Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.051181

ARTICLE

Vector Dominance with Threshold Searchable Encryption (VDTSE) for the Internet of Things

Jingjing Nie[1,*] **and Zhenhua Chen**[2]

[1]College of Safety Science and Engineering, Xi'an University of Science and Technology, Xi'an, 710054, China

[2]College of Computer Science and Technology, Xi'an University of Science and Technology, Xi'an, 710054, China

*Corresponding Author: Jingjing Nie. Email: niejingjing99@gmail.com

Received: 28 February 2024 Accepted: 19 April 2024 Published: 20 June 2024

ABSTRACT

The Internet of Medical Things (IoMT) is an application of the Internet of Things (IoT) in the medical field. It is a cutting-edge technique that connects medical sensors and their applications to healthcare systems, which is essential in smart healthcare. However, Personal Health Records (PHRs) are normally kept in public cloud servers controlled by IoMT service providers, so privacy and security incidents may be frequent. Fortunately, Searchable Encryption (SE), which can be used to execute queries on encrypted data, can address the issue above. Nevertheless, most existing SE schemes cannot solve the vector dominance threshold problem. In response to this, we present a SE scheme called Vector Dominance with Threshold Searchable Encryption (VDTSE) in this study. We use a Lagrangian polynomial technique and convert the vector dominance threshold problem into a constraint that the number of two equal-length vectors' corresponding bits excluding wildcards is not less than a threshold t. Then, we solve the problem using the proposed technique modified in Hidden Vector Encryption (HVE). This technique makes the trapdoor size linear to the number of attributes and thus much smaller than that of other similar SE schemes. A rigorous experimental analysis of a specific application for privacy-preserving diabetes demonstrates the feasibility of the proposed VDTSE scheme.

KEYWORDS

Internet of Things (IoT); Internet of Medical Things (IoMT); vector dominance with threshold searchable encryption (VDTSE); threshold comparison; electronic healthcare

1 Introduction

The Internet of Things (IoT) [1] provides safe and controllable real-time online monitoring, positioning, and other service functions. As an important IoT application in the medical field, the Internet of Medical Things (IoMT) can realize efficient and inexpensive healthcare and enhance patient comfort. However, the data involved in continuous monitoring can be massive, and the devices used to collect the data are resource-constrained. As a result, the data is often stored in the cloud, which may seriously violate patient privacy during data collection, storage, and computation. Suspecting the infringement of their healthcare privacy, patients may become reluctant to participate or encrypt their medical data before uploading it to service providers. This raises another question: how to perform a

user search about this encrypted private data. In this context, Searchable Encryption (SE) is desirable as a prominent encryption tool for privacy-preserving IoMT [2,3].

1.1 Motivation

Consider a specific example of privacy-preserving diabetes screening (see Fig. 1) in IoMT. In this example, a nurse tries to screen out diabetics from Personal Health Records (PHRs). However, PHRs contain private information, and the patient is unwilling to share them with anyone. Therefore, the patient in this example encrypts the medical record $\vec{X}$ (random blood glucose = 4, fasting blood glucose = 3, 2-h post-load glucose = 5, glycated hemoglobin = 4) and the personal information M before uploading them to the service provider. In this case, the nurse sends a search request $(\vec{\Delta}, t)$ (random blood glucose = 3, fasting blood glucose = 2, 2-h post-load glucose = 3, glycated hemoglobin = 5, $t = 2$ $(1 \leq t \leq 4)$) to the service provider using a handheld Personal Digital Assistant (PDA). When the service provider detects *number* $\left(\vec{X} \succ \vec{\Delta}\right) \geq t$ ($\vec{X} \succ \vec{\Delta}$ means that $\vec{X}$ dominates $\vec{\Delta}$ [4]), it will send the encrypted M to the nurse. In this process, the service provider has no prior knowledge of $\vec{X}$ and M.

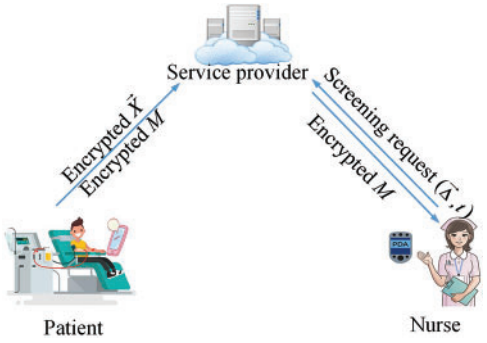

Figure 1: Privacy-preserving diabetes screening

For the application above, it is necessary to solve *number* $\left(\vec{X} \succ \vec{\Delta}\right) \geq t$ in SE. It is very helpful when we need to compare multiple attributes while counting the number of matches. We call the proposed scheme Vector Dominance [4] with Threshold Searchable Encryption (VDTSE). The vector dominance with threshold problem determines whether the number of components in vectors $\vec{X} = (X_1, \ldots, X_n)$ and $\vec{\Delta} = (\Delta_1, \ldots, \Delta_n)$ $(X_i > \Delta_i, i \in [1, n])$ reaches a certain threshold t.

Therefore, the proposed VDTSE scheme is favorable for threshold comparison queries in a public-key SE system. When *number* $\left(\vec{X} \succ \vec{\Delta}\right) \geq t$ holds, the encrypted M will be sent to the requester. Besides, our algorithm can solve the problem of existing SE schemes by controlling the threshold t, which is adaptive and can support additional functionalities such as range queries $(t = 0)$ or similarity searches $(X_i = \Delta_i)$.

1.2 Our Contribution

The main contributions of this study are outlined below:

• We design a VDTSE scheme supporting comparable attributes that can function for the (*t, n*)-threshold policy. Although existing schemes [5–8] also support this feature, they are restricted to AND-gates (or (*t, t*)-threshold). We implement the threshold using a Lagrangian polynomial technique.

• The VDTSE scheme has a shorter trapdoor that makes it more suitable for data storage on mobile devices (such as PDAs). To achieve this, we solve it using our technique modified in Hidden Vector Encryption (HVE) [7], which makes the trapdoor size linear.

• We prove the security, flexibility, and effectiveness of the proposed scheme through theoretical comparison with other schemes. Finally, the experiment shows that the trapdoor size of our method is much smaller than that of other similar SE schemes.

1.3 Outline

The rest of this paper is structured as below. Section 2 describes the related work of VDTSE. Section 3 contains the preliminaries, while Section 4 presents the scheme construction. Section 5 discusses the security proof for our scheme, Section 6 describes the performance of the proposed scheme, and Section 7 concludes the article and suggests possible future work.

2 Related Work

Song et al. popularized the SE scheme [9] that enabled a user to generate both ciphertexts and trapdoors under a symmetric system, so service providers could perform a match search on encrypted information. Under the symmetric system, SE was further improved in [10–12]. In 2004, Boneh et al. [13] considered the first asymmetric SE scheme, Public-key Encryption with Keyword Search (PEKS). The initial public-key SE schemes, however, were limited to test keyword equality. Among these developments, comparative searches received little attention.

To address the aforementioned issue, Boneh et al. [5] created a searchable public-key system called HVE in 2007, which allowed for conjunctive range queries over encrypted data. That same year, Shi et al. [6] proposed a multi-dimensional range SE scheme. However, the aforementioned methods were inefficient and disadvantageous from an operational standpoint. As a result, Park [7] proposed a novel HVE that was effective in prime-order groups significantly smaller than the composite-order groups on the identical level. Later, Park et al. [8] realized a more effective HVE scheme. Although the schemes above supported the AND gate of attribute comparison, they could not apply to arbitrary threshold gates. Fortunately, Sun et al. [14] extended the technique by combining HVE and predicate encryption (PE) for the inner product (called IPE) to achieve threshold comparison queries.

In addition to these references, other scholars addressed the threshold comparison. In 2007, Bethencourt et al. [15] presented a threshold comparison scheme to realize numeric ranges search. In 2018, Attrapadung et al. [16] developed a numeric comparison scheme under the Attribute-Based Encryption (ABE) system. In 2017, Xue et al. [17] constructed a comparable ABE scheme that was more efficient. Although these schemes [15–17] were under the public key system, they could not protect the privacy of $\vec{X}$.

To the best of our knowledge, the only scheme that supported the threshold comparison was [14], but their ciphertext and private key size increased quadratically, posing potential challenges in resource-constrained IoT devices. Therefore, we need to design a more efficient SE scheme.

3 Preliminaries

3.1 Bilinear Maps

Assume g is a generator in $\mathbb{G}$, while $\mathbb{G}$ and $\mathbb{G}_T$ are two multiplicative cyclic groups whose order is prime p. e: $\mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a function with these attributes:

1. Bilinear. We have $e\left(u^c, v^d\right) = e\left(u, v\right)^{cd}$, where $u, v \in \mathbb{G}$, $c, d \in \mathbb{Z}_p$.

2. Nondegenerate. $e\left(g, g\right) \neq 1$.

3. Computable. e is a computable map algorithm.

Therefore, we state that e is a bilinear pairing map within $\mathbb{G}$. It is worth noting that $e\left(,\right)$ is symmetric because $e\left(g^c, g^d\right) = e\left(g, g\right)^{cd} = e\left(g^d, g^c\right)$.

3.2 Complexity Assumptions

Decisional Bilinear Diffie-Hellman (DBDH) Assumption. The following describes the DBDH problem: Provided $\left(g, g^a, g^b, g^c, Z\right) \in \mathbb{G}^4 \times \mathbb{G}_T$, identify $Z = e\left(g, g\right)^{abc}$ or Z is random in $\mathbb{G}_T$.

Augmented Decision Linear (ADLIN) Assumption. Provided $(g, g^{z_1}, g^{z_2}, g^{z_2^2}, g^{z_2/z_1}, g^{z_2^2 z_3}, g^{z_4}, Z) \in \mathbb{G}^8$, find $Z = g^{z_1\left(z_3+z_4\right)}$ or Z is random in $\mathbb{G}$.

