9-2017

IND-PCA secure KEM is enough for password-based authenticated key exchange (short paper)

Haiyang XUE
Singapore Management University, haiyangxue@smu.edu.sg

Bao LI

Xianhui LU

Citation

XUE, Haiyang; LI, Bao; and LU, Xianhui. IND-PCA secure KEM is enough for password-based authenticated key exchange (short paper). (2017). *Proceedings of the 12th International Workshop on Security, IWSEC 2017 Hiroshima, Japan, August 30 - September 1,*. 231-241.
Available at: https://ink.library.smu.edu.sg/sis_research/9191

IND-PCA Secure KEM Is Enough for Password-Based Authenticated Key Exchange (Short Paper)

Haiyang Xue[1,2(✉)], Bao Li[1,2,3], and Xianhui Lu[1,2]

[1] Data Assurance and Communication Security Research Center, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2] Science and Technology on Communication Security Laboratory, Chengdu, China
{hyxue12,lb,xhlu}@is.ac.cn
[3] University of Chinese Academy of Sciences, Beijing, China

Abstract. There are several frameworks for password-based authenticated key exchange (PAKE) protocols with common reference string following the work of Katz, Ostrovsky and Yung (Eurocrypt'01), and it seems that the IND-CCA secure encryption is inevitable when constructing PAKE in standard model.

In this paper, we show that IND-PCA secure key encapsulation mechanism (KEM) is enough for PAKE, which is weaker and easier to be constructed than IND-CCA secure encryption. Our refined PAKE consists of a smooth projective hash function on IND-CPA secure encryption and an IND-PCA secure KEM. Based on DDH assumption, the total communication of PAKE consists of 6 group elements and $\log |D|$ (D is the set of password) bits, while before this, the most efficient PAKE contains 7 group elements.

Keywords: Password-based authenticated key exchange · Smooth projective hash functions · IND-PCA secure KEM

1 Introduction

Password-based authenticated key exchange (PAKE) allows two users to mutually authenticate each other and agree on a high-entropy session key based on a shared low-entropy password. The challenge in designing such protocols is to prevent *off-line* dictionary attacks where an adversary exhaustively enumerates potential passwords, attempting to match the correct password. The secure goal of PAKE is to restrict the adversary's advantage to that of *online* dictionary attack. The seminal work in the area of PAKE was given by Bellovin and Merritt [2]. After that, Bellare et al. [4], and Boyko et al. [3] proposed formal security models for PAKE. Since then, a large number of constructions were presented in the

S. Obana and K. Chida (Eds.): IWSEC 2017, LNCS 10418, pp. 231–241, 2017.
DOI: 10.1007/978-3-319-64200-0_14

random oracle model [3,4]. But the random oracle model is known to be not sound [5].

The first PAKE protocol to achieve security in standard model was given by Goldreich and Lindel [8]. There are several works to improve and simplify Goldreich and Lindel's scheme. Unfortunately, they are inefficient in terms of communication, computation and round complexity. Katz, Ostrovsky and Yung [14] demonstrated the first efficient PAKE (KOY) under DDH assumption with common reference string (CRS). On the ground of concrete construction of KOY protocol, a framework of PAKE (GL-PAKE) was abstracted by Gennaro and Lindell [9]. GL-PAKE consists of two smooth projective hash functions (SPHFs) on chosen ciphertext secure (IND-CCA) encryption. Following the work of KOY, Jiang and Gong [12] improved and gave a PAKE with mutual authentication under DDH assumption. Groce and Katz [10] abstracted the protocol of Jiang and Gong's protocol and gave a framework of GK-PAKE with SPHF on IND-CPA secure encryption and IND-CCA secure encryption.

Recently, Abdalla, Benhamouda and Pointcheval [1] pointed out that the underlying IND-CCA secure encryption in GL-PAKE and GK-PAKE can be replaced by IND-PCA secure encryption, (the adversary has the capability to query plaintext checkable oracle with (C, m) to help him to decide if C is the encryption of m or not) and the Cramer-Shoup scheme in PAKE can be simplified. As Abdalla et al. pointed out, IND-PCA secure encryption with short plaintext is actually IND-CCA secure. Since password (in Addalla et al.'s scheme, password is in the part of plaintext) is generally short in PAKE, the framework of PAKE in Abdalla et al. essentially relies on an IND-CCA secure encryption.

