Public-Key Encryption with Fuzzy Keyword Search: A Provably Secure Scheme under Keyword Guessing Attack

Peng Xu and Hai Jin, *Senior Member, IEEE*

Abstract—A lot of interest has been drawn recently into public-key encryption with keyword search (PEKS), which keeps public-key encrypted documents amendable to secure keyword search. However, PEKS resist against keyword guessing attack by assuming that the size of the keyword space is beyond the polynomial level. But this assumption is ineffective in practice. PEKS are insecure under keyword guessing attack. As we observe, the key to defend such attack is to avoid the availability of the exact search trapdoor to adversaries. Accordingly, we compromise the exactness of search trapdoor by mapping at least two different keywords into a fuzzy search trapdoor. We propose a novel concept called public-key encryption with fuzzy keyword search (PEFKS), by which the un-trusted server only obtains the fuzzy search trapdoor instead of the exact search trapdoor, and define its semantic security under chosen keyword attack (SS-CKA) and indistinguishability of keywords under non-adaptively chosen keywords and keyword guessing attack (IK-NCK-KGA). For the keyword space with and without uniform distribution, we respectively present two universal transformations from anonymous identity-based encryption to PEFKS, and prove their SS-CKA and IK-NCK-KGA securities. To our knowledge, PEFKS is the first scheme to resist against keyword guessing attack on condition that the keyword space is not more than the polynomial level.

Index Terms—Public-key encryption with keyword search, keyword guessing attack, public-key encryption with fuzzy keyword search, anonymous identity-based encryption

1 Introduction

Public-key encryption with keyword search (PEKS) [1] is the first keyword searchable encryption based on a probabilistic public key system. It is more convenient to search ciphertexts for multiple users, compared with previous schemes based on a symmetric key system, such as [2, 3, 4, 5]. Figure 1 presents a classic scenario, in which t senders send searchable ciphertexts to the proxy server of the receiver. When we employ PEKS in this scenario, all the senders produce searchable ciphertexts with the public key of the receiver. So PEKS does not need any coordination between the receiver and any sender when the sender first joins in, which is opposite to previous schemes [2, 3, 4, 5]. In addition, PEKS achieves semantic security under an adaptive chosen keyword attack (SS-CKA), which can not be achieved in previous schemes.

So far, all of proposed PEKS schemes and their expansions had proved their SS-CKA security. However under keyword guessing attack (KGA), their provable securities rely on an implicit assumption that the size of keyword space must be beyond the polynomial level. Therefore, any adversary with the limited computational ability fails to exhaustively search the keyword space to guess the correct keyword in PEKS. Nevertheless, we confirm that the implicit assumption is obviously unreasonable:

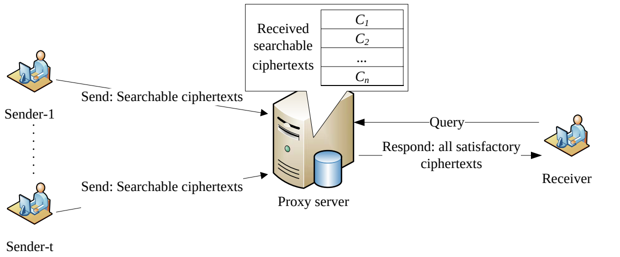
Received searchable ciphertexts
C_1
C_2
...
C_n
Sender-1
Send: Searchable ciphertexts
Query
Respond: all satisfactory ciphertexts
Receiver
Send: Searchable ciphertexts
Proxy server
Sender-t

Fig. 1. A classic scenario with searchable encryption

- In practice, keywords are mostly semantical and indexable in any dictionary. Moreover, some keywords are used in high frequency, such as 'Urgency'. Therefore keywords are non-uniformly employed. In addition, the keyword space should be carefully determined when establishing a cryptosystem [6], because for the keywords having the same meaning, like 'urgent' and 'imperative', a searcher needs different search trapdoors for the same meaning, and the time cost of search will be multiplied. Hence, when determining the keyword space, the basic rule is that its size should not be more than the polynomial level.
- In theory, let we provide a counter example that the size of keyword space is 2^k, and $\mathcal{E}_{K^i}$ denotes

- P. Xu and H. Jin are with Services Computing Technology and System Lab, Cluster and Grid Computing Lab, School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, 430074, China. E-Mail: {xupeng, hjin}@mail.hust.edu.cn.

the keyword K^i has the probability $\mathcal{E}_{K^i}$ to be used. We trivially have

$$\sum_{i=1}^{2^k} \mathcal{E}_{K^i} = 1 \tag{1}$$

Let $Poly()$ denote any polynomial. If all keywords have $\mathcal{E}_{K^i} \geq Poly^{-1}(k)$, it obviously has $\sum_{i=1}^{2^k} \mathcal{E}_{K^i} \gg 1$, which is contrary with Equation 1. So just a part of keywords have $\mathcal{E}_{K^i} \geq Poly^{-1}(k)$. Moreover, the number of this kind of keywords is not more than $Poly(k)$. In other words, the number of keywords having a practical probability to be used is not more than $Poly(k)$, even if the keyword space is the exponential level.

On the practical condition that the size of keyword space is not more than the polynomial level, Byun et al. first proposed keyword guessing attack [7], and attacks a PEKS scheme and a PECKS scheme [8] successfully in 2006. On the same condition, Jeong et al. proved that any PEKS scheme satisfying at least computationally indistinguishable consistency implies successful KGA [9] necessarily. Moreover, since it is necessary to satisfy at least computationally indistinguishable consistency for any efficient PEKS, to defend KGA seems impossible on the practical condition.

1.1 The Motivation

Referring to PEKS, we notice that a search trapdoor of a keyword is necessary for an adversary to implement KGA. Moreover, he can efficiently guesses this keyword when the size of the keyword space is not beyond polynomial level. Therefore, the key to defend KGA is to avoid to leak the the exact search trapdoor of any keyword.

We heuristically conceived an opposite scheme that is public-key encryption without keyword search (called O-PEKS). Obviously, no search trapdoor of any keyword is needed in O-PEKS scheme. Moreover, keywords encrypted using the existing cryptosystem can keep their privacy. Therefore, no adversary can successfully guess any keyword in O-PEKS scheme. In other words, O-PEKS is secure under KGA, but it is not a keyword searchable encryption. When improving O-PEKS to support keyword searchability, it is necessary to provide the search trapdoors of keywords. But when providing the exact search trapdoor of any keyword, O-PEKS also will leak the contents of keywords under KGA. In summary, we either achieve the exact keyword searchability, or maintain the privacy of keywords under KGA. So we were motivated to propose an approach to tradeoff searchability and privacy, such that searchability can be achieved as well as possible without losing the privacy of keywords.

We were motivated to propose a fuzzy keyword search, which may be the first scheme to defend KGA in PEKS as far as we know. In the fuzzy keyword search, adversaries only know the fuzzy search trapdoor of any keyword. So even if under KGA, they just know there are at least two keywords can generate the fuzzy search trapdoor, but they can not deterministically guess which one of them is the correct one. Furthermore, beyond the consideration of cryptosystem, they may have a biased advantage to guess the correct keyword according to the probability distribution of the keyword space. But we will reduce it in our best effort. Formally speaking, we think that a fuzzy keyword search is secure under KGA, if no adversary can distinguish the two keywords, which can generate the same fuzzy search trapdoor. Correspondingly, we formally propose a new security definition called indistinguishability of keywords under non-adaptively chosen keywords and KGA (IK-NCK-KGA).

1.2 IK-NCK-KGA Security

In this subsection, we further discuss IK-NCK-KGA security and illuminate its rationality. In general, the security of a cryptosystem often is a relative concept, because a cryptosystem with perfect security should be inefficient in practice, such as the encryption with one-time key. Therefore the security of a cryptosystem often has several definitions for different strength. For example, indistinguishability under chosen plaintext attack (IND-CPA) [10] is a popular security definition of public-key encryption, and it has a more secure improved version that is indistinguishability under chosen ciphertext attack (IND-CCA) [10].

Analogously, when considering the security of keyword-searchable encryption under KGA, it is intuitive to define the security as indistinguishability of keywords under KGA (IK-KGA). However the security contradicts to the searchability of encryption, and would be un-achievable in theory. Because KGA allows an adversary to know the search trapdoor of a keyword which he wants to challenge, the adversary can distinguish the keyword with any adaptively chosen keyword, which can not be searched by the search trapdoor. By this reason, we restrain the choice of the challenging keywords and propose IK-NCK-KGA security, which will be achieved by our proposed keyword-searchable encryption.

In addition, PEKS is not IK-NCK-KGA secure. An efficient PEKS scheme necessarily satisfies the consistency [11] mentioned in Subsection 2.2. It means that if an adversary knows a search trapdoor of a challenging keyword in PEKS, he can distinguish the keyword with any other keyword. Therefore, IK-NCK-KGA security can not be achieved by PEKS, no matter what NCK defined.

In conclusion, IK-NCK-KGA security is a rational definition. On one hand, a more stronger definition IK-KGA security is contradictive with the searchability of searchable encryption. On the other hand, a

IK-NCK-KGA secure searchable encryption is more secure than PEKS under KGA.

