Access Control Encryption Based on LWE

Gaosheng Tan
Institute of Information Engineering, CAS
Beijing, China
tangaosheng@iie.ac.cn

Rui Zhang
Institute of Information Engineering, CAS
Beijing, China
r-zhang@iie.ac.cn

Hui Ma
Institute of Information Engineering, CAS
Beijing, China
mahui@iie.ac.cn

Yang Tao
Institute of Information Engineering, CAS
Beijing, China
taoyang@iie.ac.cn

ABSTRACT

Damgård et al. proposed a new primitive called access control encryption (ACE) [6] which not only protects the privacy of the message, but also controls the ability of the sender to send the message. We will give a new construction based on the Learning with Error (LWE) assumption [12], which is one of the two open problems in [6]. Although there are many public key encryption schemes based on LWE and supporting homomorphic operations. We find that not every scheme can be used to build ACE. In order to keep the security and correctness of ACE, the random constant chosen by the sanitizer should satisfy stricter condition. We also give a different security proof of ACE based on LWE from it based on DDH. We will see that although the modulus of LWE should be super-polynomial, the ACE scheme is still as secure as the general public key encryption scheme based on the lattice [5].

CCS Concepts

•**Security and privacy** → **Public key encryption;** *Cryptography;*

Keywords

Access Control Encryption; Lattice; Fully Homomorphic Encryption

1. INTRODUCTION

Access Control Encryption (ACE) is a new cryptographic primitive allowing fine-gained access control. It not only controls who can read the encrypted message, but also determines who can send the message. For ACE, there are two security requirements called the no-read rule and the no-write rule. We'll give a detailed security model for these two rules in Section 3. Informally, the no-read rule is similar to the conventional public encryption scheme. The one who has no decryption key cannot read the message under the ciphertext. However, the no-write rule seems hard to be realized by the pure cryptography methods, because it always hard to forbid someone to send something.

To realize the no-write rule, Damgård et al. inserted a third party called the sanitizer between the sender and the receiver. They use the sanitizer to control the information flow. If the sender is allowed to communicate with the receiver, then the sanitizer makes no effect on the ciphertext and the receiver can decrypt and get the original plaintext. However, if the sender is forbidden to send messages to the receiver, the sanitizer will make a dominating influence on the ciphertext and the receiver only attains a random plaintext. They used some encryption scheme with homomorphic addition property to realize this functionality of the sanitizer. In a nutshell, the sender outputs the ciphertext $c = (\mathrm{Enc}(ek), \mathrm{Enc}(\mu))$ where ek is the sender's encryption secret key and μ is the message. Then the sanitizer compute a new ciphertext $c'=r(\mathrm{Enc}(ek)+\mathrm{Enc}(rk))+\mathrm{Enc}(\mu)$ where r is a random number and rk is the sanitizer's re-randomized secret key. Obviously, when ek is equal to $-rk$, then by the homomorphic addition of the encryption scheme, c' is still an ciphertext of μ. However, if ek is not equal to $-rk$, then it becomes a ciphertext of $(r(ek+rk)+\mu)$ which is a random message.

Damgård et al. constructed two ACE schemes based on DDH and Pallier. One of the advantages of these schemes is their efficiency. But under the quantum computer, DDH and Pallier are polynomial-time solvable. It is worthwhile to construct an ACE scheme based on LWE to defense the quantum attacks besides the author also set this as one of the two open problems. Fuchsbauer et al. solve the other open problem and they construct ACE schemes with polylogarithmic complexity (in the number of possible identities in the system) from standard pairing assumptions [7].

There are many excellent fully homomorphic encryption schemes based on LWE or Ring-LWE [10] such as [4] [3] [9] [1]. However, it is not fit for every scheme to serve as the building block. For example, use Brakerski's scheme in 2011 [4] as the building block. The public key is $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^m$ and the decryption key is $\mathbf{s} \in \mathbb{Z}_q^n$, where $\mathbf{b} = \mathbf{As} + 2\mathbf{e}$. The error vector $\mathbf{e}$ is from a distribution χ^m.

APKC'17, April 02 2017, Abu Dhabi, United Arab Emirates

DOI: http://dx.doi.org/10.1145/3055504.3055509

To sent the message $\mu \in \{0,1\}$, the sender computes the ciphertext $c = (c_1, c_2)$ and sents it to the sanitizer where

$$c_1 = (\mathbf{r}_{ek}^T \mathbf{A}, \mathbf{r}_{ek}^T \mathbf{b} + ek),$$

$$c_2 = (\mathbf{r}^T \mathbf{A}, \mathbf{r}^T \mathbf{b} + \mu).$$

For controlling the right of the sender to send messages, the sanitizer computes a ciphertext

$$c_3 = (\mathbf{r}_{rk}^T \mathbf{A}, \mathbf{r}_{rk}^T \mathbf{b} + rk).$$

Then the sanitizer outputs the final ciphertext

$$c' = (\mathbf{v}, w) = r(c_1 + c_3) + c_2$$

where $r \in \mathbb{Z}_q$ is a random value. We assume that the sender can send the message i.e. $ek = -rk$. To decrypt, the receiver first computes

$$w - \mathbf{v} \cdot \mathbf{s} \equiv 2r(\mathbf{r}_{ek}^T + \mathbf{r}_{rk}^T)\mathbf{e} + 2\mathbf{r}^T \mathbf{e} + m \pmod{q}.$$

In order to decrypt correctly, we need the error in the right smaller than q. But if the random value $r \in \mathbb{Z}_q$, this condition is hard to guarantee. One may think that $r \in \mathbb{Z}_q$ is overkill for just one bit message. But in order to encrypt ek and rk, the plaintext space of the encryption scheme should be $\mathbb{Z}_q$ which means that $r \in \mathbb{Z}_q$ is necessary. To overcome the incorrect decryption caused by the sanitizer's computation, we adopt GSW scheme as the building block. Fortunately, GSW scheme [9] can support $\mathbb{Z}_q$ plaintext space and most importantly the error growth factor in homomorphic scalar-multiplication is independent from the scalar and is just a fixed small constant. So we choose GSW scheme as the building block to construct ACE scheme.

The main contribution of this paper is to construct an ACE scheme based on LWE. As the scheme based on DDH and Pallier, the scheme based on LWE also need to support homomorphic addition and scalar-multiplication computation, which is very easy to satisfy. There are some difficulties to solve before we construct ACE based on LWE.

