Fully Homomorphic Distributed Identity-based Encryption Resilient to Continual Auxiliary Input Leakage

François Gérard, Veronika Kuchta, Rajeev Anand Sahu, Gaurav Sharma and Olivier Markowitch
Université Libre de Bruxelles, Belgium

Keywords: Homomorphic Encryption, Identity-based Encryption, Leakage Resilient Cryptography, LWE.

Abstract: History tells us that is not enough to base security solely on the unfeasability of solving the underlying hard problem of a cryptosystem. In the real world, it is not uncommon for an adversary to get access to some key dependent information potentially helping to perform cryptanalysis. Recently a lot of effort has been put into designing cryptosystems such that the impact of leaking key related information is minimized, this area is mostly known as *leakage-resilient cryptography*. In this work, we show how to construct a distributed fully homomorphic identity-based encryption secure in the continual auxiliary input model. Our construction is based on the fully homomorphic scheme of Gentry, Sahai and Waters and relies merely on the learning with errors assumption, which is conjectured being resistant against quantum attacks.

1 INTRODUCTION AND MOTIVATION

Leakage-Resilient Cryptography. Security of traditional public-key cryptographic schemes depends on privacy of secret keys and can be analyzed in an idealized model under the assumption that the secret keys are hidden from adversary. Nevertheless many schemes become insecure during their implementation into real systems. An adversary can often learn auxiliary information on the secret inputs of the algorithm (for example if the key is used somewhere else or by studying the physical behavior of the device performing cryptographic operations). Such attacks are known as side-channel attacks. A solution for this problem can be provided by what is called leakage resilient cryptography which guarantees security even if we assume secret key leakage during implementation procedure (Akavia et al., 2009).

An example of strong side-channel attack is the so-called ”cold-boot attack” that was defined recently (Halderman et al., 2009). Due to the fact that every cryptographic algorithm is made to be eventually used in a real environment, side-channel attacks often lead to loss of secrecy, during the implementation which enables observations like the amount of power consumption or the time required for this implementation. These observations lead to information leakage about secret-keys without breaking the underlying assumptions of the considered schemes. Those side-channel attacks which include all attacks in which leakage of information is possible when while the scheme performs any computations, are called computational side-channel attacks as showed by Micali and Reyzin (Micali and Reyzin, 2004). But not only computation on secrets leak information. Akavia et al. (Akavia et al., 2009) considered another family of side-channel attacks, the so called ”memory attack”, which is a generalization of the already mentioned ”cold-boot attack” introduced by Halderman et al. (Halderman et al., 2009). Akavia et al.'s work defined the family of memory attacks by allowing leakage of a bounded number of bits of the secret, which are computed upon applying an arbitrary function with output that is bounded by the size of the secret key. This model is called the bounded leakage model indicating that the overall amount of information the attacker can learn is bounded by a finite natural number. This leads to the main question in leakage-resilient cryptography which is exploring the suitable size of the output of the leakage function without compromising the security of cryptosystem. There are new results on public-key encryption to provide security against memory attacks. First one looks for redundant representation of secret-keys which can enable the battling memory attack. The other approach is just to consider the already existing cryptosystems and to check their consistency against memory attacks. Akavia et al. (Akavia et al., 2009) took the second approach, ex-

Gérard, F., Kuchta, V., Sahu, R., Sharma, G. and Markowitch, O.
Fully Homomorphic Distributed Identity-based Encryption Resilient to Continual Auxiliary Input Leakage.
DOI: 10.5220/0006832200410052
In *Proceedings of the 15th International Joint Conference on e-Business and Telecommunications (ICETE 2018) - Volume 2: SECRYPT*, pages 41-52
ISBN: 978-989-758-319-3

amined the learning with errors (LWE) problem and proved semantic security against memory attack of an LWE-based public-key encryption. The strength of the LWE assumption depends on the size of the leakage one would like to tolerate.

The variety of side-channel attacks leads to the conclusion that information can every time leak the cryptographic device while it performs certain computations. (Brakerski et al., 2010) presented new constructions of encryption schemes and identity-based encryption which remain secure while information of the secret key is leaked. Their constructions guarantees leakage-resilience of the secret key even if an adversary can test the memory and leak a proper fraction of this key. However their construction cannot tolerate leakage from the master secret key. Therefore the authors left the problem open of finding an IBE scheme that is resilient to the leakage of the master secret key. The problem with a model in which parameters are chosen according to the total leakage of the system during its lifetime is that it can lead to huge key size. Brakerski et al. (Brakerski et al., 2010) introduced a new model where the secret key can be refreshed during different time periods while the public key remains the same. The previously mentioned drawback is reduced by the fact that the leakage bound exists only between refresh phases. This leakage model is known as the continual leakage model. There are also scenarios where information leaks from the memory even if there is no computation taking place during the attack. In order to make public key encryption secure against leakage attacks the idea was not to store the complete secret memory on the device but to add some auxiliary device. Another attack resulting from memory leakages forms a class of ”auxiliary input attacks”, where the adversary chooses an efficiently computable leakage auxiliary function which is hard to invert in polynomial time. Dodis et al. (Dodis et al., 2010a) provided the model of such auxiliary leakage functions in case of public-key encryptions. Analogously a symmetric encryption scheme was introduced by Dodis et al. (Dodis et al., 2009). The first IBE scheme resilient to continual auxiliary leakage has been proposed by Yuen et al. (Yuen et al., 2012), meaning that the identity-based scheme remains secure even if the adversary has access to some auxiliary input, where the auxiliary input is modeled by an uninvertible function of the secret key. The auxiliary input model represents a more general model where the secret key cannot be recovered. Another differentiation between the models is given by the possibility of an adversary to see the public key either before choosing a leakage function or afterward. The first case describes an adaptive model where the adversary can adaptively choose a leakage function after she has seen the public key. The latter model is called non-adaptive model, where the leakage function has to be chosen by an adversary before seeing the public key. Over the last 17 years, leakage-resilience has become a popular research topic which can be reflected in the following research articles (Alwen et al., 2010; Canetti et al., 2000; Dodis et al., 2009; Dziembowski and Pietrzak, 2008; Micali and Reyzin, 2004; Naor and Segev, 2009; Pietrzak, 2009; Chow et al., 2010).

Fully-homomorphic Encryption. This property of cryptographic encryption schemes became one of the most fascinating research topics of modern cryptography. It allows users to perform computations on encrypted data without decrypting it in advance. Even though there were earlier attempts for a homomorphic encryption (Rivest et al., 1978), the real breakthrough came with work by Gentry (Gentry, 2009b) who introduced the first fully homomorphic encryption scheme based on a cryptographic assumption using the well-known mathematical construct called ideal lattices. Some other fully homomorphic encryption schemes which are not based on lattices but relied on ideals in rings were presented in (Smart and Vercauteren, 2010; Brakerski and Vaikuntanathan, 2011b; van Dijk et al., 2010). Brakerski and Vaikuntanathan (Brakerski and Vaikuntanathan, 2011a) presented a fully homomorphic scheme based on a well-studied assumption - known as the learning with errors assumption (LWE). A comparatively simple fully homomorphic encryption scheme also based on LWE problem has been introduced by Gentry et al. (Gentry et al., 2013). They presented a new technique called *approximate eigenvector* method where homomorphic addition and multiplication are provided by a simple matrix addition and multiplication. In contrast to previous fully homomoprhic schemes, Gentry et al.'s construction does not require any evaluation key and evaluation can even be calculated without knowing user's public key. This feature allowed the authors to construct the first fully homomorphic identity-based encryption without usage of any evaluation keys. Berkoff and Liu (Berkoff and Liu, 2014) explored for the first time a new topic of leakage-resilient fully homomorphic encryption. They instantiated their construction by making the underlying decisional learning with errors (DLWE) problem of a fully homomorphic encryption scheme leakage resilient. They defined the scheme in adaptive bounded leakage model. Later on, Goldwasser et al. (Goldwasser et al., 2010) showed in their work that the leakage resilient DLWE involves leakage resilience of symmetric-key encryption schemes which are secure under the DLWE assumption.

Our Contribution. In this work we present the first leakage resilient IBE scheme which is defined in continual auxiliary leakage model and has the fully homomorphic property. Furthermore we enhance our construction by dividing the role of one decryption server among two servers which decrypt the ciphertext by running an interactive two party decryption protocol. In contrast to the previous construction of fully homomorphic encryption by Berkoff and Liu (Berkoff and Liu, 2014), our scheme combines the concepts of continual leakage model and auxiliary leakage model where the earlier model will be achieved by refreshing the secret key shares of each server while the latter model is achievable by the minimal restriction of the leakage function between updates. The auxiliary input model provides another appealing feature being useful for composible constructions which is an interesting question for further research. We also instantiate the first fully homomorphic IBE scheme which is secure under LWE assumption. Identity-based encryption became an engrossing research topic because of its distinctive future where any public string can be used as a public key for the encryption process. Analogously to the Berkoff and Liu construction (Berkoff and Liu, 2014) our scheme also achieves adaptivity which allows an adversary to choose the leakage function after seeing the public key. For the fully homomorphic property we assume a scenario where the different ciphertexts are computed using the same public key. In order to enhance the scheme to a scenario where different ciphertext can be computed at different times using distinct public keys, we refer to the multi-key FHE technique used in (Clear and McGoldrick, 2015) and leave the topic for our further research. Furthermore our scheme profits from its distributed decryption process, where the secret key will be shared between two decryption parties. In order to realize the distribution of the secret key we use a leakage resilient symmetric encryption based on LWE. Finally, our leakage resilient lattice-based distributed IBE scheme is secure against chosen-plaintext attacks.

