Short Integrated PKE+PEKS in Standard Model

Vishal Saraswat[1(✉)] and Rajeev Anand Sahu[2]

[1] R.C. Bose Centre for Cryptology and Security, Indian Statistical Institute, Kolkata, India
vishal_v@isical.ac.in

[2] Département d'Informatique, Université Libre de Bruxelles, Brussels, Belgium
rajeev.sahu@ulb.ac.be

Abstract. At SeCrypt 2015, Buccafurri et al. [BLSS15] presented an integrated public-key encryption (PKE) and public-key encryption with keyword search (PEKS) scheme (PKE+PEKS) whose security relies on the Symmetric eXternal Diffie-Hellman (SXDH) assumption but they did not provide a security proof. We present a construction of PKE+PEKS and prove its security in the standard model under the SXDH assumption. We prove that our scheme is both IND-PKE-CCA secure, that is, it provides message confidentiality against an adaptive chosen ciphertext adversary, and IND-PEKS-CCA secure, that is, it provides keyword privacy against an adaptive chosen ciphertext adversary. Ours is the first secure PKE+PEKS construction to use asymmetric pairings which enable an extremely fast implementation useful for practical applications. Our scheme has much shorter ciphertexts than the scheme in [BLSS15] and all other publicly known PKE+PEKS schemes. Finally, we compare our scheme with other proposed PEKS and integrated PKE+PEKS schemes and provide a relative analysis of various parameters including assumption, security and efficiency.

Keywords: PKE+PEKS · Searchable encryption · Asymmetric pairings (type 3) · Provable security · Standard model · SXDH

1 Introduction

The primary goal in cryptography is message-privacy which is usually achieved by the encryption techniques. In practice, a recipient may wish to filter the messages that come to her inbox based on the message content, or a user may wish to download some encrypted files from a server whose content satisfies certain criterion. In cryptography, this functionality is achieved by searching on the encrypted data (that is, *searchable encryption*). Boneh et al. [BDOP04] introduced a method of searching for certain keyword(s) in data encrypted using public key encryption (PKE) and called it *public key encryption with keyword*

Second author is supported by the Brussels Region INNOVIRIS project SeCloud.

search (PEKS). Since then, applications of PEKS have been realized in various issues such as the design of spam filter, searchable cloud storage, time released encryption (TRE) etc.

The main advantage of this primitive is that it allows one to delegate to a third party the capability of "searching on public key encrypted data" without impacting privacy. Suppose a bank uses a third party cloud service provider (CSP) to facilitate the banking services to its account holders. To prevent fraudulent transactions, the bank must put some checks on the transactions being conducted and needs constant monitoring of the transaction "contents". For example, for a certain account holder with address in the zipcode 20500 in DC, USA, the bank may put checks as

- if the transaction location zipcode is not 20500 but the state is DC and country is USA, a sms alert must be sent to the account holder informing them of the activity outside home zipcode.
- if the transaction location state is not DC but the country is USA, a sms alert must be sent to the account holder informing them of activity outside home state and a red alert must be sent to the bank, and
- if the transaction country is not USA, the transaction must not be processed further until intervention from the bank.

However, to protect the privacy of the users, the CSP must not be able to get any information about the transactions that any of the account holders conduct. Using a PEKS, the bank can enable the CSP with the ability to test whether the zipcode, state and country values are certain values or not and then act accordingly without learning anything else about the transaction. More generally, the bank can specify a few "keywords" that the CSP can search for, but learn nothing else about the transactions.

The primitive of PEKS basically acts as a "search" function on a PKE scheme but does not retrieve any data by itself. So, in practice, a PEKS scheme is always used together with an underlying PKE scheme and such a combination of these two schemes is called *integrated PKE and PEKS* and is denoted as PKE+PEKS.

The generic approaches that simply combine a PKE scheme and a PEKS scheme (as described in [BDOP04]) work as follows. Let (pk_R, sk_R) be a receiver's (public-key, private-key) pair. A sender encrypts a message-keyword pair (M, W) as $C_{\text{M}} = PKE(\text{M}, pk_R) \| C_{\text{W}} = PEKS(\text{W}, pk_R)$. The mail server on receiving the ciphertext $C_{\text{M}} \| C_{\text{W}}$ parses C_{W} and tests it with its trapdoors $t_{\text{W}'}$ and if the result is '`TRUE`', it forwards C_{M} to the receiver, who then decrypts it using its private key sk_R. Note that in such a generic approach, the server acts only on the second component C_{W} and the receiver acts only on the first component C_{M}.

The basic objective of security in an encryption scheme is *data-privacy.* To achieve this property the standard notions like indistinguishability against-chosen-plaintext attack (IND-CPA) and chosen-ciphertext attack (IND-CCA) have been formalized [GM84, BDPR98, BF01]. The latter one is stronger. For the PKE+PEKS scheme the privacy must be achieved for both, the message (that is, data) and the keyword. Hence, the strongest security notion for a PKE+PEKS scheme corresponds to the idea to achieve CCA-security for both the message and the keyword, that is, IND-PKE-CCA and IND-PEKS-CCA.

Further, we must consider the security of the whole system PKE+PEKS rather than that of each of the components PKE and PEKS independently. As pointed out by Baek et al. [BSS06], Zhang and Imai [ZI07] and Abdalla et al. [ABN10], a PKE+PEKS scheme formed from a CCA secure PKE scheme and a CCA secure PEKS scheme may not remain CCA secure as a whole system. This is shown as follows: an adversary with a target ciphertext $C_{\text{M}}\|C_{\text{W}}$ can produce another valid ciphertext $C_{\text{M}}\|C_{\text{W}'}$ where $C_{\text{W}'}$ is a valid PEKS of some keyword W' and retrieve the plaintext M and thus breaking the CCA security of the PKE+PEKS scheme. So, a unified security model for the joint CCA-security of PKE+PEKS is desired.

1.1 Related Work

The related work on the subject of this paper is reasonably current and exhaustive in related work section of [BLSS15] and we reproduce almost verbatim from [BLSS15].

Abdalla et al. [ABC+05] have defined computational and statistical relaxations of perfect consistency for a PEKS scheme and showed that the BDOP PEKS scheme [BDOP04] is computationally consistent. They have also proposed a new statistically consistent scheme. Moreover, they have provided a transform of an anonymous identity-based encryption (IBE) scheme to a PEKS scheme that, unlike the BDOP PEKS scheme, gives consistency. Baek et al. [BSS06] have formally defined a combined scheme for PKE and PEKS (denoted as PKE/PEKS) based on the BDOP PEKS scheme and the variation of ElGamal encryption scheme with the randomness reuse technique [Kur02]. Parallel to these works, various other researchers have also studied the design and efficiency of the PEKS schemes including [BW06,FP07,ZI07,BSS08,ABN10]. Crescenzo and Saraswat [DS07] have constructed the first PEKS scheme which was not based on bilinear forms. Various other works [SVEG10,INHJ11,SR14] have studied the application aspects of PEKS.

Boneh et al. [BDOP04] formalized the security precisely for the PEKS scheme with IND-PEKS-CPA notion. Later Baek et al. [BSS08] combined PKE and PEKS with a joint security notion. But as their idea covered only data privacy and not the keyword privacy, the notion lacks completeness. Zhang and Imai [ZI07] first extended the security notion to achieve both data privacy and keyword privacy. The security notion for data privacy is IND-PKE-CCA, which is achieved in their scheme using a tag-based CCA-secure PKE scheme, and for keyword privacy the notion is IND-PEKS-CPA, which they have achieved using a CPA-secure PEKS scheme. Further their scheme achieves non-malleability with respect to the PKE component only and not with respect to the PEKS component. Hence their construction has IND-PKE-CCA security for data privacy but only IND-PEKS-CPA security for keyword privacy. Also, the joint security of their construction is built up on the key separation strategy, that is, using different keys for different cryptographic operations. Hence the construction suffers from double key size, which increases key-maintenance overheads unnecessarily during the practical implementations. However, none of the works [BSS08, ZI07] prove the joint security of a PKE+PEKS scheme in strongest notion, that is, 'IND-PKE+PEKS-CCA security'. One reason for why they are unable to give an IND-PEKS-CCA security for their schemes is that the adversary in their model is not given access to a test oracle [ABN10]. In [ABN10], Abdalla et al. introduced a new combined CCA-security notion on the standard model with a privilege to the adversary to access both, decryption oracle and test oracle. To achieve the CCA security for the PKE+PEKS scheme, they have followed the idea of [DK05], and combined two schemes, a tag-based CCA-secure PKE scheme and a tag-based CCA-secure PEKS scheme, but this idea leads to increase the computational overhead of the resulting PKE+PEKS scheme which is not appreciated at the practical platform. Additionally, their construction also suffers from double key size due to the adoption of key separation strategy. Recently, [CZLZ14] have minimized the key size of their PKE+PEKS scheme using a single key pair for both PKE and PEKS operations. They have defined data privacy and keyword privacy for PKE+PEKS schemes separately and claimed that the PKE+PEKS scheme is said to achieve the joint CCA-security if it attains keyword privacy and data privacy simultaneously.

