2011

Lightweight Delegated Subset Test with Privacy Protection

Xuhua ZHOU

Xuhua DING

Singapore Management University, xhding@smu.edu.sg

Kefei CHEN

DOI: https://doi.org/10.1007/978-3-642-21031-0_11

Lightweight Delegated Subset Test with Privacy Protection

Xuhua Zhou[1], Xuhua Ding[2], and Kefei Chen[1]

[1] Shanghai Jiao Tong University, China
xuhuazhou2010@gmail.com kfchen@sjtu.edu.cn
[2] Singapore Management University, Singapore
xhding@smu.edu.sg

Abstract. Delegated subset tests are mandatory in many applications, such as content-based networks and outsourced text retrieval, where an untrusted server evaluates the degree of matching between two data sets. We design a novel scheme to protect the privacy of the data sets in comparison against the untrusted server, with half of the computation cost and half of the ciphertext size of existing solutions based on predicate-only encryption supporting inner product.

Keywords: Private Subset Test, Predicate Encryption, Predicate Privacy

1 Introduction

Subset test, including set membership test, is a primitive computation widely used in various applications involving data matching. For instance, in a publish-subscribe network, an event is channeled to a subscriber only when the keywords of the event is a subset of the subscriber's interest filter. In keyword-based text searching, a search query is a set of keywords and a document is qualified for the query if its keyword set is a superset of the query.

The aforementioned applications are characterized by the third-party subset test. Namely, the subset test is not performed by the data owner or endusers. Instead, the data owner liaises with a service provider, e.g. a publish-subscribe broker or a search engine, to carry out the evaluation. We call this *delegated subset test* where the service provider is the delegator chosen by the data owner, so as to differentiate it from private set operations [11, 16, 9, 8, 7] which is an interactive algorithm between two participants. Privacy is the main concern in delegated subset tests as the delegator is *not* trusted. The data in computation exposes the data owner/user's privacy to the delegator. For instance, the keywords in the query set leak information to the search engine about the query issuer's interests.

In essence, a subset test for set A and B is to evaluate the predicate $A \subseteq B$. Therefore, it is a special case of predicate encryption [14] with A, B being represented by two binary vectors and the test being implemented by a vector inner product. However, Shen, Shi and Waters [18] have shown that predicate

encryption in the public key setting cannot protect predicate privacy. As a result, they propose a symmetric-key predicate encryption scheme supporting inner product computations with predicate privacy protection. When being applied for delegated subset tests, their scheme requires $2n + 2$ group elements to represent an encrypted vector and $2n + 2$ bilinear mappings for evaluation, where n is the set size. Albeit a powerful tool, the scheme in [18] is not geared for delegated subset test due to the high computation cost. In this paper, we construct a secure delegated subset test scheme in the symmetric key setting. It preserves the predicate privacy against the delegator at the cost of $n+3$ bilinear mappings and produces ciphertext of $n + 3$ group elements, which is close to the optimal efficiency for a set with n elements.

1.1 Related Work

Keyword matching is a special case of subset test. Public-key Encryption with Keyword Search (PEKS), initially proposed by Boneh et al. [3], allows searching on data encrypted under a public key. Several variants of PEKS have been proposed with various improvements. Crescenzo and Saraswatt [10] proposed PEKS by using Jacobi symbols. The requirement for secure channels were removed in [1, 6], whereby a user provides the trapdoors (or filters) for the server. Schemes proposed in [20, 21] focus on resistance to the offline keyword guessing attacks. Baek et. al. introduced the concept of keyword refreshing in [1] which allows a PEKS system to extend its keyword space. An extension of keyword matching is to test whether a keyword is in a keyword set. Such keyword membership test schemes were proposed in [12, 17, 6, 15].

To determine whether multiple keywords are in a keyword set is essentially a subset test problem. Subset test is related to predicate evaluation on encrypted data. Shi et al. proposed a scheme [19] which allows for conjunctive range queries over encrypted data. Hidden-Vector Encryption (HVE) [5, 13, 2] is another public key encryption primitive which supports conjunctive comparison query, subset query and range query. Predicate encryption [14] is a more powerful tool than HVE, as it evaluates the inner product of two vectors. However, Shen etl.al remarked in [18] that predicate encryption in the public key setting can not fully preserve predicate privacy, and proposed a symmetric-key based predicate encryption with predicate privacy.

Delegated subset tests are related to, but *different* from private set operations [11, 16, 9, 8, 7] such as set union and intersection. The latter is a special case of secure two party computation whereby two participants jointly compute a set operation without exposing their respective data. The main cryptographic building block is homomorphic encryption. Note that they are not a solution to the problem studied in this paper, due to the complete different protocol setting.

1.2 Organization

The rest of this paper is organized as follows. We formalize the delegated subset test scheme and the security notions in Section 2. Section 3 describes the pre-

liminaries and the complexity assumption we will use for security proofs. Our proposed construction and its proofs are presented in Section 4. We conclude the paper in Section 5.

2 Problem Formalization

2.1 Definitions

Let Σ denote the universe of n values. Given two sets $A, B \subset \Sigma$, a *delegated subset test* scheme is for a semi-honest delegator to evaluate the predicate $A \subseteq B$ without knowledge of the contents in A or B. If A, B are represented by two n-bit bitmap $\boldsymbol{a}, \boldsymbol{b}$ respectively, the evaluation of $A \subseteq B$ is equivalent to the evaluation of bit operation $\boldsymbol{a} \wedge \boldsymbol{b} \oplus \boldsymbol{a}$. To differentiate $\boldsymbol{a}$ from $\boldsymbol{b}$, we refer to $\boldsymbol{a}$ by *filter*, and to $\boldsymbol{b}$ by *message*. For ease of presentation in the sequel, we misuse the notation of $\subseteq$ by using "$\boldsymbol{a} \subseteq \boldsymbol{b}$" to denote "$\boldsymbol{a} \wedge \boldsymbol{b} \oplus \boldsymbol{a}$" or equivalently "$A \subseteq B$".

Intuitively, a secure delegated subset test is to ensure the privacy of $\boldsymbol{a}$ and $\boldsymbol{b}$. Recall that the privacy of $\boldsymbol{a}$ or $\boldsymbol{b}$ will not be preserved if they are encrypted using public key encryption, as the evaluator may test the encrypted mask by using subsets of her own choices. Therefore, a secure delegated subset test only works under a symmetric key setting, whose definition is formalized below.

Definition 1 (Secure Delegated Subset Test). *A secure delegated subset test scheme consists of the following four probabilistic polynomial time (PPT) algorithms.*

Setup(1^λ) *takes as input a security parameter* 1^λ *and outputs a secret key* SK.

GenFilter$(SK, \boldsymbol{a})$ *takes as input the secret key* SK *and a binary vector (filter)* $\boldsymbol{a} \in \{0,1\}^n$ *representing set* A*, and outputs a filter token* $T_{\boldsymbol{a}}$.

