ALGAES: An Authenticated Lattice-based Generic Asymmetric Encryption Scheme

Aravind Vishnu S S*[1], M Sethumadhavan[2], and Lakshmy K V[2]

[1]Department of Mathematics, Amrita School of Physical Sciences, Amrita Vishwa Vidyapeetham, Coimbatore, India
[2]TIFAC-CORE in Cyber Security, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Coimbatore, India

Abstract

In this article, we propose a generic hybrid encryption scheme providing entity authentication. The scheme is based on lossy trapdoor functions relying on the hardness of the Learning With Errors problem. The construction can be used on a number of different security requirements with minimal reconfiguration. It ensures entity authentication and ciphertext integrity while providing security against adaptive chosen ciphertext attacks in the standard model. As a desired characteristic of schemes providing entity authentication, we prove the strong unforgeability under chosen message attack for the construction. In addition, the scheme is post-quantum secure based on the hardness of the underlying assumption.

Keywords: Hybrid Encryption, Learning With Errors, IND-CCA2, Lossy Trapdoor, Lattice-based, Signcryption, Post-Quantum

1 Introduction

The National Institute of Standards and Technology (NIST) announced a competition for standardising post-quantum secure cryptosystems in 2017. The Computer Security Resource Center (CSRC) hosted the competition and selected CRYSTALS-Kyber [11], which is a lattice-based cryptosystem as a standard in Public-key Encryption and Key-Establishment Algorithms. Unlike the previous competitions hosted by NIST, the competition is still on among other

*Corresponding author email: cb.sc.d.mat17003@cb.students.amrita.edu

cryptosystems submitted based on different hard problems other than lattice-based hard problems. Thus it is evident that the world is looking at lattice-based cryptosystems as a promising candidate for security solutions in the post-quantum era. There are many complex problems based on lattices. To name a few, the shortest vector problem, the closest vector problem, the Shortest Independent Vector Problem and so on. But, a problem which can be used to construct a cryptographic one-way trapdoor function was introduced in a seminal work done by Miklós Ajtai [1] when they introduced the Shortest Integer Solution (SIS) problem. Almost a decade later, Oded Regev introduced a problem titled the Learning With Errors (LWE) problem [27]. The underlying problem for most of the initial round submissions received in NIST post-quantum standard competition was the LWE problem. The development of lattice-based cryptography until 2016 was studied and detailed by Chris Peikert in an article titled 'A decade of lattice cryptography' [25].

The hybrid encryption algorithm introduced by Victor Shoup [29], and the notion of key encapsulation scheme, now called key encapsulation mechanism (KEM), made a huge impact on asymmetric cryptographic constructions. This can be asserted from the number of hybrid KEMs submitted to the NIST competition. The Fujisaki-Okamoto transformation [16] for securely integrating a symmetric and asymmetric cryptosystem has a seminal impact on the growth and popularity of the above mentioned hybrid KEMs. In 2008, Chris Peikert and Brent Waters introduced the lossy trapdoor functions [26]. The work aimed to construct a new primitive and construction technique for asymmetric cryptosystems. They introduced a hybrid cryptosystem in which the asymmetric part was developed using lossy trapdoor functions, whereas the symmetric part was just the xor operation with a random string. The article proposes a generic construction mechanism. It also specifies how to realise Diffie-Hellman assumption and Lattice-based problems to the proposed construction. The lossy trapdoor functions find their significance in many black box-mannered cryptographic constructions [2], and some of the best examples can be found in [18, 5]. In recent literature, we can find techniques that optimize lossy trapdoor-based constructions [19] and hence the relevance of the trapdoor.

Bertoni *et al.* introduced the sponge-based hash functions [8]. In particular, a sponge-based hash function named Keccak [7] proposed by Bertoni *et al.* became SHA-3. The sponge-based constructions are preferred in integrating symmetric and asymmetric cryptosystems into a hybrid cryptosystem due to the properties offered by the construction. Yuliang Zheng [34] introduced 'signcryption' that combines entity authentication technique and message integrity in the same ciphertext. It is always preferred to have entity authentication along with message integrity and ciphertext integrity to the ciphertext. In the recent literature, we can see the applications of certificateless signcryption schemes as in [17] The sponge-based hash function named Hash-One [24] was used in one such recent construction of a certificateless hybrid signcryption scheme [4]. They used the sponge-based property of Hash-one and proposed a construction that allows users to switch among symmetric cryptosystems, according to the security requirement.

In 1993, Bellare and Rogaway introduced the Random Oracle Model [6]. The idea of a Random Oracle Model-based formal proof technique was to replace any pseudorandom functions used in a protocol or algorithm with oracles that are supposed to give truly random outputs. On the other hand, the standard model considers only complexity assumptions, which is the real-time scenario. So schemes which are having a formal security proof based on standard model are always considered superior to the ones with proofs in Random Oracle Model. This article proposes constructing a hybrid encryption mechanism, also a Key Encapsulation Mechanism (KEM), that provides entity authentication. So our proposal is similar and comparable to signcryption schemes. We employ the concept of a Master Authority in every communication channel so that a user who wants to communicate with another user in a particular network needs to register themselves under the corresponding master authority and obtain their user id from the master authority. At the time of registration, the user chooses a signature scheme along with a signing key and exchanges the verification key securely with the master authority to verify the ciphertext integrity. We use a sponge-based hash function construction that enables us to change the symmetric cryptosystem according to the security requirement. The security of the proposed construction is analysed against adversaries performing different attacks. Peikert and Waters [26] proposed a construction to use one-time strong unforgeable signatures and used that in proving IND-CCA2 security in the standard model. Our scheme allows signature reuse while maintaining the IND-CCA2 security in the standard model.

Motivation and Related works: Recently, we can see many of the signcryption schemes finds applications in networks, specifically in VANET, [3, 15, 35], IoT [28, 30] etc. Even though in earlier times, Computational Diffie-Hellman based signcryption schemes [21] were popular, some of the recent works are also relying on classical discrete logarithm assumption. Post-quantum signcryption schemes existing in the literature are very limited and one among the pioneering works on lattice-based signcryption is by Wang *et al.* [31] in 2012. Recently, Klamti and Hasan proposed a code-based hybrid signcryption scheme [20]. They used the Short Integer solution problem (SIS) and the LWE problem to come up with the construction. Recently, in 2019, Yand *et al.* [32] proposed another lattice-based signcryption mechanism based on Ring Learning with errors problem (RLWE) and Ideal-SIS problem. Apart from lattice-based constructions, in 2022, Dey *et al.* [13] proposed an isogeny-based post-quantum secure signcryption scheme. Though [13] is the recent development in post-quantum signcryption, the key recovery attack proposed by Castryck and Decru [12] on Supersingular isogeny-based Diffie Hellman made an impact on the security of isogeny-based constructions. The lattice-based constructions using SIS, LWE problems and their adaptations are secure. But our construction stands alone from those as we do not directly employ the aforementioned problems. We use the LWE problem for the construction of a lossy trapdoor function involved in our construction.

In this article, Section 2 discusses the preliminaries. The proposed construction and correctness proof is mentioned in Section 3. Section 4 describes the security

analysis, compares the proposed construction with similar schemes and Section 5 concludes the article.

2 Preliminaries

In this section, we revisit some important definitions which will be used in the article.

2.1 Basic Notions

- Statistical Distance: Let S be a countable set and let X and Y be two random variables over S.

$$\Delta(X,Y) = \frac{1}{2}\sum_{s\in S}|Pr[X=s] - Pr[Y=s]|$$

  is known as the statistical distance between X and Y.

- Minimum Entropy: For a random variable X, over a domain S,

$$H_\infty(X) = -\log_2(\max_{s\in S} Pr[X=s])$$

  is the minimum entropy of X.