1.2 Our Contribution

We present a construction of an efficient integrated PKE+PEKS scheme with short ciphertexts and prove its security in the standard model. We prove that our scheme is IND-PKE-CCA secure, that is, provides message confidentiality against an adaptive chosen ciphertext adversary, and also achieves IND-PEKS-CCA security, that is, provides keyword privacy against an adaptive chosen ciphertext adversary, under the SXDH assumption.

Up till now, although there have been lot of research on searchable encryption, the only fully secure schemes [ABN10, CZLZ14, BLSS15] are inefficient to be practical enough to be used in implementation. We propose a state of art efficient, computationally and bandwidth-wise, fully secure practical scheme which, we believe, can be used in real applications.

At SeCrypt 2015, Buccafurri et al. [BLSS15] presented an integrated public-key encryption (PKE) and public-key encryption with keyword search (PEKS) scheme (PKE+PEKS) whose security relies on the SXDH assumption. Their scheme is relatively efficient and our scheme improves upon it. Our scheme has much shorter ciphertexts and uses fewer number of the pairings. Please see Tables 1 and 2 for detailed comparison with [BLSS15] and other PEKS schemes.

Also, in comparison to the scheme in [CZLZ14], our construction has shorter public keys, shorter secret keys, shorter ciphertexts and a much improved efficiency in terms of computation. Further, we provide a unified proof of the overall security of the whole system in a much tighter way. Also, our scheme uses one unified framework for the full PKE+PEKS scheme—the security of the scheme relies on one single hardness assumption and we use the same bilinear pairing map throughout the scheme, instead of using different groups/maps/structures at different stages of the scheme which makes the implementation of our scheme much simpler both on hardware and on software.

We use the method of Paterson et al. [PSST11] of using bit prefix and a one-time signature to enable us to use the same key-pair for our integrated PKE+PEKS scheme and to obtain the joint security for our scheme. To obtain the short size and efficiency, we use the short IBE and IBS schemes of [JR13] which use asymmetric pairings to enable an extremely fast implementation useful for practical applications.

1.3 Outline of the Paper

The rest of this paper is organized as follows. In Sect. 2, we introduce some related mathematical definitions, problems and assumptions. In Sect. 3, we formally define an integrated public key encryption (PKE) and public key encryption with keyword search (PEKS) scheme (PKE+PEKS) and a unified security model for it. Our proposed PKE+PEKS scheme is presented in Sect. 4. In Sect. 5, we analyse the security of our scheme and in Sect. 6, we do an efficiency comparison with the state-of-art. Finally, in Sect. 7, we conclude our work and point a few improvements that can be made while implementing our scheme.

2 Preliminaries

In this section, we introduce some relevant definitions, mathematical problems and assumptions. Note that these definitions are standard and we reproduce these almost verbatim from [BLSS15] to maintain consistency and easier comparison.

2.1 Notations

We denote by $y \leftarrow A(x)$ the operation of running a randomized or deterministic algorithm $A(x)$ and storing the output to the variable y. If X is a set, then $v \overset{\$}{\leftarrow} X$ denotes the operation of choosing an element v of X according to the uniform random distribution on X. We say that a given function $f : N \to [0, 1]$ is *negligible in* n if $f(n) < 1/p(n)$ for any polynomial p for sufficiently large n. For a group G and $g \in G$, we write $G = \langle g \rangle$ if g is a generator of G.

2.2 Bilinear Maps

Let G_1, G_2 and G_T be multiplicative cyclic groups of the same prime order q. A map $e : G_1 \times G_2 \to G_T$ is called a *cryptographic bilinear map* or a *pairing* if it satisfies the following properties:

Bilinearity: For all $(g_1, g_2) \in G_1 \times G_2$ and for all $a, b \in \mathbb{Z}_q$, $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$.
Non-Degeneracy: There exists $(g_1, g_2) \in G_1 \times G_2$ such that $e(g_1, g_2) \neq 1$, the identity of G_T.

Computability: There exists an efficient algorithm to compute $e(g_1, g_2) \in G_T$, for all $(g_1, g_2) \in G_1 \times G_2$.

A pairing $e : G_1 \times G_2 \to G_T$ is called a *symmetric* or a *Type 1* pairing if $G_1 = G_2$ otherwise it is called *asymmetric*. Asymmetric pairings are further categorized into *Type 2* and *Type 3* pairings. If there exists an efficiently computable isomorphism between G_1 and G_2 then the pairing is referred to as Type 2, whereas if there is no efficiently computable isomorphism between G_1 and G_2, then the pairing is referred to as Type 3.

2.3 Symmetric eXternal Diffie-Hellman (SXDH) Assumption

Definition 1. Let G be a multiplicative cyclic group and g be its generator. Let $a, b, c \in \mathbb{Z}_q^\times$ be randomly chosen and kept secret. Given $g, g^a, g^b, g^c \in G$, the *decisional Diffie-Hellman problem* (DDHP) in the group G is to decide if $g^{ab} = g^c$.

Definition 2. The *DDH assumption* holds in a group G if there is no efficient polynomial time algorithm which can solve DDHP in G. Specifically, let $\mathcal{A}$ be a DDH adversary for a group G which takes as input a generator $g \in G$, and three elements $g_1 = g^a$, $g_2 = g^b$, $g_3 = g^c$ of the group G, and outputs 1 if $g_3 = g^{ab}$ and 0 otherwise. Further, let the advantage of $\mathcal{A}$ be defined as

$$\mathbf{Adv}_{\mathcal{A}} = |Pr[\mathcal{A}(g, g^a, g^b, g^{ab}) = 1]| - |Pr[\mathcal{A}(g, g^a, g^b, g^c) = 1]|$$

where $g \overset{\$}{\leftarrow} G^\times$, $a \overset{\$}{\leftarrow} \mathbb{Z}_q^\times$, $b \overset{\$}{\leftarrow} \mathbb{Z}_q^\times$ and $c \overset{\$}{\leftarrow} \mathbb{Z}_q^\times$. We say that (t, ε)-*DDH assumption* holds in the group G if any DDH adversary running in time t has an advantage at most ε.

Definition 3. Given two cyclic groups G_1 and G_2, we say the *Symmetric eXternal Diffie-Hellman* (SXDH) assumption holds if the DDH assumption is true in both the groups G_1 and G_2.

3 Integrated PKE and PEKS Scheme (PKE+PEKS)

Here we reproduce almost verbatim from [BLSS15] the formal definition of an integrated public-key encryption (PKE) and public-key encryption with keyword search (PEKS) scheme (PKE+PEKS).

In PEKS, three parties called *sender*, *receiver* and *server* are involved. The sender is a party that creates and sends encrypted keywords, which we call *PEKS ciphertexts*. The receiver is a party that creates trapdoors and sends them to the server to find the data that it wants. The server is a party that receives PEKS ciphertexts and performs search upon receiving trapdoors from the receiver.

3.1 Formal Definition of PKE+PEKS

A PKE+PEKS scheme comprises of six algorithms: *Setup*, *KeyGen*, *Encrypt*, *Decrypt*, *TokenGen* and *Test*.

$Params \leftarrow$ ***Setup***(1^k)**:** This is the system initialization algorithm run by the receiver which takes as input a security parameter 1^k and outputs public parameters *Params*. In all the algorithms from here onward, *Params* will be considered as an implicit input.

$(pk_X, sk_X) \leftarrow$ ***keyGen***(X)**:** This is the key generation algorithm run by a user X which takes input *Params* and outputs a key pair (pk_X, sk_X). For the receiver $X = R$, the key pair is its (public key, private key) pair (pk_R, sk_R) and for a sender $X = S$, the key pair is its (verification key, signing key) pair (vk_S, sk_S).

$\mathcal{U} \leftarrow$ ***Encrypt***(pk_R, m, w)**:** This is a randomized algorithm run by the sender and takes input *Params*, the receiver's public key pk_R, a message m and a keyword w, and outputs the joint PKE+PEKS ciphertext $\mathcal{U}$.

$m \leftarrow$ ***Decrypt***$(pk_R, sk_R, \mathcal{U})$**:** This is a deterministic algorithm run by the receiver and takes input *Params*, the receiver's public key pk_R and the secret key sk_R and a ciphertext $\mathcal{U}$, and outputs a message m or $\perp$.

