Accepted by IEEE Transactions on Information and Forensics & Security, 2024, DOI: 10.1109/TIFS.2024.3402073

AB-PAKE: Achieving Fine-Grained Access Control and Flexible Authentication

Mi Song and Ding Wang

***Abstract*—Two-factor authentication provides a strong defense against account compromise. However, traditional two-factor authentication schemes cannot provide users with much flexibility and fine-grained authorization. In this work, we present an efficient design of Attribute-Based Password Authenticated Key Exchange (AB-PAKE) protocol, ensuring that only two legitimate users with desired attributes and correct passwords can establish a shared session key. We, for the first time, tackle the problem of "how to enhance a peer-to-peer PAKE scheme by using a storage device (e.g., a smart-phone, a USB token, or a personal computer that the user logs in), such that even if ephemeral secret keys of two participants have been leaked, it still provides user privacy protection and truly two-factor security". AB-PAKE works well in peer-to-peer (i.e., end-to-end) scenarios where the participants expect to hide their real identity information and the peer is enforced to satisfy the defined conditions (aka authentication policy). It achieves flexibility, privacy preservation, and dynamic access control lacking in prior authentication proposals. In addition, our work mitigates a practical threat in authenticated key exchange schemes, namely, the ephemeral secret leakage attack. We aim to increase the attack difficulty and limit password leakage even if the user's long-term key or ephemeral key is leaked. The proposed protocol is also round-optimal, i.e., it is a single-round protocol consisting of only two message flows among the parties. Our new construction of AB-PAKE protocol reduces the number of pairing operations to be constant and supports richer policies. Provable security and practicality are demonstrated by comprehensive analysis.**

***Index Terms*—Two-factor authentication, key exchange, flexible access control, attribute authentication, eCK model.**

I. Introduction

In today's web environment, passwords serve as the extensively prevalent method for user authentication [1], [2]. Within the framework of password-based authenticated key exchange (PAKE) protocols (e.g., [3]–[7]), two parties with a low-entropy password can exchange a session key with high-entropy for facilitating secure data communication. Nonetheless, passwords are susceptible to various forms of attacks, including online, offline, or hybrid guessing attacks [8]. Among these, the offline guessing attack poses a significant threat as it goes undetected by the security team and application server logs, rendering conventional protection measures like account lockouts ineffective [9]. Moreover, the widespread practice of password reuse across various services amplifies the security risk, as compromising a single server can trigger a "domino effect", jeopardizing the security of all other servers safeguarded by the same or slightly modified password [10].

Recent years it has become increasingly common to hear about the compromise of high-profile web services, leading to large-scale password leaks. Some notable incidents, such as the massive 3 billion Yahoo leak [11], the staggering 8.4 billion Rockyou leak [12], and the concerning 3.8 billion DarkBeam leak [13], serve as alarming reminders of the urgent need for robust security measures in safeguarding users' passwords.

To mitigate the risk of users' passwords being exposed to servers, significant efforts have been dedicated to establishing secure password-based authentication schemes, which can be summarized into three main categories: (i) Providing password security by salted password hashing, which involves generating an authentication credential by computing the hash value of both the password and a randomly generated number (e.g., PBKDF2 [14]). (ii) Providing password security by emerging cryptographic primitives. For example, to hash the password, the server employs a memory-hard hash function (e.g., Balloon hashing [15] and Scrypt [16]). (iii) Providing password security by introducing an independent crypto server to harden the users' passwords (e.g., [17]–[20]). Specifically, the authentication credentials are generated by combining the user's password with a secret key held by the server. These password hardening schemes all conform to the perspective that, "if a password-based authentication system has already been established it is often more economically viable to harden it against possible attacks than to replace it from scratch" [18].

Although these methods [14]–[16] force attackers to spend more effort in compromising users' passwords, they rely on a critical assumption: the unwavering honesty and reliability of the server over an extended period. Unfortunately, if the server becomes compromised, the security of these schemes would be irreparably compromised as well. It has been revealed [21] that, the password-hardening (PH) protocols [17]–[20] also introduce a single-point-of-failure. If the crypto server is inaccessible (e.g., due to malicious attacks or network failure), the data would become unavailable to the clients since the online service provider cannot perform decryption alone.

To tackle the problem of password leakage resulting from a compromised server, some password-based threshold authentication schemes (e.g., [21]–[23]) were proposed. These schemes aim to overcome the single-point-of-failure issue by employing multiple servers for user authentication. In such schemes, the authentication credential is derived from

This work was supported in part by the National Natural Science Foundation of China under Grant 62222208; and in part by the Natural Science Foundation of Tianjin, China, under Grant 21JCZXJC00100 and Grant 21JCZDJC00190. The corresponding author is Ding Wang.

Mi Song and Ding Wang are with the College of Cyber Science, Nankai University, Tianjin 300350, China, and also with Key Laboratory of Data and Intelligent System Security (Ministry of Education), Nankai University, Tianjin 300350, China, and also with Tianjin Key Laboratory of Network and Data Security Technology, Nankai University, Tianjin 300350, China. (Email: songmi@mail.nankai.edu.cn; wangding@nankai.edu.cn).

a combination of the user's password and a server-side key, distributed among all servers using a threshold-based manner. Provided that attackers are unable to breach the required threshold number of servers, the password remains inaccessible. Besides, in password-based threshold single-sign-on authentication scheme [22], to resist malicious users who acquire a sufficient number of server-derived passwords and perform offline password guessing attacks, different users should generate unique server-side keys and distribute them across all servers. This significantly increases the computation and communication costs, resulting in poor usability [24].

Concurrently, it has been observed that existing schemes (e.g., PASTA [22]), are susceptible to a specific kind of attack called online password testing attack (OPTA) [23]. In this type of attack, an identity server acts as an oracle, responding to user authentication queries without having knowledge about the authentication outcomes, which severely undermines the system's security. Efforts are required to strike a balance between enhanced security against password leakage while ensuring the usability of authentication schemes.

To improve the system security and usability, the methods (e.g., [25]–[28]) employing passwords and smart cards for user authentication are proposed, which are essentially a common form of two-factor authentication (2FA). In these methods, the user is authenticated by entering a correct password and holding a smart card. In the case of password leakage [27], the auxiliary smart card forms a "second line". Besides, there are other forms of two-factor authentication methods, such as using biometrics and smart cards for authentication (e.g., [29]). These methods leverage the unique characteristics of biometric traits, such as fingerprints or facial features, in combination with smart cards to verify the user's identity. In all, two-factor authentication can provide better security for user accounts than password-only authentication.

However, existing two-factor authentication methods (e.g., [25]–[27], [29]) primarily focus on ensuring the confidentiality of user passwords, often neglecting the importance of privacy protection for the users themselves. Informally, user privacy encompasses several aspects, including maintaining user identity anonymity and untraceability, safeguarding personal information and activities, and enabling fine-grained access controls as defined by the users. User anonymity requires that the attacker cannot determine whether two authenticated key exchanges are conducted by the same user, while user untraceability can make it extremely challenging or almost impossible to link or track different sessions to the same user. In other words, it means that the user's identity is neither computable nor traceable by the attacker [30]. Fine-grained access control entails providing users with the ability to authenticate flexibly in a privacy-protection manner [31], e.g., users can pre-define a group of users with whom they communicate (but without knowing the exact identities of the individuals), which is particularly crucial in private communication scenarios [32].

To the best of our knowledge, attribute-based AKE (AB-AKE) protocol [33]–[37] is a promising solution. It can help two users establish a shared session key in the communication environment without disclosing identity privacy. The core idea is to use the user's attributes (e.g., age, gender, and hobbies) to describe and represent users, thereby achieving privacy protection for user's identities. This feature (i.e., fuzzy identity) makes the AB-AKE protocol particularly suitable for a number of application scenarios, such as voting systems [38], mobile health networks [39] and genomic testing [40].

A. Related Work

In 2010, Gorantla et al. [33] proposed a game-based definition of attribute-based authenticated key exchange (AB-AKE) and highlight that such a scheme aligns more closely with group key exchange rather than standard AKE protocols. AB-AKE protocols typically involve multiple public-key operations per gate of the policy formula. Although recent schemes [36], [41] for attribute-based encryption that support general circuits demonstrate the feasibility of AB-AKE, they are primarily of theoretical interest due to their reliance on computationally intensive underlying primitives.

Summarizing the recent studies, we observe that there is a significant trend towards, leveraging attribute-based mechanisms for enhancing privacy in key management and authentication processes across various domains, from vehicular communications to cloud services [42]–[45]. Tan et al. [42] present a privacy-preserving attribute-based authenticated key management scheme specifically designed for accountable vehicular communications. The scheme incorporates an attribute-based secret sharing scheme to enable a flexible authentication mechanism with dynamic access policies. This design allows for enhanced privacy while maintaining the ability to manage keys in vehicular communication systems effectively. Sucasas et al. [43] present a novel attribute-based pseudonymity solution for privacy-preserving authentication in cloud services, allowing users to generate unlinkable pseudonyms with embedded attributes while keeping the attribute set private. Luo et al. [44] propose an efficient attribute-based authenticated encryption with keyword search scheme that is resistant to quantum computer attacks and keyword guessing attacks.

However, the aforementioned authentication schemes fail to accomplish continuous and secure user authentication. The combination of attribute-based encryption with password authentication provides a flexible and scalable authentication mechanism. Passwords can be used to verify the user's identity, while attributes can be utilized to define access policies and authorization conditions, which offers more personalized and fine-grained access control. Regrettably, this mechanism has not received much attention in the field. In this paper, we delve into the design and analysis of AB-PAKE protocols to achieve fine-grained access control and flexible authentication.

B. Motivations

With the increasing demand of authentication and the urgency of privacy protection, authentication mechanisms combined with identity privacy (e.g., anonymity, untraceability and verifiability) have emerged. Since the seminal work of Diffie-Hellman (DH) protocol [46] was proposed, considerable research endeavors have been dedicated to designing both practical and efficient authentication key exchange schemes.

However, most two-factor authentication schemes (e.g., [25]–[28]) only require the user to enter a pre-registered password and other credentials (e.g., smart card) for successful authentication, which does not provide users with flexibility. Especially in an end-to-end authentication environment, the user cannot choose the desired entity for communication. Besides, most of the security proofs of such schemes are formally proved under the BPR model [3], they generally assume that users' ephemeral secret keys will not be compromised. However, this assumption may not scale well in practice. Compared with the long-term secret, the ephemeral secret is more vulnerable to leakage because the former may be stored in a hardware-protected area while the latter will be typically stored on disk (i.e., general memory devices) [47]. Also, there are still many similar incidents in recent years, like the Heartbleed attack [48], Meltdown attack [49], and Spectre attack [50]. Thus this paper focuses on the following question:

Can we design a flexible two-factor authentication key exchange scheme that provides user privacy protection and truly two-factor security even if ephemeral secret keys have been leaked?

To address this problem, we present a novel two-party attribute-based password authentication key exchange scheme within the enhanced Canetti-Krawczyk (eCK) model [51], which can ensure the security of the session key even if both parties simultaneously disclose their ephemeral keys. Additionally, it is worth noting that most of the existing two-factor authentication key exchange protocols (e.g., [26], [52], [53]) need at least three rounds to establish a secure session key. Despite the desire for implicit authentication alone, the communication rounds in these protocols cannot be further minimized. Consequently, the research challenge lies in finding solution which can reduce the number of communication rounds, while simultaneously meeting the security requirements of underlying cryptographic primitives.

C. Contributions

Considering the potential advantages of attribute authentication, such as fine-grained access control and flexible authentication, it makes perfect sense to design a round-optimal attribute-based password authenticated key exchange protocol. To the best of our knowledge, Abdalla-Pointcheval's work SPAKE [54] may be the closest to our work, but our work aims to design an attribute-based password authentication that not only encompasses all the inherent properties of SPAKE, but also achieves flexible mutual authentication and user anonymity. Specifically, the authentication can be performed only when the user possesses both the correct password and attributes (e.g., age, nationality and organization) that satisfy the defined authentication access policy (e.g., $18 \leq$ age ≤ 35, no nationality restrictions and Harvard University). Meanwhile, to resist ephemeral secret leakage (ESL) attacks, we propose a method that integrates both static and ephemeral secret keys simultaneously, using the trick proposed by LaMacchia et al. [51]. Overall, our work makes the following key contributions:

- **Flexible two-factor authentication and identity anonymous protection.** We present a novel construction of a flexible attribute-based password authentication scheme with constant number of pairings. In our approach, the identity of each participant is represented by a unique set of attributes. The key exchange process is only carried out when the user's attributes satisfy the intended peer's authentication policy, and the user's password is verified as correct. This design enables fine-grained access control and achieves the user identity anonymity.
- **Round-optimal attribute-based PAKE protocol.** We give the construction of a one-round AB-PAKE protocol that requires mere two pass interactions within one session. The foundation of our approach relies on the attribute-based signcryption (ABSC) structure, which serves as the building block in our protocol. A notable benefit of ABSC in our protocol is that it acts as an implicit verifier, eliminating the need for additional verification. This means that only the intended partner can successfully pass the authentication, preventing unauthorized access and reducing the communication rounds.
- **Provably secure attribute-based PAKE protocol.** Through rigorous analysis, we prove the semantic security of the proposed protocol within the eCK model [51]. Our work studies practical security risks to AKE schemes, such as the compromise of secret keys (e.g., leakage or theft), which poses a significant threat to the security provided by existing PAKE schemes. We demonstrate the effectiveness of the proposed scheme through a comprehensive security analysis, proving its security to defend against various attacks within our defined adversary model. Moreover, the scheme meets all the criteria outlined in our evaluation framework, underscoring its robustness and applicability.

II. Preliminaries

In this section, we present some relevant preliminaries necessary for the understanding of this paper.

Notation: The security parameter is represented by the symbol κ. Define $y \xleftarrow{R} Y$, with y be a random selection from the finite set Y, chosen based on a uniform distribution. The set of integers $\{1, 2, ..., m\}$ is denoted by $[m]$. To enhance the readers' comprehension of the specific construction of this paper, next, we present an understanding of bilinear map and Lagrange Interpolation, along with related complexity assumptions and an explanation of access policy.