Definition 1. The {DBDH, ADLIN} assumption holds in $\mathbb{G}$ when the advantage of any polynomial time algorithm in dealing with the {DBDH, ADLIN} problem is negligible.

3.3 Security Model

Our scheme is characterized by a chosen plaintext security, and ciphertext CT offers no information about the vector $\vec{X}$ and the message M, which is proved in a security proof that relies on the difficulty of DBDH and ADLIN. The following games are played between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. Besides, U and ℓ are given to $\mathcal{A}$, where U is the attribute universal set and ℓ is the length of vector.

• **Initialization (Init).** $\mathcal{A}$ commits $\vec{X}_0^*,\ \vec{X}_1^* \in \Sigma^{\ell/U}$ to $\mathcal{C}$.

• **Setup.** $\mathcal{C}$ runs the KeyGen algorithm to get a public key PK and a secret key SK, offers PK to $\mathcal{A}$.

• **Query Phase 1.** $\mathcal{A}$ performs a series of trapdoor queries adaptively. To any trapdoor $T_{\vec{\Delta}_i}$ from queried vectors $\vec{\Delta}_1, \ldots, \vec{\Delta}_{q_\eta} \in (\Sigma)^{\ell/U}$ and thresholds $t_1, \ldots, t_{q_\eta}$, there is a restriction of $f_{\vec{\Delta}_i, t_i}\left(\vec{X}_0^*\right) = f_{\vec{\Delta}_i, t_i}\left(\vec{X}_1^*\right)$ for all $i = 1, \ldots, q_\eta$. $\mathcal{C}$ uses trapdoors $\mathrm{TK}_{\vec{\Delta}_i, t_i} \leftarrow \mathrm{Trapdoor}\left(\mathrm{SK}, \vec{\Delta}_i, t_i\right)$ to respond to $\mathcal{A}$.

• **Challenge.** $\mathcal{A}$ offers $M_0, M_1 \in \mathcal{M}$. If $f_{\vec{\Delta}_i, t_i}\left(\vec{X}_0^*\right) = f_{\vec{\Delta}_i, t_i}\left(\vec{X}_1^*\right) = 1$, then $M_0 = M_1$. $\mathcal{C}$ responds $\mathrm{CT}^* \leftarrow PEKS\left(\mathrm{PK}, \vec{X}_\beta^*, M_\beta\right)$ to $\mathcal{A}$, where $\beta \in \{0, 1\}$.

• **Query Phase 2.** $\mathcal{A}$ makes other trapdoor queries adaptively for vectors $\vec{\Delta}_{q_{n+1}}, \ldots, \vec{\Delta}_{q_{\mathrm{TK}}}$ and thresholds $t_{q_{n+1}}, \ldots, t_{q_{\mathrm{TK}}}$. The restrictions are the same as those in Query Phase 1 and Challenge.

• **Guess.** $\mathcal{A}$ guesses β'. $\mathcal{A}$ wins the game when $\beta' = \beta$, where $\beta' \in \{0, 1\}$.

We define $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{VDTSE}} = |\Pr[\beta' = \beta] - 1/2|$ as the advantage that $\mathcal{A}$ has in breaking the VDTSE scheme.

Definition 2. If the advantage $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{VDTSE}}$ is negligible, the VDTSE scheme is selectively secure.

4 Construction

4.1 A Novel Encoding

This section presents the techniques for solving $number\left(\vec{X} \succ \vec{\Delta}\right) \geq t$ in the form of a novel encoding. We set a universal set $U = \{1, 2, \ldots, |U|\}$. Every bit x_j of a $|\vec{\Delta}| \times |U|$-dimension vector $\vec{x}$ converted from vector $\vec{X}$ is chosen from $\{0, 1\}$ following Eq. (1). In a similar way, every bit σ_j of another $|\vec{\Delta}| \times |U|$-dimension vector $\vec{\sigma}$ transformed from vector $\vec{\Delta}$ is selected from $\{1, *\}$ according to Eq. (2).

$$x_{(i-1)\times|U|+j} = \begin{cases} 1, & X_i < j, \\ 0, & otherwise, \end{cases} \tag{1}$$

where $1 \leq j \leq |U|$, $1 \leq i \leq |\vec{X}|$.

$$\sigma_{(i-1)\times|U|+j} = \begin{cases} 1, & \Delta_i = j, \\ *, & otherwise, \end{cases} \tag{2}$$

where $1 \leq j \leq |U|$, $1 \leq i \leq |\vec{\Delta}|$.

When the VDTSE scheme is used in multi-user scenarios, a trusted private service provider needs to be added to receive trapdoors from users and then send the trapdoors to an untrusted public service provider. Therefore, only single-user scenarios are considered in this paper.

Applying the example of privacy-preserving diabetes screening in the Introduction, set $\vec{X} = (X_1, X_2, X_3, X_4) = (4, 3, 5, 4)$, $\vec{\Delta} = (\Delta_1, \Delta_2, \Delta_3, \Delta_4) = (3, 2, 3, 5)$, $t = 2$, $\ell = |\vec{X}| \times |U| = 20$, where $U = \{1, 2, 3, 4, 5\}$ and $|\vec{X}| = |\vec{\Delta}| = 4$. Two 20-dimension vectors $\vec{x}$ and $\vec{\sigma}$ are transformed from $\vec{X}$ and $\vec{\Delta}$ following Eqs. (1) and (2), respectively.

For $\vec{X}$, the 20-dimension vector $\vec{x}$ is represented by

1	1	1	0	0	1	1	0	0	0	1	1	1	1	0	1	1	1	0	0

For $\vec{\Delta}$, the 20-dimension vector $\vec{\sigma}$ is represented by

*	*	1	*	*	*	1	*	*	*	*	*	1	*	*	*	*	*	*	1

Since there are 3 ($>t$) equal corresponding bits of $\vec{x}$ and $\vec{\sigma}$ excluding $*$ bits, $number\left(\vec{X} \succ \vec{\Delta}\right) \geq t$ holds. The detailed scheme is as follows.

4.2 Scheme

• **KeyGen** $(k,\ \ell,\ U)$. The KeyGen algorithm uses a security parameter $k = 1024$ bit and a type-A elliptic curve that has as input a 160-bit group order, a vector length ℓ, a universe U, and a random generator $g \in \mathbb{G}$ to generate our scheme parameters. It selects random exponents $y_1, v_1, \ldots, v_\ell$, $t_1, \ldots, t_\ell \in \mathbb{Z}_p$ as well as random elements α, $(h_1,\ u_1,\ w_1)\ , \ldots, (h_\ell,\ u_\ell,\ w_\ell) \in \mathbb{G}$. It defines $g_1 = g^\alpha$, $Y_1 = g^{y_1}$, $V_i = g^{v_i}$, $T_i = g^{t_i}$ for $i = 1, \ldots, \ell$. Furthermore, it establishes $\Omega = e\,(g_1,\ Y_1) \in \mathbb{G}_T$. The public key PK and the secret key SK are

$$\mathrm{PK} = \left(g,\ Y_1,\ (h_1, u_1, w_1, V_1, T_1)\ , \ldots, (h_\ell, u_\ell, w_\ell, V_\ell, T_\ell)\ , \Omega\right) \in \mathbb{G}^{5\ell+2} \times \mathbb{G}_T,$$
$$\mathrm{SK} = \left(\alpha, g_1 = g^\alpha, y_1, v_1, \ldots, v_\ell, t_1, \ldots, t_\ell\right) \in \mathbb{Z}_p^{2\ell+2} \times \mathbb{G}.$$

• **PEKS** $\left(\mathrm{PK},\ \vec{X}\right)$. With PK and vector $\vec{X} = \left(X_1, \ldots, X_{\ell/U}\right)$ as input, this algorithm initially converts $\vec{X}$ into $\vec{x} = (x_1, \ldots, x_\ell)$ based on Eq. (1). It chooses two random parameters $s_1,\ s_2 \in \mathbb{Z}_p$, then uses them to encrypt message $M \in \mathbb{G}_T$ and vector $\vec{x}$ for generating ciphertext:

$$\mathrm{CT} = \left(Y_1^{s_1}, g^{s_2}, \left(h_1 u_1^{x_1}\right)^{s_1} V_1^{s_2}, \ldots, \left(h_\ell u_\ell^{x_\ell}\right)^{s_1} V_\ell^{s_2}, w_1^{s_1} T_1^{s_2}, \ldots, w_\ell^{s_1} T_\ell^{s_2}, \Omega^{s_1} M\right) \in \mathbb{G}^{2\ell+2} \times \mathbb{G}_T.$$

This encryption algorithm ensures the confidentiality of M and $\vec{x}$. Their integrity can be recognized by the Message Authentication Code (MAC), public key-based Homomorphic Linear Authentication (HLA), Hash-based Message Authentication Code (HMAC), etc.