Refined Structure for PAKE. One of the most important work in cryptography is reducing security to more basic and weaker tools. This is what this paper does. In the above works, IND-CCA secure encryption scheme seems inevitable. Although there are many efficient constructions for IND-CCA secure scheme [11,15,16], this requirement is still too strong. It is meaningful to see whether there is an elegant framework to construct efficient PAKE protocol based on more basic and weaker tools.

1.1 Our Contributions

In this paper, we revisit the framework of PAKE in [10], and show that SPHF on IND-CPA secure encryption and any IND-PCA secure Key encapsulation mechanism (KEM) with short encapsulation key space is enough for PAKE. In our PAKE, the key encapsulated by KEM are used to encrypt password with one time padding. Obviously, the hybrid encryption from IND-PCA secure KEM and one time padding is not IND-CCA secure[1] (even not IND-PCA secure). Note that although the hybrid encryption is malleable, it does not hurt the security of PAKE. The adversary can only produce a meaningful plaintext by extending

[1] Let $(c, k) \leftarrow \mathsf{Enc}_{\mathsf{kem}}(pk, \lambda)$, the hybrid encryption of m is the form $(c, k \oplus m)$. It is malleable and any adversary can reproduce the ciphertext with meaningful plaintext after seeing the challenge ciphertext.

the plaintext (password), which does not add its advantage in attacking PAKE. As a by product, we show that the KEM given by Kurosawa and Desmedt [13] is IND-PCA secure, which is proved to be not IND-CCA secure [7].

Besides that, we also give concrete example based on DDH assumption and obtain a scheme with a total communication of 6 group elements and $\log|D|$ bits instead of 7 group elements in [1], and without the requirement of mutual authentication, only 5 group elements and $\log|D|$ bits are needed.

2 Preliminaries

If S is a set, we denote by $|S|$ the cardinality of S, and denote by $x \leftarrow S$ the process of sampling x uniformly from S. A function is *negligible* (negl) if for every $c > 0$ there exists a λ_c such that $f(\lambda) < 1/\lambda^c$ for all $\lambda > \lambda_c$. If A and B are distributions, $A \approx_s B$ means that the statistical distance between A and B is negligible.

We recall the definition of smooth projective hash function given in [6]. We first recall the definition of subset membership assumption (multiple versions of this assumption have appeared) following [15].

Definition 1 (Subset Membership Assumption [15]). *Let $L \subset X$ and L is called the set of YES instance, and $X\backslash L$ the set of NO instance. There are efficient sample algorithms SampY (SampN) for YES(NO) instance. For any PPT adversary A, the advantage function $Adv_A^{SMA} = |Pr[A(PP, x) = 1 : x \leftarrow L] - Pr[A(PP, x) = 1 : x \leftarrow X\backslash L]|$ is negligible.*

Definition 2 (Smooth Projective Hash Function [6]). *We assume here all the algorithms can access PP. The smooth projective hash function on $(X, X\backslash L)$ follows.*

- $\mathsf{HashKG}(PP)$ *generates a hashing key $k \in K$.*
- $\mathsf{ProjKG}(k)$ *generates the projective key $\alpha(k)$.*
- $\mathsf{Hash}(k, x)$ *outputs the hash value on any $x \in X$ from the hashing key k.*
- $\mathsf{ProjHash}(\alpha(k), w, x)$*: On input the witness w for any $x \in L$ and the projective key, outputs the hash value, such that $\mathsf{ProjHash}(\alpha(k), w, x) = \mathsf{Hash}(k, x)$.*

We say that it is smooth, if the following distributions are statistically indistinguishable: $\Delta(\{x, \alpha(k), \mathsf{Hash}(k, x)\}, \{x, \alpha(k), u\}) \leq \varepsilon$, where $k \in K$, $x \in X\backslash L$, and $u \in \Pi$ are chosen randomly.

3 Refined Framework for PAKE

As the main modification of our PAKE is the IND-PCA secure KEM, we first recall the definition of IND-PCA secure KEM and prove that the Kurosawa-Desmedt KEM in [15] are IND-PCA secure. After that, we show our refined framework of PAKE.

