Two-Round ID-PAKE with strong PFS and single pairing operation

Behnam Zahednejad[1] and Gao Chong-zhi[1]

School of Computer Science , Guangzhou University, Guangzhou , China

Abstract. IDentity-based Password Authentication and Key Establishment (ID-PAKE) is an interesting trade-off between the security and efficiency, specially due to the removal of costly Public Key Infrastructure (PKI). However, we observe that previous PAKE schemes such as Beguinet et al. (ACNS 2023), Pan et al. (ASIACRYPT 2023) , Abdallah et al. (CRYPTO 2020) etc. fail to achieve important security properties such as weak/strong Perfect Forward Secrecy (s-PFS), user authentication and resistance to replay attack. In addition, to the best of our knowledge, no previous (P)AKE (either ID- based or PKI-based (P)AKEs) could achieve s-PFS with two-rounds of communication. In this paper, we propose a highly efficient ID-PAKE scheme with s-PFS and KGC-FS using only two rounds of communication, where each party only performs a single pairing operation. We compare our work with previous single pairing-based schemes i.e. Tomida et al. (ESORICS 2019) and Lian et al. (ESORICS 2020) and show that they suffer either s-PFS, KGC-FS attack and replay attack. In order to achieve a privacy-preserving PAKE scheme, we give a fix to Lian et al. (ESORICS 2020) in terms of KGC-FS and user authentication.
We prove the security of our scheme under standard assumptions i.e., Discrete Logarithms (DL) and q-strong Diffie-Hellman(q-sDH) assumption in ID-eCK model. Finally, we conduct a proof-of-concept implementation of our scheme vs. previous single pairing-based schemes and show that our scheme imposes the least computation cost and stands in the middle of previous scheme regarding communication cost.

Keywords: IDentity-based Password Authentication and Key Establishment (ID-PAKE) · Strong-Perfect Forward Secrecy(s-PFS) · KGC-FS · User authentication · ID-eCK model

1 Introduction

Password Authentication and Key Establishment (PAKE) protocols enables two parties who share a weak password to establish a strong session key. Given the user-friendly and easy to memorize passwords, many PAKE protocols are put forth by scholars [1–8]. Further, they are on the way to be standardized by the IETF [9] and be used in standard protocols such as TLS/SSL[10]. However, TLS/SSL requires heavy-cost Public Key Infrastructure (PKI) in place. Therefore, non PKI-based methods such as Zero Knowledge Proof(ZKP) [11],

Key Encapsulation Mechanism (KEM) or identity-based PAKE are gaining more attention[7, 8]. Adi Shamir's pioneering work in the late 1970s laid the foundation for identity-based AKE protocols [12]. Shamir introduced the concept of identity-based cryptography, where public keys are derived from easily verifiable information, such as an individual's email address or username, eliminating the need for traditional public key infrastructure. Following Shamir's breakthrough, subsequent research has focused on refining and expanding identity-based AKE protocols. Noteworthy Canetti et al.[13] contributed to the formalization and analysis of IDentity-based Authentication and Key Exchange protocols (ID-AKE) protocols, ensuring their security and practical viability in diverse applications within the realm of secure communication and information exchange. Bilinear pairing was employed within ID-AKE by Smart [14] for the first time. This work ignited further works using either symmetric/asymmetric pairing. However, most of the existing pairing-based ID-AKEs [15–23] require at least two symmetric/asymmetric pairings by each participant, which reduces the efficiency of ID-AKE schemes. Previous single pairing-based AKE schemes couldn't achieve a robust security proof in id-eCK model. [24, 25] In order to cope with this efficiency and security bottleneck, single pairing-based ID-AKE scheme with security proof in id-eCK model was developed for the first time by Tomida et al. [26] Later, Lian et al. [27] put forth an Identity-Based Identity-Concealed Authentication and Key Exchange protocols (IB-CAKE) with security proof in extended id-eCK model, such that each party only performs a single pairing operation. In the PAKE context, an efficient Identity-Based Password Authenticated Key Exchange (IBPAKE) protocol using identity-based Key exchange was suggested by Choi et al. [1]. Later, Shin [2] showed that Choi et al. [1] scheme enables the malicious PKG (Private Key Generator) to obtain the clients password and impersonate the server. Then, he suggested an improved PAKE to avoid such threats and proved the security of his scheme in the random oracle model. However, we show that these schemes [1, 2] are prone to client impersonation attack, lack of s-PFS, KG-FS and no user authentication. SPEKE[3], SPAKE2[3], and TBPEKE [4] are simple password exponential key exchange. SPAKE enjoys security analysis in the random oracle model under the CDH assumption in the multiplicative groups of finite fields. TBPEKE applies to any group such that elliptic curves can be used at both client and the server to improve the efficiency. Abdallah et al.[5] revised the PAKE notion in the framework of universal composability and suggested a new functionality called lazy-extraction PAKE (lePAKE). According to argument of Abdallah et al.[5], the most efficient PAKE schemes currently known such as SPEKE[4], SPAKE2[4], and TBPEKE [5] can still realize lePAKE functionality in the random-oracle model. KEM was also suggested by Pan et al. [6] and Beguinet et al. [7] to construct secure PAKE in the Beller-Pointcheval-Rogaway (BPR) and Universal Composability (UC) model respectively.

However, we show that none of these efficient PAKE schemes [1–7] can provide s-PFS and user authentication. Further, the password can be easily recovered using side channel attack on the client's device.

1.1 ID-(P)AKE challenges and our contributions

Perfect Forward Secrecy (PFS) holds paramount importance in the realm of (P)AKEs. PFS ensures that even if a long-term secret key is compromised, past communications remain confidential [28]. This property can be classified into strong PFS(s-PFS), weak PFS (w-PFS) and KGC-FS, which are briefly described as follows:

- Weak PFS(w-PFS) ensures that no passive attacker can obtain the past session key, after the compromise of the long-term private key of the participants. The passive attackers can only eavesdrop the messages transmitted through the public channel.
- Strong PFS(s-PFS) ensures that no active attacker can obtain the past session key, upon compromise of the long-term private key of the participants. Unlike passive attackers who may only eavesdrop on communication without altering the messages, active attackers can manipulate or replay messages into the authentication process with the intent of gaining unauthorized access. Active attacks on authentication protocols may involve actions such as impersonation, where the attacker pretends to be a legitimate user, or replay attacks, where the adversary intercepts and reuses authentication messages to gain access.
- KGC-FS ensures that no active attacker can obtain the past or future session key, upon compromise of the long-term private key of the KGC. In contrast to previous attacks, the long-term private key of the participants is no longer revealed.

To the best of our knowledge all previous ID-based or non ID-based (P)AKE schemes which could satisfy strong PFS property require at least three rounds of communications. Therefore, it remains an open problem to design an efficient ID-PAKE scheme with the strong PFS property using only two rounds of communication. Other important security requirements in the context of ID-PAKE include

- **User authentication**: In order to prevent the fraud upon exposure of the client's device, the real time user should be authenticated before accessing the network.
- **Off-line guessing attack resistance**: Given the low-entropy size of the passwords, the attacker might guess the password. This requirement prevents the attacker to determine the correctness of his guess using the public transmitted messages or the extracted user's device (e.g. using side channel attack [29]).
- **On-line guessing attack resistance**: This requirement limits the attacker to run the protocol and establish the protocol with honest users with a bounded number of password trials.

In a nutshell, the main challenges on the design of an efficient ID-PAKE scheme can be summarized as follows:

1. Only a single pairing operation is used by each participant.
2. Strong and KGC-FS are achieved using only two rounds of communications.
3. Replay attack , user and server impersonation are prevented such that it doesn't give rise to other serious attacks such as s-PFS.
4. User authentication is provided such that obtaining the client's device doesn't enable the attacker to access the server.
5. It remains infeasible for the attacker to mount offline/online guessing attack on the ID-PAKE.
6. A robust security proof is provided based on id-eCK security model and standard assumptions.

Table 1. A general comparison of our proposed scheme vs. previous scheme.