• **Trapdoor** $\left(\mathrm{SK},\ \vec{\Delta},\ t\right)$. This algorithm uses SK and vector $\vec{\Delta} = \left(\Delta_1, \ldots, \Delta_J, \ldots, \Delta_{\ell/U}\right)$ as input, then chooses at random a t-1 degree polynomial $q\,(x)$ where $q\,(0) = \alpha$. It first transforms $\vec{\Delta}$ into $\vec{\sigma} = (\sigma_1, \ldots, \sigma_\ell) \in (\Sigma_*)^\ell$ based on Eq. (2). To generate a trapdoor $\mathrm{T}_{\vec{\sigma}}$ for $i \in S\left(\vec{\sigma}\right)$, where $S\left(\vec{\sigma}\right)$ is the set of all indexes i that satisfies $\sigma_i \neq *$, it computes $\mathrm{T}_{\vec{\sigma}}$ as

$$\mathrm{T}_{\vec{\sigma}} = \left(\sigma_*, \forall i \in S\left(\vec{\sigma}\right) :\ \left\{g^{q(J)} \left(h_i u_i^{\sigma_i}\right)^{r_J} w_i^{\eta_J},\ Y_1^{r_J},\ Y_1^{\eta_J},\ Y_1^{-\left(v_i r_J + t_i \eta_J\right)},\ V_i^{y_1 Z_J},\ Y_1^{Z_J},\ \left(h_i u_i^{\sigma_i}\right)^{Z_J}\right\}_{J=1}^{\ell/U}\right) \in \mathbb{G}^{7(\ell/U)},$$
$$i = (J-1)\,U + \Delta_J.$$

• **Test** $\left(\mathrm{CT}, \mathrm{T}_{\vec{\sigma}}\right)$. Let $\mathrm{CT} = (C_1,\ C_2,\ C_{3,1}, \ldots, C_{3,i}, \ldots, C_{3,\ell},\ C_{4,1}, \ldots, C_{4,i}, \ldots, C_{4,\ell},\ C_5)$ and $\mathrm{T}_{\vec{\sigma}} = \left(K_{1,J}, K_{2,J}, K_{3,J}, K_{4,J}, K_{5,J}, K_{6,J}, K_{7,J}\right)$ for $1 \leq J \leq \ell/U$. The Test algorithm computes the following:

Step 1.

$$\frac{e\left(K_{7,J}, C_1\right) \cdot e\left(K_{5,J}, C_2\right)}{e\left(K_{6,J}, C_{3,i}\right)} = \frac{e\left(\left(h_i u_i^{\sigma_i}\right)^{Z_J},\ Y_1^{s_1}\right) \cdot e\left(V_i^{y_1 Z_J}, g^{s_2}\right)}{e\left(\left(h_i u_i^{x_i}\right)^{s_1} V_i^{s_2},\ Y_1^{Z_J}\right)} = e\,(u_i, g)^{y_1 s_1 Z_J (\sigma_i - x_i)} = 1,\ \text{when } \sigma_i = x_i.$$

Step 2.

When *number* $(x_i = \sigma_i) \geq t$ holds except for $*$ bits, chooses t elements of S satisfying $x_i = \sigma_i$. So we have

$$C_5 \Big/ \prod_{i \in S} \left[\frac{e\left(C_1, K_{1,J}\right)}{e\left(C_{3,i}, K_{2,J}\right) \cdot e\left(C_{4,i}, K_{3,J}\right) \cdot e\left(C_2, K_{4,J}\right)}\right]^{A_J} \to M,$$

where A_J is the Lagrange coefficient for i and S: $A_J\,(x) = \prod_{j \in S, j \neq i}[(x-j)\ /\ (i-j)]$.

5 Security Proof

5.1 Overview

If the DBDH and ADLIN assumptions are met, the VDTSE scheme is selectively secure. The adversary chooses two vectors $\vec{x}_0^* = \left(x_{0,1}^*, \ldots, x_{0,\ell}^*\right)$ and $\vec{x}_1^* = \left(x_{1,1}^*, \ldots, x_{1,\ell}^*\right) \in \Sigma^\ell$ that are transformed from vectors $\vec{X}_0^*, \vec{X}_1^*$ at the beginning of this security game. We assume that $A = \{1, 2, \ldots, |A|\}$, where A is the set of indexes i that satisfies $A = \left\{i \in \{1, \ldots, \ell\} \mid x_{0,i}^* \neq x_{1,i}^*\right\}$. We continue to choose random elements $(R_{3,1}, \ldots, R_{3,|A|}, R_{4,1}, \ldots, R_{4,|A|})$ from $\mathbb{G}$ and R_5 from $\mathbb{G}_T$. We assume the following games:

$$\text{Game}_0\colon \text{CT}_0 = \left(C_1, C_2, C_{3,1}, \ldots, C_{3,|A|}, C_{3,|A|+1}, \ldots, C_{3,\ell}, C_{4,1}, \ldots, C_{4,|A|}, C_{4,|A|+1}, \ldots, C_{4,\ell}, C_5\right),$$

$$\text{Game}_1\colon \text{CT}_1 = \left(C_1, C_2, C_{3,1}, \ldots, C_{3,|A|}, C_{3,|A|+1}, \ldots, C_{3,\ell}, C_{4,1}, \ldots, C_{4,|A|}, C_{4,|A|+1}, \ldots, C_{4,\ell}, R_5\right),$$

$$\text{Game}_{2,1}\colon \text{CT}_{2,1} = \left(C_1, C_2, R_{3,1}, \ldots, C_{3,|A|}, C_{3,|A|+1}, \ldots, C_{3,\ell}, R_{4,1}, \ldots, C_{4,|A|}, C_{4,|A|+1}, \ldots, C_{4,\ell}, R_5\right),$$

$$\vdots,$$

$$\text{Game}_{2,|A|}\colon \text{CT}_{2,|A|} = \left(C_1, C_2, R_{3,1}, \ldots, R_{3,|A|}, C_{3,|A|+1}, \ldots, C_{3,\ell}, R_{4,1}, \ldots, R_{4,|A|}, C_{4,|A|+1}, \ldots, C_{4,\ell}, R_5\right).$$

When the adversary sends M_0, M_1, the challenger provides the ciphertext about $\left(\vec{x}_\beta^*, M_\beta\right)$ to the adversary, where $\beta \in \{0, 1\}$. If the adversary guesses β correctly, he will win the game. Game_0 is the real security game that is given to the adversary. In this game, $C_{3,1}, \ldots, C_{3,\ell}, C_{4,1}, \ldots, C_{4,\ell}$ are the ciphertext about $\vec{x}_\beta^*$, and C_5 is the ciphertext about M_β. $\text{CT}_{2,|A|}$ is the challenge ciphertext of $\text{Game}_{2,|A|}$, which reveals nothing about the attributes associated with A or message M_β. We show this in the following games, which are all computationally indistinguishable.

5.2 Type of Trapdoor Queries

In our model (the selective security), the adversary performs trapdoor queries for any vector $\vec{\Delta} = (\Delta_1, \ldots, \Delta_J, \ldots, \Delta_{\ell/U})$ and a threshold t transformed to vector $\vec{\sigma} = (\sigma_1, \ldots, \sigma_\ell) \in (\Sigma_*)^\ell$ and t, under the constraint that $f_{\vec{\Delta},t}\left(\vec{X}_0^*\right) = f_{\vec{\Delta},t}\left(\vec{X}_1^*\right)$ $\left(\text{i.e., } f_{\vec{\sigma},t}\left(\vec{x}_0^*\right) = f_{\vec{\sigma},t}\left(\vec{x}_1^*\right)\right)$. We assume S is the set of i for which σ_i is not a wildcard. We divide the queries into two types:

- **Type 1.** $\left[number\left(X_{0,J} > \Delta_J\right) \geq t\right] \wedge \left[number\left(X_{1,J} > \Delta_J\right) \geq t\right]$ (i.e., $\left[number\left(\sigma_i = x_{0,i}^*\right) \geq t\right] \wedge$ $\left[number\left(\sigma_i = x_{1,i}^*\right) \geq t\right]$). If the challenge message meets $M_0 = M_1$, this type of query can be performed.
- **Type 2.** $[number(X_{0,J} > \Delta_J < t] \wedge [number(X_{1,J} > \Delta_J) < t]$ (i.e., $[number\left(\sigma_i = x_{0,i}^*\right) < t] \wedge$ $[number(\sigma_i) = x_{1,i}^*) < t]$.

Case 1. Type 1 does not match. Additionally, there is an index $i \in S \cap A$ for which $\sigma_i \neq x_{\beta,i}^*$.

Case 2. Type 1 and Case 1 fail to hold, and there exist $i \in S \cap A$ for which $\sigma_i = x_{\beta,i}^*$, while there exists $j \in S$ for which $\sigma_i \neq x_{\beta,i}^*$.

5.3 Proof of Lemmas

Lemma 1. Suppose that the DBDH assumption holds. In other words, the distinct difference in the advantage of $\mathcal{A}$ who is an adversary of any polynomial time in Game_0 and Game_1 is negligible.

Proof. Suppose $\mathcal{A}$ has a non-negligible difference about the advantage of G. If $Z = e(g,g)^{abc}$ holds for a random instance $\left(g, g^a, g^b, g^c, Z\right)$, $\mathcal{B}$ returns 1, otherwise, $\mathcal{B}$ returns 0. $\mathcal{B}$ communicates with $\mathcal{A}$ in the following ways:

• **Init.** At the start of the game, $\mathcal{A}$ produces two vectors $\vec{X}_0^*, \vec{X}_1^*$ that have been converted to $\vec{x}_0^*, \vec{x}_1^* \in \Sigma^\ell$. $\mathcal{B}$ internally changes $\beta \in \{0,1\}$.

• **Setup.** $\mathcal{B}$ selects exponents at random $\gamma,\ y_1, v_1, \ldots, v_\ell,\ t_1, \ldots,\ t_\ell,\ \theta_1, \ldots, \theta_\ell,\ \phi_1, \ldots, \phi_\ell,\ \lambda_1, \ldots, \lambda_\ell \in \mathbb{Z}_p$. It sets $Y_1 = g^{y_1}$, $h_i = g^{\theta_i}\left(g^b\right)^{-\phi_i \cdot x^*_{\beta,i}}$, $u_i = \left(g^b\right)^{\phi_i}$, $w_i = g^{\lambda_i}$, $V_i = g^{v_i}$, $T_i = g^{t_i}$. Besides, $\mathcal{B}$ establishes $\Omega = e(g_1, Y_1) = e\left(g^a,\ g^b\right)^{y_1} e(g,\ g)^{\gamma y_1}$. Take note that $g_1 = g^{ab}g^{\gamma}$, and this is secret to $\mathcal{B}$. $\mathcal{A}$ is handed the public key $PK = (g,\ Y_1,\ (h_1,\ u_1,\ w_1,\ V_1,\ T_1), \ldots, (h_\ell,\ u_\ell,\ w_\ell,\ V_\ell,\ T_\ell),\ \Omega)$.

