt{Decision-LWE}_{n,q,\alpha}$ from [11] will be used in our proof.

Theorem 3 (Theorem 2.17 of [11]). *Let q be a power of 2, and α satisfy $1/q < \alpha < 1/\omega(\sqrt{\log_2 n})$. Then there exists an efficient reduction from* $\texttt{Search-LWE}_{n,q,\alpha}$ *to* $\texttt{Decision-LWE}_{n,q,\alpha'}$ *for $\alpha' = \alpha\omega(\log_2 n)$.*

3.2 Hardness of Compact-LWE

We first give the definition of Compact-LWE distribution, from which the public keys of our scheme are sampled. In the definition, the number of samples m is omitted, since the hardness of Compact-LWE does not depend on this parameter, as in Theorem 2 and Theorem 3 for LWE. This parameter however is used when we analyze concrete attacks to our scheme and determine the concrete security level of our scheme later.

Definition 4. *For positive integers q, n, t, w, and b, such that $b > n$, $(2\log_2 b * b + 2) * b < q$, and $2\log_2 b < n$, let sk, r, and p be secret integers satisfying the conditions specified in the algorithm* **Gen**. *Suppose $\mathbf{s}$ is a secret vector uniformly sampled from $\mathbb{Z}_q^n$. The Compact-LWE distribution $D_{n,\mathbf{s},q,b,t,w,sk,r,p}$ over $\mathbb{Z}_b^n \times \mathbb{Z}_q$ is obtained by sampling $\mathbf{a}$ from $\mathbb{Z}_b^n$ uniformly, e from $\mathbb{Z}_r$ uniformly, outputting $(\mathbf{a}, y = \langle \mathbf{a}, \mathbf{s}\rangle + e * sk_q^{-1} * p \bmod q)$.*

Similarly, there are the search and decision problems for Compact-LWE. If the secret values $\mathbf{s}$, sk, and p are found, then all ciphertexts of our scheme can de decrypted. Hence, the search problem of Compact-LWE requires the findings of all $\mathbf{s}$, sk, and p.

Definition 5. *Given a set of independent samples $(\mathbf{a}_i, y_i) \in \mathbb{Z}_b^n \times \mathbb{Z}_q$ drawn from $D_{n,\mathbf{s},q,b,t,w,sk,r,p}$, the search Compact-LWE problem* $\texttt{Search-Compact-LWE}_{n,q,b,t,w}$ *is to find $\mathbf{s}$, sk, and p.*

The hardness of $\texttt{Search-Compact-LWE}_{n,q,b,t,w}$ is proved by giving the reduction from $\texttt{Search-LWE}_{n',b',\alpha}$ for some integer n' and b' to $\texttt{Search-Compact-LWE}_{n,q,b,t,w}$. In the proof, we reduce samples in $\texttt{Search-LWE}_{n',b',\alpha}$ with dimension n' and modulus b' to samples in $\texttt{Search-Compact-LWE}_{n,q,b,t,w}$ with dimension n, modulus q, and any values for other parameters (e.g., t and w) that are not required by $\texttt{Search-LWE}_{n',b',\alpha}$. For a vector $\mathbf{s}$ and a positive integer j, let $\mathbf{s}\|\mathbb{Z}_q^j$ be a vector obtained by extending $\mathbf{s}$ with j uniformly random values from $\mathbb{Z}_q$.

Theorem 4. *Let b' be a power of 2, $b > b'$, and $\alpha \in (0, 1)$. For an integer $n' > 1$, such that $b' \geq 2^{n'/2}$, if* $\texttt{Search-Compact-LWE}_{n,q,b,t,w}$ *can be solved efficiently by the adversary, then* $\texttt{Search-LWE}_{n',b',\alpha}$ *can be solved efficiently, too.*

Proof. Suppose $\mathbf{s}' \in \mathbb{Z}_{b'}^{n'}$ is a secret vector. Given a set of independent samples $(\mathbf{a}_i, y_i) \in \mathbb{Z}_{b'}^{n'} \times \mathbb{Z}_{b'}$ drawn from $A_{n',\mathbf{s}',b',\alpha}$ for the problem $\texttt{Search-LWE}_{n',b',\alpha}$, we have

$$(\mathbf{a}_i, y_i = \langle \mathbf{a}_i, \mathbf{s}' \rangle + e_i \texttt{ mod } b' = \langle \mathbf{a}_i, \mathbf{s}' \rangle + e_i + x_i * b'),$$

for some integer $x_i < n' * b'$.

Since the above e_i is sampled from the discrete Gaussian distribution with the mean 0 modulo b', we have $e_i \in (-b', b')$. By adding b' to y_i (i.e., $y'_i = y_i + b'$), the above sample is converted into

$$(\mathbf{a}_i, y'_i = \langle \mathbf{a}_i, \mathbf{s}' \rangle + r_i),$$

where $0 < r_i = e_i + x_i * b' + b' < (n' * b' + 2) * b'$.

Next, the adversary chooses any values for parameters t, sk, r, w, and p, ensuring the conditions specified in the algorithm **Gen** are satisfied (particularly the condition $sk * (t-1) + w * r * p < q$), and also the extra condition $(2\log_2 b' * b' + 2) * b' < r$. This extra condition is satisfiable, since we have $(2\log_2 b * b + 2) * b < q$, $b' < b$, and r can be as big as q based on the choices of t, sk, w, and p.

Let $y''_i = y'_i * sk_q^{-1} * p \texttt{ mod } q$. From the above sample, the following one can be obtained

$$(\mathbf{a}_i, y''_i = \langle \mathbf{a}_i, \mathbf{s}' \cdot (sk_q^{-1} * p) \rangle + r_i * sk_q^{-1} * p \texttt{ mod } q),$$

According to the condition $b' \geq 2^{n'/2}$, we have $n' \leq 2\log_2 b'$ and then $r_i < (n' * b' + 2) * b' \leq (2\log_2 b' * b' + 2) * b' < r$. In addition, our scheme requires $n > 2\log_2 b$, so $n > 2\log_2 b' > n'$.

Thus, when $\mathbf{a}_i$ is extended with $n - n'$ zero elements, the above sample is a valid sample in $D_{n,\mathbf{s}'',q,b',t,w,sk,r,p}$, where

$$\mathbf{s}'' = (\mathbf{s}' \cdot (sk_q^{-1} * p)) \| \mathbb{Z}_q^{n-n'} \texttt{ mod } q.$$

The distribution $D_{n,\mathbf{s}'',q,b',t,w,sk,r,p}$ is a special case of the distribution $D_{n,\mathbf{s},q,b,t,w,sk,r,p}$, since $\mathbf{s}''$ is the special case of $\mathbf{s}$, $b' < b$, and the condition $sk * (t-1) + w * r * p < q$ still holds for $D_{n,\mathbf{s},q,b,t,w,sk,r,p}$.

Hence, if $\texttt{Search-Compact-LWE}_{n,q,b,t,w}$ can be solved by the adversary efficiently, then its any special case can be solved, and the same algorithm can be used to solve $\texttt{Search-LWE}_{n',b',\alpha}$.

When $b' \geq 2^{n'/2}$, Theorem 2 shows that $\texttt{Search-LWE}_{n',b',\alpha}$ is hard. Hence, $\texttt{Search-Compact-LWE}_{n,q,b,t,w}$ cannot be solved by the adversary efficiently. Note that the above proof allows the error rate α in $\texttt{Search-LWE}_{n',b',\alpha}$ to take a value infinitely close to 1, which makes the hardest case of LWE.