Scheme	**# of pairing**	**Security model**	**Properties**								**Comm round**
			P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	
AKE-Fujioka et al. [17]	2	id-eCK	✓	✓	✗	✗	✗	✗	✗	✗	2
AKE-Huang et al. [18]	2	id-eCK	✓	✓	✗	✗	✗	✗	✗	✗	2
AKE-Tomida et al. [26]	1	id-eCK	✓	✓	✗	✓	✗	✗	✗	✗	2
AKE-Lian et al. [27]	1	ext id-eCK	✓	✓	✓	✗	✓	✓	✗	✗	3
PAKE-Choi et al. [1]	1	BPR	✓	✓	✗	✓	✗	✗	✗	✗	2
PAKE-Shin et al. [2]	1	BPR	✓	✓	✗	✓	✗	✗	✗	✗	2
PAKE-Abdalla et al. [3]	0	Random oracle	✓	✓	✗	✓	✗	✗	✗	✗	2
PAKE-Pointcheval et al. [4]	0	Real-Or-Random	✓	✓	✗	✓	✗	✗	✗	✗	2/3
PAKE-Abdalla et al. [5]	0	UC	✓	✓	✗	✓	✗	✗	✗	✗	2
PAKE-Abdalla et al. [6]	0	Random oracle	✓	✓	✗	✓	✗	✗	✗	✗	2
PAKE-Pan et al. [7]	0	BPR	✓	✗	✓	✓	✗	✗	✗	✗	2
PAKE-Beguinet et al. [8]	0	UC	✓	✗	✓	✓	✗	✗	✗	✗	2
Our PAKE-PFS	1	id-eCK	✓	✓	✓	✓	✓	✗	✓	✓	2
Our fix to Lian et al. [27]	1	ext id-eCK	✓	✓	✓	✓	✓	✓	✓	✓	3

P_1: Mutual authentication P_2: Weak Perfect Forward Security(W-PFS)
P_3: Strong Perfect Forward Security(S-PFS) under leakage of participants long-term private keys
P_4: KGC Forward Security(KGC-FS) under leakage of KGC private key
P_5: Replay attack / Impersonation security
P_6: Identity concealment , P_7: User authentication/password exposure attack
P_8: Offline/online guessing attack resilience

In this paper, we make the following contributions to address the above challenges:

1. We perform cryptanalysis of previous single pairing-based ID-AKE ([26, 27]) and non-PKI based PAKE schemes [1–8] . We show that they suffer serious vulnerabilities such as user impersonation attack, no user authentication, lack of KGC-FS and weak/strong PFS.

2. We propose our ID-based PAKE protocol with strong-PFS , KGC-FS (ID-PAKE-PFS) and single pairing operation using only two rounds of communication. To the best of our knowledge, this is the first (P)AKE protocol that achieves strong PFS with two rounds of communication.
3. We prove the PAKE security i.e., session key security, strong PFS , KGC-FS and resistance to online/offline guessing attack of our scheme in id-eCK model. Our proof is based on standard security assumptions such as Discrete Logarithms (DL) and 1-strong Diffie-Hellman(1-sDH) assumptions.
4. In order to achieve a privacy-preserving PAKE, We give an improvement to Lian et al.'s scheme [27] in terms of user authentication and KGC-FS. The KGC-FS security of the improved scheme is based on Computational Diffie-Hellman(CDH) assumption.
5. We conduct a performance comparison between our scheme and previous single pairing-based schemes through a proof-of-concept implementation in Raspberry Pi 3 processors, which is widely used in IoT applications. The results indicate that our scheme imposes the least computation cost, compared to previous single pairing schemes (Tomida et al.[26] and Lian et al.[27]) In addition, the communication cost of our scheme is around one fourth of Lian et al. [27] 's scheme and very close to Tomida et al. [26]'s scheme.

A general comparison between our scheme and previous (P)AKE schemes is shown in Table 1 in terms of the number of pairing, security model, security properties and communication rounds.

2 Preliminary

2.1 Notation

The field Z/qZ is represented by Z_q^* for prime q. For each finite set $\mathcal{S}$, $s \in_U \mathcal{S}$ denotes that the value s is uniformly chosen from the finite set $\mathcal{S}$. $x||y$ denotes the concatenation of two elements $x, y \in \{0,1\}^*$.

2.2 Bilinear pairing and Assumptions

Definition 1 (Bilinear groups). *Let p be a prime value. The groups G_1, G_2, G_T of order q are called bilinear groups if an efficiently computable bilinear map $e : G_1 \times G_2 \rightarrow G_T$ exists such that the following properties hold:*

1. *Bilinearity: For all $g_1 \in G_1, g_2 \in G_2$ and $a, b \in Z_q^*$, we have $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$.*
2. *Non-degenerate: Assuming g_1 as the generator of G_1 and g_2 as the generator of G_2 , then $e(g_1, g_2)$ would be the generator of G_T.*
3. *Computable: For all g_1 as the generator of G_1 and g_2 as the generator of G_2 , the function $e(g_1, g_2)$ can be computed efficiently.*

Definition 2 (Discrete Logarithms (DL) Assumption [30]). *Assuming Z_q^* for prime q, and g as the generator of Z_q^*, Discrete Logarithms (DL) assumption states that all PPT (Probabilistic Polynominal Time bounded) attackers can not derive the value a, given $(g, q, g^a \bmod q)$ for a randomly chosen $a \in Z_q^*$ with non-negligible probability.*

Definition 3 (q-strong Diffie-Hellman(q-sDH) Assumption [31]).
Assuming g_1 as the generator of G_1 and g_2 as the generator of G_2 , for any random $x \in Z_q^*$, the q-strong Diffie-Hellman assumption states that any PPT attacker can not obtain $(g_2^{\frac{1}{x+c}}, c) \in (G_2, Z_q^*)$, given $(g_1, g_1^x, g_1^{x^2}, ..., g_1^{x^q}, g_2) \in (G_1^{q+1}, G_2)$, with non-negligible probability. In this paper, we use 1-sDH assumption, in which the attacker is only given $(g_1, g_1^x, g_2) \in (G_1^2, G_2)$.

Definition 4 (Computational Diffie-Hellman(CDH) Assumption [32]).
Assuming g_1 as the generator of G_1, for any random $x, y \in Z_q^*$, the Computational Diffie-Hellman(CDH) Assumption states that any PPT attacker can not obtain $g_1^{ab} \in G_1$, given $(g_1, g_1^x, g_1^y \in G_1)$, with non-negligible probability.

3 Security model

In this section, we briefly review the id-eCK [18] as the first ID-based version of the eCK security model developed by LaMacchia et al. [33].

3.1 Participants, Adversary and Security Experiments

In order to establish a robust security framework, it is crucial to define the main participants involved in the system. Additionally, it is imperative to delineate the adversary setting, identifying potential threats and their capabilities. This foundational understanding sets the stage for a comprehensive analysis of security measures.

Participants Each protocol participant U_i is a Probabilistic Polynomial Time (PPT) turing machine, which is identified by a unique identifier ID_i. We assume that each participant U_i can execute a polynomial number of protocol instances in parallel. For each party U_i communicating with peer U_j , we use π_i^s to denote the s-th instance of the communication (also known as a session/oracle). Each instance π_i^s has the following variables:

- pid_i^s : The identity of the communication peer with U_i in s-th session.
- $role_i^s$: The communication role of the U_i such that $role_i^s \in \{Initiator, Responder\}$.
- ss_i^s : The session state memory of U_i in s-th session.
- ls_i^s : The private long-term state memory of U_i.
- k_i^s : The session key of s-th session established by U_i.
- Ψ_i^s : This value represents whether party P_i has completed the s-th session and accepted the key or not, s.t $\Psi_i^s \in \{\emptyset, accept, reject\}$.

Adversary The adversary $\mathcal{A}$ is also a PPT turing machine who can manage and control the communication network between the participants via a **Send** (π_i^s, m) query in session s executed between ID_i and ID_j. The message m is either $(ID_i, ID_j), (ID_i, ID_j, c_i), (ID_i, ID_j, c_i, c_j)$. The adversary $\mathcal{A}$ does not access the private information of participants directly. However, it can query the following oracles to the challenger $\mathcal{C}$:

- **EphemeralKeyReveal**(π_i^s): Upon calling this query, the ephemeral private key stored in session state ss_i^s used during π_i^s is given to the adversary. In practice, this query refers to different scenarios, such as secret ephemeral values stored in unprotected memory of a device, or when the party's random number generator becomes compromised.
- **SessionKeyReveal**(π_i^s): This query models the leakage of the session key of s-th session of party U_i communicating with U_j to the adversary. If $\Psi_i^s \neq$ *accept* , this oracle returns $\perp$. Otherwise, it returns (k_i^s)
- **StaticKeyReveal**(π_i^s, ID_i): The static private key of the party ID_i and the contents of the long-term states ls_i^s is revealed to the attacker.
- **KGCMasterKeyReveal**: The KGC master private key is given to the attacker. KGC forward security can be analyzed with this query.
- **PasswordReveal**(π_i^s): This query reveals the password of the attacker used in session π_i^s.
- **Extract**(π_i^s): This query reveals the contents of the party ID_i's device stored in long term state ls_i^s of session π_i^s.
- **Test**(π_i^s) If $\Psi_i^s \neq$ *accept* , this oracle returns $\perp$. Otherwise, it sets $(k_0 = k_i^s)$ and $k_1 \in_U \{0,1\}^k$. The challenger chooses a random challenge $b \in_U \{0,1\}$ and returns k_b to the attacker$\mathcal{A}$.
- **TestPW**(π_i^s): This oracle sets $pw_0 = pw_i^s$ and $pw_1 \in_U \{0,1\}^l$, where l is the size of the password. The challenger $\mathcal{C}$ chooses a random challenge $b \in_U \{0,1\}$ and returns pw_b to the attacker $\mathcal{A}$.