$t_w \leftarrow$ ***TokenGen***(pk_R, sk_R, w)**:** This is a randomized algorithm run by the receiver and takes input *Params*, the receiver's public key pk_R and the secret key sk_R and a keyword w, and outputs a token t_w which it gives to the server.

$b \leftarrow$ ***Test***$(pk_R, t_w, \mathcal{U})$**:** This is a deterministic algorithm run by the server and takes input *Params*, the receiver's public key pk_R, a token t_w and a ciphertext $\mathcal{U}$, and outputs a bit $b \in \{0, 1\}$ or $\perp$.

From now on, where the context is clear, the inputs *Params* and the keys will be assumed to be implicit and we will not write them explicitly in the algorithms.

3.2 Security Model for PKE+PEKS

Joint data and keyword privacy for PKE+PEKS schemes is defined via the following experiment.

Setup: On input a security parameter 1^k, the challenger $\mathcal{C}$ runs $KeyGen(1^k)$ to generate the public parameter *Params* and the system key pair (pk, sk) and gives the adversary $\mathcal{A}$ the public key pk.

Phase 1: $\mathcal{A}$ can adaptively make three types of queries:

- Decryption query $\langle u \rangle$: $\mathcal{C}$ responds with $m \leftarrow Decrypt(sk, u)$.
- Token query $\langle w \rangle$: $\mathcal{C}$ responds with $t_w \leftarrow TokenGen(sk, w)$.
- Test query $\langle u, w \rangle$: $\mathcal{C}$ responds with $Test(u, t_w \leftarrow TokenGen(sk, w))$.

Challenge: $\mathcal{A}$ outputs two messages m_0^* and m_1^* and two keywords w_0^* and w_1^*. $\mathcal{C}$ picks a random bit $b \xleftarrow{\$} \{0, 1\}$ and sends $u^* \leftarrow Encrypt(pk, m_b^*, w_b^*)$ to $\mathcal{A}$ as the challenge ciphertext.

Phase 2: $\mathcal{A}$ can adaptively make more queries as in Phase 1 subject to the restrictions that it is not allowed to make

- Decryption query $\langle u^* \rangle$,
- Token queries $\langle w_0^* \rangle$ and $\langle w_1^* \rangle$, and
- Test queries $\langle u^*, w_0^* \rangle$ and $\langle u^*, w_1^* \rangle$.

$\mathcal{C}$ responds the same way as in Phase 1.

Guess: $\mathcal{A}$ outputs its guess (b^*) for (b).

Definition 4. The adversary succeeds in breaking the data privacy or the keyword privacy if $b^* = b$. We denote this event by $Succ_{\mathcal{A}}$ and define $\mathcal{A}$'s advantage as

$$Adv_{\mathcal{A}}(1^k) \overset{\text{def}}{=} |Pr[Succ_{\mathcal{A}}] - 1/2|.$$

We say a PKE+PEKS scheme is IND-PKE+PEKS-CCA secure, that is, the scheme achieves data privacy and keyword privacy simultaneously against an adaptive chosen ciphertext adversary, if $Adv_{\mathcal{A}}(1^k)$ is negligible. A PKE+PEKS scheme is said to be $(t, q_w, q_t, q_d, \varepsilon)$-IND-PKE+PEKS-CCA secure, if for all t-time adversaries making at most q_w token queries, at most q_t test queries, and at most q_d decryption queries have advantage at most ε.

Definition 5 (Data Privacy). We may define a game for just data privacy, if $w_0^* = w_1^*$ and the adversary has no restriction on Token queries and Test queries in the above game. The adversary succeeds in breaking the data privacy if $b^* = b$. We denote this event by $Succ_{\mathcal{A}}^{dp}$ and define $\mathcal{A}$'s advantage as

$$Adv_{\mathcal{A}}^{dp}(1^k) \overset{\text{def}}{=} |Pr[Succ_{\mathcal{A}}^{dp}] - 1/2|.$$

A PKE+PEKS scheme is said to have $(t, q_w, q_t, q_d, \varepsilon)$-data privacy if for all t-time adversaries making at most q_w token queries, at most q_t test queries, and at most q_d decryption queries have advantage at most ε against its data privacy. Informally, we say a PKE+PEKS scheme has data privacy if there is no PPT adversary having non-negligible advantage in 1^k in the above experiment.

Definition 6 (Keyword Privacy). We may define a game for just keyword privacy, if $m_0^* = m_1^*$ and the adversary has no restriction on Decryption queries in the above game. The adversary succeeds in breaking the keyword privacy if $b^* = b$. We denote this event by $Succ_{\mathcal{A}}^{kp}$ and define $\mathcal{A}$'s advantage as

$$Adv_{\mathcal{A}}^{kp}(1^k) \overset{\text{def}}{=} |Pr[Succ_{\mathcal{A}}^{kp}] - 1/2|.$$

A PKE+PEKS scheme is said to have $(t, q_w, q_t, q_d, \varepsilon)$-keyword privacy if for all t-time adversaries making at most q_w token queries, at most q_t test queries, and at most q_d decryption queries have advantage at most ε against its keyword privacy. Informally, we say a PKE+PEKS scheme has keyword privacy if there is no PPT adversary having non-negligible advantage in 1^k in the above experiment.

Remark 1. Note that our joint CCA-security notion for PKE+PEKS embodies both IND-PKE-CCA security and IND-PEKS-CCA security in the joint sense and is relatively unified and standard than previous ones considered in [CZLZ14, BLSS15].

4 Proposed Scheme

We present here our efficient and CCA secure integrated PKE+PEKS scheme. As described in Sect. 3, our scheme consists of the following algorithms: *Setup*, *KeyGen*, *Encrypt*, *Decrypt*, *TokenGen* and *Test*.

Setup: A receiver R wishing to receive joint PKE+PEKS messages uses a group generation algorithm for which the SXDH assumption holds to generate the public parameters of the system:

$$G := (q, G_1, G_2, G_T, e)$$

where G_1, G_2, and G_T are cyclic groups of prime order q and

$$e : G_1 \times G_2 \to G_T$$

is a Type 3 pairing. The receiver R then chooses two cryptographic collision resistant hash functions

$$H : \{0,1\}^* \to \mathbb{Z}_q^\times \quad \text{and} \quad J : \{0,1\}^* \to G_T.$$

Finally, R publishes the public parameters of the system as

$$\mathit{Params} = (G, H, J).$$

(These may be considered as part R's public key, but for sake of clarity we keep these separate.)

KeyGen: To generate the keys for the system, the receiver does the following:

- samples two random generators $g_1 \xleftarrow{\$} G_1^\times$ and $g_2 \xleftarrow{\$} G_2^\times$;
- samples $b, c, d, e, u, l, m, n, p \xleftarrow{\$} \mathbb{Z}_q^\times$;
- computes
  - $f_1 = g_1^b$,
  - $f_2 = g_2^c$,
  - $v_1 = g_1^{d-bl}$,
  - $v_2 = g_1^{e-bm}$,
  - $v_3 = g_1^{c-bn}$, and
  - $k = e(g_1, g_2)^{u-bp}$;
- sets the public key $pk_R = (g_1, f_1, v_1, v_2, v_3, k)$; and
- sets the master secret $sk_R = (g_2, f_2, l, m, n, p, d, e, u)$.

Encrypt: To encrypt a message $\text{M} \in G_T$ with a keyword $\text{W} \in \{0,1\}^*$ for the receiver R, a sender S does the following:

- samples two random generators $\tilde{g}_1 \xleftarrow{\$} G_1^\times$ and $\tilde{g}_2 \xleftarrow{\$} G_2^\times$;
- samples $\tilde{b}, \tilde{c}, \tilde{d}, \tilde{e}, \tilde{u}, \tilde{l}, \tilde{m}, \tilde{n}, \tilde{p} \xleftarrow{\$} \mathbb{Z}_q^\times$;
- computes
  - $\tilde{f}_1 = \tilde{g}_1^{\tilde{b}}$,
  - $\tilde{f}_2 = \tilde{g}_2^{\tilde{c}}$,

  - $\tilde{v}_1 = \tilde{g}_1^{\tilde{d}-\tilde{b}\tilde{l}}$,
  - $\tilde{v}_2 = \tilde{g}_1^{\tilde{e}-\tilde{b}\tilde{m}}$,
  - $\tilde{v}_3 = \tilde{g}_1^{\tilde{c}-\tilde{b}\tilde{n}}$, and
  - $\tilde{k} = e(\tilde{g}_1, \tilde{g}_2)^{\tilde{u}-\tilde{b}\tilde{p}}$;