The decision Compact-LWE problem is to distinguish between samples drawn either from $D_{n,\mathbf{s},q,b,t,w,sk,r,p}$ or from an uniform distribution in $\mathbb{Z}_b^n \times \mathbb{Z}_q$.

Definition 6. *Given a set of independent samples $(\mathbf{a}_i, y_i) \in \mathbb{Z}_b^n \times \mathbb{Z}_q$, the decision Compact-LWE problem* $\texttt{Decision-Compact-LWE}_{n,q,b,t,w}$ *is to distinguish whether a sample is drawn from $D_{n,\mathbf{s},q,b,t,w,sk,r,p}$ or drawn from the uniform distribution over $\mathbb{Z}_b^n \times \mathbb{Z}_q$ (i.e., $\mathbf{a}_i$ is uniformly drawn from $\mathbb{Z}_b^n$ and y_i is drawn uniformly from $\mathbb{Z}_q$).*

The hardness proof of $\texttt{Decision-Compact-LWE}_{n,q,b,t,w}$ is similar to the hardness proof of $\texttt{Search-Compact-LWE}_{n,q,b,t,w}$, by giving the reduction from $\texttt{Decision-LWE}_{n',b',\alpha}$ for some integer b' and n' to $\texttt{Decision-Compact-LWE}_{n,q,b,t,w}$.

Theorem 5. *For an integer $n' > 1$, such that $b \geq 2^{n'/2}$, $b > b'$, $\alpha \in (0, 1)$, and*

$$\omega(\log_2 n')/b' < \alpha < \omega(\log_2 n')/\omega(\sqrt{\log_2 n'}),$$

if $\texttt{Decision-Compact-LWE}_{n,q,b,t,w}$ *can be solved efficiently by the adversary, then* $\texttt{Decision-LWE}_{n',b',\alpha}$ *can be solved efficiently, too.*

Proof. When $\omega(\log_2 n')/b' < \alpha < \omega(\log_2 n')/\omega(\sqrt{\log_2 n'})$, there exists $\alpha' = \alpha/\omega(\log_2 n')$, such that $1/b' < \alpha' < 1/\omega(\sqrt{\log_2 n'})$. Under this condition and the condition that b' is a power of 2, Theorem 3 shows that $\texttt{Decision-LWE}_{n',b',\alpha}$ is hard.

Suppose $\mathbf{s}' \in \mathbb{Z}_{b'}^{n'}$ is a secret vector. Given a set of independent samples $(\mathbf{a}_i, y_i) \in \mathbb{Z}_{b'}^n \times \mathbb{Z}_{b'}$ drawn either from $A_{n',\mathbf{s}',b',\alpha}$ or from the uniform distribution over $\mathbb{Z}_{b'}^n \times \mathbb{Z}_{b'}$, the adversary cannot distinguish which distribution a sample comes from, according to the hardness of $\texttt{Decision-LWE}_{n',b',\alpha}$.

For a sample $(\mathbf{a}_i, y_i) \in \mathbb{Z}_{b'}^n \times \mathbb{Z}_{b'}$ drawn either from $A_{n',\mathbf{s}',b',\alpha}$ or from the uniform distribution over $\mathbb{Z}_{b'}^n \times \mathbb{Z}_{b'}$, the adversary can convert it into a sample in $(\mathbf{a}_i, y_i'') \in \mathbb{Z}_{b'}^n \times \mathbb{Z}_q$, which correspondingly belongs to either $D_{n,\mathbf{s}'',q,b',t,w,sk,r,p}$, where

$$\mathbf{s}'' = (\mathbf{s}' \cdot (sk_q^{-1} * p)) \| \mathbb{Z}_q^{n-n'} \;\texttt{mod}\; q,$$

or the uniform distribution over $U = \mathbb{Z}_{b'}^n \times \{(x + b') * sk_q^{-1} * p \;\texttt{mod}\; q | x \in \mathbb{Z}_{b'}\}$, in the same steps as in the proof of Theorem 4. The adversary still cannot distinguish whether the sample $(\mathbf{a}_i, y_i'')$ comes from $D_{n,\mathbf{s}'',q,b',t,w,sk,r,p}$ or the uniform distribution over U. Note that U is a subset of $\mathbb{Z}_b^n \times \mathbb{Z}_q$, since $b' < b$.

The adversary cannot have an efficient algorithm to solve the problem $\texttt{Decision-Compact-LWE}_{n,q,b,t,w}$; otherwise, the algorithm can distinguish whether any sample in the problem $\texttt{Decision-Compact-LWE}_{n,q,b,t,w}$ comes from $D_{n,\mathbf{s},q,b,t,w,sk,r,p}$ or from the uniform distribution over $\mathbb{Z}_b^n \times \mathbb{Z}_q$, and as a special case the adversary can use this algorithm to distinguish whether the sample $(\mathbf{a}_i, y_i'') \in \mathbb{Z}_{b'}^n \times \mathbb{Z}_q$ comes from $D_{n,\mathbf{s}'',q,b',t,w,sk,r,p}$ or from the uniform distribution over U, leading to an efficient solution to $\texttt{Decision-LWE}_{n',b',\alpha}$.

Since $\texttt{Decision-LWE}_{n',b',\alpha}$ is hard, the problem $\texttt{Decision-Compact-LWE}_{n,q,b,t,w}$ is hard, too.

3.3 Semantic Security

Based on the hardness of $\texttt{Decision-Compact-LWE}_{n,q,b,t,w}$, we prove the semantic security of our scheme, which is also called the IND-CPA security.

Given the public parameters $pp = (q, n, m, t, w, b)$, the attack game $\mathcal{G}(pp)$ shown below will be used to define the semantic security of our scheme.

1. The challenger runs the key generation algorithm $\mathbf{Gen}(pp) = (\mathbf{K}, \mathbf{PK})$ and gives the public key $\mathbf{PK}$ to the adversary, together with m samples, denoted by $\texttt{U}$, drawn uniformly from $\mathbb{Z}_b^n \times \mathbb{Z}_q$.

2. The adversary submits a message $v \in \mathbf{Z}_t$ to the challenger.
3. The challenger generates two ciphertexts $\mathbf{c}_0 = \mathbf{Enc}(\mathbf{PK}, v)$ and $\mathbf{c}_1 = \mathbf{Enc}(\mathbf{U}, 0)$. That is, $\mathbf{U}$ is used by the challenger as if it is a public key. The challenger selects a random bit $i \in \{0, 1\}$ uniformly and sends back the pair $(\mathbf{c}_i, \mathbf{c}_{1-i})$.
4. The adversary receives the pair of ciphertexts, and outputs a guess i', meaning that $\mathbf{c}_{i'}$ encrypts v with the proper public key $\mathbf{PK}$.
5. The adversary wins the game if $i' = i$.

Intuitively, this game means that if the adversary cannot distinguish a ciphertext from a random number, then the scheme does not leak much information of messages in cihertexts. Note that our game is defined differently from the usual game for IND-CPA security [26], in which the adversary selects two values and the challenger randomly encrypts one of them; however, our game and the usual game are equivalent.

Theorem 6. *Under the assumption that the decision problem* Decision-Compact-LWE$_{n,q,b,t,w}$ *is hard, the adversary cannot win the game* $\mathcal{G}(pp)$ *with a probability non-negligibly higher than* $1/2$.

