cryptography

Article

Functional Encryption for Pattern Matching with a Hidden String

Jongkil Kim *, Yang-Wai Chow, Willy Susilo, Joonsang Baek and Intae Kim

Institute of Cybersecurity and Cryptology, School of Computing and Information Technology, University of Wollongong, Wollongong, NSW 2522, Australia; caseyc@uow.edu.au (Y.-W.C.); wsusilo@uow.edu.au (W.S.); baek@uow.edu.au (J.B.); intaekim@uow.edu.au (I.K.)
* Correspondence: jongkil@uow.edu.au

Abstract: We propose a new functional encryption for pattern matching scheme with a hidden string. In functional encryption for pattern matching (FEPM), access to a message is controlled by its description and a private key that is used to evaluate the description for decryption. In particular, the description with which the ciphertext is associated is an arbitrary string w and the ciphertext can only be decrypted if its description matches the predicate of a private key which is also a string. Therefore, it provides fine-grained access control through pattern matching alone. Unlike related schemes in the literature, our scheme hides the description that the ciphertext is associated with. In many practical scenarios, the description of the ciphertext cannot be public information as an attacker may abuse the message description to identify the data owner or classify the target ciphertext before decrypting it. Moreover, some data owners may not agree to reveal any ciphertext information since it simply gives greater advantage to the adversary. In this paper, we introduce the first FEPM scheme with a hidden string, such that the adversary cannot get any information about the ciphertext from its description. The security of our scheme is formally analyzed. The proposed scheme provides both confidentiality and anonymity while maintaining its expressiveness. We prove these security properties under the interactive general Diffie–Hellman assumption (i-GDH) and a static assumption introduced in this paper.

Keywords: functional encryption; pattern matching system; searchable encryption

Citation: Kim, J.; Chow, Y.-W.; Susilo, W.; Baek, J.; Kim, I. Functional Encryption for Pattern Matching with a Hidden String. *Cryptography* **2022**, *6*, 1. https://doi.org/10.3390/cryptography6010001

Academic Editor: Carlo Blundo

Received: 11 November 2021
Accepted: 24 December 2021
Published: 31 December 2021

1. Introduction

Functional encryption was introduced to provide fine-grained access control for sensitive data. In particular, functional encryption for regular language (FERL) [1] was proposed by Waters et al. In FERL, a private key is associated with Deterministic Finite Automata (DFA) and a ciphertext is associated with a string. Access to an encrypted message is controlled by a descriptive string, such as a sentence or a genetic sequence. If the string associated with the ciphertext is satisfied by the DFA of a private key, the one holding the private key can decrypt the ciphertext. For example, medical data can be encrypted under the genetic sequence of a patient. A doctor who seeks a specific genetic disease can decrypt detailed medical data of patients by matching his private key, which is based on specific genetic sequences, to the genetic sequence that is used as the description of the ciphertext.

Although FERL is versatile, its usage is limited since the ciphertext description is not hidden. In FERL, the description of the ciphertext must be public information. This restricts the usefulness of FERL, particularly when the ciphertext description is also sensitive. For instance, based on the previous example, the genetic sequences of a patient can also be considered to be sensitive information even though the hospital still wants access control based on genetic sequences as it is a good indicator to triage patients. In this scenario, it would be difficult to use FERL as an encryption algorithm. Therefore, hiding the ciphertext description is of interest in public key encryption. Due to this reason, several public key primitives, including anonymous identity based encryption [2,3], function hiding inner production encryption [4–10], and hidden policy attribute based encryption [11–13], have been proposed to hide all information associated with the ciphertext.

In our paper, we provide functional encryption for pattern matching (FEPM) with a hidden string. Like FERL, FEPM can also be used for fine-grained access control over a string. In our proposed scheme, an arbitrary string is used to control access to the ciphertext. Therefore, it maintains the fine-grained access control that FERL provides. The difference between FERL and FEPM is that a private key is associated with a string, which is the predicate. In particular, the string used to describe a ciphertext is hidden. Hence, it is useful in the more restrictive scenario where the description of the ciphertext is also sensitive.

We propose a new FEPM scheme with a hidden string. Prior to our work, there is no functional encryption scheme that uses a hidden string (natural language) for access control of a ciphertext. Therefore, our scheme makes the following contributions to existing work:

- Our scheme is fully expressive such that it supports any string that describes the ciphertext. In our scheme, another string is used as a predicate, which is associated with a private key. This enables us to select any language, from binary to alphabet letters, for the description and the predicate. For example, we can use a binary string of which characters are simply 0 and 1, and also a string which consists of English alphabets A–Z. Moreover, the predicate can consist of wildcard letters to increase the flexibility of pattern matching.
- In our scheme, the size of the ciphertext only increases linearly with the size of the description, which is an arbitrary string that a ciphertext is associated with. Therefore, it can support a long description, such as a sentence or genetic sequences. Moreover, our scheme does not use Deterministic Finite Automata (DFA) for a private key. Instead, a private key is also associated with a string. Therefore, it does not require evaluation of the description from beginning to end. This may significantly reduce the evaluation time for decryption if the location where two strings, a predicate and a description, match are given.
- We provide formal security proofs of our scheme. In particular, we show that both confidentiality and anonymity of the encrypted message and of a hidden string property hold by providing formal security proofs in our security analysis.

We organize the rest of the paper as follows: We provide important related work in Section 2. We explain the essential preliminary knowledge needed to understand our proposed scheme in Section 3. In Section 4, we introduce a pattern matching system which our work is motivated from. In Section 5, we introduce our FEPM scheme. We prove its security in Section 6 and we conclude our paper in Section 7.

2. Related Work

Functional encryption for regular language (FERL) was introduced by Waters et al. [1]. In the FERL proposed by Waters et al., a private key is associated with Deterministic Finite Automata (DFA) and its ciphertext is associated with a string. If the string in the ciphertext is expressed by DFA of a private key, the key owner can decrypt the message encrypted in the ciphertext. More recently, Attrapadung introduced an adaptively secure FERL [14] using dual system encryption [15]. Although both schemes are quite flexible as access to the encrypted message is controlled by DFA, which can evaluate a regular language and ciphertext associated with a string, there is no scheme for hiding the string that is the description of the ciphertext.

Hidden policy attribute based encryption (ABE) [4–10] was introduced to hide an access policy controlling the access to the encrypted message. Those schemes also hide the description, which is an access policy for the ciphertext. In particular, in a Ciphertext Policy ABE (CP-ABE) scheme, the description consists of attributes that must be well defined. For example, the ciphertext can be associated with two attributes "Computer" AND "Science". A private key must also have exactly the same attributes to match them for decryption. If the private key is based on "Compute" (without r) or "Sci", it cannot be used to decrypt the ciphertext. Similarly, function hiding inner product encryption [11–13] was introduced for hiding the description associated with a ciphertext for inner product encryption.

A searchable encryption with shiftable trapdoor scheme was proposed by Desmoulins et al. [16] as a pattern matching system over encrypted data. The proposed scheme uses a shiftable trapdoor and matches any string to the encrypted data. Therefore, its search method is as flexible as that of our functional encryption for pattern matching (FEPM). More recently, Bkakria et al. [17] proposed a system that improves both efficiency and security. This achieves an anonymous trapdoor such that a string corresponding to a trapdoor is hidden. It also significantly improves the computation for matching keywords. Moreover, Kim et al. [18] showed that its efficiency can be further improved.

Middlebox searchable encryption [19–22] can also be used to detect a keyword from encrypted data. In particular, the schemes from [19–21] allow the proposed systems to match a pattern to the encrypted traffic using only AES encryption, which is fast, and show good matching performance. However, those schemes need to tokenize a message before encrypting it. This significantly increases the size of the ciphertext and prevents flexible search compared to a pattern matching system and FEPM.

