Two-round password-only authenticated key exchange in the three-party setting

Junghyun Nam[a,*], Kim-Kwang Raymond Choo[b], Juryon Paik[c], Dongho Won[c]

[a]*Department of Computer Engineering, Konkuk University, 268 Chungwondaero, Chungju, Chungcheongbukdo 380-701, Korea*
[b]*Information Assurance Research Group, Advanced Computing Research Centre, University of South Australia, Mawson Lakes, SA 5095, Australia*
[c]*Department of Computer Engineering, Sungkyunkwan University, 2066 Seoburo, Suwon, Gyeonggido 440-746, Korea*

Abstract

We present the first provably-secure 3-party password-only authenticated key exchange (PAKE) protocol that can run in only two communication rounds. Our protocol is generic in the sense that it can be constructed from any 2-party PAKE protocol. The protocol is proven secure in a variant of the widely accepted model of Bellare, Pointcheval and Rogaway (2000) without any idealized assumptions on the cryptographic primitives used. We also investigate the security of the 2-round 3-party PAKE protocol of Wang, Hu and Li (2010), and demonstrate that this protocol cannot achieve implicit key authentication in the presence of an active adversary.

Keywords: Password-only authenticated key exchange (PAKE), Three-party key exchange, Communication round, Dictionary attack, Implicit key authentication

1. Introduction

Protocols for password-only authenticated key exchange (PAKE) enable two or more parties to generate a shared, cryptographically strong key (called a session key) from their easy-to-remember passwords. PAKE protocols are increasingly popular, and perhaps due to the popularity of passwords as explained by Herley and van Oorschot that '[d]espite countless attempts to dislodge passwords [in the past 20 years], they are more widely used and firmly entrenched than ever' [1]. There has been an enormous amount of research effort expended in design and analysis of PAKE protocols and yet there are still worthwhile contributions to be made even in the simple scenario of two protocol participants (also known as clients) with an online trusted server. In such a 3-party model, the server provides its registered clients with a centralized authentication service, which allows each client to remember and manage only a single password. Password guessing attacks (also known as dictionary attacks) present a more subtle threat in the 3-party model (than a 2-party model) as a malicious client can attempt to mount such an attack against another client – see [2, 3, 4, 5, 6].

It is generally regarded that the design of secure yet efficient key exchange protocols (including PAKE protocols) is notoriously hard, and conducting security analysis for such protocols is time-consuming and error-prone; see, e.g., [7, 8, 9]. The many flaws discovered in published protocols have promoted the use of formal models and rigorous security proofs. In the provable security paradigm for key exchange protocols, a deductive reasoning process is adopted whereby

[*]E-mail: jhnam@kku.ac.kr.

emphasis is placed on a proven reduction from the problem of breaking the protocol to another problem believed to be (computationally) hard. A complete mathematical proof with respect to cryptographic definitions provides a strong assurance that a protocol is behaving as desired. It is by now standard practice for protocol designers to provide proofs of security for their protocols in widely accepted security models, in order to assure protocol implementers about the security properties of protocols. The provable security paradigm for key exchange protocols was made popular by Bellare and Rogaway [10] who provided the first formal definition for a model of adversary capabilities with an associated definition of the indistinguishability-based security. Bellare and Rogaway's model has been further revised several times, and a recent revision is the Real-Or-Random (ROR) model proposed by Abdalla, Fouque and Pointcheval [11, 12] for 3-party PAKE protocols.

A number of 3-party PAKE protocols have been proposed over the last decade [13, 14, 11, 15, 16, 12, 2, 17, 18, 19, 20, 3, 21, 22, 23, 24, 25, 5, 6]. Many of these protocols have never been proven secure in any model [13, 17, 18, 19, 20, 3, 21] and/or have been found to be vulnerable to some attack(s) [8, 2, 27, 18, 19, 28, 20, 3, 23, 29, 5, 6, 30, 31, 32, 33]. Some protocols [11, 12, 15, 2, 23, 24] have been proven secure only in a restricted model, in which the adversary is not allowed to corrupt protocol participants and thus no attacks by malicious clients can be captured.

Reducing the number of communication rounds is an important practical consideration in designing key exchange protocols. Adopting the usual convention in the 3-party (and multi-party) setting, we let a round consist of all protocol messages that can be sent in parallel; note that messages in the same round cannot be dependent on one another. So far, there have been several 2-round key exchange protocols presented in the 3-party setting.

- The protocols of [15, 24] are the only 2-round 3-party PAKE protocols published with a claimed security proof[1]. However, it was later found that both protocols are not secure against an active adversary and their associated claims of provable security are invalid (see [8, 34, 2, 33] and Section 3 of this paper).

- The protocols of [35, 36] were proven secure and require only two rounds, but these protocols assume a "hybrid" 3-party setting where a server's public key is required in addition to passwords.

- The recent protocol due to Tsai and Chang [31] can run in two rounds (without key confirmation), but this protocol only works in a hyrid setting that requires both a cryptographic key and a password pre-established between each client and the server (see [37, 38, 39, 40, 41, 4, 42, 43, 26, 30, 44] for other protocols designed to work in a hybrid setting).

Table 1 summarizes the security properties and known weaknesses of published 2-round 3-party PAKE protocols with (claimed) proofs of security. To the best of our knowledge, there exists no (provably) secure 3-party PAKE protocol running in only two rounds.

We regard our contributions of this paper to be two-fold:

1. We present the first 2-round 3-party PAKE protocol that is provably secure in a well-defined communication model - see Section 4. The communication model in which we

[1]Although the two protocols presented by Lee and Hwang [23] can run in two rounds (without key confirmation), they are insecure in the presence of a malicious client [32]; both protocols are susceptible to a man-in-the-middle attack as well as an offline dictionary attack.

Table 1: A summary of security results for existing 2-round 3-party PAKE protocols.

Protocol	Major Weaknesses	Communication Model	Security Proof
3PAKE [15]	Vulnerable to an offline dictionary attack [33]	The adversary is restricted from corrupting protocol participants	Based on an invalid assumption [34]
NWPAKE-2 [24]	Fails to achieve implicit key authentication (see Section 3)		Invalidated by an active attack (see Section 3)
S-IA-3PAKE, S-EA-3PAKE [23]	Vulnerable to an offline dictionary attack and a man-in-the-middle attack [32]		Invalidated by a passive attack (see Section 3.3 of [32])

work allows the adversary to corrupt protocol participants and therefore, captures not only the notion of forward secrecy but also attacks by malicious clients. We make no idealizing assumptions in our security proof. Similar to the protocols of [11, 12, 2, 19, 24], our protocol is generic in the sense that it can be constructed from any 2-party PAKE protocol. If the underlying 2-party protocol is round-optimal [45, 46, 47], then our 3-party protocol runs in only two communication rounds.