3.1 IND-PCA Secure KEM

We first recall the definition of (label based) KEM. For any KEM without label, we just set $\mathsf{label} = \perp$. A (label based) public key encapsulation scheme $KEM = (\mathsf{KGen}_{\mathsf{kem}}, \mathsf{Enc}_{\mathsf{kem}}, \mathsf{Dec}_{\mathsf{kem}})$ consists of three polynomial time algorithms, where $(pk_{\mathsf{kem}}, sk_{\mathsf{kem}}) \leftarrow \mathsf{KGen}_{\mathsf{kem}}(\lambda)$ produces keys for security parameter λ; for randomness r, $(K, C) \leftarrow \mathsf{Enc}_{\mathsf{kem}}(pk_{\mathsf{kem}}, \mathsf{label}, r)$ produces a key K in $KeySp(\lambda)$ together with a ciphertext C to recover the key; and $K \leftarrow \mathsf{Dec}_{\mathsf{kem}}(sk_{\mathsf{kem}}, \mathsf{label}, C)$ decapsulates ciphertext C with label to recover K with secret key sk_{kem}. For all $(K, C) \leftarrow \mathsf{Enc}_{\mathsf{kem}}(pk_{\mathsf{kem}}, \mathsf{label}, r)$, $\Pr[\mathsf{Dec}_{\mathsf{kem}}(sk_{\mathsf{kem}}, \mathsf{label}, C) = K] = 1$, where the probability is taken over the randomness of these three algorithms.

In our PAKE, we need a weak secure notion of KEM, namely security against plaintext checkable attack (PCA) [1]. Formally, for any PPT algorithm A, a KEM is said to be IND-PCA secure if the following advantage is negligible,

$$\mathbf{Adv}_A^{kem-pca} = \Pr\left[b = b' : \begin{array}{c} b \leftarrow \{0,1\}; (pk_{\mathsf{kem}}, sk_{\mathsf{kem}}) \leftarrow \mathsf{KGen}_{\mathsf{kem}}(k); \\ K_0^* \leftarrow KeySp(k), (K_1^*, C^*) \leftarrow \mathsf{Enc}_{\mathsf{kem}}(pk, \mathsf{label}), \\ b' \leftarrow A^{\mathsf{DCheck}(\cdot,\cdot)}(pk, K_b^*, C^*) \end{array}\right],$$

where the oracle $\mathsf{DCheck}(C, K)$ returns 1 if $K = \mathsf{Dec}_{\mathsf{kem}}(sk_{\mathsf{kem}}, C)$, otherwise returns 0, and the adversary A can not query DCheck with $(C, K) = (C^*, K_b^*)$.

The KEM part of Kurosawa-Desmedt scheme [13] is known to be not IND-CCA secure [7]. In the following, we prove that the KEM part of Kurosawa-Desmedt scheme is IND-PCA secure. We first recall the Kurosawa-Desmedt KEM. Let G be a group of prime order p and let g_1, g_2 be two public generators of G. Let $h_{tcr} : G \times G \rightarrow \mathbb{Z}_p$ be a target collision-resistant hash function. The key encapsulation part of the Kurosawa-Desmedt scheme is as follows:

$\mathsf{KGen}_{\mathsf{kem}}(1^n)$	$\mathsf{Enc}_{\mathsf{kem}}(pk)$;	$\mathsf{Dec}_{\mathsf{kem}}(sk, c)$
$x_1, x_2, y_1, y_2 \leftarrow \mathbb{Z}_p^*$;	$r \leftarrow \mathbb{Z}_p$	$(c_1, c_2) \leftarrow c$;
$h_1 = g_1^{x_1} g_2^{x_2}$, $h_2 = g_1^{y_1} g_2^{y_2}$	$c_1 = g_1^r, c_2 = g_2^r$;	$t = h_{tcr}(c_1, c_2)$
$pk := (g_1, g_2, h_1, h_2)$;	$t = h_{tcr}(c_1, c_2)$, $K = h_1^{tr} h_2^r$	$K = c_1^{tx_1+y_1} c_2^{tx_2+y_2}$
$sk := (x_1, x_2, y_1, y_2)$.	Return (c_1, c_2, K).	

Theorem 1. *If h_{tcr} is a collision resistant hash function, under the DDH assumption, the Kurosawa-Desmedt KEM is IND-PCA secure.*

Proof. The proof proceeds via a sequence of games.

Game 0. The adversary A is given the public key as well as an unlimited access to an Dcheck oracle with (C, K). At some point, the adversary receives an encapsulation $C^* = (c_1^*, c_2^*)$ and K_b^*. After some training on DCheck oracle, A outputs a guess of b. The ciphertext C^* is generated normally with r. Precisely speaking, $c_1^* = g_1^r$, $c_2^* = g_2^r$. On receiving $(C = (c_1, c_2), K)$, the DCheck oracle checks it using secret key. We have that $\mathbf{Adv}_{A,G_0} = \mathbf{Adv}_A^{ind-pca}$.

Game 1. In this game, the DCheck oracle rejects all queries where $C \neq C^*$ but $h_{tcr}(C) = h_{tcr}(C^*)$. This game is computationally indistinguishable from the previous one under the collision-resistance of h_{tcr}.