- sets the verification key $vk_S = (\tilde{g}_1, \tilde{f}_1, \tilde{v}_1, \tilde{v}_2, \tilde{v}_3, \tilde{k})$; and
- sets the signing key $sk_S = (\tilde{g}_2, \tilde{f}_2, \tilde{l}, \tilde{m}, \tilde{n}, \tilde{p}, \tilde{d}, \tilde{e}, \tilde{u})$;
- sets $\mathrm{V} = J(vk_S)$
- picks $x, y, z, \mathrm{TAG_M}, \mathrm{TAG_W} \xleftarrow{\$} \mathbb{Z}_q$;
- computes
  - $i_\mathrm{V} = H(0\|\mathrm{V})$ and $i_\mathrm{W} = H(1\|\mathrm{W})$;
  - $C_{\mathrm{M}0} = \mathrm{M} \cdot k^x$, $C_{\mathrm{M}1} = g_1^x$, $C_{\mathrm{M}2} = f_1^x$, and $C_{\mathrm{M}3} = v_1^x v_2^{x i_\mathrm{V}} v_3^{x\mathrm{TAG_M}}$;
  - $C_{\mathrm{W}0} = \mathrm{V} \cdot k^y$, $C_{\mathrm{W}1} = g_1^y$, $C_{\mathrm{W}2} = f_1^y$, and $C_{\mathrm{W}3} = v_1^y v_2^{y i_\mathrm{W}} v_3^{y\mathrm{TAG_W}}$;
  - $h = H(C_\mathrm{M}\|C_\mathrm{W})$;
  - $R_\sigma = \tilde{g}_2^z$, $S_\sigma = \tilde{f}_2^z$, $T_\sigma = \tilde{g}_2^{\tilde{u}+z(\tilde{d}+h\tilde{e})}$, $W_{\sigma 1} = \tilde{g}_2^{-\tilde{p}-z(\tilde{l}+h\tilde{m})}$, and $W_{\sigma 2} = \tilde{g}_2^{-z\tilde{n}}$;
- sets
  - $C_\mathrm{M} = (C_{\mathrm{M}0}, C_{\mathrm{M}1}, C_{\mathrm{M}2}, C_{\mathrm{M}3}, \mathrm{TAG_M})$;
  - $C_\mathrm{W} = (C_{\mathrm{W}0}, C_{\mathrm{W}1}, C_{\mathrm{W}2}, C_{\mathrm{W}3}, \mathrm{TAG_W})$;
  - $\sigma = (R_\sigma, S_\sigma, T_\sigma, W_{\sigma 1}, W_{\sigma 2})$;
- and finally declares the ciphertext $\mathcal{U} = (vk_S, C_\mathrm{M}, C_\mathrm{W}, \sigma)$.

Decrypt: To decrypt the ciphertext $\mathcal{U} = (u_1, u_2, u_3, u_4)$, the receiver does the following:

- obtains $\tilde{g}_1, \tilde{f}_1, \tilde{v}_1, \tilde{v}_2, \tilde{v}_3, \tilde{k}$ from u_1;
- obtains $R_\sigma, S_\sigma, T_\sigma, W_{\sigma 1}, W_{\sigma 2}$ from u_4;
- computes $h = H(u_2\|u_3)$;
- chooses $\tilde{\mathrm{M}} \xleftarrow{\$} G_T, \tilde{s} \xleftarrow{\$} \mathbb{Z}_q^\times, \mathrm{TAG}_{\tilde{\mathrm{M}}} \xleftarrow{\$} \mathbb{Z}_q^\times$;
- computes
  - $C_{\tilde{\mathrm{M}}0} := \tilde{\mathrm{M}} \cdot \tilde{k}^{\tilde{s}}$,
  - $C_{\tilde{\mathrm{M}}1} := \tilde{g}_1^{\tilde{s}}$,
  - $C_{\tilde{\mathrm{M}}2} := \tilde{f}_1^{\tilde{s}}$,
  - $C_{\tilde{\mathrm{M}}3} := \tilde{v}_1^{\tilde{s}} \tilde{v}_2^{h\tilde{s}} \tilde{v}_3^{\tilde{s}\mathrm{TAG}_{\tilde{\mathrm{M}}}}$;
- checks whether the PKE+PEKS ciphertext $\mathcal{U}$ is *valid*. That is, whether

$$\tilde{\mathrm{M}} = \frac{C_{\tilde{\mathrm{M}}0} e(C_{\tilde{\mathrm{M}}3}, R_\sigma)}{e(C_{\tilde{\mathrm{M}}1}, S_\sigma^{\mathrm{TAG}_{\tilde{\mathrm{M}}}} T_\sigma) e(C_{\tilde{\mathrm{M}}2}, W_{\sigma 1} W_{\sigma 2}^{\mathrm{TAG}_{\tilde{\mathrm{M}}}})}. \tag{1}$$

- If the above equality does not hold then outputs $\perp$.
- Otherwise it obtains $C_{\mathrm{M}0}, C_{\mathrm{M}1}, C_{\mathrm{M}2}, C_{\mathrm{M}3}, \mathrm{TAG_M}$ from u_2;
- computes $\mathrm{V} = J(u_1)$;
- sets $i_\mathrm{V} = H(0\|\mathrm{V})$;
- computes a corresponding decryption key

$$SK_{i_\mathrm{V}} = (R_\mathrm{V}, S_\mathrm{V}, T_\mathrm{V}, W_{\mathrm{V}1}, W_{\mathrm{V}2});$$

where

- $r \xleftarrow{\$} \mathbb{Z}_q^\times$,
- $R_{\text{V}} = g_2^r$,
- $S_{\text{V}} = f_2^r$,
- $T_{\text{V}} = g_2^{u+r(d+i_{\text{V}}e)}$,
- $W_{\text{V}1} = g_2^{-p-r(l+i_{\text{V}}m)}$, and
- $W_{\text{V}2} = g_2^{-rn}$;

- finally, it outputs

$$\text{M} \leftarrow \frac{C_{\text{M}0} e(C_{\text{M}3}, R_{\text{V}})}{e(C_{\text{M}1}, S_{\text{V}}^{\text{TAG}_{\text{M}}} T_{\text{V}}) e(C_{\text{M}2}, W_{\text{V}1} W_{\text{V}2}^{\text{TAG}_{\text{M}}})}. \tag{2}$$

Tokengen: To generate a token t_{W} for the keyword W to give to the server, the receiver chooses $r \xleftarrow{\$} \mathbb{Z}_q^\times$, computes

- $i_{\text{W}} = H(1\|\text{W})$,
- $R_{\text{W}} = g_2^r$,
- $S_{\text{W}} = f_2^r$,
- $T_{\text{W}} = g_2^{u+r(d+i_{\text{W}}e)}$,
- $W_{\text{W}1} = g_2^{-p-r(l+i_{\text{W}}m)}$, and
- $W_{\text{W}2} = g_2^{-rn}$,

and outputs the token:

$$t_{\text{W}} = (R_{\text{W}}, S_{\text{W}}, T_{\text{W}}, W_{\text{W}1}, W_{\text{W}2}). \tag{3}$$

Test: To test whether the ciphertext $\mathcal{U} = (u_1, u_2, u_3, u_4)$ includes the keyword W or not using the token t_{W}, the server does the following:

- obtains $\tilde{g}_1, \tilde{f}_1, \tilde{v}_1, \tilde{v}_2, \tilde{v}_3, \tilde{k}$ from u_1;
- obtains $R_\sigma, S_\sigma, T_\sigma, W_{\sigma 1}, W_{\sigma 2}$ from u_4;
- computes $h = H(u_2\|u_3)$;
- chooses $\tilde{\text{M}} \xleftarrow{\$} G_T, \tilde{s} \xleftarrow{\$} \mathbb{Z}_q^\times$, $\text{TAG}_{\tilde{\text{M}}} \xleftarrow{\$} \mathbb{Z}_q^\times$;
- computes
  - $C_{\tilde{\text{M}}0} := \tilde{\text{M}} \cdot \tilde{k}^{\tilde{s}}$,
  - $C_{\tilde{\text{M}}1} := \tilde{g}_1^{\tilde{s}}$,
  - $C_{\tilde{\text{M}}2} := \tilde{f}_1^{\tilde{s}}$,
  - $C_{\tilde{\text{M}}3} := \tilde{v}_1^{\tilde{s}} \tilde{v}_2^{h\tilde{s}} \tilde{v}_3^{\tilde{s}\text{TAG}_{\tilde{\text{M}}}}$;
- checks whether the PKE+PEKS ciphertext $\mathcal{U}$ is *valid*. That is, whether

$$\tilde{\text{M}} = \frac{C_{\tilde{\text{M}}0} e(C_{\tilde{\text{M}}3}, R_\sigma)}{e(C_{\tilde{\text{M}}1}, S_\sigma^{\text{TAG}_{\tilde{\text{M}}}} T_\sigma) e(C_{\tilde{\text{M}}2}, W_{\sigma 1} W_{\sigma 2}^{\text{TAG}_{\tilde{\text{M}}}})}. \tag{4}$$

- If the above equality does not hold then outputs 0.
- Otherwise it obtains $C_{\text{W}0}, C_{\text{W}1}, C_{\text{W}2}, C_{\text{W}3}, \text{TAG}_{\text{W}}$ from u_3 and checks if

$$J(u_1) = \frac{C_{\text{W}0} e(C_{\text{W}3}, R_{\text{W}})}{e(C_{\text{W}1}, S_{\text{W}}^{\text{TAG}_{\text{W}}} T_{\text{W}}) e(C_{\text{W}2}, W_{\text{W}1} W_{\text{W}2}^{\text{TAG}_{\text{W}}})}. \tag{5}$$

- If yes then outputs 1, else outputs 0.