2. We also present a previously unpublished flaw in an existing 2-round 3-party PAKE protocol proposed by Wang, Hu and Li [24] - see Section 3.2. The Wang-Hu-Li protocol (named NWPAKE-2) was claimed to be provably secure in a variant of the ROR model. We reveal that the NWPAKE-2 protocol fails to achieve implicit key authentication in the presence of an active adversary who is not even registered with the server, which invalidates the "claimed" security proof.

The remainder of this paper is structured as follows: Section 2 describes a communication model along with the associated security definition. In Section 3, we revisit the NWPAKE-2 protocol of Wang, Hu and Li [24] and reveal a previously unpublished flaw in the protocol. We then present our proposed 2-round 3-party PAKE protocol and prove its security in Section 4. The last section concludes the paper.

2. The communication model

We now describe a communication model adapted from the widely accepted indistinguishability-based model of Bellare, Pointcheval and Rogaway [45]. This will be the model that is used to prove the security of our proposed 3-party PAKE protocol.

Participants and long-term keys. Let S be a trusted authentication server, and $\mathcal{C}$ the set of all clients registered with S. During registration, each client $C \in \mathcal{C}$ selects a password pw_C from dictionary $\mathcal{D}$, and shares pw_C with S via a secure/authenticated channel. The password pw_C is used as the long-term secret key between C and S. Any two clients $C, C' \in \mathcal{C}$ may run a 3-party PAKE protocol P with S at any point in time to establish a session key. Let $\mathcal{U} = \mathcal{C} \cup \{S\}$. A user $U \in \mathcal{U}$ may execute the protocol multiple times (including concurrent executions) with the same or different participants. Thus, at a given time, there could be many instances of a single user. We use Π_U^i to denote instance i of user U. We say that a client instance Π_C^i *accepts* when it successfully computes its session key sk_C^i in an execution of the protocol.

Partnering. Intuitively, two instances are *partners* if they participate in a protocol execution and establish a (shared) session key. Formally, partnering between instances is defined in terms of the notions of session identifiers and partner identifiers (see [48] on the role and the possible construct of session and partner identifiers as a form of partnering mechanism that enables the right session key to be identified in concurrent protocol executions). Session identifier (*sid*) is a unique identifier of a protocol session and is usually defined as a function of the messages transmitted in the session. Let sid_U^i denotes the *sid* of instance Π_U^i. A partner identifier (*pid*) is a sequence of identities of participants of a specific protocol session. Instances are given as input a *pid* before they can run the protocol. pid_U^i denotes the *pid* given to instance Π_U^i. In a typical session, there will be three participants, namely two clients C and C' and the server S. We say that two instances Π_C^i and $\Pi_{C'}^j$ are partners if all of the following conditions are satisfied: (1) both Π_C^i and $\Pi_{C'}^j$ have accepted, (2) $sid_C^i = sid_{C'}^j$, and (3) $pid_C^i = pid_{C'}^j$.

Adversary capabilities. The probabilistic polynomial-time (PPT) adversary $\mathcal{A}$ is in complete control of all communications between users, and it's capabilities are modeled via a pre-defined set of oracle queries as described below.

- $\mathsf{Execute}(\Pi_C^i, \Pi_{C'}^j, \Pi_S^k)$: This query models passive attacks against the protocol. It prompts an execution of the protocol between the instances Π_C^i, $\Pi_{C'}^j$ and Π_S^k, and returns the transcript of the protocol execution to $\mathcal{A}$.

- $\mathsf{Send}(\Pi_U^i, m)$: This query sends message m to instance Π_U^i, modelling active attacks against the protocol. Upon receiving m, the instance Π_U^i proceeds according to the protocol specification. The message output by Π_U^i, if any, is returned to $\mathcal{A}$. A query of the form $\mathsf{Send}(\Pi_C^i, \texttt{start:}(C, C', S))$ prompts Π_C^i to initiate the protocol with $pid_C^i = (C, C', S)$.

- $\mathsf{Reveal}(\Pi_C^i)$: This query captures the notion of known key security[2], and if Π_C^i has accepted, returns the session key sk_C^i back to $\mathcal{A}$. However, this session (key) will be rendered unfresh (see Definition 1).

- $\mathsf{Corrupt}(U)$: This query returns U's password pw_U to $\mathcal{A}$. If $U = S$ (i.e., the server is corrupted), all clients' passwords stored by the server are returned. This query captures not only the notion of forward secrecy but also attacks by malicious clients.

- $\mathsf{Test}(\Pi_C^i)$: This query is used to define the indistinguishability-based security of the protocol. If Π_C^i has accepted, then depending on a randomly chosen bit b, $\mathcal{A}$ is given either the real session key sk_C^i (when $b = 1$) or a random key drawn from the session-key space (when $b = 0$). $\mathcal{A}$ is allowed to ask as many Test queries as it wishes. All Test queries are answered using the same value of the hidden bit b. Namely, the keys output by the Test oracle are either all real or all random. But, we require that for each different set of partners, $\mathcal{A}$ should access the Test oracle only once.

The number of queries asked by an adversary is referred to as the *query complexity* of the adversary (Q), and is represented as an ordered sequence of five non-negative integers,

[2]It is often reasonable to assume that the adversary will be able to obtain session keys from any session different from the one under attack.

$Q = (q_{\text{exec}}, q_{\text{send}}, q_{\text{reve}}, q_{\text{corr}}, q_{\text{test}})$. These five non-negative integers are the numbers of queries that the adversary asked respectively to the $\mathsf{Execute}$, Send, Reveal, $\mathsf{Corrupt}$, and Test oracles.

Security definition. We define the security of a 3-party PAKE protocol via the notion of *freshness*. Intuitively, a fresh instance is one that holds a session key which should not be known to the adversary $\mathcal{A}$, and an unfresh instance is one whose session key (or some information about the key) can be known by trivial means. The formal definition of freshness is explained in Definition 1.