1.3 Our Contributions

Public-Key encryption with fuzzy keyword search (PEFKS) and its IK-NCK-KGA security are novelly defined in our paper. Furthermore, we propose two universal transformations from identity-based encryption (IBE) to PEFKS respectively under different conditions, and prove their SS-CKA and IK-NCK-KGA securities. Specifically, we first present a universal transformation for the keyword space with the uniform distribution (called PEFKS-UD). We also propose an instance of PEFKS-UD based on the anonymous IBE scheme proposed by Boneh in 2001 [12]. Secondly, we propose another universal transformation for the keyword space with the non-uniform distribution (called PEFKS-ND). In addition, we cite two methods to sort keywords, which is the key to realize PEFKS-ND. Beyond the perspective of cryptosystem, we discuss the biased advantage of KGA on PEFKS-ND, which is caused by the non-uniform distribution of the keyword space, and illuminate that we have reduced the biased advantage as much as possible.

1.4 Related Works on PEKS

As Boneh et al. first proposed PEKS, he also proposed a universal transformation from anonymous IBE [1] [12, 13, 14, 15, 16] to PEKS [1]. Hereafter, PEKS has been given a lot of attention. Furthermore, Abdalla et al. completed the foundations of PEKS, presented an improved universal transformation from anonymous IBE to PEKS, and a novel expansion of PEKS, that is public-key encryption with temporary keyword search (PETKS) in 2005. To achieve combinable multi-keyword search, two public-key encryption with conjunctive keyword search (PECKS) schemes [8, 17] were respectively proposed in 2004 and 2007. Conclusively, the aforementioned schemes have a common character that they only succeeded on the equality search, rather than achieved range search and so on. Hence, Bethencourt et al. succeeded on public-key encryption with conjunctive keyword range search [18] by anonymous hierarchical IBE (HIBE) [13] in 2006, and further updated their work in 2007 [19]. In TCC'2007 , Boneh et al. proposed a novel technique called hidden vector encryption (HVE) to achieve conjunctive, range and subset searches [20]. In addition, an improved trapdoor generation of keywords was proposed by Camenisch et al. [21], who employed the committed two-part computation protocol and achieved the invisibility of keyword to generator. The scheme was called public-key encryption with oblivious keyword search (PEOKS). Although several efficiently conjunctive keyword search over encrypted datum were proposed [22, 23], they need to share a secret among users. So they are not convenient for multiple users, compared with aforementioned scheme. Conclusively, all researches on PEKS focused on the various functions of search in recent years rather than its security under KGA.

1.5 Organization

The rest of this paper is organized as follows. In Section 2, some definitions about IBE and PEKS will be given. In Section 3, we will define PEFKS and its SS-CKA and IK-NCK-KGA securities. In Section 4, we will propose PEFKS-UD, prove its SS-CKA and IK-NCK-KGA securities and give an instance of it. In Section 5, we will propose PEFKS-ND, prove its SS-CKA and IK-NCK-KGA securities and discuss the biased advantage of KGA on PEFKS-ND, which is caused by the non-uniform distribution of the keyword space. In Section 7, we will conclude our works and propose an interesting idea for a future work.

2 Preliminaries

Throughout this paper we employ $Poly()$ to denote any polynomial. The symbol $r \overset{\$}{\leftarrow} R$ means randomly choosing r from the space R or outputting r by the randomized algorithm R. (Note that unless stated otherwise, all symbols have the same meaning as when they first appear in this paper.)

2.1 IBE and Anon-ID-CPA Security

IBE [12] and its anonymity under adaptive-ID and chosen plaintext attack (Anon-ID-CPA)[13] are redefined as follows.

Definition 1 (IBE). *An IBE scheme consists of following polynomial time algorithms:*

- $Setup(k, r_1)$: *Take as input a security parameter* k *and a random tape* r_1, *and produce a pair of public-and-private system parameters* $\{Pub_{IBE}, Pri_{IBE}\}$, *in which* Pub_{IBE} *includes the message space* $\mathcal{M}$ *and the identity space* $\mathcal{ID}$.
- $Extract(Pri_{IBE}, r_2, ID)$: *Take as input* Pri_{IBE}, *a random tape* r_2 *and an identity* $ID \in \mathcal{ID}$, *and produce the private key* $PriK_{ID}$ *of* ID.
- $Encrypt(Pub_{IBE}, r_3, ID, M)$: *Take as input* Pub_{IBE}, *an identity* $ID \in \mathcal{ID}$, *a random tape* r_3 *and a message* $M \in \mathcal{M}$, *and produce a ciphertext* C.
- $Decrypt(Pub_{IBE}, PriK_{ID}, C)$: *Take as input* Pub_{IBE}, *a private key* $PriK_{ID}$ *and a ciphertext* C, *and return the decryption result of* C.

[1] The first anonymous IBE scheme was proposed by Boneh et al. in 2001 and proved its security in the random oracle (RO) model [12]. In 2006, Boyen et al. proposed an anonymous IBE scheme [13], which for the first time obtain the provable security in the standard model. In the same year, Gentry proposed the most efficient anonymous IBE scheme in [14]. Ducas proposed an anonymous IBE scheme first based on the asymmetric bilinear map in 2010 [15]. In 2010, Fan et al. proposed the first anonymous multi-receiver IBE [16].

Moreover, it satisfies the consistency that for any ciphertext $C = Encrypt(Pub_{IBE}, r_3, ID', M)$, $M = Decrypt(Pub_{IBE}, PriK_{ID}, C)$ *holds if and only if* $ID = ID'$, *where* $ID \xleftarrow{\$} \mathcal{ID}$.

Definition 2 (Anon-ID-CPA Security of IBE [24, 13]). *An IBE scheme* $IBE = (Setup, Extract, Encrypt, Decrypt)$ *is Anon-ID-CPA secure, if any adversary A, associated to the experiment* $EXP_{IBE,A}^{\text{Anon-ID-CPA-}b}(k)$ *in Figure 2 (where* $b \in \{0,1\}$*), has a negligible advantage* $Adv_{IBE,A}^{\text{Anon-ID-CPA}}$, *where*

$$Adv_{IBE,A}^{\text{Anon-ID-CPA}} = Pr(EXP_{IBE,A}^{\text{Anon-ID-CPA-1}}(k) = 1) - Pr(EXP_{IBE,A}^{\text{Anon-ID-CPA-0}}(k) = 1)$$

```
Experiment $EXP_{IBE,A}^{\text{Anon-ID-CPA-}b}(k)$
    $IDSet \leftarrow \emptyset$; $\{r_1, r_3\} \xleftarrow{\$} \{0,1\}^{Poly(k)}$;
    $\{Pub_{IBE}, Pri_{IBE}\} \leftarrow Setup(k, r_1)$;
    $\{ID^0, ID^1, M, \text{state}\} \xleftarrow{\$} A^{Trap(\cdot)}(\text{find}, Pub_{IBE})$;
    $C \leftarrow Encrypt(Pub_{IBE}, r_3, ID^b, M)$;
    $b' \xleftarrow{\$} A^{Trap(\cdot)}(\text{guess}, C, \text{state})$;
    if $\{ID^0, ID^1\} \bigcap IDSet = \emptyset$ then return $b'$
    else return 0;

Oracle $Trap(ID)$
    $IDSet = IDSet \bigcup \{ID\}$; $r_2 \xleftarrow{\$} \{0,1\}^{Poly(k)}$;
    $PriK_{ID} \leftarrow Extract(Pri_{IBE}, r_2, ID)$;
    return $PriK_{ID}$;
```

Fig. 2. Experiment on Attacking Anon-ID-CPA Security.

2.2 PEKS and Its Insecurity under KGA

We redefine PEKS [1] and illuminate its insecurity under KGA [7] as follows.

Definition 3 (PEKS). *A PEKS scheme consists of following polynomial time algorithms:*

- $SysG(k, r_1)$: *Take as input a security parameter* k *and a random tape* r_1, *and produce a pair of public-and-private system parameters* $\{Pub_{PEKS}, Pri_{PEKS}\}$, *in which* Pub_{PEKS} *includes the keyword space* $\mathcal{K}$.
- $Trapdoor(Pri_{PEKS}, r_2, K)$: *Take as input* Pri_{PEKS}, *a random tape* r_2 *and a keyword* $K \in \mathcal{K}$, *and produce a search trapdoor* T_K.
- $CipherG(Pub_{PEKS}, r_3, K)$: *Take as input* Pub_{PEKS}, *a random tape* r_3 *and a keyword* $K \in \mathcal{K}$, *and produce a searchable ciphertext* C *of* K.
- $ExactTest(Pub_{PEKS}, T_K, C)$: *Take as input* Pub_{PEKS}, *a search trapdoor* T_K *and a searchable ciphertext* $C = CipherG(Pub_{PEKS}, r_3, K')$, *and return*

$$\begin{cases} 1 & \textit{if } K' = K; \\ 0 & \textit{otherwise.} \end{cases}$$

Moreover, it satisfies the consistency that for any keyword searchable ciphertext $C' = CipherG(Pub_{PEKS}, r_3, K')$, $ExactTest(Pub_{PEKS}, T_K, C')$ *returns 1 if and only if* $K = K'$, *where* $K \xleftarrow{\$} \mathcal{K}$.

KGA is a brute force way to guessing the correct keyword. In PEKS, an adverse searcher, who knows the search trapdoors of keywords, can efficiently find out these keywords by KGA on condition that $|\mathcal{K}| \le Poly(k)$ (in which k is the security parameter of PEKS). The details of KGA on PEKS is defined as follows:

Definition 4 (KGA on PEKS). *Given a public system parameters* Pub_{PEKS} *and a valid search trapdoor* T_K *of a keyword* K, *an adversary indexes all keywords in the keyword space* $\mathcal{K}$ *as* $\{K^1, K^2, \ldots, K^{|\mathcal{K}|}\}$ *and implements keyword guessing attack as follows:*

1) *Let* $i = 1$.
2) *Generate a keyword searchable ciphertext* C *of* K^i, *where* $C = CipherG(Pub_{PEKS}, r_3, K^i)$.
3) *If* $ExactTest(Pub_{PEKS}, T_K, C) = 1$, *then return* K^i;
4) *If* $i \ne |\mathcal{K}|$, *compute* $i = i + 1$ *and go to step 2; otherwise it returns* '$\perp$' *(it means the abortion).*

Insecurity of PEKS under KGA According to the consistency of PEKS [2], the output K^i of the adversary equals to K with at most negligible error probability. In addition, when $|\mathcal{K}| \le Ploy(k)$, the adversary can efficiently and exhaustively search all keywords. Consequently, the adversary can deterministically guess the correct keyword, which is used to generate the search trapdoor.