Remark 2. Note that to maintain a "uniformity" we have used the Naor transform of the IBE of [JR13] as a signature. We could have used the signature scheme of [JR13] for a little more efficiency of our proposed PKE+PEKS. Again, to maintain "uniformity" and comparability with the previous schemes, in the *Decrypt* and *Test* algorithms, we have done a generic Naor transform verification of the ciphertext validity; we can improve the efficiency by making it more direct. Finally, in the *Decrypt* algorithm, the receiver can use its secret key sk_R to directly decrypt the ciphertext instead of generating the "secret key" corresponding to i_{V} to increase efficiency.

4.1 Correctness of the Proposed Scheme

Theorem 1. *The proposed scheme is correct.*

Proof. With the terms in the expressions below defined as in the algorithms *Setup*, *KeyGen*, *Encrypt*, *Decrypt*, *TokenGen*, and *Test* defined in the proposed scheme in Sect. 4, we note that for a correctly generated ciphertext $\mathcal{U} = (u_1, u_2, u_3, u_4)$,

- $u_1 = vk_S = (\tilde{g}_1, \tilde{f}_1, \tilde{v}_1, \tilde{v}_2, \tilde{v}_3, \tilde{k})$;
- $u_2 = C_{\mathrm{M}} = (C_{\mathrm{M}0} := \mathrm{M} \cdot k^x, C_{\mathrm{M}1} := g_1^x, C_{\mathrm{M}2} := f_1^x, C_{\mathrm{M}3} := v_1^x v_2^{x i_{\mathrm{V}}} v_3^{x\mathrm{TAG_M}}, \mathrm{TAG_M})$;
- $u_3 = C_{\mathrm{W}} = (C_{\mathrm{W}0} := \mathrm{V} \cdot k^y, C_{\mathrm{W}1} := g_1^y, C_{\mathrm{W}2} := f_1^y, C_{\mathrm{W}3} := v_1^y v_2^{y i_{\mathrm{W}}} v_3^{y\mathrm{TAG_W}}, \mathrm{TAG_W})$; and
- $u_4 = \sigma = (R_\sigma = \tilde{g}_2^z, S_\sigma = \tilde{f}_2^z, T_\sigma = \tilde{g}_2^{\tilde{u}+z(\tilde{d}+h\tilde{e})}, W_{\sigma 1} = \tilde{g}_2^{-\tilde{p}-z(\tilde{l}+h\tilde{m})}, W_{\sigma 2} = \tilde{g}_2^{-z\tilde{n}})$.

Thus, the three pairings in the Eq. (2) can be simplified as follows.

$$\begin{aligned} e(C_{\mathrm{M}3}, R_{\mathrm{V}}) &= e(v_1^x v_2^{x i_{\mathrm{V}}} v_3^{x\mathrm{TAG_M}}, g_2^r) \\ &= e((g_1^{d-bl})^x (g_1^{e-bm})^{x i_{\mathrm{V}}} (g_1^{c-bn})^{x\mathrm{TAG_M}}, g_2^r) \\ &= e(g_1^{(d-bl+e i_{\mathrm{V}}-bm i_{\mathrm{V}}+c\mathrm{TAG_M}-bn\mathrm{TAG_M})x}, g_2^r) \\ &= e(g_1, g_2)^{(d-bl+e i_{\mathrm{V}}-bm i_{\mathrm{V}}+c\mathrm{TAG_M}-bn\mathrm{TAG_M})xr}; \end{aligned} \tag{6}$$

$$\begin{aligned} e(C_{\mathrm{M}1}, S_{\mathrm{V}}^{\mathrm{TAG_M}} T_{\mathrm{V}}) &= e(g_1^x, (f_2^r)^{\mathrm{TAG_M}} g_2^{u+r(d+i_{\mathrm{V}}e)}) \\ &= e(g_1^x, (g_2^c)^{r\mathrm{TAG_M}} g_2^{u+r(d+i_{\mathrm{V}}e)}) \\ &= e(g_1^x, g_2^{cr\mathrm{TAG_M}+u+r(d+i_{\mathrm{V}}e)}) \\ &= e(g_1, g_2)^{ux+(c\mathrm{TAG_M}+d+i_{\mathrm{V}}e)xr}; \end{aligned} \tag{7}$$

$$\begin{aligned} e(C_{\mathrm{M}2}, W_{\mathrm{V}1} W_{\mathrm{V}2}^{\mathrm{TAG_M}}) &= e(f_1^x, g_2^{-p-r(l+i_{\mathrm{V}}m)} (g_2^{-rn})^{\mathrm{TAG_M}}) \\ &= e((g_1^b)^x, g_2^{-p-r(l+i_{\mathrm{V}}m)-rn\mathrm{TAG_M}}) \\ &= e((g_1, g_2)^{-bxp-bxr(l+i_{\mathrm{V}}m+n\mathrm{TAG_M})}; \end{aligned} \tag{8}$$

Hence, the decryption Eq. (2) is correct since

$$\begin{aligned}
&\frac{C_{\text{M}0}e(C_{\text{M}3}, R_{\text{V}})}{e(C_{\text{M}1}, S_{\text{V}}^{\text{TAG}_{\text{M}}}T_{\text{V}})e(C_{\text{M}2}, W_{\text{V}1}W_{\text{V}2}^{\text{TAG}_{\text{M}}})} \\
&\quad = \frac{(\text{M} \cdot k^{x})(e(g_1, g_2)^{(d-bl+ei_{\text{V}}-bmi_{\text{V}}+c\text{TAG}_{\text{M}}-bn\text{TAG}_{\text{M}})xr})}{(e(g_1, g_2)^{ux+(c\text{TAG}_{\text{M}}+d+i_{\text{V}}e)xr})(e((g_1, g_2)^{-bxp-bxr(l+i_{\text{V}}m-n\text{TAG}_{\text{M}})})} \\
&\qquad \text{(from Equations (6), (7) and (8))} \\
&\quad = (\text{M} \cdot k^{x})e(g_1, g_2)^{-x(u-bp)} \\
&\quad = (\text{M} \cdot k^{x})k^{-x} \\
&\quad = \text{M}. \qquad (9)
\end{aligned}$$

Since the terms in the Eqs. (1), (4) and (5) are generated similarly to those in the Eq. (2), the correctness of the *Test* follows similarly as that of *Decrypt*.

Hence the proposed scheme is correct.

5 Security Proof

In this section, we analyse the security of our scheme. We prove that the presented scheme is secure under the SXDH assumption.

We follow the security proof of the IBE in [JR13] using the simulation technique of [Wat09] of using a sequence of games and adopting *semi-functional keys* and *semi-functional ciphertexts*. For the notion of construction of these *semi-functional*-values, [CLL+12, JR13] can be referred. The advantage of an adversary in winning the IND-PKE+PEKS-CCA game is then shown to be bounded in terms of its advantage in distinguishing between successive games.

Remark 3. For the sake of brevity and page limitation, the parts of the proof which are already available in literature, has been cited and presented here only briefly.

Theorem 2. *If the DDH assumption holds in both the groups G_1 and G_2 then there is no IND-PKE+PEKS-CCA adversary $\mathcal{A}$ for the presented integrated PKE+PEKS scheme.*

Proof: Let $\mathcal{A}$ be a t-time IND-PKE+PEKS-CCA adversary making at most q_w token queries, at most q_t test queries and at most q_d decryption queries, and with advantage ε.

Let (m_0^*, w_0^*) and (m_1^*, w_1^*) be the target message-keyword output by $\mathcal{A}$ at the end of Phase 1. Let $b \overset{\$}{\leftarrow} \{0,1\}$, $\bar{b} = 1 - b$ and $u^* = (u_1^*, u_2^*, u_3^*, u_4^*) \leftarrow$ *Encrypt*(pk, m_b^*, w_b^*) be the challenge ciphertext. Let b^* be the guess output by $\mathcal{A}$ at the end of Phase 2.

