7-2019

Location based encryption

Tran Viet Xuan PHUONG
University of Wollongong

Willy SUSILO
University of Wollongong

Guomin YANG
Singapore Management University, gmyang@smu.edu.sg

Jun YAN

Dongxi LIU

Location Based Encryption

Tran Viet Xuan Phuong[1,2(✉)], Willy Susilo[1], Guomin Yang[1], Jun Yan[1], and Dongxi Liu[2]

[1] Institute of Cybersecurity and Cryptology,
School of Computing and Information Technology,
University of Wollongong, Wollongong, Australia
{txuan,wsusilo,gyang,jyang}@uow.edu.au
[2] Data61, CSIRO, Syndey, Australia
Dongxi.Liu@data61.csiro.au

Abstract. We first propose a 2D Location Based Encryption (LBE) scheme, where the setting includes a geography center system and the 2D triangle area including the set of locations. A user joining in the system is provided with a pre-arranged key, which belongs to her/his location. If the user's location is belonging to this area, he/she can decrypt the message. Our proposed scheme achieves a constant ciphertext size in encryption algorithm and decryption cost. Beyond the 2D-LBE scheme, we explore the 3D-LBE scheme; whereby the location is set up in the 3D dimensions. This proposed scheme is an extension of 2D-LBE scheme, which the ciphertext is also constant. Both two schemes are proved in the selective model under the decisional $\ell-$wBDHI assumption.

Keywords: 2D/3D · Location based encryption · Constant ciphertext size · w-lBDHI

1 Introduction

We consider the scenario where there is a geography center system and the 2D triangle area including the set of locations. Each location comprises the X and Y coordinator. When a user joining in the system, he/she is provided a pre-arranged key, which belongs to her/his location. At some point, the center wants to broadcast a message to a specific triangle area, which user's location belonging to this area can decrypt the message. In such a way, any other users located outside specific area cannot learn the information. In addition, according to Fig. 1, we require the user's X and Y coordinate to belong to the distance NE, NW respectively, in the NEWS triangle area if the decryption of message is processed.

The solution is that the center encrypts the message by embedding the set of locations from N to E, and from N to W, then it broadcast the ciphertext to the area NEWS. However, when the broadcasting is adjusted to cover larger area of triangle NESW, the size of the ciphertext will be increased. This requires a

J. Jang-Jaccard and F. Guo (Eds.): ACISP 2019, LNCS 11547, pp. 21–38, 2019.
https://doi.org/10.1007/978-3-030-21548-4_2

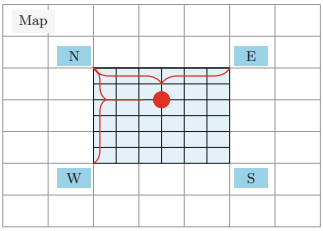

Fig. 1. Broadcasting encrypted message in the triangle area NESW. (Color figure online)

center system generating a ciphertext, which can reduce the size even the area of triangle is rescaled. Likewise, the ciphertext size is constant if the set of locations is increased or decreased.

Motivated from the above scenario, we present a solution which produces a constant ciphertext size. Even the set of locations is increased, the ciphertext size is constant. We also aim that the scheme should be computationally efficient. According Fig. 1, let's assume that the user is located in the red location p_1. If p_1 belongs to $[N, E]$, and $[N, W]$, he/she can decrypt the broadcasted message. There is an obvious solution from [9], which we consider the distance $[N, E]$, $[N, W]$ as the range queries, and the user location as the query. This approach can not deal with the reduce the ciphertext size, since the proposed scheme considers each encrypted data stored in the $(d \times l)$ matrix, and encrypts each data producing in each components in the ciphertext. In another approach, our scenario can be applied directly the Attribute-based encryption [1,5] and Predication Encryption [7]. In fact, a class of policies in Attribute Based Encryption expresses the distance [N, E], [N, W], attributes express user's location, and with the center system playing the role of attribute authority. However, in [1,5,7], the access policy embedded in ciphertext cannot be aggregated into one components, then ciphertext size increase linearly depending on the size of access policy generating by set of attributes. In order to construct a constant size of ciphertext even the resizing of the set of locations, we need to aggregate the ciphertext components corresponding to each location. Moreover, the scheme should guarantee that the successful decryption of one satisfied location belongs to the set of locations embedded in the ciphertext.

Contributions: We first propose a 2D Location Based Encryption (2D-LBE) scheme, which achieves the constant ciphertext size. We apply the Time Specific Encryption (TSE) scheme [6,8] to exploit the time interval properties to construct our idea. In TSE scheme, the decryption is processed when a time t falls into the specific distance time $[t_1, t_2]$. Consequently, we consider each distance $[p_1, p_2]$ as the interval $[t_1, t_2]$ in TSE scheme. We then produce our scheme by evaluating the location p belonging to the $[p_1, p_2]$, and $[p_3, p_4]$; achieving the constant ciphertext size in encryption algorithm and decryption cost. Beyond the 2D-LBE scheme, we explore the 3D-LBE scheme, where there are a center

system and a 3D triangle area including the set of locations. Each location comprises the X, Y, and Z coordinator. Hence, the decryption is processed when a location p is belonging to the specific distance $[p_1, p_2]$, $[p_3, p_4]$, and $[p_5, p_6]$. This proposed scheme is an extension of the 2D-LBE scheme, which the ciphertext size is also constant.

We give a detailed comparison between the aforementioned obvious solution of Multi-Search Range Queries [9] and our proposed schemes in Table 1. The schemes are compared in terms of the order of the underlying group, ciphertext size, decryption cost, and security assumption. In the table, p denotes the pairing operation, (d, l) the dimension of the matrix.

Table 1. Performance comparison

Scheme	Ciphertext size	Decryption cost	Assumption
MQRED [9]	$(6 \times d \times l)\|\mathbb{G}\| + 1\|\mathbb{G}_T\|$	$5d$p	D-BDH
2D-LBE	$5\|\mathbb{G}\| + 1\|\mathbb{G}_T\|$	5p	1 $\ell-$wBDHI
3D-LBE	$7\|\mathbb{G}\| + 1\|\mathbb{G}_T\|$	7p	1 $\ell-$wBDHI

Related Works: In 2009, Chandran et al. [4] proposed a Position Based Cryptography scheme, which utilizes the geographic location to derive the user's identity. In addition, the scheme is position-secured to hide the user's position, however, the verifier still authenticates the location of the user. Buhrman et al. [2] constructed the position based cryptography in the quantum setting, which uses the geographical position of a party as its only credential. Extension in the mobile environment, You et al. [11] proposed a novel location-based encryption model based on a fuzzy vault scheme to protect the sensitive data and the location data. Recently, Yang et at. [10] constructed a secure positioning protocol with location privacy in the bounded retrieval model deploying in a fog computing environment.

2 Preliminaries

2.1 Bilinear Map on Prime Order Groups

Let $\mathbb{G}$ and $\mathbb{G}_\mathbb{T}$ be two multiplicative cyclic groups of same prime order p, and g a generator of $\mathbb{G}$. Let $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ be a bilinear map with the following properties:

1. Bilinearity: $e(u^a, v^b) = e(u^b, v^a) = e(u, v)^{ab}$ for all $u,v \in \mathbb{G}$ and $a,\ b \in \mathbb{Z}_p$.
2. Non-degeneracy: $e(g, g) \neq 1$

Notice that the map e is symmetric since $e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$.

2.2 The Decisional $\ell-$wBDHI Assumption

The Decision ℓ-wBDHI problem in $\mathbb{G}$ is defined as follows: Let $\ell \in \mathbb{N}$, and $\mathbb{G}$ be a bilinear group of prime order p, and g, h two independent generators of $\mathbb{G}$. Denote $\overrightarrow{y}_{g,\alpha,\ell} = (g_1, g_2, \ldots, g_\ell \in \mathbb{G}^\ell$ where $g_i = g^{\alpha^i}$ for some unknown $\alpha \in \mathbb{Z}_p^*$. We say that the ℓ-wBDHI assumption holds in $\mathbb{G}$ if for any probabilistic polynomial-time algorithm A

$$|\Pr[A(g, h, \overrightarrow{y}_{g,\alpha,\ell}, e(g_{\ell+1}, h)) = 1] - \Pr[A(g, h, \overrightarrow{y}_{g,\alpha,\ell}, W) = 1]| \leq \epsilon(k)$$

where the probability is over the random choive of g, h in $\mathbb{G}$, the random choice $\alpha \in \mathbb{Z}^*{}_p$, the random choice $W \in \mathbb{G}_T$, and $\epsilon(k)$ is negligible in the security parameter k.