Proof. Let the matrix $\mathbf{A} \in \mathbf{Z}_b^{n*m}$ be formed as in lattice-based attacks from $\mathbf{PK}$ or U. Let $\mathbf{l} = (l[1], ..., l[m])$ be a column vector, where $l[i] \geq 0$ denotes the times of the ith sample selected in an encryption from $\mathbf{PK}$ or U. Let $\mathbf{c}'$ be the vector generated at the end of the second step of the encryption algorithm. Then, we have $l[1] + \cdots + l[m] = w$, w is a small value, and $0 \leq l[i] \leq w$, satisfying $\mathbf{Al} = \mathbf{c}'$. Due to the hardness of Short Integer Solution (SIS) [36], the vector $\mathbf{l}$ cannot be efficiently recovered from $\mathbf{A}$ and $\mathbf{c}'$. That is, given $\mathbf{c}'$, the adversary cannot efficiently check whether $\mathbf{c}'$ is generated from $\mathbf{PK}$ or U by only trying to find the vector $\mathbf{l}$ with $\mathbf{PK}$ and $\mathbf{c}'$ or with U and $\mathbf{c}'$. Furthermore, the hardness of Decision-Compact-LWE$_{n,q,b,t,w}$ ensures that the adversary cannot efficiently distinguish whether a single sample is drawn from $\mathbf{PK}$ or U based on their distributions.

Then, given $\mathbf{c}'_0$ and $\mathbf{c}'_1$ generated at the second step of the encryption algorithm with $\mathbf{PK}$ and U, respectively, but in a random order, the adversary cannot have an efficient algorithm to determine which is from $\mathbf{PK}$ and which is from U with a probability non-negligibly higher $1/2$; otherwise, the adversary can use the same algorithm to solve Decision-Compact-LWE$_{n,q,b,t,w}$ or to solve the SIS problem. Since the adversary cannot distinguish $\mathbf{c}'_0 = (\mathbf{a}_0, pk_0)$ and $\mathbf{c}'_1 = (\mathbf{a}_1, pk_1)$, then the adversary cannot distinguish between $(v - pk_0) \bmod q$ and $(0 - pk_1) \bmod q$, or between $(0 - pk_0) \bmod q$ and $(v - pk_1) \bmod q$, which comprise the last elements of the ciphertexts $\mathbf{c}_0$ and $\mathbf{c}_1$. Thus, every element in $\mathbf{c}_0$ and $\mathbf{c}_1$ cannot be distinguished efficiently with a probability non-negligibly higher $1/2$.

4 Resistance to Attacks

We have proved that our scheme is secure in terms of the way it is constructed. To determine the values of parameters (e.g., the value of n) that make our scheme concretely secure, we need to consider the possible attacks to our scheme.

In this section, we analyze the attacks through pubic keys and the attacks to encrypted messages. The former attacks aim to recover private keys from the corresponding public keys, while the latter attacks try to recover the plaintext values from the encrypted messages without knowing the private keys.

4.1 Attacks through Public Keys

The possible attacks to the search LWE includes algebraic attacks [4], combinatorial attacks [2, 27], and lattice-based attacks [31, 33, 28]. All these attacks are not applicable to our Compact-LWE based scheme.

4.1.1 Algebraic Attacks The algebraic attacks consider all possible error values in each LWE sample. When the errors in LWE samples are too small, the algebraic attacks [4] can lead to the subexponential algorithm to recover private keys. There are at least n^2 samples needed for such attacks for binary errors. Our scheme is not vulnerable to such attacks, since the errors e_i in our public key are not binary and the number of our public key samples is less than n^2.

In addition, if the number of samples is limited, the algebraic attacks cannot apply to LWE [36] and this result is also applicable to Compact-LWE.

4.1.2 Combinatorial Attacks The combinatorial attacks [2, 27] work by assuming that there is an LWE oracle that can be used by the adversary to generate any required number of LWE samples. If two samples have collisions on a block of elements of $\mathbf{a}$, then one sample can be subtracted by the other to get a new LWE sample, which have the collided elements eliminated. This procedure eliminates a block of unknowns, without expanding too much the error value in the new sample. When there are a small number of elements in $\mathbf{s}$ left, the adversary guesses their possible values. If the guess is correct, then the recovered errors value in the new sample should be small, following the discrete Gaussian distribution.

This attack does not apply to our scheme for the following reasons. First, the number of our public key samples is limited and the adversary cannot generate new independent samples by its own. Second, the errors in our public key are secretly scaled and can be evenly distributed over $\mathbf{Z}_q$, as to be shown in our experiment later. Third, our scheme allows a relatively big modulus q (e.g., $q = 2^{32}$ in our evaluation, bigger than $q = 4093$ or $q = 16381$ used in [31, 28]) and a big q makes the guess of even a small number of secret values less efficient. Moreover, our requirement $b > n$ ensures that b cannot be too small, so as to reduce the collision probability of $\mathbf{a}_i \in \mathbb{Z}_b^n$ among $m < n^2$ public key samples.

4.1.3 Lattice-based Attacks Given a small number of LWE samples, the lattice-based attacks [31, 33, 28] are effective and practical. For example, when $n = 100$, $q = 4093$, and $\alpha = 4/4093$, the open-source tool developed in [28] takes 2.7 hours to solve the search LWE problem $\texttt{Search-LWE}_{n,q,\alpha}$. Due to such attacks, the current LWE-based encryption schemes have to choose big dimension parameters. However, as discussed in Section 1.2 and illustrated in Fig. 1, the lattice-based attacks are not applicable to our Compact-LWE based scheme, even if the hard problem CVP in lattices can be efficiently solved.

LWE is an extension of the well-known Learning Parity with Noise (LPN) problem by extending the modulus q from 2 to bigger integers [42]. Without considering scaled errors, our Compact-LWE can be seen as a problem sitting in between LWE and LPN, in the sense that the secret vector $\mathbf{s}$ in Compact-LWE is sampled from $\mathbf{Z}_q$ (the same as LWE), while the public vector $\mathbf{a}_i$ is bounded by $b < r$ and r is much smaller than q (close to LPN).

LPN is known only vulnerable to the combinatorial attacks [7, 2, 27], not to lattice-based attacks. Like LPN, Compact-LWE is not vulnerable to lattice-based attacks; by limiting the number of samples, Compact-LWE is resistant to the combinatorial attacks, the same as LWE [36]. Thus, Compact-LWE can be regarded as a new hard problem, lying in between LPN and LWE and taking the advantages of both LPN and LWE.

4.2 Attacks to Ciphertexts

In this type of attacks, we consider whether a message can be recovered efficiently from its ciphertext and the public key, without knowing the corresponding private key.

Let $l[i] \geq 0$ denote the times of the ith public key sample being selected in an encryption. Then, we have the equation

$$l[1] + \cdots + l[m] = w,$$

since w public key samples are randomly selected in one encryption. We also have $l[i] \leq w$, since one sample can be selected at most w times. Let $\mathbf{c} = (c_1, \ldots, c_{n+1})$ be a ciphertext and v be the unknown plaintext message. Let $\mathbf{a}_i = (a_{i1}, ..., a_{in})$. Then, $\mathbf{c}$ is defined by the following $n + 1$ equations.

$$l[1] * a_{11} + \cdots + l[m] * a_{m1} = c_1 \texttt{ mod } q$$

$$\cdots$$

$$l[1] * a_{1n} + \cdots + l[m] * a_{mn} = c_n \texttt{ mod } q$$

$$l[1] * pk_1 + \cdots + l[m] * pk_m = v - c_{n+1} \texttt{ mod } q$$

Altogether, from an encryption, $n+2$ equations can be defined with $m+1$ unknowns $l[1]$,. . . ,$l[m]$, and v. Our scheme requires $m > n+1$. Thus, the $m+1$ unknowns cannot be uniquely determined by the $n+2$ equations. As to be discussed later, the parameters (e.g., w, m and n) need to be chosen to ensure there is a large number of possible combinations of public key samples when generating a ciphertext, hence allowing a large number of solutions to the above underdetermined system of linear equations.