A. Bilinear Map

Let us consider two multiplicative cyclic groups G and G_T with prime order p. Assume that g is a generator in group G. The bilinear map $e : G \times G \rightarrow G_T$ has following properties:

1) Bilinearity: If $\forall\ \alpha, \beta \in Z_p$, $g_1, g_2 \in G$, we can get $e(g_1^{\alpha}, g_2^{\beta}) = e(g_1, g_2)^{\alpha\beta}$.
2) Non-degeneracy: $\exists\ g_1, g_2 \in G$, we have $e(g_1, g_2) \neq 1$.
3) Computability: There is always an efficient algorithm exists to get $e(g_1, g_2)$, where $g_1, g_2 \in G$.

B. Lagrange Interpolation

Given any polynomial $f(z)$ of degree $d-1$, it can be reconstructed using d distinct points $(z_1, f(z_1)), (z_2, f(z_2)), \ldots, (z_d, f(z_d))$ utilizing the following approach:

$$f(z) = \sum_{i \in \Omega} f(i) \cdot \Delta_{i,\Omega}(z).$$

Here, $\Omega = \{z_1, z_2, \ldots, z_d\}, \Delta_{i,\Omega}(z) := \prod_{j \in \Omega, j \neq i} \frac{z-j}{i-j}$ and it is equal to the Lagrange coefficient of i in Ω, where z denotes the variable. Consequently, we have the ability to get the value of $f(z)$ for any $z \in Z_p$ if we have d distinct polynomial values. Nonetheless, only when $d-1$ distinct polynomial values are provided, the security of the remaining polynomial values is guaranteed unconditionally.

C. Complexity Assumptions

***Definition** 1 (Computational Diffie-Hellman Assumption):* [55] Let g be a group generator of G. We define the Computational Diffie-Hellman assumption (CDH assumption) to hold when, for any probabilistic polynomial-time (PPT) algorithm, the advantage $Adv_{\mathcal{A}}^{CDH} := |Pr[\mathcal{A}(q, G, g, g^{\alpha}, g^{\beta}) \rightarrow g^{\alpha\beta} | \alpha, \beta \xleftarrow{R} Z_q, g \xleftarrow{R} G] = \epsilon^{CDH}$ of PPT algorithm in successfully solving this problem is negligible.

***Definition** 2 (Decision modified q-BDHE Assumption):* The Decision modified q-Bilinear Diffie-Hellman Exponent (Dmq-BDHE) problem is formally defined as follows: given parameters Σ, Φ, where $\Phi := [g, g^s, g^a, g^{a^2}, \cdots, g^{a^q}, g^{a^{q+2}}, \cdots, g^{a^{2q}}, g^{s(at+a)}, g^{ta}, g^{ta^2}, \cdots, g^{ta^q}, g^{ta^{q+2}}, \cdots, g^{ta^{2q}}]$, select numbers $a, s, t \in \mathbb{Z}_p^*, Z \in \mathbb{G}_T$ are unknown, the goal is to determine whether $Z \in e(g,g)^{sa^{q+1}}$ or Z represents a random element of G_T. The Dmq-BDHE assumption is said to hold if the advantage $Adv_{\mathcal{A}}^{Dmq-BDHE} := |Pr[1 \leftarrow \mathcal{A}(\Sigma, \Phi, e(g,g)^{sa^{q+1}})] - Pr[1 \leftarrow \mathcal{A}(\Sigma, \Phi, Z)]$ of a PPT algorithm in solving this problem is considered negligible.

***Definition** 3 (Gap modified q-BDHE Assumption):* Consider the oracle $\mathcal{O}^{mq-BDHE}$, given an input $(g^a, g^{a^q}, g^s, e(g,g)^d)$, yields a value of 1 if $sa^{q+1} = d \bmod p$ holds, and 0 otherwise. Define mq-BDHE function $f^{mq-BDHE}$ as $f^{mq-BDHE}(\Phi) = e(g,g)^{sa^{q+1}}$. An adversary $\mathcal{A}$ is provided with a randomly selected input Φ and has oracle access to the $\mathcal{O}^{mq-BDHE}$ oracle. The objective of $\mathcal{A}$ is to compute $f^{mq-BDHE}(\Phi)$. Then we can use $Adv_{\mathcal{A}}^{Gmp-BDHE} := Pr[\mathcal{A}^{O^{mq-BDHE}}(\Phi) = f^{mq-BDHE}(\Phi)]$ to represent the advantage of a PPT algorithm in solving this problem. The Gmq-BDHE assumption is considered to be hold if for any PPT adversary $\mathcal{A}$, the advantage $Adv_{\mathcal{A}}^{Gmq-BDHE}$ in solving this problem is negligible.

D. Access Policy

We employ a set of attributes to represent the identity of users. To enforce access control, our access policy incorporates logical AND-gates with multi-valued attributes, and introduces the wildcards to enhance the flexibility of our access policy.

***Definition** 4 (Access Policy):* Let U denotes the attribute universe, in our formal definition, we define 2^U as the set that encompasses all nonempty subsets of U.

- Every nonempty subset in 2^U corresponds to an authentication access policy.
- If for any $X \subset 2^U$, $X \subseteq Y$ when $Y \in \Omega$, has $X \in \Omega$. namely, $X \subset \Omega \leftrightarrow X \subseteq Y$ when $Y \in \Omega$. $\Omega \subset 2^U$ is seen as a monotone access policy. In this paper, the access policies we use are all monotone access policies.
- For $X \subset U$, if $X \in \Omega$, it means that attribute set X satisfies the access policy Ω, can be expressed as $\Omega(X) = true$. If $\Omega(X) = false$, means X does not satisfy Ω. Therefore, $\Omega(X) = true \Longleftrightarrow X \subseteq Y$ for some $Y \in \Omega$.

In our approach, we adopt the disjunctive normal form (DNF) to express authenticated access policies. If $\Omega := \{Y_1, Y_2, \cdots, Y_m\}$, one can equivalently represent the access policy as $\Omega := \{Y_1 \vee Y_2 \vee \cdots \vee Y_m\}$. In this case, for $k \in \{1, 2, \cdots, m\}, \Omega(X) = true \Longleftrightarrow X \subseteq Y_k$. By employing this representation, we effectively capture the conditions necessary for granting access based on attribute containment.

III. System Architecture, Adversary Model, and Evaluation Criteria

This section aims to provide a systematic and comprehensive evaluation by elaborating on the system architecture, defining a practical adversary model, and presenting well-defined evaluation criteria. Based on principles encompassing both usability and security, Wang-Wang [27] proposed an evaluation framework, which has found extensive adoption in various authentication schemes (e.g., [26], [56]). Here we inherit the security criteria of [27] to analyze the proposed attribute-based password authentication key exchange scheme.

A. System Architecture

In this part, we present an attribute-based password authenticated key exchange system architecture. The system involves two parties and a trusted authority (TA). The TA assumes the responsibility of generating the user's private key, which is specifically associated with the user's attributes. Our authentication protocol includes five algorithms as following.

(1) Setup $(\kappa, \Omega) \rightarrow (PP, MK)$
Taking the security parameter κ and the attribute universe Ω as input, this algorithm generates the system public key PP and master key MK.

(2) Key Generation $(MK, PP, S_U) \rightarrow SK_U$
The key generation algorithm inputs the system key (PP, MK), party U's attributes S_U. It generates a static attribute secret key SK_U associated with S_U.

(3) Authentication $(S_A, S_B, \Gamma_A, \Gamma_B, PW) \rightarrow SEK$
The execution of this algorithm occurs between two users. They will compute the same session key SEK when both parties have the correct low-entropy password PW negotiated in advance, and $S_A \in \Gamma_B$, $S_B \in \Gamma_A$, where S_A and S_B denote the user attributes and Γ_A and Γ_B denote the authentication access policy, i.e., $PW_A = PW_B, \Gamma_B(S_A) = true$ and $\Gamma_A(S_B) = true$.

(4) Password change $(PW \rightarrow PW^*)$
If $Alice$ and Bob want to change their password PW, a new password PW^* can be updated by them. Then $Alice$ and Bob can update their shared password to (Bob, PW^*) and $(Alice, PW^*)$ respectively.

(5) Attribute update $(SK_U \rightarrow SK_U{}^*)$
When a user's role changes (e.g., a teacher's title changed from associate professor to full professor), TA should issue an updated attribute private key to the user, thereby modifying the user's access to the newly encrypted data. Simultaneously, it is crucial to ensure that the user is unable to reuse her outdated and obsolete secret key SK_U to access the authentication ciphertexts.

During the key generation phase, a user provides the TA with a set of attributes (e.g., age, gender, and interests), and TA issues the attribute secret key SK_U (i.e., authentication credentials) to the user, which will be used later for the authentication. Our protocol ensures robust security by requiring participants to possess *not only a correct password but also the appropriate set of attributes that satisfy the peer's policy for successful authentication*. In the attribute update phase, when a user's attributes change, her authenittication rights for new key exchange should be modified accordingly.

Remark: In certain situations, a participant may desire to maintain communication with an individual who possesses the desired attributes. For example, in a social network, each participant is characterized by their attributes (such as age, address and hobbies), they can establish a secure channel with anyone who meets their specified policy using an attribute-based authentication scheme. After discovering common topics, it is natural for the participants to establish a session key for further interaction. In our scheme, because they are pre-shared the same password, which enables them to authenticate each other instead of relying solely on policy satisfaction. This highlights the applicability of our AB-PAKE protocol in such scenarios. Consequently, our work focuses on addressing the challenge of *how to precisely and consistently establish a session key between two participants*.

B. Adversary Model

In our scheme, we summarize the potential capabilities possessed by the adversary $\mathcal{A}$, guided by the adversary model presented in [27], [28]. The specific description is as follows:

(1) The adversary $\mathcal{A}$ possesses complete control over the public channel, allowing them not only to passively eavesdrop but also to manipulate protocol messages by altering, injecting, rescheduling, or deleting them.
(2) The adversary $\mathcal{A}$ has the ability to offline enumerate all possible username-password pairs within the password and identity space. And during the protocol security evaluations, $\mathcal{A}$ can also acquire the user's identity.
(3) The adversary $\mathcal{A}$ has the capability to obtain the user's attributes or password, but not both.
(4) The adversary $\mathcal{A}$ has the capability to acquire previously established session keys.
(5) The adversary $\mathcal{A}$ can be a legitimate user.
(6) The adversary $\mathcal{A}$ has the capability to get the user's long-term key or ephemeral key, but not both.

C. Evaluation Criteria

We now introduce a comprehensive evaluation system for our Attribute-Based Password Authenticated Key Exchange (AB-PAKE) protocol. To ensure a thorough assessment, it is imperative to fulfill the following properties:

(1) **Security**. The proposed protocol should provide security for two authentication factors. Additionally, it should demonstrate resilience against various types of known attacks (see [27]). Such as online and offline guessing attacks, and ephemeral secret leakage attacks.
(2) **Efficiency**. To achieve optimal efficiency, it is crucial to minimize the communication and computation burden on users, ensuring it remains as minimal as possible. The user should not be heavily burdened with computational tasks. By this, we can enhance the usability and user experience of the system.
(3) **Functionality**. To facilitate successful authentication for mobile users, it is deemed sufficient for them to possess their passwords and corresponding attributes. This authentication process does not necessitate any additional investments or resources, and promotes convenience and accessibility for mobile users.

Fortunately, Wang-Wang [27] have provided a systematic framework for evaluating two-factor authentication schemes, which contains 12 evaluation criteria. We make some corresponding adaptation to this framework and use it as the evaluation criterion for the proposal.

C1. **Password friendly**: Users have the flexibility to freely choose their passwords and make local changes.
C2. **No password exposure**: The password remains undisclosed to the privileged administrator.
C3. **Resistance to known attacks**: The scheme demonstrates robustness against various types of known attacks, encompassing both fundamental and sophisticated techniques, including password guessing attacks, impersonation attacks and so on.
C4. **Resistance to key compromise impersonation**: Assuming that entities $Alice$ and Bob are two participants in the protocol, when $Alice$'s long-term private key is obtained by the adversary, who can obviously impersonate $Alice$ to communicate with Bob. However, if the protocol is resistant to such attack, then this key-leakage would not grant the adversary the ability to subsequently impersonate Bob to $Alice$ in turn.
C5. **Provision of key agreement**: It enables the involved parties to establish a shared session key, ensuring secure communication throughout the authentication, provided that the attributes of the user comply with the peer's authentication access policies.
C6. **Resistance to ephemeral secret leakage attack**: If the session's ephemeral secret information is leaked or the pseudo-random generator is broken, the adversary can gain the session's ephemeral key, in this case, our protocol can still keep the session key secure.
C7. **Mutual authentication**: The parties have the ability to authenticate each other's legitimacy.
C8. **User anonymity and untraceability**: This scheme enables user identity to remain anonymous and prevents the tracking of user activities.
C9. **Flexible authentication**: The parties possess the flexibility to define their own access policies, facilitating

convenient communication with other entities through the utilization of their attributes.

C10. **Collusion resistance**: Clients with different attributes are unable to generate the same authentication key, this crucial design feature ensures the prevention of any collaborative attempts to compromise the system.

C11. **Robustness**: The authentication must maintain its security as long as one factor remains uncorrupted. This fundamental security requirement is crucial for ensuring the effectiveness of two-factor authentication.

C12. **Forward secrecy**: In the event of the user's long-term key being compromised, the adversary is unable to acquire the previous session key.

IV. Formal Security Model

Formal security definitions are important even from a practical standpoint. They precisely describe what level of security can be achieved and serve as a basis for comparison between different solutions. Finding the "right" security definitions is challenging. They should be strong enough to cover all real world attacks [19]. The security of ours is built upon eCK2007 [51] model, which can provide a strong and rigorous definition of security for two-party authenticated key exchange protocols.

Participants. In our AB-PAKE protocol, each participant is assigned a unique identifier that corresponds to a attribute set. Define U to represent a participant in the authentication system, who has attribute sets S_U and corresponding attribute private key SK_U. Furthermore, each protocol participant defines an individualized authentication policy Γ_U. This policy governs the criteria and conditions for authentication, encapsulating the specific requirements and constraints that must be satisfied by participants during the authentication process.