- We see that the secret key ek and rk of the sender and sanitizer must be chosen from a super-polynomial size domain in order to defense the simple secret key guessing attack. This means that the modulus in LWE must be a super-polynomial positive integer, which is a bad news for the security and the performance of the scheme. For the security, it is the ratio of the modulus and the error in LWE to dominate the approximate factor in the lattice hardness assumptions. When the ratio is super-polynomial, the hardness assumptions must be stronger. Therefore, many techniques are found to control the ratio such as in [2] [5]. We will choose a super-polynomial modulus but still keep the ratio of the modulus and the error be a polynomial of LWE dimension n. This means our construction is as secure as the general public key encryption based on the lattice as claimed by [5].

- We can see that the random r is an important parameter in the DDH-based ACE scheme. It makes $r(ek + rk) + \mu$ to be uniform in the plaintext space. However, in our construction, r (or a value relating with r) will also be a part of the error which must be bounded for decrypting correctly. Obviously, we hope that r is also chosen from $\mathbb{Z}_q$. We find the error of the homomorphic scalar-multiplication of GSW scheme is independent from the constant and is only a fixed number. This means that GSW scheme is perhaps the optimal building block among all the LWE-based full homomorphic encryption schemes to construct ACE scheme.

- Although the lattice is a very simple concept in Algebra. It has a worse algebraic structure than the cyclic or residue class group. This takes some difficulties to the proof of the no-write role. We find that these can be solved by a carefully chosen parameters. The main idea is that we need the rows of the public key matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ of the LWE-based scheme to generate the whole space $\mathbb{Z}_q^n$. When the rows of A are randomly chosen and $m = O(n \log q)$, this condition holds except with negligible probability. We will give a detailed presentation in Section 4.

Similar as the Damgård's construction, we first construct a simple ACE (1-ACE) for one identity. Then we will construct the general scheme for multiple identities by tiling many 1-ACE schemes. In Section 2, we show some preliminaries that we need. In Section 3, we describe the model of ACE and review GSW scheme. We give the construction and analysis of ACE scheme in Section 4 and make a conclusion in Section 5.

2. THE PRELIMINARY

In this paper, the vectors are denoted as bold lower-cased letters, e.g. $\mathbf{b}$. For convenience, we don't distinguish the column and the row vector. The matrices are denoted as bold capital letters, e.g. $\mathbf{A}$ and the ith row or column vector of a matrix $\mathbf{A}$ is written as $\mathbf{A}_i$. For an odd positive integer q, let $\mathbb{Z}_q = \mathbb{Z} \cap [-\frac{q-1}{2}, \frac{q-1}{2})$. For a distribution D, $x \leftarrow D$ means that x is chosen according to the distribution D and if D is a set, it means that x is chosen uniformly from D. The set $\{1, 2, \ldots, n\}$ is denoted as $[n]$. We denote the p-norm of a vector as $||\cdot||_p$.

Definition 1. ($\mathrm{LWE}_{n,m,q,\chi}$) Let n be the dimension of a vector and q be a positive integer modulus. Some distribution over integer $\mathbb{Z}$ is denoted by χ. Then for a fixed vector $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, let $\mathcal{A}_{\mathbf{s},\chi} = \{(\mathbf{a}, b) : b = \langle \mathbf{a}, \mathbf{s} \rangle + e\}$ be a distribution where $\mathbf{a} \leftarrow \mathbb{Z}_q^n$ and $e \leftarrow \chi$. The $\mathrm{LWE}_{n,m,q,\chi}$ problem is given m samples from $\mathcal{A}_{\mathbf{s},\chi}$, to search the vector $\mathbf{s}$.

Above definition is the searching version for LWE. For the convenience of constructing the public key encryption, we usually need its decisional version $\mathrm{DLWE}_{n,m,q,\chi}$ which is to distinguish two distributions i.e. determine whether the m samples are from $\mathcal{A}_{\mathbf{s},\chi}$ or uniform distribution over $\mathbb{Z}_q^{n+1}$. It is believed that the $\mathrm{DLWE}_{n,m,q,\chi}$ problem is infeasible. There is a reduction from the GapSVP problem in lattice to the DLWE. We first give a definition of the B-bounded distribution which plays an important role on the reduction.

Definition 2. (B-bounded definition [9]) A distribution ensemble $\{\chi_n\}_{n \in \mathbb{N}}$ supported over the integers, is called B-bounded if

$$\Pr_{e \leftarrow \chi_n}[|e| > B] = negl(n) \tag{1}$$

THEOREM 1. *(Reduction theorem from [9]). Let q be either a prime power or a product of small (size poly(n)) distinct primes, and let $B \geq \omega(\log n) \cdot \sqrt{n}$. Then there exists an efficient sampleable B-bounded distribution χ such that if there is an efficient algorithm that solves the average-case LWE ($DLWE_{n,m,q,\chi}$) problem for parameters n, m, q, χ, then:*

- *There is an efficient quantum algorithm that solves $GapSVP_{\tilde{O}(nq/B)}$ on any n-dimensional lattice.*
- *If $q \geq \tilde{O}(2^{n/2})$, there is an efficient classical algorithm for $GapSVP_{\tilde{O}(nq/B)}$ on any n-dimension lattice.*

From the theorem, if q is super-polynomial and the error bound B is only polynomial, then the approximate factor of GapSVP is super-polynomial which is not optimal for the security. In order to support polynomial homomorphic multiplication in [9], the authors made q/B to be sub-exponent which requires a stronger lattice assumption. In this paper, we only need one time homomorphic addition and one time constant-multiplication. We can choose a super-polynomial q meanwhile hold q/B just polynomial. So we just need a standard lattice assumption (polynomial approximate factor).

In the proof of the no-write rule of the DDH-based ACE scheme, the authors need to prove that in some case the sanitizer is able to transfer any sender's ciphertext into a uniformly random ciphertext over the sanitizer's ciphertext space except with negligible probability. Specially, they need to prove there exists $(s_1, s_2) \in \mathbb{Z}_q$ satisfying the following equations for any fixed $(\gamma_0, \gamma_1) \in \mathbb{Z}_q$.

$$\begin{cases} \gamma_0 = \delta_2 + s_1\delta_0 + s_2 \\ \gamma_1 = \delta_3 + s_1(\delta_1 - \alpha) + s_2 x \end{cases}$$

This equations have the unique solution unless $\alpha = (\delta_1 - x\delta_0)$ which happens with negligible probability, where x is the decryption key. Under the similar case in our construction, we need to prove that for any vector $\mathbf{b} \in \mathbb{Z}_q^{n+1}$, there exists a binary vector $\mathbf{r} \in \{0,1\}^m$ such that $\mathbf{rA} = \mathbf{b}$ where $\mathbf{A}$ is uniform over $\mathbb{Z}_q^{m \times (n+1)}$. It sufficient to show that the rows of $\mathbf{A}$ generate $\mathbb{Z}_q^{n+1}$. We use the following lemma from [8] to guarantee this sufficient condition.