Pattern matching systems and middlebox searchable encryption only have a matching algorithm and do not have a decryption algorithm. Therefore, they can only be used for matching keywords to data. In the case of a pattern matching system, decryption can be performed by giving tokens for all alphabet letters. However, it cannot be used for access control as it only detects keywords or decrypts the entire data. Sophisticated access control that provides functional encryption cannot be implemented in those schemes. We compare FEPM to other encryption primitives in Table 1.

Table 1. Comparison with other primitives.

	Function	**Hidden Policy**	**Decryption**	**Flexible Predicate**
FERL [1,14]	DFA	No	Yes	Yes
HP-ABE [4–10]	Boolean policy	Yes	Yes	No
MBSE [19–22]	Matching	Yes	No	No
PA [16–18]	Matching	Yes	No	Yes
FEPM (Ours)	Matching	Yes	Yes	Yes

3. Preliminaries

We introduce important preliminaries that are needed to understand functional encryption for pattern matching (FEPM).

3.1. Bilinear Pairing

Let set $\mathcal{G}$ be a group generator that takes a security parameter λ as input and outputs a description of a bilinear group $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ where p is a prime. $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ are cyclic groups of order p, and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is an efficiently computable non-degenerate bilinear map. We assume that the group operations in $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ as well as the bilinear map e are efficiently computable in polynomial time with respect to λ, and that the group descriptions of $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ include generators of the respective cyclic groups. We call e an asymmetric pairing if $\mathbb{G}_1 \neq \mathbb{G}_2$ and no efficiently computable homomorphism exists between $\mathbb{G}_1$ and $\mathbb{G}_2$, in either direction.

We use the interactive General Diffie–Hellman (i-GDH) assumption [16] to prove our security, which is defined as follows:

Assumption 1. *(i-GDH Assumption) [16]. Let r, s, t, c and k be five positive integers and $R \in \mathbb{F}_p[X_1; \ldots; X_c]^r$, $S \in \mathbb{F}_p[X_1; \ldots; X_c]^s$, and $T \in \mathbb{F}_p[X_1; \ldots; X_c]^t$ be three tuples of multivariate polynomials over $\mathbb{F}_p$.*

Let $\mathcal{O}_R$ (resp. $\mathcal{O}_S$ and $\mathcal{O}_T$) be oracles that on input $\{\{a^{(k)}_{i_1,\ldots,i_c}\}^{d_k}_{i_j=0}\}_k$ add polynomials $\{\sum_{i_1,\ldots,i_c} a^{(k)}_{i_1,\ldots,i_c} \prod_j X_j^{i_j}\}_k$ to R (resp. S and T).

Let $(x_1,\ldots,x_c)$ be a secret vector and q_R (resp. q_S) (resp. q_T) be the number of queries to $\mathcal{O}_R$ (resp. $\mathcal{O}_S$) (resp. $\mathcal{O}_T$). The i-GDH assumption states that given the values $\{g^{R^{(i)}(x_1,\ldots,x_c)}\}^{r+k\cdot q_R}_{i=1}$, $\{\tilde{g}^{S^{(i)}(x_1,\ldots,x_c)}\}^{s+k\cdot q_S}_{i=1}$ and $\{e(g,\tilde{g})^{T^{(i)}(x_1,\ldots,x_c)}\}^{t+k\cdot q_T}_{i=1}$, it is hard to decide whether $U = g^{f(x_1,\ldots,x_c)}$ or U is random if f is independent of $< R,\ S,\ T >$.

Assumption 2. *Let g and h be random generators of $\mathbb{G}_1$ and $\mathbb{G}_2$ and u, v, c and $d_1,\ldots d_n$ be selected randomly from $\mathbb{Z}_p^*$ and T is a random element from $\mathbb{G}_T$. We define*

$$D_0 = (g, h, g^u, h^c, \{g^{d_i}, h^{d_i}, h^{v+c\cdot d_i}\}^n_{i=0}, e(g,h)^b, e(g,h)^{uv})$$

$$D_1 = (g, h, g^u, h^c, \{g^{d_i}, h^{d_i}, h^{v+c\cdot d_i}\}^n_{i=0}, e(g,h)^b, T)$$

Then, there is no PPT algorithm $\mathcal{B}$ that can distinguish D_0 from D_1 with non-negligible advantage. We denote the advantage of $\mathcal{B}$ as

$$Adv^{A2}_{\mathcal{B}} = |\Pr[\mathcal{B}(D_0 = 0)] - \Pr[\mathcal{B}(D_1 = 0)]|.$$

Lemma 1. *D_0 and D_1 in Assumption 2 are indistinguishable in a generic group model.*

Proof. The proof of Lemma 1 is straightforward because the only difference between D_0 and D_1 is whether T is a random element or $e(g,h)^{uv}$. Therefore, we can only distinguish D_1 and D_2 by testing the value of T, Because exponents u and v only appear in g^u and $h^{v+c\cdot d_i}$, T can only be tested by their pairing results, which are $e(g,h)^{uv+u\cdot c\cdot d_i}$. However, the computation of $e(g,h)^{u\cdot c\cdot d_i}$ cannot be computed from the given instances as there is no monomial that has two of those three exponents, either $u\cdot c$, $u\cdot d_i$ or $c\cdot d_i$. Due to this fact, the adversary cannot use $e(g,h)^{uv+u\cdot c\cdot d_i}$ to distinguish $e(g,h)^{uv}$ from T. □

3.2. Definitions

FEPM is defined by four algorithms that we call **Setup**, **Keygen**, **Encrypt** and **Decrypt** as follows:

Setup$(1^\lambda,\ n,\ \mathcal{S}) \rightarrow (pp, msk)$: This algorithm takes as inputs a security parameter k and an integer n defining the maximum size of a string (i.e., a pattern) that one can use as a description associated with a ciphertext. It also takes a finite set $\mathcal{S}$, which is a set of alphabet letters for the description, as input. It sets a master secret key msk and publishes public parameters pp.
KeyGen$(W, msk) \rightarrow sk_W$: This algorithm takes as inputs a string W of any size $0 < \ell_W \leq n$, along with the master secret key, and returns a private key sk_W.
Encrypt$(M, S, pp) \rightarrow CT$: This algorithm takes as inputs a message M and the public parameters pp along with a string $S = s_0\ldots s_{m-1}$ where $m \leq n$, such that $s_i \in \mathcal{S}$ for all $i \in [0, m-1]$, and returns a ciphertext CT which encrypts M.
Decrypt$(CT, sk_W) \rightarrow M$: This deterministic algorithm takes as inputs a ciphertext CT associating to a string $S = s_0\ldots s_{m-1}$ of size m, along with the private key sk_W for a string $W = w_0\ldots w_{\ell_W-1}$ of size ℓ_W. If $\ell_W > m$, then the algorithm returns $\perp$. Otherwise, the algorithm decrypts a ciphertext CT and return M.

Correctness Property. For correctness, the following property must be satisfied:

Let $(pp, msk) \leftarrow$ **Setup**$(1^k, \mathcal{S}, n)$. For a string $W = w_0\ldots w_{\ell_W-1}$ of any size less than or equal to n (i.e., $\ell_W \leq n$), $sk_W \leftarrow$ **KeyGEn**(W, msk) will be returned as a private key. For a string $S = s_0\ldots s_{m-1}$, a ciphertext $CT \leftarrow$ **Encrypt**(M, S, pp) is returned. For a ciphertext CT and a private sk_W, $M \leftarrow$ **Decrypt**(CT, td_W) will be returned if there exists an index j such that $s_j\ldots s_{j+\ell_W-1} = w_0\ldots w_{\ell_W-1}$.