- Average minimum entropy: It evaluates the conditional probability of one random variable conditioned on another [14].

$$\tilde{H}_\infty(X|Y) = -\log_2\left(\underset{y\xleftarrow{} Y}{E}\left[\max_{s\in S} Pr[X=s|Y=y]\right]\right)$$

2.2 Learning With Errors (LWE)

Oded Regev [27] introduced the Learning with Errors (LWE) problem in 2005. The problem established its role in lattice-based cryptographic applications and fetched the author the Gödel prize in 2018. This problem paves the foundation for the current post-quantum standard CRYSTALS-Kyber [11]. Let $0 < m \in \mathbb{Z}$ and $1 < n \in \mathbb{Z}$ respectively be the dimension and the modulus. Suppose that $\mathbf{s} \in \mathbb{Z}_n^m$ and χ is a probability distribution over $\mathbb{Z}_n$. $\mathcal{D}_{\mathbf{s},\chi}$ is the distribution on $\mathbb{Z}_n^m \times \mathbb{Z}_n$ that takes $a \xleftarrow{\$} \mathbb{Z}_n^m$, $e \xleftarrow{\$} \chi$ as input and outputs $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s}\rangle + e)$. Here $x \xleftarrow{\$} Y$ represents the operation of assigning x with a value from Y chosen uniformly at random. The LWE problem is to find $\mathbf{s}$ given many independent samples from the distribution $\mathcal{D}_{\mathbf{s},\chi}$.

2.3 Construction of Lossy and ABO functions

Peikert and Waters introduced the lossy and All-But-One (ABO) functions in [26]. We find these functions in many recent applications [2]. The functions are described as follows.

2.3.1 Lossy Trapdoor Function

Let λ be a security parameter, $\mu(\lambda)$ be the length of the input of the function and let $\nu(\lambda) \leq \mu(\lambda)$ be the lossiness. We also define the residual leakage $\gamma(\lambda) = \mu(\lambda) - \nu(\lambda)$. The 4-tuple of PPT algorithms $(\Phi_i, \Phi_l, F_l, F_l^{-1})$ is called a collection of (μ, ν)-lossy TDFs according to the following rules.

- sampling injective function with trapdoor: Φ_i outputs the ordered pair (ζ, η) in which the component ζ is the function index and the component η is the trapdoor. $F_l(\zeta, \cdot)$ computes the function $f_\zeta(\cdot)$ over the domain $\{0,1\}^\mu$. Similarly $F_l^{-1}(t, \cdot)$ computes $f_\zeta^{-1}(\cdot)$. The function's behaviour beyond the range of f_ζ is not defined and is redundant.

- Φ_l outputs $(\zeta, \perp)$ where ζ is as in the previous one with the difference being F_l computes $f_\zeta(\cdot)$ over $\{0,1\}^\mu$ with range size not exceeding $2^\gamma = 2^{\mu-\nu}$

For the lattice-based construction, we require a relaxed condition. Consider the functions holding the above-stated rules valid with an overwhelming probability over the randomness of Φ_i. We call such functions *almost-always* lossy trapdoor functions.

2.4 Injective and Lossy Functions from LWE

We use the construction mechanism proposed by Peikert and Waters [26] for developing a lossy trapdoor function based on the LWE problem. To begin proceedings, we introduce some pre-requisites as follows.

Concealer matrix: Suppose that m_1, m_2, m_3 and $p \in \mathbb{Z}_+$, the set of positive integers. Let $q \in \mathbb{Z}$ be prime. Let $\mathbf{A} \xleftarrow{r} \mathbb{Z}_q^{m_1 \times m_2}$, $\mathbf{S} \xleftarrow{r} \mathbb{Z}_q^{m_3 \times m_2}$ and $\mathbf{E} \leftarrow \chi^{m_1 \times m_3}$. With $\mathbf{S}^T$ denoting the transpose of $\mathbf{S}$, compute $\mathbf{B} = \mathbf{A}\mathbf{S}^T + \mathbf{E}$. The augmented matrix $\mathbf{C} = (\mathbf{A}, \mathbf{B}) \in \mathbb{Z}_q^{m_1 \times (m_2 + m_3)}$ is called the concealer matrix which we will use in our construction. Note that this is a special instance of the LWE problem mentioned in section 2.2. Let $Conceal_\chi(m_1, m_3)$ output a concealer matrix as defined above.

Encoding matrix: Another matrix which is used in our construction is defined as follows: Consider $1 < p \in \mathbb{Z}$ and $0 < m_3 \in \mathbb{Z}$. Let $n = m_3 \cdot \lceil \log(p) \rceil$. Define a vector $\mathbf{P} = (2^0, 2^1, \ldots, 2^{\lceil \log(p) \rceil - 1}) \in \mathbb{Z}^{\lceil \log(p) \rceil}$, where $\lceil x \rceil$ denote the integer that is immediately greater than x. We call $\mathbf{P}$ as the *powers of two vector*. Now define the $n \times m_3$ matrix $\bar{I} = I_{m_3} \otimes \mathbf{P}^T$ which is the tensor product of the identity matrix of order m_3(denoted by I_{m_3}) and the vector $\mathbf{P}$.

Consider an integer $p \geq 2$, which is also a power of 2. Let $m_3 \in \mathbb{Z}$ be such that $n = m_3 \log_2(p)$ is the input length of the function. Let $q \geq 4np$ be the modulus and α be the parameter for the error distribution $\chi = \Psi_\alpha$ of the LWE problem with $\frac{1}{\alpha} \geq 16np$. The rationale for the choice of parameters as above are based on Lemma 3.1 and Theorem 3.1.

For simplicity, we assume that p in section 2.4 is a power of 2 and hence $\lceil \log(p) \rceil = \log(p)$. Now define an encoding function $\rho : \mathbb{Z}_p \to \mathbb{Z}_q$ where $q \geq 4pn$.

The function ρ is defined as

$$\rho(x) = \left\lfloor q \cdot \frac{x}{p} \right\rceil \in \mathbb{Z}_q \tag{2.1}$$

The function also takes matrices as input in a natural manner by acting on each matrix element. The corresponding decoding function $\rho^{-1} : \mathbb{Z}_q \to \mathbb{Z}_p$ is defined as

$$\rho^{-1}(y) = \left\lfloor p \cdot \frac{y}{q} \right\rceil \in \mathbb{Z}_p \tag{2.2}$$

Generation of Injective and Lossy functions: To generate or sample an injective or lossy function, the function generators (both injective and lossy) invokes $Conceal_\chi(n, m_3)$ to construct a concealer matrix $\mathbf{C} = (\mathbf{A}, \mathbf{B}) = (\mathbf{A}, \mathbf{AS}^T + \mathbf{E}) \in \mathbb{Z}_q^{n \times (m_2+m_3)}$ as defined in section 2.4. The trapdoor corresponding to this construction will be $\mathbf{S}$. The injective function (Φ_i) generates the function index $\mathbf{Y} = (\mathbf{A}, \mathbf{B} + \mathbf{M}) \in \mathbb{Z}_q^{n \times (m_2+m_3)}$ with trapdoor $\mathbf{S}$; where $\mathbf{M} = \rho(\bar{I} \pmod p)$.

The lossy function (Φ_l) outputs the concealer matrix $\mathbf{C}$, same as above but there will not be any trapdoor. **Evaluation of Injective and Lossy functions:** F_l takes $(\mathbf{Y}, \mathbf{x})$ as input where $\mathbf{Y}$ is the function index and $\mathbf{x} \in \{0,1\}^n$. The output is $\mathbf{z} = \mathbf{xY} \in \mathbb{Z}_q^{m_2+m_3}$.
If the function index $\mathbf{Y}$ was generated by Φ_i, then

$$\mathbf{z} = ((\mathbf{xA}, \mathbf{x}(\mathbf{B} + \mathbf{M}))) = (\mathbf{xA}, \mathbf{xAS}^T + \mathbf{x}(\mathbf{E} + \mathbf{M})) \tag{2.3}$$

On the contrary, if $\mathbf{Y}$ was generated by Φ_l, then

$$\mathbf{z} = ((\mathbf{xA}, \mathbf{xB})) = (\mathbf{xA}, \mathbf{xAS}^T + \mathbf{xE}) \tag{2.4}$$

Inversion algorithm: F_l^{-1} takes $(\mathbf{S}, \mathbf{z} = (\mathbf{z_1}, \mathbf{z_2}))$ as input and computes

$$\mathbf{v} = \mathbf{z_2} - \mathbf{z_1}\mathbf{S}^T = \mathbf{x}(\mathbf{M} + \mathbf{E}) \tag{2.5}$$

The vector $\mathbf{m} = \rho^{-1}(\mathbf{v}) \in \mathbb{Z}_p^{m_3}$ is taken and $\mathbf{x}$ can be found from the base-2 representation of $\mathbf{m}$.