Security experiment We describe the security experiment $Exp^X_{\pi_i^s,\mathcal{A}}(k)$ for session key secrecy (SK), s-(P)FS , KGC-FS and Online/Offline Guessing Attack (OGA) such that $X \in \{SK, s-(P)FS, KGC-FS, OGA\}$. The security experiment $Exp^{SK}_{\pi_i^s,\mathcal{A}}(k)$ is run between the challenger $\mathcal{C}$ and the attacker $\mathcal{A}$ in the test session π_i^s as follows:

1. The challenger $\mathcal{C}$ outputs all public parameters such as KGC public key and identity of participants to the attacker $\mathcal{A}$.
2. The attacker $\mathcal{A}$ issues polynominal number of **SessionKeyReveal** , **EphemeralKeyReveal** and **Send** queries in any order.
3. The attacker $\mathcal{A}$ chooses a fresh session π_i^s (Definition 7) as the test session and issues **Test**(π_i^s) query only once.
4. It continues queries similar as step 2 to the test session π_i^s except queries which violate the definition of fresh session (Definition 6).
5. The attacker $\mathcal{A}$ guesses the bit b' for the value b chosen by the challenger $\mathcal{C}$. The security experiment returns $'1'$ if $b' = b$ or $'0'$ if $b' \neq b$.

The security experiment $Exp^{X}_{\pi_i^s,\mathcal{A}}(k)$ for the $X \in \{s-(P)FS, KGC-FS\}$ is similar to above experiment except that the attacker can issue the following queries in step 4:

- For $Exp^{s-FS}_{\pi_i^s,\mathcal{A}}(k)$, the attacker issues **StaticKeyReveal**(ID_i) and **PasswordReveal**(π_i^s) queries.
- For $Exp^{s-PFS}_{\pi_i^s,\mathcal{A}}(k)$, the attacker issues both **StaticKeyReveal**(ID_i) and **StaticKeyReveal**(ID_j) and **PasswordReveal**(π_i^s) queries.
- For $Exp^{KGC-FS}_{\pi_i^s,\mathcal{A}}(k)$, the attacker issues **KGCMasterKeyReveal** and **PasswordReveal**(π_i^s) queries.

The online/offline guessing attack experiment $Exp^{OGA}_{\pi_i^s,\mathcal{A}}(k)$, is run between the challenger $\mathcal{C}$ and the attacker $\mathcal{A}$ in the test session π_i^s as follows:

1. The challenger $\mathcal{C}$ outputs all public parameters such as KGC public key and identity of participants to the attacker $\mathcal{A}$.
2. The attacker $\mathcal{A}$ issues polynominal number of **Extract**(π_i^s) and **Send** queries in any order.
3. The attacker $\mathcal{A}$ chooses a random session π_i^s as the test session and issues **TestPW**(π_i^s) query only once.
4. The attacker $\mathcal{A}$ guesses the bit b' for b. The experiment returns $'1'$ if $b' = b$ or $'0'$ if $b' \neq b$.

3.2 Security Definitions

Definition 5 (Origin oracle). *The oracle π_j^t is an origin oracle of π_i^s if $\Psi_i^s =$ accept, $\Psi_j^t \neq \emptyset$ and messages sent by U_i equal the messages received by U_j i.e., $sent_i^s = recv_j^t$.*

Definition 6 (Matching oracle). *The completed oracles π_i^s and $\pi_{j,i}^t$ are matching if the following conditions hold:*

1. *Both π_i^s and $\pi_{j,i}^t$ are origin oracle for each other.*
2. *$pid_i^s = j$ and $pid_j^t = i$.*
3. *$role_i^s \neq role_j^t$.*

Definition 7 (Freshness).

The oracle π_i^s with matching oracle $\pi_{j,i}^t$ (if it exists) between honest party ID_i and its honest peer ID_j is called fresh if none of following three conditions hold:

1. *The session key of π_i^s or of its matching session $\pi_{j,i}^t$ (if it exists) is revealed through **SessionKeyReveal**(π_i^s) or **SessionKeyReveal**$(\pi_{j,i}^t)$ queries.*
2. *The matching session $\pi_{j,i}^t$ exists and the adversary reveals one of the following cases:*
   - *Both the static key of ID_i and ephemeral key of session π_i^s by calling both **StaticKeyReveal**$(ID_i)$and **EphemeralKeyReveal**(π_i^s) queries.*

- *Both the static key of ID_j and ephemeral key of session π_j^t by calling both **StaticKeyReveal**$(ID_j)$and **EphemeralKeyReveal**(π_j^t) queries.*

3. *The session π_j^t holds no matching sessions and the adversary reveals one of the following cases:*
   - *Both the static key of ID_i and ephemeral key of session π_i^s by calling both **StaticKeyReveal**$(ID_i)$and **EphemeralKeyReveal**(π_i^s) queries.*
   - *The static key of ID_j using either **StaticKeyReveal**(ID_j) or **KGC-MasterKeyReveal**.*

Definition 8 (ID-AKE Security).

Let $Pr[Exp^X_{\pi_i^s,\mathcal{A}}(k) = 1]$ denote the probability that the adversary $\mathcal{A}$ breaks the X property of the test session π_i^s. These properties include $X = SK - security, s-(P)FS, KGC-FS, OGA$. The advantage of the adversary $\mathcal{A}$ in breaking X property is defined as:

$$Adv^X_{\pi_i^s,\mathcal{A}}(k) = Pr[Exp^X_{\pi_i^s,\mathcal{A}}(k) = 1] - \frac{1}{2} \tag{1}$$

Our ID-AKE-PFS is secure if for any PPT adversary $\mathcal{A}$, the function $Adv^X_{\pi_i^s,\mathcal{A}}(k)$ is negligible in k.

4 Cryptanalysis of previous (P)AKE schemes

In this section, we describe the main vulnerabilities of previous non-PKI based PAKE schemes [1–8] and ID-AKE schemes which employ single pairing ([26, 27]). Their main limitations include user/server impersonation attack, lack of strong Perfect Forward Secrecy (s-PFS), lack of KGC-FS and no user authentication. These vulnerabilities are described as follows:

4.1 User/server impersonation attack

In Tomida et al. [26]'s scheme, any party including the attacker $\mathcal{A}$, is able to impersonate the user U_A (or U_B) by generating an ephemeral public key X_A of the user U_A (or ephemeral public key X_B of user U_B). Similarly, in Pointcheval et al.'s [4] scheme, the attacker $\mathcal{A}$ can impersonate the server S, generate the values $\{Y = g^y, \epsilon\}$, replay the seed s and respond them to the client. Also, in Abdalla et al.'s [3], Choi et al.'s [1] and Shin [2] scheme, the attacker can replay the client's first message $(X^*/ID_c, W, X/C, U_1, U_2)$ from previous session and impersonate the client. While it remains infeasible for the attacker $\mathcal{A}$ to acquire the session key K, it is imperative to acknowledge that the vulnerability posed by impersonation attacks constitutes a significant risk to the overall security of the scheme. In particular, it leads to the exposure of the scheme against s-PFS property.

4.2 Lack of weak Perfect Forward Secrecy (w-PFS)

Pan et al.'s [7] scheme as a tightly secure PAKE and Beguinet et al.[8]'s scheme as a generic scheme proven in the Universal Composability (UC) model both suffer weak PFS attack against a passive adversary. It suffices for the attacker $\mathcal{A}$ to eavesdrop public transmitted messages $M_U, M_S/Epk, Ec$ of a target session and derive the secret session key parameters as $pk = D_1(pw, e_1), c = D_1(pw, e_2), k = Decaps(c, sk), pw/pk = D_1(pw||ssid, Epk), c = D_1(pw||ssid, Ec), k = Decaps(c, sk)$ after obtaining the long term secrets of the client (pw, sk).

4.3 Lack of strong Perfect Forward Secrecy (s-PFS)

Thanks to the successful user/server impersonation attack (section 4.1), Tomida et al. [26], Pointcheval et al.'s [4], Abdalla et al.'s [3], Choi et al.'s [1] and Shin [2] 's scheme can not provide s-PFS. In Tomida et al. [26]'s scheme, the attacker can impersonate the user U_A to party U_B by generating an ephemeral private/public key x_A/X_A. Once the secret long-term key of the user U_A (K_A) is leaked, the attacker can obtain the previous established session keys as $\sigma = e((X_B(Wg_1^{i_A})^{d_B})^{x_A+d_A}, K_A), K = H_4(\sigma, ID_A, ID_B, X_A, X_B)$ and thereby violate the s-PFS property. In Pointcheval et al.'s [4] scheme, once the attacker $\mathcal{A}$ queries the ephemeral values of the server in an exposed session (y, ϵ), it can replay the server's response of the exposed session to the test session to impersonate the server. After getting the password (pw), it can compute the session key ek, sk of the test session. Similarly in Abdalla et al.'s [3], Choi et al.'s [1] and Shin [2] 's scheme, once the attacker $\mathcal{A}$ queries the ephemeral values of the client in an exposed session $(x/x, r)$, it can replay the client's request of the exposed session to the test session to impersonate the client. After getting the password (pw) and the private keys of the client (sk_{ID_c}) , it can compute the session key sk_c of the test session.

4.4 Lack of KGC Forward Secrecy(KGC-FS)

In Lian et al.[27] 's scheme, once the KGC secret key i.e., msk is revealed, thanks to the bilinear property of the bilinear pairing, not only the current session key, but also all previous and future session keys are exposed to the attacker. In order to obtain the session key between the initiator ID_A with ephemeral public key X and responder ID_B with ephemeral public key Y, the attacker can compute the session key as $(K_1, K_2) = KDF(e(X, Y)^{msk}, X||Y)$.