Private data. Party U possesses a secure device that effectively stores the private key SK_U with a high level of entropy. The possession of this device confirms the participant's ownership of the corresponding attributes, since the Trusted Authority (TA) verifies the eligibility of each specific attribute. Furthermore, a unique password PW is chosen from a predefined "dictionary" D of small size $|D|$, which follows a Zipf distribution [57] and is shared among the two parties.

Protocol execution. A range of queries are utilized to facilitate interactions between the adversary $\mathcal{A}$ and the party U, effectively capturing $\mathcal{A}$'s real-world capabilities. Throughout the process, $\mathcal{A}$ has the ability to activate multiple instances of a participant. We use the notation S_U^i to denote the i-th instance of the protocol, characterized by the attribute set S_U. Additionally, the listed oracles, which can be invoked by $\mathcal{A}$.

(1) $\mathsf{Send}(message)$: In this query, $\mathcal{A}$ possesses the capability to transmit a message in the form of $(\mathcal{I}, S_A, S_B, m_1, \cdots, m_k)/(\mathcal{R}, S_B, S_A, m_1, \ldots, m_{k+1})$, subsequently acquiring the response from the participant involved. This query simulates the $\mathcal{A}$'s control over the communication network.

(2) $\mathsf{SessionKeyReveal}(sid)$: If the session sid is considered complete and already establishes a session key, $\mathcal{A}$ will be granted access to the corresponding session key. This authorization allows $\mathcal{A}$ to retrieve and utilize the session key in accordance with the established protocol.

(3) $\mathsf{EphemeralKeyReveal}(sid)$: $\mathcal{A}$ gains access to the corresponding ephemeral key for the session sid.

(4) $\mathsf{StaticKeyReveal}(S_U)$: $\mathcal{A}$ gains knowledge of the static secret key linked to the attribute set S_U. This signifies that the party U has been compromised.

(5) $\mathsf{MasterKeyReveal}$: In this query, $\mathcal{A}$ obtains the system's master secret key MK.

(6) $\mathsf{Corrupt}(S_U, a)$: By utilizing this query, $\mathcal{A}$ gains the ability to register the attribute set S_U on behalf of the party U, thus obtaining complete control over U. It simulates the capability of $\mathcal{A}$ to corrupt participants: For $a = 1$, this query provides the individual password PW_U of U, for $a = 2$, provides U's private key SK_U that corresponds to the attribute set S_U.

(7) $\mathsf{Test}(sid^*)$: In Test query, it is essential that the session sid^* remains fresh. During this phase, choose a bit b randomly from the set $\{0, 1\}$. If $b = 0$, the session key is provided to $\mathcal{A}$; otherwise, $\mathcal{A}$ receives a randomly selected value chosen uniformly. The adversary is permitted to make only one query of this nature. However, if the test session remains fresh, the adversary may continue asking the oracles in a manner similar to the first phase. Finally, $\mathcal{A}$ provides her guess b' in the test session. The game is deemed won by $\mathcal{A}$ if the chosen test session is fresh and she correctly guesses the challenge (i.e., $b' = b$).

Matching session. Let Π be a protocol. Consider two completed sessions among $Alice$ with attribute set S_A and Bob with S_B. Let $sid = (\mathcal{I}, S_A, S_B, m_1, \ldots, m_n), sid' = (\mathcal{R}, S_B, S_A, m_1, \ldots, m_n)$ represent these sessions respectively. They are referred to as matching sessions if the sessions sid and sid' satisfy the conditions: Bob's attributes satisfy $Alice$'s authentication policy (i.e., $S_B \in \Gamma_A$). Similarly, $Alice$'s attributes satisfy Bob's authentication policy (i.e., $S_A \in \Gamma_B$).

Freshness. A session $sid = (\mathcal{I}, S_A, S_B, m_1, \ldots, m_n)$ is considered fresh when $\mathcal{A}$ is prohibited from making any of the inquiries on sid or its corresponding session $sid' = (\mathcal{R}, S_B, S_A, m_1, \ldots, m_n)$.

(1) If session sid' exists, $\mathcal{A}$ initiates $\mathsf{SessionKeyReveal}(sid)$ or $\mathsf{SessionKey\ Reveal}(sid')$ query.

(2) $\mathcal{A}$ initiates a $\mathsf{Corrupt}(S_A, a)$ query or $\mathsf{Corrupt}(S_B, a)$ query, such as: $\mathsf{Corrupt}(S_A, 1)$ or $\mathsf{Corrupt}(S_B, 1)$; $\mathsf{Corrupt}(S_A, 2)$ or $\mathsf{Corrupt}(S_B, 2)$, where S_A satisfies the policy Γ_B, represented by $S_A \in \Gamma_B$; or S_B satisfies the policy Γ_A, where $S_B \in \Gamma_A$.

(3) Assuming the existence of a matching session sid', the adversary $\mathcal{A}$ proceeds to initiate the following queries: $\mathsf{StaticKeyReveal}(S)$ and $\mathsf{EphemeralKeyReveal}(sid)$ (where $S \in \Gamma_B$); or $\mathsf{StaticKeyReveal}(S)$ and $\mathsf{EphemeralKeyReveal}(sid')$, where $S \in \Gamma_A$.

(4) In the absence of a matching session sid', the adversary $\mathcal{A}$ proceeds to issue the following queries: $\mathsf{StaticKeyReveal}(S), \mathsf{EphemeralKeyReveal}(sid)$, where $S \in \Gamma_B$; or $\mathsf{StaticKeyReveal}(S)$, where $S \in \Gamma_A$.

(5) When $\mathcal{A}$ initiates $\mathsf{MasterKeyReveal}$ query, it is considered $\mathcal{A}$ is issuing $\mathsf{StaticKeyReveal}(S)$ (where $S \in \Gamma_A$) and $\mathsf{StaticKeyReveal}(S)$, where $S \in \Gamma_B$.

In the two-factor authentication key exchange protocol that combines attributes and passwords, the parties can successfully authenticate if their attributes meet the specified policy. Hence, we can interpret the query $Corrupt(S_{A'}, 2)$ as analogous to any queries represented by $Corrupt(S_A, 2)$, given that $S_{A'}$ satisfies the authentication policy defined by $Alice$'s partner.

Semantic Security. Our scheme is considered to achieve semantic security when the advantage $Adv(\mathcal{A})$ is considered negligible, and greater than the parameter $C' \cdot q_s^{s'}$. Here, q_s represents the the overall count of queries queries performed by the adversary $\mathcal{A}$ on the Send oracle. The parameters C' and s' correspond to Zipf's parameters [57]. In the eCK model, if $\mathcal{A}$ successfully wins the game, denoted by $Succ$, then $\mathcal{A}$'s advantage $Adv^{\text{eCK}}_{\text{AB-PAKE}}(\mathcal{A})$ in compromising the semantic security of our scheme can be expressed as:

$$Adv^{\text{eCK}}_{\text{AB-PAKE}}(\mathcal{A}) = \Pr[Succ] - \frac{1}{2}$$

***Definition** 5 (eCK Security):* we define an AB-PAKE scheme as secure in the eCK model when the conditions below are satisfied:

- If two honest users successfully complete matching sessions, have $S_A \in \Gamma_B$, $S_B \in \Gamma_A$, and $PW_A = PW_B$, it can be stated that, except for a negligible probability, both parties are capable of computing a same session key.
- For any PPT adversary $\mathcal{A}$, $Adv^{\text{eCK}}_{\text{AB-PAKE}}$ is negligible.

V. Proposed Protocol

A concrete scheme for attribute-based password authentication is formally presented in this section. It successfully fulfills all 12 criteria outlined in Section III-C. Our AB-PAKE protocol is devised by skillfully integrating the attribute-based signcryption (ABSC) technique [38] with the NAXOS approach [51]. It consists of five phases: setup, user key generation (as shown in Fig. 1), authentication and key exchange (see Fig. 2), password change and attribute update. In practice, we use the numerical set $\{1, 2, \cdots, n\}$ to represent the actual attribute set $\{att_1, ..., att_n\}$. And we represent the lagrange coefficient as $\Delta_{i,n(x)} = \prod_{j \neq i, j \in N} \frac{x-j}{i-j}$, where $i \in Z_p^*$. The notations utilized in the protocol are detailed in Table I.

A. Design Idea

Under the eCK model, an adversary may acquire either a user's attribute secret key participating in the test session or that session's ephemeral secret key, but never both simultaneously. To counteract this, the NAXOS method [51] is employed, intertwining the static secret key with the ephemeral one through a hash function. Given that the adversary lacks knowledge of at least one of these secrets, the resultant value appears completely random. To elaborate, we transform the ephemeral secret keys, denoted as $\tilde{\alpha}_A$ for $Alice$ and $\tilde{\alpha}_B$ for Bob, into pseudo-ephemeral values α_A and α_B. This is achieved by utilizing the hash function $\alpha_A = H_3(\tilde{\alpha}_A || SK_{S_A})$.

In the KeyExchange process, $Alice$ and Bob each selects a distinct access policy (Γ_A for $Alice$, Γ_B for Bob) and a set of signing attributes (W_A for $Alice$, W_B for Bob). They then engage the attribute-based signcryption algorithm, which generates ephemeral public keys EPK_A for $Alice$ and EPK_B for Bob. If Bob's attributes meet the requirements of policy Γ_A in EPK_A (or $Alice$'s attributes meet policy Γ_B in EPK_B), and if the signatures σ_A and σ_B are confirmed as authentic, then each party can recover the secret key (Y^{α_A} for Bob and Y^{α_B} for $Alice$) using the decryption algorithm of the attribute-based signcryption (ABSC). This ABSC system is characterized by the public verifiability of the ciphertext, ensuring the integrity of the exchanged data between $Alice$ and Bob. The sender's legitimacy is verified through their respective signatures, σ_A and σ_B. Within our AB-PAKE framework, we define $\gamma_1 = Y^{\alpha_A}$ and $\gamma_2 = Y^{\alpha_B}$, with α_A and α_B being secret keys generated by $Alice$ and Bob. Nevertheless, γ_1 and γ_2 alone are insufficient for attaining the desired level of security under the eCK model.

The eCK framework also permits an adversary to compromise the system's master secret key. Establishing security under these conditions is unfeasible, as the simulator is incapable of incorporating the BDHE instance into the master secret key, or extracting necessary details to resolve the Gmq-BDHE challenge solely based on Y^{α_A} and Y^{α_B}. To effectively simulate this scenario, we enhance the session key by incorporating element $\gamma_3 = g^{\alpha_A \alpha_B}$.

TABLE I: Some notations in our AB-PAKE protocol

Symbol	Description
TA	Trusted authority
$\mathcal{A}$	Malicious adversary
PP, MK	The system public key and master key
CT	Ciphertext
PW_A, PW_B	Password of $Alice$ and Bob
PW^*	The updated password
d	Attribute threshold
SK_A, SK_B	The secret key of $Alice$ and Bob
Γ_U,	The authentication access policy of user U
S_A, S_B	The attribute set of $Alice$ and Bob
$f(\cdot)$	A $d-1$ degree polynomial
H, H_1, H_2, H_3, H_4	Collision-resistant one-way hash function
$Att_i^{(s)}, Att_i^{(e)}$	Used to sign and encrypt attributes i
SEK	The shared session key
$\|\|$	The string concatenation operation

B. Setup Phase

An attribute universe Ω and the security parameter κ are the inputs for this algorithm. Assuming the protocol contains n attributes, which is represented as $\Omega = \{Att_1, Att_2, \cdots, Att_n\}$, and d denotes the attribute threshold. The public key PP of our AB-PAKE scheme consists of a collection of public parameters, which are generated by a trusted party (TA) using the $Setup(1^\lambda)$ algorithm. The master key MK associated with PP is kept secret to all users. The public key PP and master key MK are created by TA as follows:

S1. Selects bilinear pairing parameters $\Phi := [p, G, G_T, e]$ and $G = \langle g \rangle$, where g is a generator of the group G.

S2. Picks hash functions $H_1 : \{0,1\}^* \to \{0,1\}^l, H_2 : \{0,1\}^* \to Z_p^*, H_3 : \{0,1\}^* \to Z_p^*, H_4 : \{0,1\}^* \to \{0,1\}^\kappa$.

S3. Samples random numbers $s, b \xleftarrow{R} Z_p^*$, sets $g_1 := g^s, g_2 := g^b$ and $Y := e(g_1, g_2)$.

S4. Chooses random numbers $r_i, z_i \xleftarrow{R} Z_p^*$ and sets $Att_i^{(e)} := g^{r_i}$, $Att_i^{(s)} := g^{z_i}$ for each $i \in \Omega$.

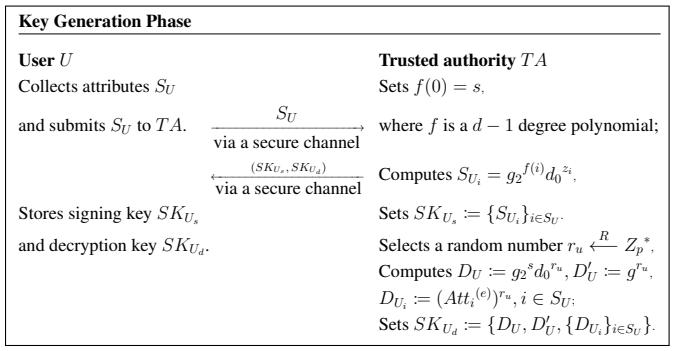

Fig. 1: The user key generation phase of our AB-PAKE protocol. This phase is equivalent to user registration. The users submit attributes to the trusted authority, TA generates the attribute key $SK_U = \{SK_{U_s}, SK_{U_d}\}$ for users, which is used for authentication and key exchange in Section V-D.

S5. Selects $M, d_0, u_0, u_1, \cdots, u_l \xleftarrow{R} G$.

S6. Defines a function $T : \{0,1\}^l \rightarrow G$ where $T(x) := u_0 \prod_{i=1}^{l} u_i^{x_i}$, $x := (x_1, \cdots, x_l) \in \{0,1\}^l$.

Therefore, the system public key can be set as $PP = \{\Phi, M, g, g_2, Y, \{Att_i^{(e)}, Att_i^{(s)}\}_{i \in \Omega}, d_0, u_0, \{u_i\}_{i=1}^{l}, H_1, H_2, T\}$, and master key $MK = \{g_1, b, \{r_i, z_i\}_{i \in \Omega}\}$.