2.3 Location Based Encryption

A Location Based Encryption (LBE) scheme consists of the following four probabilistic polynomial-time algorithms:

- **Setup**$(1^k, T = 2^\ell - 1)$: on input a security parameter 1^n, and the maximum $T = 2^\ell - 1$, the algorithm outputs a public key PK and a master secret key MSK.
- **Encrypt**$(PK, [t_{1N}, t_{2E}], [t_{2N}, t_{2W}], M)$: on input a public key PK, a message M, two distances $[t_{1N}, t_{2E}], [t_{2N}, t_{2W}]$, the algorithm outputs a ciphertext CT.
- **KeyGen**$(MSK, [t_1, t_2])$: on input a master secret key MSK, a location $[t_1, t_2]$, the algorithm outputs a decryption key SK.
- **Decrypt**(CT, SK): on input a ciphertext CT and a secret key SK, the algorithm outputs either a message M if $[t_1, t_2]$ belongs to $[t_{1N}, t_{2E}], [t_{2N}, t_{2W}]$, or a special symbol $\perp$.

2.4 Security Model

The security model for an LBE scheme is defined via the following game between an adversary A and a challenger B.

- **Setup**: The challenger B run **Setup**$(1^k, T = 2^\ell - 1)$ to generate the PK and MSK. PK is then passed to A.
- **Query Phase 1**: The challenger answers all location extraction queries t_{1_i}, t_{2_i} by generating $SK_{t_{1_i}, t_{2_i}}$ as in the KeyGen algorithm.
- **Challenge**: A submits two equal-length messages M_0 and M_1, challenge X-Y $[t^*_{1N}, t^*_{1E}][t^*_{2N}, t^*_{2W}]$ such that $t_{1_i} \notin [t^*_{1N}, t^*_{1E}], t_{2_i} \in [t^*_{2N}, t^*_{2W}]$ or $t_{1_i} \in [t^*_{1N}, t^*_{1E}], t_{2_i} \notin [t^*_{2N}, t^*_{2W}]$ that has been queried in Phase 1. The challenger then flips a coin $\beta \leftarrow \{0, 1\}$ and computes the challenge ciphertext CT^*, which is given to A.
- **Query Phase 2**: same as Query Phase 1
- **Output**: A outputs a bit β' as her guess for β.

Define the advantage of A as $\mathbf{Adv}_A^{\mathsf{LBE}}(k) = |\Pr[\beta' = \beta] - 1/2|$.

Selective Security. In the selective security model, the adversary A is required to submit the target challenge X-Y $[t^*_{1N}, t^*_{1E}][t^*_{2N}, t^*_{2W}]$ before the game setup, and A is only allowed to make private key queries for any $t_{1_i} \notin [t^*_{1N}, t^*_{1E}], t_{2_i} \in [t^*_{2N}, t^*_{2W}]$ or $t_{1_i} \in [t^*_{1N}, t^*_{1E}], t_{2_i} \notin [t^*_{2N}, t^*_{2W}]$ throughout the game.

3 2D Location Based Encryption

In this section, we propose a 2D Location Based Encryption scheme. Let $\ell \in \mathbb{N}$, we consider the binary tree B with $T = 2^\ell - 1$ nodes, where T will be the number of segments between the location N and E. By using the transformation [3], we map the distance location $[N.E]$ to the binary tree as the following:

We present the vector d_{X_t} and set $\{X_t\}$. d_{X_t} be a vector consisting of the indices corresponding to the root node of B as $2T + 1$.

Suppose that we balance a binary tree of depth $T = 2^\ell - 1$, which T segments in the specific distance from N and E. The root node is installed with $2T + 1$, and each node is labeled with a string in $\{0, 1\}^{\leq \ell}$. In addition, the left and the right children of a node is assigned 0 and 1, Without loss of generality, from the root to E we view a binary string of length $m \leq \ell$ as an N-tuple of length m as $d_{X_{t_{1E}+1}} = \{N_0, \ldots, N_m\}$, and from the root to N we view a binary string of length $n \leq \ell$ as an E-tuple of length n as $d_{X_{2T-t_{1N}}} = \{E_0, E_1, \ldots E_n\}$. Therefore, when encrypting a message with a distance $[N, E]$, the algorithm will associate the $[N, E]$ into a binary tree, and construct the path way from the root tree to the $[N, E]$ by indexing two binary strings $(N_0, \ldots, N_m), (E_0, \ldots, E_n)$. Then to retrieve $t \in [N, E]$, the algorithm invoke as:

For $t \in [1, 2T]$, $X_t = \{d_{X_t}\}$, $X_{T+1} = \{d_{X_{T+1}}\}$. Recursively, for $t \in [1, 2T] \backslash \{1, T + 1\}$, X_{t+1} is computed on X_t as: Let $s = \min\{u : X_u \in X_t\}$. If d_{X_s} is a leaf node, then X_{t+1} is retrieved by removing the vector d_{X_s} from the set X_t. Otherwise, let α_l, α_r be the index of the left, right node of node s respectively. X_{t+1} is the set obtained by removing d_{X_s} and adding $d_{X_{\alpha_l}}, d_{X_{\alpha_r}}$ to the set X_t (1). Hence, the setting of segments between the location N and W is similarly to the location N and E.

In addition, we should face another problem of a key generation when using master key $g^{\alpha\beta}$ to generate the user key securely. Hence, our idea is to share $\alpha\beta$, we re-generate r, z by randomly choosing and obtaining the re-share $(\alpha\beta + r - r), (z, -z)$, which r, z are also blinding factors. As a result, the extract keys are computed by the master key completely obviousness.

We elaborate the Setup, Encryption, Key Extraction, Decryption algorithms defined above. Let $\mathbb{G}, \mathbb{G}_\mathbb{T}, e$ be bilinear maps, and $T = 2^\ell - 1$ ($\ell \in \mathbb{N}$) be a polynomial that indicates the segments of X and Y coordinate. Our 2D Location based scheme is presented in the following:

▶ $\mathsf{Setup}(1^k, T = 2^\ell - 1)$: The algorithm first chooses randomly $\alpha, \beta \in \mathbb{Z}_p$, and chooses uniformly $g_{1N}, g_{1E}, g_{2N}, g_{2W}, h_0, \ldots, h_\ell \in \mathbb{G}$. Then it computes:

$$MSK = g^{\alpha\beta},$$
$$MPK = (g, g_1, g_{1N}, g_{1E}, g_{2N}, g_{2W}, h_0, \ldots, h_\ell, Y = e(g^\alpha, g^\beta)),$$

and returns MPK, MSK.

▶ Encryption$(MPK, [t_{1N}, t_{2E}], [t_{2N}, t_{2W}], M)$: Firstly, let

$$d_{X_{t_{1E}+1}} = (N_0, N_1, \ldots N_m), d_{X_{2T-t_{1N}}} = (E_0, E_1, \ldots E_n),$$
$$d_{Y_{t_{2W}+1}} = (\hat{N}_0, \hat{N}_1, \ldots \hat{N}_m), d_{Y_{2T-t_{2N}}} = (W_0, W_1, \ldots W_n),$$

with fixed numbers m, n. The algorithm then chooses randomly $s \in \mathbb{Z}_p$, and computes:

$$C_0 = Y^s \cdot M,$$
$$C_1 = g^s,$$
$$C_2 = (\prod_{i=0}^{m} h_i^{N_i} \cdot g_{1N})^s, C_3 = (\prod_{i=0}^{n} h_i^{E_i} \cdot g_{1E})^s$$
$$C_4 = (\prod_{i=0}^{m} h_i^{\hat{N}_i} \cdot g_{2N})^s, C_5 = (\prod_{i=0}^{m} h_i^{W_i} \cdot g_{2W})^s,$$

and returns $CT = (C_0, C_1, C_2, C_3, C_4, C_5, [t_{1N}, t_{1E}], [t_{2N}, t_{2W}])$.

▶ Extract$(MSK, [t_1, t_2])$: The algorithm chooses randomly $r, z \in \mathbb{Z}_p$.