5 Concrete Security Level and Size-Adaptable Ciphertexts

In this section, we give the estimation of the concrete security level of our scheme. The estimation is for private key security and message security, respectively, related to the attack analysis in the last section. We also discuss the size of ciphertexts, which can be adaptable to the size of message spaces, without changing the security level of private keys.

5.1 Security Level of Private Keys

The security level of a private key is determined by the parameters $pp = (q, n, m, t, w, b)$, and the distributions of three secret values sk, r and p in the private key. As discussed in the last section, the current attacks to LWE are not applicable to our scheme. Hence, to recover a private key from the corresponding public key of our scheme, the adversary needs to correctly guess sk, p, and all e_i in at least n public key samples. Let $|sk|$ and $|p|$ denote the size of distributions from which sk and p are sampled, respectively, and sk_i $(1 \leq i \leq |sk|)$ and p_j $(1 \leq j \leq |p|)$ be the ith and jth elements that can be sampled in the corresponding distributions. Then, the security level of the private key in our scheme is defined as $\log_2\left(\sum_{i=1}^{|sk|}\sum_{j=1}^{|p|} r^n\right)$ bits, where $r \leq \frac{q-1-sk_i*(t-1)}{w*p_j}$.

On the other hand, the adversary can take $sk_q^{-1} * p$ as a single secret value, without guessing sk and p individually. By this way, the adversary deals with a system of linear equations with $n+1$ unknowns, so it needs to guess e_i for $n+1$ equations. Thus, the security level will be $\log_2(r^{n+1})$ bits. The final security level for private keys is the minimal value between $\log_2\left(\sum_{i=1}^{|sk|}\sum_{j=1}^{|p|} r^n\right)$ and $\log_2(r^{n+1})$.

Given a security level of private keys in our scheme, the domains of r, sk and p are not public, making it harder for the adversary to launch a brute-force attack. That is, the adversary has no exact knowledge on what are all possible values of e_i, sk, and p that should be guessed.

5.2 Security Level of Messages

The security level of encrypted messages is determined by three public parameters m, n and w. As described in Section 4.2, from an encryption, we can construct a linear system of $n+2$ equations with $m+1$ unknown variables (i.e., the variables $l[i]$ for $1 \leq i \leq m$ and the variable v). The unknown variable v appears only in one equation. If $l[i]$ can be determined from the $n+1$ equations, in which v is not involved, then v can be recovered. Hence, we consider only those $n+1$ equations, which includes m unknown variables $l[i]$.

By using the Gaussian elimination, the $n+1$ equations can be reduced to a new equation with $m-n$ unknowns left over from $l[1], ..., l[m]$. In this new equation, let the $m-n$ unknowns be represented as $l'[i]$ $(1 \leq i \leq m-n)$. Then, the new equation after Gaussian elimination has the format

$$l'[1] * a'_1 + \cdots + l'[m-n] * a'_{m-n} = pk' \texttt{ mod } q,$$

where $0 \leq l'[i] \leq w$ and $l'[1] + \cdots + l'[m-n] \leq w$.

The value of $l'[1] + \cdots + l'[m-n]$ takes the binomial distribution, with the success probability $sp = \frac{m-n}{m}$ and w times of experiments. The probability sp means that a selection of a public key sample in encryption is accumulated by the sum $l'[1] + \cdots + l'[m-n]$, rather than by the eliminated unknowns. Let $\Pr(k)$ denote the probability of $l'[1] + \cdots + l'[m-n] = k$. Then, we have

$$\Pr(k) = \frac{w!}{k! * (w-k)!} * (sp)^k * (1-sp)^{w-k}.$$

Given a sum $k = l'[1] + \cdots + l'[m-n]$, there are different combinations of $l'[i]$ to make the sum. The number of such combination is

$$\frac{(k+m-n-1)!}{k!(m-n-1)!}.$$

The security level of a message is thus defined as

$$\log_2\left(\sum_{k=0}^{w}\Pr(k) * \frac{(k+m-n-1)!}{k!(m-n-1)!}\right).$$

5.3 Size of Public Keys and Size of Cipehrtexts

When the public vector $\mathbf{a}_i$ is shared, the size of a public key is about $m * \log_2 q$ bits. The size of a public key is increased with m, which in return leads to a higher level of message security.

The size of a ciphertext is dependent on parameters w, b , n and q. In a $(n+1)$-dimensional ciphetrext, the last element has $\log_2 q$ bits, and other n elements has either $\log_2(w * b)$ bits or $\log_2 q$ bits, depending on which one is smaller. Hence, the size of a ciphertext is $\log_2 q + n * \min(\log_2(w*b), \log_2 q)$ bits.

Moreover, the size of our ciphertexts can be adaptable to the size of a message space. That is, given a security level, the ciphertexts can have a big size for a big message space, or a small size for a small message space.

The security level of private keys and messages of our scheme is directly determined by the parameters m, n, w, r, $|sk|$, and $|p|$. Without changing the security level, one way of adapting the size of our ciphertexts to the size of a message space is to increase or decrease q, t and p, as discussed below.

The size of a message space is determined by the value of t. For a bigger t, we can increase q to ensure the correctness condition $sk * (t-1) + w * r * p < q$ still holds, leading to bigger ciphertexts. Similarly, when t becomes smaller, q can be decreased to produce smaller ciphertexts.

6 Implementation and Evaluation

In this section, we describe the parameter selection in our experiment and the adaption of the decryption algorithm to the Contiki operating system and MTM-CM5000-MS. Then, we evaluate the error distribution in public key samples and verify the attack resistance of our scheme to lattice-based attacks with a tool. At last, we report the implementation and performance of the leveled Needham-Schroeder-Lowe (NSL) public key authentication protocol over two MTM-CM5000-MSP wireless sensor nodes using our encryption scheme.

6.1 Parameter Selection

In this prototype, we use c language to implement our scheme on Contiki, which is an open-source operating system for IoT devices, such as MTM-CM5000-MSP. In Contiki, the data type `unsigned long` (of the c programming language) is the largest integer type with 4 bytes. When the sum or product of two integers of `unsigned long` type is equal to or bigger than 2^{32}, the operation of modulo 2^{32} is implicitly enforced by the underlying hardware. For example, the sum of two `unsigned long` integers 1 and 4294967295 is zero on Contiki and MTM-CM5000-MSP.

We thus choose $q = 2^{32}$ to benefit from the modulo operations enforced by the underlying hardware. The parameters in our experiment are listed in Table 1 and Table 2. Since t is 2^{16}, we can encrypt 16-bit messages. These parameters satisfy the conditions $n + 1 < m < n^2$, $b > n$, $(2\log_2 b * b + 2) * b < q$, and $2\log_2 b < n$.