C. User Key Generation Phase

In our scheme, each client U holds a set of attributes S_U and must register with TA to obtain the private key SK_U. Client U collects her attribute set S_U and securely transmits it to TA via a reliable channel. After receiving S_U, TA verifies whether U indeed possesses the valid attributes or not. If U does not, the TA rejects the request. Otherwise, the TA generates SK_U by associating S_U with its master secret key MK. Subsequently, SK_U is stored on a personal device (e.g., a smart-phone, a USB token, or a personal computer from which U will log in later on), and then sends it to the client U as Fig. 1.

R1. Let f be a random polynomial of degree $d-1$ such that $f(0) = s$. Computes $S_{U_i} := g_2^{f(i)} d_0^{z_i}$ for each $i \in S_U$ and the signing key $SK_{U_s} := \{S_{U_i}\}_{i \in S_U}$. It should be noted that in our scheme, each attribute i is considered as an element in Z_p^*.

R2. TA picks $r_u \xleftarrow{R} Z_U^*$ and sets $D_U := d_0^{r_u}, D'_U := g^{r_u}, D_{U_i} := g_2^s (Att_i^{(e)})^{r_u}, i \in S_U$ and the decryption key $SK_{U_d} := \{D_U, D'_U, D_{U_i}\}_{i \in S_U}$.

R3. TA sets the private key $SK_U := \{SK_{U_s}, SK_{U_d}\} = \{S_{U_i}, D_U, D'_U, D_{U_i}\}_{i \in S_U}$.

D. Authentication and Key Exchange Phase

For the purpose of illustrating our method effectively, we can consider the example of $Alice$ and Bob, who combine their static and ephemeral secret keys by performing the computations $\alpha_A := H_3(\tilde{\alpha}_A || SK_A)$ and $\alpha_B := H_3(\tilde{\alpha}_B || SK_B)$. In this way, it becomes challenging to calculate α_A and α_B without knowledge of the corresponding values $\tilde{\alpha}_A || SK_A$ and $\tilde{\alpha}_B || SK_B$. This combination plays a vital role in ensuring implicit entity authentication. That is, both parties use the hash value of their long-term private key and ephemeral private key combination to generate ephemeral public key. In eCK model, the attacker can query the long-term or ephemeral private key, but not both. Hence, the protocol can achieve eCK security. Fig. 2 shows an authentication between $Alice$ and Bob.

Suppose $Alice$ and Bob share the exact same password, denoted as PW. First, $Alice$ defines an authentication access policy Γ_A hoping that Bob's attributes S_B satisfy Γ_A. Similarly, Bob also defines an authentication access policy Γ_B and hopes that $Alice$'s attributes S_A satisfies Γ_B. After successfully completing the protocol, both $Alice$ and Bob acquire the shared session key SEK. Details are as follows.

$Alice$ performs the following steps:

A1. Enters public key PP, an authentication access policy Γ_A, a signing attribute set $W_A \in S_A$ with $|W_A| = d$, the private key SK_A, a shared password PW_A, and a message $msg_A \in M$, the message space $M := \{0,1\}^*$.

A2. Selects $\tilde{\alpha}_A \xleftarrow{R} Z_p^*$ as her ephemeral key, then generates the ephemeral public key $\alpha_A := H_3(\tilde{\alpha}_A || SK_A)$.

A3. Computes $C_A := msg_A \cdot Y^{\alpha_A}$, $C_{A_1} := g^{\alpha_A}$. $M^{PW_A}, C_{A_2} := \left(d_0 Att_i^{(e)}\right)^{\alpha_A}$, where $i \in \Gamma_A$.

A4. Computes $\sigma_A := T(x_A)^{\alpha_A} S_{A_i} d_0^{H_2(msg_A)}$, where $x_A = H_1(C_A || \Gamma_A || W_A)$. The signcryption ciphertext is $CT_A := \{\Gamma_A, C_A, C_{A_1}, C_{A_2}, \sigma_A, W_A\}$.

Bob performs the following verification steps:

A5. Inputs public parameters PP, ciphertext CT_A , a verification attribute set $W_B \subset S_B$, private key SK_B, and a shared low entropy password PW_B.

A6. If $|W_B \cap W_A| \geq d$ exists, and both communication parties have the same correct password PW, i.e., $PW_A = PW_B$. The following verification process is performed. Otherwise, outputs $\perp$.

A7. Selects subset $S \subset (W_B \cap W_A)$, where S contains d attributes (d is the attribute threshold), computes

$$\frac{e(C_{A_1}/M^{PW_B}, \prod_{i \in S} D_B \cdot D_{B_i}^{\Delta_{i,S(0)}})}{e(D'_B, C_{A_2})} = Y^{\alpha_A} \quad (1)$$

$$\frac{C_A}{Y^{\alpha_A}} = msg_A \quad (2)$$

A8. Computes $x'_A = H_1(CT_A || \Gamma_A || W_A)$ and verifies

$$\frac{e(g, \sigma_A) e(d_0^{-1}, \prod_{i \in W_A} Att_i^{(s)\Delta_{i,W_A(0)}} g^{H_2(msg_A)})}{e(T(x'_A), C_{A_1}/M^{PW_B})} \stackrel{?}{=} Y \quad (3)$$

A9. When the above equation (3) holds, it means that $S_B \in \Gamma_A$, if *Bob* can verify that the signcryption message originates from *Alice*, proceed with accepting the session key exchange. Otherwise, outputs $\perp$.

A10. $Alice$ and Bob calculate $\gamma_1 = Y^{\alpha_A}, \gamma_2 = Y^{\alpha_B}, \gamma_3 = g^{\alpha_A \alpha_B}$ respectively. Where α_A and α_B are the ephemeral keys of $Alice$ and Bob respectively.

A11. Computes the same session key $SEK = H_4(\gamma_1 || \gamma_2 || \gamma_3 || PW || EPK_A || EPK_B)$.

Our protocol satisfies the property of *role symmetric*, i.e., both communication parties execute the same operations. Subsequently, $Alice$ and Bob establish a mutual agreement on the shared session key SEK, ensuring the security of their subsequent data communications.

Authentication and Key Exchange Phase

Alice	**Bob**
Access policy: Γ_A, password PW_A	Access policy: Γ_B, password PW_B
Signing attribute $W_A \subseteq S_A$ with $\|W_A\| = d$	Signing attribute $W_B \subseteq S_B$ with $\|W_B\| = d$
1. Samples $\tilde{\alpha}_A \xleftarrow{R} Z_p{}^*$ and calculates $\alpha_A \coloneqq H_3(\tilde{\alpha}_A \| SK_{S_A})$	**1.** Samples $\tilde{\alpha}_B \xleftarrow{R} Z_p{}^*$ and calculates $\alpha_B \coloneqq H_3(\tilde{\alpha}_B \| SK_{S_B})$
2. Computes: $C_A \coloneqq msg_A \cdot Y^{\alpha_A}$ $C_{A_1} \coloneqq g^{\alpha_A} M^{PW_A}$ $C_{A_2} \coloneqq \left(d_0 Att_i^{(e)}\right)^{\alpha_A}, i \in \Gamma_A$ $\sigma_A \coloneqq T(x_A)^{\alpha_A} S_{A_i} d_0^{H_2(msg_A)}$ where $x_A = H_1(C_A \| \Gamma_A \| W_A)$	**2.** Computes $C_B \coloneqq msg_B \cdot Y^{\alpha_B}$ $C_{B_1} \coloneqq g^{\alpha_B} M^{PW_B}$ $C_{B_2} \coloneqq \left(d_0 Att_i^{(e)}\right)^{\alpha_B}, i \in \Gamma_B$ $\sigma_B \coloneqq T(x_B)^{\alpha_B} S_{B_i} d_0^{H_2(msg_B)}$ where $x_B = H_1(C_B \| \Gamma_B \| W_B)$
3. Lets $CT_A \coloneqq \{C_A, C_{A_1}, C_{A_2}\}$ Erases α_A	**3.** Lets $CT_B \coloneqq \{C_B, C_{B_1}, C_{B_2}\}$ Erases α_B
$\xrightarrow{EPK_A \coloneqq \{\Gamma_A, CT_A, \sigma_A\}}$	
$\xleftarrow{EPK_B \coloneqq \{\Gamma_B, CT_B, \sigma_B\}}$	
4. Inputs PP, EPK_B, private key SK_A, and password PW_A	**4.** Inputs PP, EPK_A, private key SK_B, and password PW_B
5. If $\|W_B \cap W_A\| \geq d$ exists, lets $S \subset (W_B \cap W_A)$	**5.** If $\|W_B \cap W_A\| \geq d$ exists, lets $S \subset (W_B \cap W_A)$
6. Computes : (a) $\dfrac{e(C_{B_1}/M^{PW_A}, \prod_{i \in S} D_A \cdot D_{A_i}^{\Delta_{i,S(0)}})}{e(D'_A, C_{B_2})} = Y^{\alpha_B}$ (b) $\dfrac{C_B}{Y^{\alpha_B}} = msg_B$	**6.** Computes : (a) $\dfrac{e(C_{A_1}/M^{PW_B}, \prod_{i \in S} D_B \cdot D_{B_i}^{\Delta_{i,S(0)}})}{e(D'_B, C_{A_2})} = Y^{\alpha_A}$ (b) $\dfrac{C_A}{Y^{\alpha_A}} = msg_A$
7. Computes $x'_B = H_1(CT_B \| \Gamma_B \| W_B)$ and checks (c) $\dfrac{e(g, \sigma_B) e(d_0^{-1}, \prod_{i \in W_B} Att_i^{(s)\Delta_{i,W_B(0)}} \cdot g^{H_2(msg_B)})}{e(T(x'_B), C_{B_1}/M^{PW_A})} \stackrel{?}{=} Y$	**7.** Computes $x'_A = H_1(CT_A \| \Gamma_A \| W_A)$ and checks (c) $\dfrac{e(g, \sigma_A) e(d_0^{-1}, \prod_{i \in W_A} Att_i^{(s)\Delta_{i,W_A(0)}} \cdot g^{H_2(msg_A)})}{e(T(x'_A), C_{A_1}/M^{PW_B})} \stackrel{?}{=} Y$
8. If $\neg(c)$, outputs $\perp$	**8.** If $\neg(c)$, outputs $\perp$
9. Otherwise, calculates: $\gamma_1 \coloneqq Y^{H_3(\tilde{\alpha}_A \| SK_{S_A})} = Y^{\alpha_A}$ $\gamma_2 \coloneqq Y^{\alpha_B}$ $\gamma_3 \coloneqq \left(\dfrac{C_{B_1}}{M^{PW_A}}\right)^{H_3(\tilde{\alpha}_A \| SK_{S_A})} = g^{\alpha_A \alpha_B}$	**9.** Otherwise, calculates: $\gamma_1 \coloneqq Y^{\alpha_A}$ $\gamma_2 \coloneqq Y^{H_3(\tilde{\alpha}_B \| SK_{S_B})} = Y^{\alpha_B}$ $\gamma_3 \coloneqq \left(\dfrac{C_{A_1}}{M^{PW_B}}\right)^{H_3(\tilde{\alpha}_B \| SK_{S_B})} = g^{\alpha_A \alpha_B}$
10. Computes $SEK_A \coloneqq H_4(\gamma_1 \| \gamma_2 \| \gamma_3 \| PW_A \| EPK_A \| EPK_B)$	**10.** Computes $SEK_B \coloneqq H_4(\gamma_1 \| \gamma_2 \| \gamma_3 \| PW_B \| EPK_A \| EPK_B)$

Fig. 2: Authentication and key exchange phase of our AB-PAKE protocol.

E. Password Change Phase

To ensure user friendliness if clients want to update their password PW, they can change the current password PW to a new shared password PW^* by following steps.

- **P1.** $Alice$ provides her attributes W_A to Bob, negotiates with Bob to change the shared password PW.
- **P2.** Bob submits the ciphertext CT_B to $Alice$ and requires $Alice$ to recover the message msg_B.
- **P3.** $Alice$ inputs her attribute key SK_A, computes $e(C_{B_1}/M^{PW_A}, \prod_{i \in W_A} D_A \cdot D_{A_i}^{\Delta_{i,W_A(0)}})/e(D'_A, C_{B_2}) = Y^{\alpha_B}$, $msg'_B = \frac{C_{B_1}}{Y^{\alpha_B}}$, returns the message msg'_B to Bob.
- **P4.** Bob checks whether $msg'_B = msg_B$ is true. If the verification holds, this implies that $Alice$ is authorized user. Otherwise, Bob rejects the password update request.
- **P5.** Then $Alice$ inputs a new password PW^* and transmits PW^* to Bob over the reliable channel.
- **P6.** $Alice$ and Bob update their storage to (Bob, PW^*) and $(Alice, PW^*)$ correspondingly.

F. Attribute Update Phase

When a user's role changes, she can update her attributes to satisy different authentication requirements. The trusted authority should be required to issue an updated attribute private key to the client, thereby modifying the client's access.

- **U1.** The user U sends her original registered attributes set S_U and the new attributes sets S'_U to TA.
- **U2.** The TA enters the system master key $MK = \{g_1 = g^s, b, \{r_i, z_i\}_{i \in \Omega}\}$. If $S_U, S'_U \subset \Omega$, then randomly selects $r_{i'}, z_{i'}$ and calculates $rk_{i1} = r_{i'}/r_i, rk_{i2} = z_{i'}/z_i$, stores the re-encryption key $rk_i = \{rk_{i1}, rk_{i2}\}_{i \in S_U, i' \in S'_U}$.
- **U3.** For each update attribute $i' \in S'_U$, TA calculates the updated private key SK'_U and return it to the client who needs to update attributes.
- **U4.** TA sets $Att_{i'}^{(e)} = (Att_i^{(e)})^{rk_{i1}} = g^{r_{i'}}, Att_{i'}^{(s)} = (Att_i^{(s)})^{rk_{i2}} = g^{z_{i'}}$, computes $D_{U_{i'}} = (h_{i'}^{(e)})^{r_u}$, $S_{U_{i'}} = g_2^{f(i')} d_0^{z_{i'}}$ for each $i' \in S'_U$.

U5. Then TA sets the updated private key $SK'_U = \{D_U, D'_U, D_{U_{i'}}, S_{U_{i'}}\}_{i' \in S'_U}$.