- For each $X_1 = (N_0, N_1, \ldots, N_m) \in X_{t_1+1}$, the algorithms picks randomly $r_{1N} \in \mathbb{Z}_p$, and computes:

$$d_{X_1} = (g^{\alpha\beta+r} \cdot (\prod_{i=0}^{m} h_i^{N_i} \cdot g_{1N})^{r_{1N}}, g^{r_{1N}}, h_{m+1}^{r_{1N}}, \ldots, h_\ell^{r_{1N}}).$$

- For each $X_2 = (E_0, E_1, \ldots, E_n) \in X_{2T-t_1}$, it picks randomly $r_{1E} \in \mathbb{Z}_p$. and computes:

$$d_{X_2} = (g^{-r} \cdot (\prod_{i=0}^{n} h_i^{E_i} \cdot g_{1E})^{r_{1E}}, g^{r_{1E}}, h_{n+1}^{r_{1E}}, \ldots, h_\ell^{r_{1E}}).$$

- For each $Y_1 = (\hat{N}_0, \hat{N}_1, \ldots, \hat{N}_m) \in Y_{t_2+1}$, it picks randomly $r_{2N} \in \mathbb{Z}_p$. and computes:

$$d_{Y_1} = (g^z \cdot (\prod_{i=0}^{m} h_i^{\hat{N}_i} \cdot g_{2N})^{r_{2N}}, g^{r_{2N}}, h_{m+1}^{r_{2N}}, \ldots, h_\ell^{r_{2N}}).$$

- For each $Y_2 = (W_0, W_1, \ldots, W_n) \in Y_{2T-t_2}$, it picks randomly $r_{2W} \in \mathbb{Z}_p$. and computes:

$$d_{Y_2} = (g^{-z} \cdot (\prod_{i=0}^{n} h_i^{W_i} \cdot g_{2W})^{r_{2W}}, g^{r_{2W}}, h_{n+1}^{r_{2W}}, \ldots, h_\ell^{r_{2W}}).$$

Finally, it sets:

$$SK_{t_{1,1N}} = \{d_{X_1}\}_{X_1 \in X_{t_1+1}}, SK_{t_{1,1E}} = \{d_{X_2}\}_{X_2 \in X_{2T-t_1}}$$
$$SK_{t_{2,2N}} = \{d_{Y_1}\}_{Y_1 \in Y_{t_2+1}}, SK_{t_{2,2W}} = \{d_{Y_2}\}_{Y_2 \in Y_{2T-t_2}},$$

and returns $SK_{t_1,t_2} = \{SK_{t_{1,1N}}, SK_{t_{1,1E}}, SK_{t_{2,2N}}, SK_{t_{2,2W}}, \{t_1, t_2\}\}$.

▶ Decryption(SK_{t_1,t_2}, CT): If $\{t_1, t_2\} \notin ([t_{1N}, t_{1E}], [t_{2N}, t_{2W}])$ return $\perp$. Otherwise, the algorithm retrieves:

$$d_{X_{t_{1E}+1}} = (N_1, N_2, \ldots), d_{X_{2T-t_{1N}}} = (E_1, E_2, \ldots),$$
$$d_{Y_{t_{2W}+1}} = (\hat{N}_1, \hat{N}_2, \ldots), d_{Y_{2T-t_{2N}}} = (W_1, W_2, \ldots),$$

Then, it computes:

$$\frac{C_0 \cdot e(d_{X_{12}}, C_2) \cdot e(d_{X_{22}}, C_3) \cdot e(d_{Y_{12}}, C_4) \cdot e(d_{Y_{22}}, C_5)}{e(d_{X_{11}} \cdot d_{X_{21}} \cdot d_{Y_{11}} \cdot d_{Y_{21}}, C_1)}$$

Let:

$$\begin{aligned}
A &= e(d_{X_{12}}, C_2) \cdot e(d_{X_{22}}, C_3) \cdot e(d_{Y_{12}}, C_4) \cdot e(d_{Y_{22}}, C_5) \\
&= e(g^{r_{1N}}, (\prod_{i=0}^{m} h_i^{N_i} \cdot g_{1N})^s) \cdot e(g^{r_{1E}}, (\prod_{i=0}^{n} h_i^{E_i} \cdot g_{1E})^s) \\
&\quad \cdot e(g^{r_{2N}}, (\prod_{i=0}^{m} h_i^{\hat{N}_i} \cdot g_{2N})^s) \cdot e(g^{r_{2W}}, (\prod_{i=0}^{m} h_i^{W_i} \cdot g_{2W})^s) \\
B &= e(d_{X_{11}} \cdot d_{X_{21}} \cdot d_{Y_{11}} \cdot d_{Y_{21}}, C_1) \\
&= e(g^{\alpha\beta+r} \cdot (\prod_{i=0}^{m} h_i^{N_i} \cdot g_{1N})^{r_{1N}} \cdot g^{-r} \cdot (\prod_{i=0}^{n} h_i^{E_i} \cdot g_{1E})^{r_{1E}} \\
&\quad \cdot g^z \cdot (\prod_{i=0}^{m} h_i^{\hat{N}_i} \cdot g_{2N})^{r_{2N}} \cdot g^{-z} \cdot (\prod_{i=0}^{n} h_i^{W_i} \cdot g_{2W})^{r_{2W}}, g^s)
\end{aligned}$$

To recover message M:

$$\frac{Y^s \cdot M \cdot A}{B} = M$$

4 Security Proof

Theorem 1. *Assume that the ℓ-wBDHI assumption holds, then no polynomial-time adversary against our 2D Location based Encryption scheme can have a non-negligible advantage over random guess in the Selective IND-CPA security game.*

We assume our our 2D Location based Encryption with the size of X-Y coordinate which is polynomial in the security parameter k. We consider the selective adversary the decides the challenge X-Y $[t^*_{1N}, t^*_{1E}][t^*_{2N}, t^*_{2W}]$ at the beginning of the IND-CPA game.

Let $\mathcal{A}$ be any IND-CPA adversary that attacks our proposed scheme. We then build an algorithm $\mathcal{B}$ that solves the decisional ℓ-wBDHI problem in $(\mathbb{G}, \mathbb{G}_T, e)$ by using $\mathcal{A}$.

Let g, h choose uniformly in $\mathbb{G}$, randomly $\alpha \in \mathbb{Z}_p$, and sets $y_i = g^{\alpha^{i+1}}$. $\mathcal{B}$ is given as input $(g, h, y_0, y_1, \ldots, y_\ell, Y)$, where Y is $e(g, h)^{\alpha^{l+2}}$ or a random value in $\mathbb{G}_T$. $\mathcal{B}$ interacts with $\mathcal{A}$ as follows:

– **Setup**: $\mathcal{A}$ outputs the challenge X $[t^*_{1N}, t^*_{1E}]$, and Y $[t^*_{2N}, t^*_{2W}]$. Then lets:

$$d_{X_{t^*_{1E}+1}} = (N^*_0, N^*_1, \dots N^*_m), d_{X_{2T-t^*_{1N}}} = (E^*_0, E^*_1, \dots E^*_n),$$
$$d_{Y_{t^*_{2W}+1}} = (\hat{N}^*_0, \hat{N}^*_1, \dots \hat{N}^*_m), d_{Y_{2T-t^*_{2N}}} = (W^*_0, W^*_1, \dots W^*_n),$$

$\mathcal{B}$ picks randomly $\gamma, \gamma_0, \gamma_1, \dots, \gamma_\ell, \alpha_N, \alpha_E, \alpha_{\hat{N}}, \alpha_W \in \mathbb{Z}_p$, and $g_1 = y_0$, then computes:

$$g_{1N} = g^{\alpha_N} \cdot \prod_{i=0}^{m} y_{\ell-1}^{N^*_i} g_{1E} = g^{\alpha_E} \cdot \prod_{i=0}^{n} y_{\ell-1}^{E^*_i}$$
$$g_{2N} = g^{\alpha_{\hat{N}}} \cdot \prod_{i=0}^{m} y_{\ell-1}^{\hat{N}^*_i} g_{2W} = g^{\alpha_W} \cdot \prod_{i=0}^{n} y_{\ell-1}^{W^*_i}$$
$$h_i = g^{\gamma_i} \cdot y_{\ell-i} = g^{\gamma_i - \alpha^{\ell-i+1}} Y = e(y_0, y_\ell g^\gamma),$$

where $\alpha^{\ell+1} + \gamma$ is implicitly setting as β. $\mathcal{B}$ then sets $MPK = (g, g_1, g_{1N}, g_{1E}, g_{2N}, g_{2W}, h_0, \dots, h_\ell, Y = e(g^\alpha, g^\beta))$, and gives it to $\mathcal{A}$.