4.5 No user authentication/password exposure attack

In most PAKE schemes [1–8], the password is stored directly in the client's device in plaintext. Therefore, extracting the client's device (e.g. using side channel attack) would enable the attacker to derive the password pw. In addition, no user authentication is provided such that obtaining the client's device suffices to access the network services.

5 Our ID-based PAKE Protocol with s-PFS (ID-PAKE-PFS)

In this section, we present our ID-based PAKE Protocol with s-PFS property (ID-AKE-PFS). We assume k as the security parameter, $H_1, H_2 : \{0,1\}^* \rightarrow Z_q^*$ and $H_3 : \{0,1\}^* \rightarrow \{0,1\}^k$. The symmetric encryption/decryption functions are also denoted as $Enc/Dec : \{0,1\}^* \times \{0,1\}^k \rightarrow \{0,1\}^k$

KGC registration

1. The KGC chooses a master secret key $w \in Z_q^*$ and broadcasts his public key $W = g_1^w$.
2. For each party (client/server) $U_i = C_i/U_j = S_i$ with identity ID_i/ID_j, KGC assigns a static secret key as $SK_i = g_2^{\frac{1}{w+d_i}}/SK_j = g_2^{\frac{1}{w+d_j}}$, where $d_i = H_1(ID_i, t_{reg_i}), d_j = H_1(ID_j)$. Here t_{reg_i} is the registration time of U_i. In addition, the KGC assigns and delivers a password $pw_{ij} \in \{0,1\}^l$ to both C_i and S_j, through a secure channel. Here, $10^4 \leq l \leq 10^6$ denotes the size of the password.
3. As the client receives the password pw_{ij} and static key SK_i, it chooses a small integer $2^2 \leq n_i \leq 2^3$ and stores the values $hpw_{ij} = H_3((pw_{ij}, ID_i) \mod n_i)$, $ESK_i = (SK_i \oplus pw_{ij}), n_i, t_{reg_i}$ in his memory.
4. The server S_j also stores his static key SK_j and adds a tuple $(ID_i, pw_{ij}, blist_i = \emptyset, t_{reg_i})$ in his database for each client ID_i.

Participants The mutual authentication between the user ID_A, client C_A and the server S_B is depicted in Figure 1. The main steps of the authentication consists of:

1. The user inputs his identity $ID_A^{'}$ and password $pw_{AB}^{'}$ into the client's device, which aborts if the relation $hpw_{ij} = H_3((pw_{AB}^{'}, ID_A^{'}) \mod n_i$ doesn't holds. Otherwise, it authenticates the user and retrieves the static secret key as $SK_A = ESK_A \oplus pw_{AB}$ and chooses a nonce $r_A \in \{0,1\}^k$ to compute $x_A = H_2(SK_A, r_A)$. Then, it sets the ephemeral public key as $X_A = SK_A^{x_A}$, computes $c_A = Enc((ID_A, X_A), H_3(pw_{AB}, t_A))$ and sends $(\pi_{A,B}^s, t_{reg_i}, ID_A, t_A, c_A)$ to server S_B. Here, t_A is the current time-stamp. It also erases x_A from the session state ss_A^s.
2. Upon receiving $(\pi_{A,B}^s, t_{reg_A}^{'}, ID_A, t_A, c_A)$, the server S_B makes sure over the freshness of the time-stamp if $t_A^{'} - t_A \leq \Delta$ holds. Here, $t_A^{'}$ is the current time-stamp. It retrieves the registration time t_{reg_A} and the password pw_{AB} corresponding to ID_A from his database and aborts if $t_{reg_A}^{'} \neq t_{reg_A}$. Otherwise, it decrypts $(ID_A^{'}, X_A) = Dec(c_A, H_3(pw_{AB}, t_A))$ and rejects the client's request if $ID_A^{'} \neq ID_A$ and increments the client's block list $blist_A+$. Also, it revokes the client ID_A and asks him to register again if $blist_A \geq N_i = 5$. Otherwise, it chooses a nonce $r_B \in \{0,1\}^k$ to compute

$x_B = H_2(SK_B, r_B)$, sets the ephemeral public key as $X_B = SK_B^{x_B}$ and computes $c_B = Enc((ID_B, X_B), H_3(pw_{AB}, t_B))$, where t_B denotes the current time-stamp. It sends $(\pi^s_{A,B}, ID_B, t_B, c_B)$ to client U_A and computes the session key K as:

$$d_A = H_1(ID_A, t_{reg_A}), \sigma_B = e(Wg_1^{d_A}, X_A^{x_B}), K = H_3(\sigma_B, ID_A, ID_B, t_A, t_B, X_A, X_B) \tag{2}$$

Then, it erases x_B from the session state ss^s_B.

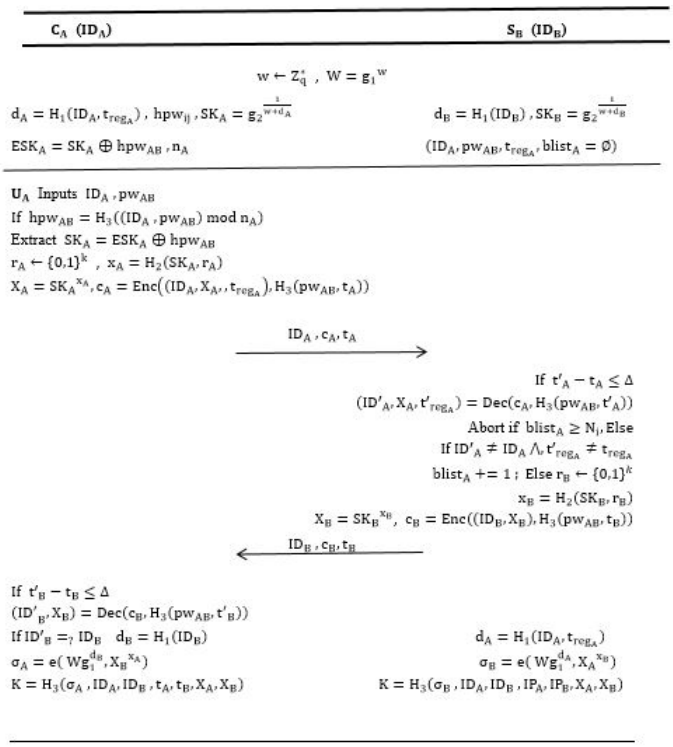

Fig. 1. A schematic of our ID-AKE-PFS

3. As the party U_A receives $(\pi^s_{A,B}, ID_B, t_B, c_B)$, it makes sure over the freshness of the time-stamp if $t^{'}_B - t_B \leq \Delta$ holds, where $t^{'}_B$ denotes the current time-stamp. It decrypts c_B as $(ID^{'}_B, X_B) = Dec(c_B, H_3(pw_{AB}, t_B))$ and rejects the server's response if $ID^{'}_B \neq ID_B$. Otherwise, it computes $x_A = H_2(SK_A, r_A)$ for one more time and computes the session key as follows:

$$d_B = H_1(ID_B), \sigma_A = e(Wg_1^{d_B}, X_B^{x_A}), K = H_3(\sigma_A, ID_A, ID_B, t_A, t_B, X_A, X_B) \tag{3}$$

Correctness It can be shown that the two participants U_A, U_B derive the same session key, if they run the protocol honestly:

$$\sigma_A = e(Wg_1^{d_A}, X_A^{x_B}) = \sigma_B = e(Wg_1^{d_B}, X_B^{x_A}) = g_T^{x_A x_B} \tag{4}$$

6 Security Proof of Our ID-AKE-PFS

Theorem 1. *Our proposed ID-AKE-PFS scheme achieves SK-security in the id-eCK model under 1-sDH and DL assumptions. The advantage of the attacker $Adv_{\mathcal{A}}^{SK}(k)$ is upper bounded by:*

$$\begin{aligned} Adv_{\mathcal{A}}^{SK}(k) = Pr[Exp_{\pi_i^s,\mathcal{A}}^{SK}(k) = 1] - \frac{1}{2} \leq 2s(k)n^2(k)t(k)\frac{Pr(\mathcal{S})}{|pw_{ij}|} + \\ [s(k)n^2(k)t(k) + \frac{s^2(k)t(k)}{2}]\frac{Pr(\mathcal{F})}{|pw_{ij}|} \end{aligned} \tag{5}$$

Here , $Pr(\mathcal{S}), Pr(\mathcal{F})$ denote the success probability of the 1-sDH and DL solvers. Given the 1-sDH and DL assumptions, we can claim that $Adv_{\mathcal{A}}^{SK}(k)$ is negligible in terms of the security parameter k.

Proof. In order to breaks the SK-secrecy in security experiment i.e. $Exp_{\pi_i^s,\mathcal{A}}^{SK}(k) = 1$, and distinguish the session key of the test session from a random string, the attacker has the following methods:

1. Key impersonation (M_1): The attacker $\mathcal{A}$ computes either the value SK_A, x_A or SK_B, x_B by itself, constructs the value σ_A, σ_B and queries H_3 in the test session $\pi_{A,B}^s$ owned by party U_A.
2. Key replication (M_2): The attacker $\mathcal{A}$ forces the session key of an exposed non-matching session to be the same as the session key of the test session $\pi_{A,B}^s$. As the non-matching session is exposed, the attacker can learn the session key of the test session $\pi_{A,B}^s$. However, the key derivation function (H_3) of the test session includes the identities and public ephemeral keys of the participants. Therefore, the success probability of the attacker $\mathcal{A}$ is negligible, as two non-matching sessions can not have the same identities and public ephemeral keys.