We prove that if the challenger $\mathcal{C}$ chooses a random message $\tilde{m} \overset{\$}{\leftarrow} G_T$ and a random keyword $\tilde{w} \overset{\$}{\leftarrow} \{0,1\}^*$ and gives $\tilde{u}^* = (\tilde{u}_1^*, \tilde{u}_2^*, \tilde{u}_3^*, \tilde{u}_4^*) \leftarrow$

Encrypt$(pk, \tilde{m}, \tilde{w})$ instead of $u^* = (u_1^*, u_2^*, u_3^*, u_4^*)$ to the adversary $\mathcal{A}$ as the challenge ciphertext, the view of the adversary will remain computationally indistinguishable (in view of the SXDH assumption on G) and $\mathcal{A}$ will not be any wiser and will keep playing the game without aborting.

Since $\tilde{u}^*$ is completely random and independent of the target message-keyword pairs (m_0^*, w_0^*) and (m_1^*, w_1^*) in the view of the adversary $\mathcal{A}$, the guess output b^* by $\mathcal{A}$ at the end of Phase 2 must also be completely random.

Hence the advantage $\mathbf{Adv}_{\mathcal{A}}$ of $\mathcal{A}$ in winning the IND-PKE+PEKS-CCA game is then bounded in terms of its advantage in distinguishing between successive games and hence must be negligible and the scheme must be IND-PKE+PEKS-CCA secure.

We achieve this through a sequence of games where each successive game differs from the preceding game in such a way that the two games are either statistically indistinguishable or computationally indistinguishable in view of the SXDH assumption on G defined as follows:

Game $\mathcal{G}_0$: This is the actual IND-PKE+PEKS-CCA security game as defined in Section 3.

Game $\mathcal{G}_1$: This game is similar to the previous game in all aspects except that instead of actual ciphertexts, the challenger outputs *partial semifunctional ciphertexts* as follows:

– Let $\mathcal{U} = (u_1, u_2, u_3, u_4)$ be the actual ciphertext with

$$\begin{aligned} u_2 = C_{\text{M}} = (&C_{\text{M}0} := \text{M} \cdot k^x, C_{\text{M}1} := g_1^x, C_{\text{M}2} := f_1^x, \\ &C_{\text{M}3} := v_1^x v_2^{x i_{\text{V}}} v_3^{x \text{TAG}_{\text{M}}}, \text{TAG}_{\text{M}}) \quad \text{and} \\ u_3 = C_{\text{W}} = (&C_{\text{W}0} := \text{V} \cdot k^y, C_{\text{W}1} := g_1^y, C_{\text{W}2} := f_1^y, \\ &C_{\text{W}3} := v_1^y v_2^{y i_{\text{W}}} v_3^{y \text{TAG}_{\text{W}}}, \text{TAG}_{\text{W}}). \end{aligned}$$

– The challenger picks $x', y' \xleftarrow{\$} \mathbb{Z}_q$ and sets the corresponding *partial semi-functional* components as:

$$\begin{aligned} u_2' = C_{\text{M}}' = (&C_{\text{M}0}' := C_{\text{M}0} \cdot e(g_1, g_2)^{ux'}, C_{\text{M}1}' := C_{\text{M}1} \cdot g_1^{x'}, C_{\text{M}2}' := C_{\text{M}2}, \\ &C_{\text{M}3}' := C_{\text{M}3} \cdot g_1^{(d + e i_{\text{V}} + c \text{TAG}_{\text{M}}) x'}, \text{TAG}_{\text{M}}) \\ = (&C_{\text{M}0}' := \text{M} \cdot k^x \cdot e(g_1, g_2)^{ux'}, C_{\text{M}1}' := g_1^x \cdot g_1^{x'}, C_{\text{M}2}' := f_1^x, \\ &C_{\text{M}3}' := v_1^x v_2^{x i_{\text{V}}} v_3^{x \text{TAG}_{\text{M}}} \cdot g_1^{(d + e i_{\text{V}} + c \text{TAG}_{\text{M}}) x'}, \text{TAG}_{\text{M}}) \end{aligned}$$

and

$$\begin{aligned} u_3' = C_{\text{W}}' = (&C_{\text{W}0}' := C_{\text{W}0} \cdot e(g_1, g_2)^{uy'}, C_{\text{W}1}' := C_{\text{W}1} \cdot g_1^{y'}, C_{\text{W}2}' := C_{\text{W}2}, \\ &C_{\text{W}3}' := C_{\text{W}3} \cdot g_1^{(d + e i_{\text{V}} + c \text{TAG}_{\text{W}}) y'}, \text{TAG}_{\text{W}}) \\ = (&C_{\text{W}0}' := \text{V} \cdot k^y \cdot e(g_1, g_2)^{uy'}, C_{\text{W}1}' := g_1^y \cdot g_1^{y'}, C_{\text{W}2}' := f_1^y, \\ &C_{\text{W}3}' := v_1^y v_2^{y i_{\text{W}}} v_3^{y \text{TAG}_{\text{W}}} \cdot g_1^{(d + e i_{\text{V}} + c \text{TAG}_{\text{M}}) y'}, \text{TAG}_{\text{W}}). \end{aligned}$$

– Finally, the challenger outputs the (partial semifunctional) ciphertext

$$\mathcal{U}' = (u_1, u_2', u_3', u_4)$$

as the ciphertext.

In view of the DDH assumption in the group G_1, the two pairs of tuples

$$(\langle g_1, g_1^b, g_1^{xb}, g_1^x\rangle, \langle g_1, g_1^b, g_1^{xb}, g_1^{x+x'}\rangle)$$

and

$$(\langle g_1, g_1^b, g_1^{yb}, g_1^y\rangle, \langle g_1, g_1^b, g_1^{yb}, g_1^{y+y'}\rangle)$$

are indistinguishable to the adversary. Hence from the view of the adversary $\mathcal{A}$, the games $\mathcal{G}_0$ and $\mathcal{G}_1$ are computationally indistinguishable.

We note here that the advantage gap between two consecutive games can be proved by the reduction to the DDH assumption following the same proofs given in [CLL+12, JR13]. From here onwards wherever we need to show this reduction we mention it as 'indistinguishable from the view of adversary'.

Game $\mathcal{G}_2$: This game is similar to the previous game in all aspects except that instead of partial semifunctional ciphertexts, the challenger outputs *semi-functional ciphertexts* as follows:

– Let $\mathcal{U} = (u_1, u_2', u_3', u_4)$ be the partial semifunctional ciphertext with

$$u_1 = vk_S = (\tilde{g}_1, \tilde{f}_1 = \tilde{g}_1^{\tilde{b}}, \tilde{v}_1 = \tilde{g}_1^{\tilde{d}-\tilde{b}\tilde{l}}, \tilde{v}_2 = \tilde{g}_1^{\tilde{e}-\tilde{b}\tilde{m}}, \tilde{v}_3 = \tilde{g}_1^{\tilde{c}-\tilde{b}\tilde{n}},$$
$$\tilde{k} = e(\tilde{g}_1, \tilde{g}_2)^{\tilde{u}-\tilde{b}\tilde{p}} \quad \text{and}$$
$$u_4 = \sigma = (R_\sigma = \tilde{g}_2^{z}, S_\sigma = \tilde{f}_2^{z}, T_\sigma = \tilde{g}_2^{\tilde{u}+z(\tilde{d}+h\tilde{e})},$$
$$W_{\sigma 1} = \tilde{g}_2^{-\tilde{p}-z(\tilde{l}+h\tilde{m})}, W_{\sigma 2} = \tilde{g}_2^{-z\tilde{n}}).$$

– The challenger sets the corresponding *semifunctional* components as:

$$u_1' = vk_S' = (\tilde{g}_1, \tilde{f}_1 = \tilde{g}_1^{\tilde{b}}, \tilde{v}_1 = \tilde{g}_1^{-\tilde{l}}, \tilde{v}_2 = \tilde{g}_1^{-\tilde{m}}, \tilde{v}_3 = \tilde{g}_1^{-\tilde{n}},$$
$$\tilde{k} = e(\tilde{g}_1, \tilde{g}_2)^{-\tilde{p}}; \text{ and}$$
$$u_4' = \sigma' = (R_\sigma = \tilde{g}_2^{z}, S_\sigma = \tilde{f}_2^{z}, T_\sigma = \tilde{g}_2^{\tilde{u}+z(\tilde{d}+h\tilde{e})},$$
$$W_{\sigma 1} = \tilde{g}_2^{-\tilde{p}-\tilde{u}-z(\tilde{l}+\tilde{d}+h(\tilde{m}+\tilde{e}))/\tilde{b}}, W_{\sigma 2} = \tilde{g}_2^{-z(\tilde{n}+\tilde{c})/\tilde{b}}).$$

– Finally, the challenger outputs the (semifunctional) ciphertext

$$\mathcal{U}' = (u_1, u_2', u_3', u_4)$$

as the ciphertext.