– **Phase 1**: If $\mathcal{A}$ submits a location extraction query t_{1_i}, t_{2_i}, $\mathcal{B}$ responds to each query by generating $SK_{t_{1_i}, t_{2_i}}$ as follows:

- Case 1: $t_{1_i} < t^*_{1N}, t_{2_i} \in [t^*_{2N}, t^*_{2W}]$
  $\mathcal{B}$ implicitly sets $r' = \alpha^{\ell+2} + r$, where r, z is chosen randomly from $\mathbb{Z}_p$, and r' is distributed uniformly in $\mathbb{Z}_p$. Then:
  * For each $X_1 = (N_0, N_1, \dots, N_m) \in X_{t_{1_i}+1}$, $\mathcal{B}$ picks r_{1N} randomly in $\mathbb{Z}_p$, and computes:

$$d_{X_1} = \left(y_0^\gamma g^{r'} (\prod_{i=0}^{m} g^{\gamma_i - \alpha^{\ell-i+1}} g^{\alpha_N} \cdot \prod_{i=0}^{m} y_{\ell-1}^{N^*_i})^{r_{1N}}, g^{r_{1N}}, h_{m+1}^{r_{1N}}, \dots, h_\ell^{r_{1N}} \right)$$
$$= \left(g^{\alpha^{\ell+2} + \gamma\alpha} \cdot g^{r' - \alpha^{\ell+2}} \cdot (\prod_{i=0}^{m} h_i^{N_i} \cdot g_{1N})^{r_{1N}}, g^{r_{1N}}, h_{m+1}^{r_{1N}}, \dots, h_\ell^{r_{1N}} \right)$$
$$= \left(g^{\alpha\beta + r} \cdot (\prod_{i=0}^{m} h_i^{N_i} \cdot g_{1N})^{r_{1N}}, g^{r_{1N}}, h_{m+1}^{r_{1N}}, \dots, h_\ell^{r_{1N}} \right).$$

  * We consider the secret keys of $X_2 \in X_{2T-t_{1_i}}$. For $X_2 = (E^*_0, \dots, E^*_{d-1}, E_d, \dots, E_n)$. $\mathcal{B}$ then generates the secret key of $(E^*_0, \dots, E^*_{d-1}, E_d)$, and use this secret key to derive the secret key of X_2. $\mathcal{B}$ picks randomly r'_{1E}, which implicitly sets $r'_{1E} = \alpha^{d+1} + r_{1E}(E^*_d - E_d)$. $\mathcal{B}$ computes:

$$d_{X_2} = \left(g^{r'} \cdot g^{\alpha^{\ell-d+1} r'_{1E}} \cdot \left\{ \left(\prod_{i=0}^{d-1} y_d^{\gamma_i E^*_i} \cdot y_d^{\gamma_d E_d} \cdot y_d^{\alpha_E} \cdot \prod_{j=d+1}^{n} y_{\ell-j+d+1}^{E^*_i} \right) \right.\right.$$
$$\left.\left. \cdot \left(\prod_{i=0}^{d-1} g^{\gamma_i E^*_i} \cdot g^{\gamma_d E_d} \cdot g^{\alpha_E} \cdot \prod_{j=d+1}^{n} (g^{\alpha^{\ell-j+d+1}})^{E^*_i} \right)^{-r'_{1E}} \right\}^{\frac{1}{E^*_d - E_d}},$$

$$
\begin{aligned}
&(g^{r'_{1E}} \cdot y_d^{-1})^{\frac{1}{E_d^* - E_d}}, (g^{\gamma_{d+1} r'_{1E}} \cdot y_d^{-\gamma_d+1} \cdot y_{\ell-d-1}^{r'_{1E}} \cdot y_\ell)^{\frac{1}{E_d^* - E_d}}, \\
&h_{d+2}^{r_{1E}}, \ldots, h_\ell^{r_{1E}} \Bigg) \\
= &\Bigg(g^{-r'} \cdot g^{\alpha^{\ell+2}} \cdot g^{\alpha^{\ell-d+1}(E_d^* - E_d) r_{1E}} \cdot \Bigg(\prod_{i=0}^{d-1} g^{\gamma_i E_i^*} \cdot g^{\gamma_d E_d} \cdot g^{\alpha_E} \cdot \\
&\prod_{j=d+1}^{n} (g^{\alpha^{\ell-j+d+1}})^{E_i^*} \Bigg)^{r_{1E}}, \\
&g^{\frac{r'_{1E} - \alpha^{d+1}}{E_d^* - E_d}}, (g^{\gamma_{d+1} - \alpha^{\ell-d}})^{\frac{r'_{1E} - \alpha^{d+1}}{E_d^* - E_d}}, h_{d+2}^{r_{1E}}, \ldots, h_\ell^{r_{1E}} \Bigg) \\
= &\Bigg(g^{-r' + \alpha^{\ell+2}} \cdot \Bigg(\prod_{i=0}^{d-1} g^{(\gamma_i - \alpha^{\ell-i+1}) E_i^*} \cdot g^{(\gamma_d - \alpha^{\ell-d+1}) E_d} \cdot g^{\alpha_E} \\
&\cdot \prod_{j=0}^{n} (g^{\alpha^{\ell-j+1}})^{E_i^*} \Bigg)^{r_{1E}}, \\
&g^{r_{1E}}, h_{d+1}^{r_{1E}}, h_{d+2}^{r_{1E}}, \ldots, h_\ell^{r_{1E}} \Bigg) \\
= &\Bigg(g^{-r} \cdot (\prod_{i=0}^{n} h_i^{E_i} \cdot g_{1E})^{r_{1E}}, g^{r_{1E}}, h_{n+1}^{r_{1E}}, \ldots, h_\ell^{r_{1E}} \Bigg)
\end{aligned}
$$

* For each $Y_1 = (\hat{N}_0, \hat{N}_1, \ldots, \hat{N}_m) \in Y_{t_2+1}$, $\mathcal{B}$ picks randomly $r_{2N} \in \mathbb{Z}_p$. and computes:

$$d_{Y_1} = (g^z \cdot (\prod_{i=0}^{m} h_i^{\hat{N}_i} \cdot g_{2N})^{r_{2N}}, g^{r_{2N}}, h_{m+1}^{r_{2N}}, \ldots, h_\ell^{r_{2N}}).$$

* For each $Y_2 = (W_0, W_1, \ldots, W_n) \in Y_{2T-t_2}$, $\mathcal{B}$ picks randomly $r_{2W} \in \mathbb{Z}_p$. and computes:

$$d_{Y_2} = (g^{-z} \cdot (\prod_{i=0}^{n} h_i^{W_i} \cdot g_{2W})^{r_{2W}}, g^{r_{2W}}, h_{n+1}^{r_{2W}}, \ldots, h_\ell^{r_{2W}}).$$

When simulating d_{X_2}, the components $h_{d+2}^{r_{1E}}, \ldots, h_\ell^{r_{1E}}$ are not involved a $\alpha^{\ell+2}$, then $\mathcal{B}$ can simulate these components similarly as in the main construction. Finally, $\mathcal{B}$ sets $SK_{t_{1_i}, t_{2_i}} = \{SK_{t_{1_i},1N}, SK_{t_{1_i},1E}, SK_{t_{2_i},2N}, SK_{t_{2_i},2W}, \{t_{1_i}, t_{2_i}\}$, and sends the $SK_{t_{1_i}, t_{2_i}}$ to $\mathcal{A}$.

– Case 2: $t_{2_i} > t_{1E}^*, t_{2_i} \in [t_{2N}^*, t_{2W}^*]$
$\mathcal{B}$ chooses r, z is chosen randomly from $\mathbb{Z}_p$. Then:

- For each $X_1 = (N_0, N_1, \ldots, N_m) \in X_{t_{1_i}+1}$, we consider the secret keys of $X_1 \in X_{t_{1_i}+1}$. For $X_1 = (N_0^*, \ldots, N_{d-1}^*, N_d, \ldots, N_m)$. $\mathcal{B}$ then generates the secret key of $(E_0^*, \ldots, E_{d-1}^*, E_d)$, and use this secret key to derive the secret key of X_1. $\mathcal{B}$ picks randomly r'_{1N}, which implicitly sets $r'_{1N} = \alpha^{d+1} + r_{1N}(E_d^* - E_d)$. $\mathcal{B}$ computes:

$$d_{X_1} = \Bigg(y_0^{\gamma} g^r \cdot g^{\alpha^{\ell-d+1} r'_{1N}} \cdot \Bigg\{ \Bigg(\prod_{i=0}^{d-1} y_d^{\gamma_i N_i^*} \cdot y_d^{\gamma_d N_d} \cdot y_d^{\alpha_N} \cdot \prod_{j=d+1}^{m} y_{\ell-j+d+1}^{N_i^*} \Bigg) \cdot \Bigg(\prod_{i=0}^{d-1} g^{\gamma_i N_i^*} \cdot g^{\gamma_d N_d} \cdot g^{\alpha_N} \cdot \prod_{j=d+1}^{m} (g^{\alpha^{\ell-j+d+1}})^{N_i^*} \Bigg)^{-r'_{1N}} \Bigg\}^{\frac{1}{N_d^* - N_d}},$$
$$(g^{r'_{1N}} \cdot y_d^{-1})^{\frac{1}{N_d^* - N_d}}, (g^{\gamma_{d+1} r'_{1N}} \cdot y_d^{-\gamma_d + 1} \cdot y_{\ell-d-1}^{r'_{1N}} \cdot y_\ell)^{\frac{1}{N_d^* - N_d}}, h_{d+2}^{r_{1N}}, \ldots, h_\ell^{r_{1N}} \Bigg)$$