2.4.1 ABO Trapdoor Functions:

Let a collection of sets $\mathcal{S} = \{S_\lambda\}_{\lambda \in \mathbb{N}}$ be called branches. A collection of (μ, ν)-ABO TDFs with branch $\mathcal{S}$ is given by an ordered triple of probabilistic polynomial time algorithms
$(\Phi_{abo}, G_{abo}, G_{abo}^{-1})$ satisfying the following:

1. Let $b^* \in S_\lambda$, $(\zeta, \eta) \leftarrow \Phi_{abo}$. For $b \neq b^*$,
   $G_{abo}(\zeta, b, \cdot)$ computes $g_{\zeta,b}(\cdot)$ over $\{0,1\}^\mu$ and $G_{abo}^{-1}(\eta, b, \cdot)$ computes $g_{\zeta,b}^{-1}(\cdot)$ and G_{abo}^{-1} is not relevant beyond the range of $g_{\zeta,b}$.

2. $G_{abo}(\zeta, b^*, \cdot)$ computes $g_{\zeta,b^*}(\cdot)$ over $\{0,1\}^\mu$ with range space not exceeding $2^\gamma = 2^{\mu-\nu}$

We have defined the *almost-always* lossy function in section 2.3. Similarly, an *almost-always* variant is applicable here in the ABO construction. If there exists a function satisfying the above conditions with an overwhelming probability, it is called an *almost-always* ABO trapdoor function.

2.4.2 ABO construction from LWE:

Consider the vector $\mathbf{v} = (v_1, v_2, \ldots, v_{m_3}) \in \mathbb{Z}^{m_3}$ and a shift operation defined as $\xi(\mathbf{v}) = (-v_{m_3}, v_1, \ldots, v_{m_3-1})$. Now, we construct a matrix using this shift operation on the chosen vector. The matrix $\mathbf{V} = \Xi(\mathbf{v}) \in \mathbb{Z}^{m_3 \times m_3}$ where the k^{th} row $\mathbf{v_k} = \xi^{(k-1)}(\mathbf{v})$ is the vector $\mathbf{v}$ shifted $(k-1)$ times or the shift operation ξ operated $k-1$ times on $\mathbf{v}$. It can be observed that $\mathbf{V}$ is a full-rank matrix. Consider the *powers of two vector* $\mathbf{P}$ defined in section 2.4. Let $p' = 2pm_3$. The functions ρ and ρ^{-1} are defined in the same way as that in section 2.4 but from $\mathbb{Z}_{p'} \to \mathbb{Z}_q$.Let $r : \{0,1\}^{m_3} \to \mathbb{Z}_q^{n \times m_3}$ defined as $r(\mathbf{v}) = \rho(\Xi(\mathbf{v}) \otimes \mathbf{P^T} \pmod{p'})$. The function generator takes the desired lossy branch ($\mathbf{v}^* \in \{0,1\}^{m_3}$) and generates a concealer matrix $\mathbf{C} \in \mathbb{Z}_q^{n \times (m_2+m_3)}$ but the function index will be $\mathbf{Y} = (\mathbf{A}, \mathbf{B} - r(\mathbf{v}^*))$ with trapdoor being the ordered pair $(\mathbf{S}, \mathbf{v}^*)$.

2.5 The Architecture and Security Models

An encryption scheme providing entity authentication generally consists of the following probabilistic polynomial time(PPT) algorithms.

- **Setup**: Input the security parameter 1^n and outputs the public parameters $Params_{Pub}$.
- **KeyGen**$(1^n, Params_{Pub})$: Input the security parameter 1^n and $Params_{Pub}$ to generate the sender's and receiver's public and secret(private) key pairs, which are represented by (pk_s, sk_s) and (pk_r, sk_r) respectively.
- **Sign and Encrypt**(m, sk_s, pk_r): Create a ciphertext c, for a message m, signed using sk_s and encrypted using pk_r.
- **Decrypt and Verify**(c, pk_s, sk_r): The ciphertext c is decrypted using sk_r and verified using pk_s. Upon successful completion of the process, the output will be m. Otherwise, it will be $\perp$.

Asymmetric encryption schemes providing entity authentication are supposed to be secure in the following security definitions [9].

IND-CCA2 Security:

Security against an adversary capable of performing an adaptive chosen ciphertext attack, also termed IND-CCA2 security, is defined as follows. The IND-CCA2 experiment consists of a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ involved in the steps below described in chronological order.

- **Initiate**: $\mathcal{C}$ runs **KeyGen**$(1^n, Params_{Pub})$ to generate (pk_r^*, sk_r^*) and sends $(pk_r^*, Params_{Pub})$ to $\mathcal{A}$.
- **Query**: $\mathcal{C}$ queries the encryption and decryption oracles adaptively. For valid ciphertexts, the oracle replies with the correct plaintext.
- **Challenge**: $\mathcal{A}$ chooses two messages m_0 and m_1 of the same length, adaptive to the queries and sends it to $\mathcal{C}$. $\mathcal{C}$ chooses $b \underset{\$}{\leftarrow} \{0,1\}$ to encrypt m_b as c^* and sends c^* to $\mathcal{A}$.
- **Adaptive Query**: $\mathcal{A}$ re-initiates the query as in the second stage adaptive to the challenge ciphertext c^* obtained, but $\mathcal{A}$ is forbidden to query the oracles with c^*.
- **Response**: $\mathcal{A}$ outputs b'. $\mathcal{A}$ wins the game if $b = b'$.

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

$$Adv(\mathcal{A}) = \left| Pr\left[b = b'\right] - \frac{1}{2} \right|$$

If the advantage of the adversary is negligible, the scheme is IND-CCA2 secure.

sUF-CMA Security:

The strong Existential Unforgeability under adaptive Chosen Message Attack, abbreviated as sUF-CMA, relates to the following experiment defined by Boneh et. al [10].

- **Setup**: $\mathcal{C}$ runs **KeyGen**$(1^n, Params_{Pub})$ to generate (pk_s^*, sk_s^*) and sends $(pk_s^*, Params_{Pub})$ to $\mathcal{A}$.
- **Signature Queries**: $\mathcal{A}$ issues signature queries $m_1, m_2, \ldots, m_n$. To each query m_i, $\mathcal{C}$ responds with a signature σ_i corresponding to m_i. The queries are made one after other and every query is made adaptive to the previous queries.
- **Response**: $\mathcal{A}$ responds with a message signature pair (m, σ) where $(m, \sigma) \neq (m_i, \sigma_i)$ for $i \in [n]$

The advantage of $\mathcal{A}$ is the probability with which $\mathcal{A}$ wins the above game. A signature scheme is (t, n, ϵ)-strongly existentially unforgeable under adaptive chosen message attack if any t-time adversary $\mathcal{A}$ making at most n queries has no more than ϵ advantage in the above game.

3 Proposed Construction

We propose a hybrid encryption scheme with entity authentication. The construction provides ciphertext integrity and is secure against adaptive chosen ciphertext attack in the standard model. The proposed construction consists of two phases, the registration phase and the communication phase. In the registration phase, the entity Alice registers at the Master Authority (MA) and obtains a user identity. Using that, Alice will communicate with other users in the network in the subsequent communication phase.