Since $\tilde{l}, \tilde{m}, \tilde{n}, \tilde{p} \xleftarrow{\$} \mathbb{Z}_q^\times$, from the view of the adversary $\mathcal{A}$, the games $\mathcal{G}_1$ and $\mathcal{G}_2$ are statistically indistinguishable.

Game $\mathcal{G}_3$: This game is similar to the previous game in all aspects except that instead of actual tokens, the challenger outputs *partial semifunctional keys/tokens* as follows:

– Given a keyword W and the corresponding identity $i_{\text{W}} = H(1\|\text{W})$, let the corresponding public key and token be:

$$pk_{\text{W}} = (g_1, f_1 = g_1^b, v_1 = g_1^{d-bl}, v_2 = g_1^{e-bm}, v_3 = g_1^{c-bn},$$
$$k = e(g_1, g_2)^{u-bp} \text{ and}$$
$$t_{\text{W}} = (R_{\text{W}} = g_2^z, S_{\text{W}} = f_2^z, T_{\text{W}} = g_2^{u+z(d+he)},$$
$$W_{\text{W}1} = g_2^{-p-z(l+hm)}, W_{\text{W}2} = g_2^{-zn}).$$

– The challenger sets the corresponding *partial semifunctional keys* as:

$$pk'_{\text{W}} = (g_1, f_1 = g_1^b, v_1 = g_1^{-l}, v_2 = g_1^{-m}, v_3 = g_1^{-n},$$
$$k = e(g_1, g_2)^{-p} \text{ and}$$
$$t'_{\text{W}} = (R_{\text{W}} = g_2^z, S_{\text{W}} = f_2^z, T_{\text{W}} = g_2^{u+z(d+he)},$$
$$W_{\text{W}1} = g_2^{-p-u-z(l+d+h(m+e))/b}, W_{\text{W}2} = g_2^{-z(n+c)/b}).$$

Since $l, m, n, p \xleftarrow{\$} \mathbb{Z}_q^\times$, from the view of the adversary $\mathcal{A}$, the games $\mathcal{G}_2$ and $\mathcal{G}_3$ are statistically indistinguishable.

Game $\mathcal{G}_4$: This is a sequence of several hybrid games, used to generate tokens on various keywords. For $j = 0$, we define the game $\mathcal{G}_{4,0}$ to be the same as $\mathcal{G}_3$. We define the j-th hybrid game $\mathcal{G}_{4,j}$ by changing the simulation of the j-th token on the keyword W_j, and outputs a *semifunctional token* instead of the actual token as follows:

– Challenger randomly picks r_j, r'_j and r''_j and sets the token t_{W_j} for the keyword W_j as:

$$R_{\text{W}} = g_2^{r_j},\ S_{\text{W}} = g_2^{r_j c + r'_j},\ T_{\text{W}} = g_2^{r''_j + r_j \cdot (d + i_{\text{W}j} e)},$$
$$W_{\text{W}1} = g_2^{[-p' - r''_j - r_j(l' + d + i_{\text{W}j}(m' + e))]/b}, W_{\text{W}2} = g_2^{-r'_j - r_j(n' + c)/b}.$$

Observe that u has completely vanished from the j-th and earlier token responses. In view of the DDH assumption in the group G_2, it can be seen [JR13] that the view of the adversary $\mathcal{A}$ in game $\mathcal{G}_{4,j}$ is *computationally indistinguishable* from the view of the adversary $\mathcal{A}$ in game $\mathcal{G}_{4,j-1}$.

Game $\mathcal{G}_5$: This game is just the game $\mathcal{G}_{4,q}$ where q is the total number of secret key queries. Observe that in the game $\mathcal{G}_4$, the only place where u is used is in the ciphertext components $C_{\text{M}0} = \text{M} \cdot k^x \cdot e(g_1, g_2)^{u \cdot x'}$ and $C_{\text{W}0} = \text{V} \cdot k^y \cdot e(g_1, g_2)^{u \cdot y'}$. Hence $C_{\text{M}0}$ and $C_{\text{W}0}$ are completely random and independent of the target message-keyword pairs (m_0^*, w_0^*) and (m_1^*, w_1^*) in the view of the adversary $\mathcal{A}$ in the game $\mathcal{G}_5$. Note that u is non-zero with high probability. Hence the SXDH assumption implies computational indistinguishability from the chosen ciphertext adversary. That is, the scheme achieves IND-PKE+PEKS-CCA security.

6 Efficiency Analysis

In this section, we provide an efficiency comparison of various parameters in existing PEKS schemes in Table 1 and the efficiency comparison of existing PKE+PEKS schemes in Table 2.

We compare various PEKS schemes with ours in the Table 1 based on the following parameters:

- $\#pk$ – number of group elements in the public parameters
- $\#sk$ – number of group elements in the master secret
- $\#ct$ – number of group elements in the ciphertext
- (a, b, c, d) denotes a elements from G_1, b elements from G_2, c elements from G_T and d elements from Z_q where $q = |G_1|$.

Table 1. Comparison of various PEKS schemes

Scheme →	[BSS06]	[ZI07]	[CZLZ14]	[BLSS15]	Our scheme
Pairing	Type 1	Type 1	Type 1	Type 3	Type 3
Security	IK-PKE-CCA IND-PEKS-CKA	IK-PKE-CCA IND-PEKS-CKA	IND-PKE+ PEKS-CCA	IND-PKE+ PEKS-CCA	IND-PKE+ PEKS-CCA
Security model	RO	STD	STD	STD	STD
Assumption	CDH	DADHE	q-ABDHE /SDH	SXDH	SXDH
$\#pk$	$(2, -, 0, 0)$	$(3, -, 2, 0)$	$(5, -, 0, 1)$	$(8, 0, 1, 0)$	$(5, 0, 1, 0)$
$\#sk$	$(0, -, 0, 1)$	$(0, -, 0, 5)$	$(0, -, 0, 1)$	$(0, 8, 0, 1)$	$(0, 2, 0, 7)$
$\#ct$	$(1, -, 0, 3)^{\#}$	$(2, -, 3, 3)^{\dagger}$	$(6, -, 4, 1)$	$(12, 8, 3, 0)$	$(11, 5, 3, 2)$

1. * in [BDOP04], ciphertext contains one element from G_1 and one element of size $\log p$, for more detail please refer [BDOP04].
2. # in [BSS06], ciphertext contains one element from G_1 and three elements of maximum bitlength $\approx l$, where $l = \max(l_1, l_3, l_4)$; for more details please Refer to [BSS06].
3. † in [ZI07], ciphertext contains one MAC output and one element of the length of the message which we have included in the integer count; for more detail please refer [ZI07].
4. RO – Random Oracle, STD – Standard Model.
5. In the row "Assumption", the standard abbreviations like BDH – Bilinear Diffie-Hellman, CDH – Computational Diffie-Hellman are used. For details of assumptions please refer respective paper.

Finally, in Table 2, we compare the efficiency of the proposed integrated PKE+PEKS scheme with the existing PEKS and integrated PKE+PEKS schemes [BSS06, BLSS15, CZLZ14, ZI07] and show that our scheme is more efficient than these schemes. In each of the four phases: Encryption, Decryption, Token Gen. and Test, we compare the total number of bilinear pairings (P),