Definition 1. An instance Π^i_C is fresh if none of the following occurs: (1) $\mathcal{A}$ queries $\mathsf{Reveal}(\Pi^i_C)$ or $\mathsf{Reveal}(\Pi^j_{C'})$, where $\Pi^j_{C'}$ is the partner of Π^i_C, and (2) $\mathcal{A}$ queries $\mathsf{Corrupt}(U)$, for some $U \in pid^i_C$, before Π^i_C or its partner $\Pi^j_{C'}$ accepts.

The security of a 3-party PAKE protocol P is defined in the context of the following experiment:

Experiment $\mathbf{Exp}_0$:

Phase 1. $\mathcal{A}$ makes any oracle queries at will as many times as it wishes, except that:

- $\mathcal{A}$ is not allowed to ask the $\mathsf{Test}(\Pi^i_C)$ query if the instance Π^i_C is unfresh.
- $\mathcal{A}$ is not allowed to ask the $\mathsf{Reveal}(\Pi^i_C)$ query if it has already made a Test query to Π^i_C or $\Pi^j_{C'}$, where $\Pi^j_{C'}$ is the partner of Π^i_C.

Phase 2. Once $\mathcal{A}$ decides that Phase 1 is over, it outputs a bit b' as a guess on the hidden bit b chosen by the Test oracle. $\mathcal{A}$ is said to succeed if $b = b'$.

Let Succ_0 be the event that $\mathcal{A}$ succeeds in the experiment $\mathbf{Exp}_0$. The advantage of $\mathcal{A}$ in breaking the security of the authenticated key exchange protocol P is $\mathsf{Adv}^{\text{ake}}_P(\mathcal{A}) = 2 \cdot \Pr_{P,\mathcal{A}}[\mathsf{Succ}_0] - 1$.

Definition 2. A 3-party PAKE protocol P is AKE-*secure* if, for any PPT adversary $\mathcal{A}$ asking at most q_{send} Send queries, $\mathsf{Adv}^{\text{ake}}_P(\mathcal{A})$ is only negligibly larger than $c \cdot q_{\text{send}}/|\mathcal{D}|$, where c is a very small constant (usually around 2 or 4) when compared with $|\mathcal{D}|$.

To quantify the security of protocol P in terms of the amount of resources expended by adversaries, we let $\mathsf{Adv}^{\text{ake}}_P(t, Q)$ denote the maximum value of $\mathsf{Adv}^{\text{ake}}_P(\mathcal{A})$ over all PPT adversaries $\mathcal{A}$ with time complexity at most t and query complexity at most Q.

3. Revisiting Wang, Hu and Li (2010)'s NWPAKE-2 protocol

Implicit key authentication is the fundamental security property that any given key exchange protocol is expected to achieve. In this section, we show that the NWPAKE-2 protocol of Wang, Hu and Li [24] does not achieve implicit key authentication.

3.1. Protocol description

Let A and B be two clients who wish to establish a session key, and pw_A and pw_B denote the respective passwords of A and B shared with a trusted server S. The public parameters of the NWPAKE-2 protocol include: (1) a cyclic group $\mathbb{G}$ of prime order q, and a generator g of $\mathbb{G}$, (2) a 2-party PAKE protocol 2PAKE, and (3) a pair of message authentication code (MAC) generation/verification algorithms ($\mathsf{Mac}, \mathsf{Ver}$), where Ver outputs a bit, with 1 meaning `accept` and

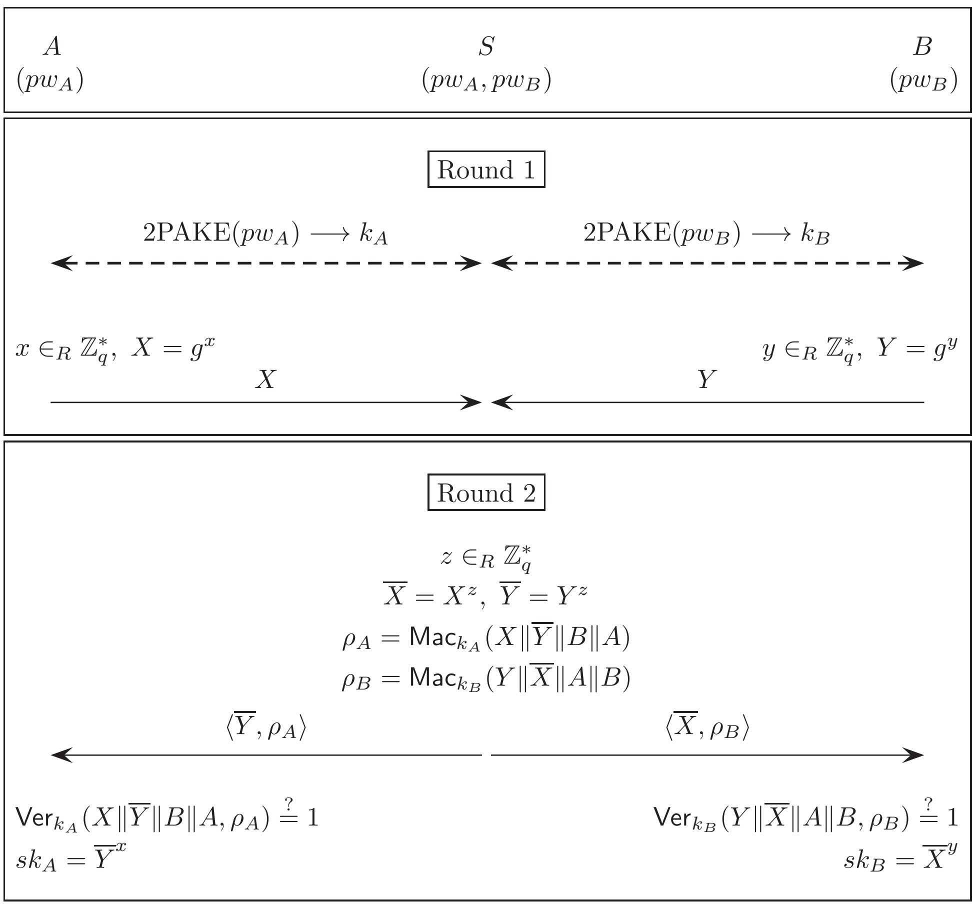
A
(pw_A)
S
(pw_A, pw_B)
B
(pw_B)
Round 1
$2\text{PAKE}(pw_A) \longrightarrow k_A$
$2\text{PAKE}(pw_B) \longrightarrow k_B$
$x \in_R \mathbb{Z}_q^*,\ X = g^x$
$y \in_R \mathbb{Z}_q^*,\ Y = g^y$
X
Y
Round 2
$z \in_R \mathbb{Z}_q^*$
$\overline{X} = X^z,\ \overline{Y} = Y^z$
$\rho_A = \mathsf{Mac}_{k_A}(X \| \overline{Y} \| B \| A)$
$\rho_B = \mathsf{Mac}_{k_B}(Y \| \overline{X} \| A \| B)$
$\langle \overline{Y}, \rho_A \rangle$
$\langle \overline{X}, \rho_B \rangle$
$\mathsf{Ver}_{k_A}(X \| \overline{Y} \| B \| A, \rho_A) \stackrel{?}{=} 1$
$sk_A = \overline{Y}^x$
$\mathsf{Ver}_{k_B}(Y \| \overline{X} \| A \| B, \rho_B) \stackrel{?}{=} 1$
$sk_B = \overline{X}^y$