3 PEFKS

In this section, we define PEFKS and its SS-CKA and IK-NCK-KGA security.

3.1 The Definition of PEFKS

PEFKS novelly contains two test algorithms: the fuzzy test algorithm $FuzzTest$ and the exact test algorithm $ExactTest$, which is the main difference with PEKS. When receiving a query, $FuzzTest$ is used to filter out most of ineffective keyword searchable ciphertexts. But the remainders still contain the keyword searchable ciphertexts which do not satisfy the query. $ExactTest$ is used to find out the correct keyword searchable ciphertexts from these remainders. In practice, a proxy server implements $FuzzTest$ to respond the query of a receiver. The receiver implements $ExactTest$ to find out the correct searchable ciphertexts from the responses.

Definition 5 (PEFKS). *PEFKS consists of following polynomial time algorithms:*

- $SysG(k, r_1)$: *Take as input a security parameter* k *and a random tape* r_1, *and produce a pair of*

[2] The consistency is computationally indistinguishable at least [11].

public-and-private system parameters $\{Pub_{PEFKS}, Pri_{PEFKS}\}$, in which Pub_{PEFKS} includes the keyword space $\mathcal{K}$ and a deterministic function $Fuz(K, \mathcal{K})$. $Fuz(K, \mathcal{K})$ deterministically returns a fuzzy value for an inputted keyword K and there are at least two different keywords have the same fuzzy value.

- *$DTrapdoor(Pri_{PEFKS}, r_2, K)$: Take as input Pri_{PEFKS}, a random tapes r_2, and a keyword $K \in \mathcal{K}$, and produce a fuzzy search trapdoor FT_K for the fuzzy value $FK = Fuz(K, \mathcal{K})$, and an exact search trapdoor ET_K for K.*
- *$CipherG(Pub_{PEFKS}, r_3, K)$: Take as input Pub_{PEFKS}, a random tape r_3 and a keyword $K \in \mathcal{K}$, and produce a fuzzy keyword searchable ciphertext C of K.*
- *$FuzzTest(Pub_{PEFKS}, FT_K, C)$: Take as input Pub_{PEFKS}, a fuzzy search trapdoor FT_K of the keyword K and a fuzzy keyword searchable ciphertext $C = CipherG(Pub_{PEFKS}, r_3, K')$, and return*

$$\begin{cases} 1 & \textit{if } Fuz(K', \mathcal{K}) = Fuz(K, \mathcal{K}); \\ 0 & \textit{otherwise.} \end{cases}$$

- *$ExactTest(Pub_{PEFKS}, ET_K, C)$: Take as input a public system parameters Pub_{PEFKS}, an exact search trapdoor ET_K of the keyword K and a fuzzy keyword searchable ciphertext $C = CipherG(Pub_{PEFKS}, r_3, K')$, and return*

$$\begin{cases} 1 & \textit{if } K' = K; \\ 0 & \textit{otherwise.} \end{cases}$$

Moreover, for any fuzzy keyword searchable ciphertext $C' = CipherG(Pub_{PEFKS}, r_3, K')$, it satisfies the following consistencies:

1) *$FuzzTest(Pub_{PEFKS}, FT_K, C')$ returns 1, if and only if $Fuz(K', \mathcal{K}) = Fuz(K, \mathcal{K})$;*
2) *$ExactTest(Pub_{PEFKS}, ET_K, C')$ returns 1, if and only if $K = K'$.*

3.2 SS-CKA and IK-NCK-KGA security definitions of PEFKS

SS-CKA security is a conventional security definition of PEKS, and used to prove the privacy of keywords under chosen keyword attack. Referring to PEFKS, we define the SS-CKA security of PEFKS as follows:

Definition 6 (SS-CKA Security of PEFKS). *A PEFKS scheme $PEFKS = (SysG, DTrapdoor, CipherG, FuzzTest, ExactTest)$ is SS-CKA secure, if any adversary A, associated to the experiment $EXP^{\text{SS-CKA-}b}_{PEFKS,A}(k)$ in Figure 3 (where $b \overset{\$}{\leftarrow} \{0, 1\}$), has a negligible advantage $Adv^{\text{SS-CKA}}_{PEFKS,A}$, where*

$$\begin{aligned} Adv^{\text{SS-CKA}}_{PEFKS,A} = & Pr(EXP^{\text{SS-CKA-1}}_{PEFKS,A}(k) = 1) \\ & - Pr(EXP^{\text{SS-CKA-0}}_{PEFKS,A}(k) = 1) \end{aligned}$$

```
Experiment EXP$^{\text{SS-CKA-}b}_{PEFKS,A}(k)$
    $KSet \leftarrow \emptyset$; $\{r_1, r_3\} \overset{\$}{\leftarrow} \{0,1\}^{Poly(k)}$;
    $\{Pub_{PEFKS}, Pri_{PEFKS}\} \leftarrow SysG(k, r_1)$;
    $\{K^0, K^1, \text{state}\} \overset{\$}{\leftarrow} A^{DTrap(\cdot)}(\text{find}, Pub_{PEFKS})$;
    $C \leftarrow CipherG(Pub_{PEFKS}, r_3, K^b)$;
    $b' \overset{\$}{\leftarrow} A^{DTrap(\cdot)}(\text{guess}, C, \text{state})$;
    if $\{K^0, K^1\} \bigcap KSet = \emptyset$ then return $b'$
    else return 0;

Oracle $DTrap(K)$
    $KSet = KSet \bigcup \{K\}$; $r_2 \overset{\$}{\leftarrow} \{0,1\}^{Poly(k)}$;
    $\{FT_K, ET_K\} \overset{\$}{\leftarrow} DTrapdoor(Pri_{PEFKS}, r_2, K)$;
    return $\{FT_K, ET_K\}$;
```

Fig. 3. Experiment on Attacking SS-CKA Security.

SS-CKA security of PEFKS defines the indistinguishability of any two keywords under chosen keyword attack. It does not consider whether these two keywords have the same output of $Fuz()$. In contrast, IK-NCK-KGA security of PEFKS defines the indistinguishability of the two keywords, which have the same output of $Fuz()$, under KGA. So these two security definitions are different, and do not have any trivial relationship.

Definition 7 (IK-NCK-KGA Security of PEFKS). *A PEFKS scheme $PEFKS = (SysG, DTrapdoor, CipherG, FuzzTest, ExactTest)$ is IK-NCK-KGA secure, if any adversary A, associated to the experiment $EXP^{\text{IK-NCK-KGA-}b}_{PEFKS,A}(k)$ in Figure 4 (where $b \overset{\$}{\leftarrow} \{0, 1\}$), has a negligible advantage $Adv^{\text{IK-NCK-KGA}}_{PEFKS,A}$, where*

$$\begin{aligned} Adv^{\text{IK-NCK-KGA}}_{PEFKS,A} = & Pr(EXP^{\text{IK-NCK-KGA-1}}_{PEFKS,A}(k) = 1) \\ & - Pr(EXP^{\text{IK-NCK-KGA-0}}_{PEFKS,A}(k) = 1) \end{aligned}$$

```
Experiment EXP$^{\text{IK-NCK-KGA-}b}_{PEFKS,A}(k)$
    $KSet \leftarrow \emptyset$; $\{r_1, r_3\} \overset{\$}{\leftarrow} \{0,1\}^{Poly(k)}$;
    $\{Pub_{PEFKS}, Pri_{PEFKS}\} \leftarrow SysG(k, r_1)$;
    $\{K^0, K^1, \text{state}\} \overset{\$}{\leftarrow} A^{DTrap(\cdot), FTrap(\cdot)}(\text{find}, Pub_{PEFKS})$;
    $C \leftarrow CipherG(Pub_{PEFKS}, r_3, K^b)$;
    $b' \overset{\$}{\leftarrow} A^{DTrap(\cdot), FTrap(\cdot)}(\text{guess}, C, \text{state})$;
    if $\{K^0, K^1\} \bigcap KSet = \emptyset$ and $Fuz(K^0, \mathcal{K}) = Fuz(K^1, \mathcal{K})$ then return $b'$
    else return 0;

Oracle $DTrap(K)$
    $KSet = KSet \bigcup \{K\}$; $r_2 \overset{\$}{\leftarrow} \{0,1\}^{Poly(k)}$;
    $\{FT_K, ET_K\} \overset{\$}{\leftarrow} DTrapdoor(Pri_{PEFKS}, r_2, K)$;
    return $\{FT_K, ET_K\}$;

Oracle $FTrap(K)$
    $r_2 \overset{\$}{\leftarrow} \{0,1\}^{Poly(k)}$;
    $\{FT_K, ET_K\} \overset{\$}{\leftarrow} DTrapdoor(Pri_{PEFKS}, r_2, K)$;
    return $FT_K$;
```

Fig. 4. Experiment on Attacking IK-NCK-KGA Security.

In next section, we will proposed two universal transformations from IBE to PEFKS, when the keyword space of PEFKS respectively has uniform and non-uniform distributions.