• **Query Phase 1.** $\mathcal{A}$ triggers the trapdoor queries. Assume $\mathcal{A}$ searches for $\vec{\Delta}$ that has been changed to $\vec{\sigma} = (\sigma_1, \ldots, \sigma_\ell) \in (\Sigma_*)^\ell$. Let $S = \{i | \sigma_i \neq *\}$, where $*$ is a wildcard. The queries of $\mathcal{A}$ are divided into the following two types.

Type 1. $\mathcal{A}$ sends a Type 1 query, and $\mathcal{B}$ answers an arbitrary guess. The query above shows that $M_0 = M_1$, and $\mathcal{B}$ provides the message's encryption. Since in this circumstance the two games, Game_0 and Game_1, are identical, there ought to be no difference in $\mathcal{A}$'s advantage.

Type 2. These two scenarios imply that there exists at least one index $j \in S$ that satisfies $\sigma_j \neq x^*_{\beta,j}$. $\mathcal{B}$ initially defines the three sets, $\Gamma,\ \Gamma',\ S$, as follows: $\Gamma = \left\{i \mid \sigma_i = x^*_{\beta,i}\right\}$, Γ' is any set such that $\Gamma \subseteq \Gamma' \subseteq S$ and $|\Gamma'| = t-1$, and $S = \Gamma' \cup \{0\}$. Following that, $\mathcal{B}$ specifies the trapdoor K_J, where $J = (i - \Delta_J)/U + 1$. $\mathcal{B}$ selects a random t-1 degree polynomial $q(x)$ by randomly selecting its value for the t-1 points and having $q(0) = ab + \gamma$.

If $i \in \Gamma'$, $\mathcal{B}$ chooses a random $r_J, \eta_J \in \mathbb{Z}_p$, $q(J) = \tau_J$.

$$K_{1,J} = g^{q(J)}\left(h_i u_i^{\sigma_i}\right)^{r_J} w_i^{\eta_J} = (g)^{\tau_J + \theta_i + \lambda_i \eta_J}\left(g^b\right)^{r_J\left(\sigma_i - x^*_{\beta,\ i}\right)\phi_i},$$

$$K_{2,J} = Y_1^{r_J} = (g)^{y_1 r_J},$$

$$K_{3,J} = Y_1^{\eta_J} = (g)^{y_1 \eta_J},$$

$$K_{4,J} = Y_1^{-\left(r_J v_i + \eta_J t_i\right)} = (g)^{-y_1\left(r_J v_i + \eta_J t_i\right)},$$

$$K_{5,J} = V_i^{y_1 Z_J} = (g)^{y_1 v_i Z_J},$$

$$K_{6,J} = Y_1^{Z_J} = (g)^{y_1 Z_J},$$

$$K_{7,J} = \left(h_i u_i^{\sigma_i}\right)^{Z_J} = \left((g)^{\theta_i}\left(g^b\right)^{\left(\sigma_i - x^*_{\beta,i}\right)\phi_i}\right)^{Z_J}$$

If $i \in S - \Gamma'$: $\sigma_i - x^*_{\beta,i} \neq 0$, $q(J) = \Sigma_{i \in \Gamma'} q(J)\,\Delta_J + q(0)\,\Delta_0$, $r_J = \left(\tilde{r}_J - a/\left[\left(\sigma_i - x^*_{\beta,i}\right)\phi_i\right]\right)\Delta_0$.

$$K_{1,J} = g^{q(J)}\left(h_i u_i^{\sigma_i}\right)^{r_J} w_i^{\eta_J} = g^{\Sigma_{i \in \Gamma'} q(J)\Delta_J + q(0)\Delta_0}\left((g)^{\theta_i}\left(g^b\right)^{\left(\sigma_i - x^*_{\beta,i}\right)\phi_i}\right)^{r_J}\left(g^{\lambda_i}\right)^{\eta_J}$$

$$= (g)^{\Sigma_{i \in \Gamma'} q(J)\Delta_J}\, g^{(ab+\gamma)\Delta_0}\left((g)^{\theta_i}\left(g^b\right)^{\left(\sigma_i - x^*_{\beta,i}\right)\phi_i}\right)^{\left(\tilde{r}_J - a/\left[\left(\sigma_i - x^*_{\beta,i}\right)\phi_i\right]\right)\Delta_0}\left(g^{\lambda_i}\right)^{\eta_J}$$

$$= (g)^{\Sigma_{i \in \Gamma'} q(J)\Delta_J + \gamma\Delta_0}\, g^{ab\Delta_0}\,(g)^{\theta_i \tilde{r}_J \Delta_0}\left(g^a\right)^{\left(\theta_i/\left[\left(\sigma_i - x^*_{\beta,i}\right)\phi_j\right]\right)\Delta_0}\left(g^b\right)^{\left(\sigma_i - x^*_{\beta,i}\right)\phi_i \tilde{r}_J \Delta_0}\left(g^{-ab\Delta_0}\right)\left(g^{\lambda_i}\right)^{\eta_J}$$

$$= (g)^{\Sigma_{i \in \Gamma'} q(J)\Delta_J + \gamma\Delta_0 + \theta_i \tilde{r}_J \Delta_0 + \lambda_i \eta_J}\left(g^a\right)^{\left(\theta_i/\left[\left(\sigma_i - x^*_{\beta,i}\right)\phi_j\right]\right)\Delta_0}\left(g^b\right)^{\left(\sigma_i - x^*_{\beta,i}\right)\phi_i \tilde{r}_J \Delta_0},$$

$$K_{2,J} = Y_1^{r_J} = (g^{y_1})^{\left(\tilde{r}_J - a/\left[\left(\sigma_i - x^*_{\beta,i}\right)\phi_i\right]\right)\Delta_0} = (g)^{y_1 \tilde{r}_J \Delta_0} \, (g^a)^{-\left(y_1 \Delta_0\right)/\left[\left(\sigma_i - x^*_{\beta,i}\right)\phi_i\right]},$$

$$K_{3,J} = Y_1^{\eta_J} = (g)^{y_1 \eta_J},$$

$$K_{4,J} = Y_1^{-\left(r_J v_i + \eta_J t_i\right)} = (g^{y_1})^{-\left(\left(\tilde{r}_J - a/\left[\left(\sigma_i - x^*_{\beta,i}\right)\phi_i\right]\right)\Delta_0 v_i + \eta_J t_i\right)} = (g)^{-y_1\left(\tilde{r}_J \Delta_0 v_i + \eta_J t_i\right)} \, (g^a)^{\left(y_1 \Delta_0 v_i\right)/\left(\sigma_i - x^*_{\beta,i}\right)\phi_i},$$

$$K_{5,J} = V_i^{y_1 Z_J} = (g)^{y_1 v_i Z_J},$$

$$K_{6,J} = Y_1^{Z_J} = (g)^{y_1 Z_J},$$

$$K_{7,J} = \left(h_i u_i^{\sigma_i}\right)^{Z_J} = \left((g)^{\theta_i} \left(g^b\right)^{\left(\sigma_i - x^*_{\beta,i}\right)\phi_i}\right)^{Z_J} = (g)^{\theta_i Z_J} \left(g^b\right)^{\left(\sigma_i - x^*_{\beta,i}\right)\phi_i Z_J}.$$

• **Challenge.** $\mathcal{A}$ sends M_0 and M_1 to $\mathcal{B}$. If $M_0 = M_1$, $\mathcal{B}$ finishes the game and guesses $\beta' \in \{0, 1\}$. Otherwise, $\mathcal{B}$ chooses $s_1, s_2 \in \mathbb{Z}_p$ randomly and generates the challenge ciphertext CT^*, where $s_1 = c$.

$$C_1^* = Y_1^{s_1} = (g^{y_1})^c = (g^c)^{y_1},$$

$$C_2^* = g^{s_2} = (g)^{s_2},$$

$$C_{3,1}^* = \left(h_1 u_1^{x^*_{\beta,1}}\right)^{s_1} V_1^{s_2} = \left((g)^{\theta_1} \left(g^b\right)^{\left(x^*_{\beta,1} - x^*_{\beta,1}\right)\phi_1}\right)^c g^{v_1 s_2} = (g^c)^{\theta_1} \, (g^{v_1})^{s_2}, \ldots,$$

$$C_{3,i}^* = \left(h_i u_i^{x^*_{\beta,i}}\right)^{s_1} V_i^{s_2} = (g^c)^{\theta_i} \, (g)^{v_i s_2}, \ldots,$$

$$C_{3,\ell}^* = \left(h_\ell u_\ell^{x^*_{\beta,\ell}}\right)^{s_1} V_\ell^{s_2} = (g^c)^{\theta_\ell} \, (g)^{v_\ell s_2},$$

$$C_{4,1}^* = w_1^{s_1} T_1^{s_2} = \left(g^{\lambda_1}\right)^c \left(g^{t_1}\right)^{s_2} = (g^c)^{\lambda_1} \, (g)^{t_1 s_2}, \ldots,$$

$$C_{4,i}^* = w_i^{s_1} T_i^{s_2} = (g^c)^{\lambda_i} \, (g)^{t_i s_2}, \ldots,$$

$$C_{4,\ell}^* = w_\ell^{s_1} T_\ell^{s_2} = (g^c)^{\lambda_\ell} \, (g)^{t_\ell s_2},$$

$$C_5^* = e\left(g_1, Y_1\right)^{s_1} M_\beta = e\left(g^{ab+\gamma}, g^{y_1}\right)^c M_\beta = Ze\left(g^c, g\right)^{y_1 \gamma} M_\beta,$$

where $Z = e\left(g, g\right)^{abc}$.

• **Query Phase 2.** $\mathcal{A}$ carries on issuing questions that were not asked in Query Phase 1. $\mathcal{B}$ reacts the same way as previously.

• **Guess.** For the challenge ciphertext, $\mathcal{A}$ provides a guess $\beta' \in \{0, 1\}$. If $\beta' = \beta$, $\mathcal{B}$ returns 1, else $\mathcal{B}$ returns 0.