Applications. Our leakage-resilient fully homomorphic IBE scheme has several applications. As mentioned shortly in the introduction, an identity-based encryption has the attractive feature of no need to managing a public key infrastructure since only recipient's identity and some public parameters are required to encrypt a message. Regarding privacy, IBE schemes are especially suitable for those systems which require anonymity of communicating parties. Having an IBE scheme, a user can choose an anonymous certificate to achieve anonymity between those parties. Fully homomorphic property is particularly useful in applications concerning cloud security. Outsourcing private data to the cloud services increases concerns about data owner's privacy. Storing the data on cloud in encrypted form and providing the homomorphic property, which allows the cloud servers to perform arbitrary computations on encrypted data, addresses the question of privacy issues. In particular, cloud services are appealing in the medical and financial sectors. Li et al. (Li et al., 2010) presented an attribute-based encryption (ABE) scheme which allows to secure personal health records in cloud computing. Sahai and Waters (Sahai and Waters, 2005) showed in their work how to generalize an IBE encryption to fuzzy identity-based encryption, which on its part can be generalized to and ABE scheme, as showed by Goyal et al. (Goyal et al., 2006). This applies that our leakage-resilient fully homomorphic distributed IBE scheme can be generalized to an ABE scheme, which on its part can be applied to secure cloud services of medical and financial sectors.

2 PRELIMINARIES

Notations. In our paper, we follow common notations used in the recent literature. A column vector $\mathbf{v}$ (with coefficients v_i) is written as a bold lowercase letter. A matrix $\mathbf{M}$ is written as a bold uppercase letter. For a value v sampled from a distribution $\mathcal{D}$, we write $v \leftarrow \mathcal{D}$. This notation is extended to vector, $\mathbf{v} \leftarrow \mathcal{D}^n$ indicates that the coefficients of $\mathbf{v}$ are sampled independently from $\mathcal{D}$. The uniform distribution over a set S is written as $\mathcal{U}(S)$. Matrix multiplication will be denoted with either juxtaposition $\mathbf{AB}$ or with a small dot $\mathbf{A} \cdot \mathbf{B}$ when it enhances readability. The dot product between two (same size) vectors $\langle \mathbf{a}, \mathbf{b} \rangle$ or $\mathbf{a}^T \mathbf{b}$ corresponds to $\sum_i a_i b_i$ with the operations performed in the algebraic structure of the vectors' coefficients.

2.1 Learning with Errors

For some positive integers parameters q,n, a vector $\mathbf{s} \leftarrow \mathcal{U}\left(\mathbb{Z}_q^n\right)$ and an error distribution $\mathcal{D}_\sigma$ with standard deviation σ over the integers (usually a discrete Gaussian), we define the LWE distribution $A_{s,\mathcal{D}_\sigma}$ which is obtained by sampling a vector $\mathbf{a} \leftarrow \mathcal{U}(\mathbb{Z}_q^n)$, an error term $e \leftarrow \mathcal{D}_\sigma$ and outputting the tuple $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e) \leftarrow \mathcal{U}\left(\mathbb{Z}_q^n \times \mathbb{Z}_q\right)$.

The LWE problem, first introduced by Regev (Regev, 2005), comes in two flavors, a search problem and a decision problem.

Definition 1 (Search-LWE$_{n,m,q,\sigma}$)**.** Given m samples $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e) \leftarrow A_{s,\mathcal{D}_\sigma}$, find $\mathbf{s}$.

This can be seen as solving an over-determined system of noisy linear equations over $\mathbb{Z}_q$. Indeed, if we write the samples as a matrix, the problem becomes: given a public matrix $\mathbf{A} \in \mathbb{Z}^{m \times n}$ and a target vector $\mathbf{t} = \mathbf{A}\mathbf{s} + \mathbf{e} \in \mathbb{Z}_q^m$, find $\mathbf{s}$.

Definition 2 (Decision-LWE$_{n,m,q,\sigma}$)**.** Given m samples over $\mathbb{Z}_q^n \times \mathbb{Z}_q$, determine if they come from $\mathcal{U}(\mathbb{Z}_q^n \times \mathbb{Z}_q)$ or $A_{s,\mathcal{D}_\sigma}$.

Stated otherwise, it is the problem of distinguishing the LWE distribution from the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$.

Surprisingly, both problems are equivalently hard, even for an adversary in possession of a quantum computer.

Definition 3 (Goldreich-Levin Theorem(Dodis et al., 2010a))**.** For a prime q, a set $H \subseteq \mathbb{Z}(q)$, a function $h : H^m \to \{0,1\}^*$, a *secret* vector $\mathbf{s} \in H^m$, a random vector $\mathbf{r} \leftarrow \mathcal{U}(\mathbb{Z}_q)^m$ if there exists a distinguisher between $\langle \mathbf{s}, \mathbf{r} \rangle$ and $\mathcal{U}(\mathbb{Z}_q)$ given $h(\mathbf{s})$, there exists an efficient inverter for h.

Definition 4 (Goldreich-Levin Theorem for LWE (Goldwasser et al., 2010))**.** Let $k > log\ q$ and $h : \{0,1\}^n \to \{0,1\}^*$ be a function that no polynomial adversary can invert with probability greater than 2^{-k}. For a super-polynomial q, a polynomial m, $0 \leq \sigma_1, \sigma_2 \leq q$ such that σ_2/σ_1 is negligible,

$$(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e}, h(\mathbf{s})) \approx_c (\mathbf{A}, \mathbf{u}, h(\mathbf{s})) \qquad (1)$$

where $\mathbf{A} \leftarrow \mathcal{U}(\mathbb{Z}_q^{m \times n}), \mathbf{s} \leftarrow \mathcal{U}(\mathbb{Z}_q^n), \mathbf{u} \leftarrow \mathcal{U}(\mathbb{Z}_q^m)$ are uniformly random and the error vector $\mathbf{e} \leftarrow \mathcal{D}_{\sigma_1}$ if the Decision-LWE$_{\ell,m,q,\sigma_2}$ assumption holds with $\ell = \frac{k - \omega(log\ n)}{log\ q}$.

This result was first introduced in (Goldwasser et al., 2010). It express the leakage-resilient property of LWE in the auxiliary input model. In it first form it required required a super-polynomial q but, later on, Alwen et al. (Alwen et al., 2013) showed that the LWE-to-LWE reduction works even for larger parameters where the modulus and modulus-to-error ratio have polynomial size.

2.2 Homomorphic Encryption

For several decades(Rivest et al., 1978), cryptographers asked themselves the following question: is it possible to perform arbitrary computation on encrypted data without revealing it? The final answer came from Gentry in 2009(Gentry, 2009a). In his thesis, he solved the open problem of constructing a cryptosystem allowing to compute an arbitrary function on encrypted data without decrypting first. Despite being still quite inefficient, this technique, called *fully homomorphic encryption*(FHE), fits really well in the context of distributed computing, or more generally, cloud applications. As surprising as it can sounds, a small device can now ask a powerful cloud to perform data analysis without altering the privacy of the data. In the following, we recall the basic definitions. A survey of the field can be found in (Armknecht et al., 2015).

A homomorphic encryption scheme is made of four algorithms $(\texttt{FHEKeyGen}, \texttt{FHEEncrypt}, \texttt{FHEDecrypt}, \texttt{Eval})$ defined as follow:

$\texttt{FHEKeyGen}(\lambda)$: On input a security parameter λ, it outputs the secret key *sk*, the public key *pk* and the evaluation key *evk*.

$\texttt{FHEEncrypt}(pk, m)$: On input the public key *pk* and a message *m*, it outputs a ciphertext *c*.

$\texttt{FHEDecrypt}(sk, c)$: On input the secret key *sk* and a ciphertext *c*, it outputs a message *m*.

$\texttt{Eval}(evk, f, (c_0, ..., c_n))$: On input the evaluation key *evk*, a function f and some ciphertexts $c_0, ..., c_n$ such that $c_i = \texttt{Encrypt}(pk, m_i)$, it outputs $c' = \texttt{Encrypt}(pk, f(m_0, ..., m_n))$.