3.3. Security Models

Our proposed FEPM scheme pursues both confidentiality of message (i.e., confidentiality) and hidden predicate (i.e., anonymity). These security properties can be proved by using two different security models. We use sIND-CPA and sANON-CPA, which are defined in this section, to prove confidentiality and anonymity, respectively.

3.3.1. IND-CPA

We first define the indistinguishable chosen plaintext security (IND-CPA) for our FEPM. IND-CPA of FEPM is defined by an experiment $\mathbf{Exp}_{\mathcal{A}_1}^{\text{IND-CPA-}\beta}(1^\lambda, \ell)$ where $\beta \in \{0,1\}$ defined as follows:

- **Setup**: The challenger runs **Setup**$(1^\lambda, n, \mathcal{S})$ to obtain a public parameter pp. It gives $\mathcal{A}_1$ the public parameter pp.
- **Phase I**: The adversary $\mathcal{A}_1$ queries private keys $sk_{W_1}, \dots, sk_{W_{q_1}}$ for strings $W_1, \dots, W_{q_1}$.
- **Challenge**: If **Phase I** is over, $\mathcal{A}_1$ outputs messages M_0 and M_1 with a string $S = s_0 \dots s_{m-1}$ with the restriction that there is no private key queried in **Phase I** to be matched with S. More formally, the challenger outputs $\perp$ if $\exists W = w_0 \dots w_{\ell_W - 1} \in \{W_1, \dots, W_{q_1}\}$ and $\exists i \in \{0, \dots, m - \ell_w\}$ such that
$$s_i \dots s_{i+\ell_W-1} = w_0 \dots w_{\ell_W-1}.$$
The challenger randomly selects $\beta \in \{0,1\}$ and runs **Encrypt** algorithm to obtain $C = \mathbf{Encrypt}(M_\beta, S, pp)$ and returns C to $\mathcal{A}_1$.
- **Phase I**: The adversary $\mathcal{A}_1$ continues to query private keys $sk_{W_{q_1+1}}, \dots, sk_{W_q}$ for strings $W_{q_1+1}, \dots, W_q$ under the same restriction that
$$s_i \dots s_{i+\ell_W-1} \neq w_0 \dots w_{\ell_W-1}.$$
for all $W = w_0 \dots w_{\ell_W-1} \in \{W_{q_1+1}, \dots, W_q\}$ and all $i \in \{0, \dots, m - \ell_w\}$.
- **Guess**: Finally, the adversary $\mathcal{A}_1$ outputs a guess $\beta' \in \{0,1\}$ and wins the game if $\beta = \beta'$.

We define the advantage of an adversary $\mathcal{A}_1$ as follows:
$$Adv_{\mathcal{A}_1}^{\text{IND-CPA}}(1^\lambda, \ell) = |\Pr[\mathbf{Exp}_{\mathcal{A}_1}^{\text{IND-CPA-1}}(1^\lambda, \ell)] - \Pr[\mathbf{Exp}_{\mathcal{A}_1}^{\text{IND-CPA-0}}(1^\lambda, \ell)]|.$$

A functional encryption for pattern matching is IND-CPA secure if this advantage is negligible for any polynomial-time adversary. A weaker notion, which is selective security, sIND-CPA, can be defined with an adversary giving S to the challenger before the challenge gives pp to the adversary in **Setup**.

3.3.2. ANON-CPA

We also define the ANON-CPA security of the functional encryption for pattern matching, namely FEPM-ANON-CPA security.

FEPM-ANON-CPA is defined by an experiment $\mathbf{Exp}_{\mathcal{A}_2}^{\text{ANON-CPA-}\beta}(1^\lambda, \ell)$ the adversary $\mathcal{A}_2$ where $\beta \in \{0,1\}$ defined as follows:

- **Setup**: The challenger runs **Setup**$(1^\lambda, n, \mathcal{S})$ to obtain a public parameter pp. It gives $\mathcal{A}_2$ the public parameter pp.
- **Phase I**: The adversary $\mathcal{A}_2$ queries private keys $sk_{W_1}, \dots, sk_{W_{q_1}}$ for strings $W_1, \dots, W_{q_1}$.
- **Challenge**: If **Phase I** is over, $\mathcal{A}_2$ outputs $S_0 = s_0^{(0)} \dots s_{m-1}^{(0)}$ and $S_1 = s_0^{(1)} \dots s_{m-1}^{(1)}$ with the restriction that there is no trivial trapdoors queried in **Phase I** to distinguish S_0 and S_1. More formally, the challenger outputs $\perp$ if $\exists W = w_0 \dots w_{\ell_W-1} \in \{W_1, \dots, W_q\}$ and i, j such that
$$s_j^{(i)} \dots s_{j+\ell_W-1}^{(i)} = w_0 \dots w_{\ell_W-1} \neq s_j^{(i-1)} \dots s_{j+\ell_W-1}^{(i-1)}.$$

The challenger randomly selects $\beta \in \{0,1\}$ and runs **Encrypt** algorithm to obtain $C = \mathbf{Encrypt}(M, S_\beta, pp)$ and returns C to $\mathcal{A}_2$.

- **Phase II**: The adversary $\mathcal{A}_2$ having accesses to the oracle $\mathcal{O}_{\text{Issue}}$ continue to query private keys $sk_{W_{q_1+1}}, \ldots, sk_{W_q}$ for strings $W_1, \ldots, W_q$ with the same restriction that there is no trivial $W \in \{W_1, \ldots, W_q\}$ to distinguish M_0 and M_1 was queried in **Phase I**.
- **Guess**: Finally, the adversary $\mathcal{A}_2$ outputs a guess $\beta' \in \{0,1\}$ and wins the game if $\beta = \beta'$.

We define the advantage of an adversary $\mathcal{A}_2$ as follows:

$$Adv_{\mathcal{A}_2}^{\text{ANON-CPA}}(1^\lambda, \ell) = |\Pr[\mathbf{Exp}_{\mathcal{A}_2}^{\text{ANON-CPA-1}}(1^\lambda, \ell)] - \Pr[\mathbf{Exp}_{\mathcal{A}_2}^{\text{ANON-CPA-0}}(1^\lambda, \ell)]|.$$

A functional encryption for pattern matching is FEPM-ANON-CPA secure if this advantage is negligible for any polynomial-time adversary.

A weaker notion, which is selective security, FEPM-sANON-CPA, can be defined with an adversary giving S_0 and S_1 to the challenger before the challenge gives pp to the adversary in **Setup**.