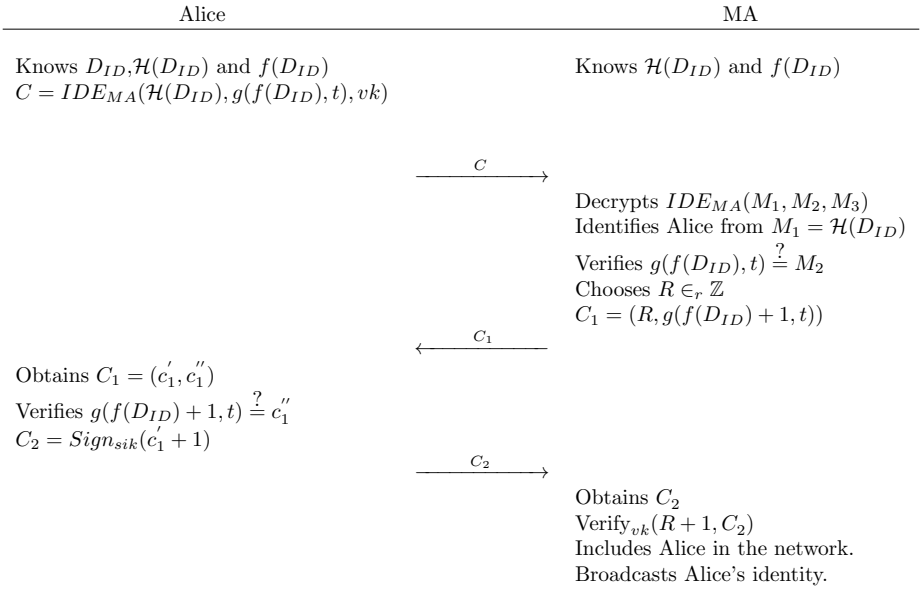
3.1 Registration

This section explains how a new entity (Alice) can register with Master Authority(MA) and become part of the network to communicate with other entities. Without loss of generality, suppose that a user, Alice, has a device identifier (D_{ID}). The following are known to the participating entity, Alice and the Master Authority (MA):

1. $\mathcal{H}(D_{ID})$, a hash value of the (D_{ID})
2. $f(D_{ID})$, a function of D_{ID}
3. A time stamp, t
4. A function g defined on an extended domain formed by the cartesian product $Range(f) \times Domain(t)$.The function g uniquely determines a value after taking $f(D_{ID})$ and t as inputs.

This shared information helps an entity to authenticate themselves to MA. It is to be noted that the value of the time stamp corresponding to each handshake will be different. When the communication initiates, Alice uses an Identity-based encryption scheme (IDE) and computes the encryption of the following: $\mathcal{H}(D_{ID})$, $g(f(D_{ID}, t))$ and the verification key (vk) corresponding to the signing key sik chosen by Alice for the strongly unforgeable signature scheme. Upon receiving the ciphertext, MA identifies Alice from $\mathcal{H}(D_{ID})$ and computes $g(f(D_{ID}), t)$ and compares it with the obtained component for ensuring a thwart against replay attack. Upon successful verification, MA accepts vk but has to ensure that Alice possesses the signing key corresponding to vk. To ensure the same MA chooses a random number R and sends it to Alice along with $g(f(D_{ID}) + 1, t)$ for Alice to verify the origin of the message is from MA. Alice obtains an ordered pair $C_1 = (c_1', c_1'')$. Alice can compute the value of $g(f(D_{ID}) + 1, t)$ since g, t and $f(D_{ID})$ is known to Alice. She then compares whether the computed value matches with c_2'' for a successful verification. After verifying the component and the time stamp, Alice replies with the signature on $(c_1' + 1)$ using the signing key (sik) corresponding to vk. This proves to MA that Alice possesses the signing key corresponding to vk. (Note: For clarity in certain contexts, we use sik for signing key although sk serves the same purpose.)

MA successfully verifies the signature with $R+1$ as the message corresponding to the signature C_2 and updates the user's list by adding Alice's Public parameters $(\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3) = (F_{Alice}, G_{Alice}, \mathcal{H}_{Alice})$ to the repository and broadcasting Alice's User ID (ID_A) and the signature verification key vk to be used in the network. The construction of F, G and h are explained in Section 3.2 and the diagrammatic representation of the registration phase is shown in the Protocol 1.



Protocol 1: Registration

3.2 Communication

This section describes the communication between two entities, Alice and Bob, already registered with MA under the same network. Consider the case where Alice encrypts a message m to Bob using Bob's Public key. Bob's public key is the 3-tuple $(\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3) = (F_{Bob}, G_{Bob}, h_{Bob})$, where F_{Bob} is the lossy trapdoor function, G_{Bob} is the ABO-trapdoor function and h_{Bob} is the hash function chosen by Bob. Since we are using the LWE-based construction, the injective function is the same for all the users, but only the trapdoor varies from user to user. Bob invokes the $Conceal_{\chi}(n, m_3)$ and outputs a function index $F_{Bob} = (A_{Bob}, B_{Bob} + M_{Bob}) \in \mathbb{Z}_q^{n \times (m_2 + m_3)}$ as described in Section 2.4 with the trapdoor S_{Bob}. Consider $\emptyset \neq \mathbf{ID}, \mathbf{K} \subsetneq \{0,1\}^{m_3}$ with $\mathbf{ID} \cap \mathbf{K} = \emptyset$. $\mathbf{ID}$ is the set from which the user identities are assigned. Before starting the encryption process, Bob chooses an element $\mathbf{v}^* \in \mathbf{K}$ to form the ABO function. G_{Bob} corresponds to the ABO function chosen by Bob in which $G_{Bob} = (A_{Bob}, B'_{Bob})$. That is, $G_{Bob} = (A_{Bob}, B_{Bob} - r(\mathbf{v}^*))$ and hence

$G_{Bob} = (A_{Bob}, A_{Bob}S_{Bob}^T + E - r(\mathbf{v}^*))$. $\mathcal{B}_3 = \mathcal{H}_{Bob}$ is a sponge based hash function. The diagrammatic representation of the protocol is shown in Protocol 2.

Alice		Bob
Obtain $\mathcal{B} = \{\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3\}$		
$x \xleftarrow[r]{} \{0,1\}^n$		
Computes $\mathcal{B}_3(x)$		
$C_1 := \mathcal{B}_1(x), C_2 := \mathcal{B}_2(ID_A, x)$		
For the message m, computes $C_3 = \mathcal{E}^{\text{sy}}_{\mathcal{B}_3(x)}(m)$		
$\sigma = Sign(C_1, C_2, C_3)$		
$C = (ID_A, C_1, C_2, C_3, \sigma)$		
	$\xrightarrow{C}$	
		Receives $(ID', C_1'C_2', C_3', \sigma')$
		Obtains $vk \leftrightarrow ID'$
		Verify σ'
		$\mathcal{B}_1^{-1}(C_1) = x'$
		$C_2' \stackrel{?}{=} \mathcal{B}_2(ID', x')$ or $\perp$
		$m := \mathcal{D}^{sym}_{\mathcal{B}_3(x)}(C_3)$ or $m := \perp$

Protocol 2: Encryption and Decryption

3.2.1 Encryption

1. Alice takes F_{Bob} and $\mathbf{x} \xleftarrow[r]{} \{0,1\}^n$ as input and computes

$$C_1 = xF_{Bob} = (xA_{Bob}, x(B_{Bob} + M_{Bob})) \tag{3.1}$$

2. Alice then takes $G_{Bob}, ID_A \in \mathbf{ID}$ (assigned for Alice by MA) and x as input and computes $C_2 = x(A_{Bob}, B'_{Bob} + r(ID_A))$.

3. Alice takes x as input and computes the hash $\mathcal{B}_3(x)$.

4. Now takes the message m and encrypts it using the symmetric encryption algorithm with the key being $\mathcal{B}_3(x)$. The ciphertext is labelled C_3.

5. Now having the triplet (C_1, C_2, C_3), Alice computes $Sign(C_1, C_2, C_3) = \sigma$.

6. Finally $C = (ID_A, C_1, C_2, C_3, \sigma)$ is the ciphertext sent to Bob.

3.2.2 Decryption

1. After receiving the ciphertext C, Bob will parse it into five components.

2. Bob obtains the verification key corresponding to the first component (ID_A) and verifies the signature (σ) in the fifth component.

3. Upon successful verification, Bob extracts x from C_1 using the trapdoor S_{Bob}

4. Bob will reconstruct C_2 using the x obtained and the ID_A in the first component.

5. If the reconstructed value matches the C_2 value obtained from Alice, Bob constructs $\mathcal{B}_3(x)$ and decrypts C_3 using the same as the decryption key and obtains m.

3.3 Correctness of the Scheme

Let us verify the correctness of the registration phase first.

- Since both Alice and MA knows $\mathcal{H}(D_{ID})$, MA can check which user does the first component, M_1 in the triplet maps to and can identify Alice is the entity.
- From the shared information about g and $f(D_{ID})$, MA can compute the correct value of $g(f(D_{ID}), t)$ and compare with the second component M_2.
- By the same logic as the previous point, Alice also authenticates the message from MA by checking whether the value of $c_1^{''}$ matches with the one computed by Alice.
- For the final step, Alice possesses the signing key sik corresponding to vk as well as the signature is on the value $c_1^{'} + 1 = R + 1$. So the final verification also will turn out successful.