The *correctness* requirement for homomorphic encryption scheme is the following:

$$\texttt{FHEDecrypt}(sk, \texttt{Eval}(evk, f, (c_0, ..., c_n))) \\ = f(m_0, ..., m_n).$$

Correct decryption of evaluated ciphertext is sufficient in the sense that one can always evaluate the identity function on a ciphertext without explicitly asking for the output of FHEEncrypt to be decryptable. (Brakerski and Vaikuntanathan, 2011a)

Definition 5. A fully homomorphic encryption scheme is said to be leveled if the depth of the circuit evaluating the function f is bounded by a prespecified parameter.

2.3 Gentry-Sahai-Waters FHE

Let us now describe the fully homomorphic encryption scheme of Gentry, Sahai and Waters(Gentry et al., 2013)(GSW). This scheme has the beautiful property that no evaluation key is required to perform homomorphic operations. One can directly add and multiply ciphertexts together (which is enough to describe a **NAND** gate) and get the expected result. The construction is based on LWE.

Ciphertext flattening. We start by recalling the technique which keeps ciphertexts strongly bounded. We call a ciphertext C B-strongly-bounded if its associated messages μ and the coefficients of the ciphertext C all have magnitude at most 1, while the coefficients of the error vector e all have magnitude at most B ((Gentry et al., 2013)). The technique was used to realize the first (leveled) fully homomorphic identity-based and (leveled) fully homomorphic attribute-based encryption as showed in (Gentry et al., 2013). Using transformations from (Brakerski and Vaikuntanathan, 2011a), vectors can be modified without affecting dot products.

We assume two vectors $\mathbf{a},\mathbf{b} \leftarrow \mathcal{U}\left(\mathbb{Z}_q^k\right)$ and set $l = \lfloor \log_2 q \rfloor + 1$ and $N = k \cdot l$. Let $\texttt{BitDecomp}(\mathbf{a})$ be the N-dimensional vector $(a_{1,0},\dots,a_{1,l-1},\dots,a_{k,0},\dots,a_{k,l-1})$, where $a_{i,j}$ is the j-th bit in a_i's binary representation. For some vector $\mathbf{a}' = (a_{1,0},\dots,a_{1,l-1},\dots,a_{k,0},\dots,a_{k,l-1})$, let

$$\texttt{BitDecomp}^{-1}(\mathbf{a}') = \left(\sum_{j=0}^{l-1} 2^j \cdot a_{1,j}, \dots, \sum_{j=0}^{l-1} 2^j \cdot a_{k,j} \right)$$

be the inverse of $\texttt{BitDecomp}$, which is well defined. That means even if the input is not a bit-vector, the inverse is well-defined. For an N-dimensional vector $\mathbf{a}'$, let $\texttt{Flatten}(\mathbf{a}') = \texttt{BitDecomp}(\texttt{BitDecomp}^{-1}(\mathbf{a}'))$ a N-dimensional bit vector. For a matrix $\mathbf{A}$, let $\texttt{BitDecomp}(\mathbf{A})$, $\texttt{BitDecomp}^{-1}(\mathbf{A})$, $\texttt{Flatten}(\mathbf{A})$ be applied to each row of $\mathbf{A}$. Let $\texttt{Powerof2}(\mathbf{b}) = (b_1, 2b_1, \dots, 2^{l-1}b_1, \dots, b_k, 2b_k, \dots, 2^{l-1}b_k)$. We observe the following properties for any N-dimensional $\mathbf{a}'$:

- $\langle \texttt{BitDecomp}(\mathbf{a}), \texttt{Powerof2}(\mathbf{b}) \rangle = \langle \mathbf{a}, \mathbf{b} \rangle$
- $\left\langle \mathbf{a}', \texttt{Powerof2}(\mathbf{b}) \right\rangle = \left\langle \texttt{BitDecomp}^{-1}(\mathbf{a}'), \mathbf{b} \right\rangle$
$= \left\langle \texttt{Flatten}(\mathbf{a}'), \texttt{Powerof2}(\mathbf{b}) \right\rangle.$

Encryption and Decryption. The fully homomorphic encryption (FHE) scheme from (Gentry et al., 2013) works as follows. For suitable parameters $q, n, m = O(n \log q)$ the LWE instance consists of a $m \times (n+1)$ matrix $\mathbf{A}$ such that there is a vector $\mathbf{s} \leftarrow \mathcal{U}\left(\mathbb{Z}_q^{n+1}\right)$, where the first entry is 1 and $\mathbf{e} = \mathbf{A} \cdot \mathbf{s}$ is a small error vector. We assume that $\mathbf{A}$ is public and $\mathbf{s}$ is secret. A ciphertext $\mathbf{C}$ encrypts μ if $\mathbf{C} \cdot \mathbf{v} = \mu \mathbf{v} + \mathbf{e}$, where $\mathbf{v}$ is a N-dimensional secret key. To decrypt message μ, the i-th row $\mathbf{C}_i$ is extracted from $\mathbf{C}$ and $x \leftarrow \langle \mathbf{C}_i, \mathbf{v} \rangle = \mu v_i + e_i$ computed. The vector $\mathbf{v}$ is called approximate eigenvector. Let $\mathbf{v} = \texttt{Powerof2}(\mathbf{s})$, which is a vector of dimension $N = (n+1) \cdot l$ for $l = \lfloor \log_2 q \rfloor + 1$. Consider the following property:

$$\texttt{Flatten}(\mathbf{C}) \cdot \mathbf{v} = \mathbf{C} \cdot \mathbf{v}.$$

To encrypt a message $\mu \in \mathbb{Z}_q$, a random matrix $\mathbf{R} \in \{0,1\}^{N \times m}$ is generated and $\mathbf{C} = \texttt{Flatten}(\mu \cdot \mathbf{I}_N + \texttt{BitDecomp}(\mathbf{R} \cdot \mathbf{A}))$ computed. Note that $\texttt{Flatten}$ operation does not affect the product with $\mathbf{v}$, i.e.

$$\begin{aligned} \mathbf{C} \cdot \mathbf{v} &= \mu \cdot \mathbf{v} + \texttt{BitDecomp}(\mathbf{R} \cdot \mathbf{A}) \cdot \mathbf{v} \\ &= \mu \cdot \mathbf{v} + \mathbf{R} \cdot \mathbf{A} \cdot \mathbf{s} = \mu \cdot \mathbf{v} + \mathit{small}. \end{aligned}$$

Homomorphic Properties. Homomorphism comes naturally with the definition of a ciphertext $\mathbf{C}_i$ on a message μ_i being a matrix such that $\mathbf{C}_i \cdot \mathbf{v} = \mu_i \cdot \mathbf{v} + \mathbf{e}_i$. Indeed, for addition,

$$\mathbf{C}^+ = \texttt{Add}(\mathbf{C}_1, \mathbf{C}_2) = \mathbf{C}_1 + \mathbf{C}_2$$

implies $\mathbf{C}^+ \cdot \mathbf{v} = (\mu_1 + \mu_2) \cdot \mathbf{v} + (\mathbf{e}_1 + \mathbf{e}_2)$ and for multiplication,

$$\mathbf{C}^\times = \texttt{Mult}(\mathbf{C}_1, \mathbf{C}_2) = \mathbf{C}_1 \cdot \mathbf{C}_2$$

implies $\mathbf{C}^\times \cdot \mathbf{v} = \mathbf{C}_1 \cdot (\mathbf{C}_2 \cdot \mathbf{v}) = \mathbf{C}_1 \cdot (\mu_2 \cdot \mathbf{v} + \mathbf{e}_2) = \mu_2 \cdot (\mu_1 \cdot \mathbf{v} + \mathbf{e}_1) + \mathbf{C}_1 \cdot \mathbf{e}_2 = \mu_1 \cdot \mu_2 \cdot \mathbf{v} + \mu_2 \cdot \mathbf{e}_1 + \mathbf{C}_1 \cdot \mathbf{e}_2 = \mu_1 \cdot \mu_2 \cdot \mathbf{v} + \mathit{small}$. Unfortunately, it is impossible to prevent the error term to grow while performing homomorphic operations. We refer to (Gentry et al., 2013) for more details.

2.4 Identity-based Encryption

An identity-based encryption scheme is made of four algorithms ($\texttt{Setup}$, $\texttt{Extract}$, $\texttt{Encrypt}$, $\texttt{Decrypt}$) defined as follow:

$\texttt{IBE.Setup}(\lambda)$: On input a security parameter λ it outputs a key pair $(\mathit{msk}, \mathit{mpk})$.

$\texttt{Extract}(\mathit{msk}, \mathit{ID})$: On input the master secret key msk and an identity ID, it outputs a secret key for the given ID $\mathit{sk}_{\mathit{ID}}$.

$\texttt{IBE.Encrypt}(\mathit{mpk}, \mathit{ID}, m)$: On input the master public key mpk, an identity ID and a message m, it outputs a ciphertext c_{ID} intended to ID.

$\texttt{IBE.Decrypt}(c_{\mathit{ID}}, \mathit{sk}_{\mathit{ID}})$: On input a ciphertext c_{ID} and a secret key $\mathit{sk}_{\mathit{ID}}$ for both the same ID, it outputs a message m.