LEMMA 1. *(Lemma 5.1 from [8]) Let $m \geq 2n \log q$. Then for all but an at most q^{-n} fraction of $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, the subset-sums of the rows of $\mathbf{A}$ generate $\mathbb{Z}_q^n$; i.e. for every syndrome $\mathbf{b} \in \mathbb{Z}_q^n$ there is a $\mathbf{r} \in \{0,1\}^m$ such that $\mathbf{rA} = \mathbf{b} \mod q$.*

From this lemma, we just need to choose $m \geq 2(n+1)\log q$.

3. ACCESS CONTROL ENCRYPTION AND GSW SCHEME

ACE scheme is a quintuple of the PPT algorithms (**Setup, KeyGen, Enc, San, Dec**). We adopt the description in [6]. We use GSW leveled fully homomorphic encryption scheme as the building block for our ACE scheme. For the completeness, we will give a review about GSW scheme.

3.1 Access Control Encryption

An access control encryption (ACE) scheme is defined by the following algorithms [6]:

Setup: The Setup algorithm on input the security parameter λ and a policy $P : [n] \times [n] \to \{0,1\}$ outputs a master secret key msk and public parameters pp. The message space is $\mathcal{M}$ and the ciphertext spaces are $\mathcal{C}, \mathcal{C}'$ of the sender and sanitizer respectively.

KeyGen: A Gen sub-algorithm on input the master secret key msk, an identity $i \in \{0, \ldots, n+1\}$, and a type $t \in \{sen, rec, san\}$ outputs a key k. We use the following notation for the three kind of keys in the system:

- $ek_i \leftarrow Gen(msk, i, sen)$ and call it an *encryption key* for $i \in [n]$;
- $dk_j \leftarrow Gen(msk, j, rec)$ and call it a *decryption key* for $j \in [n]$;
- $ek_0 = dk_0 = pp$;
- $rk \leftarrow Gen(msk, n+1, san)$ and call it the *sanitizer key*;

Enc: The Enc algorithm on input an encryption key ek_i and a message m outputs a ciphertext $c \in \mathcal{C}$.

San: San transforms an incoming ciphertext $c \in \mathcal{C}$ into a sanitizer ciphertext $c' \in \mathcal{C}'$ using the sanitizer key rk.

Dec: Dec recovers a message $m' \in \mathcal{M} \cup \{\perp\}$ from a ciphertext $c' \in \mathcal{C}'$ using a decryption key dk_j.

Note that for $(i,j) \in [n] \times [n]$, if $P(i,j) = 1$ we say sen_i can write down to rec_j, and if $P(i,j) = 0$, we say sen_i cannot write down to rec_j. KeyGen algorithm defines two special identities $i = 0$ or $j = 0$ which represent a sender or receiver with no rights such that $P(0,j) = 0 = P(i,0)$ for all $i, j \in [n]$; $i = n+1$ denotes sanitizer identity, which cannot receive from anyone but can send to all i.e. $P(n+1, j) = 1$ and $P(i, n+1) = 0$ for $\forall i \in [n]$.

Definition 3. (Correctness) For all $m \in \mathcal{M}$, $i, j \in [n]$ such that $P(i,j) = 1$:

$$\Pr[\mathrm{Dec}(dk_j, \mathrm{San}(rk, \mathrm{Enc}(ek_i, m))) \neq m] \leq negl(\lambda) \quad (2)$$

with $(pp, msk) \leftarrow \mathrm{Setup}(1^\lambda, P)$, $ek_i \leftarrow \mathrm{Gen}(msk, i, sen)$, $dk_j \leftarrow \mathrm{Gen}(msk, j, rec)$, and $rk \leftarrow \mathrm{Gen}(msk, n+1, san)$, and the probabilities are taken over the random coins of all algorithms.

The security definition of ACE is a little complex and divided into two parts. The one is for the no-read rule and the other one is for the no-write rule. Before describing the definition of the no-read rule, we first give a no-read rule game between a challenger C and a stateful adversary A as Table 1.

Definition 4. (**No-Read Rule**) Consider the no-read rule game of Table 1 between a challenger C and a stateful adversary A: We say that A wins the no-read rule game if $b = b'$, $|m_0| = |m_1|$, $i_0, i_1 \in \{0, \ldots, n\}$ and one of the following holds:

Payload Privacy: For all queries q to $\mathcal{O}_G$ with $q = (j, rec)$ it holds that

$$P(i_0, j) = P(i_1, j) = 0$$

Table 1: No-Read Rule Game

Game Definition	Oracle Definition
1. $(pp, msk) \leftarrow \mathbf{Setup}(1^\lambda, P)$;	$\mathcal{O}_G(j,t)$:
2. $(m_0, m_1, i_0, i_1) \leftarrow A^{\mathcal{O}_G(\cdot),\mathcal{O}_E(\cdot)}(pp)$;	1. Output $k \leftarrow \mathrm{Gen}(msk, j, t)$;
3. $b \leftarrow \{0,1\}$;	
4. $c \leftarrow \mathbf{Enc}(\mathrm{Gen}(msk, i_b, sen), m_b)$;	$\mathcal{O}_E(i,m)$:
5. $b' \leftarrow A^{\mathcal{O}_G(\cdot),\mathcal{O}_E(\cdot)}(c)$;	1. $ek_i \leftarrow \mathrm{Gen}(msk, i, sen)$;
	2. Output $c \leftarrow \mathbf{Enc}(ek_i, m)$;

Sender Anonymity: For all queries q to $\mathcal{O}_G$ with $q = (j, rec)$ it holds that

$$P(i_0, j) = P(i_1, j) \quad and \quad m_0 = m_1$$

We say an ACE scheme satisfies the no-read rule if for all PPT A

$$\mathrm{Adv}^A = 2|\Pr[\text{A wins the No-Read game}] - \frac{1}{2}| \leq \mathrm{negl}(\lambda).$$

In a nutshell, the above definition means that the one who has no decryption secret key will never read the encrypted message and no one can know the sender's identity. We give two remarks about the no-read rule.

Remark 1: If we don't constrain the access rights of the parties queried by the adversary in the security model, there are some trivial attacks. Therefore, in the payload privacy, there is condition $P(i_0, j) = P(i_1, j) = 0$ for the the query of the decryption key of the receiver j. In the sender anonymity, there are conditions $P(i_0, j) = P(i_1, j)$ and $m_0 = m_1$.

Remark 2: We find that in the sender anonymity, for multiple users, the condition $P(i_0, j) = P(i_1, j)$ is not sufficient. For the adversary who represents the corrupted sanitizer and can query some receivers' decryption key, the condition should be $P(i_0, j) = P(i_1, j) = 0$. Because this adversary attains senders' ciphertext as the challenge ciphertext, for some receiver j such that $P(i_0, j) = P(i_1, j) = 1$ which satisfies the original condition $P(i_0, j) = P(i_1, j)$, the adversary can easily get different encryption keys to distinguish the challenge ciphertexts with the help of the receiver j . Although the security model cannot exclude this trivial case which is noted in [6], it should be easily remedied when consider the security proof for multiple users.