Game 2. We now change the way of generating challenge key. The key K_1^* encapsulated is generated as $(c_1^*)^{t^* x_1 + y_1}(c_2^*)^{t^* x_2 + y_2}$, where $t^* = h_{tcr}(c_1^*, c_2^*)$. The Game 2 is exactly the same with Game 1.

Game 3. We now change the generation algorithm of challenge ciphertext C^*. $r_1, r_2 \leftarrow \mathbb{Z}_p$ and let $c_1^* = g_1^{r_1}, c_2^* = g_1^{r_2}$. The difference between Game 3 and Game 2 is bounded by the DDH assumption. Note that the randomness for c_1^* and c_2^* do not needed to generate the challenge key K_1^*, we can perfectly simulate the game given a DDH challenge. Thus the difference between Game 3 and Game 2 is bounded by DDH assumption.

Game 4. In this Game, the simulator holds the secret a s.t. $g_2 = g_1^a$ during the key generation algorithm. The DCheck oracle rejects all queries (c_1, c_2, K) where $c_2 \neq c_1^a$ with the knowledge of a. It can make a difference when $c_2 \neq c_1^a$ but $K = (c_1)^{tx_1 + y_1}(c_2)^{tx_2 + y_2}$. First, if $(c_1, c_2) = (c_1^*, c_2^*)$ but $K \neq K_1^*$, since that implies $t = t^*$, we can safely answer negatively. We thus now have to deal with the cases $(c_1, c_2) \neq (c_1^*, c_2^*)$ and $K = (c_1)^{tx_1 + y_1}(c_2)^{tx_2 + y_2}$.

Consider the map $f(x_1, x_2, y_1, y_2) = (h_1, h_2, K_1^*, K)$ mapping hashing secret keys. If we show that this map is injective then we are done.

$$\begin{cases} \log_{g_1} h_1 = x_1 + a \cdot x_2 \\ \log_{g_1} h_2 = y_1 + a \cdot y_2 \\ \log_{g_1} K_1^* = r_1(t^* x_1 + y_1) + a r_2(t^* x_2 + y_2) \\ \log_{g_1} K = r_1(t x_1 + y_1) + a r_2(t x_2 + y_2) \end{cases}$$

$$\begin{pmatrix} \log_{g_1} h_1 \\ \log_{g_1} h_2 \\ \log_{g_1} K_1^* \\ \log_{g_1} K \end{pmatrix} = \begin{pmatrix} 1 & 0 & a & 0 \\ 0 & 1 & 0 & a \\ r_1^* t^* & r_1^* & a r_2^* t^* & a r_2^* \\ r_1 t & r_1 & a r_2 t & a r_2 \end{pmatrix} \times \begin{pmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \end{pmatrix}$$

Since $det(A) = a^2(r_2^* - r_1^*)(r_2 - r_1)(t^* - t)$, f is injective if $r_2^* \neq r_1^*$, $r_2 \neq r_1$. The two assumption holds as both the challenge ciphertext and query ciphertext are not DDH subset member.

Game 5. In this game, K_1^* is randomly chosen from G. This is statically indistinguishable form Game 4. Now, the challenge ciphertext doesn't contain any information of b. To sum up, we finish this proof. □

Remark 1. The Kurosawa-Desmedt method also works for a more general class of universal 2 hash proof systems on subset membership problem [6]. As shown in next subsection, in the application of PAKE, the key with low entropy in KEM (long enough to extract $\log |D|$ bits) is enough.

3.2 PAKE from IND-PCA Secure KEM

We now present the refined framework for PAKE. As the space limits, we omit the secure definition of BPR model [4] with mutual authentication which is added