G. Correctness Analysis

The correctness of our protocol means that if the protocol is executed honestly according to the above process, both communication parties will get the same session key, i.e., $SEK_A = SEK_B$. We assume that both communication entities satisfy the peer's authentication policy and have the same shared low-entropy password, i.e., $S_A \in \Gamma_B$, $S_B \in \Gamma_A$ and $PW_A = PW_B$. Let W_A and W_B be the signcryption attributes of $Alice$ and Bob respectively. When $|W_B \cap W_A| \geq d$, selects a subset $S \subset (W_B \cap W_A)$, where S contains d attributes (d is the attribute threshold). We can verify the correctness of our proposed scheme by the following equations. For equation (1), (2), (3), the user performs the following calculation to authenticate the other party.

$$\begin{aligned}
&\frac{e(C_{A_1}/M^{PW_B}, \prod_{i\in S} D_B \cdot D_{B_i}^{\Delta_{i,S(0)}})}{e(D'_B, C_{A_2})} \\
=&\prod_{i\in S}\left(\frac{e(g^{\alpha_A}, g_2^s d_0^{r_B} \cdot (Att_i^{(e)})^{r_B})}{e(g^{r_B}, (d_0 Att_i^{(e)})^{\alpha_A})}\right)^{\Delta_{i,S(0)}} \\
=&\prod_{i\in S}\left(\frac{e(g^{\alpha_A}, g_2^s) e(g^{\alpha_A}, (d_0 Att_i^{(e)})^{r_B})}{e(g^{\alpha_A}, (d_0 Att_i^{(e)})^{r_B})}\right)^{\Delta_{i,S(0)}} \\
=&e(g, g_2^s)^{\alpha_A} = Y^{\alpha_A}
\end{aligned}$$

Then recovers the correct message: $\frac{C_A}{Y^{\alpha_A}} = \frac{msg_A \cdot Y^{\alpha_A}}{Y^{\alpha_A}} = msg_A$. If the message has not been modified or forged, computes $x'_A = H_1(CT_A||\Gamma_A||W_A)$, there are:

$$\begin{aligned}
&\frac{e(g,\sigma_A) e(d_0^{-1}, \prod_{i\in W_A} Att_i^{(s)\Delta_{i,W_A(0)}} g^{H_2(msg_A)})}{e(T(x'_A), C_{A_1}/M^{PW_B})} \\
=&\frac{e(g, \prod_{i\in W_A} T(x_A)^{\alpha_A} S_{A_i}^{\Delta_{i,W_A(0)}} d_0^{H_2(msg_A)})}{e(d_0, g^{z_i} g^{H_2(msg_A)}) e(T(x'_A), g^{\alpha_A})} \\
=&\frac{e(g, T(x_A)^{\alpha_A}) e(g, \prod_{i\in W_A} g_2^{f(i)} d_0^{z_i})^{\Delta_{i,W_A(0)}} e(g, d_0^{H_2(msg_A)})}{e(d_0, g^{z_i}) e(d_0, g^{H_2(msg_A)}) e(T(x'_A), g^{\alpha_A})} \\
=&e(g, \prod_{i\in W_A} g_2^{f(i)})^{\Delta_{i,S(0)}} \\
=&e(g, g_2)^s = Y
\end{aligned}$$

$Alice$ calculates the session key as:

$$SEK_A = \begin{cases} \gamma_1 = Y^{H_3(\tilde{\alpha}_A||SK_A)} = Y^{\alpha_A} \\ \gamma_2 = \dfrac{e(C_{B_1}/M^{PW_A}, \prod_{i\in S} D_A D_{A_i}^{\Delta_{i,S(0)}})}{e(D'_A, C_{B_2})} = Y^{\alpha_B} \\ \gamma_3 = (\dfrac{C_{B_1}}{M^{PW_A}}) = g^{\alpha_A \alpha_B} \\ PW_A||EPK_A||EPK_B \end{cases}$$

Similarly, Bob calculates the session key as:

$$SEK_B = \begin{cases} \gamma_1 = \dfrac{e(C_{A_1}/M^{PW_B}, \prod_{i\in S} D_B D_{B_i}^{\Delta_{i,S(0)}})}{e(D'_B, C_{A_2})} = Y^{\alpha_A} \\ \gamma_2 = Y^{H_3(\tilde{\alpha_B}||SK_B)} = Y^{\alpha_B} \\ \gamma_3 = (\dfrac{C_{A_1}}{M^{PW_B}}) = g^{\alpha_A \alpha_B} \\ PW_B||EPK_A||EPK_B \end{cases}$$

Therefore, $Alice$ and Bob independently calculate the same session key, that is, $SEK_A = SEK_B$, it implies that both parties have negotiated a same session key successfully.

H. Applications

Our proposed AB-PAKE scheme has a variety of potential applications. As highlighted in [58], one of the most compelling usage scenarios could be friend-searching based on shared interests in online social networks (OSNs). A user registering on an OSN is required to obtain attribute permissions, verified by the respective OSN administrator. Through the integration of the AB-PAKE protocol within the OSN platform, a secure connection can be established between two users when there is a mutual match in their profile lists. Users can verify if their communication partner is the entity they seek by conducting OSN profile matching in the AB-PAKE protocol. It is worth noting that in certain contexts, such as mobile health networks [39], the sharing of privacy information becomes necessary. Consequently, attribute-based password authentication holds significant potential for extensive applications, including genomic testing [36], [40], privacy-preserving data aggregation [59], and keyword searching [60]. Notably, main cryptographic schemes like Idemix [61] have successfully deployed attribute-based authentication. These techniques are now integrated into lightweight infrastructures [62], [63]. We are of the belief that attribute-based password authentication will find broad applicability in the future.

VI. Formal Security Analysis

The security of our scheme is proved through the widely accepted eCK model [51], which can provide a solid proof of security for two-party authenticated key exchange protocols. In addition, we apply Zipf's law [57] to choose a password for communication participants.

A. AKE Security

***Theorem** 1 (eCK security):* If Gmq-BDHE assumption and CDH assumption are both hold, assuming that H_1 and H_2 are collision-resistant hash functions, and H_3 and H_4 are random oracles, then our AB-PAKE protocol is selectively secure in the eCK model.

Proof. Let $\mathcal{A}$ be the adversary of AB-PAKE protocol and $\mathcal{S}$ be the simulator. $\mathcal{S}$ aims to solve the CDH problem or Gmq-BDHE problem by using $\mathcal{A}$ as a black-box. The idea of proving the theorem is as follows: For the proposed AB-PAKE protocol, if $\mathcal{A}$ has the ability to distinguish the session key of the test session sid^* from a randomly chosen session key in polynomial time, then $\mathcal{S}$ can address either the CDH problem or the Gmq-BDHE problem with a non-negligible probability of winning. Let $Succ$ denote the event that $\mathcal{A}$ wins. Denote the probability of a successful attack by $\mathcal{A}$ on the proposed protocol as $\Pr[Succ]$. The probability of giving a correct guess at the sid^* session key. Let sid be the session identifier of an honest party such that $sid = sid^*$, the test session, and sid is not a matching session to sid^*. In this case, since sid and sid^* are distinct non-matching sessions, the output of the

hash function H_4 is different for $sid = sid^*$. Since, H_4 is assumed to be a random oracle, the adversary cannot learn any information about the session key of the test session. We give the following definition:

- Event $AskH$: $\mathcal{A}$ sends a query to the H_4 oracle for $(\gamma_1 ||\gamma_2||\gamma_3||PW||EPK_A||EPK_B)$ corresponding to sid^*.
- Event $\overline{AskH}$: $\mathcal{A}$ doesn't send query to H_4 oracle for $(\gamma_1 ||\gamma_2||\gamma_3||PW||EPK_A||EPK_B)$ corresponding to sid^*.

Since $\Pr[Succ \wedge \overline{AskH}] \leq 1/2$. Then it holds that:

$$\Pr[Succ] = \Pr[Succ \wedge AskH] + \Pr[Succ \wedge \overline{AskH}]$$

$$\leq \Pr[Succ \wedge AskH] + 1/2.$$

Let $Succ^*$ denotes the event $Succ \wedge AskH$, then we can define the events as follows:

- Event $AskS$: The adversary $\mathcal{A}$ makes queries of $SK_U := \{SK_{U_s}, SK_{U_d}\} = \{S_{U_i}, D_U, D_U^{'}, D_{U_i}\}_{i \in S_U}$ to the H_3 oracle (for party $U \in \{Alice, Bob\}$) before querying StaticKeyReveal or MasterKeyReveal.
- Event $\overline{AskS}$: The complement of $AskS$.

Since the test session sid must be a fresh session, and depending on the $\mathcal{A}$'s attack capability, we define the events:

- Event E_1: If no corresponding matching session $\overline{sid}^*$ for sid^*, $\mathcal{A}$ queries StaticKeyReveal $(S_U), S_U \in \Gamma_B$.
- Event E_2: If no corresponding matching session $\overline{sid}^*$ for sid^*, $\mathcal{A}$ makes queries to EphemeralKeyReveal(sid^*).
- Event E_3: $\mathcal{A}$ makes queries to either MasterKeyReveal or StaticKeyReveal$(S_U), S_U \in \Gamma_B$ and StaticKeyReveal $(S_U), S_U \in \Gamma_A$, while there is a corresponding matching session $\overline{sid}^*$ for sid^*.
- Event E_4: The adversary $\mathcal{A}$ makes queries to EphemeralKeyReveal(sid^*), EphemeralKeyReveal$(\overline{sid}^*)$, while there is a corresponding matching session $\overline{sid}^*$.
- Event E_5: The adversary $\mathcal{A}$ queries StaticKeyReveal (S_U), $S_U \in \Gamma_B$ and EphemeralKeyReveal$(\overline{sid}^*)$, while there is a corresponding matching session $\overline{sid}^*$ for sid^*.
- Event E_6: The adversary $\mathcal{A}$ queries EphemeralKey Reveal(sid^*) and StaticKeyReveal$(S_U), S_U \in \Gamma_A$, while there is a corresponding matching session $\overline{sid}^*$ for sid^*.

We analyze the probability of $\mathcal{A}$ in solving Gmq-BDHE and CDH problems through a series of events $E_1 \sim E_6$. There are N users in the system and at most L sessions are activated for each user. Simulator $\mathcal{S}$ randomly selects two users $Alice$ and Bob, and guesses with the probability of $1/N^2L$ that i-th session is the test session sid^* selected by attacker $\mathcal{A}$.

Assuming that the StaticKeyReveal (S_U) query on the private key has been executed, the occurrence of the event indicates that no password corruption has taken place. This is because there should only be a maximum of one password associated with the value PW that leads to the successful execution of $\mathcal{O}^{mq-BDHE}(C_{A_1}/PW_A, \alpha_1, \alpha_q, \gamma_1) = 1, \mathcal{O}^{mq-BDHE}(C_{B_1}/M_{PW_B}, \alpha_1, \alpha_q, \gamma_2) = 1$ and $e(C_{A_1}/PW_A, C_{B_1}/M_{PW_B}) = e(g, \gamma_3)$. In other words, in each protocol transcript, $\mathcal{A}$ can verify only one password.

Thus, using Zipf's law for passwords $Succ^*, \mathcal{A}$ queries correctly formed $\gamma_1, \gamma_2, \gamma_3$ to H_4. Therefore, $\mathcal{S}$ is successful. Hence, $\mathcal{S}$ is successful with probability Pr[$\mathcal{S}$ solves the Gmq-BDHE problem] $\geq p_1/N^2L + 2C' \cdot q_s^{s'}$, Where C' and s' are the Zipf's parameters, q_s represents the amount of Send queries, p_1 is the probability that the event $E_1 \wedge \overline{AskS} \wedge Succ^*$. The detailed proof can be referred to Appendix A, which is accessible in the supplementary material. To sum up, we can conclude with: if Gmq-BDHE assumption and CDH assumption are hold, the AB-PAKE protocol is selective security under eCK model.

B. Message Confidentiality

***Theorem** 2 (IND-sAP-CCA2 security):* In our protocol, assuming that the access structure is represented by an Linear Secret Sharing Scheme (LSSS) matrix, the maximum number of rows and columns is q. Under the Dmq-BDHE assumption, our AB-PAKE protocol achieves IND-sAP-CCA2 security.

Proof. The formalization of message confidentiality is defined based on the indistinguishability of ciphertexts under selective encryption access policies and adaptive chosen ciphertext attacks (IND-sAP-CCA2). It is formalized through a game $\mathbf{Game}_{\text{AB-PAKE}}^{\text{IND-sAP-CCA2}}$ that plays between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$.

Commit. $\mathcal{A}$ sends an encryption access policy Γ_e^* to $\mathcal{C}$.

Setup Phase. $\mathcal{C}$ gets $(PP, MK) \leftarrow$ Setup(κ, Ω) and sends PP to A.

Ouery Phase 1. $\mathcal{A}$ dynamically generates and submits the subsequent queries in polynomial time.

- DecKey Query: $\mathcal{A}$ gives a decryption attribute set $A_d \subset \Omega$ such that $\Gamma_e^*(A_d) = false$, and obtains $SK_{Ad} \leftarrow$ DecKeyGen (PP, MK, Ad) from $\mathcal{C}$.
- SignKey Query: $\mathcal{A}$ sends a signing attribute set $A_s \subset \Omega$ of size at least n, and receives $SK_{As} \leftarrow$ SignKeyGen (PP, MK, A_s) from $\mathcal{C}$.
- Unsigncrypt Query: $\mathcal{A}$ submits $\{CT, W_v, A_d\}$ as inputs to the challenger $\mathcal{C}$ and receives the result of Unsigncrypt$(PP, CT, W_d, \mathcal{SK}_{Ad})$, where $\mathcal{SK}_{Ad} \leftarrow$ DecKeyGen (PP, MK, A_d).

Challenge. $\mathcal{A}$ generates two messages of equal length, referred to as $msg_0^*, msg_0^* \in \mathcal{M}$, and provides a signing attribute set $W_s^* \subset \Omega$. The challenger $\mathcal{C}$ selectsa random bit $\mu \in [0, 1]$ and sends the challenge ciphertext $CT^* \leftarrow$ Signcrypt $(PP, \Gamma_w^*, SK_A, W_s^*, msg_\mu^*)$ to $\mathcal{A}$, where $W_s^* \subseteq A_s$ with $|W_s^*| = d$ and $SK_{A_s} \leftarrow$ SignKeyGen(PP, MK, A_s).