Definition 5. (**No-Write Rule**) Consider the no-write rule game of Table 2 between a challenger C and a stateful adversary A. Let Q_S (resp. Q) be the set of all queries $q = (j, t)$ that A issue to $\mathcal{O}_S$ (resp. both $\mathcal{O}_S$ and $\mathcal{O}_R$). Let I_S be the set of all $i \in [n]$ such that $(i, sen) \in Q_S$ and let J be the set of all $j \in [n]$ such that $(j, rec) \in Q$. Then we say that A wins the no-write rule game if $b' = b$ and all of the following hold:
1.$(n+1, san) \notin Q$;
2.$i' \in I_S \cup \{0\}$;
3.$\forall i \in I_S, j \in J, P(i, j) = 0$;
We say an ACE scheme satisfies the no-write rule if for all PPT A

$$\mathrm{Adv}^A = 2 \cdot |\Pr[\text{A wins the No-Write game}] - \frac{1}{2}| \leq \mathrm{negl}(\lambda).$$

Remark 1: In the no-write rule, the target ciphertext c in (c, i') either is chosen uniformly from ciphertext space $\mathcal{C}$ or generated by legal encryption key queried for oracle $\mathcal{O}_S$.

Remark 2: In the no-write rule, the sanitizer must be trusted.

The no-write rule definition means that the one $(i' \in I_S \cup \{0\})$ who has no rights $(\forall i \in I_S, j \in J, P(i, j) = 0)$ to send the message to some receivers $j \in J$, he or she cannot send any valuable information to these receivers. Note that no matter in the no-read rule or the no-write rule game, if the sender has no encryption key (ek), he or she can only choose a random ciphertext from the ciphertext space $\mathcal{C}$.

3.2 GSW Scheme

Before giving the construction of ACE scheme, we first recap GSW scheme which serves as the building block of our construction. GSW scheme is very pellucid. Its homomorphic addition and multiplication are just matrix addition and multiplication which enjoys a superior error control. It needs some operations such as bit decomposition (BitDecomp) and the inverse of the bit decomposition ($\mathrm{BitDecomp}^{-1}$) and the combination (Flatten) of the both operations and so on. We just give the definition of these operations and the detailed presentation are referred to [9].

Definition 6. Let k be a positive integer, q be a prime modulus, $\ell = \lfloor \log q \rfloor + 1$ and $N = k \cdot \ell$. Let $\mathbf{a}$ be a vector of dimension k over $\mathbb{Z}_q$ and $\mathbf{a}'$ be a vector of dimension N over $\{0,1\}$, denoted as $(a_{1,0}, \ldots, a_{1,\ell-1}, \ldots, a_{k,0}, \ldots, a_{k,\ell-1})$.

$$\mathrm{BitDecomp}(\mathbf{a}) = (a_{1,0}, \ldots, a_{1,\ell-1}, \ldots, a_{k,0}, \ldots, a_{k,\ell-1})$$

where $a_{i,j}$ is the j-th bit in a_i's binary representation.

$$\mathrm{BitDecomp}^{-1}(\mathbf{a}') = (\sum_{j=0}^{\ell-1} 2^j \cdot a_{1,j}, \ldots, \sum_{j=0}^{\ell-1} 2^j \cdot a_{k,j}).$$

Note that $a_{i,j}$ is not necessary in 0/1 and the computation is over $\mathbb{Z}_q$.

$$\mathrm{Flatten}(\mathbf{a}') = \mathrm{BitDecomp}(\mathrm{BitDecomp}^{-1}(\mathbf{a}')).$$

$$\mathrm{Powerof2}(\mathbf{a}) = (a_1, 2a_1, \ldots, 2^{\ell-1}a_1, \ldots, a_k, 2a_k, \ldots, 2^{\ell-1}a_k).$$

In order to control the error growth during the homomorphic encryption, the message space of the GSW scheme prefers to be $\{0,1\}$ and the circuits are transferred to NAND gate circuits. In our construction, the message space can be $\mathbb{Z}_q$ because we just need one time homomorphic addition and one time homomorphic scalar-multiplication. We know that if the modulus q is not in a form of a power of two and the message is not in $\{0,1\}$, the decryption will be a little complex and need the nearest plane algorithm described in [11]. In the construction of GSW scheme, we just give the special case of the message space of $\{0,1\}$.

Definition 7. (the GSW scheme [9])

Table 2: No-Write Rule Game

Game Definition	Oracle Definition
1. $(pp, msk) \leftarrow \mathbf{Setup}(1^\lambda, P)$;	$\mathcal{O}_S(j,t)$:
2. $(c, i') \leftarrow A^{\mathcal{O}_E(\cdot),\mathcal{O}_S(\cdot)}(pp)$;	1. Output $k \leftarrow \mathrm{Gen}(msk, j, t)$;
3. $ek_{i'} \leftarrow \mathrm{Gen}(msk, i', sen)$;	
4. $rk \leftarrow \mathrm{Gen}(msk, n+1, san)$;	$\mathcal{O}_R(j,t)$:
5. $r \leftarrow \mathcal{M}$	1. Output $k \leftarrow \mathrm{Gen}(msk, j, t)$;
6. $b \leftarrow \{0,1\}$;	
	$\mathcal{O}_E(i,m)$:
- if $b = 0, c' \leftarrow \mathbf{San}(rk, \mathbf{Enc}(ek_{i'}, r))$;	1. $ek_i \leftarrow \mathrm{Gen}(msk, i, sen)$;
- if $b = 1, c' \leftarrow \mathbf{San}(rk, c)$;	2. $c \leftarrow \mathbf{Enc}(ek_i, m)$;
	3. Output $c' \leftarrow \mathbf{San}(rk, c)$;
7. $b' \leftarrow A^{\mathcal{O}_E(\cdot),\mathcal{O}_R(\cdot)}(c')$	

- Setup$(1^\lambda, 1^L)$: Choose a modulus q of $\kappa = \kappa(\lambda, L)$ bits, lattice dimension parameter $n = n(\lambda, L)$, and the error distribution $\chi = \chi(\lambda, L)$ appropriately for DLWE that achieves at least 2^λ security against known attacks. Also, choose parameter $m = m(\lambda, L) = O(n \log q)$. Let $params = (n, q, \chi, m)$, $\ell = \lfloor \log q \rfloor + 1$ and $N = (n+1) \cdot \ell$.