For correctness to hold, it should be the case that when $\mathit{sk}_{\mathit{ID}}$ is output by the algorithm $\texttt{Extract}(\mathit{msk}, \mathit{ID})$, we have

$$\texttt{IBE.Decrypt}(\texttt{IBE.Encrypt}(\mathit{mpk}, \mathit{ID}, m), \mathit{sk}_{\mathit{ID}}) = m.$$

3 MODELING CONTINUAL KEY-LEAKAGE ATTACKS

We propose our scheme in the continual auxiliary leakage model. A cryptosystem is secure in this model if it remains secure even when an adversary is given a computationally uninvertible function on input a secret key as an auxiliary input. The scheme achieves continual leakage resistance by refreshing the secret key in each time period with a restriction that there is no polynomial time algorithm which can invert the leakage function between two time periods. The auxiliary input model was introduced by Dodis et al. (Dodis et al., 2010a), while the continual leakage model was presented in (Dodis et al., 2010b). We present a model which is defined in the chosen ciphertext security setting. In this case an adversary has access to four oracles: extraction, leakage and refresh oracle. The model is provided in the following paragraph. Let $\mathcal{A}_{ind}$ be an adversary which is playing the following experiment $\mathbf{Exp}^{\mathbf{IND-ID-CPA-b}}_{\mathbf{LR-FHIBE},\mathcal{A}_{\mathbf{ind}}}$ for a bit $b \in \{0,1\}$:

Setup: The challenger runs `Setup` algorithm and outputs mpk, msk. The adversary specifies the leakage function h. The challenger constructs storage list which consists of extracted secret keys $\mathcal{L}_{ex}$ given by $(ID, sk_{ID,1}, sk_{ID,2})$, where $sk_{ID,1}, sk_{ID,2}$ are the two secret shares.

Phase 1: Adversary $\mathcal{A}_{ind}$ queries the following oracles:

- **Extraction Oracle:** Taking as input ID, index $i \in \{1,2\}$, the oracle checks if $(ID, sk_{ID,i}, \cdot) \in \mathcal{L}_{ex}$. If so it takes the corresponding secret key $sk_{ID} = (sk_{ID,1}, sk_{ID,2})$ and gives the secret share $sk_{ID,i}$ to the adversary. Otherwise it runs the `Extract` algorithm on input master public, master secret key and an identity ID. It outputs then the computed secret key shares $sk_{ID,i}$ to the adversary and stores $(ID, sk_{ID,i}, \cdot)$ in the list $\mathcal{L}_{ex}$.
- **Leakage Oracle:** On input a PPT computable leakage function h and an identity ID the oracle returns $h(mpk, ID, sk_{ID,1}, sk_{ID,2})$.
- **Refresh Oracle:** On input an identity ID and, the oracle checks if $(ID, sk_{ID,1}, sk_{ID,2}) \in \mathcal{L}_{ex}$, if not, it sets a counter $count = 1$. The oracle generates $\{sk_{ID,i}\}_{i\in\{1,2\}} \leftarrow \texttt{Extract}(msk, ID)$, adds $(ID, sk_{ID,1}, sk_{ID,2})$ to the list, runs the `Refresh` algorithm on input $\{sk_{ID,1}, sk_{ID,2}\}$ and returns $(sk^{fresh}_{ID,1}, sk^{fresh}_{ID,2})$ to $\mathcal{A}_{ind}$. Otherwise, if $(ID, sk_{ID,1}, sk_{ID,2})$ is already in the list it runs the `Refresh` algorithm on the existing secret key shares from the list $\mathcal{L}_{ex}$ and returns $\{sk^{fresh}_{ID,i}\}_{i\in\{1,2\}}$ to $\mathcal{A}$.

Challenge: $\mathcal{A}_{ind}$ outputs m_0, m_1 and an identity ID^*. The challenger computes $\texttt{FHIB.Encrypt}(mpk, m_b, ID^*)$, where ID^* has not been queried before to the leakage and refresh oracles.

Phase 2: The adversary is allowed to issue further extraction oracle queries excluding the queries on input ID^*.

Output: $\mathcal{A}_{ind}$ returns a bit b' as a guess for b.

We say that an identity-based encryption is secure against chosen-plaintext attacks in the continual auxiliary input model if for any probabilistic polynomial time attacker $\mathcal{A}$ running the above experiment the advantage to win, defined as

$$\mathbf{Adv}^{\mathbf{IND-ID-CPA}}_{\mathbf{LR-FHIBE},\mathcal{A}_{\mathbf{ind}}} = |\mathbf{Pr}[\mathbf{Exp}^{\mathbf{IND-ID-CPA-0}}_{\mathbf{LR-FHIBE},\mathcal{A}_{\mathbf{ind}}} = \mathbf{1}] \\ - Pr[\mathbf{Exp}^{\mathbf{IND-ID-CPA-1}}_{\mathbf{LR-FHIBE},\mathcal{A}_{\mathbf{ind}}} = \mathbf{1}]| \leq \varepsilon(\lambda),$$

where $\varepsilon(\cdot)$ is a negligible function in λ.

4 LEAKAGE-RESILIENT FULLY-HOMOMORPHIC DISTRIBUTED IDENTITY-BASED ENCRYPTION WITH AUXILIARY INPUTS (LRFHIBE)

Intuition. Motivated by the scheme of (Akavia et al., 2012) which introduced distributed public key encryption schemes secure against continual leakage, we present the new construction of a leakage-resilient fully homomorphic IBE scheme (LRFHIBE) in the distributed setting. We note that the secret key will be shared among two computing devices which can communicate with each other via a public channel. This distributed setting allows to reduce the risk of manipulating the single device, by distributing the power of a single device among two ones. Our construction guarantees security against continual leakage of the secret key where the leakage function consists of the secret key and some auxiliary input. Decryption of ciphertext is given by a two-party protocol. In order to ensure a secure communication between the two parties we use symmetric encryption during the execution of the 2-party protocol. We use the approximate eigenvector technique in order to achieve the fully homomorphic property of our scheme, avoiding the existence of any evaluation keys which makes our scheme favorable in contrast to the so far existing homomorphic encryption schemes with evaluation keys.

Definition 6 (Model for LRFHIBE Scheme)**.** Let $\hat{\mathbf{C}}$ denote the evaluated ciphertext which can be either $\hat{\mathbf{C}}^{+}$ or $\hat{\mathbf{C}}^{\times}$, as introduced in Chapter 2.3. A LRFHIBE consists of the following six algorithms:

$\texttt{FHIB.Setup}(\lambda)$**:** On input security parameter λ it generates the master public key and master secret key mpk, msk.

$\texttt{FHIB.Extract}(mpk, msk, ID)$: On input mpk, msk, ID the algorithm consists of two stages:

- **Stage 1.** Takes as input mpk, msk, ID. Outputs $\mathbf{sk}_{ID}$.
- **Stage 2.** On input secret key $\mathbf{sk}_{ID}$, it outputs secret shares $\mathbf{sk}_{ID,1}, \mathbf{sk}_{ID,2}$ for party P_1 and party P_2, respectively.

FHIB.Encrypt(mpk, ID, μ_i) : On input master public key mpk, identity ID and message μ_i it outputs a ciphertext $C_{ID}(\mu_i)$.

FHIB.Eval$(mpk, \mathbf{C}_{ID}(\mu_1), \ldots, \mathbf{C}_{ID}(\mu_n), F)$: Take as input the ciphertexts, $\mathbf{C}_{ID}(\mu_1), \ldots, \mathbf{C}_{ID}(\mu_n)$ and an evaluation function F and output the evaluated ciphertext $\hat{\mathbf{C}}$.

FHIB.Decrypt$(mpk, \hat{\mathbf{C}}, \mathbf{sk}_{ID,1}, \mathbf{sk}_{ID,2})$: On input master public key mpk, evaluated ciphertext $\hat{\mathbf{C}}$, two secret shares $\mathbf{sk}_{ID,1}, \mathbf{sk}_{ID,2}$ it runs an interactive 2-party protocol executed between P_1 and P_2 where at the end of the protocol one of the parties outputs $\hat{\mu}$, which is an evaluated value of messages $\mu_1, \ldots, \mu_n$, i.e. $\hat{\mu} = F(\mu_1, \ldots, \mu_n)$.

Refresh$(\mathbf{sk}_{ID,1}, \mathbf{sk}_{ID,2})$: Is a 2-party protocol executed between P_1 and P_2 taking as input their secret shares $\mathbf{sk}_{ID,1} = \mathbf{s}$ and $\mathbf{sk}_{ID,2}$ and outputting updated secret shares $sk_{ID,1}^{fresh}, sk_{ID,2}^{fresh}$.