GenSet$(SK, \boldsymbol{b})$ *takes as inputs the secret key* SK *and a binary vector (plaintext)* $\boldsymbol{b} \in \{0,1\}^n$ *representing set* B*, and outputs an encrypted message vector* $C_{\boldsymbol{b}}$.

Test$(C_{\boldsymbol{b}}, T_{\boldsymbol{a}})$ *(run by the delegator) takes as input an encrypted vector* $C_{\boldsymbol{b}}$*, and a token* $T_{\boldsymbol{a}}$*, and outputs* 0 *if* $\boldsymbol{a} \not\subseteq \boldsymbol{b}$*; otherwise outputs 1.*

The definition above describes a delegated subset test in a generalized fashion. The executioner of individual algorithms may vary from application to application. Taking a publish-subscribe network as an example, the publisher runs Setup, GenSet and the subscribers run their GenFilter, while the brokers, i.e. the delegators in our terminology, run Test. A secret key is shared between the publisher and subscribers. In other applications, such as searches on the outsourced document corpus, a user may run all algorithms except Test, which is executed by the search engine (i.e. a delegator). The timing of the execution of Test also depends on applications. Both the filters and the data sets can be static or dynamically updated. For example, in publish-subscribe applications, filters are static while data sets are dynamic. Note that the timing of running Test does not affect its security. Our study focuses on protecting data privacy against the delegators.

2.2 Notion of Security

Informally, the privacy notion implies that the delegator learns no additional information about the two sets A and B in evaluation except the test result computed from the respective filter token $T_{\boldsymbol{a}}$ and the encrypted vector $C_{\boldsymbol{b}}$. More formally, we define the privacy notions for the filter and the message separately, because the subset test computation is not commutative. The formal notions are defined using two privacy game similar to the ones used in [18]. The games are between a challenger which is a simulator and an adversary $\mathcal{A}$ attacking a delegated subset test scheme. We first define $Game_P$ for the plaintext privacy.

Init: The adversary chooses an arbitrary index set $I^* \subseteq [1, n]$, and sends it to the challenger.

Setup: The challenger runs Setup(1^λ) and keeps SK to itself.

Query Phase 1: $\mathcal{A}_P$ adaptively issues queries of the following two types:

- Filter query: On the i-th filter query, $\mathcal{A}_P$ outputs a bit $t = 0$ (indicating a filter query) and a binary vector $\boldsymbol{a_i}$ with the restriction that $\boldsymbol{a_i}[i^*] = 0$ for all $i^* \in I^*$. The challenger responds with $\mathsf{GenFilter}(SK, \boldsymbol{a_i})$.
- Ciphertext query: On the j-th ciphertext query, $\mathcal{A}_P$ outputs a bit $t = 1$ (indicating a ciphertext query) and a binary vector $\boldsymbol{b_j}$. The challenger responds with $\mathsf{GenSet}(SK, \boldsymbol{b_j})$.

Challenge: $\mathcal{A}_P$ makes a ciphertext challenge as below:

- $\mathcal{A}_P$ outputs two binary vectors $\boldsymbol{b_0^*}$ and $\boldsymbol{b_1^*}$ such that, the set $I := \{i | \boldsymbol{b_0^*}[i] \neq \boldsymbol{b_1^*}[i], 1 \leq i \leq n\}$ equals I^*. Due to these restrictions, for all previous filter queries $\boldsymbol{a_i}$, $\mathsf{Test}(\boldsymbol{a_i}, \boldsymbol{b_0^*}) = \mathsf{Test}(\boldsymbol{a_i}, \boldsymbol{b_1^*})$. The challenge picks a random bit β and responds with $\mathsf{GenSet}(SK, \boldsymbol{b_\beta^*})$.

Query Phase 2: $\mathcal{A}_P$ adaptively issues additional queries as in **Query Phase 1**, subject to the same restriction with respect to the challenge index set as above.

Guess: $\mathcal{A}_P$ outputs a guess β' of β.

The advantage of $\mathcal{A}_P$ is defined as $\mathrm{Adv}_{\mathcal{A}_P} = |\Pr[\beta' = \beta] - 1/2|$.

Definition 2 (Plaintext Privacy). *A delegated subset test scheme has* plaintext privacy *if, for all PPT adversaries $\mathcal{A}_P$, the advantage of $\mathcal{A}_P$ in winning $Game_P$ is negligible in λ.*

Next, we describe $Game_F$ for the notion of filter privacy.

Init: The adversary chooses an arbitrary index set $I^* \subseteq [1, n]$, and sends it to the challenger.

Setup: The challenger runs Setup(1^λ) and keeps SK to itself.

Query Phase 1: $\mathcal{A}_F$ adaptively issues queries of the following two types:

- Filter query: On the i-th filter query, $\mathcal{A}_F$ outputs a bit $t = 0$ (indicating a filter query) and a binary vector $\boldsymbol{a_i}$. The challenger responds with $\mathsf{GenFilter}(SK, \boldsymbol{a_i})$.

– Ciphertext query: On the j-th ciphertext query, $\mathcal{A}_F$ outputs a bit $t = 1$ (indicating a ciphertext query) and a binary vector $\boldsymbol{b_j}$ with the restriction that $\boldsymbol{b_j}[i^*] = 1$ for all $i^* \in I^*$. The challenger responds with $\mathsf{GenSet}(SK, \boldsymbol{b_j})$.

Challenge: $\mathcal{A}_F$ makes a filter challenge as below:

– $\mathcal{A}$ outputs two binary vectors $\boldsymbol{a_0^*}$ and $\boldsymbol{a_1^*}$ such that, the set $I := \{i | \boldsymbol{a_0^*}[i] \neq \boldsymbol{a_1^*}[i], 1 \leq i \leq n\}$ equals I^*. Due to these restrictions, for all previous ciphertext queries $\boldsymbol{b_j}$, $\mathsf{Test}(\boldsymbol{a_0^*}, \boldsymbol{b_j}) = \mathsf{Test}(\boldsymbol{a_1^*}, \boldsymbol{b_j})$ The challenger picks a random bit β and responds with $\mathsf{GenFilter}(SK, \boldsymbol{a_\beta^*})$.

Query Phase 2: $\mathcal{A}$ adaptively issues additional queries as in **Query Phase 1**, subject to the same restriction with respect to the challenge as above.

Guess: $\mathcal{A}_F$ outputs a guess β' of β.

The advantage of $\mathcal{A}_F$ is defined as $\mathrm{Adv}_{\mathcal{A}_F} = |\Pr[\beta' = \beta] - 1/2|$.

Definition 3 (Filter Privacy). *A delegated subset test scheme has* filter privacy *if, for all PPT adversaries $\mathcal{A}_F$, the advantage of $\mathcal{A}_F$ in winning $Game_F$ is negligible in λ.*

3 Background and Complexity Assumptions

In this section, we briefly review some known facts about bilinear groups of a composite order and the complexity assumption we will use for a formal proof.