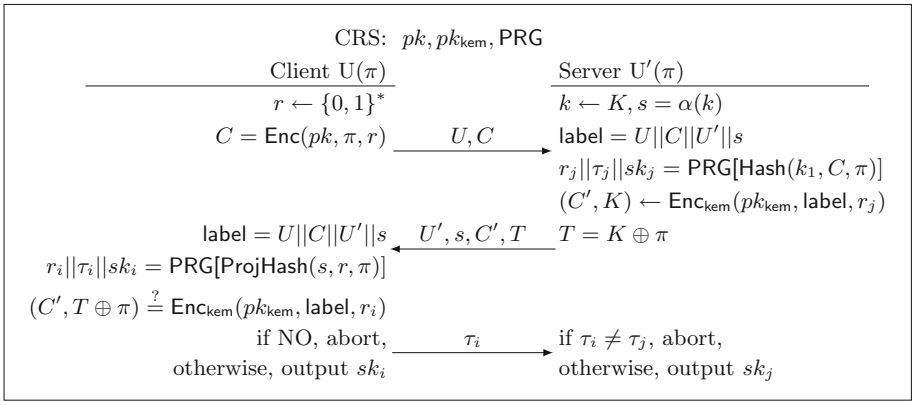

Fig. 1. Refined framework of PAKE

by [10]. For more details, please refer [10]. In this construction, the following primitives are required: A SPHF on IND-CPA secure encryption; An IND-PCA secure KEM with short key space. Let $PKE = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ be an IND-CPA secure encryption. Take the ciphertext space as X, and the ciphertexts of π as L, Let SPHF be a SPHF on it. Let $KEM = (\mathsf{KGen}_{\mathsf{kem}}, \mathsf{Enc}_{\mathsf{kem}}, \mathsf{Dec}_{\mathsf{kem}})$ be an IND-PCA secure KEM with $KeySp = D$.

Initialization: The CRS consists of public keys pk for IND-CPA secure scheme PKE and public keys pk_{kem}. Let PRG be a pseudorandom generator.

Protocol execution. Figure 1 demonstrates the execution of the protocol.

Stage 1: When a client U (holds π) wants to authenticate to the server U' (holds π), it chooses $r \leftarrow \{0,1\}^*$, computes $C = \mathsf{Enc}(pk, \pi, r)$, and sends $U||C$ to U'.

Stage 2: On receiving the message $U||C$, U' chooses $k \leftarrow K$ and computes $s = \alpha(k)$ and $\mathsf{Hash}(k, C, \pi)$. It decomposes the PRG value on $\mathsf{Hash}(k, C, \pi)$ as three bit strings r_j, τ_j, sk_j. It sets $\mathsf{label} = U||C||U'||s$, computes $(C', K) \leftarrow \mathsf{Enc}_{\mathsf{kem}}(pk_{\mathsf{kem}}, \mathsf{label}, r_j)$ and $T = K \oplus \pi$. Then U' sends $U'||s||C'||T$ to U.

Stage 3: On receiving the message $U'||s||C'|T$, user U computes and decomposes the hash value $r_i||\tau_i||sk_i \leftarrow \mathrm{PRG}[\mathsf{ProjHash}(s, r, \pi)]$. It sets $\mathsf{label} = (U||C||U'||s)$ and checks $(C', T \oplus \pi) \stackrel{?}{=} \mathsf{Enc}_{\mathsf{kem}}(pk_{\mathsf{kem}}, \mathsf{label}, r_i)$. If no, aborts else sends τ_i to U' and outputs sk_i which means that U' has successfully authenticated to U.

Stage 4: On receiving the message τ_i, U' checks that if $\tau_i = \tau_j$ or not. If $\tau_i \neq \tau_j$, U' aborts, otherwise U has successfully authenticated to U' and U' outputs sk_j.

If both parties are honest and there is no adversarial interference, then the projection of the hash proof guarantees that it holds $r_i||\tau_i||sk_i = r_j||\tau_j||sk_j$. Both parties will accept and output the same session key.

Theorem 2. *Assume PKE is an IND-CPA secure encryption scheme, SPHF is a ε_{smooth} SPHF over PKE, and KEM is an IND-PCA secure KEM, the*

PAKE is secure in the BPR model. In particular, let q_e be the number of Execute queries, q_s be the number of Send queries, and $q_e + q_s \leq t$, we have

$$\boldsymbol{Adv}_{A,\Pi}(n) \leq \frac{1}{D} + t\boldsymbol{Adv}^{CPA}_{B,Enc} + t\varepsilon_{smooth} + t\boldsymbol{Adv}^{kem-pca}_{E}.$$

Proof. The proof proceeds via a sequence of experiments. Let "G_i" denote the sequence of experiments and denote the advantage of adversary A in "G_i" as $\mathbf{Adv}_{A,G_i}(n) = 2\mathbf{Pr}[A \text{ succeeds in } G_i] - 1$. Let G_0 be the experiment of BPR challenge.

The proof is separated into two phases: the first phase (from G_1 to G_5) bounds out the advantage of *Execute* queries, and the second phase (from G_6 to G_{10}) bounds out the advantage of *Send* queries.