Having established the correctness of the registration phase, let us move on to the communication phase. First, let us prove the correct inversion of the LTDF function. That is, we show that $x' = x$ in protocol 2. We use the following lemma from [26] in the process.

Lemma 3.1. *Let $0 < n, p, m_3 \in \mathbb{Z}$. Let $q \geq 4pn$, let $\frac{1}{\alpha} \geq 8p(n+g)$ for some $g > 0$ and let $\chi = \bar{\Psi}_\alpha$, then $\forall x \in \{0,1\}^n$, every element of $\frac{xE}{q}$ belongs to $(\frac{-1}{4p}, \frac{1}{4p})$ except with probability $m_3 \cdot 2^{-g}$ over the choice of $E \leftarrow \chi^{n \times m_3}$.*

Now we prove the correctness of $\mathcal{B}_1^{-1}(C_1)$ with the help of a theorem by Peikert and Waters [26]. With parameters being the same as in Section 2.4, the theorem is as follows:

Theorem 3.1. *Let $q \geq 4pn$ and $\chi = \bar{\Psi}_\alpha; \frac{1}{\alpha} \geq 16np$, then the correct inversion of the function $\mathcal{B}_1$ follows.*

Proof. Suppose that Bob obtained $C_1 = (k_1, k_2) = (xA_{Bob}, x(B_{Bob} + M))$. For

obtaining x from C_1, Bob computes

$$\begin{aligned}
k &= k_2 - k_1 S_{Bob}^T && \text{(Eq. 2.5)}\\
&= xB_{Bob} + xM - xA_{Bob}S_{Bob}^T \\
&= xA_{Bob}S_{Bob}^T + x(E+M) - xA_{Bob}S_{Bob}^T \\
&= xE + xM \\
&= xE + x\rho(\bar{I} \mod p) && \text{(from 2.4)}\\
&= xE + x\left\lfloor q \cdot \frac{\bar{I}}{p} \right\rceil && \text{(Eq.2.1)}
\end{aligned}$$

Let $I_1 = \left[\frac{-1}{2}, \frac{1}{2}\right] \subset \mathbb{R}$ and $T = \rho^{-1}(k)$.

$$\begin{aligned}
T &= \left\lfloor p \cdot \frac{xE + x\left\lfloor q \cdot \frac{\bar{I}}{p} \right\rceil}{q} \right\rceil && \text{(Eq.2.2)}\\
&\in \left\lfloor p \cdot \frac{xE}{q} + \frac{p}{q} \cdot x\left(I_1^{n\times m_3} + q \cdot \frac{\bar{I}}{p}\right) \right\rceil \\
&\in \left\lfloor \frac{1}{2} \cdot I_1^{m_3} + \frac{p}{q} \cdot xI_1^{n\times m_3} + x\bar{I} \right\rceil && \text{(Lemma 3.1)}
\end{aligned}$$

We have, by the triangle inequality, $xI_1^{n\times m_3} \in n \cdot I_1^{m_3}$. Also by the choice of parameters from section 2.4, we have $q \geq 4pn \implies \frac{p}{q} \leq \frac{p}{4pn}$. Therefore,

$$\begin{aligned}
T &\in \left\lfloor \frac{1}{2} \cdot I_1^{m_3} + \frac{1}{4} \cdot I_1^{m_3} + x\bar{I} \right\rceil \\
T &\in \left\lfloor \frac{3}{4} \cdot I_1^{m_3} + x\bar{I} \right\rceil \\
T &= x\bar{I}
\end{aligned}$$

Now let $k' = x\bar{I} \pmod p$. The binary representation of k' will yield x. □

The trapdoor for the function G_{Bob} is available and hence the correct inversion is possible. But we only need to recreate the same for verification of ciphertext integrity. We already proved that $x' = \mathcal{B}_1^{-1}(C_1) = x$. Now, $ID = ID' \implies C_2' = B_2(ID', x') = B_2(ID, x) = C_2$ and the verification will be successful.
The symmetric cryptosystem to create C_3 will be chosen according to the security requirements. But, irrespective of the symmetric cryptosystem used, the key used by Alice to encrypt m as $\mathcal{E}^{sym}_{\mathcal{B}_3(x)}(m) = C_3$ will be $\mathcal{B}_3(x)$. Since $\mathcal{B}_1^{-1}(C_1) = x$, we have $\mathcal{D}^{sym}_{\mathcal{B}_3(x')}(C_3) = \mathcal{D}^{sym}_{\mathcal{B}_3(x)}\left(\mathcal{E}^{sym}_{\mathcal{B}_3(x)}(m)\right) = m$. Hence we prove the correctness of the proposed construction.

3.3.1 Entity Authentication

The entity authentication techniques used in the scheme are as follows.

- The verification key corresponding to the user's identity(ID_A) in the first component of the ciphertext should verify the fifth component(σ) as the signature of the corresponding entity on the second third and fourth (C_1, C_2, C_3) components.
- The reconstruction of $B_2\left(ID', B_1^{-1}(C_1)\right)$ should match with the third component (C_2') of the ciphertext.

4 Security Analysis

In this section, we prove that the system is secure against adaptive chosen ciphertext attack in the standard model. We also prove that the scheme is non-malleable and secure against replay attacks and man-in-the-middle attacks.

4.1 IND-CCA2 Security

We prove the security of the proposed construction in the standard model using the game-hopping technique. Consider the indistinguishability against adaptive chosen ciphertext attack (IND-CCA2) experiment defined in Section 2.5. We prove that our construction is secure against an adversary capable of performing an IND-CCA2 attack.
We are employing a game-hopping technique model for the proof. Before defining the hybrids, we are quoting two very important lemmas proposed in [14]

Lemma 4.1. *If Y takes at most 2^r possible values and X is any random variable, then*

$$\tilde{H}_\infty(X|Y) \geq H_\infty(X) - r$$

Definition 4.1. *A collection $\mathbb{H}$ of functions from $\{0,1\}^n \to \{0,1\}^l$ is an average case (n, k, l, ϵ) strong extractor if for all the pairs of random variables (X, Y) such that $X \in \{0,1\}^n$ and $\tilde{H}_\infty(X|Y) \geq k$, it holds that for $\mathcal{H} \leftarrow \mathbb{H}$ and $r \leftarrow \{0,1\}^l$,*

$$\Delta((\mathcal{H}, \mathcal{H}(X), Y), (\mathcal{H}, r, Y)) \leq \epsilon$$

Lemma 4.2. *Let X and Y be random variables such that $x \in \{0,1\}^n$ and $\tilde{H}_\infty(X|Y) \geq k$. Let $\mathbb{H}$ be a family of universal hash functions from $\{0,1\}^n \to \{0,1\}^l$, where $l \leq k - 2\log_2\left(\frac{1}{\epsilon}\right)$. Then $\mathbb{H}$ is an average case (n, k, l, ϵ)-strong extractor.*

We define the function triplets $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ for the following theorem.

- $\mathcal{G}$, also known as the key generation algorithm, takes a security parameter as input and outputs an ordered pair (*pk,sk*) where the first component is the public key and the second is the secret key.

- $\mathcal{E}$, the encryption algorithm takes pk and $n \in \mathcal{M}$ as input and outputs a ciphertext c. Here, $\mathcal{M}$ is the message space, a.k.a the plaintext space.
- $\mathcal{D}$ takes the ciphertext c and the secret key sk as input and outputs $m \in \mathcal{M}'$ where $\mathcal{M}' = \mathcal{M} \cup \{\perp\}$

Theorem 4.1. *The triplet of algorithms key generation, encryption and decryption* $(\mathcal{G}, \mathcal{E}, \mathcal{D})$*gives an IND-CCA2-secure cryptosystem.*

Proof. We engage the game-hopping technique to prove the claim. As we already have proven the correctness of the scheme, we will now set up the paraphernalia required to design the experiments. Each experiment contains primarily three algorithms, which we will alter according to the experiment. They are

- Set: The algorithm runs and outputs a public key pk.
- Dec: This is an algorithm which works as a decryption oracle. This one takes a ciphertext c as input and outputs a value m from the message space or outputs $\perp$.
- Chal: This algorithm corresponds to the challenge phase where the adversary provides two values m_0 and m_1 which Chal takes as input and outputs c^*.