3.1 Bilinear Groups of Composite Order

Let $\mathcal{G}$ denote a group generation algorithm that takes as input a security parameter 1^λ and outputs a tuple $(p, q, r, s, \mathbb{G}, \mathbb{G}_T, e)$ where p, q, r, s are distinct large primes; $\mathbb{G}$ and $\mathbb{G}_T$ are two cyclic groups of order $N = pqrs$; and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ satisfies the following properties:

– (Bilinear) $\forall u, v \in \mathbb{G}$, $\forall a, b \in \mathbb{Z}_N$, $e(u^a, v^b) = e(u, v)^{ab}$.
– (Non-degenerate) $\exists g \in \mathbb{G}$ such that $e(g, g)$ has order N in $\mathbb{G}_T$.

We assume that group operations in $\mathbb{G}$ and $\mathbb{G}_T$ as well as the bilinear map e can be computed efficiently. We use $\mathbb{G}_p, \mathbb{G}_q, \mathbb{G}_r, \mathbb{G}_s$ to denote the subgroups of $\mathbb{G}$ with order p, q, r, s, respectively. In addition, elements in $\mathbb{G}$ have the following properties.

– $\forall a \in \mathbb{G}_x, b \in \mathbb{G}_{x'}$ where $x, x' \in \{p, q, r, s\}$ and $x \neq x'$, $e(a, b) = 1$.
– Let $\mathbb{G}_{pq} = \mathbb{G}_p \times \mathbb{G}_q$. $\forall a, b \in \mathbb{G}_{pq}$, there exist unique $a_p, b_p \in \mathbb{G}_p, a_q, b_q \in \mathbb{G}_q$ satisfying $a = a_p a_q$, $b = b_p b_q$, and $e(a, b) = e(a_p, b_p)e(a_q, b_q)$.

In the sequel, we will continue to use the group setting of $(p, q, r, s, \mathbb{G}, \mathbb{G}_T, e)$ with the same notations described above.

3.2 Our Assumptions

The complexity assumptions used in this paper are the same as Assumption W and ℓ-DLinear Assumption previously stated in [18], and the former could be reduced to Assumption 1 in [14].

Assumption W Let $(p, q, r, s, \mathbb{G}, \mathbb{G}_T, e)$ be the group setting described in Section 3.1. Let g_p, g_q, g_r, g_s be random generators of $\mathbb{G}_p, \mathbb{G}_q, \mathbb{G}_r, \mathbb{G}_s$, respectively. Choose random $R'_1, R'_2 \in \mathbb{G}_r$, random $a, b, s \in \mathbb{Z}_q$ and a random bit γ. If $\gamma = 0$, $\beta = 0$, otherwise β is chosen from $\mathbb{Z}_p$. Adversary $\mathcal{A}$ is given the description of the bilinear group $(N, \mathbb{G}, \mathbb{G}_T, e)$, along with the following values:

$$\left(g_q, g_r, g_s, g_p R'_1, g_p g_q^a, g_q^b, g_q^c, T = g_p^\beta g_q^{bc} R'_2\right)$$

The adversary $\mathcal{A}$ outputs a guess γ' of γ. The advantage of $\mathcal{A}$ is defined as $\mathrm{Adv}_{\mathcal{A}} = |\Pr[\gamma' = \gamma] - 1/2|$.

Definition 4. *We say that $\mathcal{G}$ satisfies the above assumption if, for all PPT algorithm $\mathcal{A}$, the advantage of $\mathcal{A}$ in winning the above game is negligible in the security parameter λ.*

ℓ-DLinear Assumption Let $\mathcal{G}$ be a group generation algorithm. Run $\mathcal{G}(1^\lambda)$ to obtain $(p, q, r, s, \mathbb{G}, \mathbb{G}_T, e)$. Let $N = pqrs$ and let g_p, g_q, g_r, g_s be random generators of $\mathbb{G}_p, \mathbb{G}_q, \mathbb{G}_r, \mathbb{G}_s$, respectively. Let ℓ be an integer greater than 2. Choose two random vectors $\boldsymbol{y} = (y_1, y_2, \ldots, y_\ell) \xleftarrow{R} \mathbb{F}_q^\ell$ and $\boldsymbol{z} = (z_1, z_2, \ldots, z_\ell) \xleftarrow{R} \mathbb{F}_q^\ell$. Choose a random bit β. Choose a vector $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \ldots, \gamma_\ell)$ in one of two ways, depending on the value of β. If $\beta = 0$, choose $\gamma_1, \gamma_2, \ldots, \gamma_\ell$ independently at random from $\mathbb{F}_q$. In other words, the vector $\boldsymbol{\gamma}$ is picked at random from the vector space $\mathbb{F}_q^\ell$. If $\beta = 1$, choose the vector $\boldsymbol{\gamma}$ from the 2-dimensional subspace $\mathbf{span}(\boldsymbol{y}, \boldsymbol{z})$ of $\mathbb{F}_q^\ell$ generated by $\boldsymbol{y}, \boldsymbol{z}$. Specially, choose random $w, t \xleftarrow{R} \mathbb{Z}_q$ and let $\boldsymbol{\gamma} = w\boldsymbol{y} + t\boldsymbol{z}$. Define the following notation:

$$g_q^{\boldsymbol{x}} := \left(g_q^{x_1}, g_q^{x_2}, \ldots, g_q^{x_\ell}\right) \qquad \text{where } \boldsymbol{x} \in \mathbb{F}_q^\ell.$$

Give the adversary the description of the group, $(N = gprs, \mathbb{G}, \mathbb{G}_T, e)$, the generators of each subgroup, g_p, g_q, g_r, g_s, and the following tuple:

$$\left(g_q^{\boldsymbol{y}}, g_q^{\boldsymbol{z}}, g_q^{\boldsymbol{\gamma}}\right).$$

The adversary outputs a guess β' of the bit β. The advantage of $\mathcal{A}$ is defined as $\mathrm{Adv}_{\mathcal{A}} = |\Pr[\beta' = \beta] - 1/2|$.

Definition 5. *We say that $\mathcal{G}$ satisfies the ℓ-DLinear assumption if, for all PPT algorithms $\mathcal{A}$, the advantage of $\mathcal{A}$ in winning the above game is negligible in the security parameter λ.*

4 Secure Delegated Subset Test

4.1 The Rationale

We begin with an intuitive explanation of our construction, which is different from a direct application of inner product computation. Given two binary vectors $\boldsymbol{a} = \langle a_1, \cdots, a_n \rangle, \boldsymbol{b} = \langle b_1, \cdots, b_n \rangle$ representing set $A, B \subset \Sigma$ respectively, let $\Gamma_{a,b}$ denote the index set $\{i | 1 \leq i \leq n, a_i = 1, b_i = 0\}$. Thus, $\boldsymbol{a} \subseteq \boldsymbol{b}$ is true if and only if $\Gamma = \emptyset$. Based on this observation, we construct an encryption scheme for $\boldsymbol{a}, \boldsymbol{b}$ whereby the randomness has the following property: if $\Gamma \neq \emptyset$, the randomness will not be nullified when testing $\boldsymbol{a} \subseteq \boldsymbol{b}$. Otherwise, the randomness is removed.