- SecretKeyGen($params$): Sample $\mathbf{t} \leftarrow \mathbb{Z}_q^n$. Output $sk = \mathbf{s} \leftarrow (1, -t_1, \ldots, -t_n) \in \mathbb{Z}_q^{n+1}$.

- PublicKeyGen($param, sk$): Generate a matrix $\mathbf{B} \leftarrow \mathbb{Z}_q^{m \times n}$ uniformly and a vector $\mathbf{e} \leftarrow \chi^m$. Set $\mathbf{b} = \mathbf{B} \cdot \mathbf{t} + \mathbf{e}$. Set $\mathbf{A} = (\mathbf{b}|\mathbf{B})$. Set the public key $pk = \mathbf{A}$.(Remark: Observe that $\mathbf{A} \cdot \mathbf{s} = \mathbf{e}$.)

- Enc($params, pk, \mu$): To encrypt a message $\mu \in \mathbb{Z}_q$, sample a uniform matrix $\mathbf{R} \in \{0,1\}^{N \times m}$ and output the ciphertext $\mathbf{C}$ given bellow.

$$C = \mathrm{Flatten}(\mu \cdot \mathbf{I}_N + \mathrm{BitDecomp}(\mathbf{R} \cdot \mathbf{A})) \in \mathbb{Z}_q^{N \times N}.$$

- Dec($params, sk, \mathbf{C}$): Let $\mathbf{v} = \mathrm{Powerof2}(\mathbf{s})$. Observe that the first ℓ coefficients of $\mathbf{v}$ are $(1, 2, \ldots, 2^{\ell-1})$. Among these coefficients, let $v_i = 2^i$ be in $(q/4, q/2]$. Let $\mathbf{C}_i$ be the ith row of $\mathbf{C}$. Compute $x_i \leftarrow \langle \mathbf{C}_i, \mathbf{v} \rangle$. Output $\mu' = \lfloor x_i / v_i \rceil$. This method only outputs one bit message and for $\mu \in \mathbb{Z}_q$, we need the nearest plane algorithm [11].

We make a simple analysis on correctness, security and homomorphic properties of the GSW scheme for completeness. For correctness, we know that

$$\mathbf{C} \cdot \mathbf{v} = \mu \cdot \mathbf{v} + \mathbf{R} \cdot \mathbf{A} \cdot \mathbf{s} = \mu \cdot \mathbf{v} + \mathbf{R} \cdot \mathbf{e}.$$

Then we have $x_i = \mu \cdot v_i + \langle \mathbf{R}_i, \mathbf{e} \rangle$. The error $\langle \mathbf{R}_i, \mathbf{e} \rangle$ is at most $\| \mathbf{e} \|_1$. If the error e' in $x_i = \mu \cdot v_i + e'$ bounded by $q/8$, and $v_i \in (q/4, q/2]$, we have $(q/8)/v_i < 1/2$. After rounding in Dec, we attain the correct message μ. For the security, we use the following lemma in [9].

LEMMA 2. *(Lemma 1 [9]) Let $params = (n, q, \chi, m)$ be such that the $DLWE_{n,m,q,\chi}$ assumption holds. Then, for $m = O(n \log q)$ and $\mathbf{A}, \mathbf{R}$ as generated above, the joint distribution $(\mathbf{A}, \mathbf{R} \cdot \mathbf{A})$ is computationally indistinguishable from the uniform over $\mathbb{Z}_q^{m \times (n+1)} \times \mathbb{Z}_q^{N \times (n+1)}$.*

In [9], they take $m > 2n \log q$. In order to use Lemma 1, which is necessary in the proof of the no-write rule of ACE, we choose $m = 2(n+1) \log q$. For the construction of ACE scheme, we need the homomorphic addition and scalar-multiplication properties of GSW scheme. The homomorphic addition of $\mathbf{C}_1, \mathbf{C}_2$ is just $\mathbf{C}_{add} = \mathrm{Add}(\mathbf{C}_1, \mathbf{C}_2) = \mathrm{Flatten}(\mathbf{C}_1 + \mathbf{C}_2)$ and the growth factor of the errors in $\mathbf{C}_{add}$ is just 2. The scalar-multiplication can be computed by a sequence of addition. However, if so, the error growth factor is obviously the constant, which is bad when the constant $\alpha \in \mathbb{Z}_q$. Fortunately, the error growth factor of scalar-multiplication of GSW scheme is a fixed constant and independent from α. The following is the analysis.

Definition 8. (MultiConst($\mathbf{C}, \alpha$) from [9]): To multiply a ciphertext $\mathbf{C} \in \mathbb{Z}_q^{N \times N}$ by known constant $\alpha \in \mathbb{Z}_q$, set $\mathbf{M}_\alpha \leftarrow \mathrm{Flatten}(\alpha \cdot \mathbf{I}_N)$ and output $\mathrm{Flatten}(\mathbf{M}_\alpha \cdot \mathbf{C})$. Observe that:

$$\begin{aligned} \mathrm{MultiConst}(\mathbf{C}, \alpha) \cdot \mathbf{v} &= \mathbf{M}_\alpha \cdot (\mu \cdot \mathbf{v} + \mathbf{e}) \\ &= \alpha \cdot \mu \cdot \mathbf{v} + \mathbf{M}_\alpha \cdot \mathbf{e} \end{aligned} \tag{3}$$

By the definition of Flatten, $\mathbf{M}_\alpha$ is a $N \times N$ matrix with coefficients in $\{0,1\}$. So the error growth factor of MultiConst is just N which is independent from α.

4. THE CONSTRUCTION OF ACE

We will follow the strategy of [6] to construct ACE. Firstly, we will construct a ACE for one identity. Then, through a simple parallel arrangement, we can construct a general ACE for many identities. However, there is a same flaw as [6]. In this extension from one identity to many identities, the complexity of the scheme has a linear relation with the number of identities. It is also the other open problem in [6] that whether there is a construction of the general ACE based on the standard assumption with polylog relation with the number of identities. We also put it as the future work. We denote the ACE scheme for one identity as 1-ACE.

4.1 The Construction of 1-ACE

Let 1-ACE = (**Setup**, **KeyGen**, **Enc**, **San**, **Dec**) be the access control encryption scheme for one identity. We denote the algorithms of GSW scheme as (**GSW.Setup**, **GSW.SKGen**, **GSW.PKGen**, **GSW.Enc**, **GSW.Dec**).