As the second method is not feasible ($Pr[M_2] = 0$), the success probability of the attacker is represented as

$$\begin{aligned} Pr[Exp_{\pi_i^s,\mathcal{A}}^{SK}(k) = 1] = Pr[M_1]Pr[Exp_{\pi_i^s,\mathcal{A}}^{SK}(k) = 1|M_1] + \\ Pr[M_2]Pr[Exp_{\pi_i^s,\mathcal{A}}^{SK}(k) = 1|M_2] + Pr[1-(M_1+M_2)]Pr[Exp_{\pi_i^s,\mathcal{A}}^{SK}(k) = 1|\neg(M_1 \vee M_2)] \\ \leq Pr[M_1](Pr[Exp_{\pi_i^s,\mathcal{A}}^{SK}(k) = 1|M_1] - \frac{1}{2}) + \frac{1}{2} \end{aligned}$$

In the following, we focus on the possible scenarios of the first method ($Pr[M_1] = Pr[S_1] + Pr[S_2]$):

Scenario 1 (S_1): The test session π_i^s has no matching session owned by an honest party.

Scenario 2 (S_2): The test session π_i^s holds a matching session owned by an honest party.

6.1 Scenario 1 (S_1)

For this scenario, two cases can be considered such that ($Pr[S_1] = Pr[S_{1.1}] + Pr[S_{1.2}]$):

Case 1.1 ($S_{1.1}$): The static private key owned by the party $ID_i = A$ has never been revealed to the adversary $\mathcal{A}$. (According to the freshness definition, the ephemeral secret keys of the party $ID_i = A$ can be obtained to the attacker.) In order to obtain the session key of the test session π_i^s , the attacker needs to guess the password (with probability $\frac{1}{|pw_{ij}|}$) and the static key of either the owner of the test session (SK_A) or its peer (SK_B) to compute x_A or x_B, respectively. Therefore, we construct the solver $\mathcal{S}$ of the 1-sDH problem which aims to compute SK_A or SK_B. Given $n(k)$ honest parties for the security parameter k and $s(k)$ sessions for each party, the solver $\mathcal{S}$ guesses that party A is communicating with party B with probability $\frac{1}{n(k)^2}$, and chooses session s as the test session with probability $\frac{1}{s(k)}$. Further, it chooses random static private keys for the remaining n-1 parties (including B). Then, it responds the $\mathcal{A}$'s queries as follows:

1. **Send** $(\pi_i^s, (ID_i, ID_j))$: If $ID_i \neq A/B$, the solver $\mathcal{S}$ follows the protocol honestly and outputs the first message $c_i = Enc_{H_3(pw_{ij},t_i)}(SK_i^{x_i})$ to the attacker. If $ID_i = A/B$, the solver $\mathcal{S}$ computes the static private key SK_A/SK_B by solving 1-sDH problem. Then, it outputs $c_A = Enc_{H_3(pw_{AB},t_A)}(SK_A^{x_A})$ or $c_B = Enc_{H_3(pw_{AB},t_B)}(SK_B^{x_B})$ to the attacker.
2. **Send** $(\pi_i^s, (ID_i, ID_j, c_i))$: If $ID_i \neq A/B$, the solver $\mathcal{S}$ executes the protocol honestly and outputs the second message $c_j = Enc_{H_3(pw_{ij},t_j)}(SK_j^{x_j})$ to the attacker. If $ID_j = A/B$, the solver computes the static private key SK_A/SK_B by solving 1-sDH problem, and outputs $c_A = Enc_{H_3(pw_{ij},t_i)}(SK_A^{x_A})$ or $c_B = Enc_{H_3(pw_{ij},t_i)}(SK_B^{x_B})$ to the attacker.
3. **Send** $(\pi_i^s, (ID_i, ID_j, c_i, c_j))$: If this query has been used before or query item 1 is not executed, the solver $\mathcal{S}$ aborts. Otherwise, it computes the session key K according to the protocol specification.
4. $H_1(ID_j)$: If this query is called for the first time, a random string $h_j \in Z_q^*$ is given to the attacker. Otherwise, the same value is output as before.
5. $H_2(SK_j, r_j)$: If this query is called before, the solver $\mathcal{S}$ outputs the same value. Otherwise, it outputs the value $e(g_1, SK_j)$ if $e(Wg_1^{d_j}, SK_j) = g_T$ holds. Else, it returns a random string $x_j \in Z_q^*$ to the attacker if the relation $e(Wg_1^{d_j}, SK_j) = g_T$ doesn't hold.

6. $H_3(\sigma, ID_i, ID_j, X_i, X_j)$: If the oracle $H_3(\sigma, ID_i, ID_j, X_i, X_j)$ is called before, the solver $\mathcal{S}$ outputs the same value. Otherwise, it checks whether the output of the query **Send** $(\pi_i^s, (ID_i, ID_j, X_i))$ is X_j. If so, it gives the session key K computed in query **Send** $(\pi_i^s, (ID_i, ID_j, c_i, c_j))$ to the attacker $\mathcal{A}$.
7. **SessionKeyReveal**$(\pi_i^s, ID_i, ID_j, X_i, X_j)$: If π_i^s is the test session, the simulator $\mathcal{S}$ aborts. Otherwise, it checks if the records of the query **Send** $(\pi_i^s, (ID_i, ID_j, c_i, c_j))$ exists. If they indeed exist, it gives the session key K computed in this oracle to the attacker.
8. **StaticKeyReveal** (ID_i): If $ID_i = A/B$, the simulator $\mathcal{S}$ aborts. Otherwise, it returns the corresponding static private keys to the attacker $\mathcal{A}$.
9. **EphemeralKeyReveal**(π_i^s): The simulator $\mathcal{S}$ returns the ephemeral private keys r_i, r_j used in the first two queries i.e., **Send** $(\pi_i^s, (ID_i, ID_j))$, **Send**$(\pi_i^s, (ID_i, ID_j, c_i)$.
10. **Test**(π_i^s): If π_i^s is not the test session, the simulator $\mathcal{S}$ aborts. Otherwise, it returns a random string $K \in \{0,1\}^k$ to the adversary $\mathcal{A}$.

As the test session $s(k)$ and participants are chosen among $n(k)$ honest parties in a random manner, the success probability of the simulator $\mathcal{S}$ would be as follows:

$$Pr(\mathcal{S}) \geq \frac{|pw_{ij}|}{s(k)n^2(k)t(k)} Pr[S_{1.1}](k) \tag{6}$$

Here, $Pr[S_{1.1}](k)$ denotes the probability that the case 1.1 ($S_{1.1}$) occurs and $t(k)$ is the bound of hash $H_i(\cdot), 1 \leq i \leq 3$ queries called by the attacker $\mathcal{A}$.

Case 1.2 ($S_{1.2}$): The static private key owned by the party $ID_i = A$ has been revealed to the adversary $\mathcal{A}$, but the ephemeral private keys i.e., r_i, r_j or x_i, x_j of the test session π_i^s are not exposed. Given the static private key of party A i.e.,SK_A and the public ephemeral key X_A, the attacker needs the ephemeral private keys x_A to compute the session key K. Therefore, we construct the DL problem simulator $\mathcal{F}$. All the query responses are the same as case 1.1, except the following queries:

1. Queries **Send** $(\pi_i^s, (ID_i, ID_j)$ and **Send** $(\pi_i^s, (ID_i, ID_j, c_i))$ are the same as case 1.1 if π_i^s is not the test session. Otherwise, the solver $\mathcal{F}$ aborts.
2. **Send** $(\pi_i^s, (ID_i, ID_j, c_i, c_j))$: If π_i^s is not test session, or $ID_i \neq A/B$, the same response is given as case 1.1. Otherwise, if $ID_i = B$, the solver $\mathcal{F}$ aborts as no matching session exists in this case. If $ID_i = A$, it obtains the private ephemeral key x_i corresponding to X_i by solving DL problem. Then, it computes the session key K according to the protocol specification.
3. **StaticKeyReveal**(ID_i): If $ID_i = B$, the simulator $\mathcal{F}$ aborts. Otherwise, it returns the corresponding static keys to the attacker.
4. **EphemeralKeyReveal**(π_i^s): If π_i^s is not the test session, the same response is given as case 1.1. Otherwise, it aborts.

The success probability of the simulator $\mathcal{F}$ is:

$$Pr(\mathcal{F}) \geq \frac{|pw_{ij}|}{s(k)n^2(k)t(k)} Pr[S_{1.2}](k) \tag{7}$$

Here, $Pr[S_{1.2}](k)$ denotes the probability that the case 1.2 ($S_{1.2}$) occurs and $t(k)$ is the bound of hash $H_i(\cdot), 1 \leq i \leq 3$ queries called by the attacker $\mathcal{A}$.

6.2 Scenario 2

For this scenario, a series of cases emerge such that ($Pr[S_2] = Pr[S_{2.1}] + Pr[S_{2.2}] + Pr[S_{2.3}] + Pr[S_{2.4}]$):

Case 2.1 ($S_{2.1}$): The static private key owned by the party $ID_i = A$ in π_i^s and $ID_j = B$ in matching session π_j^t has been revealed to the adversary $\mathcal{A}$.