More specifically, $\boldsymbol{a}$ is represented with an $n+2$ dimension vector $\boldsymbol{a}'$ whereby all "1"s in $\boldsymbol{a}$ are represented by a common random number and each "0" is replaced by a unique random number. The remaining two coordinates in $\boldsymbol{a}'$ contain information about the sum of these two types of random numbers. In contrast, all "0"s in $\boldsymbol{b}$ are replaced by a common random number whereas each "1" in $\boldsymbol{b}$ is replaced by a unique random number. This setup allows all randomness to be exactly annihilated when $\Gamma_{a,b} = \emptyset$. The privacy of $\boldsymbol{a}$ and $\boldsymbol{b}$ are then protected by padding with $\mathbb{G}_r$ and $\mathbb{G}_q$ elements, a technique used many schemes [4, 14].

REMARK Our subset test has no false negatives. Namely, if $\boldsymbol{a} \subseteq \boldsymbol{b}$, Test always returns *true*. However, it has negligible false positives. If $\boldsymbol{a} \not\subseteq \boldsymbol{b}$, it is possible for Test to output *true*. This only occurs when those random numbers happen to cancel each other. We argue that the false positive rate is negligible because the random numbers are drawn from a significantly large domain.

4.2 Our Construction

We now describe our construction in detail. The five algorithms of the proposed secure delegated subset test scheme are described as below.

Setup(1^λ): The setup algorithm proceeds as follows.

- run $\mathcal{G}(1^\lambda)$ to generate $N = pqrs, \mathbb{G}, \mathbb{G}_T, e$ with $\mathbb{G} = \mathbb{G}_p \times \mathbb{G}_q \times \mathbb{G}_r \times \mathbb{G}_s$ and picks random generators g_p, g_q, g_r, g_s of $\mathbb{G}_p, \mathbb{G}_q, \mathbb{G}_r, \mathbb{G}_s$ respectively.
- for each $i \in [1, n+2]$, select $h_i \in_R \mathbb{G}_q$, where n is the size of binary vectors.

The secret key is

$$SK = (g_p, g_q, g_r, g_s, \{h_i\}_{i=1}^{n+2}).$$

GenFilter($SK, \boldsymbol{a}$): Let $\boldsymbol{a} = \langle a_1, \ldots, a_n \rangle$ be a binary filter vector. Define a set $W := \{i | a_i = 1, 1 \leq i \leq n\}$. Then set $\{q_i\}_{i=1}^{n+2} \xleftarrow{R} \mathbb{Z}_N^{n+2}$ and $\{S_i\}_{i=1}^{n+2} \xleftarrow{R} \mathbb{G}_s^{n+2}$ and choose a random w from $\mathbb{Z}_N$ uniformly.
For each $i = 1$ to n,

- If $i \notin W$ ($a_i = 0$), set $w_i = w$; Otherwise (i.e. $a_i = 1$), set $w_i \xleftarrow{R} \mathbb{Z}_N$;
- Compute $T_i = g_p^{w_i} g_q^{q_i} S_i$.

Choose a random R_{n+3} from $\mathbb{G}_r$. Set

$$T_{n+1} = g_p^{\sum_{i=1}^{n} w_i} g_q^{q_{n+1}} S_{n+1}$$
$$T_{n+2} = g_p^{-w} g_q^{q_{n+2}} S_{n+2}$$
$$T_{n+3} = \prod_{i=1}^{n+2} h_i^{-q_i} R_{n+3}.$$

The output filter token is $T_{\boldsymbol{a}} = \{T_i\}_{i=1}^{n+3}$.

GenSet($SK, \boldsymbol{b}$): Let $\boldsymbol{b} = \langle b_1, \ldots, b_n \rangle$ be a binary message vector. Define a set $V := \{i | b_i = 0, 1 \leq i \leq n\}$. Choose $\delta, v \xleftarrow{R} \mathbb{Z}_N$ and $\{R_i\}_{i=1}^{n+2} \xleftarrow{R} \mathbb{G}_r^{n+2}$. For each $i = 1$ to n,

- If $i \notin V$ ($b_i = 1$), set $v_i = v$; Otherwise (i.e. $b_i = 0$), set $v_i \xleftarrow{R} \mathbb{Z}_N$;
- Compute $C_i = g_p^{v_i} h_i^{\delta} R_i$.

Choose a random S_{n+3} from $\mathbb{G}_s$. Set

$$C_{n+1} = g_p^{-v} h_{n+1}^{\delta} R_{n+1}$$
$$C_{n+2} = g_p^{\sum_{i \in V} (v_i - v)} h_{n+2}^{\delta} R_{n+2}$$
$$C_{n+3} = g_q^{\delta} S_{n+3}.$$

The ciphertext is $C_{\boldsymbol{b}} = \{C_i\}_{i=1}^{n+3}$

Test($T_{\boldsymbol{a}}, C_{\boldsymbol{b}}$): Suppose $T_{\boldsymbol{a}} = \{T_i\}_{i=1}^{n+3}$ and $C_{\boldsymbol{b}} = \{C_i\}_{i=1}^{n+3}$. Compute

$$\alpha = \prod_{i=1}^{n+3} e\left(T_i, C_i\right).$$

If $\alpha = 1$, output 1; otherwise output 0.

The correctness of the scheme can be easily verified as follows. Given $T_{\boldsymbol{a}}$ and $C_{\boldsymbol{b}}$ properly produced by GenFilter and GenSet respectively. Then

$$\begin{aligned}
\alpha = \prod_{i=1}^{n+3} e(T_i, C_i) &= \prod_{i=1}^{n} e\left(g_p^{w_i} g_q^{q_i} S_i, \quad g_p^{v_i} h_i^{\delta} R_i\right) \\
&\cdot e\left(g_p^{\sum_{i=1}^{n} w_i} g_q^{q_{n+1}} S_{n+1}, \ g_p^{-v} h_{n+1}^{\delta} R_{n+1}\right) \\
&\cdot e\left(g_p^{-w} g_q^{q_{n+2}} S_{n+2}, \quad g_p^{\sum_{i \in V} (v_i - v)} h_{n+2}^{\delta} R_{n+2}\right) \\
&\cdot e\left(\prod_{i=1}^{n+2} h_i^{-q_i} R_{n+3}, \quad g_q^{\delta} S_{n+3}\right) \\
&= e(g_p, g_p)^{\sum_{i=1}^{n} v_i w_i} \cdot e(g_p, g_p)^{-\sum_{i=1}^{n} w_i v} \cdot e(g_p, g_p)^{-\sum_{i \in V} w(v_i - v)} \\
&= e(g_p, g_p)^{\sum_{i=1}^{n} v_i w_i} \cdot e(g_p, g_p)^{(|W| + |V| - n) w v - \sum_{i \in W} w_i v - \sum_{i \in V} w v_i}.
\end{aligned}$$