- For $X_2 \in X_{2T-t_{1_i}}$. For $X_2 = (E_0^*, \ldots, E_{d-1}^*, E_d, \ldots, E_n)$. $\mathcal{B}$ picks randomly r_{1E}, then computes:

$$d_{X_2} = \left(g^{-r} \cdot (\prod_{i=0}^{n} h_i^{E_i} \cdot g_{1E})^{r_{1E}}, g^{r_{1E}}, h_{n+1}^{r_{1E}}, \ldots, h_\ell^{r_{1E}} \right)$$

- For each $Y_1 = (\hat{N}_0, \hat{N}_1, \ldots, \hat{N}_m) \in Y_{t_2+1}$, $\mathcal{B}$ picks randomly $r_{2N} \in \mathbb{Z}_p$. and computes:

$$d_{Y_1} = (g^z \cdot (\prod_{i=0}^{m} h_i^{\hat{N}_i} \cdot g_{2N})^{r_{2N}}, g^{r_{2N}}, h_{m+1}^{r_{2N}}, \ldots, h_\ell^{r_{2N}}).$$

- For each $Y_2 = (W_0, W_1, \ldots, W_n) \in Y_{2T-t_2}$, $\mathcal{B}$ picks randomly $r_{2W} \in \mathbb{Z}_p$. and computes:

$$d_{Y_2} = (g^{-z} \cdot (\prod_{i=0}^{n} h_i^{W_i} \cdot g_{2W})^{r_{2W}}, g^{r_{2W}}, h_{n+1}^{r_{2W}}, \ldots, h_\ell^{r_{2W}}).$$

– Case 3: $t_{1_i} \in [t_{1N}^*, t_{1E}^*], t_{2_i} < t_{2N}^*$
$\mathcal{B}$ chooses r, z is chosen randomly from $\mathbb{Z}_p$. Then:

- For each $X_1 = (N_0, N_1, \ldots, N_m) \in X_{t_{1_i}+1}$, $\mathcal{B}$ picks r_{1N} randomly in $\mathbb{Z}_p$, and computes:

$$d_{X_1} = \left(y_0^{\gamma} g^r (\prod_{i=0}^{m} g^{\gamma_i - \alpha^{\ell-i+1}} g^{\alpha_N} \cdot \prod_{i=0}^{m} y_{\ell-1}^{N_i^*})^{r_{1N}}, g^{r_{1N}}, h_{m+1}^{r_{1N}}, \ldots, h_\ell^{r_{1N}} \right)$$

- For $X_2 \in X_{2T-t_{1_i}}$. For $X_2 = (E_0^*, \ldots, E_{d-1}^*, E_d, \ldots, E_n)$. $\mathcal{B}$ picks randomly r_{1E}, then computes:

$$d_{X_2} = \left(g^{-r} \cdot (\prod_{i=0}^{n} h_i^{E_i} \cdot g_{1E})^{r_{1E}}, g^{r_{1E}}, h_{n+1}^{r_{1E}}, \ldots, h_\ell^{r_{1E}} \right)$$

- For each $Y_1 = (\hat{N}_0, \hat{N}_1, \ldots, \hat{N}_m) \in Y_{t_2+1}$, $\mathcal{B}$ picks randomly $r_{2N} \in \mathbb{Z}_p$. and computes:

$$d_{Y_1} = (g^z \cdot (\prod_{i=0}^{m} h_i^{\hat{N}_i} \cdot g_{2N})^{r_{2N}}, g^{r_{2N}}, h_{m+1}^{r_{2N}}, \ldots, h_\ell^{r_{2N}}).$$

- We consider the secret keys of $Y_2 \in Y_{2T-t_{2_i}}$. For $Y_2 = (W_0^*, \ldots, W_{d-1}^*, W_d, \ldots, W_n)$. $\mathcal{B}$ then generates the secret key of $(W_0^*, \ldots, W_{d-1}^*, W_d)$, and use this secret key to derive the secret key of Y_2. $\mathcal{B}$ picks randomly r'_{2W}, which implicitly sets $r'_{2W} = \alpha^{d+1} + r_{2W}(W_d^* - W_d)$. $\mathcal{B}$ computes:

$$\begin{aligned} d_{Y_2} = & \left(g^{-z} \cdot g^{\alpha^{\ell-d+1} r'_{2W}} \cdot \left\{ \left(\prod_{i=0}^{d-1} y_d^{\gamma_i W_i^*} \cdot y_d^{\gamma_d W_d} \cdot y_d^{\alpha_W} \cdot \prod_{j=d+1}^{n} y_{\ell-j+d+1}^{W_i^*} \right) \right. \right. \\ & \left. \left. \cdot \left(\prod_{i=0}^{d-1} g^{\gamma_i W_i^*} \cdot g^{\gamma_d W_d} \cdot g^{\alpha_W} \cdot \prod_{j=d+1}^{n} (g^{\alpha^{\ell-j+d+1}})^{W_i^*} \right)^{-r'_{2W}} \right\}^{\frac{1}{W_d^* - W_d}}, \right. \\ & (g^{r'_{2W}} \cdot y_d^{-1})^{\frac{1}{W_d^* - W_d}}, (g^{\gamma_{d+1} r'_{2W}} \cdot y_d^{-\gamma_d + 1} \cdot y_{\ell-d-1}^{r'_{2W}} \cdot y_\ell)^{\frac{1}{W_d^* - W_d}}, \\ & \left. h_{d+2}^{r_{2W}}, \ldots, h_\ell^{r_{2W}} \right) \\ = & \left(g^{-z} \cdot (\prod_{i=0}^{n} h_i^{W_i} \cdot g_{2W})^{r_{2W}}, g^{r_{2W}}, h_{n+1}^{r_{2W}}, \ldots, h_\ell^{r_{2W}} \right) \end{aligned}$$

– Case 4: $t_{1_i} \in [t_{1N}^*, t_{1E}^*], t_{2_i} > t_{2W}^*$
$\mathcal{B}$ chooses r, z is chosen randomly from $\mathbb{Z}_p$. Then:

- For each $X_1 = (N_0, N_1, \ldots, N_m) \in X_{t_{1_i}+1}$, $\mathcal{B}$ picks r_{1N} randomly in $\mathbb{Z}_p$, and computes:

$$d_{X_1} = \left(y_0^\gamma g^r (\prod_{i=0}^{m} g^{\gamma_i - \alpha^{\ell-i+1}} g^{\alpha_N} \cdot \prod_{i=0}^{m} y_{\ell-1}^{N_i^*})^{r_{1N}}, g^{r_{1N}}, h_{m+1}^{r_{1N}}, \ldots, h_\ell^{r_{1N}} \right)$$

- For $X_2 \in X_{2T-t_{1_i}}$. For $X_2 = (E_0^*, \ldots, E_{d-1}^*, E_d, \ldots, E_n)$. $\mathcal{B}$ picks randomly r_{1E}, then computes:

$$d_{X_2} = \left(g^{-r} \cdot (\prod_{i=0}^{n} h_i^{E_i} \cdot g_{1E})^{r_{1E}}, g^{r_{1E}}, h_{n+1}^{r_{1E}}, \ldots, h_\ell^{r_{1E}} \right)$$

- We consider the secret keys of $Y_1 \in Y_{t_{2_i}+1}$. For $Y_1 = (\hat{N}_0^*, \ldots, \hat{N}_{d-1}^*, \hat{N}_d, \ldots, \hat{N}_m)$. $\mathcal{B}$ then generates the secret key of $(\hat{N}_0^*, \ldots, \hat{N}_{d-1}^*, \hat{N}_d)$, and use this secret key to derive the secret key of Y_1. $\mathcal{B}$ picks randomly $r'_{2\hat{N}}$, which implicitly sets $r'_{2\hat{N}} = \alpha^{d+1} + r_{1\hat{N}}(\hat{N}_d^* - \hat{N}_d)$. $\mathcal{B}$ computes:

$$d_{Y_1} = \Bigg(g^z \cdot g^{\alpha^{\ell-d+1} r'_{2\hat{N}}} \cdot \Bigg\{ \Bigg(\prod_{i=0}^{d-1} y_d^{\gamma_i \hat{N}_i^*} \cdot y_d^{\gamma_d \hat{N}_d} \cdot y_d^{\alpha_{\hat{N}}} \cdot \prod_{j=d+1}^{m} y_{\ell-j+d+1}^{\hat{N}_i^*} \Bigg)$$
$$\cdot \Bigg(\prod_{i=0}^{d-1} g^{\gamma_i \hat{N}_i^*} \cdot g^{\gamma_d \hat{N}_d} \cdot g^{\alpha_{\hat{N}}} \cdot \prod_{j=d+1}^{m} (g^{\alpha^{\ell-j+d+1}})^{\hat{N}_i^*} \Bigg)^{-r'_{2\hat{N}}} \Bigg\}^{\frac{1}{\hat{N}_d^* - \hat{N}_d}},$$
$$(g^{r'_{2\hat{N}}} \cdot y_d^{-1})^{\frac{1}{\hat{N}_d^* - \hat{N}_d}}, (g^{\gamma_{d+1} r'_{2\hat{N}}} \cdot y_d^{-\gamma_d + 1} \cdot y_{\ell-d-1}^{r'_{2\hat{N}}} \cdot y_\ell)^{\frac{1}{\hat{N}_d^* - \hat{N}_d}},$$
$$h_{d+2}^{r_{2\hat{N}}}, \ldots, h_\ell^{r_{2\hat{N}}} \Bigg)$$
$$= \Bigg(g^s \cdot (\prod_{i=0}^{m} h_i^{\hat{N}_i} \cdot g_{2\hat{N}})^{r_{2\hat{N}}}, g^{r_{2\hat{N}}}, h_{m+1}^{r_{2\hat{N}}}, \ldots, h_\ell^{r_{2\hat{N}}} \Bigg)$$