- **Setup**(1^λ) Input the security parameter 1^λ. Invoke **GSW.Setup**$(1^\lambda, 1)$ to attain the parameters $params$ for DLWE assumption. Let $\beta \leftarrow \mathbb{Z}_q$ be a uniform random element. Invoke $\mathbf{s} \leftarrow$ **GSW.SKGen**$(params)$ to

attain the decryption secret key. Then invoke $\mathbf{A} \leftarrow \mathbf{GSW.PKGen}(params, \mathbf{s})$ to attain another common parameter. Output $pp = (params, A)$ and $msk = (\mathbf{s}, \beta)$. The message space is $\mathcal{M} = \mathbb{Z}_q$ and the ciphertext spaces are $\mathcal{C} = \{0,1\}^{N\times N} \times \{0,1\}^{N\times N}$ and $\mathcal{C}' = \{0,1\}^{N\times N}$.

- **KeyGen**(msk): Input the master secret key msk. Output the encryption secret key ek, decryption secret dk and sanitizer key rk as follows:
  - $ek = \beta$;
  - $rk = -\beta$;
  - $dk = \mathbf{s}$;

- **Enc**$(pp, ek, \mu \in \mathbb{Z}_q)$: Input the public parameters pp, the encryption key ek, and the message μ. Invoke $\mathbf{C}_1 \leftarrow \mathbf{GSW.Enc}(\mathbf{A}, ek)$ and $\mathbf{C}_2 \leftarrow \mathbf{GSW.Enc}(\mathbf{A}, \mu)$ to get two ciphertexts. Output
$$\mathbf{C} = (\mathbf{C}_1, \mathbf{C}_2).$$

- **San**$(pp, rk, \mathbf{C})$: Input the public parameters pp, the sanitizer key rk and the ciphertext $\mathbf{C} \in \mathcal{C}$, Choose a random $r \in \mathbb{Z}_q$ and invoke $\mathbf{C}_3 \leftarrow \mathbf{GSW.Enc}(\mathbf{A}, rk)$. Compute $\mathbf{C}_3^{'} \leftarrow \text{MultiConst}(\text{Add}(\mathbf{C}_3, \mathbf{C}_1), r)$. Output
$$\mathbf{C}' = \text{Add}(\mathbf{C}_3^{'}, \mathbf{C}_2).$$

- **Dec**$(\mathbf{s}, \mathbf{C}')$: Input the decryption secret key $\mathbf{s}$ and the ciphertext $\mathbf{C}'$. Invoke $\mu' \leftarrow \mathbf{GSW.Dec}(\mathbf{s}, \mathbf{C}')$ and output μ'.

The correctness and the security of the scheme mainly depend on GSW scheme. The following two theorems are the main results about our 1-ACE scheme based on LWE.

THEOREM 2. *(Correctness) Let N, m be the parameters defined in the scheme. And assume that the error distribution χ is a B-bounded distribution. Let q be the prime modulus with super-polynomial magnitude. For the valid ciphertext, if $3NmB < \frac{q}{8}$, it can be decrypted correctly.*

PROOF. We can see that the final ciphertext output by sanitizer is

$$\mathbf{C}' = \text{Flatten}(\text{Flatten}(\mathbf{M}_r \cdot \text{Flatten}(\mathbf{C}_1 + \mathbf{C}_3)) + \mathbf{C}_2)$$

where $\mathbf{M}_r = \text{Flatten}(r \cdot \mathbf{I}_N)$. Then we have

$$\begin{aligned}\mathbf{C}' \cdot \mathbf{v} &= \mathbf{M}_r \cdot (\mathbf{C}_1 + \mathbf{C}_3) \cdot \mathbf{v} + \mathbf{C}_2 \cdot \mathbf{v} \\ &= \mathbf{M}_r \cdot ((ek + rk)\mathbf{v} + (\mathbf{R}_1 + \mathbf{R}_3) \cdot \mathbf{e}) + \mu\mathbf{v} + \mathbf{R}_2 \cdot \mathbf{e} \\ &= \mu\mathbf{v} + r(ek + rk)\mathbf{v} + (\mathbf{M}_r \cdot (\mathbf{R}_1 + \mathbf{R}_3) + \mathbf{R}_2) \cdot \mathbf{e}\end{aligned} \tag{4}$$

where $\mathbf{v} = \text{Powerof2}(\mathbf{s})$, $\mathbf{R}_i \in \{0,1\}^{N\times m}$ for $i \in \{1,2,3\}$ and $\mathbf{e} \leftarrow \chi^m$. For the valid ciphertext, we have $r(ek + rk) = 0$. For the error

$$\begin{aligned}&||(\mathbf{M}_r \cdot (\mathbf{R}_1 + \mathbf{R}_3) + \mathbf{R}_2) \cdot \mathbf{e}||_\infty \\ &\leq ||\mathbf{M}_r||_1||(\mathbf{R}_1 + \mathbf{R}_3) \cdot \mathbf{e}||_\infty + ||\mathbf{R}_2 \cdot \mathbf{e}||_\infty \\ &\leq 3N||\mathbf{R}_1||_1||\mathbf{e}||_\infty \\ &\leq 3NmB < \frac{q}{8}\end{aligned} \tag{5}$$

by the correctness of GSW scheme, we attain the correctness of 1-ACE scheme. □

THEOREM 3. *(Security) Let q be super-polynomial. And the parameters defined in the **Setup** make $DLWE_{n,m,q,\chi}$ assumption holding. We choose $m = 2(n+1)\log q$. Then 1-ACE scheme above satisfies the **No-Read Rule** and the **No-Write Rule**.*

Before the formal proof, we make a simple analysis. We mainly use the indistinguishability of the ciphertexts generated by the sender or sanitizer to prove the security of our ACE scheme. In the security model, the oracle queries represent the collusion attack in the system. In the proof, we can simulate the whole system. Therefore, we can answer any legal oracle query correctly. In a word, we want to prove that the challenge ciphertexts are indistinguishable even given some parties' secret keys which satisfy the condition requirements.

PROOF. Our scheme is 1-ACE scheme, which means that there are only one sender, one sanitizer and one receiver. And the policy is P(1,1)=1. We firstly prove the **No-Read Rule**. So we need to prove that the scheme satisfies the payload privacy and the sender anonymity.