A brief explanation is as follows. All indexes are divided into four subsets S_1, S_2, S_3 and $\Gamma_{a,b}$, where $S_1 := \{i | i \notin W \text{ and } i \notin V\}$, $S_2 := \{i | i \in W \text{ and } i \notin$

$V\}$, $S_3 := \{i | i \notin W \text{ and } i \in V\}$, $\Gamma_{a,b} := \{i | i \in W \text{ and } i \in V\}$. Recall that if $\Gamma_{a,b} \neq \emptyset$, then $\boldsymbol{a} \not\subseteq \boldsymbol{b}$. In fact, when $\Gamma_{a,b} = W \cap V$. Therefore, If $\Gamma_{a,b} = \emptyset$, $|W| + |V| = |W \cup V|$ and $\{1, \cdots, n\} = S_1 \cup S_2 \cup S_3$. As a result, $\sum_{i=1}^{n} v_i w_i = (n - |W| + |V|)wv + \sum_{i \in W} w_i v + \sum_{i \in V} w v_i$, since the exponent of the last term of the last equation corresponds to the joint set of S_1, S_2 and S_3. Thus, $\alpha = 1$ when $\boldsymbol{a} \subseteq \boldsymbol{b}$.

Note that, if $\Gamma_{a,b} \neq \emptyset$, it is still likely that $\alpha = 1$ since $\sum_{i=1}^{n} v_i w_i$ may happen to be the negative of $(|W| + |V| - n)wv - \sum_{i \in W} w_i v - \sum_{i \in V} w v_i$. However, as w_i-s and v_i-s are randomly generated, the probability is $1/N$ which is negligible in λ.

4.3 Proof of Security

As defined in Section 2, the privacy notion in delegated subset test scheme implies both the plaintext privacy and the filter privacy. Since GenFilter is a dual of GenSet and the proofs for plaintext privacy and filter privacy are similar, we only focus on proving plaintext privacy in order to avoid verbosity. The proof for filter privacy can be achieved under ℓ-DLinear Assumption using a proving technique in the extended version of [18].

Plaintext Privacy. Informally, the plaintext privacy of a delegate subset test scheme means that there exists no PPT adversary $\mathcal{A}_P$ which could distinguish two plaintext from ciphertexts generated by GenSet. We start with a special case where the two plaintext vectors only have one bit difference.

Lemma 1. *Under Assumption W, for all $\boldsymbol{b_0}, \boldsymbol{b_1}$ satisfying that there exists unique $i \in [1, n]$ and $\boldsymbol{b_0}[i] \neq \boldsymbol{b_1}[i]$ and $\boldsymbol{b_0}[j] = \boldsymbol{b_1}[j]$ for all $j \neq i, j \in [1, n]$, the ciphertext $C_0^* \leftarrow GenSet(SK, \boldsymbol{b}_0)$ is computationally indistinguishable from another ciphertext $C_1^* \leftarrow GenSet(SK, \boldsymbol{b}_1)$ for all polynomial time adversary, who could issue filter token inquiries $\boldsymbol{a}$ with restriction that the i-th element of $\boldsymbol{a}$ is 0, i.e. $a_i = 0$.*

Proof. In the following proof, without loss of generality we fix an arbitrary $i^* \in [1, n]$ as the index of different bits. Suppose there exists an adversary $\mathcal{A}_P$ can distinguish the ciphertexts of two vectors with one bit difference in $Game_P$ defined in Section 2.2, we can leverage its ability to build a simulator $\mathcal{B}$ that can break Assumption W.

A high level view of the proof is as follows. The simulator $\mathcal{B}$ is given an instance of Assumption W, and it plays $Game_P$ with the adversary $\mathcal{A}_P$. The adversary issues ciphertext queries and filter queries subject to the stated restriction. To respond to these queries, $\mathcal{B}$ computes corresponding ciphertexts and filters using parameters of the assumption instance. The resulting ciphertexts and filter tokens are distributed identically as normal. In the challenge phase, $\mathcal{A}_P$ sends $\mathcal{B}$ a challenge plaintext pair $(\boldsymbol{b}_0^*, \boldsymbol{b}_1^*)$, satisfying $\boldsymbol{b}_0^*, \boldsymbol{b}_1^*$ only have one bit difference at i^*. $\mathcal{B}$ responds with an encryption embedding the challenge from Assumption W, such that if $\mathcal{A}_P$ wins the game, $\mathcal{B}$ breaks the assumption in the same probability of $\mathcal{A}_P$. Next, we present the details of the simulator $\mathcal{B}$.

Initial. The adversary $\mathcal{A}_P$ chooses a challenge index $i^* \in [1, n]$, and sends it to the simulator $\mathcal{B}$.

Setup. The simulator $\mathcal{B}$ is given an instance of Assumption W with the following parameters:

$$\left[(N, \mathbb{G}, \mathbb{G}_T, e), g_q, g_r, g_s, g_p R'_1, g_p g_q^a, g_q^b, g_q^c, T = g_p^\beta g_q^{bc} R'_2\right]$$

$\mathcal{B}$'s objective is to determine whether $\beta = 0$ or not.

$\mathcal{B}$ uses these parameters to create a secret key for a secure delegated subset test:

$$SK = \left(g_q, g_r, g_s, \{h_i\}_{i=1}^{n+2}\right),$$

where $\{h_i = g_q^{\mu_i}\}_{1 \le i \le n+1, i \ne i^*}$ and $h_{i^*} = g_q^{c+\mu_{i^*}}, h_{n+2} = g_q^{c+\mu_{i_{n+2}}}$ are generated as follows. For each $i \in [1, n+2]$, chooses random $\mu_i \in \mathbb{Z}_N$ and computes $h_{i^*} = g_q^c \cdot g_q^{\mu_{i^*}}, h_{n+2} = g_q^c \cdot g_q^{\mu_{n+2}}$ and $h_i = g_q^{\mu_i}$ for other i, where g_q^c, g_q are inherited from the assumption instance. Note that h_i are distributed in $\mathbb{G}_q$ randomly and uniformly.