Since there is no change in the signature and verification key pairs throughout a session, it will be the same throughout all the hybrids. Now we define the hybrids one by one and will prove that each hybrid is either computationally or statistically indistinguishable from the preceding one. For the notation convenience, we define that

- Hybrid_1: This hybrid is the actual CCA2 experiment where the adversary offers two messages and the challenger challenges the adversary in return with the encryption of only one random message chosen from them. Formally, Set_1 gets $(pk,sk) \leftarrow \mathcal{G}$ and outputs pk. $\text{Dec}_1(c) \rightarrow \mathcal{D}(sk, c)$ and $\text{Chal}_1(m_0, m_1) \rightarrow \mathcal{E}(pk, m_b)$. In addition, the lossy branch will be 0^ν and the inversion will be done using the injective function trapdoor.
- Hybrid_2: The change is in Dec_2. In the ciphertext $c = (ID, C_1, C_2, C_3, \sigma)$ if $ID = v^*$ output $\perp$. else return $\text{Dec}_1(c)$
- Hybrid_3: The change is in Set_3. The lossy branch of the ABO function is v^* instead of 0^ν. In $\mathcal{G}$, $(\zeta', \eta') \leftarrow \Phi_{abo}(v^*)$.
- Hybrid_4: The change is in Dec_4. The witness recovery was previously made using the trapdoor for the injective function. But here, in this hybrid, the trapdoor for the ABO inversion function and trapdoor is used. Mathematically, $x = G_{abo}^{-1}(\eta', ID, C_2)$. Here the verification check will be $C_1 \stackrel{?}{=} F_l(\zeta, x)$.

- Hybrid$_5$: The change is in Set$_5$ and it is that the ordered pair $(\zeta, \eta) \leftarrow \Phi_i$ is replaced with $(\zeta, \perp) \leftarrow \Phi_l$. that is the injective function is replaced with a lossy function. The setup in Hybrid$_5$ is as follows: $(\zeta', \eta') \leftarrow \Phi_{abo}(v^*)$, $(\zeta, \eta) \leftarrow \Phi_l, \mathcal{H} \leftarrow \mathcal{H}$ and outputs $pk = (\zeta, \zeta', h)$.

- Hybrid$_6$: The change is in Chal$_6$. The C_3 component is replaced with a uniform random value of the same length. The final description will be $x \leftarrow \{0,1\}^{\mu}$, $C_1 = F_l(\zeta, x), C_2 = G_{abo}(\zeta', v^*, x)$, $C_3 \underset{r}{\leftarrow} \{0,1\}^{l}, \sigma = \text{Sign}(sik_\sigma, (C_1, C_2, C_3))$ and output $c = (v^*, C_1, C_2, C_3, \sigma)$. Observe that the adversary's view is the same for either choice from m_0 and m_1 in Hybrid$_6$.

Claim 4.1. *The adversary has only a negligible advantage in computationally distinguishing between Hybrid$_1$ and Hybrid$_2$.*

Proof of claim: It can be easily observed that the adversary's view is unaltered unless the event mentioned in Hybrid$_2$ happens. That is, unless the first component of the ciphertext matches with the lossy branch chosen by the owner of the public key, both events are the same. Note that we are choosing the ID and v^* from $\{0,1\}^{m_3}$. So we are choosing values from a set of cardinality 2^{m_3}. Consider a finite set S having cardinality K. Let $\alpha \in_r S$ be taken, noted down and replaced. Now take $\beta \in_r S$. Since the events are independent and identically distributed, $Pr[|\alpha - \beta| = 0] = \frac{1}{K}$.
Here in our case, if we denote H_i to be the event in Hybrid$_i$, $|Pr[H_1] - Pr[H_2]| = \frac{1}{2^{m_3}}$.
Hence claim 4.1 holds.

Claim 4.2. *The adversary has only a negligible advantage in computationally distinguishing between Hybrid$_2$ and Hybrid$_3$.*

Proof of claim: To prove this claim, we are using a PPT simulator $\mathcal{T}$ to interact in the hidden lossy branch experiment of the ABO collection. Suppose that $\mathcal{T}$ gives the experiment two branches 0^ν and v^* and obtains ζ' as the output of either $\Phi_{abo}(0^\nu)$ or $\Phi_{abo}(v^*)$. Now the process works as follows $\mathcal{T}$ implements Set and $(\zeta, \eta) \leftarrow \Phi_i, \mathcal{H} \leftarrow \mathbb{H}$ and outputs $pk = (\zeta, \zeta', h)$. The algorithms Dec and Chal are unchanged between both the hybrids. Now observe that the view generated by $\mathcal{T}$ is exactly Hybrid$_3$ provided $\zeta \leftarrow \Phi_{abo}(v^*)$ and is of Hybrid$_2$ when $\zeta \leftarrow \Phi_{abo}(0^\nu)$ and hence the claim.

Claim 4.3. *The adversary has only a negligible advantage in distinguishing between Hybrid$_3$ and Hybrid$_4$.*

Proof of claim: The implementation of the algorithm Dec is the sole difference between Hybrid$_3$ and Hybrid$_4$. In Hybrid$_3$, the injective function is used for witness recovery, whereas in Hybrid$_4$, ABO function is used for witness recovery. If $ID = v^*$, Dec outputs $\perp$. So assume that $ID \neq v^*$. We can also observe that both implementations check whether $C_1 \stackrel{?}{=} F_l(\zeta, x) = f_\zeta(x)$ and $C_2 \stackrel{?}{=} G_{abo}(\zeta', ID, x) = g_{\zeta', ID}(x)$.

We now show that the value of x (if exists) is unique in both Hybrid$_3$ and Hybrid$_4$. $(\zeta, \eta) \leftarrow \Phi_i$ and
$(\zeta', \eta') \leftarrow \Phi_{abo}(v^*)$. Hence it is evident that $f_\zeta(\cdot)$ and $g_{\zeta', ID}(\cdot)$ are both injective, and hence x is unique since it satisfies both the decryptions and both the verification simultaneously, though obtained in both the hybrids in different ways. Hence the claim.

Claim 4.4. *The adversary has only a negligible advantage in distinguishing between Hybrid$_4$ and Hybrid$_5$.*

Proof of claim: To prove this claim, we engage a PPT simulator $\mathcal{T}$ once again. Let $\zeta \leftarrow \Phi_i$ be a function index, then $\mathcal{T}$ simulates Hybrid$_4$ and if $\zeta \leftarrow \Phi_l$, then $\mathcal{T}$ simulates Hybrid$_5$.
$\mathcal{T}$ takes ζ as input and implements Set, Dec and Chal as in Hybrid$_4$. $(\zeta', \eta') \leftarrow \Phi_{abo}(v^*), \mathcal{H} \leftarrow \mathbb{H}$ and $pk = (\zeta, \zeta', h)$. So $\mathcal{T}$ knows the ABO trapdoor η', which is used for witness recovery in both Hybrid$_4$ and Hybrid$_5$. Without knowing the trapdoor, η corresponding to the function index ζ, $\mathcal{T}$ is able to implement successfully both Hybrid$_4$ and Hybrid$_5$. Hence by the indistinguishability of injective and lossy functions, the claim follows.

Claim 4.5. *The adversary has only a negligible advantage in statistically distinguishing between Hybrid$_5$ and Hybrid$_6$.*

Proof of claim: Assume that except for the hash function h and x used by Chal, all the randomness is fixed. Note that we are now on the lossy function instead of the injective function and on the lossy branch of the ABO function, also. Hence $F_l(\zeta, \cdot) = f_\zeta(\cdot)$ has an image size not exceeding $2^{\mu-\nu}$ and $G_{abo}(\zeta', v^*, \cdot) = g_{\zeta', v^*}(\cdot)$ has an image size not more than $2^{\mu-\nu'}$. Hence the random variables (c_1^*, c_2^*) can take at most $2^{\gamma+\gamma'} \leq 2^{\mu-\kappa}$
By lemma 4.1, we have

$$\tilde{H}_\infty(x|(c_1^*, c_2^*)) \geq H_\infty(x) - (\mu - \kappa) = \mu - (\mu - \kappa) = \kappa$$

Now by the assumption that $l \leq \kappa - 2\log_2\left(\frac{1}{\epsilon}\right)$ and lemma 4.2, we have

$$\Delta((c_1^*, c_2^*, h, \mathcal{E}^{\text{sy}}_{h(x)}(m)), (c_1^*, c_2^*, h, r')) \leq \epsilon = \text{negl}(\lambda)$$

where r' is independent of all other variables. Hence the claim. □

4.2 Unforgeability

We do not confine ourselves to any specific signature scheme in our construction. We urge the user to use a strongly unforgeable signature scheme. The sUF-CMA security of the scheme as defined in Section 2.5 depends on the security of the signature scheme used [9]. So it is evident that our scheme, if used as directed, provides sUF-CMA security.