4.1 Building Block: Leakage-resilient Symmetric Encryption

We recall a building block, termed leakage-resilient symmetric encryption introduced in (Goldwasser et al., 2010):
SymGen(λ) : On input λ outputs a uniformly random secret key $\mathbf{s} \leftarrow \mathcal{U}(\{0,1\}^{\lambda})$.
SymEnc$(\mathbf{s}, \mu)$: On input a secret key $\mathbf{s}$ and a message $\mu \in \{0,1\}^m$, outputs the ciphertext $\mathbf{C} = (\mathbf{A}, \mathbf{As} + \mathbf{e} + \mu\lfloor \frac{q}{2} \rfloor)$, where $\mathbf{A} \leftarrow \mathcal{U}(\mathbb{Z}_q^{m \times n})$ and $\mathbf{e} \leftarrow \mathcal{D}_{\sigma}^{m}$.
SymDec$(\mathbf{s}, \mathbf{C})$: On input a secret key $\mathbf{s}$ and a ciphertext $(\mathbf{A}, \mathbf{y})$, computes $\mathbf{y} - \mathbf{As}$ and outputs the message using a threshold decoder.

4.2 The Scheme

Intuition. Let Σ be an LWE-based IBE scheme consisting of four algorithms (IBE.Setup, IBE.Extract, IBE.Encrypt, IBE.Decrypt) as described in Section 2.4. To apply the transformation technique presented via "compiler" in (Gentry et al., 2013), the underlying IBE scheme needs to have the following three properties:

- The decryption key for identity ID and the corresponding ciphertext for ID, are $\mathbf{sk}_{ID}, \mathbf{c}_{ID} \leftarrow \mathcal{U}\left(\mathbb{Z}_q^{n'}\right)$. We extend the decryption key by adding 1 as the first component.
- If $\mathbf{c}_{ID}$ encrypts 0, then $\langle \mathbf{c}_{ID}, \mathbf{sk}_{ID} \rangle$ is small.
- Encryptions of 0 are indistinguishable from uniform vectors over $\mathbb{Z}_q$ (under LWE assumption).

Our transformation of Σ into a distributed LRFHIBE scheme follows in the next paragraph. The main idea of the compiler is that the encryption of a message $\mu \in \{0,1\}$ is represented by encryption of 0 using the encryption algorithm of the underlying IBE scheme. The decryptor uses the decryption algorithm of the fully homomorphic scheme FHE to recover the message μ. In order to make our scheme resilient against continual leakage our construction contains an additional algorithm to refresh the secret keys. At the beginning of the refresh protocol, P_1 holds a secret key which is represented by a ciphertext of the symmetric encryption and encrypts the secret key of the underlying public key encryption. The other party P_2 holds the secret key of the symmetric encryption scheme. The role of the refresh algorithm is to update the secret shares. As a result the new key of P_1 represents the new random symmetric encryption of secret key of the underlying asymmetric scheme using the new symmetric secret key which is to be held by P_2. During the refreshing process, P_2 chooses a new random secret and sends an encryption combining the old ciphertext with the new one and sends it to P_1. For his part, P_1 chooses a new secret key and sends a ciphertext to P_2 using its own share. After decrypting the received ciphertext, P_1 will receive a new secret share.

Note that the extraction algorithm first extracts identity-based secret key, then it generates the key shares for both parties. P_2 obtains encryption of the identity-based secret key, using symmetric secret key which is hold by P_1 as his secret key. Furthermore we point out that the decryption algorithm does not reconstruct the initial master secret key. The idea is that the two servers interact in a protocol sending each other certain values, where the final decryption procedure is done by one of the servers.

As showed by Goldwasser et al. (Goldwasser et al., 2010) the decisional LWE problem is leakage resilient. Regev (Regev, 2005) proved the search version of LWE (worst-case) is as hard as several lattice problems in the worst case. Eventually we know that decisional and search LWE are equivalent up to a polynomial in q factors, where q is a prime number. In order to achieve leakage resilience of our fully homomorphic IBE scheme we need to change the parameters of the underlying scheme which is based on Gentry et al. construction (Gentry et al., 2013). We assume that the circuit depth of the scheme is given by $L = poly(\lambda)$ and is a polynomial in the security parameter λ. The observance in the parameter setting is the fact that with a higher parameter σ of error distribution $\mathcal{D}_\sigma$ the security increases too. But according to correctness constraint, σ must be set low while q must grow exponentially with the depth of circuit. This contradicts to the hardness constraint where with an increasing circuit depth L, solving the underlying gapSVP becomes easy. This problem involves a solution where the parameter n has to be polynomial in the depth L. A deeper discussion on parameter settings has been provided by Berkoff and Liu (Berkoff and Liu, 2014). Following the proofs in (Berkoff and Liu, 2014) we set $m = 2n\log q + 3\lambda$, where $n = \tau^2$ and $\tau = \max\{L, \lambda^2\}$. The suitable error distribution parameter σ of $\mathcal{D}_\sigma$ is set equal to $\sigma = 2^{\log^2 \tau}$ and the prime q is bounded by $q \geq 2^{\tau \log^2 \tau}$. These parameters guarantee correctness of our leakage resilient fully homomorphic IBE scheme according to the analysis in (Berkoff and Liu, 2014).

Construction. A leakage-resilient fully-homomorphic distributed IBE scheme consists of the following six algorithms:

$\texttt{FHIB.Setup}(\lambda)$**:** It runs the $\texttt{IBE.Setup}$ algorithm of Σ to generate the master public key and master secret key mpk, msk.

$\texttt{FHIB.Extract}(mpk, msk, ID)$: The algorithm consists of 2 stages:

- **Stage 1.** It runs the extraction algorithm $\texttt{IBE.Extract}$ of Σ scheme in order to compute $\mathbf{sk}_{ID} \leftarrow \mathcal{U}\left(\mathbb{Z}_q^m\right)$, which is the decryption key of IBE scheme. It supplements the calculations by setting $\mathbf{sk}'_{ID} = (1, \mathbf{sk}_{ID}) \leftarrow \mathcal{U}\left(\mathbb{Z}_q^{m+1}\right)$. It computes the decryption key of LRFHIBE scheme as $\texttt{Powerof2}(\mathbf{sk}'_{ID}) = \mathbf{v}_{ID}$, where $\mathbf{v}_{ID} = (v_{ID,1}, \ldots, v_{ID,N}) \in \mathbb{Z}_q^{l\cdot(m+1)}$, $N = l(m+1)$. It outputs $\mathbf{v}_{ID}$.

- **Stage 2.** The secret key shares of the two parties are generated as follows: given the security parameter and the identity-based secret key $\mathbf{v}_{ID}$ it runs the $\texttt{SymGen}$ algorithm of symmetric scheme and outputs a random value $\mathbf{s} \leftarrow \mathcal{U}(\mathbb{Z}_q^n)$ which is the secret share of P_1, i.e. $\mathbf{sk}_{ID,1} = \mathbf{s}$. The secret share of P_2 is given by running the symmetric algorithm $\texttt{Encrypt}(\mathbf{s}, \mathbf{v}_{ID}) = \mathbf{As} + \mathbf{e} + \lfloor \frac{q}{2} \rfloor \mathbf{v}_{ID} = \mathbf{sk}_{ID,2}$, where $\mathbf{A} \leftarrow \mathcal{U}\left(\mathbb{Z}_q^{N\times n}\right)$, $\mathbf{e} \leftarrow \mathcal{D}_\sigma^N$.

$\texttt{FHIB.Encrypt}(mpk, ID, \mu_i)$: To encrypt the message $\mu_i \leftarrow \mathcal{U}(\{0,1\})$, the algorithm invokes $\texttt{IBE.Encrypt}$ of Σ to compute $N = l\cdot(m+1)$ encryptions of 0. The resulted ciphertext is denoted by $\mathbf{C}'_{ID}$. Taking $\mathbf{C}'_{ID}$ it computes the ciphertext of FHIBE by the following calculation:

$$\mathbf{C}_{ID} = \texttt{Flatten}\left(\mu \cdot I_N + \texttt{BitDecomp}(\mathbf{C}'_{ID})\right) \leftarrow \mathcal{U}\left(\mathbb{Z}_q^{N\times N}\right).$$

$\texttt{FHIB.Eval}(mpk, \mathbf{C}_{ID}(\mu_1), \ldots, \mathbf{C}_{ID}(\mu_n), F)$: Take as input the ciphertexts, $\mathbf{C}_{ID}(\mu_1), \ldots, \mathbf{C}_{ID}(\mu_n)$ and an evaluation function $F \in \{\texttt{Add}, \texttt{Mult}\}$. If $F = \texttt{Add}$, output

$$\begin{aligned}\hat{\mathbf{C}} &= \texttt{Add}(\mathbf{C}_{ID}(\mu_1), \ldots, \mathbf{C}_{ID}(\mu_n)) \\ &= \texttt{Flatten}(\mathbf{C}_{ID}(\mu_1) + \ldots + \mathbf{C}_{ID}(\mu_n)),\end{aligned}$$

else if $F = \texttt{Mult}$, output

$$\begin{aligned}\hat{\mathbf{C}} &= \texttt{Mult}(\mathbf{C}_{ID}(\mu_1), \ldots, \mathbf{C}_{ID}(\mu_n)) \\ &= \texttt{Flatten}(\mathbf{C}_{ID}(\mu_1) \times \ldots \times \mathbf{C}_{ID}(\mu_n)).\end{aligned}$$