Ouery Phase 2. $\mathcal{A}$ persistently generates DecKey, SignKey, and Unsigncrypt queries, similar to Query phase 1. However, certain limitations are imposed on $\mathcal{A}$ regarding Unsigncrypt Queries. Specifically, $\mathcal{A}$ is prohibited from issuing an Unsigncrypt Query (PP, MK, A_d) if it results in $\Gamma_e^*(A_d) =$ true.

Guess. $\mathcal{A}$ provides guess $\mu\prime \in [0, 1]$ and achieves victory in the game if $\mu\prime$ is equal to μ.

The advantage of $\mathcal{A}$ wins in this game is defined as $Adv_{\mathcal{A}}^{IND-sAP-CCA2} := |\Pr[\mu\prime = \mu] - 1/2|$.

In the above game, it is not necessary for $\mathcal{A}$ to issue signcryption queries because she can obtain the signing key for any signature attribute set of size n, and hence she can signcrypt on her own.

The aforementioned theorem can be proven by employing a similar approach used to establish the message confidentiality in the CP-ABE scheme [38].

C. Ciphertext Unforgeability

***Theorem** 3 (EUF-sSAS-CMA security):* Given the assumption that the hash functions H_1 and H_2 have collision resistance. Our AB-PAKE protocol achieves EUF-sSAS-CMA security under the CDH assumption.

Proof. Ciphertext unforgeability is formalized by establishing the existential unforgeability under selective signing attribute set and adaptive chosen message attack (EUF-SSAS-CMA). It is formalized through a game $\mathbf{Game}_{\text{AB-PAKE}}^{\text{EUF-sSAS-CMA}}$ that plays between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ captures EUF-sAP-CMA security. Commit. $\mathcal{A}$ submits a signing attribute set $W_s^* \subset \Omega$ of size n to $\mathcal{C}$.

Setup Phase. $\mathcal{C}$ obtains $(PP, MK) \leftarrow \text{Setup}(k, U)$ and gives PP to $\mathcal{A}$.

Ouery Phase. $\mathcal{A}$ adaptively issues the following queries a polynomial number of times.

- DecKey Query: $\mathcal{A}$ gives a decryption attribute set $A_d \subset \Omega$, and obtains $SK_{A_d} \leftarrow \text{DecKeyGen}\ (PP, MK, A_d)$ from $\mathcal{C}$.
- SignKey Query: $\mathcal{A}$ sends a signing attribute set $A_s \subset \Omega$ of size at least n such that $|A_s \cap W_s^*| < d$, and then receives $SK_{A_s} \leftarrow \text{SignKeyGen}\ (PP, MK, A_s)$ from $\mathcal{C}$.
- Signcrypt Query: $\mathcal{A}$ submits $\{msg, \Gamma_e, A_s\}$ to $\mathcal{C}$ (where $|A_s| \geq d$) and gets back a ciphertext $CT \leftarrow \text{Signcrypt}\ (PP, \Gamma_e, SK_{A_s}, W_s, msg)$, where $SK_{A_s} \leftarrow \text{SignKeyGen}\ ((PP, MK, A_s))$ and $W_s \subseteq A_s$, with $|W_s| = d$.

Forgery. $\mathcal{A}$ returns a ciphertext $CT^* := [\Gamma_e^*, \cdots]$ for the signing attribute set W_s^*.

The adversary $\mathcal{A}$ achieves victory in the game if the ciphertext $\mathcal{CT}^* := [\Gamma_e^*, \cdots]$ is both valid and not obtained from a Signcrypt query. That is, $\text{Unsigncrypt}(PP, CT^*, W_s^*, SK_{A_d}) = msg^* \neq \perp$, where $\Gamma(A_d)^* = true$ and $\mathcal{A}$ did not issue a Signcrypt query$[msg^*, \Gamma_e^*, A_s^*]$ where $W_s^* \subseteq A_s^*$.

The advantage of adversary $\mathcal{A}$ wins in this game is defined as "$succ$", where "$succ$" represents the success of $\mathcal{A}$. $Adv_{\mathcal{A}}^{EUF-sSAS-CMA} := \Pr\ [\mathcal{A}\ \text{wins}]$.

In the given game, there is no requirement for $\mathcal{A}$ to submit unsigncryption query since she has the capability to acquire the decryption key for any decryption attribute set, which allowing $\mathcal{A}$ to perform unsigncryption autonomously.

VII. Heuristic Analysis

In this section, we discuss about the security of the scheme through the heuristic analysis.

Ephemeral secret leakage (ESL) attack. In the eCK model, the adversary has the capability to acquire either the user's secret attribute key or the ephemeral secret key, but not both simultaneously. Therefore, in our protocol, by using NAXOS trick [51] to bind together the long-term secret key SK and the ephemeral secret key $\tilde{\alpha}$, we set the ephemeral public key $\alpha = H_3(\tilde{\alpha}||SK)$. Because the adversary can't get both $\tilde{\alpha}$ and SK, she cannot perform the off-line attack. Despite the attacker's ability to obtain either the long-term secret key SK or the ephemeral secret key $\tilde{\alpha}$ and successfully guess the user's correct password, acquiring the value of α remains beyond her reach. So she cannot verify if her guess is correct. As a result, the proposal is resilient against the ESL attack.

Password guessing attack. Due to $C = g^{\alpha} M^{PW}$, $\alpha = H_3(\tilde{\alpha}||SK)$. We now analyze the following two cases. 1) Without the correct attribute information to generate attribute key SK. The attacker can't compute correct α, so password validity can't be verified. 2) With the correct attribute information to generate attribute key SK. Since $\tilde{\alpha}$ is a random number, the attacker's task to generate α is computationally infeasible. The attacker still can't determine whether the password is correct. Therefore our scheme can resist password guessing attack.

Replay attack. This attack refers to the attacker using the intercepted public session information to obtain secret information of a new session. In our AB-PAKE protocol, $Alice$ and Bob need to select new random numbers $\tilde{\alpha}_A \in Z_p^*$ and $\tilde{\alpha}_B \in Z_p^*$ as the ephemeral private keys for each session, and recalculate fresh α_A and α_B respectively. Since the information used to generate the session key varies for each session, replay attacks are effectively avoided.

User anonymity and untraceability. It requires that the attacker remains unaware of the user's real identity, nor can she judge a specific user's session from multiple sessions. In attribute-based systems, the property of anonymity is inherent as individuals possessing different attribute sets can satisfy the same access policy. Hence, the attacker is unable to ascertain the user's real identity, ensuring their anonymity. Moreover, the message EPK involves a random number α, it becomes computationally infeasible to track the activities of a user across multiple sessions and extract valid information. Hence, our proposed scheme effectively achieves the principles of untraceability and anonymity.

Forward security. In our AB-PAKE protocol, the session key is not completely generated by both parties' long-term keys, but also contains the ephemeral private keys of the current session, so the new protocol can achieve forward security: If the session remains uncompromised by an adversary before the leakage of the long-term private key, even the adversary obtains the participant's long-term private key, they remain unable to obtain the previous sessions keys.

Mutual authentication. In our protocol, the user's attributes are embedded in the user's private key SK, and the building block of our protocol is attribute-based signcryption. $Alice$ uses her private key SK_A to decrypt Bob's ciphertext CT_B. If the decryption succeeds, means $Alice$'s attributes S_A and password PW_A satisfy the authentication requirements of Bob (i.e., $S_A \in \Gamma_B$, $PW_A = PW_B$). Then, $Alice$ uses Bob's public key $Att_i^{(s)}$ to verify the signature σ_B. If the verification succeeds, it indicates that Bob is a legitimate user, Bob uses the same method to authenticate $Alice$. Therefore, our scheme can realize mutual authentication.

Krawczyk [47] pointed out that, perfect forward security (PFS) cannot be attained by any two-pass key exchange protocols that rely on public key authentication. Therefore, for two-pass AKE protocols, the concept of weak PFS (wPFS) is often adopted. It asserts that even if static keys are compromised, the session key stays confidential after the session finishes.

Discussion. The recent advancements in quantum computing [64], [65] indicate that the quantum era is arriving, and standards organizations (e.g., IETF, IEEE, and NIST) are preparing solutions in the post-quantum age. These facts highlight the importance and urgency of constructing quantum-secure authentication schemes. There are several cryptographic approaches to resist quantum attacks, such as hash-based, code-based, multivariate polynomial-based, lattice-based, and supersingular isogeny-based cryptosystems [66]. Among them, the lattice-based cryptosystem is considered to be one of the most promising solutions, which can be best illustrated by the overwhelming fraction of lattice-based candidates submitted and selected into the first round (41% = 26/64), second round (46% = 12/26), and the newly third round (71%= 5/7) competition issued by NIST. Besides, NIST regards lattice-based schemes as the most promising general-purpose algorithms for public-key encryption [67]. Dozens of quantum-resistant authentication protocols [68]–[70] have been proposed over lattices. Some encryption schemes [71], [72] also achieve quantum computer attack resistance by exploiting the lattice-based techniques. Since our research focuses on cryptographic mechanisms (i.e., how to achieve flexible user authentication and fine-grained access control), quantum-related countermeasures are beyond the scope of our focus. We will investigate quantum-resistant techniques and their integration into flexible user authentication and fine-grained access control as a potential future work.

Also note that, as with any non-tamper proof device-enhanced PAKE schemes (e.g., [8], [73]) except for those using additional secure channels (e.g., [25]), an inherent limitation is that the security of our AB-PAKE cannot be ensured when used on untrusted terminals (e.g., infected with malware): the attacker can simultaneously compromise the low-entropy password and the high-entropy attribute key of the infected user. In contrast, as revealed in [ISC'13] [74], password authentication using (conditionally) tamper-proof devices (e.g., smartcards and trusted execution environments) can be secure on untrusted terminals.

VIII. Performance Analysis

In the following, we conduct a thorough performance analysis of our protocol in terms of functionality, computation and communication overhead by comparing it with the previous AKE schemes [26], [42]–[44], [52], [53], [55], [75]–[81]. Since we focus on two users secretly and privately authenticate to each other and jointly agree on a key for further communication, and there is no relevant authentication on the server side. The experiments are executed on a personal computer (Intel Core(TM) i7-10700F CPU 2.9 GHz, 8 GB RAM). The pairing-based cryptography (PBC) serves as a proficient resource for generating elliptic curve parameters and performing pairing computations. Based on the PBC cryptographic library, the related algorithms are implemented to facilitate various of cryptographic operations. Where T_M, T_H, T_P and T_E denote scalar multiplication, hash operation, bilinear pairing and modular exponential operation. T_{Enc} denotes the execution of symmetric encryption. For clarity, we describe the running times of the cryptographic primitives in Table II.

TABLE II: Some cryptographic primitive operations

Notation	Description
$T_H \approx 1.436$ ms	One hash operation
$T_{Enc} \approx 1.826$ ms	One block encryption
$T_P \approx 5.064$ ms	One bilinear pairing operation
$T_E \approx 2.132$ ms	One modular exponential operation
$T_M \approx 3.603$ ms	One elliptic curve scalar multiplication

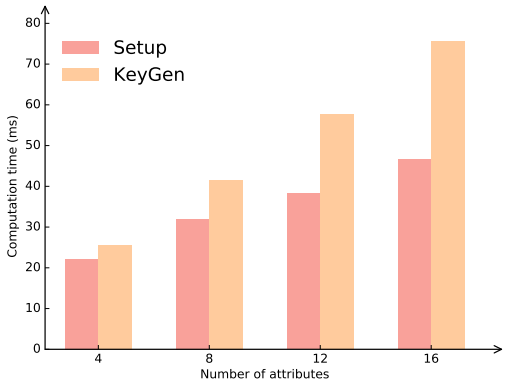

Fig. 3: The computation cost of system setup and key generation algorithms under different number of attributes.

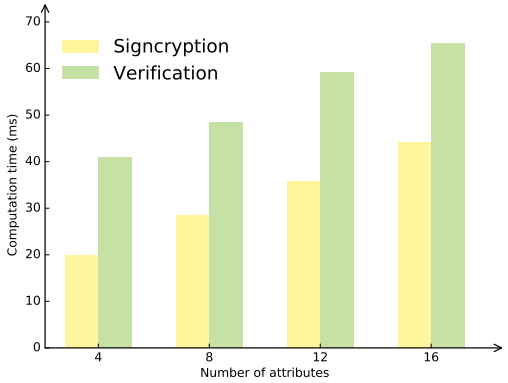

Fig. 4: The computation cost of signcryption and verification algorithms under different number of attributes.

A. Functional Analysis

To understand the effectiveness and practicability of our scheme, we compared this scheme with similar AKE schemes in protocol rounds, authentication factors, security model and 12 evaluation criteria. The comparison result in Table III shows that the properties mentioned are all effectively achieved by our scheme. In addition to the literature in Table III, we also analyze some related AB-AKE schemes [33]–[37] about the number of communication rounds. Among them, [33] has one round, [34] has three rounds, [35] has five rounds, [36] has four rounds, and [37] has one round. Despite [33] and [37] requires only one communication round each, both authentication protocols lack privacy protection for user attributes. This makes it easy for attackers to obtain users' attribute information from the public channel, leading to privacy breaches for authenticated users. A second limitation is that insider security is not considered in the security model.