4.3 Secure against Malleability

From the first component, the decryptor will get the sender's identity and can obtain the verification key of the signature corresponding to the associated user. If the fifth component (σ') is manipulated, the verification key will not match. Hence, σ' will fail the verification procedure. Similarly, any change in the components C_1, C_2 and C_3 will also result in the abortion of the session.
Consider the adversary changing the first component (user ID) and replacing the fifth component (σ') with a signature corresponding to the replaced user ID. This can result in the adversarial action being undetected in the above-mentioned case. But there is a stage where component $\mathcal{B}_2$ is being recreated and cross-checked to verify whether it is the same as C_2'. Alice created C_2 with her identity along with the value x. So without knowing x, the adversary cannot alter C_2, and hence the change made by the adversary on the signature and ID will be detected in the recreating stage. Hence we conclude that the scheme is non-malleable.

4.4 Secure against Man-in-the-Middle Attack

Suppose that an adversary is trying to perform Man-in-the-Middle Attack on the proposed construction. We list the possibilities of the adversary and why the adversary will not emerge successful in each case.

- The adversary can replace $C = IDE_{MA}(h(D_{ID}), g(f(D_{ID}), t), vk)$ with $C' = (h(D'_{ID}), g(f(D_{ID}), t), vk)$, formed using the hash of his device id. Since the MA verifies C' by checking whether $M_2 \stackrel{?}{=} g(f(D'_{ID}), t)$, it will turn out to be a mismatch if the adversary doesn't change the second component (originally $g(f(D_{ID}), t)$). Once the adversary changes the component M_2 also to $g(f(D'_{ID}), t)$, it will turn out to be a legitimate registration of the adversary and not an impersonation as user Alice.

- The adversary can try to impersonate as MA to Alice by sending a random ordered pair C_1' instead of C_1. Since the second component of the ordered pair is a function involving $f(D_{ID})$, known only to Alice and MA, this attempt will not pass the verification process.

- If the adversary attempts to send a random value instead of C_2 to MA, it will not be a successful attempt because for successful verification, it should be the signature on $R+1$ using the signing key sik known only to Alice.

- In the communication phase, each user verifies the signature of the other user using the verification key broadcasted by MA.

Hence the proposed construction is secure against a man-in-the-middle attack.

4.5 Secure against Replay Attack

Suppose that the adversary is trying to replay a previous message to impersonate as Alice to MA.

- Each of the values $C = IDE_{MA}(h(D_{ID}), g(f(D_{ID}), t), vk)$ and $C_1 = (R, g(f(D_{ID}) + 1, t))$ contains a time stamp and hence the replay will be an unsuccessful attempt.
- The components C_1 and $C_2 = Sign_{sik}(R+1)$ contain a fresh and randomly chosen value R and hence the replay can be identified.

Thus we can conclude that the scheme is secure against Replay attacks.

4.6 Forward Secrecy

A new user will be provided with a fresh user identity. The verification key of the signature will also be made public. Any user who comes into the network afresh won't be able to learn any communication in the network which he is not part of. Similarly, a user who leaves the network will be removed from the repository and hence will be forbidden from observing communication inside the network. Let us examine the different scenarios in which a withdrawn adversary can make an impact inside the system are addressed and mitigated one by one as follows.

- Suppose that a user who is withdrawing from the system. The MA immediately removes the ID and verification key of the user from the list of valid users. So even if the removed user Alice tries to communicate with a valid user Bob, the ciphertext $C = (ID_A, C_1, C_2, C_3, \sigma)$ contains ID_A, which is invalid when Bob checks. So the communication will not be possible.
- A newly joined user will not be assigned the user ID of a previously removed user. Hypothetically assume that a malicious user assumes the user ID of a previous user and contacting MA to be a part of the network again. In this case, there is a very negligible probability that the malicious user also choose the same signature scheme with the same signing key. Thus failing to impersonate as the previous user to MA.
- Since all the new users are given a fresh user ID and chooses a new signature with a fresh signing key, the previous user cannot sign and impersonate as the new user. So we rule out the possibility of a compromise in this regard also.

Thus we prove that the scheme provides forward secrecy.

4.7 Post Quantum Security

Oded Regev [27] showed that the LWE problem is as hard as some standard worst-case lattice problems for quantum algorithms. A modified version of the main theorem by Regev is given by Peikert and Waters in [26], which can be interpreted as follows.

Theorem 4.2. *Let $\alpha(d) \in (0,1)$ and $q(d)$ be a prime number such that $\alpha \cdot q > 2\sqrt{d}$. There is a quantum polynomial time reduction from solving either SIVP or GapSVP problems in the worst case to solving $LWE_{q,\bar{\Psi}_\alpha}$*

It can be observed that the security of our scheme is established in such a way that the scheme is vulnerable to an adversary capable of solving the Learning With Errors (LWE) problem. By virtue of theorem 4.2, it is clear that solving an LWE instance is as hard as solving some hard lattice problems like the Shortest Vector Problem (SVP) or the approximate Shortest Vector Problem (GapSVP). In literature, there are no successful algorithms available that solves the above problems and hence our construction is post-quantum secure.

4.8 Comparison with similar schemes

In this section, we compare our scheme with some of the similar schemes in literature regarding the security they provide and the properties they possess. Most of the schemes similar to our scheme are based on elliptic curve pairings. So, in Comparison 1, we compare our construction with some of the recent or standard schemes on the basis of the security properties, whether it involves pairing computation, whether it provides post-quantum security etc. To resolve ambiguity, by privacy, we mean that the original identity of the user is never shared in the protocol. Instead, an alias identity or a function output of the device identifier will be used in the protocol. The computation assumptions which the schemes are based on are also listed in the same table.

Scheme	IND-CCA2	EUF-CMA	Privacy/ Anonimity	Pairing free	Post-Quantum Security	Computation Assumption
Xiaoguang Liu *et al.*[23]	✓	✓	✓	✓	✗	IFP
YW Zhou *et al.*[36]	✓	✓	✓	✓	✗	CDH, DLP
Fuxiao Zhou *et al.*[35]	✓	✓	✗	✗	✗	DDH-CDH
Insaf Ullah *et al.*[30]	✓	✓	✓	✓	✗	DLP
Ahmed Elkhalil *et al.*[15]	✓	✓	✓	✓	✗	CDH
Mutaz Elradi S Saeed *et al.*[28]	✓	✓	✓	✗	✗	q-BDHIP
Bo Zhang*et al.* [33]	✓	✓	✓	✓	✗	ECDLP
Ikram Ali *et al.*[3]	✓	✓	✓	✗	✗	q-BDHIP,q-SDH
Aravind Vishnu *et al.* [4]	✓	✓	✗	✗	✗	VDP
ALGAES	✓	✓	✓	✓	✓	LWE

Comparison 1: Properties

The abbreviations used in Comparison 1 are listed below.

We compared our scheme with some of the existing similar schemes for security features like confidentiality, authentication, integrity, privacy, non-repudiation, traceability, unlinkability, resistance to replay attacks and resis-

Notation	Description
DDH	Decisional Diffie-Hellman
CDH	Computational Diffie-Hellman
DLP	Discrete Logarithm Problem
IFP	Integer Factorisation Problem
LWE	Learning With Errors
ECDLP	Elliptic Curve DLP
q-BDHIP	q-Bilinear Diffie-Hellman Inversion Problem
q-SDH	q-Strong Diffie Hellman
VDP	Vector Decomposition Problem

tance to impersonation represented in Comparison 2 using S-1, S-2, S-3, S-4, S-5, S-6, S-7, S-8 and S-9 respectively.