4. DFOS's Pattern Matching System

To introduce our FEPM system, we first give an overview of the pattern matching system proposed by Desmoulines et al. [16]. In short, the pattern matching system encrypts a string by each letter so that one who has a valid token to search for a keyword (or a pattern) in the encrypted data can determine if the encrypted data contains a keyword (or a pattern) that matches the token. In particular, the pattern matching system [16] consists of five algorithms: **Setup**, **KeyGen**, **Issue**, **Encrypt** and **Test** as follows:

- $\mathbf{Setup}(1^\lambda, n, \mathcal{S}) \to (pp, msk)$: The algorithm takes as input a security parameter λ and the maximum size of the ciphertext n. It selects $g \overset{R}{\leftarrow} \mathbb{G}_1$ and $h \overset{R}{\leftarrow} \mathbb{G}_2$ and publish $pp = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, h, n)$.
- $\mathbf{KeyGen}(\mathcal{S}) \to (pk, sk)$: It takes as input the set of alphabet letters $\mathcal{S}$. It selects $|\mathcal{S}| + 1$ random values z and $\{\sigma_s\}_{s \in \mathcal{S}}$ from $\mathbb{Z}_p^*$ and sets $g_i \leftarrow g^{z^i}$. It publishes a public key $pk = (\{g_i, \{g_i^{\sigma_s}\}_{s \in \mathcal{S}}\}_{i=0}^{n-1})$ and outputs a secret key $sk = (z, \{\sigma_s\}_{s \in \mathcal{S}})$.
- $\mathbf{Issue}(W, sk) \to sk_W$: To create a token sk_W for a string $W = w_0 w_1 \ldots w_{\ell_W - 1}$ of length $\ell_W (\leq n)$, the algorithm generates $r, r_0, r_1, \ldots, r_{\ell_W - 1}$. For $i \in \{0, \ldots, \ell_W - 1\}$, it sets $K_i = h^{r_i}$ and $K' = h^{\sum_{i=0}^{\ell_w - 1} \sigma_{w_i} \cdot z^i \cdot r_i}$. It sets a private key sk_W for W as $(\{K_i\}_{i \in \{0, \ldots, \ell_W - 1\}}, K')$.
- $\mathbf{Encrypt}(S, pp) \to CT$: To encrypt a string $S = s_0, \ldots, s_{m-1}$, for $i \in \{0, \ldots, m-1\}$, the algorithm selects a random value $a \in \mathbb{Z}_p^*$ and sets $C_i = g_i^a$ and $C_i' = g_i^{a \cdot \sigma_{s_i}}$. It outputs the ciphertext $CT := (\{C_i, C_i'\}_{i=0}^{m-1})$.
- $\mathbf{Test}(pk, td_W, CT) \to \mathcal{J}$: The algorithm takes as inputs a public key pk, a token td_W, and a ciphertext CT. The algorithm sets $\mathcal{J} = \emptyset$. For all $i \in \{0, \ldots, m - \ell\}$, it computes $D_i = e(C_i, K')$ and $E_i = \prod_{j=0}^{\ell_W - 1} e(C_{i+j}', K_j)$. If $D_i = E_i$, it add i to $\mathcal{J}$. Finally, it outputs the set of indexes $\mathcal{J}$.

It should be noted that the above pattern matching system from DFOS is selectively secure. This means that if oracles $\mathcal{O}_{PMS}^{(0)}$ and $\mathcal{O}_{PMS}^{(1)}$ respectively encrypt two strings S_0 and S_1 using the above pattern matching system, $\mathcal{O}_{PMS}^{(0)}$ and $\mathcal{O}_{PMS}^{(1)}$ are indistinguishable for any polynomial time adversary.

Proposition 1. *We let $\mathcal{O}_{PMS}^{(\beta)}$ denote an oracle to simulate a description S_β for $\beta \in \{0,1\}$ using the DFOS pattern matching system. Then, $\mathcal{O}_{PMS}^{(0)}$ and $\mathcal{O}_{PMS}^{(1)}$ are indistinguishable if S_β is given before the system set-up.*

It should be noted that it is straightforward to prove Proposition 1 using the definition of sIND-CPA for the pattern matching system given in [16].

5. Our Construction

In this section, we explain the technique that we used to construct a FEPM scheme. We then provide the construction of our proposed scheme.

5.1. Our Technique

FEPM with a hidden string needs two security proofs: one for confidentiality of the message and the other for anonymity of the ciphertext. Proving both properties in a single encryption system is difficult.

Therefore, we take an idea from a pattern matching system [16,17] which was recently introduced. We observe that a pattern matching system exhibits some similarity to FERL as it evaluates encrypted data using a search token based on alphabet letters. More precisely, the pattern matching system has a trapdoor and a ciphertext, which are associated with a search string and a message, respectively. If a search string matches with a message encrypted in a ciphertext, then, a trapdoor can be used to reveal a location (i.e., an index) where they match in the ciphertext. Thus, it can detect a search string in the message without full decryption of the ciphertext.

However, a pattern matching system is also different from FERL as it does not have a decryption algorithm. As it is designed to only detect a string from the encrypted message, it naturally does not have a decryption algorithm but has a match algorithm. This means that data that does not match the search token will remain encrypted. One of the trivial solutions to decrypt all encrypted data is by giving trapdoors (i.e., search tokens) to all individual alphabet letters. However, this makes detection inefficient as all tokens representing alphabet letters must be matched against each encrypted letter in the ciphertext until one of them matches. Moreover, each letter in a message will be encrypted by multiple group elements to search for a pattern in the whole message. It may not be suitable when a message to be encrypted is long, as several hundred bits are required for each alphabet letter (1 bit for binary or 1 byte for English letter) in the ciphertext. The efficiency of the system will be significantly improved if we can extract some strings, which can be used to control access to a whole message, and use the extracted strings as a description of the ciphertext.

Based on these observations, in our paper, we devise a new functional encryption (FE) scheme that controls access via pattern matching. We consider a message in the pattern match system as a description of a ciphertext in FE. This naturally hides the description of a ciphertext so that it guarantees the anonymity of FERL. Using this idea, we construct FEPM based on one of the simplest pattern matching systems [16] and show that the IND-CPA of the pattern matching system actually implies anonymity, called ANON-CPA, in our proposed FEPM with a hidden string.

Although the hidden string property can be proved relatively easily, proving confidentiality remains demanding. Anonymity in FEPM does not directly imply confidentiality. A ciphertext of FEPM with a hidden string contains two types of information, a description S and a message M, which it aims to hide. Assume that there are two ciphertexts, C_1 and C_2 encrypted under (S_1, M_1) and (S_2, M_2), respectively, where S_1 and S_2 are strings to describe ciphertexts and M_1 and M_2 are messages to be encrypted. S_1 and S_2 have the same length as M_1 and M_2 does so that the C_1 and C_2 cannot be trivially distinguished by the difference of their sizes. Anonymity implies that the adversary cannot distinguish between the two ciphertexts C_1 and C_2 if $M_1 = M_2$ but $S_1 \neq S_2$. However, confidentiality requires that the adversary cannot distinguish between C_1 and C_2 if $M_1 \neq M_2$, but $S_1 = S_2$. Therefore, we need a separate proof for IND-CPA of FEPM. As this proof is not straightforward, we propose a new static assumption, which can be used to prove the security of the proposed FEPM, then show that the security can be reduced to the assumption. We also utilize the strategy that is used to prove anonymity as a part of the confidentiality proof. This implies that we first prove that the adversary cannot distinguish the ciphertext from the original ciphertext even if a string that the ciphertext is associated with is replaced with a random string. We then show that the message of this ciphertext also cannot be distinguished from a ciphertext containing a random message.