Experiment G_1. We first modify the way *Execute* queries between two users are answered. The hash value is computed using hashing key k instead of witness r in the client side. This does not change anything as the correctness of SPHF. We have that $\mathbf{Adv}_{A,G_0}(n) = \mathbf{Adv}_{A,G_1}(n)$

Experiment G_2. We replace C by the encryption of π_0 rather than π, where π_0 represent some password not in D. This is indistinguishable from G_1 under the IND-CPA property of PKE.

We first assume that only one *Execute* query is allowed. We now construct an IND-CPA attacker B using a distinguisher A of G_1 and G_2. In the IND-CPA game of PKE, on receiving public key pk, B generates real password π and fake password π_0 as challenge message. On receiving challenge ciphertext C^*, B simulates the entire game for A, including the KEM and so on (note that the randomness r for C is not needed now). In response to the *Execute* query, it returns the challenge ciphertext C^* which is the encryption of π or π_0. At the end, B outputs 1 if A succeeds. The advantage of B is exactly the difference between G_1 and G_2. If q_e is the bound of the number of *Execute* queries, using the classical hybrid technique, the difference between G_1 and G_2 is bounded by $q_e\mathbf{Adv}^{CPA}_{B,Enc}$

Experiment G_3. We replace the hash value by truly random elements in Π in *Execute* query. Since when answering the *Execute* queries in G_2 the ciphertext in the first message is an encryption of π_0, the hash value is statistically close to uniform even conditional on s. Using the hybrid technique, we have that $\mathbf{Adv}_{A,G_2}(n) - \mathbf{Adv}_{A,G_3}(n) \leq q_e\varepsilon_{smooth}$.

Experiment G_4. Here, we continue to modify the *Execute* query. The key generated by $\mathsf{Enc}_{\mathsf{kem}}$ is replaced by a random key in $KeySp$.

The indistinguishability between G_3 and G_4 is bound by the IND-PCA security of KEM (actually, the IND-CPA security of KEM is enough). We now construct an IND-PCA attacker E using a distinguisher A of G_3 and G_4. In the IND-PCA game of KEM, on receiving public key pk_{kem}, D generates real password π and fake password π_0 as challenge message. On receiving challenge ciphertext and key C^*, K^*, E simulates the entire game for A, including the PKE and so on. In response to the *Execute* query, it returns the challenge

ciphertext and key C^*, K^*, where K^* is the encapsulated by C^* (corresponding to G_3) or a random key (corresponding to G_4). At the end, E outputs 1 if A succeeds. The advantage of E is exactly the difference between G_3 and G_4. If q_e is the upper bound of $Execute$ queries, by the hybrid technique, $\mathbf{Adv}_{A,G_3}(n) - \mathbf{Adv}_{A,G_4}(n) \leq q_e Adv_D^{kdm-pca}$.

Experiment G_5. Here when answering an $Exacute$ query, T is replaced by a random string. Obviously G_5 and G_6 is exactly same.

Now the answers of $\underline{Execute}$ queries reveal no information of actual password. We handle the $\underline{Send}$ queries in the following experiments. Let $Send_0$ denote sending the prompt message that causes the client U to initiate the protocol with U'. Let $Send_1$ denote sending the first message, $Send_2$ denote sending the second message, $Send_3$ denote sending the finial message.

Experiment G_6. On answering the $Send_2(U'||s||x||T)$ queries, we do not use r_i to check (C', T) but query the Dcheck oracle with $(C', T \oplus \pi)$. If $U'||s||C'||T$ is not previously used, and Dcheck returns 1, we declare the attacker successful. This just adds the advantage of adversary. We have that $\mathbf{Adv}_{A,G_5}(n) \leq \mathbf{Adv}_{A,G_6}(n)$.

Experiment G_7. On answering $Send_0$ queries, we replace $C = Enc(pk, \pi, r)$ by the encryption of π_0.

Note that the smooth hash value on instance C is not needed to simulate the entire experiment now. We now construct an IND-CPA attacker B using a distinguisher A of G_6 and G_7. In the IND-CPA game of PKE, on receiving public key pk, B generates real password π and fake password π_0 as challenge message. On receiving challenge ciphertext C^*, B can simulate the entire game for A, including the HPS and KEM (note that the randomness r for C^* is not needed now). In response to the $Send_0$ query, it returns the challenge ciphertext C^* which is the encryption of π or π_0. At the end, B outputs 1 if A succeeds. The advantage of D is exactly the difference between G_6 and G_7.

If q_s is the upper bound of $Send$ queries, using the classical hybrid technique, the difference between G_6 and G_7 is bounded by $q_s \mathbf{Adv}_{D,Enc}^{CPA}$.