The NSL public key authentication protocol includes two parties, called A and B, to be mutually authenticated. In Table 2, we specify the domains of private parameters sk and p for each of them. As described below, this selection of private parameters can keep the security level of private keys above 138-bits.

q	t	m	w	n	b
2^{32}	2^{16}	74	86	13	16

Table 1. Public Parameters

	Domain of sk
A	$\{2 * x + 1 \| 0 \leq x \leq 50\}$
B	$\{2 * x + 1 \| 0 \leq x \leq 500\}$

	Domain of p
A	$\{t + 2 * x + 1 \| 0 \leq x \leq 500\}$
B	$\{t + 2 * x + 1 \| 0 \leq x \leq 50\}$

Table 2. Private Parameters for A and B in NSL Protocol

For the private parameters for A, when $sk = 101$ and $p = 65536 + 1001 = 66537$, r takes its minimal value

$$\frac{2^{32} - 1 - 101 * (65536 - 1)}{86 * 66537} = 749.43,$$

which satisfies the condition $b < r$ specified in the key generation algorithm. Thus, the security level of the private key for A is above

$$\log_2 \left(50 * 500 * (749.43)^{13}\right) = 138 \text{ bits}.$$

Similarly, for B, the minimal value of r is

$$\frac{2^{32} - 1 - 1001 * (65536 - 1)}{86 * 65637} = 749.25,$$

satisfying the condition $b < r$, and the security level of $B's$ private key is at least

$$\log_2 \left(500 * 50 * (749.25)^{13}\right) = 138 \text{ bits.}$$

The above configuration shows that given the security level, the secret domains of sk and p cannot be exactly determined, hence making it harder for the adversary to guess all possible combinations of sk and p in brute-force attacks.

Let $sp = \frac{74-13}{74}$. The security level of messages for this configuration is

$$\log_2 \left(\sum_{k=0}^{86} Pr(k) * \frac{(74+k-13-1)!}{k!(74-13-1)!} \right) = 129.12 \text{ bits},$$

where

$$\Pr(k) = \frac{86!}{k! * (86-k)!} * (sp)^k * (1-sp)^{86-k}.$$

There are 74 public key samples. When $\mathbf{a}_i$ is shared, the public key samples include only the components pk_i $(1 \leq i \leq 74)$, which have $74 * 32 = 2368$ bits. Since in this configuration $\log_2(w * b) < \log_2 q$, a ciphertext has at least $\log_2 q + n * \log_2(w * b) = 32 + 13 * \log_2(1376) = 32 + 13 * 10.42 = 167.46$ bits; in our implementation, the actual size of ciphertexts is 176 bits.

6.2 Adaption of Decryption Algorithm to Contiki

As described above, we exploit the hardware to execute the modular operation $\texttt{mod}\ q$ in our implementation. For the third step of decryption, i.e., the calculation of $v = sk_p^{-1} * skv \texttt{ mod } p$, if $sk_p^{-1} * skv \geq q$, then the hardware executes an operation of $\texttt{mod}\ q$ unexpectedly.

To solve this implementation problem, we revise the decryption algorithm by dividing sk_p^{-1} into h shares, i.e., $sk_p^{-1} = sk_{p1}^{-1} + \cdots + sk_{ph}^{-1}$, such that $sk_{p1}^{-1} * (p-1) < q$,..., and $sk_{ph}^{-1} * (p-1) < q$. We have $h = 2$ in our implementation. The revised decryption algorithm decrypts a ciphertext $\mathbf{c} = (\mathbf{a}, d)$ in the following revised steps.

- Calculate $c' = \langle \mathbf{a}, \mathbf{s} \rangle + d \texttt{ mod } q$.
- Calculate $skv = sk * c' \texttt{ mod } q$.
- Calculate $skv' = skv \texttt{ mod } p$.
- For $i = 1$ to h, calculate $v_i = sk_{pi}^{-1} * skv' \texttt{ mod } p$.
- Calculate $v = v_1 + \cdots + v_h \texttt{ mod } p$.

6.3 Performance of Encryption and Decryption

To evaluate the concrete performance of our scheme, we run the implementation on a MTM-CM5000-MSP device, which has 8MHz CPU frequency. A pair of private key and public key is generated with the private parameters for A. Then, we encrypt a set of integers, and after all encryptions, we decrypt and print the decrypted message to check decryption correctness.

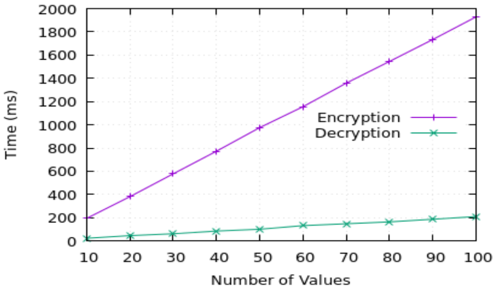

Fig. 2. Performance of Encryptions and Decryptions

The performance of encryption and decryption is shown in Fig. 2. In one second, the device can encrypt about 50 messages and decrypt about 500 ciphertexts. This performance is higher than the performance of the lightweight scheme proposed in [13], where 33 encryptions and 79.4 decryptions performed per second on a 32MHz ARM Cortex-M0 processor for 84-bit security.

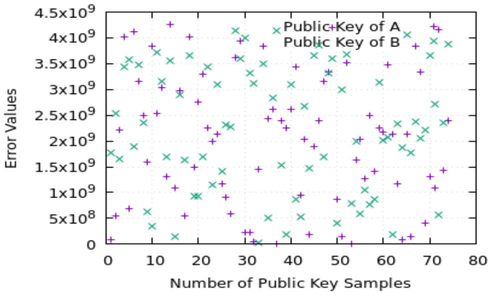

Fig. 3. Secretly-Scaled Errors in Public Keys

6.4 Secretly-Scaled Errors in Public Keys

The secretly-scaled errors in our public key samples can be as big as q. With the configured parameters, we check the errors used in the generation of public keys with experiments. In the experiment, the key generation algorithm runs for A and B, respectively, with the public parameters in Table 1 and their private parameters in Table 2.

The errors $e_i * sk_q^{-1} * p \mod q$ are shown in Fig. 3. This experiment shows that the scaled errors in our scheme are spread evenly across $\mathbb{Z}_q$. For such secretly scaled errors, if an adversary wants any chance of launching attacks to our scheme, the secret

$sk_q^{-1} * p$ must be firstly guessed. In the next experiment, we will assume $sk_q^{-1} * p$ has been guessed correctly.

6.5 Evaluation of Lattice-Based Attacks

The tool[3] (referred to as the KMW tool) developed in [28] is used in this experiment. The first step of this tool is to generate LWE samples. We change this step, letting the tool generate Compact-LWE samples $(\mathbf{a}_i, \langle \mathbf{a}_i, \mathbf{s} \rangle + k * e_i \bmod q)$, where $k = sk_q^{-1} * p$. Since k is assumed to be guessed by the adversary, we then change the Compact-LWE samples into $(k^{-1}\mathbf{a}_i, \langle k^{-1}\mathbf{a}_i, \mathbf{s} \rangle + e_i \bmod q)$ as the output samples. The rest of steps in the tool are not changed.

The errors in Compact-LWE are sampled uniformly, while in LWE they are usually drawn from a Gaussian distribution. We evaluate the errors from both the uniform distribution and the Gaussian distribution. For the standard deviation of the Gaussian distribution, we let it be 187, so that 95% error values drawn from the Gaussian distribution is within $[-374, 374]$, roughly matching the possible number of e_i in our configuration (i.e., [0, 749]). For the uniform distribution, we let the KMW tool to sample the errors from $[-374, 374]$ uniformly.