Fig. 1. Wang et al.'s 2-round 3-party PAKE protocol (NWPAKE-2) [24].

0 meaning `reject`. If the underlying 2-party protocol, 2PAKE, is round-optimal, NWPAKE-2 completes in 2 communication rounds as depicted in Fig. 1. The protocol description is as follows:

Step 1. A and S establish a secret key k_A by running the 2-party protocol 2PAKE. Likewise, B and S establish a secret key k_B.

Step 2. A (resp. B) selects a random $x \in \mathbb{Z}_q^*$ (resp. $y \in \mathbb{Z}_q^*$) and sends $X = g^x$ (resp. $Y = g^y$) to S.

Step 3. S chooses a random $z \in \mathbb{Z}_q^*$, computes

$$\overline{X} = X^z, \quad \overline{Y} = Y^z,$$
$$\rho_A = \mathsf{Mac}_{k_A}(X \| \overline{Y} \| B \| A), \quad \rho_B = \mathsf{Mac}_{k_B}(Y \| \overline{X} \| A \| B),$$

and sends $\langle \overline{Y}, \rho_A \rangle$ and $\langle \overline{X}, \rho_B \rangle$ to A and B, respectively.

Step 4. A and B abort if their received MAC is invalid. Otherwise, they will compute their respective session keys, $sk_A = \overline{Y}^x$ and $sk_B = \overline{X}^y$.

At the end of the protocol execution, A and B will compute the same session key $sk_A = sk_B = g^{xyz}$.

3.2. Violating implicit key authentication

We now assume that there exists an adversary C who is not registered with the server, and demonstrate how C can easily violate the implicit key authentication property of NWPAKE-2.

1. C chooses a random $x' \in \mathbb{Z}_q^*$, computes $X' = g^{x'}$, and replaces X (sent by A to S) with X'.
2. Upon receipt of the "replaced" message, S will compute $\overline{X}$ as $\overline{X} = X'^z$ and therefore, B's session key sk_B will be set to $g^{x'yz}$.
3. C intercepts the message $\langle \overline{Y}, \rho_A \rangle$ sent by S to A, and then computes $sk_C = \overline{Y}^{x'} = g^{x'yz} = sk_B$. In other words, C is able to compute B's session key even though C is not B's partner.

Note that NWPAKE-2 exhibits this security weakness no matter which protocol is used for the instantiation of 2PAKE. Protocols proven secure in a model that allows Send queries should be secure against the above mentioned attack. NWPAKE-2 was claimed to be provably secure in a variant of Abdalla et al.'s ROR model [11, 12] where the adversary is allowed to query $\mathsf{Execute}$, Send, Reveal and Test oracles. This means that the claim of provable security for NWPAKE-2 is invalid[3].

4. Our proposed protocol

This section presents our 2-round 3-party PAKE protocol, which we denote as 2R3PAKE ("R" is for Round), and proves its security in the communication model described in Section 2. The 2R3PAKE protocol is generic in the sense that it can be constructed from any secure 2-party PAKE protocol. Our generic construction takes only one round of communication in addition to the number of rounds required to perform the underlying 2-party protocol. Hence, applying our construction to a round-optimal 2-party PAKE protocol immediately yields a 3-party PAKE protocol running in two communication rounds.

4.1. Preliminaries

The security of 2R3PAKE is based on the decisional Diffie-Hellman assumption and the security of a message authentication code scheme, a 2-party PAKE protocol, and a symmetric encryption scheme.

Decisional Diffie-Hellman (DDH) assumption. Consider a cyclic group $\mathbb{G}$ having prime order q. Informally stated, the DDH problem for $\mathbb{G}$ is to distinguish between two distributions (g^x, g^y, g^{xy}) and (g^x, g^y, g^z), where g is a random generator of $\mathbb{G}$ and x, y, z are chosen at random from $\mathbb{Z}_q^*$. We say that the DDH assumption holds in $\mathbb{G}$ if it is computationally infeasible to solve the DDH problem for $\mathbb{G}$. More formally, we define the advantage of an algorithm $\mathcal{D}$ in solving the DDH problem for $\mathbb{G}$ as $\mathsf{Adv}_{\mathbb{G}}^{\mathrm{ddh}}(\mathcal{D}) = |\Pr[\mathcal{D}(\mathbb{G}, g, g^x, g^y, g^{xy}) = 1] - \Pr[\mathcal{D}(\mathbb{G}, g, g^x, g^y, g^z) = 1]|$. We say that the DDH assumption holds in $\mathbb{G}$ if $\mathsf{Adv}_{\mathbb{G}}^{\mathrm{ddh}}(\mathcal{D})$ is negligible for all PPT algorithms $\mathcal{D}$. $\mathsf{Adv}_{\mathbb{G}}^{\mathrm{ddh}}(t)$ denotes the maximum value of $\mathsf{Adv}_{\mathbb{G}}^{\mathrm{ddh}}(\mathcal{D})$ over all algorithms $\mathcal{D}$ running in time at most t. A standard way of generating $\mathbb{G}$ where the DDH assumption is assumed to hold is to choose two primes p, q such that $p = rq + 1$ for some small $r \in \mathbb{N}$ (e.g., $r = 2$) and let $\mathbb{G}$ be the subgroup of order q in $\mathbb{Z}_p^*$.

[3]Due to space limitation, Wang, Hu and Li [24] provide only a quick sketch of the security proof for their NWPAKE-2 protocol.