$\mathcal{B}$'s advantage in solving the DBDH problem is related to $\mathcal{A}$'s advantage of distinguishing Game_0 and Game_1. Allow $\mathrm{Game}_1 = \mathrm{Game}_{2,0}$, and the following lemma is true for $j = 0, 1, \ldots, |A| - 1$.

Lemma 2. Suppose that the ADLIN assumption is true. In other words, the difference in the advantage of $\mathcal{A}$ in $\mathrm{Game}_{2,\mathrm{j}}$ and $\mathrm{Game}_{2,\mathrm{j}+1}$ is negligible for every polynomial time adversary $\mathcal{A}$.

Proof. Consider $\mathcal{A}$ with a non-negligible difference ε about its advantage between $\mathrm{Game}_{2,\mathrm{j}}$ and $\mathrm{Game}_{2,\mathrm{j}+1}$. We hope to create an algorithm in which $\mathcal{B}$ employs $\mathcal{A}$'s ability to work out the ADLIN issue in $\mathbb{G}$. Given $\left(g, g^{z_1}, g^{z_2}, g^{z_2^2}, g^{z_2/z_1}, g^{z_2^2 z_3}, g^{z_4}, Z\right) \in \mathbb{G}^8$, $\mathcal{B}$ outputs 1 when $Z = g^{z_1\left(z_3+z_4\right)}$, otherwise $\mathcal{B}$ generates 0. $\mathcal{B}$ communicates with $\mathcal{A}$ using the subsequent process:

• **Init.** At the start of the game, $\mathcal{A}$ commits two vectors $\vec{X}_0^*, \vec{X}_1^*$ converted to $\vec{x}_0^*, \vec{x}_1^* \in \Sigma^\ell$. $\mathcal{B}$ internally throws a coin $\beta \in \{0, 1\}$.

• **Setup.** δ denotes the $(j+1)$-th index in A. $\mathcal{B}$ randomly selects $\gamma, y_1, v_1, \ldots, v_\ell, t_1, \ldots, t_\ell, \theta_1, \ldots, \theta_\ell, \phi_1, \ldots, \phi_\ell, \lambda_1, \ldots, \lambda_\ell$, then places $g_1 = g^\gamma$, $Y_1 = \left(g^{z_2^2}\right)^{y_1}$, and $\Omega = e(g_1, Y_1)$. Take note that $\tilde{y}_1 = y_1 z_2^2$. Following that, $\mathcal{B}$ sets $h_\delta = (g^{z_1})^{\theta_\delta} (g^{z_2})^{-\phi_\delta \cdot x^*_{\beta,\delta}}$, $u_\delta = (g^{z_2})^{\phi_\delta}$, $w_\delta = (g^{z_1})^{\lambda_\delta}$, $V_\delta = (g^{z_1})^{\theta_\delta} g^{v_\delta}$, $T_\delta = (g^{z_1})^{\lambda_\delta} g^{t_\delta}$. For all $i \neq \delta$, $\mathcal{B}$ sets $h_i = \left(g^{z_2^2}\right)^{\theta_i} (g^{z_2})^{-\phi_i \cdot x^*_{\beta,i}}$, $u_i = (g^{z_2})^{\phi_i}$, $w_i = \left(g^{z_2^2}\right)^{\lambda_i}$, $V_i = g^{v_i}$, and $T_i = g^{t_i}$. $\mathcal{B}$ sends the public key $\mathrm{PK} = (g, Y_1, (h_1, u_1, w_1, V_1, T_1), \ldots, (h_\ell, u_\ell, w_\ell, V_\ell, T_\ell), \Omega)$ to $\mathcal{A}$. Because g and all exponents are chosen at random, the public key PK has the same distribution as the original construction.

• **Query Phase 1.** $\mathcal{A}$ offers trapdoor queries, where a trapdoor vector $\vec{\Delta}$ from $\mathcal{A}$ is changed to $\vec{\sigma} = (\sigma_1, \ldots, \sigma_\ell) \in (\Sigma_*)^\ell$ and S is the set of indexes i such that $\sigma_i \neq *$. We consider the two types of trapdoor queries stated previously.

Type 1. This describes the situation with $\delta \notin S$. $\mathcal{B}$ then selects at random $r_1, \ldots, r_J, \eta_1, \ldots, \eta_J, Z_1, \ldots, Z_J \in Z_J$ for all $i \in S$, and makes $r_J = \tilde{r}_J, \eta_J = \tilde{\eta}_J, Z_J = \tilde{Z}_J$. Next, $\mathcal{B}$ calculates $i \notin S, q(J) = \gamma$.

$$K_{1,J} = g^{q(J)} \left(h_i u_i^{\sigma_i}\right)^{r_J} w_i^{\eta_J} = g^\gamma \left(\left(g^{z_2^2}\right)^{\theta_i} (g^{z_2})^{\left(\sigma_i - x^*_{\beta,i}\right)\phi_i}\right)^{\tilde{r}_J} \left(g^{z_2^2}\right)^{\lambda_i \tilde{\eta}_J} = g^\gamma \left(g^{z_2^2}\right)^{\theta_i \tilde{r}_J + \lambda_i \tilde{\eta}_J} (g^{z_2})^{\left(\sigma_i - x^*_{\beta,i}\right)\phi_i \tilde{r}_J},$$

$$K_{2,J} = Y_1^{r_J} = \left(g^{z_2^2}\right)^{y_1 \tilde{r}_J},$$

$$K_{3,J} = Y_1^{\eta_J} = \left(g^{z_2^2}\right)^{y_1 \tilde{\eta}_J},$$

$$K_{4,J} = Y_1^{-\left(r_J v_i + \eta_J t_i\right)} = \left(g^{z_2^2}\right)^{-y_1\left(r_J v_i + \eta_J t_i\right)},$$

$$K_{5,J} = V_i^{y_1 Z_J} = (g)^{y_1 v_i \tilde{Z}_J},$$

$$K_{6,J} = Y_1 Z_J = \left(g^{z_2^2}\right)^{y_1 \tilde{Z}_J},$$

$$K_{7,J} = \left(h_i u_i^{\sigma_i}\right)^{Z_J} = \left(\left(g^{z_2^2}\right)^{\theta_i \tilde{Z}_J} (g^{z_2})^{\left(\sigma_i - x^*_{\beta,i}\right)\phi_i \tilde{Z}_J}\right).$$

Type 2. Case 1. Assume $\delta \in S$ and $\sigma_\delta \neq x^*_{\beta,\delta}$. Then, for each $i (\neq \delta) \in S$, $\mathcal{B}$ chooses random $r_1, \ldots, r_J, \eta_1, \ldots, \eta_J, Z_1, \ldots, Z_J \in \mathbb{Z}_p$. $K_{1,J}, \ldots, K_{7,J}$ are identical as those in Type 1. When $i = \delta$, $\mathcal{B}$ then calculates $q(J) = \gamma$, $r_J = \left(\tilde{r}_J / z_2 + \tilde{\eta}_J / z_2^2\right)\lambda_\delta$, $\eta_J = -\left(\tilde{r}_J / z_2 + \tilde{\eta}_J / z_2^2\right)\theta_\delta - [\phi_\delta \tilde{\eta}_J \left(\sigma_\delta - x^*_{\beta,\delta}\right)]/(z_1 z_2)$, and $Z_J = \tilde{Z}_J$.

$$K_{1,J} = g^{q(J)} \left(h_\delta u_\delta^{\sigma_\delta}\right)^{r_J} w_\delta^{\eta_J} = g^\gamma \left((g^{z_1})^{\theta_\delta} (g^{z_2})^{\left(\sigma_\delta - x^*_{\beta,\delta}\right)\phi_\delta}\right)^{\left(\tilde{r}_J / z_2 + \tilde{\eta}_J / z_2^2\right)\lambda_\delta} \left(g^{z_1 \lambda_\delta}\right)^{-\left(\tilde{r}_J / z_2 + \tilde{\eta}_J / z_2^2\right)\theta_\delta - \left[\phi_\delta \tilde{\eta}_J \left(\sigma_\delta - x^*_{\beta,\delta}\right)\right]/z_1 z_2}$$

$$= g^\gamma (g)^{\left(\sigma_\delta - x^*_{\beta,\delta}\right)\phi_\delta \lambda_\delta \tilde{r}_J},$$

$$K_{2,J} = Y_1^{r_J} = \left(g^{z_2^2 y_1}\right)^{\left(\tilde{r}_J / z_2 + \tilde{\eta}_J / z_2^2\right)\lambda_\delta} = (g^{z_2})^{y_1 \tilde{r}_J \lambda_\delta} (g)^{y_1 \tilde{\eta}_J \lambda_\delta},$$

$$K_{3,J} = Y_1^{\eta_J} = \left(g^{z_2^2 y_1}\right)^{-\left(\tilde{r}_J / z_2 + \tilde{\eta}_J / z_2^2\right)\theta_\delta - \left[\phi_\delta \tilde{\eta}_J \left(\sigma_\delta - x^*_{\beta,\delta}\right)\right]/(z_1 z_2)} = (g^{z_2})^{-y_1 \tilde{r}_J \theta_\delta} (g)^{-y_1 \tilde{\eta}_J \theta_\delta} \left(g^{z_2 / z_1}\right)^{-y_1 \phi_\delta \tilde{\eta}_J \left(\sigma_\delta - x^*_{\beta,\delta}\right)},$$

$$K_{4,J} = Y_1^{-\left(r_J v_\delta + \eta_J t_\delta\right)} = \left(g^{z_2^2 y_1}\right)^{-\left(\left(\tilde{r}_J / z_2 + \tilde{\eta}_J / z_2^2\right)\lambda_\delta v_\delta + \left[-\left(\tilde{r}_J / z_2 + \tilde{\eta}_J / z_2^2\right)\theta_\delta - \left[\phi_\delta \tilde{\eta}_J \left(\sigma_\delta - x^*_{\beta,\delta}\right)\right]/\left(z_1 z_2\right)\right] t_\delta\right)}$$

$$= (g^{z_2})^{-y_1 \tilde{r}_J \left(v_\delta \lambda_\delta - t_\delta \theta_\delta\right)} (g)^{-y_1 \tilde{\eta}_J \left(\lambda_\delta v_\delta - t_\delta \theta_\delta\right)} \left(g^{z_2 / z_1}\right)^{y_1 \tilde{\eta}_J \phi_\delta t_\delta \left(\sigma_\delta - x^*_{\beta,\delta}\right)},$$

$$K_{5,J} = V_{\delta}^{y_1 Z_J} = (g)^{y_1 v_\delta \tilde{Z}_J},$$
$$K_{6,J} = Y_1^{Z_J} = \left(g^{z_2^2}\right)^{y_1 \tilde{Z}_J},$$
$$K_{7,J} = \left(h_\delta u_\delta^{\sigma_i}\right)^{Z_J} = (g^{z_1})^{\theta_\delta \tilde{Z}_J} (g^{z_2})^{\left(\sigma_\delta - x^*_{\beta,\delta}\right)\phi_\delta \tilde{Z}_J}.$$