5.2. FEPM

In this section, we introduce our FEPM scheme. We use λ and n to denote a security parameter and the maximum size of the description for a ciphertext. It also sets a symmetric encryption $\mathsf{Sym} := (\mathsf{Enc}_{\mathsf{sym}}, \mathsf{Dec}_{\mathsf{sym}})$ and a oneway function $H : \mathbb{G}_T \rightarrow \mathcal{K}$ where $\mathcal{K}$ is a key space of Sym. Our scheme consists of four algorithms, **Setup**, **KeyGen**, **Encrypt** and **Decrypt** as follows:

- **Setup**$(1^\lambda, n, \mathcal{S}) \rightarrow (pp, msk)$: The algorithm takes as input a security parameter λ, the maximum size of the description n and set of alphabet letters $\mathcal{S}$. It selects $g \overset{R}{\leftarrow} \mathbb{G}_1$ and $h \overset{R}{\leftarrow} \mathbb{G}_2$. It selects $|\mathcal{S}| + 2$ random values γ, z and $\{\sigma_s\}_{s\in\mathcal{S}}$ from $\mathbb{Z}_p^*$ and set $g_i \leftarrow g^{z^i}$. It publishes a public parameter:

$$pp = (\mathsf{Enc}_{\mathsf{sym}}, \mathsf{Dec}_{\mathsf{sym}}, H_1, \{e(g_i, h)^\gamma, g_i, \{g_i^{\sigma_s}\}_{s\in\mathcal{S}}\}_{i=0}^{n-1})$$

  and sets a master secret key $msk = (z, \gamma, h, \{\sigma_s\}_{s\in S})$.
- **KeyGen**$(W, msk) \rightarrow sk_W$: To create a token sk_W for a string $W = w_0 w_1 \dots w_{\ell_W - 1}$ of length $\ell_W(\leq n)$, the algorithm generates $r, r_0, r_1, \dots, r_{\ell_W-1}$. For $i \in \{0, \dots, \ell_W - 1\}$, it sets $K_i = h^{r_i}$ and $K' = h^{\gamma + \sum_{i=0}^{\ell_w - 1} \sigma_{w_i} \cdot z^i \cdot r_i}$. It sets the private key sk_W for W as $(\{K_i\}_{i\in\{0,\dots,\ell_W-1\}}, K')$.
- **Encrypt**$(M, S, pp) \rightarrow CT$: To encrypt a string $S = s_0, \dots, s_{m-1}$. It selects random value a and b from $\mathbb{Z}_p^*$ and $K \in \mathcal{K}$. It sets $C = \mathsf{Enc}_{\mathsf{sym}}(K, M)$. For $i \in \{0, \dots, m-1\}$, the algorithm sets $C_i = \mathsf{Enc}_{\mathsf{sym}}(H(e(g_i, h)^{\gamma\cdot a}), K), C_i' = g_i^a, C_i'' = g_i^{a\cdot\sigma_{s_i}}$. It outputs the ciphertext $CT := (C, \{C_i, C_i', C_i''\}_{i=0}^{m-1})$.
- **Decrypt**$(pp, sk_W, CT) \rightarrow M$: The algorithm takes as inputs a public parameter pp, a private key sk_W, and a ciphertext CT. If $\ell_W > m$, it outputs $\perp$. Otherwise, for all $i \in \{0, \dots, m - \ell_W\}$, it computes $D_i = \left(e(C_i', K') / \prod_{j=0}^{\ell_W - 1} e(C_{i+j}'', K_j)\right)$. It then outputs $M = \mathsf{Dec}_{\mathsf{sym}}(\mathsf{Dec}_{\mathsf{sym}}(H(D_i), C_i), C)$.

Correctness. Let W be a substring of S (i.e., $\exists i$ s.t. $s_i \dots s_{i+\ell_W-1} = w_0 \dots w_{\ell_W-1}$). First, one can compute D_i as follows:

$$\begin{aligned}
D_i &= e(C_i', K') / \prod_{j=0}^{\ell_W-1} e(C_{i+j}'', K_j) \\
&= e(g_i^a, h^{\gamma + \sum_{j=0}^{\ell_w-1} \sigma_{w_j} \cdot z^j \cdot r_j}) / \prod_{j=0}^{\ell_W-1} e(g_{i+j}^{a\cdot\sigma_{s_{i+j}}}, h^{r_j}) \\
&= e(g,h)^{a\cdot\gamma\cdot z^i + a\cdot\sum_{j=0}^{\ell_w-1} \sigma_{w_j} \cdot z^{i+j} \cdot r_j} / \prod_{j=0}^{\ell_W-1} e(g,h)^{a\cdot\sigma_{s_{i+j}} \cdot z^{i+j} \cdot r_j} \\
&= e(g,h)^{a\cdot\gamma\cdot z^i} \cdot e(g,h)^{a\cdot\sum_{j=0}^{\ell_w-1} \sigma_{w_j} \cdot z^{i+j} \cdot r_j} / e(g,h)^{a\cdot\sum_{j=0}^{\ell_W-1} \sigma_{s_{i+j}} \cdot z^{i+j} \cdot r_j} \\
&= e(g,h)^{a\cdot\gamma\cdot z^i}.
\end{aligned}$$

The last equality holds as W is a substring of S. Finally, we can compute M as

$$\begin{aligned}
\mathsf{Dec}_{\mathsf{sym}}(\mathsf{Dec}_{\mathsf{sym}}(H(D_i), C_i), C) &= \mathsf{Dec}_{\mathsf{sym}}(\mathsf{Dec}_{\mathsf{sym}}(H(e(g,h)^{a\cdot\gamma\cdot z^i}), C_i), C) \\
&= \mathsf{Dec}_{\mathsf{sym}}(K, C) \\
&= M.
\end{aligned}$$

6. Security Analysis

We formally prove the security of our proposed FEPM scheme.

Theorem 1. *Our FEPM scheme is FEPM-sANON-CPA secure.*

Proof. We prove that the selective security sANON-CPA of FEPM using the security of sIND-CPA of DFOS. More formally, we prove the following claim to show the security of our FEPM scheme:

Claim: Suppose that there is a polynomial time algorithm $\mathcal{A}_1$ breaking sANON-CPA of our FEPM with non-negligible advantage ϵ. Then, we can construct a polynomial time algorithm $\mathcal{B}$ distinguishing between $\mathcal{O}_{PMS}^{(0)}$ and $\mathcal{O}_{PMS}^{(1)}$ (i.e., breaking sIND-CPA of DFOS's pattern matching system with advantage ϵ) using $\mathcal{A}_1$.

Before starting **Setup**, the challenger provides two strings $S^{(0)} = s_0^{(0)} \dots s_{m-1}^{(0)}$ and $S^{(1)} = s_0^{(1)} \dots s_{m-1}^{(1)}$ to $\mathcal{B}$. $\mathcal{B}$ will also set the oracle $\mathcal{O}_{PMS}^{(\beta)}$ by giving $S^{(0)}$ and $S^{(1)}$. $\mathcal{B}$ will simulate sANON-CPA with $\mathcal{A}$ to distinguish between $\mathcal{O}_{PMS}^{(0)}$ and $\mathcal{O}_{PMS}^{(1)}$.

Setup: First, the oracle, $\mathcal{O}_{PMS}^{(\beta)}$, will give a public parameter pp' and a public key pk' to $\mathcal{B}$ where $pp' = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, h, n)$ and $pk' = (\{g_i, \{g_i^{\sigma_s}\}_{s\in\mathcal{S}}\}_{i=0}^{n-1})$. To simulate the Setup algorithm of FEPM with $\mathcal{A}$, $\mathcal{B}$ randomly selects γ from $\mathbb{Z}_p^*$. It computes $\{e(g_i, h)^\gamma\}_{i=0}^{n-1}$. It returns a public parameters $pp = (\{e(g_i, h)^\gamma, g_i, \{g_i^{\sigma_s}\}_{s\in\mathcal{S}}\}_{i=0}^{n-1})$ to $\mathcal{A}_1$.