Message authentication codes. A message authentication code (MAC) scheme Σ is a triple of efficient algorithms (Gen, Mac, Ver) where: (1) the key generation algorithm Gen takes as input a security parameter 1^ℓ and outputs a key k chosen uniformly at random from $\{0,1\}^\ell$; (2) the MAC generation algorithm Mac takes as input a key k and a message m, and outputs a MAC (also known as a tag) σ; and (3) the MAC verification algorithm Ver takes as input a key k, a message m, and a MAC σ, and outputs 1 if σ is valid for m under k or outputs 0 if σ is invalid. Let $\mathsf{Adv}_{\Sigma}^{\mathrm{suf-cma}}(\mathcal{A})$ be the advantage of an adversary $\mathcal{A}$ in violating the strong existential unforgeability of Σ under an adaptive chosen message attack. More precisely, $\mathsf{Adv}_{\Sigma}^{\mathrm{suf-cma}}(\mathcal{A})$ is the probability that an adversary $\mathcal{A}$, who mounts an adaptive chosen message attack against Σ with oracle access to $\mathsf{Mac}_k(\cdot)$ and $\mathsf{Ver}_k(\cdot)$, outputs a message/tag pair (m,σ) such that: (1) $\mathsf{Ver}_k(m,\sigma) = 1$ and (2) σ was not previously output by the oracle $\mathsf{Mac}_k(\cdot)$ as a MAC on the message m. We say that the MAC scheme Σ is secure if $\mathsf{Adv}_{\Sigma}^{\mathrm{suf-cma}}(\mathcal{A})$ is negligible for every PPT adversary $\mathcal{A}$. Let $\mathsf{Adv}_{\Sigma}^{\mathrm{suf-cma}}(t, q_{\mathrm{mac}}, q_{\mathrm{ver}})$ denotes the maximum value of $\mathsf{Adv}_{\Sigma}^{\mathrm{suf-cma}}(\mathcal{A})$ over all adversaries $\mathcal{A}$ running in time at most t and asking at most q_{mac} and q_{ver} queries to $\mathsf{Mac}_k(\cdot)$ and $\mathsf{Ver}_k(\cdot)$ respectively.

2-party PAKE protocols. 2R3PAKE takes as input a 2-party PAKE protocol 2PAKE. We assume that the given 2-party protocol 2PAKE outputs session keys distributed in $\{0,1\}^n$, where $n = 2\ell$, and is AKE-secure against an adversary who is given access to all the oracles: $\mathsf{Execute}$, Send, Reveal, $\mathsf{Corrupt}$ and Test. Let $\mathsf{Adv}_{\mathrm{2PAKE}}^{\mathrm{ake}}(\mathcal{A})$ be the advantage of an adversary $\mathcal{A}$ in breaking the AKE security of 2PAKE. We require that, for any PPT adversary $\mathcal{A}$ asking at most q_{send} Send queries, $\mathsf{Adv}_{\mathrm{2PAKE}}^{\mathrm{ake}}(\mathcal{A})$ is only negligibly larger than $q_{\mathrm{send}}/|\mathcal{D}|$. $\mathsf{Adv}_{\mathrm{2PAKE}}^{\mathrm{ake}}(t, Q)$ denotes the maximum value of $\mathsf{Adv}_{\mathrm{2PAKE}}^{\mathrm{ake}}(\mathcal{A})$ over all adversaries $\mathcal{A}$ with time complexity at most t and query complexity at most Q.

Symmetric encryption schemes. A symmetric encryption scheme Ω is a triple of efficient algorithms (Gen, Enc, Dec) where: (1) the key generation algorithm Gen takes as input a security parameter 1^ℓ and outputs a key k chosen uniformly at random from $\{0,1\}^\ell$; (2) the encryption algorithm Enc takes as input a key k and a plaintext message m, and outputs a ciphertext c; and (3) the decryption algorithm Dec takes as input a key k and a ciphertext c, and outputs a message m. We require that $\mathsf{Dec}_k(\mathsf{Enc}_k(m)) = m$ holds for all $k \in \{0,1\}^\ell$ and all $m \in \mathcal{M}$, where $\mathcal{M}$ is the plaintext space. For an eavesdropping adversary $\mathcal{A}$ against Ω and for a random bit $b \in_R \{0,1\}$, consider the following indistinguishability experiment:

```
Experiment $\mathbf{Exp}_{\Omega}^{\mathrm{ind-seav}}(\mathcal{A}, b)$
    $k \leftarrow \mathsf{Gen}(1^\ell)$
    $(m_0, m_1) \leftarrow \mathcal{A}(\Omega)$, where $|m_0| = |m_1|$
    $c \leftarrow \mathsf{Enc}_k(m_b)$
    $b' \leftarrow \mathcal{A}(c)$, where $b' \in \{0,1\}$
    return $b'$
```

For simplicity, we assume, in this experiment, that the security parameter 1^ℓ is implicit in the description of Ω. Let $\mathsf{Adv}_{\Omega}^{\mathrm{ind-seav}}(\mathcal{A})$ be the advantage of a single eavesdropper $\mathcal{A}$ in breaking the indistinguishability of Ω, and let it be defined as

$$\mathsf{Adv}_{\Omega}^{\mathrm{ind-seav}}(\mathcal{A}) = |\Pr[\mathbf{Exp}_{\Omega}^{\mathrm{ind-seav}}(\mathcal{A}, 0) = 1] - \Pr[\mathbf{Exp}_{\Omega}^{\mathrm{ind-seav}}(\mathcal{A}, 1) = 1]|.$$

We say that the symmetric encryption scheme Ω is secure (with respect to a single encryption) if $\mathsf{Adv}_{\Omega}^{\mathrm{ind-seav}}(\mathcal{A})$ is negligible for every PPT adversary $\mathcal{A}$. We use $\mathsf{Adv}_{\Omega}^{\mathrm{ind-seav}}(t)$ to denote the maximum value of $\mathsf{Adv}_{\Omega}^{\mathrm{ind-seav}}(\mathcal{A})$ over all adversaries $\mathcal{A}$ running in time at most t.