Case 2. Assume $\delta \in S$ and $\sigma_\delta = x^*_{\beta,\delta}$, yet there exists an index $j \in S$ such that $\sigma_j \neq x^*_{\beta,j}$. As in Case 1, $\mathcal{B}$ chooses at random $r_1, \ldots, r_J, \eta_1, \ldots, \eta_J, Z_1, \ldots, Z_J \in \mathbb{Z}_p$ for each $i\,(\neq \delta) \in S$. $K_{1,J}, \ldots, K_{7,J}$ are identical to those in Type 1. Otherwise (i.e., $i\,(=\delta) \in S$), $\mathcal{B}$ generates $q(J) = \gamma$, $r_J = (\tilde{r}_J/z_2^2 + \tilde{\eta}_J/z_2)\lambda_\delta$, $\eta_J = -\left(\tilde{r}_J/z_2^2 + \tilde{\eta}_J/z_2\right)\theta_\delta - \phi_j \tilde{\eta}_J \left(\sigma_j - x^*_{\beta,j}\right)$, $Z_J = \tilde{Z}_J$.

$$K_{1,J} = g^{q(J)} \left(h_\delta u_\delta^{\sigma_\delta}\right)^{r_J} w_\delta^{\eta_J} = g^{\gamma} \left((g^{z_1})^{\theta_\delta} (g^{z_2})^{\left(\sigma_\delta - x^*_{\beta,\delta}\right)\phi_\delta}\right)^{r_J} \left(g^{z_1 \lambda_\delta}\right)^{\eta_J}$$
$$= g^{\gamma} \left((g^{z_1})^{\theta_\delta} (g^{z_2})^{\left(\sigma_\delta - x^*_{\beta,\delta}\right)\phi_\delta}\right)^{\left(\tilde{r}_J/z_2^2 + \tilde{\eta}_J/z_2\right)\lambda_\delta} \left(g^{z_1\lambda_\delta}\right)^{-\left(\tilde{r}_J/z_2^2 + \tilde{\eta}_J/z_2\right)\theta_\delta - \phi_j \tilde{\eta}_J \left(\sigma_j - x^*_{\beta,j}\right)}$$
$$= g^{\gamma} (g^{z_1})^{-\lambda_\delta \phi_j \tilde{\eta}_J \left(\sigma_j - x^*_{\beta,j}\right)},$$
$$K_{2,J} = Y_1^{r_J} = \left(g^{z_2^2 y_1}\right)^{\left(\tilde{r}_J/z_2^2 + \tilde{\eta}_J/z_2\right)\lambda_\delta} = (g)^{y_1 \tilde{r}_J \lambda_\delta} (g^{z_2})^{y_1 \tilde{\eta}_J \lambda_\delta},$$
$$K_{3,J} = Y_1^{\eta_J} = \left(g^{z_2^2 y_1}\right)^{-\left(\tilde{r}_J/z_2^2 + \tilde{\eta}_J/z_2\right)\theta_\delta - \phi_j \tilde{\eta}_J \left(\sigma_j - x^*_{\beta,j}\right)} = (g)^{-y_1 \tilde{r}_J \theta_\delta} (g^{z_2})^{-y_1 \tilde{\eta}_J \theta_\delta} \left(g^{z_2^2}\right)^{-y_1 \phi_j \tilde{\eta}_J \left(\sigma_j - x^*_{\beta,j}\right)},$$
$$K_{4,J} = Y_1^{-\left(r_J v_\delta + \eta_J t_\delta\right)} = \left(g^{z_2^2 y_1}\right)^{-\left(\left(\tilde{r}_J/z_2^2 + \tilde{\eta}_J/z_2\right)\lambda_\delta v_\delta + \left[-\left(\tilde{r}_J/z_2^2 + \tilde{\eta}_J/z_2\right)\theta_\delta - \phi_j \tilde{\eta}_J \left(\sigma_j - x^*_{\beta,j}\right)\right] t_\delta\right)}$$
$$= (g)^{-y_1 \tilde{r}_J \left(\lambda_\delta v_\delta - t_\delta \theta_\delta\right)} (g^{z_2})^{-y_1 \tilde{\eta}_J \left(\lambda_\delta v_\delta - t_\delta \theta_\delta\right)} \left(g^{z_2^2}\right)^{y_1 t_\delta \phi_j \tilde{\eta}_J \left(\sigma_j - x^*_{\beta,j}\right)},$$
$$K_{5,J} = V_{\delta}^{y_1 Z_J} = (g)^{y_1 v_\delta Z_J},$$
$$K_{6,J} = Y_1^{Z_J} = \left(g^{z_2^2}\right)^{y_1 \tilde{Z}_J},$$
$$K_{7,J} = \left(h_i u_i^{\sigma_i}\right)^{Z_J} = (g^{z_1})^{\theta_\delta} \tilde{Z}_J (g^{z_2})^{\left(\sigma_\delta - x^*_{\beta,\delta}\right)\phi_\delta \tilde{Z}_J}.$$

We verify that it is a well-formed trapdoor via random exponents $i \in S$ (including δ). Based on the exponents above, the subsequent formulas hold for every $i \in S$ as in the actual construction.

• **Challenge.** $\mathcal{A}$ sends $\mathcal{B}$ with M_0 and M_1. $\mathcal{B}$ chooses random $R_6 \in \mathbb{G}_T$, $R_{3,1}, \ldots, R_{3,\delta-1}$, $R_{4,1}, \ldots, R_{4,\delta-1} \in \mathbb{G}$. If $s_1 = z_3$, $s_2 = z_4$, $\mathcal{B}$ gives the challenge ciphertext CT^* as

$$C_1^* = Y_1^{s_1} = \left(g^{z_2^2}\right)^{y_1 z_3} = \left(g^{z_2^2 z_3}\right)^{y_1},$$
$$C_2^* = g^{s_2} = g^{z_4},$$
$$C_{3,1}^* = R_{3,1}, \cdots,$$
$$C_{3,\delta-1}^* = R_{3,\delta-1},$$

$$C_{3,\delta}^* = (h_\delta u_\delta^{x^*_{\beta,\delta}})^{s_1} V_\delta^{s_2} = ((g^{z_1})^{\theta_\delta} (g^{z_2})^{(x^*_{\beta,i} - x^*_{\beta,\delta})\phi_\delta})^{z_3} ((g^{z_1})^{\theta_\delta} (g^{v_\delta}))^{z_4} = (g^{z_1(z_3+z_4)})^{\theta_\delta} (g^{z_4})^{v_\delta} = (Z)^{\theta_\delta} (g^{z_4})^{v_\delta},$$

$$C^*_{3,\delta+1} = \left(h_{\delta+1} u^{x^*_{\beta,\delta+1}}_{\delta+1}\right)^{s_1} V^{s_2}_{\delta+1} = \left(\left(g^{z_2^2}\right)^{\theta_{\delta+1} z_3} \left(g^{z_2}\right)^{\left(x^*_{\beta,\delta+1} - x^*_{\beta,\delta+1}\right)\phi_{\delta+1} z_3}\right) \left(g^{v_{\delta+1} z_4}\right)$$

$$= \left(g^{z_2^2 z_3}\right)^{\theta_{\delta+1}} \left(g^{z_4}\right)^{v_{\delta+1}}, \cdots,$$

$$C^*_{3,\ell} = \left(h_\ell u^{x^*_{\beta,\ell}}_{\ell}\right)^{s_1} V^{s_2}_{\ell} = \left(g^{z_2^2 z_3}\right)^{\theta_\ell} \left(g^{z_4}\right)^{v_\ell},$$

$$C^*_{4,1} = R_{4,1}, \cdots,$$

$$C^*_{4,\delta-1} = R_{4,\delta-1},$$

$$C^*_{4,\delta} = w^{s_1}_{\delta} T^{s_2}_{\delta} = \left(g^{z_1}\right)^{\lambda_\delta z_3} \left(g^{z_1 \lambda_\delta} g^{t_\delta}\right)^{z_4} = \left(g^{z_1 \left(z_3 + z_4\right)}\right)^{\lambda_\delta} \left(g^{z_4}\right)^{t_\delta} = (Z)^{\lambda_\delta} \left(g^{z_4}\right)^{t_\delta},$$

$$C^*_{4,\delta+1} = w^{s_1}_{\delta+1} T^{s_2}_{\delta+1} = \left(g^{z_2^2}\right)^{\lambda_{\delta+1} z_3} \left(g^{t_{\delta+1}}\right)^{z_4} = \left(g^{z_2^2 z_3}\right)^{\lambda_{\delta+1}} \left(g^{z_4}\right)^{t_{\delta+1}}, \cdots,$$

$$C^*_{4,\ell} = w^{s_1}_{\ell} T^{s_2}_{\ell} = \left(g^{z_2^2 z_3}\right)^{\lambda_\ell} \left(g^{z_4}\right)^{t_\ell},$$

$$C^*_5 = e\left(g_1, Y_1\right)^{s_1} M_\beta = R_6,$$

where $s_1 = c$ and $Z = e\left(g, g\right)^{abc}$.