Table 2. Efficiency comparison

Operation	Scheme	P	E(Z_q)	I(Z_q)	E(G_1)	M(G_1)	E(G_2)	M(G_2)	E(G_T)	M(G_T)
Encryption	[BSS06]	1	0	0	2	0	-	-	1	0
	[ZI07]	2	0	0	2	1	-	-	6	1
	[CZLZ14]	5	0	1	7	2	-	-	5	2
	[BLSS15]	1	0	0	16	0	12	0	3	2
	Our scheme	1	0	0	15	4	6	0	3	2
Decryption	[BSS06]	0	0	0	1	0	-	-	0	0
	[ZI07]	0	0	0	0	0	-	-	2	1
	[CZLZ14]	3	0	1	3	2	-	-	0	1
	[BLSS15]	8	0	0	4	4	4	0	0	7
	Our scheme	6	0	0	5	2	9	4	1	7
Token Gen	[BSS06]	0	0	0	1	0	-	-	0	0
	[ZI07]	0	0	0	1	1	-	-	0	0
	[CZLZ14]	0	0	1	1	1	-	-	0	0
	[BLSS15]	0	0	0	0	0	4	0	0	0
	Our scheme	0	0	0	0	0	5	0	0	0
Test	[BSS06]	1	0	0	0	0	-	-	0	0
	[ZI07]	1	0	0	0	0	-	-	1	1
	[CZLZ14]	4	1	0	2	2	0	0	2	3
	[BLSS15]	8	0	0	4	4	0	0	0	7
	Our scheme	6	0	0	5	2	9	4	1	7
Overall comparison	[BSS06]	2	0	0	4	1	-	-	1	0
	[ZI07]	3	0	0	3	2	-	-	9	3
	[CZLZ14]	12	1	3	13	7	-	-	7	6
	[BLSS15]	17	0	0	24	8	20	8	3	16
	Our scheme	13	0	0	25	8	29	8	5	16

exponentiations and inverse in Z_q denoted as $E(Z_q)$ and $I(Z_q)$, exponentiations and multiplications in G_1 (resp. G_2 and G_T) denoted as $E(G_1)$ (resp. $E(G_2)$ and $E(G_T)$) and $M(G_1)$ (resp. $M(G_2)$ and $M(G_T)$). Since [BLSS15] is the only construction of PEKS other than ours with *asymmetric pairing*, that is, Type 3 pairing ($e : G_1 \times G_2 \rightarrow G_T$), for these schemes, we have considered operations in all the three different groups, that is, in G_1, G_2 and G_T, and since all the previous schemes use symmetric pairings, that is, Type 1 pairing ($e : G_1 \times G_1 \rightarrow G_T$) [GPS08], we have counted operations in groups G_1 and G_2 only for these schemes, considering $|G_1| \approx |G_2|$.

From the efficiency comparison Table 1, it is evident that the proposed integrated PKE+PEKS scheme is (computationally) more efficient than the schemes given in [BSS06, ZI07, CZLZ14, BLSS15]. Note that first two schemes [BSS06, ZI07] provide only CPA security so they are naturally a bit more efficient. The third scheme [CZLZ14] uses symmetric pairings and even though the numbers in some cells of the table show smaller number of operations, the operations are much more expensive in their case. Finally, in the fourth scheme [BLSS15] the smaller numbers in some cells are adequately compensated by the smaller number of pairings in our scheme.

7 Conclusion

We have proposed an efficient and practical integrated PKE+PEKS scheme and proved its security in the strongest security notion for PKE+PEKS schemes. The security of our scheme relies on SXDH assumption which is a much simpler and more standard hardness assumption than the ones used in most of the comparable schemes. Ours is the first fully secure integrated PKE+PEKS scheme using asymmetric pairings which enable an extremely fast implementation useful for practical applications. Finally, providing a relative analysis of parameters, assumptions, securities and efficiency, we have compared our scheme with the existing similar schemes and shown that our scheme is more efficient than those schemes.

Acknowledgements. We thank the anonymous reviewers for the constructive and helpful comments. We thank Francesco Buccafurri and Gianluca Lax for the useful discussions. We are thankful to Olivier Markowitch for the support.

References

[ABC+05] Abdalla, M., Bellare, M., Catalano, D., Kiltz, E., Kohno, T., Lange, T., Malone-Lee, J., Neven, G., Paillier, P., Shi, H.: Searchable Encryption Revisited: Consistency Properties, Relation to Anonymous IBE, and Extensions. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 205–222. Springer, Heidelberg (2005). https://doi.org/10.1007/11535218_13

[ABN10] Abdalla, M., Bellare, M., Neven, G.: Robust encryption. In: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 480–497. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-11799-2_28

[BDOP04] Boneh, D., Di Crescenzo, G., Ostrovsky, R., Persiano, G.: Public key encryption with keyword search. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 506–522. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_30

[BDPR98] Bellare, M., Desai, A., Pointcheval, D., Rogaway, P.: Relations among notions of security for public-key encryption schemes. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 26–45. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0055718

[BF01] Boneh, D., Franklin, M.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_13

[BLSS15] Buccafurri, F., Lax, G., Sahu, R.A., Saraswat, V.: Practical and secure integrated PKE+PEKS with keyword privacy. In: SECRYPT, pp. 448–453. SciTePress (2015)

[BSS06] Baek, J., Safavi-Naini, R., Susilo, W.: On the integration of public key data encryption and public key encryption with keyword search. In: Katsikas, S.K., López, J., Backes, M., Gritzalis, S., Preneel, B. (eds.) ISC 2006. LNCS, vol. 4176, pp. 217–232. Springer, Heidelberg (2006). https://doi.org/10.1007/11836810_16

[BSS08] Baek, J., Safavi-Naini, R., Susilo, W.: Public key encryption with keyword search revisited. In: Gervasi, O., Murgante, B., Laganà, A., Taniar, D., Mun, Y., Gavrilova, M.L. (eds.) ICCSA 2008. LNCS, vol. 5072, pp. 1249–1259. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-69839-5_96

[BW06] Boyen, X., Waters, B.: Anonymous hierarchical identity-based encryption (without random oracles). In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 290–307. Springer, Heidelberg (2006). https://doi.org/10.1007/11818175_17

[CLL+12] Chen, J., Lim, H.W., Ling, S., Wang, H., Wee, H.: Shorter IBE and signatures via asymmetric pairings. In: Abdalla, M., Lange, T. (eds.) Pairing 2012. LNCS, vol. 7708, pp. 122–140. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36334-4_8

[CZLZ14] Chen, Y., Zhang, J., Lin, D., Zhang, Z.: Generic constructions of integrated PKE and PEKS. In: Designs, Codes and Cryptography, pp. 1–34 (2014)

[DK05] Dodis, Y., Katz, J.: Chosen-ciphertext security of multiple encryption. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 188–209. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-30576-7_11

[DS07] Di Crescenzo, G., Saraswat, V.: Public key encryption with searchable keywords based on Jacobi symbols. In: Srinathan, K., Rangan, C.P., Yung, M. (eds.) INDOCRYPT 2007. LNCS, vol. 4859, pp. 282–296. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-77026-8_21

[FP07] Fuhr, T., Paillier, P.: Decryptable searchable encryption. In: Susilo, W., Liu, J.K., Mu, Y. (eds.) ProvSec 2007. LNCS, vol. 4784, pp. 228–236. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-75670-5_17

[GM84] Goldwasser, S., Micali, S.: Probabilistic encryption. J. Comput. Syst. Sci. **28**(2), 270–299 (1984)

[GPS08] Galbraith, S.D., Paterson, K.G., Smart, N.P.: Pairings for cryptographers. Discrete Appl. Math. **156**(16), 3113–3121 (2008). Applications of Algebra to Cryptography

[INHJ11] Ibraimi, L., Nikova, S., Hartel, P., Jonker, W.: Public-key encryption with delegated search. In: Lopez, J., Tsudik, G. (eds.) ACNS 2011. LNCS, vol. 6715, pp. 532–549. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21554-4_31

[JR13] Jutla, C.S., Roy, A.: Shorter Quasi-adaptive NIZK proofs for linear subspaces. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013. LNCS, vol. 8269, pp. 1–20. Springer, Heidelberg (2013)

[Kur02] Kurosawa, K.: Multi-recipient public-key encryption with shortened ciphertext. In: Naccache, D., Paillier, P. (eds.) PKC 2002. LNCS, vol. 2274, pp. 48–63. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45664-3_4

[PSST11] Paterson, K.G., Schuldt, J.C.N., Stam, M., Thomson, S.: On the joint security of encryption and signature, revisited. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 161–178. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25385-0_9

[SR14] Strizhov, M., Ray, I.: Multi-keyword Similarity Search over Encrypted Cloud Data. In: Cuppens-Boulahia, N., Cuppens, F., Jajodia, S., Abou El Kalam, A., Sans, T. (eds.) SEC 2014. IAICT, vol. 428, pp. 52–65. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-55415-5_5

[SVEG10] Shmueli, E., Vaisenberg, R., Elovici, Y., Glezer, C.: Database encryption: an overview of contemporary challenges and design considerations. ACM SIGMOD Rec. **38**(3), 29–34 (2010)

[Wat09] Waters, B.: Dual system encryption: realizing fully secure IBE and HIBE under simple assumptions. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 619–636. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03356-8_36

[ZI07] Zhang, R., Imai, H.: Generic combination of public key encryption with keyword search and public key encryption. In: Bao, F., Ling, S., Okamoto, T., Wang, H., Xing, C. (eds.) CANS 2007. LNCS, vol. 4856, pp. 159–174. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-76969-9_11