$\texttt{FHIB.Decrypt}(mpk, \hat{\mathbf{C}}, \mathbf{sk}_{ID,1}, \mathbf{sk}_{ID,2})$: Is the following 2-party protocol executed between P_1 and P_2 on a given ciphertext $\hat{\mathbf{C}}$. Assume that both parties know index i, which represents the required row i of ciphertext $\hat{\mathbf{C}}$:

- P_2 picks a uniformly random $\mathbf{s}' \leftarrow \mathcal{U}(\mathbb{Z}_q^n)$, takes the $i-$th row of ciphertexts $\hat{\mathbf{C}}_i$ of the underlying asymmetric scheme, and computes $\mathbf{d} = \mathbf{As}'\hat{\mathbf{C}}_i + \mathbf{sk}_{ID,2}\hat{\mathbf{C}}_i = \mathbf{As}'\hat{\mathbf{C}}_i + \mathbf{As}\hat{\mathbf{C}}_i + \mathbf{e}\hat{\mathbf{C}}_i + \left\langle \mathbf{v}_{ID}, \hat{\mathbf{C}}_i \right\rangle \lfloor \frac{q}{2} \rfloor$. Finally P_2 sends $\mathbf{d}$ to P_1.
- P_1 computes $\mathbf{d}' = \mathbf{d} - \mathbf{AsC}_i$ and sends it back to P_2.
- P_2 decrypts by executing the simple computation $\mathbf{d}' - \mathbf{As}'\hat{\mathbf{C}}_i = \left\langle \mathbf{v}_{ID}, \hat{\mathbf{C}}_i \right\rangle \lfloor \frac{q}{2} \rfloor + \mathbf{e}$.

$\texttt{Refresh}(\mathbf{sk}_{ID,1}, \mathbf{sk}_{ID,2})$: Is a 2-party protocol executed between P_1 and P_2 taking as input their secret shares $\mathbf{sk}_{ID,1} = \mathbf{s}$ and $\mathbf{sk}_{ID,2} = SymEnc_{\mathbf{s}}(\mathbf{v}_{ID})$.

- P_2 uses the previously picked secret $\mathbf{s}'$, picks a new error vector $\mathbf{e}' \leftarrow (\mathcal{D}_\sigma)^N$, computes $\mathbf{f} = \mathbf{sk}_{ID,2} + \mathbf{As}' + \mathbf{e}'$ and sends it to P_1.
- Using its secret key, P_1 computes $\mathbf{As} + \mathbf{e}$ and taking $\mathbf{f}$ it sets $\mathbf{f} - (\mathbf{As} + \mathbf{e}) = \mathbf{f}'$. In the next step P_1 chooses a new secret share $\mathbf{sk}_{ID,1}^{fresh}$ given by $\mathbf{s}'' \leftarrow \mathcal{U}(\mathbb{Z}_q^n), \mathbf{e}'' \leftarrow (\mathcal{D}_\sigma)^N$, computes $\mathbf{f}'' = \mathbf{As}'' + \mathbf{e}'' + \mathbf{f}'$ and sends it to P_2.
- Upon receiving $\mathbf{f}''$, P_2 subtracts $\mathbf{As}' + \mathbf{e}'$ and obtains the refreshed secret key $\mathbf{sk}_{ID,2}^{fresh} = \mathbf{As}'' + \mathbf{e}'' + \mathbf{v}_{ID} \lfloor \frac{q}{2} \rfloor$.

Correctness. Correctness of the scheme follows due to the fact that: $\left\langle \mathbf{v}_{ID}, \hat{\mathbf{C}}_i \right\rangle = \mu \mathbf{v}_{ID} + small$.

5 SECURITY ANALYSIS

Our scheme offers an extension of Berkoff and Liu (Berkoff and Liu, 2014) scheme to a distributed setting by introducing a 2-party protocol between two servers who run the decryption process. Furthermore our scheme remains secure even if an adversary can leak a part of the secret key at each time period, while the leakage is represented by an uninvertible function. This security model is called continual auxiliary leakage model and is particularly attractive in contrast to the previous leakage models reviewed in the introduction of our work. Analogously to (Yuen et al., 2012) our model does not require erasure of the secret key after each update in every period.

Theorem 7. *The fully homomorphic distributed identity-based encryption is resilient to continual auxiliary input leakage under the assumption that the LWE problem is hard, the underlying identity-based encryption and symmetric encryption are both secure under chosen-plaintext attacks.*

Proof. Assume an adversary $\mathcal{A}$ which plays the security experiment from section 4. The security holds even for auxiliary leakage functions $h(mpk, \mathbf{sk}_{ID,1}, \mathbf{sk}_{ID,2})$ which are hard to invert. The goal is to generate secret key shares which cannot be used to decrypt the challenge ciphertext, in other words those secret key shares will always decrypt to $\perp$. We show that if the leakage function is uninvertible, an adversary will not be able to gain information about the secret key which can be used to distinguish the ciphertext from a random ciphertext. Later on we have to show how to extend it to the continual leakage model where the secret key shares will be updated in each time period. We will use a CPA adversary $\mathcal{A}$ against our scheme to construct a polynomial time algorithm $\mathcal{B}$ against the LWE problem.

Additionally we prove that our scheme achieves the appealing property of adaptivity, where an adversary can choose the leakage function after seeing the public key. In order to achieve this property we need to show that the ciphertext remains computational indistinguishable form random even if we assume that the public key can be distinguished from a random value. In the decisional LWE problem, the adversary is provided with the access to a sampling oracle. Note that this oracle can be either a pseudorandom oracle O_{pr} with some included secret s' or a truly random oracle, with all random samples over $\mathbb{Z}_q^{\cdot}$ (where the exponent differs according to the required vectors). $\mathcal{B}$ samples $N+1$ vectors $\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_N \leftarrow \mathcal{U}\left(\mathbb{Z}_q^n\right)^N$ and a vector $\mathbf{v}$ from $\mathbb{Z}_q^N$. Similarly it samples more vectors for the simulation of master public key and master secret key. Before we proceed with the proof, we describe how $\mathcal{B}$ simulates the received queries from $\mathcal{A}$.

Extraction Queries: Whenever $\mathcal{A}$ issues extraction queries on identity ID and an index $i \leftarrow \mathcal{U}(\{1,2\})$, $\mathcal{B}$ picks the sampled vector $\mathbf{v}$. Then it takes the sampled n-dimensional vector $\mathbf{a}_0$ and the remained N vectors $\mathbf{a}_1, \ldots, \mathbf{a}_N$ and defines $\mathbf{A} = [\mathbf{a}_1 | \ldots | \mathbf{a}_N]$. It simulates $\mathbf{sk}_{ID,1} = \mathbf{a}_0$ and $\mathbf{sk}_{ID,2} = \mathbf{A}\mathbf{a}_0 + \mathbf{e} + \mathbf{v}$, where $\mathbf{e} \leftarrow \mathcal{D}_\sigma^N$ is a sample corresponding to the Gaussian distribution $\mathcal{D}_\sigma^N$. It returns $\{\mathbf{sk}_{ID,i}\}_{i \in \{1,2\}}$ to the adversary $\mathcal{A}_{ind}$, sets $\mathbf{sk}_{ID} = \mathbf{v}_{ID} = \mathbf{v}$ and stores $(ID, \mathbf{sk}_{ID,1}, \mathbf{sk}_{ID,2})$ in the list $\mathcal{L}_{ex}$.

Leakage Queries: Whenever $\mathcal{A}$ issues leakage queries on input an identity ID and a leakage function h, $\mathcal{B}$ simulates the input of the leakage function by first running the simulation of the secret share queries described above and returns $h(mpk, ID, \mathbf{sk}_{ID,1}, \mathbf{sk}_{ID,2})$.

Refresh Queries: The challenger $\mathcal{B}$ simulates the queries as follows: Whenever $\mathcal{A}_{ind}$ issues a refresh query on an identity ID, $\mathcal{B}$ first checks whether $(ID, \mathbf{sk}_{ID,1}, \mathbf{sk}_{ID,2}) \in \mathcal{L}_{ex}$. If so, it picks a random value $\mathbf{s}' \in \mathcal{U}(\mathbb{Z}_q^n)$ and builds a matrix $\mathbf{A}$ of sampled vectors $\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_N$. It picks a value $\mathbf{e}' \leftarrow \mathcal{D}_\sigma^N$ and sets $\mathbf{sk}_{ID,1}^{fresh} = (\mathbf{s}', \mathbf{e}')$. By picking another random $\mathbf{s}'' \in \mathcal{U}(\mathbb{Z}_q^n)$ and $\mathbf{e}'' \leftarrow \mathcal{D}_\sigma^N$, the challenger simulates $\mathbf{sk}_{ID,2}^{fresh} = \mathbf{A}\mathbf{s}'' + \mathbf{e}'' + \mathbf{v}_{ID}$ (where $\mathbf{v}_{ID} = \mathbf{v}$ is the previously sampled vector.