Ciphertext query: Given a query vector $\boldsymbol{b}$ from $\mathcal{A}_P$, $\mathcal{B}$ responds with ciphertext $C_{\boldsymbol{b}}$ which is generated in the same fashion as in GenSet, except that g_p is replaced by $g_p R'_1$ from the assumption instance.

$$\begin{aligned}
C_i &= (g_p R'_1)^{v_i} \cdot h_i^\delta \cdot R''_i = g_p^{v_i} \cdot h_i^\delta \cdot R_i, \quad \text{for} 1 \le i \le n \\
C_{n+1} &= (g_p R'_1)^{-v} \cdot h_{n+1}^\delta \cdot R''_{n+1} = g_p^{-v} \cdot h_{n+1}^\delta \cdot R_{n+1} \\
C_{n+2} &= (g_p R'_1)^{\sum_{i \in V}(v_i - v)} \cdot h_{n+2}^\delta \cdot R''_{n+2} = g_p^{\sum_{i \in V}(v_i - v)} \cdot h_{n+2}^\delta \cdot R_{n+2} \\
C_{n+3} &= g_q^\delta \cdot S_{n+3}.
\end{aligned}$$

Filter query: Given a query vector $\boldsymbol{a}$ from $\mathcal{A}_P$, $\mathcal{B}$ chooses randoms w, w_i, S_i in the same fashion as in GenFilter, and randoms $q'_i \in \mathbb{Z}_N$ for each $i \in [1, n+2]$, then constructs the response as follows:

$$\begin{aligned}
T_i &= (g_p g_q^a)^{w_i} \cdot g_q^{q'_i} \cdot S_i = g_p^{w_i} \cdot g_q^{a w_i + q'_i} \cdot S_i \quad \text{for } 1 \le i \le n \\
T_{n+1} &= (g_p g_q^a)^{\sum_{i=1}^n w_i} \cdot g_q^{q'_{n+1}} \cdot S_{n+1} = g_p^{\sum_{i=1}^n w_i} \cdot g_q^{a \sum_{i=1}^n w_i + q'_{n+1}} \cdot S_{n+1} \\
T_{n+2} &= (g_p g_q^a)^{-w} \cdot g_q^{q'_{n+2}} \cdot S_{n+2} = g_p^{-w} \cdot g_q^{-aw + q'_{n+2}} \cdot S_{n+2}.
\end{aligned}$$

Apparently, the randoms $\{q_i\}_{i=1}^{n+2}$ in GenFilter have the following values:

$$\begin{aligned}
q_i &= a w_i + q'_i, \qquad \text{for each } i \in [1, n] \\
q_{n+1} &= a \sum_{i=1}^n w_i + q'_{n+1} \\
q_{n+2} &= -aw + q'_{n+2}.
\end{aligned}$$

Due to the aforementioned restriction, we have $w_{i^*} = w$. Next we continue to address the last component of the response.

$$
\begin{aligned}
T_{n+3} &= \prod_{i=1}^{n+2} h_i^{-q_i} R_{n+3} = \prod_{i=1, i\neq i^*}^{n+1} \left(g_q^{\mu_i}\right)^{-q_i} \cdot \left(g_q^{c+\mu_{i^*}}\right)^{-q_{i^*}} \cdot \left(g_q^{c+\mu_{n+2}}\right)^{-q_{n+2}} \cdot R_{n+3} \\
&= \prod_{i=1, i\neq i^*}^{n} g_q^{-\mu_i(aw_i+q_i')} \cdot g_q^{-(c+\mu_{i^*})(aw_{i^*}+q_{i^*}')} \\
&\quad \cdot g_q^{-\mu_{n+1}(a\sum_{i=1}^{n} w_i + q_{n+1}')} \cdot g_q^{(c+\mu_{n+2})(aw-q_{n+2}')} \cdot R_{n+3} \\
&= \prod_{i=1}^{n} g_q^{-\mu_i(aw_i+q_i')} \cdot g_q^{-\mu_{n+1}(a\sum_{i=1}^{n} w_i + q_{n+1}')} \cdot g_q^{-caw_{i^*}} \cdot g_q^{caw} \\
&\quad \cdot g_q^{-cq_{i^*}' + aw\mu_{n+2} - q_{n+2}'(c+\mu_{n+2})} \cdot R_{n+3}.
\end{aligned}
$$

Note that since $w_{i^*} = w$, the two components $g_q^{-caw_{i^*}}$ and g_q^{caw} are canceled out, and the rest components could be calculated using $g_q^a R^* = \frac{g_p g_q^a}{g_p R_1'}$ and g_q^c easily, where $g_p g_q^a, g_p R_1', g_q^c$ are inherent from the assumption instance. Thus, T_{n+3} could be calculated, and has correct distribution.

Challenge. After a polynomial number of filter queries and ciphertext queries, the adversary $\mathcal{A}_P$ outputs the challenge plaintext pair $(\boldsymbol{b}_0^*, \boldsymbol{b}_1^*)$ such that $\boldsymbol{b}_0^*[i^*] \neq \boldsymbol{b}_1^*[i^*]$ and $\boldsymbol{b}_0^*[j] = \boldsymbol{b}_1^*[j]$ for all $j \in [1, n], j \neq i^*$. Without loss of generality, let $\boldsymbol{b}_0^*[i^*] = 0$ and $\boldsymbol{b}_1^*[i^*] = 1$. In response, $\mathcal{B}$ selects v_i, v, R_i'' in the same way as in GenSet, and returns the following to $\mathcal{A}_P$.

$$
\begin{aligned}
C_i^* &= (g_p R_1')^{v_i} \cdot (g_q^b)^{\mu_i} \cdot R_i'' = g_p^{v_i} \cdot h_i^b \cdot R_i \text{ for } 1 \leq i \leq n,\ i \neq i^* \\
C_{i^*}^* &= T \cdot (g_p R_1')^{v} \cdot (g_q^b)^{\mu_{i^*}} \cdot R_{i^*}'' = g_p^{\beta+v} \cdot h_{i^*}^b \cdot R_{i^*} \\
C_{n+1}^* &= (g_p R_1')^{-v} \cdot (g_q^b)^{\mu_{n+1}} \cdot R_{n+1}'' = g_p^{-v} \cdot h_{n+1}^b \cdot R_{n+1} \\
C_{n+2}^* &= T \cdot \prod_{i \in V \setminus \{i^*\}} (g_p R_1')^{v_i - v} \cdot (g_q^b)^{\mu_{n+2}} \cdot R_{n+2}'' \\
&= g_p^{\sum_{i \in V}(v_i - v)} \cdot h_{n+2}^b \cdot R_{n+2} \\
C_{n+3}^* &= g_q^b \cdot S_{n+3}.
\end{aligned}
$$

$\mathcal{A}_P$ may continue to issue queries and $\mathcal{B}$ responds as explained above. In the end, $\mathcal{A}_P$ outputs the plaintext of $\{C_i^*\}_{i=1}^{n+3}$. If it outputs $\boldsymbol{b}_0^*$, $\mathcal{B}$ outputs $\beta \neq 0$. If it outputs $\boldsymbol{b}_1^*$, $\mathcal{B}$ outputs $\beta = 0$. This is because when $\beta \neq 0$, $\{C_i^*\}_{i=1}^{n+3}$ is the ciphertext of $\boldsymbol{b}_0^*$, and when $\beta = 0$, it is the ciphertext of $\boldsymbol{b}_1^*$.