TABLE III: Functionality Comparison among Relevant Authentication Schemes

Scheme	Year	Ref.	Rounds/Flows‡	Authentication factor	Security model	The proposed twelve evaluation criteria†											
						C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
Li et al.	2019	[52]	3/3	Password+Smart card	BPR	✓	✓	✓	×	✓	×	✓	✓	×	✓	✓	✓
Zhang et al.	2019	[53]	4/4	Password+Biometrics	ROR	×	×	×	×	✓	×	✓	×	✓	✓	✓	✓
Srinivas et al.	2020	[75]	3/3	Password+Biometrics+Smart card	ROR	✓	✓	✓	×	✓	×	✓	×	×	✓	✓	✓
Li et al.	2021	[76]	3/3	Password+Biometrics+Smart card	ROM	✓	✓	✓	×	✓	×	✓	✓	×	✓	✓	✓
Tseng et al.	2021	[77]	3/3	ID-PKC	eCK	–	–	×	✓	✓	✓	✓	×	×	✓	✓	✓
Chakraborty et al.	2021	[37]	1/2	Attribute-based authentication	AB-eCK	–	–	✓	✓	✓	✓	✓	×	✓	×	×	✓
Huang et al.	2022	[35]	5/5	Attribute-based authentication	ROM	–	–	×	✓	✓	×	✓	×	✓	✓	✓	✓
Abbasinezhad et al.	2022	[78]	3/3	ID-PKC	ROM	–	–	×	✓	✓	✓	✓	✓	×	✓	×	✓
Vijayakumar et al.	2022	[55]	3/3	ID-PKC	–	–	–	×	✓	✓	×	✓	×	✓	✓	×	✓
Wang et al.	2023	[26]	3/3	Password+Smart card	BPR	✓	✓	✓	✓	✓	×	✓	✓	×	×	✓	✓
Tan et al.	2023	[42]	6/6	Attribute-based authentication	ROM	–	–	×	✓	✓	×	✓	×	✓	✓	✓	✓
Sucasas et al.	2023	[43]	4/5	Attribute-based authentication	ROM	–	–	×	✓	✓	×	✓	✓	✓	✓	×	✓
Luo et al.	2023	[44]	5/6	Attribute-based authentication	ROM	–	–	✓	✓	✓	×	✓	×	✓	✓	✓	✓
Gong et al.	2024	[81]	6/6	Attribute-based authentication	SM	–	–	✓	✓	×	✓	×	✓	✓	✓	✓	✓
Our scheme	2024	–	1/2	Password+Attributes	eCK	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

†The details of criteria C1∼C12 are referred to Section III-C. Note that "✓" means achieving the corresponding goal, while "×" not, "–" means not applicable.
‡The protocol rounds/flows means the number of two-way and one-way communications between the two parties, which are equal if the message is transmitted asynchronously. If it is synchronous message transmission, the message flows is twice the number of communication rounds. "SM" denotes the standard model.

TABLE IV: Performance Comparison among Relevant Authentication Schemes*

Scheme	Ref.	Computation cost			Communication cost		
		User (or Alice)	Server (or Bob)	Total	User (or Alice)	Server (or Bob)	Total
Li et al.	[52]	$8T_M + 12T_H$	$12T_M + 8T_H + 2T_{Enc}$	$\approx$ 104.432 ms	2,928 bits	3,536 bits	6,464 bits
Zhang et al.	[53]	$3T_{Enc} + 12T_M$	$16T_M + 13T_E$	$\approx$ 134.078 ms	3,152 bits	4,280 bits	7,432 bits
Srinivas et al.	[75]	$13T_M + 15T_H$	$16T_H + 2T_{Enc}$	$\approx$ 95.007 ms	2,240 bits	3,664 bits	5,904 bits
Li et al.	[76]	$12T_E + 9T_H$	$5T_M + 14T_E + 16T_H + 3T_P$	$\approx$ 124.539 ms	3,528 bits	4,216 bits	7,744 bits
Tseng et al.	[77]	$15T_E + T_P$	$12T_E + 15T_H + 9T_P$	$\approx$ 129.744 ms	2,808 bits	4,048 bits	6,856 bits
Abbasinezhad et al.	[78]	$12T_M + 10T_H + T_{Enc}$	$9T_M + 14T_H + 3T_{Enc}$	$\approx$ 117.431 ms	2,928 bits	4,632 bits	7,560 bits
Vijayakumar et al.	[55]	$6T_P + 13T_H$	$13T_M + 9T_P$	$\approx$ 141.467 ms	3,344 bits	3,216 bits	6,560 bits
Wang et al.	[26]	$14T_M + 12T_H$	$13T_M + 9T_H$	$\approx$ 127.437 ms	3,088 bits	4,408 bits	7,496 bits
Tan et al.	[42]	$9T_P + 12T_M + 11T_H$	$lT_M + (2l+2)T_H$	$\approx$ 139.855 ms	3,092 bits	4,728 bits	7,820 bits
Sucasas et al.	[43]	$(3l+8)T_E + l \cdot T_P$	$(5l+6)T_E + 2l \cdot T_M$	$\approx$ 176.478 ms	3,018 bits	4,408 bits	7,426 bits
Luo et al.	[44]	$2l \cdot T_{Enc} + 2l \cdot T_H + 2l \cdot T_M$	$5l \cdot T_M + 3l \cdot T_P$	$\approx$ 234.685 ms	3,152 bits	4,458 bits	7,610 bits
Guo et al.	[79]	$(2l+1)T_E + (2l+3)T_P + T_H$	$(3l+5)T_E + (3l+2)T_P + 3T_H$	$\approx$ 223.756 ms	3,640 bits	5,440 bits	9,080 bits
Belguith et al.	[80]	$(2l+6)T_E + (2l+7)T_P$	$(l+3)T_E + (4l+2)T_P + (l+5)T_M$	$\approx$ 284.694 ms	3,890 bits	6,464 bits	10,354 bits
Our scheme	–	$5T_P + (2l+9)T_E + (2l+4)T_M + 5T_H$	$5T_P + (2l+9)T_E + (2l+4)T_M + 5T_H$	$\approx$ 246.900 ms	4,160 bits	4,160 bits	8,320 bits

* In peer-to-peer authentication scenarios, both $Alice$ and Bob serve as the authentication server for each other. For the sake of clarity and without loss of generality, we see Bob as the authentication server. "l" represents the number of attributes submitted by the user, the computation cost is linearly increasing with the number of attributes. In our scheme, assuming $l = 5$, the corresponding computation time of $Alice$ or Bob is 123.450 ms and the corresponding communication cost is 4,160 bits.

Therefore, the model only covers outsider adversaries and does not account for attacks by malicious insiders who may try to get secret information. In comparison, our proposal not only achieves higher security but also optimizes the number of communication rounds.

B. Computational Analysis

We conducted a comprehensive evaluation of the computational overhead associated with fundamental cryptographic operations, namely bilinear pairing, modular exponentiation, and hash operations. The results of this analysis are presented in Table IV, which provides a theoretical comparison. Our scheme exhibits a linear increase in algorithmic time corresponding to the number of attributes. Fig. 3 illustrates the execution of the system initialization and key generation algorithms by the trusted authority (TA), while Fig. 4 demonstrates the execution of the signcryption and verification algorithms by the participating entities. It is apparent that the computational cost of our proposed scheme scales proportionally with the number of attributes. These experiments demonstrate the efficiency of our scheme, affirming its practical applicability for user authentication in real-world scenarios.

Note that, for peer to peer authentication scenarios, Bob serves as the authentication server for $Alice$, and vice versa. Without loss of generality, we see Bob as the authentication server. Since our protocol satisfies the property of *role symmetric*, i.e., both communication parties execute the same operations, so $Alice$ and Bob have the same computation overhead and communication overhead.

C. Communication Analysis

Furthermore, we thoroughly evaluate the communication costs, considering specific parameter lengths for various components. In our analysis, we set the lengths as follows: $|G| = |G_T| = 1024$ bits, $|Z_p^*| = 160$ bits, $|\kappa| = 160$ bits, the authentication access policy $|\Gamma| = 64$ bits. In the PBC Library, Type A is denoted as $E(F_q) : y^2 = x^3 + x$, where the groups G and G_T of order p are subgroups of $E(F_q)$. The parameters p and q correspond to 160 bits and 512 bits, respectively. For consistency and simplicity, we assume one password length of 64 bits, and one block encryption or hash operation is 256 bits. In our AB-PAKE protocol, the parameters $\{\Gamma_A, C_A, C_{A_1}, C_{A_2}, \sigma_A\}$ are delivered, where $C_A, C_{A_1}, \sigma_A \in G_T$ and $C_{A_2} \in G$. Consequently, the communication cost for single authentication process is $3|G_T| + |G| + |\Gamma| = 3 \times 1024 + 1024 + 64 = 4160$ bits. Table IV shows that the communication overhead of our proposed scheme is quite reasonable and acceptable.

IX. Conclusion

We present AB-PAKE, a one-round attribute-based password authenticated key exchange protocol, where a client can authenticate to the other party by a shared password and a set of expected attributes. In contrast to previous two-factor authentication schemes, our protocol offers flexibility in user authentication and finer-grained access control, achieving truly two-factor security even if ephemeral secret keys of two participants have been leaked. Neither participant discloses any privacy information to the other party who fails to satisfy the authentication policy requirements. Additionally, our work adds a desirable feature to two-factor authentication, that allows clients to change their secret key according to their preferences (i.e., attribute update), thereby the reproducibility of two authentication factors is realized. To thoroughly evaluate our scheme, we conduct a comprehensive comparative analysis considering communication and computation overhead, as well as other criteria. The comparison results demonstrate the priority of our proposed scheme.

References

[1] J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano, "Passwords and the evolution of imperfect authentication," *Commun. ACM*, vol. 58, no. 7, pp. 78–87, 2015.

[2] J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano, "The quest to replace passwords: A framework for comparative evaluation of web authentication schemes," in *Proc. IEEE S&P 2012*, pp. 553–567.

[3] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in *Proc. EUROCRYPT 2000*, vol. 1807, pp. 139–155.

[4] F. Benhamouda, O. Blazy, C. Chevalier, D. Pointcheval, and D. Vergnaud, "New techniques for sphfs and efficient one-round PAKE protocols," in *Proc. CRYPTO 2013*, vol. 8042, pp. 449–475.

[5] V. Boyko, P. D. MacKenzie, and S. Patel, "Provably secure password-authenticated key exchange using diffie-hellman," in *Proc. EUROCRYPT 2000*, vol. 1807, pp. 156–171.

[6] J. Katz and V. Vaikuntanathan, "Round-optimal password-based authenticated key exchange," *J. Cryptol.*, vol. 26, no. 4, pp. 714–743, 2013.

[7] S. Jarecki, H. Krawczyk, and J. Xu, "OPAQUE: an asymmetric PAKE protocol secure against pre-computation attacks," in *Proc. EUROCRYPT 2018*, vol. 10822, pp. 456–486.

[8] S. Jarecki, H. Krawczyk, M. Shirvanian, and N. Saxena, "Device-enhanced password protocols with optimal online-offline protection," in *Proc. ACM ASIACCS 2016*, pp. 177–188.

[9] S. M. Bellovin and M. Merritt, "Encrypted key exchange: password-based protocols secure against dictionary attacks," in *Proc. IEEE S&P 1992*, pp. 72–84.

[10] A. Das, J. Bonneau, M. Caesar, N. Borisov, and X. Wang, "The tangled web of password reuse," in *Proc. NDSS 2014*, vol. 14, pp. 23–26.

[11] L. H. Newman, "Yahoo's 2013 email hack actually compromised three billion accounts," 2019, https://www.wired.com/story/yahoo-breach-three-billion-accounts/.

[12] E. Mikalauskas, "Rockyou2021: largest password compilation of all time leaked online with 8.4 billion entries," 2021, https://cybernews.com/security/rockyou2021-alltime-largest-password-compilation-leaked/.

[13] O. Powell, "More than 3.8 billion records exposed in darkbeam data leak," 2023, https://www.cshub.com/data/news/darkbeam-data-leak.

[14] P. Grassi, E. Newton, J. Fenton, R. Perlner, A. Regenscheid, W. Burr, J. Richer, N. Lefkovitz, J. Danker, Y.-Y. Choong, K. Greene, and M. Theofanos, "NIST 800-63B digital identity guidelines: Authentication and lifecycle management," McLean, VA, Tech. Rep., June 2017.

[15] D. Boneh, H. Corrigan-Gibbs, and S. E. Schechter, "Balloon hashing: A memory-hard function providing provable protection against sequential attacks," in *Proc. ASIACRYPT 2016*, vol. 10031, pp. 220–248.

[16] J. Alwen, B. Chen, C. Kamath, V. Kolmogorov, K. Pietrzak, and S. Tessaro, "On the complexity of scrypt and proofs of space in the parallel random oracle model," in *Proc. EUROCRYPT 2016*, vol. 9666, pp. 358–387.

[17] A. Everspaugh, R. Chatterjee, S. Scott, A. Juels, and T. Ristenpart, "The pythia PRF service," in *Proc. USENIX SEC 2015*, pp. 547–562.

[18] J. Schneider, N. Fleischhacker, D. Schröder, and M. Backes, "Efficient cryptographic password hardening services from partially oblivious commitments," in *Proc. ACM CCS 2016*, pp. 1192–1203.

[19] R. W. F. Lai, C. Egger, D. Schröder, and S. S. M. Chow, "Phoenix: Rebirth of a cryptographic password-hardening service," in *Proc. USENIX SEC 2017*, pp. 899–916.

[20] R. W. F. Lai, C. Egger, M. Reinert, S. S. M. Chow, M. Maffei, and D. Schröder, "Simple password-hardened encryption services," in *Proc. USENIX SEC 2018*, pp. 1405–1421.

[21] J. Brost, C. Egger, R. W. F. Lai, F. Schmid, D. Schröder, and M. Zoppelt, "Threshold password-hardened encryption services," in *Proc. ACM CCS 2020*, pp. 409–424.

[22] S. Agrawal, P. Miao, P. Mohassel, and P. Mukherjee, "PASTA: password-based threshold authentication," in *Proc. ACM CCS 2018*, pp. 2042–2059.

[23] Y. Zhang, C. Xu, H. Li, K. Yang, N. Cheng, and X. Shen, "PROTECT: efficient password-based threshold single-sign-on authentication for mobile users against perpetual leakage," *IEEE Trans. Mob. Comput.*, vol. 20, no. 6, pp. 2297–2312, 2021.

[24] J. Li, X. Chen, M. Li, J. Li, P. P. Lee, and W. Lou, "Secure deduplication with efficient and reliable convergent key management," *IEEE Trans. Parallel. Distrib. Syst.*, vol. 25, no. 6, pp. 1615–1625, 2014.

[25] S. Jarecki, M. Jubur, H. Krawczyk, N. Saxena, and M. Shirvanian, "Two-factor password-authenticated key exchange with end-to-end security," *ACM Trans. Priv. Secur.*, vol. 24, no. 3, pp. 1–37, 2021.