4 PEFKS-UD

Let k be a security parameter, $\sum$ be an alphabet and the keyword space $\mathcal{K} = \sum^n$, where $|\mathcal{K}| \leq Ploy(k)$. Moreover, $\mathcal{K}$ has the uniform distribution that each symbol of $\sum$ has the identical or computationally indistinguishable probability to be used. The function $Fuz(K, \mathcal{K})$ of PEFKS-UD works as follows:

1) Take as input a keyword $K \in \mathcal{K}$, and parse K as $K = K_1||\ldots||K_n$ where $K_i \in \sum$ for $i \in [1, n]$.
2) Finally return $FK = K_1||\ldots||K_{(n-1)}$.

The complete description of PEFKS-UD is as follows.

4.1 A Universal Transformation from IBE to PEFKS-UD

Let $IBE = (Setup, Extract, Encrypt, Decrypt)$ be an IBE scheme. Let $H_1 : \mathcal{FK} \bigwedge \mathcal{K} \rightarrow \mathcal{ID}$ be a collision resistant function. PEFKS-UD consists of following algorithms:

- $SysG(k, r_1)$: Take as input a security parameter k and a random tape r_1, and run algorithm $Setup(k, r_1)$ of IBE to generate a pair of public-and-private system parameters that

$$Pub_{PEFKS\text{-}UD} = \langle Pub_{IBE}, Fuz, H_1, \mathcal{K} \rangle$$
$$Pri_{PEFKS\text{-}UD} = Pri_{IBE}$$

- $DTrapdoor(Pri_{PEFKS\text{-}UD}, r_2, r_2', K)$: Take as input $Pri_{PEFKS\text{-}UD}$, two random tapes r_2 and r_2', and a keyword $K \in \mathcal{K}$, and generate a fuzzy search trapdoor FT_K and an exact search trapdoor ET_K, where

$$FT_K = Extract(Pri_{IBE}, r_2', H_1(Fuz(K, \mathcal{K})))$$
$$ET_K = Extract(Pri_{IBE}, r_2, H_1(K))$$

- $CipherG(Pub_{PEFKS\text{-}UD}, r_3, r_3', K)$: Take as input $Pub_{PEFKS\text{-}UD}$, two random tapes r_3 and r_3', and a keyword $K \in \mathcal{K}$, choose $M \overset{\$}{\leftarrow} \mathcal{M}$, and generate a fuzzy keyword searchable ciphertext $\langle M, C_F, C_E \rangle$, where

$$C_F = Encrypt(Pub_{IBE}, r_3', H_1(Fuz(K, \mathcal{K})), M)$$
$$C_E = Encrypt(Pub_{IBE}, r_3, H_1(K), M)$$

- $FuzzTest(Pub_{PEFKS\text{-}UD}, FT_K, \langle M, C_F, C_E \rangle)$: Take as input $Pub_{PEFKS\text{-}UD}$, a fuzzy search trapdoor FT_K and a fuzzy keyword searchable ciphertext $\langle M, C_F, C_E \rangle$, and return

$$\begin{cases} 1 & \text{if } M = Decrypt(Pub_{IBE}, FT_K, C_F); \\ 0 & \text{otherwise.} \end{cases}$$

- $ExactTest(Pub_{PEFKS\text{-}UD}, ET_K, \langle M, C_F, C_E \rangle)$: Take as input $Pub_{PEFKS\text{-}UD}$, an exact search trapdoor ET_K and a fuzzy keyword searchable ciphertext $\langle M, C_F, C_E \rangle$, and return

$$\begin{cases} 1 & \text{if } M = Decrypt(Pub_{IBE}, ET_K, C_E); \\ 0 & \text{otherwise.} \end{cases}$$

The consistency of PEFKS-UD According to Theorem 4.2 in [11], it is easy to find that PEFKS-UD is consistent, when the anonymous IBE scheme IBE satisfies the semantic security.

4.2 Provable SS-CKA Security of PEFKS-UD

Theorem 1. *For a PEFKS-UD scheme $PEFKS\text{-}UD = (SysG, DTrapdoor, CipherG, FuzzTest, ExactTest)$ based on an IBE scheme $IBE = (Setup, Extract, Encrypt, Decrypt)$, if IBE is Anon-ID-CPA secure, then $PEFKS\text{-}UD$ is SS-CKA secure.*

Proof: Assuming that there is an adversary A with non-negligible advantage $\mathrm{Adv}^{\text{SS-CKA}}_{PEFKS\text{-}UD,A}$ to break $PEFKS\text{-}UD$, we prove this theorem by inducing a contradiction with the Anon-ID-CPA securiry of IBE. Therefore, we construct an adversary B who employs adversary A to break the Anon-ID-CPA security of IBE.

```
Adversary $B^{Trap(\cdot)}(\text{find}, Pub_{IBE})$
    configure $Fuz, H_1, \mathcal{K}$;
    $Pub_{PEFKS\text{-}UD} = \langle Pub_{IBE}, Fuz, H_1, \mathcal{K} \rangle$;
    $\{K^0, K^1, \text{state}\} \overset{\$}{\leftarrow} A^{DTrap(\cdot)}(\text{find}, Pub_{PEFKS\text{-}UD})$;
    $M \overset{\$}{\leftarrow} \mathcal{M}$; $ID^0 = H_1(K^0)$, $ID^1 = H_1(K^1)$;
    return $\{ID^0, ID^1, M, \text{state}\}$;

Adversary $B^{Trap(\cdot)}(\text{guess}, C, \text{state}))$
    $C_E = C$; $r_3' \overset{\$}{\leftarrow} \{0,1\}^{Poly(k)}$;
    $C_F = Encrypt(Pub_{IBE}, r_3', H_1(Fuz(K^0, \mathcal{K})), M)$;
    $b' \overset{\$}{\leftarrow} A^{DTrap(\cdot)}(\text{guess}, \langle M, C_F, C_E \rangle, \text{state})$;
    return $b'$;

Oracle $DTrap(K)$
    $FT_K = Trap(H_1(Fuz(K, \mathcal{K})))$;
    $ET_K = Trap(H_1(K))$;
    return $\{FT_K, ET_K\}$;
```

Fig. 5. Adversary B Employs Adversary A of SS-CKA Security of $PEFKS\text{-}UD$ to Break Anon-ID-CPA Security of IBE.

Figure 5 presents the construction of adversary B. In it, Adversary B employs the oracle $Trap(\cdot)$ and the public key Pub_{IBE} of IBE to simulate a PEFKS-UD scheme with $Pub_{PEFKS\text{-}UD} = \langle Pub_{IBE}, Fuz, H_1, \mathcal{K} \rangle$. Moreover, he simulates the challenging ciphertext $\langle M, C_F, C_E \rangle$ of $Pub_{PEFKS\text{-}UD}$ by concatenating the challenging ciphertext C of IBE with his produced $C_F = Encrypt(Pub_{IBE}, r_3', H_1(Fuz(K^0, \mathcal{K})), M)$, in which $C_E = C$. It is obvious that $\langle M, C_F, C_E \rangle$ is effective, if both C_E and C_F were generated by K^0. In other word, $\langle M, C_F, C_E \rangle$ is a real challenging ciphertext of $Pub_{PEFKS\text{-}UD}$ with $\frac{1}{2}$ probability. Except the challenging ciphertext, adversary B simulates $Pub_{PEFKS\text{-}UD}$ as a real PEFKS-UD scheme. Consequently, we have

$$\mathrm{Adv}^{\text{Anon-ID-CPA}}_{IBE,B} \geq \frac{1}{2} \cdot \mathrm{Adv}^{\text{SS-CKA}}_{Pub_{PEFKS\text{-}UD},A}$$

But it contradicts to the Anon-ID-CPA security of IBE, when $\mathrm{Adv}^{\text{SS-CKA}}_{Pub_{PEFKS\text{-}UD},A}$ is non-negligible. So if IBE is Anon-ID-CPA secure, then $Pub_{PEFKS\text{-}UD}$ is SS-CKA secure. □

4.3 Provable IK-NCK-KGA Security of PEFKS-UD

Theorem 2. *For a PEFKS-UD scheme $PEFKS\text{-}UD = (SysG, DTrapdoor, CipherG, FuzzTest, ExactTest)$ based on an IBE scheme $IBE = (Setup, Extract, Encrypt, Decrypt)$, if IBE is Anon-ID-CPA secure, then $PEFKS\text{-}UD$ is IK-NCK-KGA secure.*

Proof: Assuming that there is an adversary A with non-negligible advantage $\mathrm{Adv}^{\text{IK-NCK-KGA}}_{PEFKS\text{-}UD,A}$ to break $PEFKS\text{-}UD$, we prove this theorem by inducing a contradiction with the Anon-ID-CPA security of IBE. So we construct an adversary B who employs adversary A to break Anon-ID-CPA security of IBE.

```
Adversary $B^{Trap(\cdot)}(\text{find}, Pub_{IBE})$
    configure $Fuz, H_1, \mathcal{K}$;
    $Pub_{PEFKS\text{-}UD} = \langle Pub_{IBE}, Fuz, H_1, \mathcal{K}\rangle$;
    $\{K^0, K^1, \text{state}\} \xleftarrow{\$} A^{DTrap(\cdot),FTrap(\cdot)}(\text{find}, Pub_{PEFKS\text{-}UD})$;
    $M \xleftarrow{\$} \mathcal{M}$; $ID^0 = H_1(K^0)$, $ID^1 = H_1(K^1)$;
    return $\{ID^0, ID^1, M, \text{state}\}$;

Adversary $B^{Trap(\cdot)}(\text{guess}, C, \text{state}))$
    $C_E = C$; $r'_3 \xleftarrow{\$} \{0,1\}^{Poly(k)}$;
    $C_F = Encrypt(Pub_{IBE}, r'_3, H_1(Fuz(K^0, \mathcal{K})), M)$;
    $b' \xleftarrow{\$} A^{DTrap(\cdot),FTrap(\cdot)}(\text{guess}, \langle M, C_F, C_E\rangle, \text{state})$;
    return $b'$;

Oracle $DTrap(K)$
    $FT_K = Trap(H_1(Fuz(K, \mathcal{K})))$;
    $ET_K = Trap(H_1(K))$;
    return $\{FT_K, ET_K\}$;

Oracle $FTrap(K)$
    $FT_K = Trap(H_1(Fuz(K, \mathcal{K})))$;
    return $FT_K$;
```

Fig. 6. Adversary B Employs Adversary A of IK-NCK-KGA Security of $PEFKS\text{-}UD$ to Break Anon-ID-CPA Security of IBE.