If $M_0 = M_1$, $\mathcal{B}$ substitutes R_6 with $e\left(g, g^{z_2^2 z_3}\right)^{\gamma y_1} M_0$; otherwise, it follows the preceding procedure. If $Z = g^{z_1\left(z_3+z_4\right)}$, we can deduce that $Z^{\theta_\delta} \left(g^{z_4}\right)^{v_\delta} = \left(g^{z_1\left(z_3+z_4\right)}\right)^{\theta_\delta} \left(g^{z_4}\right)^{v_\delta} = \left(\left(g^{z_1}\right)^{\theta_\delta} \left(g^{z_2}\right)^{-\phi_\delta x^*_{\beta,\delta}} \left(g^{z_2}\right)^{\phi_\delta x^*_{\beta,\delta}}\right)^{z_3}$ $\left(\left(g^{z_1}\right)^{\theta_\delta} g^{v_\delta}\right)^{z_4} = \left(h_\delta u^{x^*_{\beta,\delta}}_{\delta}\right)^{s_1} V^{s_2}_{\delta}$. In this case, $\mathcal{B}$ plays $\text{Game}_{2,\text{j}}$. Otherwise, Z is chosen at random and $\mathcal{B}$ performs $\text{Game}_{2,\text{j}+1}$.

• **Query Phase 2.** $\mathcal{A}$ keeps asking queries that were not asked in Query Phase 1. $\mathcal{B}$ responds in the same manner as before.

• **Guess.** $\mathcal{A}$ returns a guess $\beta' \in \{0, 1\}$ to the challenge ciphertext. If $\beta' = \beta$, $\mathcal{B}$ offers 1, else 0.

If $\mathcal{A}$ guesses properly, $\mathcal{B}$ also guesses correctly, implying that $Z = g^{z_1\left(z_3+z_4\right)}$ holds in the ADLIN problem. In addition, $\mathcal{B}$ considers that $Z \neq g^{z_1\left(z_3+z_4\right)}$. Consequently, any advantage obtained by $\mathcal{A}$ in distinguishing between $\text{Game}_{2,\text{j}}$ and $\text{Game}_{2,\text{j}+1}$ is transferred to $\mathcal{B}$'s advantage when dealing with the ADLIN problem.

6 Performance

6.1 Theoretical Comparison

To analyze the vector dominance threshold problem in a public key system, we compare the proposed scheme with the schemes in [5–8] and [14–17] and show the results in Table 1. In [5–8], the authors create searchable public-key systems that support comparison. Unfortunately, these schemes cannot conduct threshold comparisons, which means some of their search results are not flexible. Although the schemes in [14–17] can perform it, the schemes in [15–17] are not attribute-hiding. Besides, the scheme in [14] is not efficient. Our scheme requires a ciphertext size of $O\left(wn\right)$ and merely a trapdoor size of $O\left(w\right)$ when any threshold comparison queries are constructed. Overall, our scheme performs better than the other SE methods in terms of computational efficiency and resource utilization.

Table 1: Storage overhead

Scheme	System	CT size	Trapdoor size	Threshold comparison	Attribute-hiding	Standard model
[5]	SE	$O(wn)$	$O(w)$	×	√	√
[6]	SE	$O(w \log n)$	$O(w \log n)$	×	√	√
[7]	SE	$O(wn)$	$O(1)$	×	√	√
[8]	SE	$O(wn)$	$O(1)$	×	√	√
[14]	SE	$O(n^w)$	$O(n^w)$	√	√	√
[15]	ABE	$O(w \log n)$	$O(w \log n)$	√	×	×
[16]	ABE	$O(w \log n)$	$O(w \log n)$	√	×	×
[17]	ABE	$O(w \log n)$	$O(w \log n)$	√	×	√
Ours	SE	$O(wn)$	$O(w)$	√	√	√

Note: w is the number of query keywords; n is the universal set (the length of the maximum vector).

6.2 Computation Overhead

Since the scheme in [14] deals with the same problem and system as our scheme, we make a further comparison between the two in Table 2. The computation tasks involve pairing and exponentiation operations where pairing operations cost the most time.

Table 2: Computation overhead

Scheme	PK size	CT size	Trapdoor size	Decryption cost
[14]	$O(n^w)$	$(n^w + 3)\,\mathbb{G} + \mathbb{G}_T$	$(n^w + 2)\,\mathbb{G}$	$(n^w + 4)\,p1$
Ours	$O(wn)$	$(2nw)\,\mathbb{G} + \mathbb{G}_T$	$7w\mathbb{G}$	$(w)\,p1 + (w)\,e$

Note: w is the number of query keywords; $\mathbb{G}$, $\mathbb{G}_T$ is the measured length for each element of $\mathbb{G}$, $\mathbb{G}_T$; n is the universal set; $p1$ and e are the pairing and exponentiation of $\mathbb{G}$, respectively.

For [14], its PK size expands exponentially with w, while the PK size of our scheme only increases linearly. The ciphertext size (CT size) of [14] requires $(n^w + 3)\,\mathbb{G}+\mathbb{G}_T$ actions, and ours is $(2nw)\,\mathbb{G}+\mathbb{G}_T$. Furthermore, to compute a trapdoor, reference [14] requires $(n^w + 2)\,\mathbb{G}$ group operations, whereas our scheme requires just $7w\mathbb{G}$ group operations. Finally, reference [14] requires $(n^w + 4)\,p1$ operations and our scheme needs $(w)\,p1 + (w)\,e$. Because $p1$ requires more processing resources than e in general, our technique outperforms the scheme in [14]. Overall, it is clear that our system outperforms the scheme of [14] in terms of computation efficiency.

6.3 Storage Overhead

To compute the storage overhead, we set $|\mathbb{G}| = 1024$ and $|Z_p| = 160$ in the simulation. We assume that the number of our query keywords ranges from 0 to 50, then count the storage overhead of the parameters in the algorithm. Fig. 2 compares the results between our scheme and [14]. According to Fig. 2a, our PK size is smaller than that of [14]. In Fig. 2b, the CT size of our scheme grows linearly with the number of query keywords while the CT size of the scheme in [14] increases exponentially and is larger than ours. From Fig. 2c, the storage overheads of our method and the scheme in [14] also

grow linearly and exponentially, respectively. The simulation results show that the overheads of the proposed scheme are much smaller than those of [14], indicating higher efficiency of our scheme.

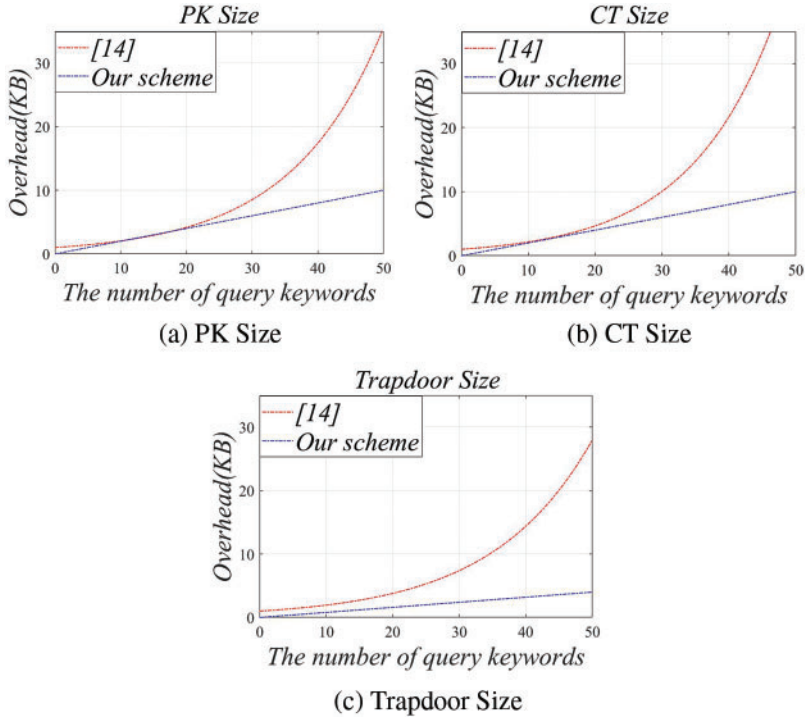

Figure 2: Storage overhead of each parameter as a function of the number of query keywords

6.4 Experimental Evaluation

We implement the algorithm with C language employing the GNU Multiple Precision Arithmetic (GMP) and Pairing-Based Cryptographic (PBC) libraries. Furthermore, this experiment makes use of the Pima Indians Diabetes Dataset at https://www.kaggle.com/uciml/pima-indians-diabetes-database.

The execution time of the scheme of [14] and our scheme for the KeyGen, PEKS, Trapdoor, and Test algorithms is displayed in Figs. 3a–3d, respectively. The universal set is set to 100. According to [14], as the number of query keywords grows from 0 to 100, the cost time required for KeyGen creation rises from 0.024 to 50 s and the cost time for PEKS generation increases from 0.37 to 300 s. Considering that the generation time of the Trapdoor and Test algorithms is excessive, we change the number of query keywords to 0–50. In [14], the cost time required for trapdoor generation climbs from 0.24 to 500 ms, while the test generation time rises from 0.37 to 600 ms. Our algorithm, however, takes almost no time. The execution time for KeyGen represents the time for PK and SK to be generated, and the PEKS time indicates the ciphertext generation time. The Trapdoor and Test algorithms involve the time of trapdoor generation and the time of cloud server search, respectively.

It can be seen from Fig. 3 that the runtime of our algorithm increases slowly with the increase of query keywords, while the runtime of [14] increases exponentially. The experiment shows that our scheme is more efficient. In addition, our scheme uses real data sets and is carried out on a cloud outsourcing platform, therefore it will be feasible and effective in real-world scenarios.