- For each $Y_2 = (W_0, W_1, \ldots, W_n) \in Y_{2T-t_{2_i}}$, $\mathcal{B}$ picks randomly $r_{2W} \in \mathbb{Z}_p$. and computes:

$$d_{Y_2} = (g^{-z} \cdot (\prod_{i=0}^{n} h_i^{W_i} \cdot g_{2W})^{r_{2W}}, g^{r_{2W}}, h_{n+1}^{r_{2W}}, \ldots, h_\ell^{r_{2W}}).$$

– **Challenge**: When $\mathcal{A}$ decides that Phase 1 is over, it outputs the challenge plaintexts M_0, M_1. $\mathcal{B}$ picks a random bit $b \in_U \{0,1\}$, and computes the challenge ciphertext by

$$CT^* = (M_b \cdot T \cdot e(y_0, h^\gamma), h, h^{\alpha_N + \sum_{i=0}^{m} N_i^* \gamma_i}, h^{\alpha_E + \sum_{i=0}^{n} E_i^* \gamma_i}, h^{\alpha_{\hat{N}} + \sum_{i=0}^{m} \hat{N}_i^* \gamma_i},$$
$$h^{\alpha_W + \sum_{i=0}^{n} NW_{*i} \gamma_i}, [t_{1N}^*, t_{1E}^*][t_{2N}^*, t_{2W}^*])$$

If $T = e(g,h)^{\alpha^\ell + 2}$, then CT^* is of the following form by letting $\log_g h = s$

$$CT^* = (M_b \cdot e(g,h)^{\alpha^{\ell+2} s} \cdot e(g,g)^{\alpha\gamma s}, g^s, (g^{\alpha_N} \cdot \prod_{i=0}^{m} y_{\ell-1}^{N_i^*})^s, (g^{\alpha_E} \cdot \prod_{i=0}^{n} y_{\ell-1}^{E_i^*})^s,$$
$$(g^{\alpha_{\hat{N}}} \cdot \prod_{i=0}^{m} y_{\ell-1}^{\hat{N}_i^*})^s, (g^{\alpha_W} \cdot \prod_{i=0}^{n} y_{\ell-1}^{W_i^*})^s)$$

$\mathcal{B}$ sends the following challenge ciphertext to $\mathcal{A}$:

$$CT^* = (M_b T, C_1, C_2, C_3, C_4, C_5, [t_{1N}^*, t_{1E}^*][t_{2N}^*, t_{2W}^*]).$$

– **Phase II**: Same as Phase I.
– **Guess**: $\mathcal{A}$ outputs $b' \in \{0,1\}$. If $b' = b$ then $\mathcal{B}$ outputs 1, otherwise outputs 0.

Analysis: If $T = e(g,h)^{\alpha^{\ell+2}}$, then the simulation is the same as in the real game. Hence, $\mathcal{A}$ will have the probability $\frac{1}{2} + \epsilon$ to guess b correctly. If T is a random element of $\mathbb{G}_T$, then $\mathcal{A}$ will have probability $\frac{1}{2}$ to guess b correctly. Therefore, $\mathcal{B}$ can solve the Decisionℓ-wBDHI assumption also with advantage ϵ. □

5 3D Location Based Encryption

We extend the 2D-LBE scheme to 3D-LBE scheme, where each location comprises the X, Y, and Z coordinator. Hence, the decryption is processed when a location t is belonging to the specific distance $[t_1, t_2]$, $[t_3, t_4]$, and $[t_5, t_6]$. In order to share $\alpha\beta$, we re-generate r, z, k by randomly choosing and obtaining the re-share $(\alpha\beta + r - r), (z, -z), (k, -k)$, which r, z, k are also blinding factors. We elaborate the Setup, Encryption, Key Extraction, Decryption algorithms defined above.

Let $\mathbb{G}, \mathbb{G}_{\mathbb{T}}, e$ be bilinear maps, and $T = 2^{\ell} - 1$ ($\ell \in \mathbb{N}$) be a polynomial that indicates the segments of X, Y and Z coordinate. Our 3D Location Based scheme is presented in the following:

▶ Setup$(1^k, T = 2^{\ell} - 1)$: The algorithm first chooses randomly $\alpha, \beta \in \mathbb{Z}_p$, and chooses uniformly $g_{1N}, g_{1E}, g_{2N}, g_{2W}, h_0, \ldots, h_{\ell} \in \mathbb{G}$. Then it computes:

$$MSK = g^{\alpha\beta},$$
$$MPK = (g, g_1, g_{1N}, g_{1E}, g_{2N}, g_{2W}, g_{3N}, g_{3E}, h_0, \ldots, h_{\ell}, Y = e(g^{\alpha}, g^{\beta})),$$

and returns MPK, MSK.

▶ Encryption$(MPK, [t_{1N}, t_{1E}], [t_{2N}, t_{2W}], [t_{3N}, t_{3E}], M)$: Firstly, let

$$d_{X_{t_{1E}+1}} = (N_0, N_1, \ldots N_m), d_{X_{2T-t_{1N}}} = (E_0, E_1, \ldots E_n),$$
$$d_{Y_{t_{2W}+1}} = (\hat{N}_0, \hat{N}_1, \ldots \hat{N}_m), d_{Y_{2T-t_{2N}}} = (W_0, W_1, \ldots W_n),$$
$$d_{Z_{t_{3E}+1}} = (N'_0, N'_1, \ldots N'_m), d_{Z_{2T-t_{3N}}} = (E'_0, E'_1, \ldots E'_n),$$

with fixed numbers m, n. The algorithm then chooses randomly $s \in \mathbb{Z}_p$, and computes:

$$C_0 = Y^s \cdot M,$$
$$C_1 = g^s,$$
$$C_2 = (\prod_{i=0}^{m} h_i^{N_i} \cdot g_{1N})^s, C_3 = (\prod_{i=0}^{n} h_i^{E_i} \cdot g_{1E})^s$$
$$C_4 = (\prod_{i=0}^{m} h_i^{\hat{N}_i} \cdot g_{2N})^s, C_5 = (\prod_{i=0}^{m} h_i^{W_i} \cdot g_{2W})^s,$$
$$C_6 = (\prod_{i=0}^{m} h_i^{N'_i} \cdot g_{3N})^s, C_7 = (\prod_{i=0}^{m} h_i^{E'_i} \cdot g_{3E})^s,$$

and returns $CT = (C_0, C_1, C_2, C_3, C_4, C_5, C_6, C_7, [t_{1N}, t_{1E}], [t_{2N}, t_{2W}], [t_{3N}, t_{3E}])$.

▶ Extract$(MSK, [t_1, t_2, t_3])$: The algorithm chooses randomly $r, z, k \in \mathbb{Z}_p$.