Scheme	S-1	S-2	S-3	S-4	S-5	S-6	S-7	S-8	S-9
Fuxiao Zhou *et al.*[35]	✓	✓	✓	✗	✓	✗	✓	✗	✓
Ahmed Elkhalil *et al.*[15]	✓	✓	✓	✓	✓	✓	✓	✗	✓
Ikram Ali *et al.*[3]	✓	✓	✓	✓	✓	✓	✓	✓	✓
Fagen Li and Pan Xiong [22]	✓	✓	✓	✗	✓	✗	✓	✗	✓
ALGAES	✓	✓	✓	✓	✓	✓	✓	✓	✓

Comparison 2: Security feature comparison

5 Conclusion

The article proposes a hybrid encryption algorithm based on lossy trapdoor functions. The conventional lossy trapdoor based constructions uses XOR function as the symmetric encryption. We introduced a method to incorporate any symmetric encryption scheme into the hybrid construction instead of simple XOR operation. This makes the proposed construction stand out from similar constructions. The choice for the symmetric encryption scheme makes the system compatible to various user requirements. In our construction, we allowed signature reuse, resulting in improved efficiency. The scheme is proved to be IND-CCA2 secure in the standard model using game-hopping technique and it performs witness recovery. The strongly existential unforgeable (sUF-CMA) signature scheme under chosen message attack ensures ciphertext integrity and unforgeability. Typically in hybrid encryption schemes, there is no entity authentication, but, we incorporated an entity authentication technique in our hybrid construction. The hard problem that holds the scheme is the Learning With Errors problem on lattices which is post-quantum secure. Therefore, the construction is an efficient post-quantum IND-CCA2 secure hybrid encryption with entity authentication.

Acknowledgement: The authors would like to thank Prof. Veni Madhavan

for his valuable suggestions.

References

[1] Miklós Ajtai. Generating hard instances of the short basis problem. In *International Colloquium on Automata, Languages, and Programming*, pages 1–9. Springer, 1999.

[2] Navid Alamati, Hart Montgomery, Sikhar Patranabis, and Arnab Roy. Minicrypt primitives with algebraic structure and applications. *Journal of Cryptology*, 36(1):1–106, 2023.

[3] Ikram Ali, Tandoh Lawrence, Anyembe Andrew Omala, and Fagen Li. An efficient hybrid signcryption scheme with conditional privacy-preservation for heterogeneous vehicular communication in vanets. *IEEE Transactions on Vehicular Technology*, 69(10):11266–11280, 2020.

[4] SS Aravind Vishnu, I Praveen, and M Sethumadhavan. An IND-CCA2 Secure Certificateless Hybrid Signcryption. *Wireless Personal Communications*, 119(4):3589–3608, 2021.

[5] Mihir Bellare, Dennis Hofheinz, and Scott Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 1–35. Springer, 2009.

[6] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pages 62–73, 1993.

[7] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Keccak. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 313–314. Springer, 2013.

[8] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. Sponge functions. In *ECRYPT hash workshop*, volume 2007. Citeseer, 2007.

[9] Tor E Bjørstad and Alexander W Dent. Building better signcryption schemes with tag-kems. In *Public Key Cryptography-PKC 2006: 9th International Conference on Theory and Practice in Public-Key Cryptography, New York, NY, USA, April 24-26, 2006. Proceedings 9*, pages 491–507. Springer, 2006.

[10] Dan Boneh, Emily Shen, and Brent Waters. Strongly unforgeable signatures based on computational diffie-hellman. In *Public Key Cryptography-PKC 2006: 9th International Conference on Theory and Practice in Public-Key Cryptography, New York, NY, USA, April 24-26, 2006. Proceedings 9*, pages 229–240. Springer, 2006.

[11] Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-kyber: a cca-secure module-lattice-based KEM. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 353–367. IEEE, 2018.

[12] Wouter Castryck and Thomas Decru. An efficient key recovery attack on sidh (preliminary version). *Cryptology ePrint Archive*, pages Paper–2022, 2022.

[13] Kunal Dey, Sumit Kumar Debnath, Pantelimon Stănică, and Vikas Srivastava. A post-quantum signcryption scheme using isogeny based cryptography. *Journal of Information Security and Applications*, 69:103280, 2022.

[14] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM journal on computing*, 38(1):97–139, 2008.

[15] Ahmed Elkhalil, Jiashu zhang, Rashad Elhabob, and Nabeil Eltayieb. Poosc: Provably online/offline signcryption scheme for vehicular communication in vanets. *Computing*, 105(11):2539–2561, 2023.

[16] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. *Journal of cryptology*, 26(1):80–101, 2013.

[17] Bei Gong, Yong Wu, Qian Wang, Yu-heng Ren, and Chong Guo. A secure and lightweight certificateless hybrid signcryption scheme for internet of things. *Future Generation Computer Systems*, 127:23–30, 2022.

[18] Vipul Goyal, Omkant Pandey, and Silas Richelson. Textbook non-malleable commitments. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 1128–1141, 2016.

[19] Shuichi Katsumata, Toi Tomita, and Shota Yamada. Direct computation of branching programs and its applications to more efficient lattice-based cryptography. *Designs, Codes and Cryptography*, 91(2):391–431, 2023.

[20] Jean Belo Klamti and M Anwarul Hasan. A code-based hybrid signcryption scheme. *Journal of Mathematical Cryptology*, 17(1):20220002, 2023.

[21] Chung Ki Li, Guomin Yang, Duncan S Wong, Xiaotie Deng, and Sherman SM Chow. An efficient signcryption scheme with key privacy and its extension to ring signcryption. *Journal of Computer Security*, 18(3):451–473, 2010.

[22] Fagen Li and Pan Xiong. Practical secure communication for integrating wireless sensor networks into the internet of things. *IEEE Sensors Journal*, 13(10):3677–3684, 2013.

[23] Xiaoguang Liu, Ziqing Wang, Yalan Ye, and Fagen Li. An efficient and practical certificateless signcryption scheme for wireless body area networks. *Computer Communications*, 162:169–178, 2020.

[24] Puliparambil Megha Mukundan, Sindhu Manayankath, Chungath Srinivasan, and Madathil Sethumadhavan. Hash-one: a lightweight cryptographic hash function. *IET Information Security*, 10(5):225–231, 2016.

[25] Chris Peikert et al. A decade of lattice cryptography. *Foundations and trends in theoretical computer science*, 10(4):283–424, 2016.

[26] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. *SIAM Journal on Computing*, 40(6):1803–1844, 2011.

[27] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):1–40, 2009.

[28] Mutaz Elradi S Saeed, Qingyin Liu, GuiYun Tian, Bin Gao, and Fagen Li. Hoosc: heterogeneous online/offline signcryption for the internet of things. *Wireless Networks*, 24:3141–3160, 2018.

[29] Victor Shoup. Using hash functions as a hedge against chosen ciphertext attack. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 275–288. Springer, 2000.

[30] Insaf Ullah, Abdullah Alomari, Noor Ul Amin, Muhammad Asghar Khan, and Hizbullah Khattak. An energy efficient and formally secured certificate-based signcryption for wireless body area networks with the internet of things. *Electronics*, 8(10):1171, 2019.

[31] Fenghe Wang, Yupu Hu, and Chunxiao Wang. Post-quantum secure hybrid signcryption from lattice assumption. *Applied Mathematics & Information Sciences*, 6(1):23–28, 2012.

[32] Xiaopeng Yang, Hao Cao, Weichun Li, and Hejun Xuan. Improved lattice-based signcryption in the standard model. *IEEE Access*, 7:155552–155562, 2019.

[33] Bo Zhang, Zhongtian Jia, and Chuan Zhao. An efficient certificateless generalized signcryption scheme. *Security and Communication Networks*, 2018, 2018.

[34] Yuliang Zheng. Digital signcryption or how to achieve cost (signature & encryption) $\ll$ cost (signature)+ cost (encryption). In *Annual international cryptology conference*, pages 165–179. Springer, 1997.

[35] Fuxiao Zhou, Yanping Li, and Yong Ding. Practical v2i secure communication schemes for heterogeneous vanets. *Applied Sciences*, 9(15):3131, 2019.

[36] YW Zhou, B Yang, and WZ Zhang. Provably secure and efficient certificateless generalized signcryption. *Chinese Journal of Computers*, 39(3):543–551, 2016.