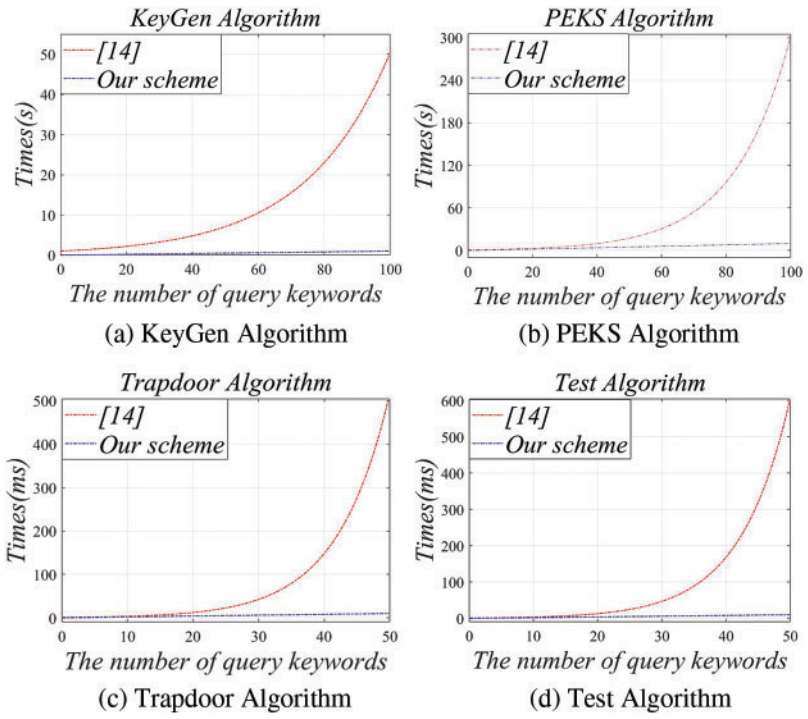

Figure 3: Time cost of each algorithm as a function of the number of query keywords

In addition to the number of query keywords, the varying size of encrypted diabetes data also has a great effect on algorithm performance. As the encrypted data increases, the service provider will continuously run the VDTSE scheme to search until it finds all matched ciphertexts. Therefore, the performance of VDTSE degrades linearly with the increase of encrypted data. Note that although the threshold t has a range, it takes a fixed value every time the algorithm runs so it has no impact on the performance of the scheme.

7 Conclusion

In theoretical comparison to existing schemes, the proposed VDTSE scheme obtains a shorter trapdoor that makes it more suitable for storage on mobile devices. It supports comparable attributes that can work for the (t, n)-threshold policy. Then, its security, flexibility, and effectiveness are proved through comparison with other SE schemes.

However, there are also some limitations in this research. Although our scheme is more efficient than other existing SE schemes, it does not work well on large real-world datasets. The larger data is transformed into a longer vector, lowering the efficiency. Besides, when dealing with floating data, the scheme converts the floating data to integer data through multiple expansions. For example, 0.1 is expanded 10 times and converted to 1. In addition, the Lagrangian polynomial technique is not efficient in dealing with the vector dominance threshold problem, but it is currently the most suitable technique. We will look for a better technique to solve these issues.

More fascinating, this work inspires some excellent open problems. Firstly, our research does not address video encryption [18,19], which is an engaging research direction. In the future, we will apply SE algorithms to more scenarios, such as images [20] and other situations [21–23]. Secondly, it will be

an intriguing path to demonstrate how to reduce the ciphertext size, which appears difficult to achieve at the moment.

Acknowledgement: Not applicable.

Funding Statement: This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 61872289 and 62172266, and in part by the Henan Key Laboratory of Network Cryptography Technology LNCT2020-A07 and the Guangxi Key Laboratory of Trusted Software under Grant No. KX202308.

Author Contributions: The authors confirm that the contributions to the paper are as follows: Study conception and design: Jingjing Nie, Zhenhua Chen; draft manuscript preparation: Jingjing Nie. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are openly available at https://www.kaggle.com/uciml/pima-indians-diabetes-database.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Z. Zhou *et al.*, "Blockchain-based secure and efficient secret image sharing with outsourcing computation in wireless networks," *IEEE Trans. Wirel. Commun.*, vol. 23, no. 1, pp. 423–435, Jan. 2024. doi: 10.1109/TWC.2023.3278108.

[2] Z. Shen, F. Ding, Y. Yao, A. Bhardwaj, Z. Guo and K. Yu, "A privacy-preserving social computing framework for health management using federated learning," *IEEE Trans. Comput. Soc. Syst.*, vol. 10, no. 4, pp. 1666–1678, Aug. 2023. doi: 10.1109/TCSS.2022.3222682.

[3] Z. Shen, F. Ding, A. Jolfaei, K. Yadav, S. Vashisht and K. Yu, "DeformableGAN: Generating medical images with improved integrity for healthcare cyber physical systems," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 5, pp. 2584–2596, Jul. 2022. doi: 10.1109/TNSE.2022.3190765.

[4] M. J. Atallah and W. Du, "Secure multi-party computational geometry," in *Proc. 7th Int. Workshop Algorithms Data Struct. (WADS)*, Providence, RI, USA, Aug. 2001, pp. 165–179.

[5] D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data," in *Proc. 4th Theory Cryptogr. Conf. (TCC)*, Amsterdam, The Netherlands, Feb. 2007, pp. 535–554.

[6] E. Shi, J. Bethencourt, T. H. Chan, D. Song, and A. Perrig, "Multi-dimensional range query over encrypted data," in *Proc. IEEE Symp. Secur. Priv. (SSP)*, Berkeley, CA, USA, May 2007, pp. 350–364.

[7] H. Park, "Efficient hidden vector encryption for conjunctive queries on encrypted data," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 10, pp. 1483–1497, Oct. 2011. doi: 10.1109/TKDE.2010.206.

[8] J. H. Park, K. Lee, W. Susilo, and D. H. Lee, "Fully secure hidden vector encryption under standard assumptions," *Inf. Sci.*, vol. 232, pp. 188–207, May 2013. doi: 10.1016/j.ins.2012.12.034.

[9] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proc. IEEE Symp. Secur. Priv. (SSP)*, Berkeley, CA, USA, May 2000, pp. 44–55.

[10] Y. Zheng, R. Lu, J. Shao, F. Yin, and H. Zhu, "Achieving practical symmetric searchable encryption with search pattern privacy over cloud," *IEEE Trans. Serv. Comput.*, vol. 15, no. 3, pp. 1358–1370, May 2020. doi: 10.1109/TSC.2020.2992303.

[11] Z. Gui, K. G. Paterson, and S. Patranabis, "Rethinking searchable symmetric encryption," in *Proc. IEEE Symp. Secur. Priv. (SSP)*, San Francisco, CA, USA, May 2023, pp. 1401–1418.

[12] E. Molla, P. Rizomiliotis, and S. Gritzalis, "Efficient searchable symmetric encryption supporting range queries," *Int. J. Inf. Secur.*, vol. 22, no. 4, pp. 785–798, Feb. 2023. doi: 10.1007/s10207-023-00667-1.

[13] D. Boneh, D. G. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Proc. Int. Conf. Theory Appl. Cryptogr. Tech.*, Interlaken, Switzerland, May 2004, pp. 506–522.
[14] D. Sun, C. Boyd, and J. M. G. Nieto, "Predicate encryption for multi-inner-products," *Secur. Commun. Netw.*, vol. 6, no. 3, pp. 325–339, Mar. 2013. doi: 10.1002/sec.566.
[15] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext–policy attribute-based encryption," in *Proc. IEEE Symp. Secur. Priv. (SSP)*, Berkeley, CA, USA, May 2007, pp. 321–334.
[16] N. Attrapadung, G. Hanaoka, K. Ogawa, G. Ohtake, H. Watanabe and S. Yamada, "Attribute based encryption for range attributes," *IEICE Trans. Fundam. Electron., Commun. Comput.*, vol. 101, no. 9, pp. 1440–1455, Sep. 2018. doi: 10.1587/transfun.E101.A.1440.
[17] K. Xue, J. Hong, Y. Xue, D. S. Wei, N. Yu and P. Hong, "CABE: A new comparable attribute-based encryption construction with 0-encoding and 1-encoding," *IEEE Trans. Comput.*, vol. 66, no. 9, pp. 1491–1503, Sep. 2017. doi: 10.1109/TC.2017.2693265.
[18] H. Li, Z. Gu, L. Deng, Y. Han, C. Yang and Z. Tian, "A fine-grained video encryption service based on the cloud-fog-local architecture for public and private videos," *Sensors*, vol. 19, no. 24, pp. 5366, Dec. 2019. doi: 10.3390/s19245366.
[19] K. M. Hosny, M. A. Zaki, N. A. Lashin, and H. M. Hamza, "Fast colored video encryption using block scrambling and multi-key generation," *Vis. Comput.*, vol. 39, no. 12, pp. 6041–6072, Dec. 2023. doi: 10.1007/s00371-022-02711-y.
[20] M. Liu, J. Fan, and Q. Liu, "Biomedical image segmentation algorithm based on dense atrous convolution," *Math. Biosci. Eng.*, vol. 21, no. 3, pp. 4351–4369, Feb. 2024. doi: 10.3934/mbe.2024192.
[21] H. A. Li, L. Wang, and J. Liu, "Application of multi-level adaptive neural network based on optimization algorithm in image style transfer," *Multimed. Tools Appl.*, vol. 23, pp. 1–23, Feb. 2024. doi: 10.1007/s11042-024-18451-1.
[22] Y. Ju *et al.*, "Reliability-security tradeoff analysis in mmWave ad hoc-based CPS," *ACM Trans. Sens. Netw.*, vol. 20, no. 2, pp. 1–23, Jan. 2024. doi: 10.1145/3582556.
[23] Z. Zhou *et al.*, "Generative steganography via auto-generation of semantic object contours," *IEEE Trans. Inf. Forensics Secur.*, vol. 18, pp. 2751–2765, Apr. 2023. doi: 10.1109/TIFS.2023.3268843.