We now claim that if a symmetric encryption scheme is secure with respect to a single encryption, then it is also secure with respect to multiple encryptions under different keys. For an integer $n \geq 1$, consider the indistinguishability experiment below:

```
Experiment $\mathbf{Exp}_{\Omega}^{\mathrm{ind-meav}}(\mathcal{A}, b, n)$
    for i=1 to n
        $k_i \leftarrow \mathsf{Gen}(1^{\ell})$
        $(m_{0,i}, m_{1,i}) \leftarrow \mathcal{A}(\Omega)$, where $|m_{0,i}| = |m_{1,i}|$
        $c_i \leftarrow \mathsf{Enc}_{k_i}(m_{b,i})$
        $\mathcal{A}(c_i)$
    $b' \leftarrow \mathcal{A}$, where $b' \in \{0,1\}$
    return $b'$
```

Then we define $\mathsf{Adv}_{\Omega}^{\mathrm{ind-meav}}(\mathcal{A})$ and $\mathsf{Adv}_{\Omega}^{\mathrm{ind-meav}}(t)$ respectively as

$$\mathsf{Adv}_{\Omega}^{\mathrm{ind-meav}}(\mathcal{A}) = |\Pr[\mathbf{Exp}_{\Omega}^{\mathrm{ind-meav}}(\mathcal{A}, 0, n) = 1] - \Pr[\mathbf{Exp}_{\Omega}^{\mathrm{ind-meav}}(\mathcal{A}, 1, n) = 1]|$$

and

$$\mathsf{Adv}_{\Omega}^{\mathrm{ind-meav}}(t) = \max_{\mathcal{A}} \{\mathsf{Adv}_{\Omega}^{\mathrm{ind-meav}}(\mathcal{A})\},$$

where the maximum is over all $\mathcal{A}$ running in time at most t.

Lemma 1. *For any symmetric encryption scheme Ω,*

$$\mathsf{Adv}_{\Omega}^{\mathrm{ind-meav}}(t) \leq n \cdot \mathsf{Adv}_{\Omega}^{\mathrm{ind-seav}}(t),$$

where n is as defined for experiment $\mathbf{Exp}_{\Omega}^{\mathrm{ind-meav}}(\mathcal{A}, b, n)$.

Proof. Let $\mathcal{A}$ be a multiple eavesdropper attacking the indistinguishability of Ω, with advantage $\mathsf{Adv}_{\Omega}^{\mathrm{ind-meav}}(\mathcal{A})$ and time complexity t. The proof proceeds with a standard hybrid argument [49]. Consider a sequence of $n+1$ hybrid experiments $\mathbf{Exp}_{\Omega,\xi}^{\mathrm{ind-meav}}(\mathcal{A}, b, n)$, $0 \leq \xi \leq n$, where each $\mathbf{Exp}_{\Omega,\xi}^{\mathrm{ind-meav}}(\mathcal{A}, b, n)$ is different from $\mathbf{Exp}_{\Omega}^{\mathrm{ind-meav}}(\mathcal{A}, b, n)$ only in that each c_i is set as follows:

$$c_i \leftarrow \begin{cases} \mathsf{Enc}_{k_i}(m_{1,i}) & \text{if } i \leq \xi \\ \mathsf{Enc}_{k_i}(m_{0,i}) & \text{otherwise.} \end{cases}$$

The experiments $\mathbf{Exp}_{\Omega,0}^{\mathrm{ind-meav}}(\mathcal{A}, b, n)$ and $\mathbf{Exp}_{\Omega,n}^{\mathrm{ind-meav}}(\mathcal{A}, b, n)$ at the extremes of the sequence are identical to $\mathbf{Exp}_{\Omega}^{\mathrm{ind-meav}}(\mathcal{A}, 0, n)$ and $\mathbf{Exp}_{\Omega}^{\mathrm{ind-meav}}(\mathcal{A}, 1, n)$ respectively. As we move from $\mathbf{Exp}_{\Omega,\xi-1}^{\mathrm{ind-meav}}(\mathcal{A}, b, n)$ to $\mathbf{Exp}_{\Omega,\xi}^{\mathrm{ind-meav}}(\mathcal{A}, b, n)$ in the sequence, we change the ξ-th ciphertext c_{ξ} from the encryption of $m_{0,\xi}$ to the encryption of $m_{1,\xi}$. Since there are n such moves from $\mathbf{Exp}_{\Omega,0}^{\mathrm{ind-meav}}(\mathcal{A}, b, n)$ to $\mathbf{Exp}_{\Omega,n}^{\mathrm{ind-meav}}(\mathcal{A}, b, n)$, the inequality of the lemma follows immediately if we prove that the difference between the probabilities that $\mathcal{A}$ outputs 1 in any two neighboring experiments $\mathbf{Exp}_{\Omega,\xi-1}^{\mathrm{ind-meav}}(\mathcal{A}, b, n)$ and $\mathbf{Exp}_{\Omega,\xi}^{\mathrm{ind-meav}}(\mathcal{A}, b, n)$ is at most $\mathsf{Adv}_{\Omega}^{\mathrm{ind-seav}}(t)$. To complete the proof, it suffices to show that for any $1 \leq \xi \leq n$,

$$|\Pr[\mathbf{Exp}_{\Omega,\xi-1}^{\mathrm{ind-meav}}(\mathcal{A}, b, n) = 1] - \Pr[\mathbf{Exp}_{\Omega,\xi}^{\mathrm{ind-meav}}(\mathcal{A}, b, n) = 1]| \leq \mathsf{Adv}_{\Omega}^{\mathrm{ind-seav}}(t). \quad (1)$$

Let $\varepsilon = |\Pr[\mathbf{Exp}_{\Omega,\xi-1}^{\mathrm{ind-meav}}(\mathcal{A}, b, n) = 1] - \Pr[\mathbf{Exp}_{\Omega,\xi}^{\mathrm{ind-meav}}(\mathcal{A}, b, n) = 1]|$. We will prove Eq. (1) by constructing, from $\mathcal{A}$, a single eavesdropper $\mathcal{A}_{\xi}$ who breaks the indistinguishability of Ω with advantage ε and time complexity t.

$\mathcal{A}_\xi$ begins by invoking adversary $\mathcal{A}$, then proceeds to simulate the indistinguishability experiment for $\mathcal{A}$, and finally ends by outputting whatever bit $\mathcal{A}$ eventually outputs. In the simulated experiment, $\mathcal{A}_\xi$ generates the ciphertexts exactly as in the hybrid experiment $\mathbf{Exp}^{\text{ind-meav}}_{\Omega,\xi}(\mathcal{A}, b, n)$ except that it generates the ξ-th ciphertext c_ξ as follows:

> When $\mathcal{A}$ outputs the ξ-th plaintext-pair $(m_{0,\xi}, m_{1,\xi})$, $\mathcal{A}_\xi$ outputs this as its own plaintext-pair in experiment $\mathbf{Exp}^{\text{ind-seav}}_{\Omega}(\mathcal{A}_\xi, b)$, receives in return a ciphertext c, and sets $c_\xi = c$.