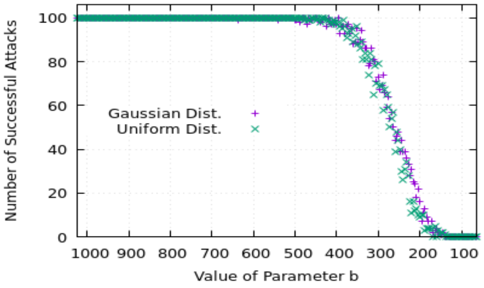

Fig. 4. Effectiveness of Attack Resistance

6.5.1 Effectiveness of Attack Resistance In this experiment, we still let $q = 2^{32}$, $n = 13$, and $m = 74$, as in our parameter configuration. If k is guessed and both uniform distribution and Gaussian distribution are considered, the only difference between LWE and Compact-LWE is the range of $\mathbf{a}_i$, which is bounded by b.

Hence, to evaluate the attack resistance effectiveness of Compact-LWE, we change the values b from 1024 to 64, and for each value we run the KMW tool 100 times, recording the number of successful attacks. Each time the KMW tool generates 74 new Compact-LWE samples to attack. A successful attack means that the vector $\mathbf{A}^T\mathbf{s}$ is found and thus the secret vector $\mathbf{s}$ can be recovered.

[3] https://github.com/pfasante/cvp-enum

Fig 4 shows the evaluation result for both the Gaussian distribution and the uniform distribution. For both of them, when b is bigger than about 450, almost every attack is a successful attack. From 450 downward, the number of successful attacks starts decreasing. When b is decreased to 136 and smaller values, there is no successful attacks for both distributions.

In our configuration, we have $b = 16$. Hence, the lattice-based attacks cannot succeed on Compact-LWE with our configuration. Since LWE takes $b = q = 2^{32}$, those attacks to LWE succeed easily for the same dimension parameter $n = 13$.

6.5.2 Attack Resistance with a Bonus Secret In our encryption scheme, a public key sample is defined as

$$(\mathbf{a}_i, pk_i = \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i * sk_q^{-1} * p \bmod q).$$

Let $bs \in \mathbf{Z}_q$ be a new secret value. In this experiment, we change the public key sample into

$$(\mathbf{a}_i, pk_i = \langle \mathbf{a}_i, \mathbf{s} \rangle + bs + e_i * sk_q^{-1} * p \bmod q).$$

With this revised public key, our encryption algorithm does not change and more importantly the size of ciphertexts does not change. Hence, we call bs a bonus secret. The implementation of our scheme is based on this revision. Since w times of bs are added up in a ciphertext, the decryption algorithm needs to slightly revise its first step into $c' = \langle \mathbf{a}, \mathbf{s} \rangle + w * bs + d \bmod q$.

Recall that in the lattice-based attacks, a matrix $\mathbf{A} \in \mathbb{Z}_b^{n \times m}$ is constructed by taking $\mathbf{a}_i$ as the columns of $\mathbf{A}$. When bs is included in the public key samples, the matrix $\mathbf{A}$ is actually extended with a new column with all entries being 1. Thus, the lattice generated with the extended matrix is more dense than the lattice formed when bs is not considered, hence preventing the lattice-based attacks further in principle. Even if the bonus secret bs can be removed by subtracting two public key samples, this subtraction however widens the range of error values in the new sample and hence the attack-resistant effect should still be improved. This improvement is confirmed by the following experiment.

In this experiment, we let KMW tool generate 74 Compact-LWE samples with the bonus secret, and subtract the first sample from each of the rest samples, obtaining 73 samples with the bonus secret removed. k is still assumed to be guessed by the adversary. Fig 5 shows the evaluation result, which confirms the attack-resistant effect is improved significantly, in particular for the uniformly distributed errors.

For the uniformly distributed errors, there is no any successful attacks when b is less than 836. There is one successful attack for the Gaussian distributed errors when b is as small as 156, but the total number of successful attacks are obviously reduced. This experiment shows that our scheme is more concretely secure when errors e_i are sampled from a uniform distribution when a bonus secret is used. Our scheme indeed selects e_i from a uniform distribution. Hence, our scheme in this configuration ($b = 16$) is protected from the lattice-based attacks very effectively, even when k is assumed to be guessed correctly.

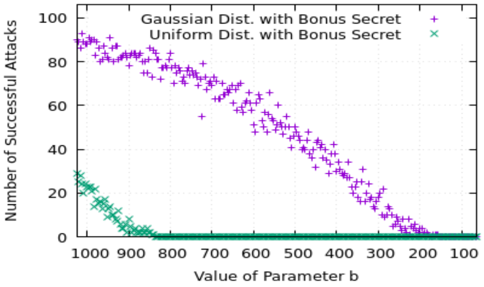

Fig. 5. Effectiveness of Attack Resistance

6.6 Leveled Needham-Schroeder Public-Key Authentication Protocol

We implement the leveled authentication of two MTM-CM5000-MSP devices with the NSL public key authentication protocol and our lightweight public key encryption scheme. Let PK_A and PK_B be the public keys of A and B, respectively. In the NSL protocol, A and B exchange three messages to authenticate each other, where n_A and n_B are fresh nonces.

$A \rightarrow B$: $Enc(PK_B, (n_A, A))$
$B \rightarrow A$: $Enc(PK_A, (n_A, n_B, B))$
$A \rightarrow B$: $Enc(PK_B, (n_B))$

To achieve leveled authentication, the NSL protocol needs to be revised to allow A and B to exchange a variational number of nonces, depending on the required authentication level of IoT devices. Let M indicate the authentication level. The NSL protocol revised for leveled authentication is shown below, where the operations $L(\cdot)$ and $H(\cdot)$ return the first and second byte of its parameter, respectively, and $\oplus$ denotes the XOR operation.

$A \rightarrow B$: $Enc(PK_B, (n_A^1, A))$
$B \rightarrow A$: $Enc(PK_A, (L(n_A^1 * B) \oplus H(n_A^1 * B), n_B^1))$
$A \rightarrow B$: $Enc(PK_B, (n_A^2, L(n_B^1 * A) \oplus H(n_B^1 * A)))$
$B \rightarrow A$: $Enc(PK_A, (L(n_A^2 * B) \oplus H(n_A^2 * B), n_B^2))$
$\ldots$
$B \rightarrow A$: $Enc(PK_A, (L(n_A^M * B) \oplus H(n_A^M * B), n_B^M))$
$A \rightarrow B$: $Enc(PK_B, (L(n_B^M * A) \oplus H(n_B^M * A)))$

Note that A can verify the message $L(n_A^i * B) \oplus H(n_A^i * B)$, since A knows both n_A^i and the identity of B, vice versa for B.

To be efficient, the authentication level should be fine-grained, such that each authentication level does not consume too much computation and communication resources of IoT Devices. For this purpose, we choose 1-byte nonces for A and B to exchange. The leveled NSL protocol can benefit from the ciphertext-size adaptability of our scheme for this small nonce space.

The identities of A and B are also supposed to be 1-byte long. Thus each message in the revised NSL protocol has two bytes, which can be encrypted by our scheme with the above configuration. Each ciphertext is 176-bit long.

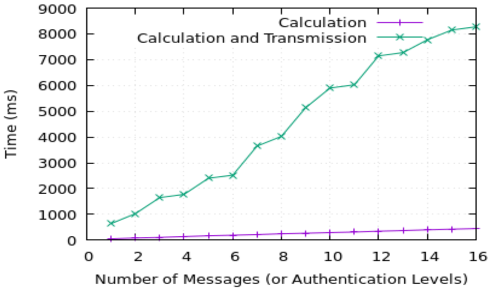

Fig. 6. Performance of Leveled NSL Authentication Protocol

The protocol is evaluated with two MTM-CM5000-MSP devices. Fig 6 shows the time of message calculation (i.e, nonce generation, encryption, decryption and nonce verification) and the total authentication time of the protocol. The latter includes the message calculation time and the time for transmitting messages with the 802.15.4 and 6LoWPAN protocols. We can see the execution of our scheme takes a very small fraction of the total authentication time.