- For each $X_1 = (N_0, N_1, \ldots, N_m) \in X_{t_1+1}$, the algorithms picks randomly $r_{1N} \in \mathbb{Z}_p$, and computes:

$$d_{X_1} = (g^{\alpha\beta+r} \cdot (\prod_{i=0}^{m} h_i^{N_i} \cdot g_{1N})^{r_{1N}}, g^{r_{1N}}, h_{m+1}^{r_{1N}}, \ldots, h_{\ell}^{r_{1N}}).$$

- For each $X_2 = (E_0, E_1, \ldots, E_n) \in X_{2T-t_1}$, it picks randomly $r_{1E} \in \mathbb{Z}_p$. and computes:

$$d_{X_2} = (g^{-r} \cdot (\prod_{i=0}^{n} h_i^{E_i} \cdot g_{1E})^{r_{1E}}, g^{r_{1E}}, h_{n+1}^{r_{1E}}, \ldots, h_\ell^{r_{1E}}).$$

- For each $Y_1 = (\hat{N}_0, \hat{N}_1, \ldots, \hat{N}_m) \in Y_{t_2+1}$, it picks randomly $r_{2N} \in \mathbb{Z}_p$. and computes:

$$d_{Y_1} = (g^{z} \cdot (\prod_{i=0}^{m} h_i^{\hat{N}_i} \cdot g_{2N})^{r_{2N}}, g^{r_{2N}}, h_{m+1}^{r_{2N}}, \ldots, h_\ell^{r_{2N}}).$$

- For each $Y_2 = (W_0, W_1, \ldots, W_n) \in Y_{2T-t_2}$, it picks randomly $r_{2W} \in \mathbb{Z}_p$. and computes:

$$d_{Y_2} = (g^{-z} \cdot (\prod_{i=0}^{n} h_i^{W_i} \cdot g_{2W})^{r_{2W}}, g^{r_{2W}}, h_{n+1}^{r_{2W}}, \ldots, h_\ell^{r_{2W}}).$$

- For each $Z_1 = (N'_0, N'_1, \ldots, N'_m) \in Z'_{t_3+1}$, the algorithms picks randomly $r_{3N} \in \mathbb{Z}_p$, and computes:

$$d_{Z_1} = (g^{k} \cdot (\prod_{i=0}^{m} h_i^{N'_i} \cdot g_{3N})^{r_{3N}}, g^{r_{3N}}, h_{m+1}^{r_{3N}}, \ldots, h_\ell^{r_{3N}}).$$

- For each $Z_2 = (E'_0, E'_1, \ldots, E'_n) \in Z'_{2T-t_3}$, it picks randomly $r_{3E} \in \mathbb{Z}_p$. and computes:

$$d_{Z_2} = (g^{-k} \cdot (\prod_{i=0}^{n} h_i^{E'_i} \cdot g_{3E})^{r_{3E}}, g^{r_{3E}}, h_{n+1}^{r_{3E}}, \ldots, h_\ell^{r_{3E}}).$$

Finally, it sets:

$$SK_{t_{1,1N}} = \{d_{X_1}\}_{X_1 \in X_{t_1+1}} SK_{t_{1,1E}} = \{d_{X_2}\}_{X_2 \in X_{2T-t_1}}$$
$$SK_{t_{2,2N}} = \{d_{Y_1}\}_{Y_1 \in Y_{t_2+1}} SK_{t_{2,2W}} = \{d_{Y_2}\}_{Y_2 \in Y_{2T-t_2}},$$
$$SK_{t_{3,3N}} = \{d_{Z_1}\}_{Z_1 \in X_{t_3+1}} SK_{t_{3,3E}} = \{d_{Z_2}\}_{Z_2 \in X_{2T-t_3}}$$

and returns $SK_{t_1,t_2} = \{SK_{t_{1,1N}}, SK_{t_{1,1E}}, SK_{t_{2,2N}}, SK_{t_{2,2W}}, SK_{t_{3,3N}}, SK_{t_{3,3E}}, \{t_1, t_2, t_3\}\}$.

▶ Decryption(SK_{t_1,t_2,t_3}, CT): If $\{t_1, t_2, t_3\} \notin ([t_{1N}, t_{1E}], [t_{2N}, t_{2W}], [t_{3N}, t_{3E}])$ return $\perp$. Otherwise, the algorithm retrieves:

$$d_{X_{t_{1E}+1}} = \{N_1, N_2, \ldots\}, d_{X_{2T-t_{1N}}} = \{E_1, E_2, \ldots\},$$
$$d_{Y_{t_{2W}+1}} = \{\hat{N}_1, \hat{N}_2, \ldots\}, d_{Y_{2T-t_{2N}}} = \{W_1, W_2, \ldots\},$$
$$d_{Z_{t_{3E}+1}} = \{N'_1, N'_2, \ldots\}, d_{Z_{2T-t_{3N}}} = \{E'_1, E'_2, \ldots\},$$

Then, it computes:

$$\frac{C_0 \cdot e(d_{X_{12}}, C_2) \cdot e(d_{X_{22}}, C_3) \cdot e(d_{Y_{12}}, C_4) \cdot e(d_{Y_{22}}, C_5) \cdot e(d_{Z_{12}}, C_6) \cdot e(d_{Z_{22}}, C_7)}{e(d_{X_{11}} \cdot d_{X_{21}} \cdot d_{Y_{11}} \cdot d_{Y_{21}} \cdot d_{Z_{11}} \cdot d_{Z_{21}}, C_1)}$$

Let:

$$\begin{aligned}
A &= e(d_{X_{12}}, C_2) \cdot e(d_{X_{22}}, C_3) \cdot e(d_{Y_{12}}, C_4) \cdot e(d_{Y_{22}}, C_5) \cdot e(d_{Z_{12}}, C_6) \cdot e(d_{Z_{22}}, C_7) \\
&= e(g^{r_{1N}}, (\prod_{i=0}^{m} h_i^{N_i} \cdot g_{1N})^s) \cdot e(g^{r_{1E}}, (\prod_{i=0}^{n} h_i^{E_i} \cdot g_{1E})^s) \\
&\quad \cdot e(g^{r_{2N}}, (\prod_{i=0}^{m} h_i^{\hat{N}_i} \cdot g_{2N})^s) \cdot e(g^{r_{2W}}, (\prod_{i=0}^{m} h_i^{W_i} \cdot g_{2W})^s) \\
&\quad \cdot e(g^{r_{3N}}, (\prod_{i=0}^{m} h_i^{N'_i} \cdot g_{3N})^s) \cdot e(g^{r_{3E}}, (\prod_{i=0}^{n} h_i^{E'_i} \cdot g_{3E})^s) \\
B &= e(d_{X_{11}} \cdot d_{X_{21}} \cdot d_{Y_{11}} \cdot d_{Y_{21}} \cdot d_{Z_{11}} \cdot d_{Z_{21}}, C_1) \\
&= e(g^{\alpha\beta+r} \cdot (\prod_{i=0}^{m} h_i^{N_i} \cdot g_{1N})^{r_{1N}} \cdot g^{-r} \cdot (\prod_{i=0}^{n} h_i^{E_i} \cdot g_{1E})^{r_{1E}} \\
&\quad \cdot g^{z} \cdot (\prod_{i=0}^{m} h_i^{\hat{N}_i} \cdot g_{2N})^{r_{2N}} \cdot g^{-z} \cdot (\prod_{i=0}^{n} h_i^{W_i} \cdot g_{2W})^{r_{2W}} \\
&\quad \cdot e(g^{k} \cdot (\prod_{i=0}^{m} h_i^{N'_i} \cdot g_{3N})^{r_{3N}} \cdot g^{-k} \cdot (\prod_{i=0}^{n} h_i^{E'_i} \cdot g_{3E})^{r_{3E}}, g^s)
\end{aligned}$$

To recover message M:

$$\frac{Y^s \cdot M \cdot A}{B} = M$$

Security Proof

Theorem 2. *Assume that the ℓ-wBDHI assumption holds, then no polynomial-time adversary against our 3D Location Based Encryption scheme can have a non-negligible advantage over random guess in the Selective IND-CPA security game.*

We assume our 3D Location Based Encryption with the size of X,Y, and Z coordinate which is polynomial in the security parameter k. We consider the selective adversary the decides the challenge X,Y, and Z $[t^*_{1N}, t^*_{1E}][t^*_{2N}, t^*_{2W}][t^*_{3N}, t^*_{3E}]$ at the beginning of the IND-CPA game.

Let $\mathcal{A}$ be any IND-CPA adversary that attacks our proposed scheme. We then build an algorithm $\mathcal{B}$ that solves the decisional ℓ-wBDHI problem in $(\mathbb{G}, \mathbb{G}_T, e)$ by using $\mathcal{A}$ as in Theorem 1. Let g, h choose uniformly in $\mathbb{G}$, randomly $\alpha \in \mathbb{Z}_p$, and sets $y_i = g^{\alpha^{i+1}}$. $\mathcal{B}$ is given as input $(g, h, y_0, y_1, \ldots, y_\ell, Y)$, where Y is $e(g,h)^{\alpha^{l+2}}$ or a random value in $\mathbb{G}_T$. $\mathcal{B}$ interacts with $\mathcal{A}$ as follows:

– **Setup**: $\mathcal{A}$ outputs the challenge X $[t^*_{1N}, t^*_{1E}]$, and Y $[t^*_{2N}, t^*_{2W}]$. Then lets:

$$X_{t^*_{1E}+1} = (N^*_0, N^*_1, \dots N^*_m), X_{2T-t^*_{1N}} = (E^*_0, E^*_1, \dots E^*_n),$$
$$Y_{t^*_{2W}+1} = (\hat{N}^*_0, \hat{N}^*_1, \dots \hat{N}^*_m), Y_{2T-t^*_{2N}} = (W^*_0, W^*_1, \dots W^*_n),$$
$$Z_{t^*_{3E}+1} = (N'^*_0, N'^*_1, \dots N'^*_m), Z_{2T-t^*_{3N}} = (E'^*_0, E'^*_1, \dots E'^*_n),$$