1. **Payload Privacy**. We know, in the payload privacy, the condition for decryption key oracle is $P(i_0, j) = P(i_1, j) = 0$ where (i_0, i_1) are the identities of the senders of attack target and j is the identities for the secret key query. And $(i_0, i_1) \in \{0,1\} \times \{0,1\}$. According to the value of (i_0, i_1), we divide into three cases to prove.

case 1: $(i_0, i_1) = (0,0)$**.** By the condition, we know that $j \in \{0,1,2\}$ and j can be the sender, receiver and sanitizer $(j = 2)$. By the rule for 0 identity sender, the output of
$$\mathbf{Enc}(\text{Gen}(msk, i_b, Sen), m_b)$$
is uniformly random over the ciphertext space $\mathcal{C}$, where $b \in \{0,1\}$. It is independent from m_0, m_1. We have $\text{Adv}^A = 0$.

case 2: $(i_0, i_1) = (1,1)$**.** By the condition, $j \in \{0,2\}$. So the adversary A has at most a sanitizer's secret key rk. The output of
$$\mathbf{Enc}(\text{Gen}(msk, i_b, Sen), m_b)$$
is $(\text{GSW.Ecn}(ek), \text{GSW.Enc}(m_b))$, where ek is the encryption key of sender 1. By the indistinguishability of GSW ciphertext, we attain $\text{Adv}^A \leq \text{nelg}(\lambda)$.

case 3: $i_0 \neq i_1$**.** W.l.o.g let $i_0 = 0$ and $i_1 = 1$. By the condition, $j \in \{0,2\}$. So the adversary has at most a sanitizer's secret key rk. For $b = 0$, the output of
$$\mathbf{Enc}(\text{Gen}(msk, 0, Sen), m_0)$$
is a uniformly random ciphertext denoted as $(\mathbf{C}_1^0, \mathbf{C}_2^0)$ over $\{0,1\}^{N\times N} \times \{0,1\}^{N\times N}$. For $b = 1$, the output of
$$\mathbf{Enc}(\text{Gen}(msk, 1, Sen), m_1)$$
is $(\text{GSW.Ecn}(ek), \text{GSW.Enc}(m_1)) \in \{0,1\}^{N\times N} \times \{0,1\}^{N\times N}$ denoted $(\mathbf{C}_1^1, \mathbf{C}_2^1)$. By Lemma 2, we have $(\mathbf{C}_1^0, \mathbf{C}_2^0)$ is computational indistinguishable from $(\mathbf{C}_1^1, \mathbf{C}_2^1)$. Therefore, $\text{Adv}^A \leq \text{negl}(\lambda)$.

From above, the scheme satisfies **Payload Privacy**. Next we prove the sender anonymity.

2. **Sender Anonymity**. In the sender anonymity, we know the conditions are $P(i_0, j) = P(i_1, j)$ and $m_0 = m_1$. Also according to $(i_0, i_1) \in \{0,1\} \times \{0,1\}$, we divide into three cases to prove.

case 1: $(i_0, i_1) = (0,0)$**.** By the conditions, we have $j \in \{0,1,2\}$. This case is the same as it in the payload privacy. So $\text{Adv}^A = 0$.

case 2: $(i_0, i_1) = (1, 1)$**.** By the conditions, we have $j \in \{0, 1, 2\}$. When $j = 1$, the adversary can decrypt the challenge ciphertexts completely. However, the encryption key ek and the message m in the challenge ciphertexts are completely identical. Obviously, the ciphertexts

$$\mathbf{Enc}(\mathrm{Gen}(msk, i_b, Sen), m_b)$$

for $b = 0$ and $b = 1$ are identical. Therefore $\mathrm{Adv}^A = 0$.

case 3: $i_0 \neq i_1$**.** W.l.o.g let $i_0 = 0$ and $i_1 = 1$. By the condition, we have $j \in \{0, 2\}$. It is also identical to the case 3 in the payload privacy. So $\mathrm{Adv}^A \leq \mathrm{negl}(\lambda)$. Note that for this case, when consider the multiple users, we need to exclude the trivial case.

From the above three cases, we prove that 1-ACE scheme satisfies the sender anonymity. Combining the proof of the payload privacy and the sender anonymity, we complete the proof of the no-read rule.

For the **No-Write Rule**, by the conditions in the security model, we have that $(2, San) \notin Q$ (condition 1) where Q is all the secret key queries. In our 1-ACE scheme, the identities of the sender are just 0 or 1. So $I_S = \emptyset$ or $I_S = \{1\}$, where I_S is the set of the identities of the true sender ($i \in [n]$) in the secret key query before giving the attack target (c, i'). According to I_S, we divide into two cases to prove.

case 1: $I_S = \{1\}$**.** By condition 2 ($i' \in I_S \cup \{0\}$), $i' \in \{0, 1\}$. By condition 3 ($\forall i \in I_S, \forall j \in J, P(i, j) = 0$), where J is the set of identities of receivers in the decryption key query in the whole attack, we have $j = 0$. Note that for a GSW ciphertext

$$\mathbf{C} = \mathrm{Flatten}(\mathrm{BitDecomp}(\mathbf{R} \cdot \mathbf{A}) + \mu \mathbf{I}_N) \in \{0, 1\}^{N \times N},$$

we attain

$$\mathbf{C}^* = \mathrm{BitDecomp}^{-1}(\mathbf{C}) = \mathbf{R} \cdot \mathbf{A} + \mu \mathbf{G} \in \mathbb{Z}_q^{N \times (n+1)},$$

which is unique determined by $\mathbf{C}$, and vice versa. The matrix $\mathbf{G} = \mathrm{BitDecomp}^{-1}(\mathbf{I}_N)$ is the primitive matrix using in [11]. For the concision, we use the form of $\mathbf{C}^*$ to denote the GSW ciphertext in the following proof.

In this case, no matter the attack target ciphertext $c \in (c, i')$ is chosen uniformly from $\mathbb{Z}_q^{N \times (n+1)} \times \mathbb{Z}_q^{N \times (n+1)}$ or generated by the encryption key ek of the sender 1. For any $b \in \{0, 1\}$, we have the challenge ciphertext $c' = r(\mathbf{C}_1 + \mathbf{C}_3) + \mathbf{C}_2$. Where $r \in \mathbb{Z}_q$ is chose randomly by the sanitizer and $(\mathbf{C}_1, \mathbf{C}_2)$ is uniformly from $\mathbb{Z}_q^{N \times (n+1)} \times \mathbb{Z}_q^{N \times (n+1)}$ or generated by the encryption key ek of the sender 1 and $\mathbf{C}_3 = rk\mathbf{G} + \mathbf{R}_3\mathbf{A}$ is the GSW ciphertext of the sanitizer secret key rk. By Lemma 2, $\mathbf{C}_3$ is indistinguishable from the uniform distribution over $\mathbb{Z}_q^{N \times (n+1)}$ except with negligible probability, which implies that c' is also indistinguishable from the uniform distribution over $\mathbb{Z}_q^{N \times (n+1)}$ except with negligible probability. Therefore, the challenge ciphertext c' under $b = 0$ or $b = 1$ is indistinguishable except with a negligible probability. Therefore we have $\mathrm{Adv}^A \leq \mathrm{negl}(\lambda)$.

case 2: $I_S = \emptyset$**.** By condition 2, $i' = 0$. By condition 3, $j \in \{0, 1\}$. If $j = 0$, it is the special case in $I_S = \{1\}$. If $j = 1$, the adversary A at most can get the decryption key dk of the receiver 1 and get the encryption key ek of the sender 1 after the challenge ciphertext. Because the adversary has a decryption key dk, we cannot determine the distribution of the attack target ciphertext $c \in \{c, i'\}$ (the input ciphertext for $b = 1$). For $b = 0$, the input ciphertext $\mathbf{Enc}(ek_0, r)$ is chosen uniformly from GSW ciphertext space.