Figure 6 presents the construction of adversary B. In it, Adversary B employs the oracle $Trap(\cdot)$ and the public key Pub_{IBE} of IBE to simulate a PEFKS-UD scheme with $Pub_{PEFKS\text{-}UD} = \langle Pub_{IBE}, Fuz, H_1, \mathcal{K}\rangle$. Moreover he simulates the challenging ciphertext $\langle M, C_F, C_E\rangle$ of $Pub_{PEFKS\text{-}UD}$ by concatenating the challenging ciphertext C of IBE with his produced $C_F = Encrypt(Pub_{IBE}, r'_3, H_1(Fuz(K^0, \mathcal{K})), M)$, in which $C_E = C$. And $\langle M, C_F, C_E\rangle$ is effective, because it has $Fuz(K^0, \mathcal{K}) = Fuz(K^1, \mathcal{K})$ according to the definition of IK-NCK-KGA security. So adversary B simulates $Pub_{PEFKS\text{-}UD}$ as a real PEFKS-UD scheme. Consequently, we have

$$\mathrm{Adv}^{\text{Anon-ID-CPA}}_{IBE,B} \geq \mathrm{Adv}^{\text{IK-NCK-KGA}}_{Pub_{PEFKS\text{-}UD},A}$$

But it contradicts to the Anon-ID-CPA security of IBE, when $\mathrm{Adv}^{\text{IK-NCK-KGA}}_{Pub_{PEFKS\text{-}UD},A}$ is non-negligible. So if IBE is Anon-ID-CPA secure, then $Pub_{PEFKS\text{-}UD}$ is IK-NCK-KGA secure. □

4.4 An Instance of PEFKS-UD

According to the above universal transformation, we construct an instance of PEFKS-UD based on the anonymous IBE scheme BF01 in [12].

Let $\mathbb{G}$ and $\mathbb{G}_t$ denote two multiplicative groups of prime order q, and g be a generator of $\mathbb{G}$. Let the bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_t$ [12, 25, 26] be an efficiently computable and non-degenerate function with the bilinearity that $e(g^a, g^b) = e(g, g)^{ab}$, where $\{a, b\} \xleftarrow{\$} \mathbb{Z}_q^*$. $e(g, g)$ is the generator of $\mathbb{G}_t$. Let the bilinear map generator $BGen(1^k)$ be an efficient algorithm that returns $\langle q, \mathbb{G}, \mathbb{G}_t, g, e\rangle$ with a given security parameter k.

- $SysG(k, r_1)$: It takes as input a security parameter k and a random tape r_1, and works as follows:
  1) Run $BGen(1^k)$ to generate $\langle q, \mathbb{G}, \mathbb{G}_t, g, e\rangle$.
  2) Set $g_{pub} = g^s$, where $s \xleftarrow{\$} \mathbb{Z}_q^*$.
  3) Choose a collision resistance function $H_1 : \sum^{n-1} \bigvee \mathcal{K} \rightarrow \mathbb{G}$.
  4) Choose a pseudo-random function $H_2 : \mathbb{G}_t \rightarrow \mathcal{M}$.
  5) Return a pair of public-and-private system parameter

  $$Pub_{PEFKS\text{-}UD} = \langle q, \mathbb{G}, \mathbb{G}_t, g, e, g_{pub}, Fuz, H_1, H_2, \mathcal{K}, \mathcal{M}\rangle$$

  $$Pri_{PEFKS\text{-}UD} = s$$

  where $\mathcal{K} = \sum^n$ and $\mathcal{M} = \{0, 1\}^{k_1}$.
- $DTrapdoor(Pri_{PEFKS\text{-}UD}, K)$: It takes as input $Pri_{PEFKS\text{-}UD}$ and $K \in \mathcal{K}$, and works as follows:
  1) Compute $g_{FK} = H_1(Fuz(K, \mathcal{K}))$ and the fuzzy search trapdoor $FT_K = g_{FK}^s$.
  2) Compute $g_{EK} = H_1(K)$ and the exact search trapdoor $ET_K = g_{EK}^s$.
  3) Return $\{FT_K, ET_K\}$
- $CipherG(Pub_{PEFKS\text{-}UD}, K)$: It takes as input $Pub_{PEFKS\text{-}UD}$ and $K \in \mathcal{K}$, and works as follows:
  1) Choose $M \xleftarrow{\$} \mathcal{M}$ and $\{t, t'\} \xleftarrow{\$} \mathbb{Z}_q^*$.
  2) Compute $g_{FK} = H_1(Fuz(K, \mathcal{K}))$ and $g_{EK} = H_1(K)$.
  3) Return a fuzzy keyword searchable ciphertext $\langle M, C_F, C_E\rangle$, where

  $$C_F = \langle g^{t'}, H_2(e(g_{FK}, g_{pub})^{t'}) \oplus M\rangle$$

  $$C_E = \langle g^t, H_2(e(g_{EK}, g_{pub})^t) \oplus M\rangle$$

- $FuzzTest(Pub_{PEFKS\text{-}UD}, FT_K, \langle M, C_F, C_E\rangle)$: It takes as input $Pub_{PEFKS\text{-}UD}$, FT_K and $\langle M, C_F, C_E\rangle$, and works as follows:
    1) Parse C_F as $\langle C_{F1}, C_{F2}\rangle$.
    2) Return 1 if $M = C_{F2} \oplus H_2(e(FT_K, C_{F1}))$; otherwise return 0.
- $ExactTest(Pub_{PEFKS\text{-}UD}, ET_K, \langle M, C_F, C_E\rangle)$: It takes as input $Pub_{PEFKS\text{-}UD}$, ET_K and $\langle M, C_F, C_E\rangle$, and works as follows:
    1) Parse C_E as $\langle C_{E1}, C_{E2}\rangle$.
    2) Return 1 if $M = C_{E2} \oplus H_2(e(ET_K, C_{E1}))$; otherwise return 0.

The consistency of the instance of PEFKS-UD According to the IND-ID-CPA (indistinguishability under adaptive-ID and chosen plaintext attacks) security of BF01 scheme, it is easy to deduce that the instance has the computationally indistinguishable consistency.

5 PEFKS-ND

Let k be a security parameter. Let $\mathcal{E}_K$ denote the probability of the event that the keyword $K \in \mathcal{K}$ is employed in practice, where $|\mathcal{K}| \leq Ploy(k)$. We partition all keywords into several subsets as follows:

1) Sort all keywords in descending order of their probabilities and denote them by $\{K^1, K^2, \ldots, K^{|\mathcal{K}|}\}$.
2) Partition $\{K^1, K^2, \ldots, K^{|\mathcal{K}|}\}$ into P, where $P =$
$$\begin{cases} \{\{K^1, K^2\}, \{K^3, K^4\}, \ldots, \{K^{|\mathcal{K}|-1}, K^{|\mathcal{K}|}\}\} \\ \quad if\ |\mathcal{K}|\ is\ even; \\ \{\{K^1, K^2\}, \ldots, \{K^{|\mathcal{K}|-4}, K^{|\mathcal{K}|-3}\}, \{K^{|\mathcal{K}|-2}, K^{|\mathcal{K}|-1}, K^{|\mathcal{K}|}\}\} \\ \quad if\ |\mathcal{K}|\ is\ odd. \end{cases}$$

Let $H_1 : \{0,1\}^* \rightarrow \{0,1\}^{k_2}$ be a collision resistance function. The function $Fuz(K^i, \mathcal{K})$ of PEFKS-ND is redefined as follows:

1) If $|\mathcal{K}|$ is even, then return
$$\begin{cases} H_1(K^{i-1}||K^i) & if\ i\ is\ even; \\ H_1(K^i||K^{i+1}) & if\ i\ is\ odd. \end{cases}$$
2) Otherwise, return
$$\begin{cases} H_1(K^{|\mathcal{K}|-2}||K^{|\mathcal{K}|-1}||K^{|\mathcal{K}|}) & if\ i \geq |\mathcal{K}| - 2; \\ H_1(K^{i-1}||K^i) & if\ i\ is\ even; \\ H_1(K^i||K^{i+1}) & if\ i\ is\ odd. \end{cases}$$

Referring to the function Fuz, we can easily verify the following properties:

- For any subset $\{K^i, K^{i+1}\}$ in P, $Fuz(K^i, \mathcal{K}) = Fuz(K^{i+1}, \mathcal{K})$ holds.
- For any two subsets $\{K^i, K^{i+1}\}$ and $\{K^j, K^{j+1}\}$ in P ($i \neq j$), $Fuz(K^i, \mathcal{K}) \neq Fuz(K^j, \mathcal{K})$ holds by the collision resistance of H_1.