Phase I and II: When $\mathcal{A}$ requests a key for $W = w_0, \dots w_{\ell_W - 1} \in \{W_1, \dots W_q\}$ to $\mathcal{B}$, first $\mathcal{B}$ check if W is a string that can trivially distinguish $S^{(0)}$ or $S^{(1)}$. It can test it by checking if there exist $\beta \in \{0, 1\}$ and $k \in \{0, \dots, |S^\beta| - \ell_W\}$ such that

$$s_{k-j}^{(\beta)} - s_{k-j+\ell_W-1}^{(\beta)} = w_0 \dots w_{\ell_W-1} \neq s_{k-j}^{(1-\beta)} - s_{k-j+\ell_W-1}^{(1-\beta)}.$$

If β and k exists, it aborts. Otherwise, $\mathcal{B}$ also requests a trapdoor $td_W = (\{K_i\}_{i=0}^{\ell_W-1}, K) = (\{h^{r_i}\}_{i=0}^{\ell_W-1}, h^{\sum_{i=0}^{\ell_W-1} \sigma_{w_i}\cdot z^i \cdot r_i})$ to the oracle running the pattern matching system. It sets $sk_W = (\{K_i\}_{i=0}^{\ell_W-1}, h^\gamma \cdot K)$ and returns sk_W to $\mathcal{A}$.

Challenge: When the challenger requests a ciphertext to $\mathcal{B}$. $\mathcal{B}$ first requests challenge ciphertext to the oracle and receives $\{g_i^a, g_i^{a\cdot\sigma_{s_i^{(\beta)}}}\}$. $\mathcal{B}$ randomly selects $b \in \mathbb{Z}_p^*$, $K \in \mathcal{K}$ and a message $M \in \mathcal{M}$ and sets $C = \mathsf{Enc}_{\mathsf{sym}}(K, M)$, $C_i = \mathsf{Enc}_{\mathsf{sym}}(H(e(g_i^a, h)^\gamma), K)$, $C_i' = g_i^a$ and $C_i'' = g_i^{a\cdot\sigma_{s_i}}$. It sends $CT := (C, \{C_i, C_i', C_i''\}_{i=0}^{m-1})$ to $\mathcal{A}$.

If $\beta = 0$, then $\mathcal{B}$ simulates $\mathcal{O}_{PMS}^{(0)}$ with $\mathcal{A}_1$. Otherwise, $\beta = 1$, it simulates $\mathcal{O}_{PMS}^{(1)}$. Because $\mathcal{A}_1$ have non-negligible advantage ϵ to distinguish β, $\mathcal{B}$ also can distinguish between $\mathcal{O}_{PMS}^{(0)}$ and $\mathcal{O}_{PMS}^{(1)}$ with non-negligible advantage ϵ. $\square$

Theorem 2. *Our FEPM scheme is FEPM-sIND-CPA secure.*

Proof. We will prove the selective security of FEPM by defining security games and showing that they are indistinguishable from $\mathbf{Exp}_{\mathcal{A}}^{\text{sIND-CPA}}(1^\lambda, n)$. First, we set Game_0 to be identical to $\mathbf{Exp}_{\mathcal{A}}^{\text{sIND-CPA}}(1^\lambda, n)$. Then, let $S = s_0, \dots, s_{m-1}$ denote the description for the challenge ciphertext. For all $j \in \{1, \dots, m-1\}$, we define Game_j by switching the first j elements $C_{(\cdot)}''$ (i.e., $C_0'', \dots, C_{j-1}''$) of the challenge ciphertext to random elements of $\mathbb{G}_1$ in Game_0. This allows us to replace all elements C_j'' for all $j \in \{0, \dots, m-1\}$ in the ciphertext on Game_m, which is the last game in the proof, to random values. It means that the adversary $\mathcal{A}$ only has negligible advantage to distinguish between Game_0 and Game_m. Then, we will show that Game_m is indistinguishable from the interim final game $\mathsf{Game}_{final'}$, where the keys (in C_i) encrypting the message encryption key (in C) of symmetric encryption in the challenge ciphertext are replaced by random keys. Finally, we will show that this is equivalent to the final game Game_{final} where the message is replaced by a random message in the proofs. Therefore, the adversary cannot distinguish the message in the challenge ciphertext. $\square$

First, we will show that Game_0 and Game_m are indistinguishable in Lemma 2. Lemmas 3 and 4 will show the indistinguishabilities among Game_m, $\mathsf{Game}_{final'}$ and Game_{final}.

Lemma 2. *For all $j \in \{0, \ldots, m-1\}$, $\mathsf{Game}_{(j)}$ and $\mathsf{Game}_{(j+1)}$ are indistinguishable.*

Proof. We will show that $\mathsf{Game}_{(j)}$ and $\mathsf{Game}_{(j+1)}$ for $j \in \{0, \ldots, m-1\}$ are indistinguishable using the i-GDH assumption. The parameters of the assumption are initially set as follows: $\mathsf{R} = \{(z^i, x_j \cdot z^i, a \cdot z^i)\}_{i=0,j=0}^{2m-1,|\mathcal{S}|-1}, \mathsf{S} = \emptyset, \mathsf{T} = \{z^i, a \cdot z^i\}_{i=0}^{2m-1}$ and $f = a \cdot x_0 \cdot z^{2m-1}$.

It should be noted that the simulator receives a string S before generating any parameters so that this will prove selective security of FEPM.

Setup: From the i-GDH assumption, the following parameters are given along with $U \in \mathbb{G}_1$:

$$\{g^{z^i}, g^{x_j \cdot z^i}, g^{a \cdot z^i}\}_{i=0,j=0}^{2m-1,|\mathcal{S}|-1} \text{ and } \{e(g,h)^a, (e(g,h)^{a \cdot z^i}, e(g,h)^{z^i})_{i=0}^{2m-1}\}.$$

The algorithm $\mathcal{B}$ randomly generates γ from $\mathbb{Z}_p^*$. It, then, generates the public parameters pp as follows:

1. It sets by first defining $g_i = g^{z^{m+i-j^*}}$. This results in $g_{j^*} = g^{z^m}$.
2. It sets $g_i^{\sigma_{s_{j^*}}} = g^{x_0 \cdot z^{m+i-j^*}}$ and $g_i^{\sigma_s} = g^{x_{f(s)} \cdot z^{m+i-j^*}}$ for all $s \in \mathcal{S} \setminus \{s_{j^*}\}$ where a function f be a random permutation from $\forall s \in \mathcal{S} \setminus \{s_{j^*}\}$ to $\{1, \ldots, |\mathcal{S}|-1\}$ (i.e., $f : \mathcal{S} \setminus \{s_{j^*}\} \to \{1, \ldots, |\mathcal{S}|-1\}$).
3. It also sets $(e(g,h)^{z^i})^\gamma$.

The above setting allows the simulator to return pp.

Phase I: Upon receiving a query for a private key with a string $W = w_0 \ldots w_{\ell-1} \in \{W_1, \ldots, W_q\}$, the simulator checks that the string complies with the restriction where there does not exist $j \in \{0, \ldots, m-\ell-1\}$ such that

$$s_{j^*-j}^{(\beta)} - s_{j^*-j+\ell-1}^{(\beta)} = w_0 \ldots w_{\ell-1}.$$

It, then, queries the key to $\mathcal{O}_S$ to receive $\{(h^{r_i})_{i=0}^{\ell-1}, h^{\sum_{i=0}^{\ell w-1} \sigma_{w_i} \cdot z^i \cdot r_i}\}$. It returns a private key $\{(h^{r_i})_{i=0}^{\ell-1}, h^{\gamma + \sum_{i=0}^{\ell-1} \sigma_{w_i} \cdot z^i \cdot r_i}\}$.