Due to the unreliability of message transmission, the total authentication time of the protocol is quite variable. Generally, a lower level of authentication takes less time to complete, such as 640ms for the authentication at the first level and 8373ms for the 16th level authentication. Thus, the IoT devices can choose appropriate authentication levels according to the sensitivity of application messages to be protected.

7 Related Work

The secret vector $\mathbf{s}$ in LWE can be sampled from the set $\{0,1\}^n$ or $\{-1,0,1\}^n$ [36]. This variant is called binary-LWE and it does not induce dense lattices as Compact-LWE. Though it is also proven hard asymptotically, the 140-dimensional binary-LWE can be attacked successfully in a number of hours, as shown in [28]. Hence, the schemes based on binary-LWE must choose big dimension parameters for concrete security.

Lindner and Peikert [31] analyzed the concrete security of the LWE-based encryption schemes by giving a lattice-based attack. For the security level of 128 bits, the public key size shall be 1120K bits. To enable a shorter public key size, Ring Learning With Errors (Ring-LWE) was introduced by Lyubashevsky, Peikert and Regev [35]. The Ring-LWE based encryption schemes are able to reduce the public key size by a factor of n. For example, the public key size of [31] needs 2-5K bits by using Ring-LWE problem. However, the lightweight public encryption based on Ring-LWE still produces ciphertexts too big ciphertexts (e.g., 2.8K bits).

The error distribution of LWE can be replaced by other distributions. For example, Buchmann et al. [14] studied a variant of LWE with binary errors which are chosen from the uniform distribution on $\{0, 1\}$. It is able to improve the performance of lightweight lattice-based public key encryption schemes [13]. The proposed scheme [13] achieves 84 bits security level by using the private key and public key with the size 256 bits and 2048 bits, respectively, based on the hardness of Ring-Binary LWE problem.

As a lattice-based hard problem, LWE has been used in various public key cryptographic primitives, such as oblivious transfer protocols [40], identity-based encryption [1], attribute-based encryption [10] and fully homomorphic encryption [12, 19]. Due to the inherent security requirements of LWE, the public key and ciphertext sizes are not comparable to traditional constructions of these primitives. Compact-LWE might also be applied in other cryptographic primitives to reduce their ciphertext sizes.

Hoffstein, Pipher and Silverman [25] proposed a lattice-based public key encryption scheme NTRU. It is considered as a practical lattice-based encryption scheme without known efficient attacks, while the security is not formally proved. Stehlé and Steinfeld [43] modified the NTRU encryption scheme and then it is provably secure in the standard model. However, Cabarcas, Weiden and Buchmann [15] pointed out that such scheme is not practical.

Algebraic Eraser (AE) was introduced by Anshel, Anshel, Goldfeld and Lemieux [3]. AE is claimed an efficient and secure lightweight (Diffie-Hellman like) key exchange protocol for resource constrained devices. Specifically, the shared secret size needs to be 728 bits for 80 bits security. Therefore, the ciphertext size of AE-based encryption scheme (on the same security level) is estimated larger than 728 bits. The security of AE is still questionable [5, 6]. A limitation of AE is that the system setup must be conducted by a trusted third party.

Elliptic Curve Cryptography (ECC) is suitable to lightweight devices due to the short key size and efficiency. Many ECC-based lightweight protocols (e.g., [29, 30, 24]) were proposed for RFID and other lightweight devices. However, the underlying computation complexity prohibits the significant improvement of encryption and decryption performance in practice.

Multilevel authentication in Oracle Application Server[4] allows different authentication levels to be assigned to different applications. Their authentication levels are determined by a fixed number of authentication factors. For example, user names and passwords are used for a low level authentication, while a high level authentication is achieved with certificates. For our leveled authentication, there can be any number of authentication levels, which can be increased gradually with fine granularity.

8 Conclusion

In this paper, we proposed the Compact-LWE problem, based on which a lightweight public encryption scheme has been constructed. The hardness of Compact-lWE was proven by giving the reduction from LWE to Compact-LWE. As a big deviation from other LWE-based or lattice-based encryption schemes, our scheme is still secure, even if the well-known closest vector problem in lattices can be solved.

[4] http://docs.oracle.com/cd/B14099_19/idmanage.1012/b14078/multilevel.htm

Due to the attack-resistance of Compact-LWE, we can choose small dimension parameters for our encryption scheme to generate short ciphertexts. The concrete security of our scheme was confirmed with a lattice-based attack tool.

We implemented our scheme on the Contiki operating system and used it to implement the leveled authentication based on the Needham-Schroeder-Lowe public key authentication protocol. The evaluation on MTM-CM5000-MSP wireless sensor devices showed that our lightweight public key encryption scheme and the leveled NSL authentication protocol can be practically deployed in small IoT devices.

References

1. Agrawal, S., Boneh, D., Boyen, X.: Efficient lattice (H)IBE in the standard model. In: Gilbert, H. (ed.) Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6110, pp. 553–572. Springer (2010)
2. Albrecht, M.R., Cid, C., Faugère, J.C., Fitzpatrick, R., Perret, L.: On the complexity of the bkw algorithm on lwe. Des. Codes Cryptography 74(2), 325–354 (Feb 2015)
3. Anshel, I., Anshel, M., Goldfeld, D., Lemieux, S.: Key agreement, the Algebraic Eraser^{TM}, and lightweight crytography. Contemporary Mathematics 416(2006), 1–34 (2006)
4. Arora, S., Ge, R.: New algorithms for learning in presence of errors. In: Proceedings of the 38th International Colloquim Conference on Automata, Languages and Programming - Volume Part I. pp. 403–415. ICALP'11, Springer-Verlag, Berlin, Heidelberg (2011), http://dl.acm.org/citation.cfm?id=2027127.2027170
5. Ben-Zvi, A., Blackburn, S.R., Tsaban, B.: A practical cryptanalysis of the algebraic eraser. In: Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part I. pp. 179–189 (2016)
6. Blackburn, S.R., Robshaw, M.J.B.: On the security of the algebraic eraser tag authentication protocol. In: Applied Cryptography and Network Security - 14th International Conference, ACNS 2016, Guildford, UK, June 19-22, 2016. Proceedings. pp. 3–17 (2016)
7. Blum, A., Kalai, A., Wasserman, H.: Noise-tolerant learning, the parity problem, and the statistical query model. J. ACM 50(4), 506–519 (2003)
8. Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J., Seurin, Y., Vikkelsoe, C.: Present: An ultra-lightweight block cipher. In: Proceedings of the 9th International Workshop on Cryptographic Hardware and Embedded Systems. pp. 450–466. CHES '07 (2007)
9. Bos, J., Costello, C., Ducas, L., Mironov, I., Naehrig, M., Nikolaenko, V., Raghunathan, A., Stebila, D.: Frodo: Take off the ring! practical, quantum-secure key exchange from lwe. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. pp. 1006–1018. CCS '16, ACM (2016)
10. Boyen, X., Li, Q.: Attribute-based encryption for finite automata from LWE. In: Au, M.H., Miyaji, A. (eds.) Provable Security - 9th International Conference, ProvSec 2015, Kanazawa, Japan, November 24-26, 2015, Proceedings. Lecture Notes in Computer Science, vol. 9451, pp. 247–267. Springer (2015)
11. Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: Classical hardness of learning with errors. In: Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing. pp. 575–584. STOC '13 (2013)

12. Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) LWE. SIAM J. Comput. 43(2), 831–871 (2014)
13. Buchmann, J.A., Göpfert, F., Güneysu, T., Oder, T., Pöppelmann, T.: High-performance and lightweight lattice-based public-key encryption. In: Chow, R., Saldamli, G. (eds.) Proceedings of the 2nd ACM International Workshop on IoT Privacy, Trust, and Security, IoTPTS@AsiaCCS, Xi'an, China, May 30, 2016. pp. 2–9. ACM (2016)
14. Buchmann, J.A., Göpfert, F., Player, R., Wunderer, T.: On the hardness of LWE with binary error: Revisiting the hybrid lattice-reduction and meet-in-the-middle attack. In: Pointcheval, D., Nitaj, A., Rachidi, T. (eds.) Progress in Cryptology - AFRICACRYPT 2016 - 8th International Conference on Cryptology in Africa, Fes, Morocco, April 13-15, 2016, Proceedings. Lecture Notes in Computer Science, vol. 9646, pp. 24–43. Springer (2016)
15. Cabarcas, D., Weiden, P., Buchmann, J.A.: On the efficiency of provably secure NTRU. In: Mosca, M. (ed.) Post-Quantum Cryptography - 6th International Workshop, PQCrypto 2014, Waterloo, ON, Canada, October 1-3, 2014. Proceedings. Lecture Notes in Computer Science, vol. 8772, pp. 22–39. Springer (2014)
16. Chan, H., Perrig, A., Song, D.X.: Random key predistribution schemes for sensor networks. In: 2003 IEEE Symposium on Security and Privacy (S&P 2003), 11-14 May 2003, Berkeley, CA, USA. p. 197 (2003), http://dx.doi.org/10.1109/SECPRI.2003.1199337
17. Ducas, L., Durmus, A., Lepoint, T., Lyubashevsky, V.: Lattice Signatures and Bimodal Gaussians, pp. 40–56. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
18. Eschenauer, L., Gligor, V.D.: A key-management scheme for distributed sensor networks. In: Proceedings of the 9th ACM Conference on Computer and Communications Security. pp. 41–47. CCS '02, ACM (2002)
19. Gentry, C., Sahai, A., Waters, B.: Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In: Canetti, R., Garay, J.A. (eds.) Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I. Lecture Notes in Computer Science, vol. 8042, pp. 75–92. Springer (2013)
20. Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of Computer and System Sciences 28(2), 270–299 (1984)
21. Guo, J., Peyrin, T., Poschmann, A., Robshaw, M.: The led block cipher. In: Proceedings of the 13th International Conference on Cryptographic Hardware and Embedded Systems. pp. 326–341. CHES'11, Springer-Verlag, Berlin, Heidelberg (2011), http://dl.acm.org/citation.cfm?id=2044928.2044958
22. Gura, N., Patel, A., Wander, A., Eberle, H., Shantz, S.C.: Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs, pp. 119–132. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)
23. Hanrot, G., Pujol, X., Stehlé, D.: Algorithms for the shortest and closest lattice vector problems. In: Proceedings of the Third International Conference on Coding and Cryptology. pp. 159–190. IWCC'11 (2011)
24. Hermans, J., Peeters, R., Preneel, B.: Proper RFID privacy: Model and protocols. IEEE Trans. Mob. Comput. 13(12), 2888–2902 (2014)
25. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: A ring-based public key cryptosystem. In: Buhler, J. (ed.) Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon, USA, June 21-25, 1998, Proceedings. Lecture Notes in Computer Science, vol. 1423, pp. 267–288. Springer (1998)
26. Katz, J., Lindell, Y.: Introduction to Modern Cryptography. Chapman & Hall/CRC (2014)
27. Kirchner, P., Fouque, P.: An improved BKW algorithm for LWE with applications to cryptography and lattices. In: Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part I. pp. 43–62 (2015)

28. Kirshanova, E., May, A., Wiemer, F.: Parallel implementation of BDD enumeration for LWE. In: Applied Cryptography and Network Security - 14th International Conference, ACNS 2016, Guildford, UK, June 19-22, 2016. Proceedings. pp. 580–591 (2016)
29. Lee, Y.K., Batina, L., Singelée, D., Verbauwhede, I.: Low-cost untraceable authentication protocols for RFID. In: Wetzel, S., Nita-Rotaru, C., Stajano, F. (eds.) WISEC. pp. 55–64. ACM (2010)
30. Lee, Y.K., Batina, L., Verbauwhede, I.: Untraceable RFID authentication protocols: Revision of EC-RAC. In: RFID, 2009 IEEE International Conference on. pp. 178 –185 (2009)
31. Lindner, R., Peikert, C.: Better key sizes (and attacks) for lwe-based encryption. In: Kiayias, A. (ed.) Topics in Cryptology - CT-RSA 2011 - The Cryptographers' Track at the RSA Conference 2011, San Francisco, CA, USA, February 14-18, 2011. Proceedings. Lecture Notes in Computer Science, vol. 6558, pp. 319–339. Springer (2011)
32. Liu, D., Ning, P.: Establishing pairwise keys in distributed sensor networks. In: Proceedings of the 10th ACM Conference on Computer and Communications Security. pp. 52–61. CCS '03, ACM, New York, NY, USA (2003), http://doi.acm.org/10.1145/948109.948119
33. Liu, M., Nguyen, P.Q.: Solving bdd by enumeration: An update. In: Proceedings of the 13th International Conference on Topics in Cryptology. pp. 293–309. CT-RSA'13 (2013)
34. Lowe, G.: An attack on the needham-schroeder public-key authentication protocol. Inf. Process. Lett. 56(3), 131–133 (1995)
35. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: Gilbert, H. (ed.) Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6110, pp. 1–23. Springer (2010)
36. Micciancio, D., Peikert, C.: Hardness of SIS and LWE with small parameters. In: Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I. pp. 21–39 (2013)
37. Needham, R.M., Schroeder, M.D.: Using encryption for authentication in large networks of computers. Commun. ACM 21(12), 993–999 (1978)
38. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem: Extended abstract. In: Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing. pp. 333–342. STOC '09 (2009)
39. Peikert, C.: A decade of lattice cryptography. Foundations and Trends in Theoretical Computer Science 10(4), 283–424 (2016), http://dx.doi.org/10.1561/0400000074
40. Peikert, C., Vaikuntanathan, V., Waters, B.: A framework for efficient and composable oblivious transfer. In: Wagner, D. (ed.) Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings. Lecture Notes in Computer Science, vol. 5157, pp. 554–571. Springer (2008)
41. Pöppelmann, T., Oder, T., Güneysu, T.: High-performance ideal lattice-based cryptography on 8-bit atxmega microcontrollers. In: Proceedings of the 4th International Conference on Progress in Cryptology – LATINCRYPT 2015 - Volume 9230. pp. 346–365 (2015)
42. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Gabow, H.N., Fagin, R. (eds.) Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005. pp. 84–93. ACM (2005)
43. Stehlé, D., Steinfeld, R.: Making NTRU as secure as worst-case problems over ideal lattices. In: Paterson, K.G. (ed.) Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings. Lecture Notes in Computer Science, vol. 6632, pp. 27–47. Springer (2011)
44. Zhang, J., Zhang, Z., Ding, J., Snook, M., Dagdelen, Ö.: Authenticated Key Exchange from Ideal Lattices, pp. 719–751. Springer Berlin Heidelberg, Berlin, Heidelberg (2015)