Experiment G_8. On answering the $Send_1(U||C)$ queries, we decrypt C using sk and clear success if $\pi = Dec(sk, C)$. This just adds the advantage of adversary. We have that $\mathbf{Adv}_{A,G_7}(n) \leq \mathbf{Adv}_{A,G_8}(n)$.

Experiment G_9. Here we again modify the answer of the $Send_1$ oracle. In response to a query $Send_1(U||C)$ we check whether $\pi = Dec(sk, C)$ or not as in experiment G_9. If so, the adversary is declared to succeed as before. If not, however, we now choose the hash value uniformly and thus r_j, τ_j and sk_j at random (rather than compute these values as the output of $\mathrm{PRG}(\mathsf{Hash}(k, C, \pi))$, and then continue as before. In particular, if there is a subsequent $Send_3$ query using the correct value of τ_j, the server accepts and outputs the session key sk_j. By the classical hybrid technique, we have that $\mathbf{Adv}_{A,G_8}(n) - \mathbf{Adv}_{A,G_9}(n) \leq q_s \varepsilon$.

Experiment G_{10}. We continue to change the answer of $Send_1$ queries. If $\pi \neq Dec(sk, C)$, the hash value is chosen uniformly as before, but after $(C', K) \leftarrow \mathsf{Enc}_{\mathsf{kem}}(pk_{\mathsf{kem}}, \mathsf{label}, r_j)$, we set $K \leftarrow KeySp$. The difference between

G_9 and G_{10} is bounded by the advantage of the IND-PCA attack on the KEM. $\mathbf{Adv}_{A,G_9}(n) - \mathbf{Adv}_{A,G_{10}}(n) \leq q_s \mathbf{Adv}_E^{kdm-pca}$.

In the final experiment, the adversary succeeds in four cases: (1) $Send_1(U', U||C)$ is queried, such that $Dec(sk, C) = \pi$; (2) $Send_2(U, U'||s||C'||T)$ is queried, such that $\mathsf{DCheck}(C', T \oplus \pi) = 1$. (3) $Send_3(\tau)$ is queried, such that $\tau = \tau_j$. (4) The adversary successfully guesses that bit used by the Test oracle.

Note that the execution of the experiment 10 is independent of password π. $\Pr[success] \leq \frac{1}{2} + \frac{1}{D}$. And so, $\mathbf{Adv}_{A,G_{10}}(n) \leq \frac{q}{D}$. By summing up all the gap advantages, $\mathbf{Adv}_{A,\Pi}(n) \leq \frac{1}{D} + t\mathbf{Adv}_{B,Enc}^{CPA} + t\varepsilon_{smooth} + t\mathbf{Adv}_D^{kem-pca}$. □

4 Instantiation and Efficiency Comparison

We instantiate the framework in Sect. 3 based on DDH assumption and subgroup membership assumptions (SGA). In case of DDH, we get a scheme with communication complexity of 6 group elements and $\log|D|$ bits; in case of SGA, we obtain a scheme with 4 group elements and $\log|D|$ bits.

Please refer Fig. 2 for the PAKE based on DDH assumption. The SPHF over ElGamal is that given in [10,12]. The KEM here is the one in [15] that improved [13] with 4-wise independent hash function rather than collision resistant hash function, and the only difference is that the length of key encapsulated is only $\log|D|$. Meanwhile, let $H_4 : \{0,1\}^* \times G \to D$.