Challenge: When the challenger gives M_0 and M_1. Finally, the algorithm $\mathcal{B}$ flip a coin to get $\beta \in \{0,1\}$ and creates the challenge ciphertext as follows:

It selects a random symmetric key $K \in \mathcal{K}$ and sets $C = \mathsf{Enc}_{\mathsf{sym}}(K, M_\beta)$. For $i \in \{0, \ldots, m-1\}$, the algorithm sets $\mathsf{C}_i = \mathsf{Enc}_{\mathsf{sym}}(H(e(g,h)^{z^{m+i-j^*} \cdot a}), K), \mathsf{C}'_i = g^{z^{m+i-j^*} \cdot a}$. For the first j indexes, C''_i are set as random values from $\mathbb{G}_1$. It then uses the $\mathcal{O}_R$ oracle to get valid C''_i for all indexes $i > j^*$. It sets C''_{j^*} as U.

If $U = g^{a \cdot x_0 \cdot z^{2m}}$, then C''_{j^*} is a valid element and the simulator is simulating $\mathsf{Game}_{(j)}$. Otherwise, if C''_{j^*} is a random value from $\mathbb{G}_1$, it is simulating $\mathsf{Game}_{(j+1)}$. This implies that if an adversary $\mathcal{A}$ is able to distinguish $\mathsf{Game}_{(j)}$ from $\mathsf{Game}_{(j+1)}$, it also can break the i-GDH assumption.

Now, we need to show that $f = a \cdot x_0 \cdot z^{2m}$ is independent of the sets R, S and T. This proof is identical with Lemma 5 in [16] except that T is not an empty. In this proof, T includes $\{z^i, a \cdot z^i\}_{i=0}^{2m-1}$. Since those are the exponent of $e(g,h)$, we simply need to show that there are no outputs from $\mathcal{O}_S$ that can be used to distinguish U. This holds obviously as monomials in T do not include x_0 and the outputs of $\mathcal{O}_S$ $(\in G_2)$ also do not have $1/x_0$, which can be taken as input of pairing together with U $(\in \mathbb{G}_1)$ to evaluate U using an element in T $(\in \mathbb{G}_T)$. Therefore, $\{z^i, a \cdot z^i\}_{i=0}^{2m-1}$ in T cannot be used to distinguish U, which has x_0 as an exponent. □

Lemma 3. *Game_m and $\mathsf{Game}_{final'}$ are indistinguishable.*

Proof. Given $\{g, h, g^u, h^c, \{g^{d_i}, h^{d_i}, h^{v+c\cdot d_i}\}_{i=0}^{|\mathcal{S}|}, e(g,h)^v\}$ from Assumption 2, the algorithm $\mathcal{B}$ will simulate either Game_m or $\mathsf{Game}_{final'}$.

Setup: The algorithm $\mathcal{B}$ randomly generates z. It implicitly sets $\gamma = b$ and $\sigma_{s_i} = d_i$ for all $i \in \{0, \ldots, |\mathcal{S}| - 1\}$ where s_i is the ith element in $\mathcal{S}$. It publishes the public parameter:

$$pp = (\{(g_i, h)^\gamma = (e(g,h)^v)^{z^i}, g_i = g^{z^i}, \{g^{\sigma_{s_j}} = (g^{d_j})^{z^i}\}_{j\in\mathcal{S}}\}_{i=0}^{n-1}).$$

Phase I and II: When the adversary requests a private key for $W = w_0, \ldots, w_{\ell_W - 1}$, it randomly generates $r_0, \ldots, r_{\ell_W}$ and sets $K_0 = h^c \cdot h^{r_0}$ and $K_i = h^{r_i}$ for all $i \in \{1, \ldots, \ell_W - 1\}$. It also sets $K' = h^{v+c\cdot\sigma_{w_0}} \cdot (h^c)^{r_0} \cdot \prod_{i=1}^{\ell_W-1} g^{\sigma_{w_i}\cdot z^i \cdot r_i}$.

It should be noted that $\sigma_{w_0} \in d_1, \ldots, d_{|\mathcal{S}|}$ so that $h^{b+c\cdot\sigma_{w_0}}$ is the one of the elements given in the assumption. It sets $sk_W = (\{K_i\}_{i=0}^{\ell_W-1}, h^\gamma \cdot K)$ and returns sk_W to $\mathcal{A}$. This process can be repeated q times for each $W \in \{W_1, \ldots, W_q\}$.

Challenge: The adversary $\mathcal{A}$ requests a challenge ciphertext with M_0 and M_1. To generate ciphertext, first, it randomly select $\beta \in \{0,1\}$. It selects random value $b \in \mathbb{Z}_p^*$ and $\{R_0, \ldots, R_{n-1}\} \in \mathbb{G}_1$ and implicitly sets $a = u$. It, then, sets $C = \mathsf{Enc}_{\mathsf{sym}}(K, M_\beta)$. For $i \in \{0, \ldots, m-1\}$, the algorithm sets $C_i = \mathsf{Enc}_{\mathsf{sym}}(H(T^{z_i}), K), C_i' = (g^u)^{z^i}$, $C_i'' = R_i$. It outputs the ciphertext $CT := (C, \{C_i, C_i', C_i''\}_{i=0}^{m-1})$. It should be noted that if $T = e(g,h)^{uv}$, this simulates Game_m. Otherwise, it simulates $\mathsf{Game}_{final'}$. □

Lemma 4. *$\mathsf{Game}_{final'}$ and Game_{final} are indistinguishable.*

Proof. This holds straightforwardly due to the security of symmetric encryption Sym. Since all keys used in $C_i = \mathsf{Enc}_{\mathsf{sym}}(H(R^{z_i}), K)$ replaced to random keys where R is a random element from $\mathbb{G}_T$ in Game_{final}, the adversary cannot differentiate if K is replaced by $K' \xleftarrow{R} \mathcal{K}$ which is a random key. This means K in $C = \mathsf{Enc}_{\mathsf{sym}}(K, M_\beta)$ does not appear anywhere else. Therefore, the adversary cannot distinguish M_β from a random message $M_R \in \mathcal{M}$. It should be noted that the keys in C_i are correlated to each other because they are all based on the random element $R \in \mathcal{K}$. This is natural in our scheme as the ciphertext is decrypted by decrypting any of C_i, not all C_i. Moreover, generally, symmetric key cryptography provides strong security, which provides enough permutation results even for those correlated keys. □

7. Conclusions

In this paper, we presented new functional encryption for pattern matching scheme with a hidden string. In the proposed scheme, we concealed the description of the ciphertext so that the ciphertext does not need to present any public information related to the message without the loss of expressiveness compared to existing functional encryption schemes. This is extremely useful for the scenario where the description of data used for access control is also sensitive. Moreover, our FEPM is the first scheme that achieves all those practical requirements together. To present formal security proofs for the proposed scheme, we define two security models for anonymity (ANON-CPA) and confidentiality (IND-CPA). We showed that the security of the proposed scheme under these security models. For future work, it would be interesting if can we achieve FEPM under static assumptions. The security of our scheme is proved under the *i-GDH* assumption which is considered to be stronger than static assumptions. Developing a scheme under static assumptions may guarantee better security. Additionally, it would be greatly valuable if FEPM is achieved with a non-pairing group. The proposed version of FEPM needs computationally expensive pairing computations. This computation overhead may become significant if the size of the predicate increases. Therefore, constructing a similar scheme without pairing computations will be beneficial in practice.