For the continuation of the proof we use the hybrid argument with a sequence of games. The oracle queries in each game are simulated as showed above. The initial game $Game_0$ is the real game as described in section 4. Each game profits from adversary's computational boundary and from the fact that this adversary cannot distinguish the ciphertext from random. $Game_i$ is different from the initial (real) $Game_0$ by definition of secret key shares and ciphertext which are so modified that the secret key shares do not decrypt correctly the modified ciphertext.
Gentry et al. (Gentry et al., 2013) defined a compiler that transforms any LWE-based IBE scheme into a fully-homomorphic IBE scheme. Since we exploit their technique, we recall the three main properties of that should have the underlying IBE scheme which we apply to our proof:

Ciphertext and Decryption Key Vectors: The ID-based secret key $\mathbf{sk}_{ID}$ and a ciphertext for $\mathbf{c}_{ID}$, are vectors in $\mathbb{Z}_q^{n+1}$, where the first coefficient is 1.
Small Dot Product: If $\mathbf{c}_{ID}$ encrypts 0, then $\langle \mathbf{c}_{ID}, \mathbf{sk}_{ID} \rangle$ is small.

Security: Encryptions of 0 are indistinguishable from uniform vectors over $\mathbb{Z}_q$ under LWE assumption.

To guarantee these properties we assume an underlying identity-based encryption, which has the required properties and is secure against chosen-plaintext attacks. The most famous schemes satisfying the mentioned features have been introduced in (Gentry et al., 2008; Agrawal et al., 2010; Cash et al., 2010).

We have to show that the compiler is leakage-resilient in continual leakage model with auxiliary input. Due to the fact that this compiler uses encryption algorithm of the underlying IBE scheme and generates N encryptions of 0, the ciphertext is indistinguishable from random according to the third property mentioned above. Since the underlying IBE scheme is assumed to be secure under LWE assumption, which is known to be leakage resilient, we conclude that leakage resilience of the compiler is guaranteed too.

Furthermore, in order to prove resistance against the continual auxiliary leakage we first define a class of function family $\mathcal{F}$ having a minimal entropy ξ_{min} of the ID-based secret shares, which at the same time denotes the length of the secret shares. Observe a set $\mathcal{S}$ of all queries secret key shares for both servers, (sk_{ID_1}, sk_{ID_2}). Assuming that $\mathcal{S}^*$ denotes the set of all secret keys on the challenge identity ID^*, then the intersection set $\mathcal{S} \cap \mathcal{S}^* = \emptyset$. We denote by $\mathcal{F}(g(\xi_{min}))$ the class of all probabilistic polynomial functions h such that for all leakage queries $i \in [1, \dots, q_l]$, with access to the master public key, challenge identity ID^*, set $\mathcal{S}$ and the leakage function $h(mpk, ID, sk_{ID_1}, sk_{ID_2})$, where ID is one of the queried identities to the leakage oracle, no PPT adversary $\mathcal{A}$ can find the valid secret shares $sk_{ID_1^*}, sk_{ID_2^*}$ with a greater probability than the hardness parameter $g(\xi_{min})$, where $g(\xi_{min}) \geq 2^{-\xi_{min}}$. We say that the underlying IBE and symmetric encryption schemes are supposed to be CPA secure against continual auxiliary leakages if they are indistinguishable CPA secure with respect to the family $\mathcal{F}(g(\xi_{min}))$ described above.

Let ID^* be the challenge identity. The adversary's view in the real game is represented by the tuple $(mpk, \mathbf{C}_{ID}, h(mpk, \mathbf{sk}_{ID,1}, \mathbf{sk}_{ID,2}))$, where $\mathbf{C}_{ID} = \mathtt{Flatten}(\mu \cdot \mathbf{I}_N + \mathtt{BitDecomp}(\mathbf{C}'_{ID}))$, with $\mathbf{C}'_{ID}$ being a ciphertext from the underlying identity-based encryption which represents N encryptions of 0 and $\mu \leftarrow \mathcal{U}(\{0,1\})$. Upon applying $\mathtt{BitDecomp}^{-1}$, we obtain $\mathbf{C}_{ID} = \mathtt{BitDecomp}^{-1}(\mu \cdot \mathbf{I}_N) + \mathbf{C}'_{ID}$. We note that $\mathtt{BitDecomp}$ and thus $\mathtt{BitDecomp}^{-1}$ are deterministic operation, it is easier to assume an adversary $\mathcal{A}$ who runs the security experiment with $\hat{\mathbf{C}}_{ID}$. That means we have to show that $\mathcal{A}$ cannot distinguish $\mathbf{C}'_{ID}$ from a randomly picked value $\mathbf{V} \leftarrow \mathcal{U}(\mathbb{Z}_q^{N \times (n+1)})$.

Adversary's view in the real game $Game0$ is given by the following hybrid tuple:

$$Hybrid_0 := \left(mpk, \mathbf{A}, \mathbf{b}, \mathbf{C}'_{ID}, h(mpk, \mathbf{sk}_{ID,1}, \mathbf{sk}_{ID,2})\right),$$

where a random hybrid tuple of a $Game_r$ is given by $Hybrid_r := (mpk, \mathbf{A}, \mathbf{u}, V, h(mpk, \mathbf{sk}_{ID,1}, \mathbf{sk}_{ID,2}))$, where $\mathbf{A} \leftarrow \mathcal{U}\left(\mathbb{Z}_q^{N \times n}\right)$ and $\mathbf{b} = \mathbf{As} + \mathbf{e}$ is the public key of the underlying symmetric scheme, which is used during the key distribution process. We have to show that the two hybrid tuples are computationally equivalent, i.e $Hybrid_0 \approx_c Hybrid_r$. To represent the view of adversary $\mathcal{A}$ in each other game $Game_i$ we define a tuple $Hybrid_i$ where each row of $\mathbf{C}_{ID}$ is replaced by a random vector $\mathbf{v}_{ID,i} \leftarrow \mathcal{U}\left(\mathbb{Z}_q^{n+1}\right)$. So we have to show that $Hybrid_i \approx_c Hybrid_{i+1}$, where $0 < i \leq N$.

According to the Definition 4 (after amending the parameters) the following approximation holds:

$$\begin{aligned}(\mathbf{A}, \mathbf{As} + \mathbf{e}, h(mpk, \mathbf{sk}_{ID,1}, \mathbf{sk}_{ID,2}))& \\ \approx_c (\mathbf{A}, \mathbf{u}, h(mpk, \mathbf{sk}_{ID,1}, \mathbf{sk}_{ID,2}))&,\end{aligned}$$

where $\mathbf{u} \leftarrow \mathcal{U}(\mathbb{Z}_q^N)$. According to the security property of our IBE compiler, $\mathbf{C}'_{ID}$ is indistinguishable from uniform vectors $\mathbf{v}_{ID,i} \leftarrow \mathcal{U}\left(\mathbb{Z}_q^{n+1}\right)$, such that we can say $\mathbf{C}'_{ID} \approx_c \{\mathbf{v}_{ID,i}\}_{i \in [N]}$. Eventually, we obtain the following result:

$$\begin{aligned}\left(mpk, \mathbf{A}, \mathbf{b}, \mathbf{C}'_{ID}, h(mpk, \mathbf{sk}_{ID,1}, \mathbf{sk}_{ID,2})\right)& \\ \approx (mpk, \mathbf{A}, \mathbf{u}, \mathbf{V}, h(mpk, \mathbf{sk}_{ID,1}, \mathbf{sk}_{ID,2}))&.\end{aligned}$$

According to the second property of the IBE compiler, the dot product of a ciphertext with the secret key is small. That means an adversary cannot distinguish the modified secret key from the real one, assuming that the leakage function is uninvertible.

As next we have to handle with the fact that there is a continual leakage, meaning existence of a certain sequence of phases where leakage of the secret key shares occurs. Each new phase follows after running a refresh algorithm and generating fresh secret key shares. Considering the refresh algorithm and choosing a random $\mathbf{s} \leftarrow \mathcal{U}\left(\mathbb{Z}_q^n\right)$ as the first secret share, and a random $\mathbf{v}' \leftarrow \mathcal{U}\left(\mathbb{Z}_q^N\right)$ it computes $\mathbf{As} + \mathbf{e} + \lfloor \frac{q}{2} \rfloor \cdot \mathbf{v}'$, where $\mathbf{e} \leftarrow (\mathcal{D}_\sigma)^N$ an error. After running the refresh algorithm, we see that the new secret shares are indistinguishable from random.