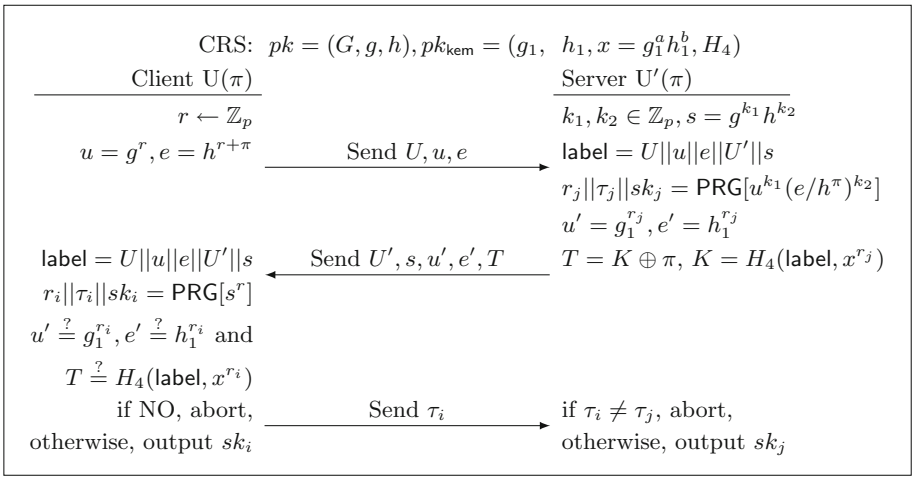

Fig. 2. PAKE based on DDH assumption

5 Conclusion

In this paper, we revisit GK-PAKE, and show that IND-PCA secure KEM is enough for PAKE. We also give concrete examples based on DDH assumptions. The instantiation based on DDH assumption need only 6 group elements and $\log |D|$ bits.

Acknowledgement. Haiyang Xue are supported by the Foundation of Science and Technology on Communication Security Laboratory (9140C110206150C11049) and National Natural Science Foundation of China (No. 61602473, 61502480, 61672019). Bao Li is supported by the Foundation of Science and Technology on Communication Security Laboratory (9140C110206150C11049) and the National Natural Science Foundation of China (No. 61379137). Xianhui Lu is supported by the National Natural Science Foundation of China (No. 61572495).

References

1. Abdalla, M., Benhamouda, F., Pointcheval, D.: Public-key encryption indistinguishable under plaintext-checkable attacks. In: Katz, J. (ed.) PKC 2015. LNCS, vol. 9020, pp. 332–352. Springer, Heidelberg (2015). doi:10.1007/978-3-662-46447-2_15
2. Bellovin, M., Merritt, M.: Encrypted key exchange: password-based protocols secure against dictionary attacks. In: 1992 IEEE Symposium on Security and Privacy, pp. 72–84 (1992)
3. Boyko, V., MacKenzie, P., Patel, S.: Provably secure password-authenticated key exchange using Diffie-Hellman. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 156–171. Springer, Heidelberg (2000). doi:10.1007/3-540-45539-6_12
4. Bellare, M., Pointcheval, D., Rogaway, P.: Authenticated key exchange secure against dictionary attacks. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 139–155. Springer, Heidelberg (2000). doi:10.1007/3-540-45539-6_11
5. Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited. J. ACM **51**(4), 557–594 (2004)
6. Cramer, R., Shoup, V.: Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (2002). doi:10.1007/3-540-46035-7_4
7. Choi, S.G., Herranz, J., Hofheinz, D., Hwang, J.Y., Kiltz, E., Lee, D.H., Yung, M.: The Kurosawa-Desmedt key encapsulation is not chosen-ciphertext secure. Inf. Process. Lett. **109**(16), 897–901 (2009)
8. Goldreich, O., Lindell, Y.: Session-key generation using human passwords only. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 408–432. Springer, Heidelberg (2001). doi:10.1007/3-540-44647-8_24
9. Gennaro, R., Lindell, Y.: A framework for password-based authenticated key exchange. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 524–543. Springer, Heidelberg (2003). doi:10.1007/3-540-39200-9_33
10. Groce, A., Katz, J.: A new framework for efficient password-based authenticated key exchange. In: ACM Conference on Computer and Communications Security, pp. 516–525 (2010)

11. Hofheinz, D., Kiltz, E.: The group of signed quadratic residues and applications. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 637–653. Springer, Heidelberg (2009). doi:10.1007/978-3-642-03356-8_37
12. Jiang, S., Gong, G.: Password based key exchange with mutual authentication. In: Handschuh, H., Hasan, M.A. (eds.) SAC 2004. LNCS, vol. 3357, pp. 267–279. Springer, Heidelberg (2004). doi:10.1007/978-3-540-30564-4_19
13. Kurosawa, K., Desmedt, Y.: A new paradigm of hybrid encryption scheme. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 426–442. Springer, Heidelberg (2004). doi:10.1007/978-3-540-28628-8_26
14. Katz, J., Ostrovsky, R., Yung, M.: Efficient password-authenticated key exchange using human-memorable passwords. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 475–494. Springer, Heidelberg (2001). doi:10.1007/3-540-44987-6_29
15. Kiltz, E., Pietrzak, K., Stam, M., Yung, M.: A new randomness extraction paradigm for hybrid encryption. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 590–609. Springer, Heidelberg (2009). doi:10.1007/978-3-642-01001-9_34
16. Mei, Q., Li, B., Lu, X., Jia, D.: Chosen ciphertext secure encryption under factoring assumption revisited. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 210–227. Springer, Heidelberg (2011). doi:10.1007/978-3-642-19379-8_13