In this case, the attacker possess SK_A, SK_B. Given the public values X_A, X_B, the attacker needs to solve the DL problem to compute x_A or x_B, in order to compute the session key. Therefore, we construct the solver $\mathcal{F}$ which responds the attacker's queries similar to case 1.2, except the following queries:

1. **Send** $(\pi_i^s, (ID_i, ID_j, c_i, c_j), \pi_j^t, (ID_j, ID_i, c_j, c_i))$: If π_i^s or π_j^t is not test session, or $ID_i \neq A/B$, the same response is given as case 1.1. Otherwise, it obtains the private ephemeral key x_i corresponding to X_i by solving DL problem. Then, it computes the session key K according to the protocol specification.
2. **StaticKeyReveal**(ID_i or ID_j): The simulator $\mathcal{F}$ returns the corresponding static keys SK_i or SK_j to the attacker.

The attacker $\mathcal{A}$ chooses one of two sessions as the test session and the other as the matching session among $s(k)$ sessions. Therefore, the success probability of the simulator $\mathcal{F}$ would be:

$$Pr(\mathcal{F}) \geq \frac{2|pw_{ij}|}{s^2(k)t(k)} Pr[S_{2.1}](k) \tag{8}$$

Here, $Pr[S_{2.1}](k)$ denotes the probability that the case 2.1 ($S_{2.1}$) occurs and $t(k)$ is the bound of hash $H_i(\cdot), 1 \leq i \leq 3$ queries called by the attacker $\mathcal{A}$.

Case 2.2 ($S_{2.2}$): The static private key of the owner of π_i^s (SK_A) and the ephemeral key owned by the owner of π_j^t (r_B) have been revealed to the adversary $\mathcal{A}$. In this case, the attacker has two ways to derive the session key K. She can either obtain the static private key SK_B using the 1-sDH solver $\mathcal{S}$ or the ephemeral private key r_A using the DL solver $\mathcal{F}$. The response of the solver $\mathcal{S}$ is similar to case 1.1, except to change the role of $ID_i = A$ to $ID_i = B$ and π_i^s to π_j^t. The response of the solver $\mathcal{F}$ is also similar to case 1.2, except **EphemeralKeyReveal**(π_i^s) can reveal r_B.

Case 2.3 ($S_{2.3}$): The static private key of the owner of π_j^t (SK_B) and the ephemeral key owned by the owner of $\pi_{i,j}^s$ (r_A) have been revealed to the adversary $\mathcal{A}$. Similar to case 2.2, the attacker has two options to derive the session key K. She can either obtain the static private key SK_A using the 1-sDH solver $\mathcal{S}$ or the ephemeral private key r_B using the DL solver $\mathcal{F}$. The response of the solver $\mathcal{S}$ is similar to case 1.1. The response of the solver $\mathcal{F}$ is also similar to case 1.2, except changing $\pi_{i,j}^s$ to π_j^t and SK_A to SK_B.

Case 2.4 ($S_{2.4}$): The ephemeral private key of the owner of π_i^s (r_A) and the ephemeral key owned by the owner of π_j^t (r_B) have been revealed to the adversary $\mathcal{A}$.

In cases 2.2 , 2.3 and 2.4, the attacker $\mathcal{A}$ chooses the test session randomly among $s(k)$ sessions. It chooses participant A as the owner of the test session and B as its peer randomly among $n(k)$ honest parties.

Therefore, together with equation (6), the success probability of the solver $\mathcal{S}$ would be:

$$Pr(\mathcal{S}) \geq max_{i=1.1,2.2,2.3,2.4}\{\frac{|pw_{ij}|}{s(k)n^2(k)t(k)}Pr[S_i](k)\} \tag{9}$$

Here, $Pr[S_{2.2}](k), Pr[S_{2.3}](k), Pr[S_{2.4}](k)$ correspond to probability of cases 2.2 , 2.3 and 2.4, respectively. In addition, the success probability of the solver $\mathcal{F}$ would be:

$$Pr(\mathcal{F}) \geq max\{\frac{|pw_{ij}|}{s(k)n^2(k)t(k)}Pr[S_{1.2}](k), \frac{2|pw_{ij}|}{s^2(k)t(k)}Pr[S_{2.1}](k)\} \tag{10}$$

Therefore, the success probability of the attacker $\mathcal{A}$ to impersonate the key ($Pr[M_1](k)$ would be:

$$Pr[M_1](k) \leq \frac{s(k)n^2(k)t(k)}{|pw_{ij}|}[Pr(\mathcal{S}) + Pr(\mathcal{F})] \tag{11}$$

if $Pr[S_{2.1}] \leq Pr[S_{1.2}]$ or

$$Pr[M_1](k) \leq \frac{s(k)n^2(k)t(k)}{|pw_{ij}|}Pr(\mathcal{S}) + \frac{s^2(k)t(k)}{2|pw_{ij}|}Pr(\mathcal{F}) \tag{12}$$

if $Pr[S_{1.2}] \leq Pr[S_{2.1}]$.

Given the DL and 1-sDH assumptions, the success probability of the solvers $\mathcal{F}$ and $\mathcal{S}$ are negligible. Therefore, the attacker has negligible chance to break the SK-security of our scheme in both scenarios 1 and 2, which completes our proof.

Theorem 2. *Our proposed ID-AKE-PFS scheme achieves s-PFS and KGC-FS security in the id-eCK model under DL and 1-sDH assumption. That is, the functions $Adv_{\mathcal{A}}^{s-PFS}(k), Adv_{\mathcal{A}}^{KGC-FS}(k)$ are negligible in terms of security parameter k.*

Proof. According to security experiment $Exp^{s-(P)FS}_{\pi^s_i,\mathcal{A}}(k)$ (section 3.1), after getting the public parameters in the first step, the attacker may call the following queries:

1. **EphemeralKeyReveal** oracle reveals the nonces r_l, r_m of non-matching sessions to the test session π^s_i corresponding to previous public ephemeral keys X_l, X_m such that $X_l \neq X_i, X_m \neq X_j$.
2. **SessionkeyReveal** oracle reveals the session key K of non-matching sessions to the test session π^s_i.
3. It applies **Send** query to non-matching sessions to the test session π^s_i to obtain the public parameters c_l/c_m such that $c_l \neq c_i, c_m \neq c_j$. Here, c_i, c_j are the messages of the test session π^s_i.
4. It might try to apply **Send** query to the test session π^s_i and replay public values c_l/c_m of the non-matching sessions to the test session π^s_i. However, the time dependency of the values c_l/c_m, prevent such replay attack.
5. It might try to forge a secret/public ephemeral key $x_i = H_2(SK_i, r_i), X_i = SK_i^{x_i}$ encrypt them as $c_i = Enc(X_i, H_3(pw_{ij}, t_i))$ and apply **Send**(c_i) query to the test session π^s_i. Toward this goal, the attacker needs the 1-sDH problem simulator $\mathcal{S}$ to obtain the static secret key SK_i and the password pw_{ij} (with probability $\frac{1}{|pw_{ij}|}$) for encryption of the public ephemeral key. Based on the 1-sDH assumption, the attacker has negligible success probability in this step. As the test session s(k) and participants are chosen among n(k) honest parties in a random manner, the success probability of the simulator $\mathcal{S}$ would be as follows:

$$Pr(\mathcal{S}) \geq \frac{|pw_{ij}|}{s(k)n^2(k)} Pr[\textbf{Send}(c_i)](k) \tag{13}$$

After choosing a fresh instance π^s_i in the third step, the attacker $\mathcal{A}$ obtains SK_A/SK_B in case of s-FS and SK_A, SK_B in case of s-PFS in the fourth step. It can still query oracles as step 2. In this case, the attacker faces similar scenario as the case 2.1 ($S_{2.1}$) in $Exp^{SK}_{\pi^s_i,\mathcal{A}}(k)$ experiment. Therefore, the success probability of the attacker in this experiment would be:

$$Pr[Exp^{s-(P)FS}_{\pi^s_i,\mathcal{A}}(k) = 1] \leq \frac{s(k)n^2(k)}{|pw_{ij}|} Pr(\mathcal{S}) + \frac{s^2(k)t(k)}{2|pw_{ij}|} Pr(\mathcal{F}) \tag{14}$$

Further, the **KGCMasterKeyReveal** enables the attacker to obtain the static secret keys of any participant such as SK_A and SK_B in the fourth step of $Exp^{KGC-FS}_{\pi^s_i,\mathcal{A}}(k)$ experiment. As no other parameter of the session key depend upon the KGC secret key w, this experiment would resemble $Exp^{s-PFS}_{\pi^s_i,\mathcal{A}}(k)$ experiment.