In summary, $\mathcal{B}$ has the same success probability in breaking Assumption W as the success probability for $\mathcal{A}_P$ to distinguish two encrypted vectors with only one bit difference. □

Next we remove the restriction on the bit difference $\boldsymbol{b}_0, \boldsymbol{b}_1$ and prove that no PPT adversary can distinguish any $\boldsymbol{b}_0$ and $\boldsymbol{b}_1$.

Theorem 1 (plaintext privacy). *Under Assumption W, for all n-bit vector* $\boldsymbol{b_0^*}, \boldsymbol{b_1^*}$*, the ciphertext* $C_0^* \leftarrow_R GenSet(SK, \boldsymbol{b_0^*})$ *is computationally indistinguishable from another ciphertext* $C_1^* \leftarrow_R GenSet(SK, \boldsymbol{b_1^*})$ *for all polynomial time adversary, who could issue polynomial ciphertext queries and filter token quiries, where for each filter token query* $\boldsymbol{a}$ *has the restriction that* $\boldsymbol{a}[i] = 0$ *for all* $i \in \{i | 1 \leq i \leq n, \boldsymbol{b_0^*}[i] \neq \boldsymbol{b_1^*}[i]\}$ *and* $Test(\boldsymbol{a}, \boldsymbol{b_0^*}) = Test(\boldsymbol{a}, \boldsymbol{b_1^*})$.

Proof. The theorem can be proved by using a series of games and Lemma 1. Without loss of generality, suppose $\boldsymbol{b_0^*}$ and $\boldsymbol{b_1^*}$ are different in m bit positions $i_1, \cdots, i_m$, $1 \leq i_1 < i_2 \cdots < i_m \leq n$. We define a list of $m+1$ vectors $\boldsymbol{b_0'} = \boldsymbol{b_0^*}, \boldsymbol{b_1'}, \boldsymbol{b_2'}, \cdots, \boldsymbol{b_m'} = \boldsymbol{b_1^*}$, such that the only difference between two adjacent vectors $\boldsymbol{b_j'}$ and $\boldsymbol{b_{j-1}'}$ is at the i_j-th bit and $\boldsymbol{b_j'}[i_j] = \boldsymbol{b_1}[i_j]$, $1 \leq j \leq m$. Note that for any $\boldsymbol{a}$ satisfying $\boldsymbol{a} \subseteq \boldsymbol{b_0}$ and $\boldsymbol{a} \subseteq \boldsymbol{b_1}$, then $\boldsymbol{a} \subseteq \boldsymbol{b_i'}$ for all $1 \leq i \leq m$. This is because none of the bit positions with differences in $\boldsymbol{b_0}, \boldsymbol{b_1}$ impacts the subset test of $\boldsymbol{a}$.

The proof is the same as in Lemma 1 except the Challenge phase. In the Challenge phase, $\mathcal{A}_P$ outputs two vectors $\boldsymbol{b_0^*}$ and $\boldsymbol{b_1^*}$ satisfying the filter restriction defined in $Game_P$. Instead of returning C^* for either $\boldsymbol{b_0^*}$ and $\boldsymbol{b_1^*}$, the simulator $\mathcal{B}$ first returns C_1^* for $\boldsymbol{b_0'^*}$ and $\boldsymbol{b_1'^*}$ to $\mathcal{A}_P$ in the same way as in Lemma 1. If $\mathcal{A}_P$ fails, $\mathcal{B}$ first returns C_2^* for $\boldsymbol{b_1'^*}$ and $\boldsymbol{b_2'^*}$, and so on until returning C_m^*. Note that if $\mathcal{A}_P$ can distinguish $\boldsymbol{b_0^*}$ and $\boldsymbol{b_1^*}$, there exists $i \in [1, m]$ such that $\mathcal{A}_P$ succeeds in distinguishing C_i^*, which leads to breaking the assumption as in Lemma 1. □

Using the proving technique in the extended version of [18], we can prove the filter privacy under ℓ-DLinear Assumption. The main difference is that $\mathcal{A}_F$'s challenge vectors are treated as filters. Correspondingly, $\mathcal{B}$ embeds the assumption challenge into the generation of filter token T^*. We omit the proof here to avoid redundancy.

Theorem 2 (filter privacy). *Under Assumption W and* ℓ*-DLinear Assumption, for all n-bit vector* $\boldsymbol{a_0^*}, \boldsymbol{a_1^*}$*, the filter token* $T_0^* \leftarrow_R GenFilter(SK, \boldsymbol{a_0^*})$ *is computationally indistinguishable from another filter token* $T_1^* \leftarrow_R GenFilter(SK, \boldsymbol{a_1^*})$ *for all polynomial time adversary, who could issue polynomial filter token queries and ciphertext queries, where for each ciphertext query* $\boldsymbol{b}$ *has the restriction that* $\boldsymbol{v}[i] = 1$ *for all* $i \in \{i | 1 \leq i \leq n, \boldsymbol{a_0^*}[i] \neq \boldsymbol{a_1^*}[i]\}$ *and* $Test(\boldsymbol{a_0^*}, \boldsymbol{b}) = Test(\boldsymbol{a_1^*}, \boldsymbol{b})$.

4.4 Efficiency Analysis

To show the improvement of efficiency of the proposed scheme, we compare it with a solution based on predicate-only encryption supporting inner product (POE) described in [18].

Firstly, we give a brief description of the POE-based solution. The SSW scheme proposed in [18] consists of 4 algorithms: $SK \leftarrow \mathsf{Setup}(1^\lambda)$, $CT_{\boldsymbol{x}} \leftarrow \mathsf{Encrypt}(SK, \boldsymbol{x})$, $TK_{\boldsymbol{v}} \leftarrow \mathsf{GenToken}(SK, \boldsymbol{v})$, $b \leftarrow \mathsf{Query}(TK_{\boldsymbol{v}}, CT_{\boldsymbol{x}})$. $\mathsf{Encrypt}$ and $\mathsf{GenToken}$ take as input 2 vectors $\boldsymbol{x}$ and $\boldsymbol{v}$, and output a ciphertext $CT_{\boldsymbol{x}}$ and a token $TK_{\boldsymbol{v}}$, separately; b equals to 1 if $\langle \boldsymbol{x}, \boldsymbol{v} \rangle = 0$, otherwise 0.