The complete description of PEFKS-ND is as follows.

5.1 A Universal Transformation From IBE to PEFKS-ND

Let $IBE = (Setup, Extract, Encrypt, Decrypt)$ be an IBE scheme. Let $H_2 : \{0,1\}^{k_2} \bigvee \mathcal{K} \rightarrow \mathcal{ID}$ be a collision resistant function. PEFKS-ND consists of following polynomial time algorithms:

- $SysG(k, r_1)$: Take as input a security parameter k and a random tape r_1, and run $Setup(k, r_1)$ of IBE to generate a pair of public-and-private system parameters that
$$Pub_{PEFKS\text{-}ND} = \langle Pub_{IBE}, Fuz, H_1, H_2, \mathcal{K}\rangle$$
$$Pri_{PEFKS\text{-}ND} = Pri_{IBE}$$
- $DTrapdoor(Pri_{PEFKS\text{-}ND}, r_2, r_2', K^i)$: Take as input $Pri_{PEFKS\text{-}ND}$, two random tapes r_2 and r_2', and a keyword $K^i \in \mathcal{K}$, and generate a fuzzy search trapdoor FT_{K^i} and an exact search trapdoor ET_{K^i}, where
$$FT_{K^i} = Extract(Pri_{IBE}, r_2', H_2(Fuz(K^i, \mathcal{K}))$$
$$ET_{K^i} = Extract(Pri_{IBE}, r_2, H_2(K^i))$$
- $CipherG(Pub_{PEFKS\text{-}ND}, r_3, r_3', K^i)$: Take as input $Pub_{PEFKS\text{-}ND}$, two random tapes r_3 and r_3' and a keyword $K^i \in \mathcal{K}$, choose a message $M \overset{\$}{\leftarrow} \mathcal{M}$, and generate a fuzzy keyword searchable ciphertext $\langle M, C_F, C_E\rangle$, where
$$C_F = Encrypt(Pub_{IBE}, r_3', H_2(Fuz(K^i, \mathcal{K})), M)$$
$$C_E = Encrypt(Pub_{IBE}, r_3, H_2(K^i), M)$$
- $FuzzTest(Pub_{PEFKS\text{-}ND}, FT_{K^i}, \langle M, C_F, C_E\rangle)$: Take as input $Pub_{PEFKS\text{-}ND}$, a fuzzy search trapdoor FT_{K^i} and a fuzzy keyword searchable ciphertext $\langle M, C_F, C_E\rangle$, and return
$$\begin{cases} 1 & \text{if } M = Decrypt(Pub_{IBE}, FT_{K^i}, C_F); \\ 0 & \text{otherwise.} \end{cases}$$
- $ExactTest(Pub_{PEFKS\text{-}ND}, ET_{K^i}, \langle M, C_F, C_E\rangle)$: Take as input $Pub_{PEFKS\text{-}ND}$, an exact search trapdoor ET_{K^i} and a fuzzy keyword searchable ciphertext $\langle M, C_F, C_E\rangle$, and return
$$\begin{cases} 1 & \text{if } M = Decrypt(Pub_{IBE}, ET_{K^i}, C_E); \\ 0 & \text{otherwise.} \end{cases}$$

The consistency of PEFKS-ND According to Theorem 4.2 in [11], it is easy to find that PEFKS-ND has the consistency, when the IBE scheme IBE satisfies the semantic security.

5.2 How To Evaluate The Probability Distributions of Keywords

Compared with PEFKS-UD, the key to realize PEFKS-ND is how to evaluate the probability distributions or frequencies of keywords. Recently, SubtlexUS [27] project counted more than 60,384 words' frequencies using an improved word frequency measure, and the

No.	States	Results (million)
1	New York	2430
2	California	1750
3	Washington	1440
4	Texas	1370
5	Florida	1200
6	Wisconsin	1190
7	Georgia	930
8	Virginia	760
9	Ohio	715
10	Michigan	672
11	Colorado	638
12	Arizona	610
13	Illinois	597
14	Indiana	545
15	New Jersey	536
16	Pennsylvania	525
17	Oregon	512
18	Kansas	401
19	Hawaii	399
20	Minnesota	382
21	Massachusetts	375
22	Maryland	369
23	Montana	365
24	Tennessee	362
25	North Carolina	361
26	Oklahoma	358
27	Missouri	353
28	Alabama	351
29	Alaska	343
30	Kentucky	324
31	Maine	324
32	Nevada	318
33	Iowa	305
34	Utah	305
35	Connecticut	302
36	Louisiana	294
37	Mississippi	261
38	Arkansas	252
39	South Carolina	251
40	Delaware	230
41	Nebraska	219
42	Idaho	214
43	New Mexico	214
44	Vermont	201
45	New Hampshire	186
46	Rhode Island	179
47	West Virginia	179
48	Wyoming	177
49	North Dakota	156
50	South Carolina	152

TABLE 1
The Search Results of 50 States of America by Internet Search Engine Google.

results can be downloaded on webpage [28]. So it can be used to sort the keywords, which consist of a single word. But it is ineffective for the keywords consisted of more than one word. For this case, we recommend to estimate their frequencies by internet search engine Google (this method was proposed in [29]).

For example, let the keyword space $\mathcal{K}$ of PEFKS-ND consist of 50 states of America. We search them by the exact search of Google, and sort them in descending order of their frequencies in Table 1. So we can partition them as the description of PEFKS-ND, and realize a PEFKS-ND scheme based on an anonymous IBE scheme, such as BF01 scheme [12].

In this subsection, we proposed two basic methods to sort keywords. But they do not consider any special application. So in practice, it is better to choose an improved sort method according to the speciality of applications. For example, when the American government employs PEFKS-ND to provide a public service, and its keyword space also is $\mathcal{K}$, every American people should applies the service with the same probability. So the state names in $\mathcal{K}$ are used with the frequencies linear with their population, and $\mathcal{K}$ should be sorted in descending order of their population in this application.

5.3 Provable SS-CKA Security of PEFKS-ND

Theorem 3. *For a PEFKS-ND scheme $PEFKS\text{-}ND = (SysG, DTrapdoor, CipherG, FuzzTest, ExactTest)$ based on an IBE scheme $IBE = (Setup, Extract, Encrypt, Decrypt)$, if IBE is Anon-ID-CPA secure, then $PEFKS\text{-}ND$ is SS-CKA secure.*

Proof: Assuming that there is an adversary A with non-negligible advantage $\mathrm{Adv}^{\text{SS-CKA}}_{PEFKS\text{-}ND,A}$ to break $PEFKS\text{-}ND$, we prove this theorem by inducing a contradiction with the Anon-ID-CPA security of IBE. So we construct an adversary B who employs adversary A to break Anon-ID-CPA security of IBE.

```
Adversary $B^{Trap(\cdot)}(\text{find}, Pub_{IBE})$
    configure $Fuz, H_1, H_2, \mathcal{K}$;
    $Pub_{PEFKS\text{-}ND} = \langle Pub_{IBE}, Fuz, H_1, H_2, \mathcal{K}\rangle$;
    $\{K^0, K^1, \text{state}\} \xleftarrow{\$} A^{DTrap(\cdot)}(\text{find}, Pub_{PEFKS\text{-}ND})$;
    $M \xleftarrow{\$} \mathcal{M}$; $ID^0 = H_2(K^0)$; $ID^1 = H_2(K^1)$;
    return $\{ID^0, ID^1, M, \text{state}\}$;

Adversary $B^{Trap(\cdot)}(\text{guess}, C, \text{state}))$
    $C_E = C$; $r'_3 \xleftarrow{\$} \{0,1\}^{Poly(k)}$;
    $C_F = Encrypt(Pub_{IBE}, r'_3, H_2(Fuz(K^0, \mathcal{K})), M)$;
    $b' \xleftarrow{\$} A^{DTrap(\cdot)}(\text{guess}, \langle M, C_F, C_E\rangle, \text{state})$;
    return $b'$;

Oracle $DTrap(K)$
    $FT_K = Trap(H_2(Fuz(K, \mathcal{K})))$;
    $ET_K = Trap(H_2(K))$;
    return $\{FT_K, ET_K\}$;

Note: $K^0$ and $K^1$ is not relative to the sort of keywords.
They can denote any two keywords.
```

Fig. 7. Adversary B Employs Adversary A of SS-CKA Security of $PEFKS\text{-}ND$ to Break Anon-ID-CPA Security of IBE.