Author Contributions: Conceptualization, J.K. and J.B.; formal analysis, J.K. and I.K.; writing—original draft preparation, J.K.; writing—review and editing, Y.-W.C., W.S., J.B. and I.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Waters, B. Functional Encryption for Regular Languages. In *Lecture Notes in Computer Science*; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7417, pp. 218–235.
2. Boneh, D.; Franklin, M.K. Identity-Based Encryption from the Weil Pairing. In *Advances in Cryptology—CRYPTO 2001, Proceedings of the 21st Annual International Cryptology Conference, Santa Barbara, CA, USA, 19–23 August 2001*; Lecture Notes in Computer Science; Kilian, J., Ed.; Springer: Berlin/Heidelberg, Germany, 2001; Volume 2139, pp. 213–229.
3. Boyen, X.; Waters, B. Anonymous Hierarchical Identity-Based Encryption (Without Random Oracles). In *Advances in Cryptology—CRYPTO 2006, Proceedings of the 26th Annual International Cryptology Conference, Santa Barbara, CA, USA, 20–24 August 2006*; Lecture Notes in Computer Science; Dwork, C., Ed.; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4117, pp. 290–307.
4. Lai, J.; Deng, R.H.; Li, Y. Expressive CP-ABE with partially hidden access structures. In Proceedings of the 7th ACM Symposium on Information, Compuer and Communications Security, ASIACCS '12, Seoul, Korea, 2–4 May 2012; Youm, H.Y., Won, Y., Eds.; ACM: New York, NY, USA, 2012; pp. 18–19.
5. Xiong, H.; Zhao, Y.; Peng, L.; Zhang, H.; Yeh, K. Partially policy-hidden attribute-based broadcast encryption with secure delegation in edge computing. *Future Gener. Comput. Syst.* **2019**, *97*, 453–461. [CrossRef]
6. Belguith, S.; Kaaniche, N.; Laurent, M.; Jemai, A.; Attia, R. PHOABE: Securely outsourcing multi-authority attribute based encryption with policy hidden for cloud assisted IoT. *Comput. Netw.* **2018**, *133*, 141–156. [CrossRef]
7. Phuong, T.V.X.; Yang, G.; Susilo, W. Hidden Ciphertext Policy Attribute-Based Encryption Under Standard Assumptions. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 35–45. [CrossRef]
8. Xu, R.; Lang, B. A CP-ABE scheme with hidden policy and its application in cloud computing. *Int. J. Cloud Comput.* **2015**, *4*, 279–298. [CrossRef]
9. Zhou, Z.; Huang, D.; Wang, Z. Efficient Privacy-Preserving Ciphertext-Policy Attribute Based-Encryption and Broadcast Encryption. *IEEE Trans. Comput.* **2015**, *64*, 126–138. [CrossRef]
10. Hao, J.; Huang, C.; Ni, J.; Rong, H.; Xian, M.; Shen, X.S. Fine-grained data access control with attribute-hiding policy for cloud-based IoT. *Comput. Netw.* **2019**, *153*, 1–10. [CrossRef]
11. Bishop, A.; Jain, A.; Kowalczyk, L. Function-Hiding Inner Product Encryption. In *Advances in Cryptology—ASIACRYPT 2015 Part I, Proceedings of the 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, 29 November–3 December 2015*; Lecture Notes in Computer Science; Iwata, T., Cheon, J.H., Eds.; Springer: Berlin/Heidelberg, Germany, 2015; Volume 9452, pp. 470–491.
12. Kim, S.; Lewi, K.; Mandal, A.; Montgomery, H.; Roy, A.; Wu, D.J. Function-Hiding Inner Product Encryption Is Practical. In *Security and Cryptography for Networks, Proceedings of the 11th International Conference, SCN 2018, Amalfi, Italy, 5–7 September 2018*; Lecture Notes in Computer Science; Catalano, D., Prisco, R.D., Eds.; Springer: Berlin/Heidelberg, Germany, 2018; Volume 11035, pp. 544–562.
13. Tomida, J. Tightly Secure Inner Product Functional Encryption: Multi-input and Function-Hiding Constructions. In *Advances in Cryptology—ASIACRYPT 2019, Part III, Proceedings of the 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, 8–12 December 2019*; Lecture Notes in Computer Science; Galbraith, S.D., Moriai, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2019; Volume 11923, pp. 459–488.
14. Attrapadung, N. Dual System Encryption via Doubly Selective Security: Framework, Fully Secure Functional Encryption for Regular Languages, and More. In *Advances in Cryptology—EUROCRYPT 2014, Proceedings of the 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Copenhagen, Denmark, 11–15 May 2014*; Lecture Notes in Computer Science; Nguyen, P.Q., Oswald, E., Eds.; Springer: Berlin/Heidelberg, Germany, 2014; Volume 8441, pp. 557–577.
15. Waters, B. Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. In *Advances in Cryptology—CRYPTO 2009, Proceedings of the 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, 16–20 August 2009*; Lecture Notes in Computer Science; Halevi, S., Ed.; Springer: Berlin/Heidelberg, Germany, 2009; Volume 5677, pp. 619–636.

16. Desmoulins, N.; Fouque, P.; Onete, C.; Sanders, O. Pattern Matching on Encrypted Streams. In *Advances in Cryptology—ASIACRYPT 2018, Part I, Proceedings of the 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, 2–6 December 2018*; Lecture Notes in Computer Science; Peyrin, T., Galbraith, S.D., Eds.; Springer: Berlin/Heidelberg, Germany, 2018; Volume 11272, pp. 121–148.
17. Bkakria, A.; Cuppens, N.; Cuppens, F. Privacy-Preserving Pattern Matching on Encrypted Data. In *Advances in Cryptology—ASIACRYPT 2020, Part II, Proceedings of the 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, Korea, 7–11 December 2020*; Lecture Notes in Computer Science; Moriai, S., Wang, H., Eds.; Springer: Berlin/Heidelberg, Germany, 2020; Volume 12492, pp. 191–220.
18. Kim, J.; Susilo, W.; Chow, Y.W.; Baek, J.; Kim, I. Pattern Matching over Encrypted Data with a Short Ciphertext. In Proceedings of the Information Security Applications—WISA 2021, Jeju, Korea, 11–13 August 2021; Lecture Notes in Computer Science; Kim, H., Ed.; Springer: Berlin/Heidelberg, Germany, 2021; Volume 13009, pp. 132–143.
19. Sherry, J.; Lan, C.; Popa, R.A.; Ratnasamy, S. BlindBox: Deep Packet Inspection over Encrypted Traffic. In Proceedings of the ACM SIGCOMM 2015, London, UK, 17–21 August 2015; ACM: New York, NY, USA, 2015; pp. 213–226.
20. Kim, J.; Camtepe, S.; Baek, J.; Susilo, W.; Pieprzyk, J.; Nepal, S. P2DPI: Practical and Privacy-Preserving Deep Packet Inspection. In Proceedings of the ASIA CCS '21: ACM Asia Conference on Computer and Communications Security, Virtual Event, Hong Kong, China, 7–11 June 2021; Cao, J., Au, M.H., Lin, Z., Yung, M., Eds.; ACM: New York, NY, USA, 2021; pp. 135–146.
21. Ning, J.; Poh, G.S.; Loh, J.; Chia, J.; Chang, E. PrivDPI: Privacy-Preserving Encrypted Traffic Inspection with Reusable Obfuscated Rules. In Proceedings of the 2019 ACMCCS 2019, London, UK, 11–15 November 2019; Cavallaro, L., Kinder, J., Wang, X., Katz, J., Eds.; ACM: New York, NY, USA, 2019; pp. 1657–1670.
22. Canard, S.; Diop, A.; Kheir, N.; Paindavoine, M.; Sabt, M. BlindIDS: Market-Compliant and Privacy-Friendly Intrusion Detection System over Encrypted Traffic. In Proceedings of the ACM AsiaCCS 2017, Abu Dhabi, United Arab Emirates, 2–6 April 2017; ACM: New York, NY, USA, 2017; pp. 561–574.