[26] Q. Wang, D. Wang, C. Cheng, and D. He, "Quantum2fa: Efficient quantum-resistant two-factor authentication scheme for mobile devices," *IEEE Trans. Depend. Secur. Comput.*, vol. 20, no. 1, pp. 193–208, 2023.

[27] D. Wang and P. Wang, "Two birds with one stone: two-factor authentication with security beyond conventional bound," *IEEE Trans. Depend. Secur. Comput.*, vol. 15, no. 4, pp. 708–722, 2018.

[28] D. Wang, D. He, P. Wang, and C. Chu, "Anonymous two-factor authentication in distributed systems: certain goals are beyond attainment," *IEEE Trans. Depend. Secur. Comput.*, vol. 12, no. 4, pp. 428–442, 2015.

[29] D. Liu, Q. Wang, M. Zhou, P. Jiang, Q. Li, C. Shen, and C. Wang, "Soundid: securing mobile two-factor authentication via acoustic signals," *IEEE Trans. Depend. Secur. Comput.*, vol. 20, no. 2, pp. 1687–1701, 2023.

[30] C. Wang, D. Wang, Y. Duan, and X. Tao, "Secure and lightweight user authentication scheme for cloud-assisted internet of things," *IEEE Trans. Inf. Forensics Secur.*, vol. 18, pp. 2961–2976, 2023.

[31] L. Zhang, Y. Zhang, S. Tang, and H. Luo, "Privacy protection for e-health systems by means of dynamic authentication and three-factor key agreement," *IEEE Trans. Ind. Electron.*, vol. 65, pp. 2795–2805, 2018.

[32] A. de la Piedra, M. Venema, and G. Alpár, "Acabella: Automated (crypt) analysis of attribute-based encryption leveraging linear algebra," in *Proc. ACM CCS 2023*, pp. 3269–3283.

[33] M. C. Gorantla, C. Boyd, and J. M. G. Nieto, "Attribute-based authenticated key exchange," in *Proc. ACISP 2010*, vol. 6168, pp. 300–317.

[34] K. Yoneyama, "Strongly secure two-pass attribute-based authenticated key exchange," in *Proc. Pairing 2010*, vol. 6487, 2010, pp. 147–166.

[35] Q. Huang, W. Yue, Y. Yang, and L. Chen, "P2GT: fine-grained genomic data access control with privacy-preserving testing in cloud computing," *IEEE ACM Trans. Comput. Biol. Bioinform.*, vol. 19, no. 4, pp. 2385–2398, 2022.

[36] V. Kolesnikov, H. Krawczyk, Y. Lindell, A. J. Malozemoff, and T. Rabin, "Attribute-based key exchange with general policies," in *Proc. ACM CCS 2016*, pp. 1451–1463.

[37] S. Chakraborty, Y. S. Rao, and C. P. Rangan, "Efficient single round attribute-based authenticated key exchange protocol," *Int. J. Comput. Math. Comput. Syst. Theory*, vol. 6, no. 4, pp. 313–336, 2021.

[38] H. Zheng, J. Qin, J. Hu, and Q. Wu, "Threshold attribute-based signcryption and its application to authenticated key agreement," *Secur. Commun. Networks*, vol. 9, no. 18, pp. 4914–4923, 2016.

[39] L. Guo, C. Zhang, J. Sun, and Y. Fang, "A privacy-preserving attribute-based authentication system for mobile health networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 9, pp. 1927–1941, 2013.

[40] S. Aerts, D. Lambrechts, S. Maity, P. Van Loo, B. Coessens, F. De Smet, L.-C. Tranchevent, B. De Moor, P. Marynen, B. Hassan *et al.*, "Gene prioritization through genomic data fusion," *Nature biotechnol.*, vol. 24, no. 5, pp. 537–544, 2006.

[41] S. Garg, C. Gentry, S. Halevi, A. Sahai, and B. Waters, "Attribute-based encryption for circuits from multilinear maps," in *CRYPTO 2013*, pp. 479–499.

[42] H. Tan, W. Zheng, Y. Guan, and R. Lu, "A privacy-preserving attribute-based authenticated key management scheme for accountable vehicular

communications," *IEEE Trans. Veh. Technol.*, vol. 72, no. 3, pp. 3622–3635, 2023.
[43] V. Sucasas, G. Mantas, M. Papaioannou, and J. Rodriguez, "Attribute-based pseudonymity for privacy-preserving authentication in cloud services," *IEEE Trans. on Cloud Comput.*, vol. 11, pp. 168–184, 2023.
[44] F. Luo, H. Wang, C. Lin, and X. Yan, "Abaeks: Attribute-based authenticated encryption with keyword search over outsourced encrypted data," *IEEE Trans. Inf. Forensics Secur.*, vol. 18, pp. 4970–4983, 2023.
[45] Y. Zolotavkin, J. J. Jeong, V. Kuchta, M. Slavnenko, and R. Doss, "Improving unlinkability of attribute-based authentication through game theory," *ACM Trans. Priv. Secur.*, vol. 25, no. 2, pp. 2471–2566, 2022.
[46] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory*, vol. 22, no. 6, pp. 644–654, 1976.
[47] H. Krawczyk, "Hmqv: A high-performance secure diffie-hellman protocol," in *Proc. CRYPTO 2005*, vol. 3621, pp. 546–566.
[48] W. Wang, "Heartbleed – openssl zero-day bug leaves millions of websites vulnerable," 2014, https://thehackernews.com/2014/04/heartbleed-openssl-zero-day-bug-leaves.html.
[49] J. Horn, W. Haas, T. Prescher, D. Gruss, M. Lipp, S. Mangard, and M. Schwarz, "Meltdown: Reading kernel memory from user space," in *Proc. USENIX SEC 2018*, vol. 18.
[50] P. Kocher, J. Horn, A. Fogh, D. Genkin, D. Gruss *et al.*, "Spectre attacks: Exploiting speculative execution," in *Proc. IEEE S&P 2019*, pp. 1–19.
[51] B. A. LaMacchia, K. E. Lauter, and A. Mityagin, "Stronger security of authenticated key exchangee," in *Proc. ProvSec 2007*, pp. 1–16.
[52] X. Li, D. Yang, X. Zeng, B. Chen, and Y. Zhang, "Comments on "provably secure dynamic id-based anonymous two-factor authenticated key exchange protocol with extended security model," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 12, pp. 3344–3345, 2019.
[53] R. Zhang, Y. Xiao, S. Sun, and H. Ma, "Efficient multi-factor authenticated key exchange scheme for mobile communications," *IEEE Trans. Depend. Secur. Comput.*, vol. 16, no. 4, pp. 625–634, 2019.
[54] M. Abdalla and D. Pointcheval, "Simple password-based encrypted key exchange protocols," in *Proc. CT-RSA 2005*, vol. 3376, pp. 191–208.
[55] P. Vijayakumar, M. Azees, S. A. Kozlov, and J. J. P. C. Rodrigues, "An anonymous batch authentication and key exchange protocols for 6g enabled vanets," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 1630–1638, 2022.
[56] M. Wazid, A. K. Das, N. Kumar, and J. J. P. C. Rodrigues, "Secure three-factor user authentication scheme for renewable-energy-based smart grid environment," *IEEE Trans. Ind. Informatics*, vol. 13, no. 6, pp. 3144–3153, 2017.
[57] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, "Zipf's law in passwords," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 11, pp. 2776–2791, 2017.
[58] G. Ateniese, J. Kirsch, and M. Blanton, "Secret handshakes with dynamic and fuzzy matching," in *Proc. NDSS 2007*, vol. 7, no. 24, pp. 43–54.
[59] Z. Guan, Y. Zhang, L. Zhu, L. Wu, and S. Yu, "EFFECT: an efficient flexible privacy-preserving data aggregation scheme with authentication in smart grid," *Sci. China Inf. Sci.*, vol. 62, no. 3, pp. 1–14, 2019.
[60] J. Sun, H. Xiong, X. Nie, Y. Zhang, and P. Wu, "On the security of privacy-preserving attribute-based keyword search in shared multi-owner setting," *IEEE Trans. Depend. Secur. Comput.*, vol. 18, no. 5, pp. 2518–2519, 2021.
[61] J. Camenisch and E. V. Herreweghen, "Design and implementation of the *idemix* anonymous credential system," in *Proc. ACM CCS 2002*, pp. 21–30.
[62] S. Xia, X. Tao, N. Li, S. Wang, T. Sui, H. Wu, J. Xu, and Z. Han, "Multiple correlated attributes based physical layer authentication in wireless networks," *IEEE Trans. Veh. Technol.*, vol. 70, no. 2, pp. 1673–1687, 2021.
[63] X. Yin, X. Fang, N. Zhang, P. Yang, X. Sha, and J. Qiu, "Online learning aided adaptive multiple attribute-based physical layer authentication in dynamic environments," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 1106–1116, 2021.
[64] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell *et al.*, "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, no. 7779, pp. 505–510, 2019.
[65] M. Ringbauer, M. Meth, L. Postler, R. Stricker, R. Blatt, P. Schindler, and T. Monz, "A universal qudit quantum processor with trapped ions," *Nature Phys.*, vol. 18, no. 9, pp. 1053–1057, 2022.
[66] S. Pirandola *et al.*, "Advances in quantum cryptography," *Adv. Opt. Photon.*, vol. 12, no. 4, pp. 1012–1236, 2020.
[67] Y. Qin, C. Cheng, X. Zhang, Y. Pan, L. Hu, and J. Ding, "A systematic approach and analysis of key mismatch attacks on lattice-based nist candidate kems," in *Proc. ASIACRYPT 2021*, pp. 92–121.
[68] Z. Li, D. Wang, and E. Morais, "Quantum-safe round-optimal password authentication for mobile devices," *IEEE Trans. Depend. Sec. Comput.*, vol. 19, no. 3, pp. 1885–1899, 2022.
[69] J. W. Bos, C. Costello, M. Naehrig, and D. Stebila, "Post-quantum key exchange for the tls protocol from the ring learning with errors problem," in *Proc. IEEE S&P 2015*, pp. 553–570.
[70] Y. Qin, R. Ding, C. Cheng, N. Bindel, Y. Pan, and J. Ding, "Light the signal: Optimization of signal leakage attacks against lwe-based key exchange," in *Proc. ESORICS 2022*, pp. 677–697.
[71] D. Boneh, C. Gentry, S. Gorbunov, S. Halevi, V. Nikolaenko, G. Segev, V. Vaikuntanathan, and D. Vinayagamurthy, "Fully key-homomorphic encryption, arithmetic circuit abe and compact garbled circuits," in *Proc. EUROCRYPT 2014*, pp. 533–556.
[72] S. Gorbunov, V. Vaikuntanathan, e. R. Wee, Hoeteck", and M. Robshaw, "Predicate encryption for circuits from lwe," in *Proc. CRYPTO 2015*, pp. 503–523.
[73] Y. Wang, "Password protected smart card and memory stick authentication against off-line dictionary attacks," in *Proc. SEC 2012*, pp. 489–500.
[74] D. Wang and P. Wang, "Offline dictionary attack on password authentication schemes using smart cards," in *Proc. ISC 2013*, pp. 221–237.
[75] J. Srinivas, A. K. Das, M. Wazid, and N. Kumar, "Anonymous lightweight chaotic map-based authenticated key agreement protocol for industrial internet of things," *IEEE Trans. Depend. Secur. Comput.*, vol. 17, no. 6, pp. 1133–1146, 2020.
[76] W. Li, X. Li, J. Gao, and H. Wang, "Design of secure authenticated key management protocol for cloud computing environments," *IEEE Trans. Depend. Secur. Comput.*, vol. 18, no. 3, pp. 1276–1290, 2021.
[77] C. Wang and C. Wang, "A highly secure three-party authentication key exchange protocol and its application in e-business with ECK model," *Int. J. Inf. Comput. Secur.*, vol. 16, no. 3, pp. 399–419, 2021.
[78] D. Abbasinezhad-Mood, S. M. Mazinani, M. Nikooghadam, and A. Ostad-Sharif, "Efficient provably-secure dynamic id-based authenticated key agreement scheme with enhanced security provision," *IEEE Trans. Depend. Secur. Comput.*, vol. 19, no. 2, pp. 1227–1238, 2022.
[79] N. Guo, C. Zhao, and T. Gao, "An anonymous authentication scheme for edge computing-based car–home connectivity services in vehicular networks," *Future Gener. Comput. Syst.*, vol. 106, pp. 659–671, 2020.
[80] S. Belguith, N. Kaaniche, M. Hammoudeh, and T. Dargahi, "Proud: Verifiable privacy-preserving outsourced attribute based signcryption supporting access policy update for cloud assisted iot applications," *Future Gener. Comput. Syst.*, vol. 111, pp. 899–918, 2020.
[81] B. Gong, C. Guo, C. Guo, C. Guo, Y. Sun, M. Waqas, and S. Chen, "Slim: A secure and lightweight multi-authority attribute-based signcryption scheme for iot," *IEEE Trans. Inf. Forensics Secur.*, vol. 19, pp. 1299–1312, 2024.

Mi Song received the MS degree in information security from the Northwest Normal University, Lanzhou, P. R. China, in Jun. 2022. She is currently working toward the PhD degree from the College of Cyber science, Nankai University, Tianjin, P. R. China. Her research interests include applied cryptography and password-based authentication.

Ding Wang received his Ph.D. degree in Information Security at Peking University in 2017, and was supported by the "Boya Postdoctoral Fellowship" in Peking University from 2017 to 2019. Currently, he is a Full Professor at Nankai University. As the first author (or corresponding author), he has published more than 90 papers at venues like IEEE S&P, ACM CCS, NDSS, USENIX Security, IEEE TIFS and IEEE TDSC. His research has been reported by over 200 medias like Daily Mail, Forbes, IEEE Spectrum and Communications of the ACM, appeared in the Elsevier 2017 "Article Selection Celebrating Computer Science Research in China", and resulted in the revision of the authentication guideline NIST SP800-63-2. He has been involved in the community as a PC Chair/TPC member for over 60 international conferences such as NDSS 2023-2025, ACM CCS 2022, RAID 2024/2023, PETS 2022-2024, ACSAC 2020-2024, AsiaCCS 2022/2021, ICICS 2019-2024, and SPNCE 2020–2022. His research interests include passwords, authentication, and provable security.