Figure 7 presents the construction of adversary B. In it, adversary B employs the oracle $Trap(\cdot)$ and the public key Pub_{IBE} of IBE to simulate a PEFKS-ND scheme with $Pub_{PEFKS\text{-}ND} = \langle Pub_{IBE}, Fuz, H_1, H_2, \mathcal{K}\rangle$. Moreover he simulates the challenging ciphertext $\langle M, C_F, C_E\rangle$ of $Pub_{PEFKS\text{-}ND}$ by concatenating the challenging ciphertext C of IBE with his produced $C_F = Encrypt(Pub_{IBE}, r'_3, H_2(Fuz(K^0, \mathcal{K})), M)$, in which $C_E = C$. It is obvious that $\langle M, C_F, C_E\rangle$ is effective, if both C_E and C_F were generated by K^0. In other word, $\langle M, C_F, C_E\rangle$ is a real challenging ciphertext of $Pub_{PEFKS\text{-}ND}$ with $\frac{1}{2}$ probability. Except the challenging ciphertext, adversary B simulates $Pub_{PEFKS\text{-}ND}$ as a real PEFKS-ND scheme. Consequently, we have

$$\mathrm{Adv}^{\text{Anon-ID-CPA}}_{IBE,B} \geq \frac{1}{2} \cdot \mathrm{Adv}^{\text{SS-CKA}}_{Pub_{PEFKS\text{-}ND},A}$$

But it contradicts to the Anon-ID-CPA security of IBE, when $\mathrm{Adv}^{\text{SS-CKA}}_{Pub_{PEFKS\text{-}ND},A}$ is non-negligible. So if IBE is Anon-ID-CPA secure, then $Pub_{PEFKS\text{-}ND}$ is SS-CKA secure. □

5.4 Provable IK-NCK-KGA Security of PEFKS-ND

Theorem 4. *For a PEFKS-ND scheme* $PEFKS\text{-}ND = (SysG, DTrapdoor, CipherG, FuzzTest, ExactTest)$ *based on an IBE scheme* $IBE = (Setup, Extract, Encrypt, Decrypt)$*, if* IBE *is Anon-ID-CPA secure, then* $PEFKS\text{-}ND$ *is IK-NCK-KGA secure.*

Proof: Assuming that there is an adversary A with non-negligible advantage $\mathrm{Adv}^{\text{IK-NCK-KGA}}_{PEFKS\text{-}ND,A}$ to break $PEFKS\text{-}UD$, we prove this theorem by inducing a contradiction with the Anon-ID-CPA security of IBE. So we construct an adversary B who employs adversary A to break Anon-ID-CPA security of IBE.

```
Adversary $B^{Trap(\cdot)}(\text{find}, Pub_{IBE})$
    configure $Fuz, H_1, H_2, \mathcal{K}$;
    $Pub_{PEFKS\text{-}ND} = \langle Pub_{IBE}, Fuz, H_1, H_2, \mathcal{K}\rangle$;
    $\{K^0, K^1, \text{state}\} \xleftarrow{\$} A^{DTrap(\cdot),FTrap(\cdot)}(\text{find}, Pub_{PEFKS\text{-}ND})$;
    $M \xleftarrow{\$} \mathcal{M}$; $ID^0 = H_2(K^0)$, $ID^1 = H_2(K^1)$;
    return $\{ID^0, ID^1, M, \text{state}\}$;

Adversary $B^{Trap(\cdot)}(\text{guess}, C, \text{state}))$
    $C_E = C$; $r_3' \xleftarrow{\$} \{0,1\}^{Poly(k)}$;
    $C_F = Encrypt(Pub_{IBE}, r_3', H_2(Fuz(K^0, \mathcal{K})), M)$;
    $b' \xleftarrow{\$} A^{DTrap(\cdot),FTrap(\cdot)}(\text{guess}, \langle M, C_F, C_E\rangle, \text{state})$;
    return $b'$;

Oracle $DTrap(K)$
    $FT_K = Trap(H_2(Fuz(K, \mathcal{K})))$;
    $ET_K = Trap(H_2(K))$;
    return $\{FT_K, ET_K\}$;

Oracle $FTrap(K)$
    $FT_K = Trap(H_2(Fuz(K, \mathcal{K})))$;
    return $FT_K$;

Note: $K^0$ and $K^1$ is not relative to the sort of keywords.
They can denote any two keywords.
```

Fig. 8. Adversary B Employs Adversary A of IK-NCK-KGA Security of $PEFKS\text{-}ND$ to Break Anon-ID-CPA Security of IBE.

Figure 8 presents the construction of adversary B. In it, Adversary B employs the oracle $Trap(\cdot)$ and the public key Pub_{IBE} of IBE to simulate a PEFKS-ND scheme with $Pub_{PEFKS\text{-}ND} = \langle Pub_{IBE}, Fuz, H_1, H_2, \mathcal{K}\rangle$. Moreover he simulates the challenging ciphertext $\langle M, C_F, C_E\rangle$ of $Pub_{PEFKS\text{-}ND}$ by concatenating the challenging ciphertext C of IBE with his produced $C_F = Encrypt(Pub_{IBE}, r_3', H_2(Fuz(K^0, \mathcal{K})), M)$, in which $C_E = C$. And $\langle M, C_F, C_E\rangle$ is effective, because it has $Fuz(K^0, \mathcal{K}) = Fuz(K^1, \mathcal{K})$ according to the definition of IK-NCK-KGA security. So adversary B simulates $Pub_{PEFKS\text{-}ND}$ as a real PEFKS-ND scheme. Consequently, we have

$$\mathrm{Adv}^{\text{Anon-ID-CPA}}_{IBE,B} \geq \mathrm{Adv}^{\text{IK-NCK-KGA}}_{Pub_{PEFKS\text{-}ND},A}$$

But it contradicts to Anon-ID-CPA security of IBE, when $\mathrm{Adv}^{\text{IK-NCK-KGA}}_{Pub_{PEFKS\text{-}ND},A}$ is non-negligible. So if IBE is Anon-ID-CPA secure, then $Pub_{PEFKS\text{-}ND}$ is IK-NCK-KGA secure. □

5.5 The Essence of Sorting Keywords

Recall that the keyword space $\mathcal{K}$ of PEFKS-ND has non-uniform distribution. It means that the keywords of $\mathcal{K}$ are non-uniformly used. But referring to the definitions of SS-CKA and IK-NCK-KGA securities, we can find that the challenging keywords $\{K^0, K^1\}$ are uniformly chosen to generate the challenging ciphertext. So it seems that these security definitions is not suitable for PEFKS-ND. But the fact is inverse. Roughly speaking, a secure cryptosystem means that no adversary can learn anything from the cryptosystem. So the security of a cryptosystem does not consider what an adversary can learn beyond the cryptosystem, such as the non-uniform distribution of keywords.

Specifically, an adversary can find two candidate keywords under KGA on PEFKS-ND, which are denoted by $\{K^i, K^{i+1}\} \in P$ without loss of generality. And without considering any others, the adversary practically has the biased advantage $|\mathcal{E}_{K^i} - \mathcal{E}_{K^{i+1}}|$ to decide which one of them is the correct one. Moreover, the biased advantage is caused only by the non-uniform distribution of keywords, so it can not be avoided by any keyword searchable encryption. But PEFKS-ND has decreased the biased advantage by sorting and sequentially partitioning keywords. By this method, the candidate keywords K^i and K^{i+1} have the similar distribution as much as possible. Correspondingly, the biased advantage $|\mathcal{E}_{K^i} - \mathcal{E}_{K^{i+1}}|$ can be much decreased.

Partitions	Probability Difference	Partitions	Probability Difference	Partitions	Probability Difference
{1, 2}	0.163	{11, 12}	0.022	{21, 22}	0.008
{17, 18}	0.122	{19, 20}	0.022	{47, 48}	0.006
{7, 8}	0.101	{45, 46}	0.019	{5, 6}	0.004
{13, 14}	0.046	{37, 38}	0.018	{23, 24}	0.004
{39, 40}	0.044	{35, 36}	0.013	{25, 26}	0.004
{9, 10}	0.031	{49, 50}	0.013	{27, 28}	0.003
{43, 44}	0.031	{41, 42}	0.012	{33, 34}	0.000
{29, 30}	0.028	{15, 16}	0.010		
{3, 4}	0.025	{31, 32}	0.009		
Note: all states in Table 1 are denoted by their numbers in this table.					

TABLE 2
The Partitions of The Keyword Space in Table 1 and Their Probability Differences.

For example, we partition the keywords listed in Table 1 according to PEFKS-ND, and compute the biased advantage (probability difference) of each pair of keywords in Table 2. In theory, except {33, 34}, the others have the non-negligibly biased advantage. But all of them are much smaller than 1.

In conclusion, PEFKS-ND not only achieves IK-NCK-KGA security from the side of cryptography, but also decreases the biased advantage under KGA on PEFKS-ND, which is caused only by the non-uniform distribution of keywords. In next subsection, we will illuminate why the biased advantage has been decreased as much as possible.

5.6 The Insecurity of A Further Decrease in The Biased Advantage

At the end of Subsection 5.5, we stated that we have decreased the biased advantage as much as possible. To prove it, we will propose a method, which is the only one method that can further decrease the biased advantage, and illuminate that it is not an secure universal method.

Referring to the partition P of $\mathcal{K}$ at the beginning of this section, we sorted all keywords in descending order of their probabilities and sequentially partitioned them. Therefore, we did not consider to combine a keyword with one of its two neighboring keywords, which has the minimum probability difference with it. When partitioning keywords according to this rule, however this partition method is not an secure universal method. For example, let $\mathcal{K} = \{K^1, K^2, K^3, K^4\}$ denote the sorted keywords in descending order of their probabilities. We sequently combine each keyword with one of its two neighboring keywords, which has the minimum probability difference with them. Assuming $|\mathcal{E}_{K^1} - \mathcal{E}_{K^2}| > |\mathcal{E}_{K^2} - \mathcal{E}_{K^3}|$, therefore we result the partition $\{\{K^1, K^2\}, \{K^2, K^3\}, \{K^3, K^4\}\}$. By implementing KGA on PEFKS-ND, an adversary can easily find out $\{K^1, K^2\}$ when he knows the fuzzy search trapdoor generated by $\{\{K^1, K^2\}$. Furthermore, he can deterministically guess K^1 rather than with any biased advantage, because a fuzzy search trapdoor of K^2 should be generated by $\{K^2, K^3\}$. Hence, if an adversary finds out $\{K^1, K^2\}$ under KGA on PEFKS-ND, he can deterministically decide that the fuzzy search trapdoor was generated by K^1 rather than K^2.

Consequently, this improved method is not a secure universal method. However, it obviously is the only one method that has chance to further decrease the biased advantage. So we think there is not any other secure universal method can further decrease the biased advantage.