$\mathcal{B}$ picks randomly $\gamma, \gamma_0, \gamma_1, \dots, \gamma_\ell, \alpha_N, \alpha_E, \alpha_{\hat{N}}, \alpha_W, \alpha'_N, \alpha'_E, \in \mathbb{Z}_p$, and $g_1 = y_0$, then computes:

$$g_{1N} = g^{\alpha_N} \cdot \prod_{i=0}^{m} y_{\ell-1}^{N^*_i} g_{1E} = g^{\alpha_E} \cdot \prod_{i=0}^{n} y_{\ell-1}^{E^*_i}$$
$$g_{2N} = g^{\alpha_{\hat{N}}} \cdot \prod_{i=0}^{m} y_{\ell-1}^{\hat{N}^*_i} g_{2W} = g^{\alpha_W} \cdot \prod_{i=0}^{n} y_{\ell-1}^{W^*_i}$$
$$g_{3N} = g^{\alpha'_N} \cdot \prod_{i=0}^{m} y_{\ell-1}^{N'^*_i} g_{3E} = g^{\alpha'_E} \cdot \prod_{i=0}^{n} y_{\ell-1}^{E'^*_i}$$
$$h_i = g^{\gamma_i} \cdot y_{\ell-i} = g^{\gamma_i - \alpha^{\ell-i+1}}, Y = e(y_0, y_\ell g^\gamma),$$

where $\alpha^{\ell+1} + \gamma$ is implicitly setting as β. $\mathcal{B}$ then sets $MPK = (g, g_1, g_{1N}, g_{1E}, g_{2N}, g_{2W}, h_0, \dots, h_\ell, Y = e(g^\alpha, g^\beta))$, and gives it to $\mathcal{A}$.

– **Phase 1**: If $\mathcal{A}$ submits a location extraction query $t_{1_i}, t_{2_i}, t_{3_i}$, $\mathcal{B}$ responds to each query by generating $SK_{t_{1_i}, t_{2_i}}$ as follows:
  - Case 1: $t_{1_i} < t^*_{1N}, t_{2_i} \in [t^*_{2N}, t^*_{2W}], t_{3_i} \in [t^*_{3N}, t^*_{3E}]$
  - Case 2: $t_{1_i} > t^*_{1E}, t_{2_i} \in [t^*_{2N}, t^*_{2W}], t_{3_i} \in [t^*_{3N}, t^*_{3E}]$
  - Case 3: $t_{1_i} \in [t^*_{1N}, t^*_{1E}], t_{2_i} < t^*_{2N}, t_{3_i} \in [t^*_{3N}, t^*_{3E}]$
  - Case 4: $t_{1_i} \in [t^*_{1N}, t^*_{1E}], t_{2_i} > t^*_{2W}, t_{3_i} \in [t^*_{3N}, t^*_{3E}]$
  - Case 5: $t_{1_i} \in [t^*_{1N}, t^*_{1E}], t_{2_i} \in [t^*_{2N}, t^*_{2W}], t_{3_i} < t^*_{3N}$
  - Case 6: $t_{1_i} \in [t^*_{1N}, t^*_{1E}], t_{2_i} \in [t^*_{2N}, t^*_{2W}], t_{3_i} > t^*_{3E}$

  This query phase 1 is simulated similarly as in Phase 1.

– **Challenge**: When $\mathcal{A}$ decides that Phase 1 is over, it outputs the challenge plaintexts M_0, M_1. $\mathcal{B}$ picks a random bit $b \in_U \{0,1\}$, and computes the challenge ciphertext by

$$CT^* = (M_b \cdot T \cdot e(y_0, h^\gamma), h, h^{\alpha_N + \sum_{i=0}^{m} N^*_i \gamma_i}, h^{\alpha_E + \sum_{i=0}^{n} E^*_i \gamma_i}, h^{\alpha_{\hat{N}} + \sum_{i=0}^{m} \hat{N}^*_i \gamma_i},$$
$$h^{\alpha_W + \sum_{i=0}^{n} NW*_i \gamma_i}, h^{\alpha'_N + \sum_{i=0}^{m} N'^*_i \gamma_i}, h^{\alpha'_E + \sum_{i=0}^{n} E'^*_i \gamma_i}, [t^*_{1N}, t^*_{1E}][t^*_{2N}, t^*_{2W}][t^*_{3N}, t^*_{3E}])$$

If $T = e(g,h)^{\alpha^\ell + 2}$, then CT^* is of the following form by letting $\log_g h = s$

$$CT^* = (M_b \cdot e(g,h)^{\alpha^{\ell+2} s} \cdot e(g,g)^{\alpha\gamma s}, g^s, (g^{\alpha_N} \cdot \prod_{i=0}^{m} y_{\ell-1}^{N^*_i})^s, (g^{\alpha_E} \cdot \prod_{i=0}^{n} y_{\ell-1}^{E^*_i})^s,$$
$$(g^{\alpha_{\hat{N}}} \cdot \prod_{i=0}^{m} y_{\ell-1}^{\hat{N}^*_i})^s, (g^{\alpha_W} \cdot \prod_{i=0}^{n} y_{\ell-1}^{W^*_i})^s, (g^{\alpha'_N} \cdot \prod_{i=0}^{m} y_{\ell-1}^{N'^*_i})^s, (g^{\alpha'_E} \cdot \prod_{i=0}^{n} y_{\ell-1}^{E'^*_i})^s)$$

$\mathcal{B}$ sends the following challenge ciphertext to $\mathcal{A}$:

$$CT^* = (M_b T, C_1, C_2, C_3, C_4, C_5, C_6, C_7, [t^*_{1N}, t^*_{1E}][t^*_{2N}, t^*_{2W}][t^*_{3N}, t^*_{3E}]).$$

- **Phase II**: Same as Phase I.
- **Guess**: $\mathcal{A}$ outputs $b' \in \{0,1\}$. If $b' = b$ then $\mathcal{B}$ outputs 1, otherwise outputs 0. **Analysis**: If $T = e(g,h)^{\alpha^{\ell+2}}$, then the simulation is the same as in the real game. Hence, $\mathcal{A}$ will have the probability $\frac{1}{2} + \epsilon$ to guess b correctly. If T is a random element of $\mathbb{G}_T$, then $\mathcal{A}$ will have probability $\frac{1}{2}$ to guess b correctly. Therefore, $\mathcal{B}$ can solve the Decisionℓ-wBDHI assumption also with advantage ϵ. □

6 Conclusion

This work is the first endeavor to develop Location Based Encryption with constant ciphertext size. We proposed two new schemes, called 2D Location Based Encryption and 3D Location Based Encryption. Both of them are constant ciphertext size and are proven under in the selective model under the decisional $\ell-$wBDHI assumption. In future work, we will consider the privacy of the area purposed for encryption, since the ciphertext component can disclose the area information. This leads to a privacy preserving location scheme to protect the information encryption, and guarantee the user's location. Furthermore, we will deploy our proposed schemes on IoT devices to analyze the efficiency of transmitting message protocol in the practical scenario.

References

1. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: 2007 IEEE Symposium on Security and Privacy (SP 2007), pp. 321–334 (2007)
2. Buhrman, H., et al.: Position-based quantum cryptography: impossibility and constructions. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 429–446. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22792-9_24
3. Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 207–222. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_13
4. Chandran, N., Goyal, V., Moriarty, R., Ostrovsky, R.: Position based cryptography. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 391–407. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03356-8_23
5. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006, pp. 89–98 (2006)
6. Kasamatsu, K., Matsuda, T., Emura, K., Attrapadung, N., Hanaoka, G., Imai, H.: Time-specific encryption from forward-secure encryption. In: Visconti, I., De Prisco, R. (eds.) SCN 2012. LNCS, vol. 7485, pp. 184–204. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32928-9_11

7. Katz, J., Sahai, A., Waters, B.: Predicate encryption supporting disjunctions, polynomial equations, and inner products. In: Smart, N. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 146–162. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78967-3_9
8. Paterson, K.G., Quaglia, E.A.: Time-specific encryption. In: Garay, J.A., De Prisco, R. (eds.) SCN 2010. LNCS, vol. 6280, pp. 1–16. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15317-4_1
9. Shi, E., Bethencourt, J., Chan, T.H.H., Song, D., Perrig, A.: Multi-dimensional range query over encrypted data. In: Proceedings of the 2007 IEEE Symposium on Security and Privacy, SP 2007, pp. 350–364 (2007)
10. Yang, R., Xu, Q., Au, M.H., Yu, Z., Wang, H., Zhou, L.: Position based cryptography with location privacy: a step for fog computing. Future Gener. Comput. Syst. **78**, 799–806 (2018)
11. You, L., Chen, Y., Yan, B., Zhan, M.: A novel location-based encryption model using fuzzy vault scheme. Soft Comput. **22**, 3383–3393 (2018)