Theorem 3. *Our proposed ID-AKE-PFS scheme achieves offline and online guessing attack security under 1-sDH assumption. That is, the functions $Adv^{OGA}_{\mathcal{A}}(k)$ is negligible in terms of security parameter k.*

Proof. In order to mount offline/online guessing attack, the attacker $\mathcal{A}(k)$ performs the experiment $Exp^{OGA}_{\pi_i^s,\mathcal{A}}(k)$ mentioned in section 3.1 and either:

1. Obtain the user's device parameters by querying $\mathbf{Extract}(\pi_i^s)$ oracle to obtain hpw_{ij}, ESK_i, n_i. Given the modular property of hpw_{ij}, the success probability of the attacker $\mathcal{A}(k)$ to derive pw_{ij} would be $\frac{1}{|pw_{ij}-n_i|}$. In addition, obtaining the password pw_{ij} from the value ESK_i would also require 1-sDH problem solver $\mathcal{S}$ to derive SK_i and $pw_{ij} = SK_i \oplus ESK_i$
2. It can issue **Send** and $\mathbf{EphemeralKey}(\pi_i^s)$ query to instance π_i^s as the test session to obtain c_i, c_j, r_i, r_j. It queries $\mathbf{TestPW}(\pi_i^s)$ and guesses pw_{ij} as $pw_{ij}^{'}$. However, in order to compute X_i/X_j and verify his guess by decrypting c_i/c_j, the attacker needs 1-sDH problem solver $\mathcal{S}$ to obtain SK_i/SK_j. The test session π_i^s is chosen among s sessions and n^2 parties.
3. It might generate a nonce $r_i^{'}$, guess pw_{ij} as $pw_{ij}^{'}$, compute X_i using 1-sDH problem solver $\mathcal{S}$ and issue $\mathbf{Send}(X_i)$ query to instance π_i^s. As the responder U_j revokes the user U_i after $N_i = 5$ unvalid trials, the success probability of the attacker would be $\frac{n(k)^2 s(k) Pr(\mathcal{S})}{N_i |pw_{ij}-n_i|}$.

Given the above ways, the success probability of the attacker would be:

$$Pr[Exp^{OGA}_{\pi_i^s,\mathcal{A}}(k) = 1] = \frac{n(k)^2 s(k) Pr(\mathcal{S})}{N_i |pw_{ij} - n_i|} + \frac{1}{|pw_{ij} - n_i|} \tag{15}$$

7 Our fix to Lian et al.'s scheme [27]

As shown in Figure 2, we suggest a fix to Lian et al.'s scheme [27] to achieve KGC-FS and user authentication. Toward this goal, each party U_i multiplies the nonce r_i with the generator g_1 to compute R_i. The key derivation function (KDF) takes the value $R_j^{r_i}$ as input. Therefore, compromise of the KGC master key msk doesn't enable the attacker to derive the past session keys, as the adversary has negligible chance to derive the KDF input $R_j^{r_i}$, based on the DH assumption. In addition, to prevent device stolen attack, each party retrieves the static key SK_i from the user's device after inputting his password PW such that $SK_i = ESK_i \oplus PW$.

8 Implementation

In this section, we analyze and compare the performance of our ID-AKE-PFS scheme with previous single pairing-based AKE schemes [26, 27] through a proof-of-concept implementations. Using Raspberry Pi 3 Model B+ with 8 GB RAM and ARM Cortex-A53 @ 1.4 GHz processor, we implement the main primitives of our scheme and previous single pairing-based AKE schemes [26, 27] by employing MIRACLE library for pairing operation and HMAC method from OPENSSL library for the hash function H. We use Koblitz curve secp256k1 and secp521r1 for G_1 and G_2, respectively. The curve parameters are chosen

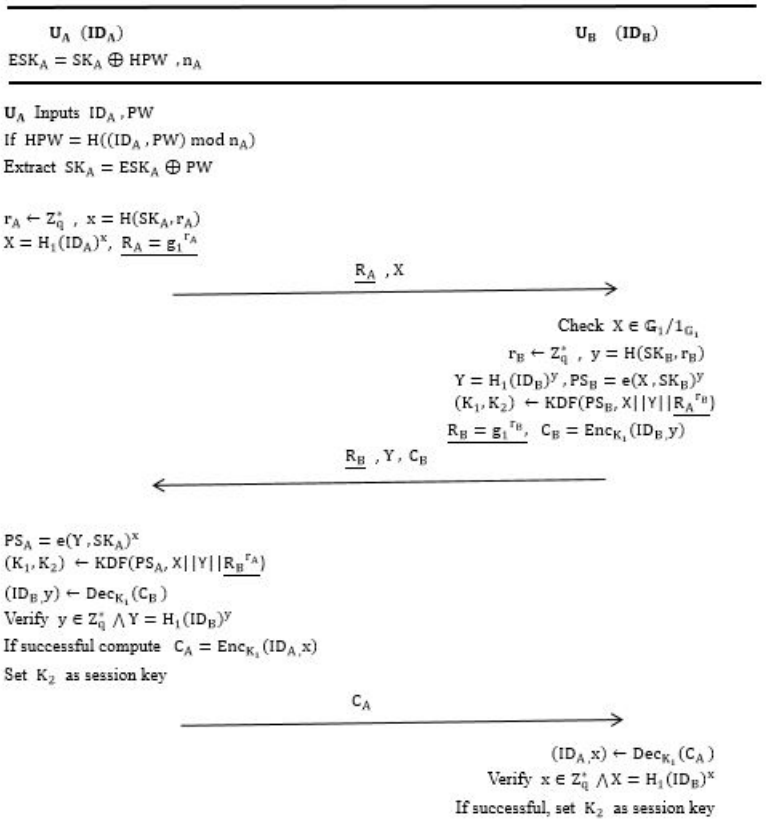

Fig. 2. Lian et al.'s scheme [26] with KGC-FS

according to standards for efficient cryptography [34]. In Table 2, we summarize the main results of our performance comparison. Our implementation results for each primitive are the average time out of 100 iterations. As shown in the computation column, our ID-AKE-PFS scheme has two less curve multiplication operation and two less addition operations than Tomida et al.[26] In addition, no encryption/decryption operation is employed in contrast to scheme Lian et al. [27]. In overall, our scheme has reduced CPU clock cycle and imposes the least computation cost compared to previous schemes [26, 27]. Further the bit length of transmitted messages are shown in the bit length column. The group order G_1 of Tomida et al.[26] scheme is 462 bits, with overall overhead of $2 \times 462 = 924$ bits. Lian et al. [27] scheme imposes $2H_1 + 2C = 3968$ bits , as SHA-256 is used for hash function H_1 with ciphertext of size 216 bytes. In our scheme, we choose 521-bit groups G_2 for the public keys leading to $2 \times 512 = 1024$ bits of communication overhead. This is a remarkable reduction compared to Lian et al. scheme [27].

Table 2. The performance comparison of our scheme vs. previous single pairing-based schemes

Scheme	Assumption	Pairing type	Computation (ms)	Clock cycle(million)	Bit length(bit)
Tomida et al.[26]	XDHT, q-Gap-BCA	Asym	$1P + 3Add_{G_1}$ $4H + 5Mult_{G_1} = 28.407$	3.977	$2G_1 = 924$
Lian et al. [27]	Gap-BDH, AEAD	Asym , sym	$1P + E + D + 2Mult_{G_1}$ $+3H + 1Mult_{G_T} = 25.68$	3.596	$2H_1 + 2C = 3968$
Our scheme	1-sDH , DL	Asym	$1P + 1Add_{G_1} + 3H$ $+1Mult_{G_1}$ $+2Mult_{G_2} = 24.464$	3.425	$2G_2 = 1024$

9 Conclusion

In conclusion, our research delves into the realm of Identity-based Password Authentication and Key Establishment (ID-PAKE). Despite the existence of previous PAKE schemes, including those proposed by Beguinet et al. (ACNS 2023), Pan et al. (ASIACRYPT 2023), Abdallah et al. (CRYPTO 2020), we identified shortcomings in terms of crucial security properties such as weak/strong Perfect Forward Secrecy (s-PFS), user authentication, and resistance to replay attacks.

Addressing these limitations, our contribution introduces a highly efficient ID-PAKE scheme boasting both s-PFS and KGC-FS with the remarkable achievement of requiring only two rounds of communication, each involving a single pairing operation. In contrast to previous single pairing-based schemes, such as those by Tomida et al. (ESORICS 2019) and Lian et al. (ESORICS 2022), our proposed scheme demonstrates resilience against s-PFS, KGC-FS attacks, and replay attacks.

To substantiate the security claims, we provide rigorous proofs under standard assumptions like Discrete Logarithms (DL) and q-strong Diffie-Hellman (q-sDH) within the ID-eCK model. Furthermore, a proof-of-concept implementation of our scheme against previous single pairing-based schemes showcases not only the robust security but also the least computation cost. In terms of communication cost, our scheme holds a favorable position, striking a balance between efficiency and security compared to its predecessors.

This work advances the landscape of ID-PAKE, providing a secure and efficient solution with notable improvements over existing schemes, thus contributing to the ongoing discourse on password authentication and key establishment.