In order to complete the proof we show that inversion of leakage function h can be reduced to the fact that the consecutive hybrid games are indistinguishable from each other. For reason of simplicity we shot that the advantage between $Hybrid_0$ and $Hybrid_r$ is negligible. We show it by a contradiction assuming that the advantage between the hybrids is non-negligible, i.e. $|Adv(Hybrid_0) - Adv(Hybrid_r)| \geq negl(\lambda)$. This assumption would mean that there is an adversary $\mathcal{B}$ against LWE, such that

$$\begin{aligned}&Pr[\mathcal{B}\left(\mathbf{A}, \mathbf{As} + \mathbf{e}, h(mpk, \mathbf{sk}_{ID,1}, \mathbf{sk}_{ID,2}), \mathbf{C}'_{ID}\mathbf{s}\right) = 1] \\ &- Pr[\mathcal{B}\left(\mathbf{A}, \mathbf{u}, h(mpk, \mathbf{sk}_{ID,1}, \mathbf{sk}_{ID,2}), \mathbf{V}\right) = 1] \geq \varepsilon(\lambda),\end{aligned}$$

This is a contradiction due to the computational indistinguishabilities we showed above. Thus we can conclude

$$|\mathbf{Adv}(Hybrid_0) - \mathbf{Adv}(Hybrid_r)| \leq negl(\lambda).$$

□

6 CONCLUSION

In this work, we described how to construct a leakage-resilient distributed identity-based encryption scheme having the fully homomorphic property making the scheme appealing to such applications like cloud security of medical and financial data. The leakage model we considered is called continual auxiliary leakage model. It aims at allowing a constant leakage of information on the secret key. To achieve that, the lifetime of the system is split in time frames during which the adversary has access to an auxiliary input represented by some uninvertible function. At the end of each frame, the key is replaced by a new one and the process can continue for an unbounded amount of time.

Our construction lives in a distributed setting where a secret key is shared between two devices. The refreshing procedure is made through a two party protocol updating the shares while keeping the same public key. Security is proven under the LWE assumption which enjoy strong leakage-resilient properties and is believed to resist attacks from quantum adversaries.

REFERENCES

Agrawal, S., Boneh, D., and Boyen, X. ((2010)). Efficient lattice (H)IBE in the standard model. In *EUROCRYPT*, volume 6110 of *LNCS*, pages 553–572. Springer.

Akavia, A., Goldwasser, S., and Hazay, C. (2012). Distributed public key schemes secure against continual leakage. In *ACM Symposium on Principles of Distributed Computing, PODC, 2012*, pages 155–164. ACM.

Akavia, A., Goldwasser, S., and Vaikuntanathan, V. (2009). Simultaneous hardcore bits and cryptography against memory attacks. In *Theory of Cryptography, TCC, 2009. Proceedings*, volume 5444 of *Lecture Notes in Computer Science*, pages 474–495. Springer.

Alwen, J., Dodis, Y., Naor, M., Segev, G., Walfish, S., and Wichs, D. (2010). Public-key encryption in the bounded-retrieval model. In *Advances in Cryptology - EUROCRYPT 2010. Proceedings*, volume 6110 of *Lecture Notes in Computer Science*, pages 113–134. Springer.

Alwen, J., Krenn, S., Pietrzak, K., and Wichs, D. (2013). Learning with rounding, revisited - new reduction, properties and applications. In *Advances in Cryptology - CRYPTO 2013 - Proceedings, Part I*, volume 8042 of *Lecture Notes in Computer Science*, pages 57–74. Springer.

Armknecht, F., Boyd, C., Carr, C., Gjøsteen, K., Jäschke, A., Reuter, C. A., and Strand, M. (2015). A guide to fully homomorphic encryption. Cryptology ePrint Archive, Report 2015/1192. http://eprint.iacr.org/2015/1192.

Berkoff, A. and Liu, F. (2014). Leakage resilient fully homomorphic encryption. In *Theory of Cryptography - 11th Theory of Cryptography Conference, TCC 2014. Proceedings*, volume 8349 of *Lecture Notes in Computer Science*, pages 515–539. Springer.

Brakerski, Z., Kalai, Y. T., Katz, J., and Vaikuntanathan, V. (2010). Overcoming the hole in the bucket: Public-key cryptography resilient to continual memory leakage. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010*, pages 501–510. IEEE Computer Society.

Brakerski, Z. and Vaikuntanathan, V. (2011a). Efficient fully homomorphic encryption from (standard) LWE. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS, 2011*, pages 97–106. IEEE Computer Society.

Brakerski, Z. and Vaikuntanathan, V. (2011b). Fully homomorphic encryption from ring-lwe and security for key dependent messages. In *Advances in Cryptology - CRYPTO 2011, Proceedings*, pages 505–524.

Canetti, R., Dodis, Y., Halevi, S., Kushilevitz, E., and Sahai, A. (2000). Exposure-resilient functions and all-or-nothing transforms. In *Advances in Cryptology - EUROCRYPT 2000, Proceeding*, volume 1807 of *Lecture Notes in Computer Science*, pages 453–469. Springer.

Cash, D., Hofheinz, D., Kiltz, E., and Peikert, C. ((2010)). Bonsai trees, or how to delegate a lattice basis. In *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 523–552. Springer.

Chow, S. S. M., Dodis, Y., Rouselakis, Y., and Waters, B. (2010). Practical leakage-resilient identity-based encryption from simple assumptions. In *Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS 2010, Chicago, Illinois, USA, October 4-8, 2010*, pages 152–161. ACM.

Clear, M. and McGoldrick, C. (2015). Multi-identity and multi-key leveled FHE from learning with errors. In *Advances in Cryptology - CRYPTO 2015 - Proceedings, Part II*, volume 9216 of *LNCS*, pages 630–656. Springer.

Dodis, Y., Goldwasser, S., Kalai, Y. T., Peikert, C., and Vaikuntanathan, V. (2010a). Public-key encryption schemes with auxiliary inputs. In *Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010. Proceedings*, volume 5978 of *Lecture Notes in Computer Science*, pages 361–381. Springer.

Dodis, Y., Haralambiev, K., López-Alt, A., and Wichs, D. (2010b). Cryptography against continuous memory attacks. In *51th Annual IEEE Symposium on Founda-*

tions of Computer Science, FOCS 2010, pages 511–520. IEEE Computer Society.

Dodis, Y., Kalai, Y. T., and Lovett, S. (2009). On cryptography with auxiliary input. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 621–630. ACM.

Dziembowski, S. and Pietrzak, K. (2008). Leakage-resilient cryptography. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, pages 293–302. IEEE Computer Society.

Gentry, C. (2009a). *A fully homomorphic encryption scheme*. PhD thesis, Stanford University.

Gentry, C. (2009b). Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009*, pages 169–178. ACM.

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

Gentry, C., Sahai, A., and Waters, B. ((2013)). Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *CRYPTO 2013*, volume 8042 of *LNCS*, pages 75–92. Springer.

Goldwasser, S., Kalai, Y. T., Peikert, C., and Vaikuntanathan, V. (2010). Robustness of the learning with errors assumption. In *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*, pages 230–240. Tsinghua University Press.

Goyal, V., Pandey, O., Sahai, A., and Waters, B. (2006). Attribute-based encryption for fine-grained access control of encrypted data. In *CCS 2006*, pages 89–98. ACM.

Halderman, J. A., Schoen, S. D., Heninger, N., Clarkson, W., Paul, W., Calandrino, J. A., Feldman, A. J., Appelbaum, J., and Felten, E. W. (2009). Lest we remember: cold-boot attacks on encryption keys. *Commun. ACM*, 52(5):91–98.

Li, M., Yu, S., Ren, K., and Lou, W. (2010). Securing personal health records in cloud computing: Patient-centric and fine-grained data access control in multi-owner settings. pages 89–106.

Micali, S. and Reyzin, L. (2004). Physically observable cryptography (extended abstract). In *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004, Proceedings*, volume 2951 of *Lecture Notes in Computer Science*, pages 278–296. Springer.

Naor, M. and Segev, G. (2009). Public-key cryptosystems resilient to key leakage. In *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, volume 5677 of *Lecture Notes in Computer Science*, pages 18–35. Springer.

Pietrzak, K. (2009). A leakage-resilient mode of operation. In *Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings*, volume 5479 of *Lecture Notes in Computer Science*, pages 462–482. Springer.

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

Rivest, R. L., Adleman, L., and Dertouzos, M. L. (1978). On data banks and privacy homomorphisms. *Foundations of Secure Computation, Academia Press*, pages 169–179.

Sahai, A. and Waters, B. (2005). Fuzzy identity-based encryption. In *EUROCRYPT 2005, Proceedings*, volume 3494 of *LNCS*, pages 457–473. Springer.

Smart, N. P. and Vercauteren, F. (2010). Fully homomorphic encryption with relatively small key and ciphertext sizes. In *Public Key Cryptography - PKC 2010, Proceedings*, volume 6056 of *Lecture Notes in Computer Science*, pages 420–443. Springer.

van Dijk, M., Gentry, C., Halevi, S., and Vaikuntanathan, V. (2010). Fully homomorphic encryption over the integers. In *Advances in Cryptology - EUROCRYPT, 2010, Proceedings*, pages 24–43.

Yuen, T. H., Chow, S. S. M., Zhang, Y., and Yiu, S. (2012). Identity-based encryption resilient to continual auxiliary leakage. In *Advances in Cryptology - EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 117–134. Springer.