It follows that:

- The probability that $\mathcal{A}_\xi$ outputs 1 when the given ciphertext c is the encryption of $m_{0,\xi}$ is equal to the probability that $\mathcal{A}$ outputs 1 in the experiment $\mathbf{Exp}^{\text{ind-meav}}_{\Omega,\xi-1}(\mathcal{A}, b, n)$.
- The probability that $\mathcal{A}_\xi$ outputs 1 when the given ciphertext c is the encryption of $m_{1,\xi}$ is equal to the probability that $\mathcal{A}$ outputs 1 in the experiment $\mathbf{Exp}^{\text{ind-meav}}_{\Omega,\xi}(\mathcal{A}, b, n)$.

This means that:

$$\mathsf{Adv}^{\text{ind-seav}}_{\Omega}(\mathcal{A}) = |\Pr[\mathbf{Exp}^{\text{ind-meav}}_{\Omega,\xi-1}(\mathcal{A}, b, n) = 1] - \Pr[\mathbf{Exp}^{\text{ind-meav}}_{\Omega,\xi}(\mathcal{A}, b, n) = 1]|.$$

Since $\mathcal{A}_\xi$ has time complexity t, it follows that $\mathsf{Adv}^{\text{ind-seav}}_{\Omega}(\mathcal{A}) \leq \mathsf{Adv}^{\text{ind-seav}}_{\Omega}(t)$ by definition. This completes the proof of Eq. (1) and hence the proof of Lemma 1. □

4.2. The 2R3PAKE protocol

We assume that the following information has been pre-established and is known to all parties in the network: (1) a cyclic group $\mathbb{G}$ of prime order q, and a generator g of $\mathbb{G}$, (2) a MAC scheme $\Sigma = (\mathsf{Gen}, \mathsf{Mac}, \mathsf{Ver})$, (3) a 2-party PAKE protocol 2PAKE, and (4) a symmetric encryption scheme $\Omega = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. These public parameters can be determined by the server and broadcast to all registered clients. Let A and B be two clients who wish to establish a session key, and S be the trusted server with which A and B have registered their passwords pw_A and pw_B respectively. The partner identifier assigned to (an instance of) A (resp. B) is pid_A (resp. pid_B). Recall that pid is a sequence of identities of protocol participants; for simplicity, we assume that $pid_A = pid_B = (A, B, S)$. Our 2R3PAKE protocol is depicted in Fig. 2 and its description is as follows:

Step 1. A (resp. B) selects a random $x \in \mathbb{Z}_q^*$ (resp. $y \in \mathbb{Z}_q^*$), computes $X = g^x$ (resp. $Y = g^y$), and sends $\langle A, pid_A, X\rangle$ (resp. $\langle B, pid_B, Y\rangle$) to S.

Step 2. A and S establish a 2ℓ-bit key k_A by running the 2-party protocol 2PAKE. Likewise, B and S establish a 2ℓ-bit key k_B. Let $k_A = k_A^{enc} \| k_A^{mac}$ and $k_B = k_B^{enc} \| k_B^{mac}$.

Step 3. A computes $\sigma_A = \mathsf{Mac}_{k_A^{mac}}(A \| pid_A \| X)$ and sends $\langle A, \sigma_A\rangle$ to S. Similarly, B computes $\sigma_B = \mathsf{Mac}_{k_B^{mac}}(B \| pid_B \| Y)$ and sends $\langle B, \sigma_B\rangle$ to S.

Step 4. S sets $pid_S = pid_A$, chooses a random $z \in \mathbb{Z}_q^*$, and computes

$$\overline{X} = X^z, \quad \overline{Y} = Y^z,$$
$$\alpha_A = \mathsf{Enc}_{k_A^{enc}}(\overline{Y}), \quad \alpha_B = \mathsf{Enc}_{k_B^{enc}}(\overline{X}),$$
$$\rho_A = \mathsf{Mac}_{k_A^{mac}}(S \| pid_S \| \alpha_A \| \alpha_B), \quad \rho_B = \mathsf{Mac}_{k_B^{mac}}(S \| pid_S \| \alpha_A \| \alpha_B).$$

S then sends $\langle S, \alpha_A, \alpha_B, \rho_A\rangle$ and $\langle S, \alpha_A, \alpha_B, \rho_B\rangle$ to A and B respectively.