6 Performance Comparison

Generally speaking, PEFKS divides a query from a receiver into two processes: first, a proxy server implements a fuzzy keyword search over all its stored ciphertexts, and returns the results to the receiver; secondly, the receiver implements an exact keyword search over these results. In contrast, PEKS only has an exact keyword search over all stored ciphertexts, which is implemented by a proxy server. So in order to achieve IK-NCK-KGA security, PEFKS increases the workload of the receiver and the communication cost. In Table 3, we compare the performance of PEFKS and PEKS when they receive a query.

	The Workload of The Proxy Server	The Cost of Communication	The Workload of The Receiver
PEKS	n	t	0
PEFKS	n	$2t$	$2t$

- Note: the workload and communication cost are denoted by the number of keyword searchable ciphertexts.
- n: the total number of keyword searchable ciphertexts stored in the proxy sever.
- t: the number of keyword searchable ciphertexts satisfied the query of the receiver.

TABLE 3
The Performance Comparison between PEKS and PEFKS.

7 Conclusion and Future Work

In PEKS, a proxy sever, who responds the keyword queries of a receiver, can know the content of keywords by implementing KGA. Moreover, it is efficient under the practical condition that the size of the keyword space is not more than the polynomial level. In order to resist against KGA, we novelly defined public-key encryption with fuzzy keyword search (PEFKS) and its IK-NCK-KGA security. And we proposed two universal transformations from IBE to PEFKS under different conditions. Under the condition that the keyword space has uniform distribution, we proposed a SS-CKA and IK-NCK-KGA secure transformation PEFKS-UD, and provided an instance based on BF01 scheme [12]. Under the condition that the keyword space has non-uniform distribution, we proposed another SS-CKA and IK-NCK-KGA secure transformation PEFKS-ND, and provided two methods to sort keywords, which is the key to realize PEFKS-ND. Beyond the perspective of cryptosystem, we discussed the biased advantage of KGA on PEFKS-ND, which is caused only by the non-uniform distribution of the keyword space. We illuminate that the biased advantage has been decreased as much as possible. So we made PEFKS-ND secure in a broad sense.

The future work Both in PEKS and PEFKS, their searches take time linear in the number of ciphertexts. Hence it is a crucial problem of performance that how to build indexes in PEKS and PEFKS. Moreover, it is more useful to improve the performance of PEFKS than PEKS. But it is a challenging work, because the SS-CKA securities of PEKS and PEFKS are contradictive to establish indexes. So a tradeoff between security and search performance should be a basic idea. Moreover, we think that a dynamic tradeoff controlled by the owner of datum is a promising method.

References

[1] D. Boneh, G. D. Crescenzo, and R. O. et al., "Public key encyrption with keyword search," in *Advances in Cryptology-EUROCRYPT 2004*, ser.

LNCS, C. Cachin and J. Camenisch, Eds., vol. 3027. Springer-Verlag, 2004, pp. 506–522.

[2] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proceedings of the 2000 IEEE Symposium on Security and Privacy*, Berkeley, CA , USA, 2000, pp. 44–55.

[3] E.-J. Goh, "Secure indexes," 2003, http://eprint.iacr.org/2003/216.pdf.

[4] R. Agrawal, J. Kiernan, and R. S. et al., "Order preserving encryption for numeric data," in *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*. Paris, France: ACM, 2004, pp. 563–574.

[5] R. Curtmola, J. Garay, and S. K. et al., "Searchable symmetric encryption: Improved definitions and efficient constructions," in *Proceedings of the 13th ACM conference on Computer and communications security*. ACM, 2006, pp. 79–88.

[6] W. Harrower, "Searching encrypted data," Department of Computing, Imperial College London, Tech. Rep., 2009.

[7] J. W. Byun, H. S. Rhee, and H.-A. P. et al., "Offline keyword guessing attacks on recent keyword search schemes over encrypted data," in *Secure Data Management*, ser. LNCS, W. Jonker and M. Petkovic, Eds., vol. 4165. Springer-Verlag, 2006, pp. 75–83.

[8] D. J. Park, K. Kim, and P. J. Lee, "Public key encryption with conjunctive field keyword search," in *WISA 2004*, ser. LNCS, C. Lim and M. Yung, Eds., vol. 3325. Spring-Verlag, 2004, pp. 73–86.

[9] I. R. Jeong, J. O. Kwon, and D. H. et al., "Constructing peks schemes secure against keyword guessing attacks is possible?" *Computer Communications*, vol. 32, no. 2, pp. 394–396, 2009.

[10] S. Goldwasser and S. Micali, "Probabilistic encyrption," in *Proceedings of the fourteenth annual ACM symposium on Theory of computing*. San Francisco, California, United States: ACM, 1982, pp. 365–377.

[11] M. Abdalla, M. Bellare, and D. C. et al., "Searchable encryption revisited: Consistency properties, relation to anonymous ibe, and extensions," in *Advances in Cryptology-CRYPTO 2005*, ser. LNCS, V. Shoup, Ed., vol. 3621. Santa Barbara, California, United States: Springer-Verlag, 2005, pp. 205–222.

[12] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Advances in Cryptology-CRYPTO 2001*, ser. LNCS, J. Kilian, Ed., vol. 2139. Santa Barbara, California, United States: Springer-Verlag, 2001, pp. 213–239.

[13] X. Boyen and B. Waters, "Anonymous hierarchical identity-based encryption (without random oracles)," in *Advances in Cryptology-CRYPTO 2006*, ser. LNCS, C. Dwork, Ed., vol. 4117. Santa Barbara, California, United States: Springer-Verlag, 2006, pp. 290–307.

[14] C. Gentry, "Practical identity-based encyrption without random oracles," in *Advances in Cryptology-EUROCRYPT 2006*, ser. LNCS, S. Vaudenay, Ed., vol. 4004. Russia: Springer-Verlag, 2006, pp. 445–464.

[15] L. Ducas, "Anonymity from asymmetry: New constructions for anonymous hibe," in *The Cryptographers' Track at the RSA Conference 2010*, ser. LNCS, J. Pieprzyk, Ed., vol. 5985. San Francisco, CA, USA: Springer Berlin, 2010, pp. 148–164.

[16] C.-I. Fan, L.-Y. Huang, and P.-H. Ho, "Anonymous multireceiver identity-based encryption," *IEEE TRANSACTIONS ON COMPUTERS*, vol. 59, no. 9, pp. 1239–1249, 2010.

[17] Y. H. Hwang and P. J. Lee, "Public key encryption with conjunctive keyword search and its extension to a multi-user system," in *Pairing 2007*, ser. LNCS, T. Takagi, Ed., vol. 4575. Springer-Verlag, 2007, pp. 2–22.

[18] J. Bethencourt, T.-H. H. Chan, and A. P. et al., "Anonymous multi-attribute encryption with range query and conditional decryption," Carnegie Mellon University, Tech. Rep. CMU-CS-06-135, 2006.

[19] E. Shi, J. Bethencourt, and T.-H. H. C. et al., "Multi-dimensional range query over encrypted data," Carnegie Mellon University, Tech. Rep. CMU-CS-06-135, 2007.

[20] D. Boneh and B. Waters, "conjunctive, subset, and range queries on ecrypted data," in *proceedings of TCC'07*, ser. LNCS, S. P. Vadhan, Ed., vol. 4392. Springer-Verlag, 2007, pp. 535–554.

[21] J. Camenisch, M. Kohlweiss, and A. R. et al., "Blind and anonymous identity-based encryption and authorised private searches on public key encrypted data," in *Proceedings of the 12th International Conference on Practice and Theory in Public Key Cryptography: PKC '09*, ser. LNCS, vol. 5443. CA: Springer-Verlag, 2009, pp. 196–214.

[22] L. Ballard, S. Kamara, and F. Monrose, "Achieving efficient conjunctive keyword searches over encrypted data," in *ICICS 2005*, ser. LNCS, S. et al., Ed., vol. 3783. Springer-Verlag, 2005, pp. 414–426.

[23] E.-K. Ryu and T. Takagi, "Efficient conjunctive keyword-searchable encryption," in *Advanced Information Networking and Applications Workshops, 2007, AINAW '07. 21st International Conference on*. Niagara Falls, Ontario, Canada: IEEE Computer Society, 2007, pp. 409 – 414.

[24] M. Bellare, A. Boldyreva, and A. D. et al., "Key-privacy in public-key encryption," in *Advances in Cryptology-ASIACRYPT 2001*, ser. LNCS, C. Boyd, Ed., vol. 2248. Australia: Springer-Verlag, 2001, pp. 566–582.

[25] A. J. Menezes, T. Okamoto, and S. A. Vanstone, "Reducing elliptic curve logarithms to log-

arithms in a finite field," *IEEE TRANSACTIONS ON INFORMATION THEORY*, vol. 39, no. 5, pp. 1639–1646, 1993.

[26] G. Frey, M. Muller, and H.-G. Ruck, "The tate pairing and the discrete logarithm applied to elliptic curve cryptosystems," *IEEE TRANSACTIONS ON INFORMATION THEORY*, vol. 45, no. 5, pp. 1717–1719, 1999.

[27] M. Brysbaert and B. New, "Moving beyond kucera and francis: A critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for american english," *Behavior Research Methods*, vol. 49, no. 4, pp. 977–990, 2009.

[28] ——, "Subtlexus: American word frequencies," http://subtlexus.lexique.org/, 2009.

[29] I. V. BLAIR, G. R. URLAND, and J. E. MA, "Using internet search engines to estimate word frequency," *Behav Res Methods Instrum Comput*, vol. 34, no. 2, pp. 286–290, 2002.