Given two vectors $\boldsymbol{a} = (a_1, \ldots, a_n)$ and $\boldsymbol{b} = (b_1, \ldots, b_n)$ for testing whether $\boldsymbol{a} \subseteq \boldsymbol{b}$ (as described in 2.1), we extend $\boldsymbol{a}$ and $\boldsymbol{b}$ and obtain two new vectors $\boldsymbol{a}' = (a_1, \ldots, a_n, n_a)$ and $\boldsymbol{b}' = (b_1, \ldots, b_n, -1)$, where n_a is the count of bit 1, i.e. $a_i = 1$. Clearly, if $\boldsymbol{a} \subseteq \boldsymbol{b}$, $\langle \boldsymbol{a}', \boldsymbol{b}' \rangle = 0$. To test the same vectors of length n, both ciphertext and token consist of $2n + 4$ elements from a group $\mathbb{G}$ of order N, and there are $2n+4$ times pairings in the algorithm Query in the POE-based solution. In the meantime, both ciphertext and token of our proposed scheme consist of $n + 3$ elements from a group $\mathbb{G}$ of order N, and there are $n + 3$ times pairings in the algorithm Test. In conclusion, our scheme is almost more efficient with half of the computation cost and half of the ciphertext size and the token size.

5 Conclusion

To summarize, we have proposed an efficient scheme for delegated subset test. Our scheme protects the privacy of both operands of a subset test, which falls in the general predicates privacy encryption [18]. Our construction is based on the characteristics of binary vectors. Therefore, it only costs $n + 3$ bilinear mapping operations and $n + 3$ group elements to achieve the same strength as using the inner-product based scheme.

Acknowledgements

We are grateful to the anonymous reviewers for their helpful comments. This work is supported by the Office of Research, Singapore Management University.

References

1. Baek, J., Safavi-Naini, R., Susilo, W.: Public key encryption with keyword search revisited. In: Computational Science and Its Applications - ICCSA 2008, Lecture Notes in Computer Science, vol. 5072, pp. 1249–1259. Springer Berlin / Heidelberg (2008)
2. Blundo, C., Iovino, V., Persiano, G.: Private-key hidden vector encryption with key confidentiality. In: Garay, J., Miyaji, A., Otsuka, A. (eds.) Cryptology and Network Security, Lecture Notes in Computer Science, vol. 5888, pp. 259–277. Springer Berlin / Heidelberg (2009)
3. Boneh, D., Di Crescenzo, G., Ostrovsky, R., Persiano, G.: Public key encryption with keyword search. In: Cachin, C., Camenisch, J. (eds.) Advances in Cryptology - EUROCRYPT 2004, Lecture Notes in Computer Science, vol. 3027, pp. 506–522. Springer Berlin / Heidelberg (2004)
4. Boneh, D., Goh, E.J., Nissim, K.: Evaluating 2-dnf formulas on ciphertexts. In: Kilian, J. (ed.) Theory of Cryptography, Lecture Notes in Computer Science, vol. 3378, pp. 325–341. Springer Berlin / Heidelberg (2005)
5. Boneh, D., Waters, B.: Conjunctive, subset, and range queries on encrypted data. In: Vadhan, S. (ed.) Theory of Cryptography, Lecture Notes in Computer Science, vol. 4392, pp. 535–554. Springer Berlin / Heidelberg (2007)

6. Byun, J., Lee, D., Lim, J.: Efficient conjunctive keyword search on encrypted data storage system. In: Public Key Infrastructure, Lecture Notes in Computer Science, vol. 4043, pp. 184–196. Springer Berlin / Heidelberg (2006)
7. Camenisch, J., Zaverucha, G.: Private intersection of certified sets. In: Financial Cryptography and Data Security, Lecture Notes in Computer Science, vol. 5628, pp. 108–127. Springer Berlin / Heidelberg (2009)
8. Cristofaro, E.D., Tsudik, G.: Practical private set intersection protocols with linear computational and bandwidth complexity. Cryptology ePrint Archive, Report 2009/491 (2009), http://eprint.iacr.org/
9. Dachman-Soled, D., Malkin, T., Raykova, M., Yung, M.: Efficient robust private set intersection. In: Applied Cryptography and Network Security, Lecture Notes in Computer Science, vol. 5536, pp. 125–142. Springer Berlin / Heidelberg (2009)
10. Di Crescenzo, G., Saraswat, V.: Public key encryption with searchable keywords based on jacobi symbols. In: Progress in Cryptology - INDOCRYPT 2007, Lecture Notes in Computer Science, vol. 4859, pp. 282–296. Springer Berlin / Heidelberg (2007)
11. Freedman, M.J., Nissim, K., Pinkas, B.: Efficient private matching and set intersection. In: Advances in Cryptology - EUROCRYPT 2004, Lecture Notes in Computer Science, vol. 3027, pp. 1–19. Springer Berlin / Heidelberg (2004)
12. Golle, P., Staddon, J., Waters, B.: Secure conjunctive keyword search over encrypted data. In: Applied Cryptography and Network Security, Lecture Notes in Computer Science, vol. 3089, pp. 31–45. Springer Berlin / Heidelberg (2004)
13. Iovino, V., Persiano, G.: Hidden-vector encryption with groups of prime order. In: Pairing-Based Cryptography - Pairing 2008, Lecture Notes in Computer Science, vol. 5209, pp. 75–88. Springer Berlin / Heidelberg (2008)
14. Katz, J., Sahai, A., Waters, B.: Predicate encryption supporting disjunctions, polynomial equations, and inner products. Cryptology ePrint Archive, Report 2007/404 (2007), http://eprint.iacr.org/
15. Khader, D.: Public key encryption with keyword search based on k-resilient ibe. In: Computational Science and Its Applications - ICCSA 2007, Lecture Notes in Computer Science, vol. 4707, pp. 1086–1095. Springer Berlin / Heidelberg (2007)
16. Kissner, L., Song, D.: Privacy-preserving set operations. In: Advances in Cryptology - CRYPTO 2005, Lecture Notes in Computer Science, vol. 3621, pp. 241–257. Springer Berlin / Heidelberg (2005)
17. Park, D.J., Kim, K., Lee, P.J.: Public key encryption with conjunctive field keyword search. In: Information Security Applications, Lecture Notes in Computer Science, vol. 3325, pp. 73–86. Springer Berlin / Heidelberg (2005)
18. Shen, E., Shi, E., Waters, B.: Predicate privacy in encryption systems. In: Reingold, O. (ed.) Theory of Cryptography, Lecture Notes in Computer Science, vol. 5444, pp. 457–473. Springer Berlin / Heidelberg (2009)
19. Shi, E., Bethencourt, J., Chan, T.H., Song, D., Perrig, A.: Multi-dimensional range query over encrypted data. pp. 350 –364 (may 2007)
20. Tang, Q.: Revisit the concept of peks: Problems and a possible solution. Technical Report TR-CTIT-08-54, Enschede (2008)
21. Tang, Q., Chen, L.: Public-key encryption with registered keyword search. In: Sixth European Workshop on Public Key Services, Applications and Infrastructures. Lecture Notes in Computer Science, Springer Verlag, London (September 2009), http://doc.utwente.nl/67563/