In this case, it is sufficient to prove that the output of San is independent from its input ciphertext except with negligible probability. So for $b = 0$ or $b = 1$, the output of San has the same distribution except with negligible probability.

We assume that the input ciphertext of San is as following:

$$\mathbf{C}_1 = \mathbf{R}_1 \cdot \mathbf{A} + \beta' \cdot \mathbf{G}$$

$$\mathbf{C}_2 = \mathbf{R}_2 \cdot \mathbf{A} + \mu \cdot \mathbf{G}$$

Where β' is the encryption key and μ is the plaintext chosen by the adversary A. We will prove that there exists $\mathbf{R}_3 \in \{0, 1\}^{N \times m}$ and $r \in \mathbb{Z}_q$ to make the output of San be any ciphertext of any message in $\mathbb{Z}_q$. Then the decryption key dk of A is invalid. The output of San is as follows:

$$r \cdot (\mathbf{R}_1 \cdot \mathbf{A} + \mathbf{R}_3 \cdot \mathbf{A}) + \mathbf{R}_2 \cdot \mathbf{A} + (r(\beta' + \beta) + \mu) \cdot \mathbf{G},$$

where β is the secret key of the sanitizer. It sufficient to prove that there is a solution $\mathbf{R}_3 \in \{0, 1\}^{N \times m}, r \in \mathbb{Z}_q$ for any $\mathbf{D} \in \mathbb{Z}_q^{N \times (n+1)}$ and $\gamma \in \mathbb{Z}_q$ such that

$$\begin{cases} r \cdot (\mathbf{R}_1 \cdot \mathbf{A} + \mathbf{R}_3 \cdot \mathbf{A}) + \mathbf{R}_2 \cdot \mathbf{A} = \mathbf{D} \\ r(\beta' + \beta) + \mu = \gamma \end{cases}$$

If $\beta' \neq -\beta$ (happening with negligible property since the modulus is super-polynomial), then $r \in \mathbb{Z}_q$ must exist. By Lemma 1 and a appropriate chosen parameters, the rows of $\mathbf{A}$ generate $\mathbb{Z}_q^{n+1}$ except with negligible probability. Then $\mathbf{R}_3 \in \{0, 1\}^{N \times m}$ must exist. Then we complete the proof of case 2. Combining the case 1 and 2, the no-write rule follows.

Now we complete the proof of theorem 3. □

We have constructed a 1-ACE scheme for one identity i.e. a sender, a sanitizer and a receiver. By the diagram in [6], we can attain a ACE scheme for multiple identities. It is a simple parallel arrangement of n 1-ACE schemes. For $(i, j) \in [n] \times [n]$ satisfying the policy $P(i, j) = 1$, the encryption secret key of sender i includes an encryption key ek_j of the j-th 1-ACE scheme. The ciphertext of a sender is an ordered sequence of n ciphertexts of the 1-ACE scheme. For the slots where i having no encryption key, i will choose a random ciphertext from the ciphertext space. More detailed information are referred to [6].

4.2 The Parameters

From the construction of 1-ACE scheme, we see that we don't need the homomorphic multiplication. So we can choose a small modulus q. Unfortunately, $\mathbb{Z}_q$ is also the encryption and sanitizer secret key space. We have to enforce q to be super-polynomial to against secret key guessing attack. For correctness, we just need $3NmB < \frac{q}{8}$, which means q/B can be only polynomial about the dimension n of $\mathrm{DLWE}_{n,m,q,\chi}$. It implies that the approximate factor in GapSVP can be only polynomial in n, which is as secure as the general public key encryption scheme based on the lattice [5]. For the security of the our scheme, we just need the $\mathrm{DLWE}_{n,m,q,\chi}$ assumption holding. By [12], we just need the error bound $B > O(\sqrt{n})$. We choose the same lower bound $\omega(\log n)\sqrt{n}$ of B as in [9]. For the security parameter λ, we let $n = O(\lambda)$, $\log q = \omega(\log \lambda)$, $N = (n + 1) \log q = \tilde{O}(\lambda)$, $m \geq 2(n + 1) \log q = \tilde{O}(\lambda)$, and $\omega(\log n)\sqrt{n} < B < \frac{q}{24Nm}$, and B should be choose a little bigger to make q/B to be $\mathrm{poly}(n)$.

5. CONCLUSIONS

We construct an access control encryption scheme first defined in [6] based on DLWE assumption. We adopt the leveled fully homomorphic encryption scheme proposed in [9] as the building block. We first construct 1-ACE scheme for one identity, then use a simple parallel arrangement of n 1-ACE schemes to construct a ACE scheme for multiple identities. We need a super-polynomial modulus for the security which decreases the performance of the scheme. We try to extend the scheme to Ring-LWE assumption [10], which may decrease the magnitude of modulus and improve the performance. However, Lemma 1 which is important to prove the no-write rule doesn't hold for Ring-LWE. This can be a future work to be settled.

6. REFERENCES

[1] J. Alperin-Sheriff and C. Peikert. Faster bootstrapping with polynomial error. In *CRYPTO*, pages 297–314. Springer, 2014.

[2] Z. Brakerski. Fully homomorphic encryption without modulus switching from classical gapsvp. In *CRYPTO*, pages 868–886. Springer, 2012.

[3] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *ITCS*, pages 309–325. ACM, 2012.

[4] Z. Brakerski and V. Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In *FOCS*, pages 97–106. IEEE Computer Society, 2011.

[5] Z. Brakerski and V. Vaikuntanathan. Lattice-based FHE as secure as PKE. In *ITCS*, pages 1–12. ACM, 2014.

[6] I. Damgård, H. Haagh, and C. Orlandi. Access control encryption: Enforcing information flow with cryptography. In *TCC*, pages 547–576. ACM, 2016.

[7] G. Fuchsbauer, R. Gay, L. Kowalczyk, and C. Orlandi. Access control encryption for equality, comparison, and more. In *http://eprint.iacr.org/2017/009.pdf*.

[8] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 197–206. ACM, 2008.

[9] C. Gentry, A. Sahai, and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *CRYPTO*, pages 75–92. Springer, 2013.

[10] V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. In *EUROCRYPT*, pages 1–23. Springer, 2010.

[11] D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT*, pages 700–718. Springer, 2012.

[12] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, pages 84–93. ACM, 2005.