References

1. K. Y. Choi, J. Cho, J. Y. Hwang, T. Kwon, Constructing efficient PAKE protocols from identity-based KEM/DEM, in: H. Kim, D. Choi (Eds.), Information Security Applications - 16th International Workshop, WISA 2015, Jeju Island, Korea, August 20-22, 2015, Revised Selected Papers, Vol. 9503 of Lecture Notes in Computer Science, Springer, 2015, pp. 411–422. doi:10.1007/978-3-319-31875-2_34.
URL https://doi.org/10.1007/978-3-319-31875-2_34

2. S. Shin, A strengthened PAKE protocol with identity-based encryption, IEICE Trans. Inf. Syst. 105-D (11) (2022) 1900–1910. doi:10.1587/TRANSINF.2022NGP0009.
URL https://doi.org/10.1587/transinf.2022ngp0009
3. M. Abdalla, D. Pointcheval, Simple password-based encrypted key exchange protocols, in: A. Menezes (Ed.), Topics in Cryptology - CT-RSA 2005, The Cryptographers' Track at the RSA Conference 2005, San Francisco, CA, USA, February 14-18, 2005, Proceedings, Vol. 3376 of Lecture Notes in Computer Science, Springer, 2005, pp. 191–208. doi:10.1007/978-3-540-30574-3_14.
URL https://doi.org/10.1007/978-3-540-30574-3_14
4. D. Pointcheval, G. Wang, VTBPEKE: verifier-based two-basis password exponential key exchange, in: R. Karri, O. Sinanoglu, A. Sadeghi, X. Yi (Eds.), Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, AsiaCCS 2017, Abu Dhabi, United Arab Emirates, April 2-6, 2017, ACM, 2017, pp. 301–312. doi:10.1145/3052973.3053026.
URL https://doi.org/10.1145/3052973.3053026
5. M. Abdalla, M. Barbosa, T. Bradley, S. Jarecki, J. Katz, J. Xu, Universally composable relaxed password authenticated key exchange, in: D. Micciancio, T. Ristenpart (Eds.), Advances in Cryptology - CRYPTO 2020 - 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17-21, 2020, Proceedings, Part I, Vol. 12170 of Lecture Notes in Computer Science, Springer, 2020, pp. 278–307. doi:10.1007/978-3-030-56784-2_10.
URL https://doi.org/10.1007/978-3-030-56784-2_10
6. J. Pan, R. Zeng, A generic construction of tightly secure password-based authenticated key exchange, IACR Cryptol. ePrint Arch. (2023) 1334.
URL https://eprint.iacr.org/2023/1334
7. J. Pan, R. Zeng, A generic construction of tightly secure password-based authenticated key exchange, IACR Cryptol. ePrint Arch. (2023) 1334.
URL https://eprint.iacr.org/2023/1334
8. H. Beguinet, C. Chevalier, D. Pointcheval, T. Ricosset, M. Rossi, Get a CAKE: generic transformations from key encaspulation mechanisms to password authenticated key exchanges, in: M. Tibouchi, X. Wang (Eds.), Applied Cryptography and Network Security - 21st International Conference, ACNS 2023, Kyoto, Japan, June 19-22, 2023, Proceedings, Part II, Vol. 13906 of Lecture Notes in Computer Science, Springer, 2023, pp. 516–538. doi:10.1007/978-3-031-33491-7_19.
URL https://doi.org/10.1007/978-3-031-33491-7_19
9. Ieee standard specification for password-based public-key cryptographic techniques, IEEE Std 1363.2-2008 (2009) 1–140doi:10.1109/IEEESTD.2009.4773330.
10. D. R. Stinson, Cryptography - theory and practice, Discrete mathematics and its applications series, CRC Press, 1995.
11. F. Hao, P. Y. A. Ryan, J-PAKE: authenticated key exchange without PKI, Trans. Comput. Sci. 11 (2010) 192–206. doi:10.1007/978-3-642-17697-5_10.
URL https://doi.org/10.1007/978-3-642-17697-5_10
12. A. Shamir, Identity-based cryptosystems and signature schemes, in: Annual International Cryptology Conference, 1984.
URL https://api.semanticscholar.org/CorpusID:1402295
13. R. Canetti, H. Krawczyk, Security analysis of ike's signature-based key-exchange protocol, in: Annual International Cryptology Conference, 2002.
URL https://api.semanticscholar.org/CorpusID:7870446

14. N. P. Smart, An identity based authenticated key agreement protocol based on the weil pairing, IACR Cryptol. ePrint Arch. 2001 (2002) 111.
URL https://api.semanticscholar.org/CorpusID:6745379
15. H. Krawczyk, Hmqv: A high-performance secure diffie-hellman protocol.
URL https://api.semanticscholar.org/CorpusID:264401565
16. D. Fiore, R. Gennaro, Making the diffie-hellman protocol identity-based, IACR Cryptol. ePrint Arch. 2009 (2010) 174.
URL https://api.semanticscholar.org/CorpusID:10018265
17. A. Fujioka, F. Hoshino, T. Kobayashi, K. Suzuki, B. Ustaoglu, K. Yoneyama, id-eck secure id-based authenticated key exchange on symmetric and asymmetric pairing, IEICE Trans. Fundam. Electron. Commun. Comput. Sci. 96-A (2013) 1139–1155.
URL https://api.semanticscholar.org/CorpusID:40966179
18. H. Huang, Z. Cao, An id-based authenticated key exchange protocol based on bilinear diffie-hellman problem, in: ACM Asia Conference on Computer and Communications Security, 2009.
URL https://api.semanticscholar.org/CorpusID:11921408
19. L. Chen, C. Kudla, Identity based authenticated key agreement protocols from pairings, 16th IEEE Computer Security Foundations Workshop, 2003. Proceedings. (2003) 219–233.
URL https://api.semanticscholar.org/CorpusID:6182192
20. R. M. Daniel, E. B. Rajsingh, S. Silas, An efficient eck secure identity based two party authenticated key agreement scheme with security against active adversaries, Inf. Comput. 275 (2020) 104630.
URL https://api.semanticscholar.org/CorpusID:225344592
21. T.-Y. Wu, Y.-Q. Lee, C. Chen, Y. Tian, N. A. Al-Nabhan, An enhanced pairing-based authentication scheme for smart grid communications, Journal of Ambient Intelligence and Humanized Computing (2021) 1–13.
URL https://api.semanticscholar.org/CorpusID:234279961
22. M. Nikravan, A. Reza, A multi-factor user authentication and key agreement protocol based on bilinear pairing for the internet of things, Wireless Personal Communications 111 (2020) 463–494.
URL https://api.semanticscholar.org/CorpusID:208126942
23. S. Rajaram, T. Maitra, S. Vollala, N. Ramasubramanian, R. Amin, euasbp: enhanced user authentication scheme based on bilinear pairing, Journal of Ambient Intelligence and Humanized Computing 11 (2020) 2827–2840.
URL https://api.semanticscholar.org/CorpusID:198310826
24. S. S. M. Chow, K. R. Choo, Strongly-secure identity-based key agreement and anonymous extension, in: J. A. Garay, A. K. Lenstra, M. Mambo, R. Peralta (Eds.), Information Security, 10th International Conference, ISC 2007, Valparaíso, Chile, October 9-12, 2007, Proceedings, Vol. 4779 of Lecture Notes in Computer Science, Springer, 2007, pp. 203–220. doi:10.1007/978-3-540-75496-1_14.
URL https://doi.org/10.1007/978-3-540-75496-1_14
25. Y. Wang, Efficient identity-based and authenticated key agreement protocol, IACR Cryptol. ePrint Arch. (2005) 108.
URL http://eprint.iacr.org/2005/108
26. J. Tomida, A. Fujioka, A. Nagai, K. Suzuki, Strongly secure identity-based key exchange with single pairing operation, in: European Symposium on Research in Computer Security, 2019.
URL https://api.semanticscholar.org/CorpusID:202579607

27. H. Lian, T. G. Pan, H. Wang, Y. Zhao, Identity-based identity-concealed authenticated key exchange, in: European Symposium on Research in Computer Security, 2021.
URL https://api.semanticscholar.org/CorpusID:238417161
28. R. Canetti, H. Krawczyk, Analysis of key-exchange protocols and their use for building secure channels, in: International Conference on the Theory and Application of Cryptographic Techniques, 2001.
URL https://api.semanticscholar.org/CorpusID:432763
29. D. Lightbody, D. Ngo, A. Temko, C. C. Murphy, E. M. Popovici, Attacks on iot: Side-channel power acquisition framework for intrusion detection, Future Internet 15 (5) (2023) 187. doi:10.3390/FI15050187.
URL https://doi.org/10.3390/fi15050187
30. A. M. Odlyzko, Discrete logarithms: The past and the future, Designs, Codes and Cryptography 19 (2000) 129–145.
URL https://api.semanticscholar.org/CorpusID:3201066
31. N. Tanaka, T. Saito, On the q-strong diffie-hellman problem, IACR Cryptol. ePrint Arch. 2010 (2010) 215.
URL https://api.semanticscholar.org/CorpusID:39140774
32. I. Shparlinski, Computational Diffie-Hellman Problem, Springer US, Boston, MA, 2011, pp. 240–244. doi:10.1007/978-1-4419-5906-$5_8$82.
$URLhttps://doi.org/10.1007/978-1-4419-5906-5_882$
33. B. A. LaMacchia, K. E. Lauter, A. Mityagin, Stronger security of authenticated key exchange, IACR Cryptol. ePrint Arch. 2006 (2006) 73.
URL https://api.semanticscholar.org/CorpusID:2548277
34. Sec1. elliptic curve cryptography. standards for efficient cryptography group, https://www.secg.org/sec2-v2.pdf, accessed: 2023-12-12.