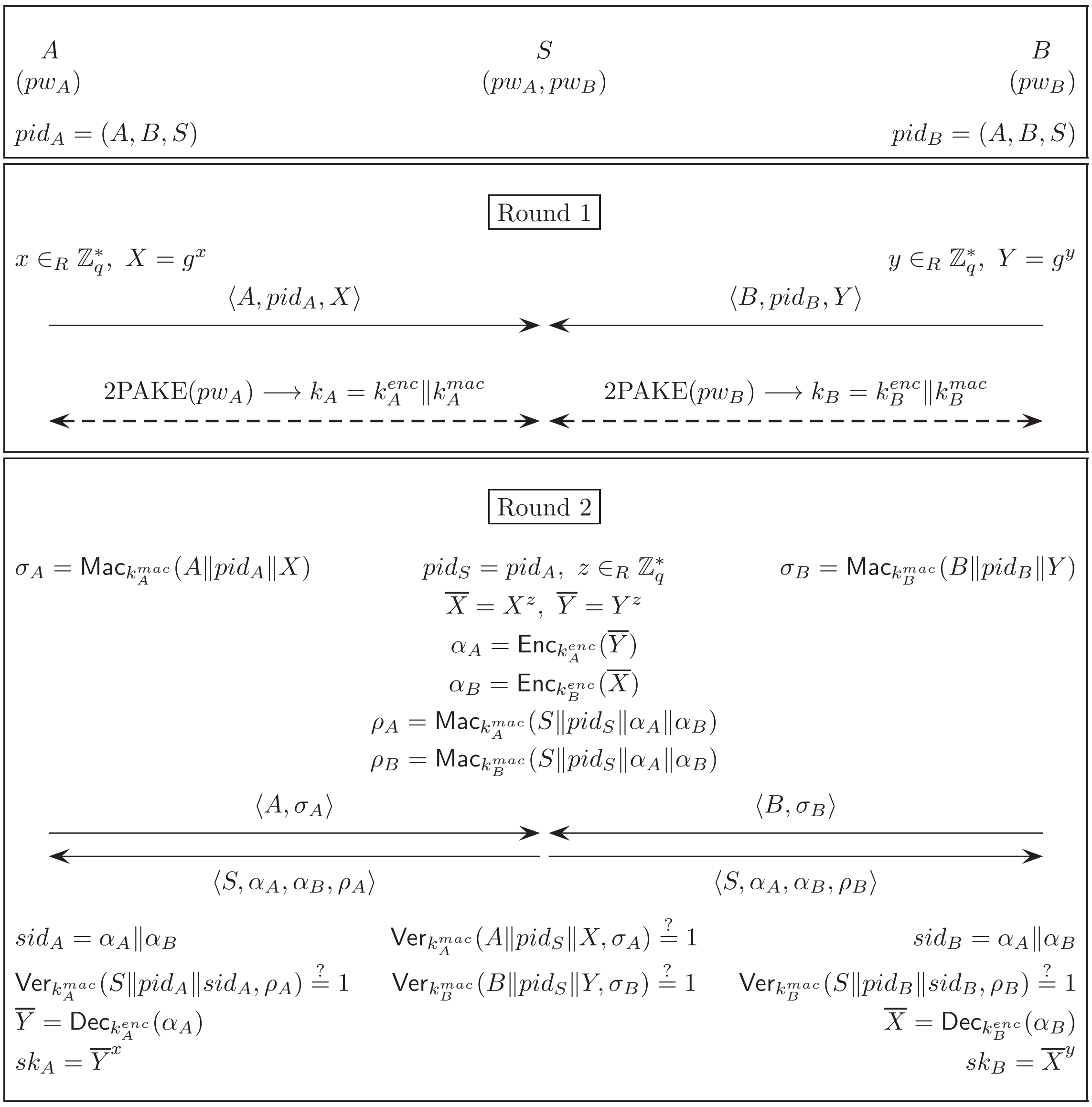
A
(pw_A)
S
(pw_A, pw_B)
B
(pw_B)
$pid_A = (A, B, S)$
$pid_B = (A, B, S)$
Round 1
$x \in_R \mathbb{Z}_q^*,\ X = g^x$
$y \in_R \mathbb{Z}_q^*,\ Y = g^y$
$\langle A, pid_A, X\rangle$
$\langle B, pid_B, Y\rangle$
$2\text{PAKE}(pw_A) \longrightarrow k_A = k_A^{enc} \| k_A^{mac}$
$2\text{PAKE}(pw_B) \longrightarrow k_B = k_B^{enc} \| k_B^{mac}$
Round 2
$\sigma_A = \mathsf{Mac}_{k_A^{mac}}(A \| pid_A \| X)$
$pid_S = pid_A,\ z \in_R \mathbb{Z}_q^*$
$\sigma_B = \mathsf{Mac}_{k_B^{mac}}(B \| pid_B \| Y)$
$\overline{X} = X^z,\ \overline{Y} = Y^z$
$\alpha_A = \mathsf{Enc}_{k_A^{enc}}(\overline{Y})$
$\alpha_B = \mathsf{Enc}_{k_B^{enc}}(\overline{X})$
$\rho_A = \mathsf{Mac}_{k_A^{mac}}(S \| pid_S \| \alpha_A \| \alpha_B)$
$\rho_B = \mathsf{Mac}_{k_B^{mac}}(S \| pid_S \| \alpha_A \| \alpha_B)$
$\langle A, \sigma_A\rangle$
$\langle B, \sigma_B\rangle$
$\langle S, \alpha_A, \alpha_B, \rho_A\rangle$
$\langle S, \alpha_A, \alpha_B, \rho_B\rangle$
$sid_A = \alpha_A \| \alpha_B$
$\mathsf{Ver}_{k_A^{mac}}(A \| pid_S \| X, \sigma_A) \stackrel{?}{=} 1$
$sid_B = \alpha_A \| \alpha_B$
$\mathsf{Ver}_{k_A^{mac}}(S \| pid_A \| sid_A, \rho_A) \stackrel{?}{=} 1$
$\mathsf{Ver}_{k_B^{mac}}(B \| pid_S \| Y, \sigma_B) \stackrel{?}{=} 1$
$\mathsf{Ver}_{k_B^{mac}}(S \| pid_B \| sid_B, \rho_B) \stackrel{?}{=} 1$
$\overline{Y} = \mathsf{Dec}_{k_A^{enc}}(\alpha_A)$
$\overline{X} = \mathsf{Dec}_{k_B^{enc}}(\alpha_B)$
$sk_A = \overline{Y}^x$
$sk_B = \overline{X}^y$

Fig. 2. Our proposed 2-round 3-party PAKE (2R3PAKE) protocol.

STEP 5. A sets the session identifier, $sid_A = \alpha_A \| \alpha_B$, and verifies that $\mathsf{Ver}_{k_A^{mac}}(S \| pid_A \| sid_A, \rho_A) = 1$. If the verification fails, A aborts the protocol. Otherwise, A recovers $\overline{Y}$ as $\overline{Y} = \mathsf{Dec}_{k_A^{enc}}(\alpha_A)$ and computes the session key, $sk_A = \overline{Y}^x$. B proceeds correspondingly; it aborts if $\mathsf{Ver}_{k_B^{mac}}(S \| pid_B \| sid_B, \rho_B) = 0$, where $sid_B = \alpha_A \| \alpha_B$, and otherwise, computes $\overline{X} = \mathsf{Dec}_{k_B^{enc}}(\alpha_B)$ and $sk_B = \overline{X}^y$.

STEP 6. S checks that $\mathsf{Ver}_{k_A^{mac}}(A \| pid_S \| X, \sigma_A) = 1$ and $\mathsf{Ver}_{k_B^{mac}}(B \| pid_S \| Y, \sigma_B) = 1$. If either of these checks fails, S aborts the protocol.

Steps 1 & 2 constitute the first round of communication while Steps 3 & 4 constitute the second communication round. It is trivial to note that in the presence of a passive adversary, A and B will compute session keys of the same value g^{xyz}. We do not require 2PAKE to be instantiated with a protocol that provides either unilateral or mutual authentication, as 2R3PAKE already provides mutual authentication between the server and the clients (via the

MAC values exchanged in the second round). Hence, any 2-party protocol that provides implicit key authentication, including one-round protocols, will be suitable candidates to instantiate 2PAKE.

4.3. Security proof

Theorem 1. *For any adversary with time complexity at most t and query complexity at most $Q = (q_{\text{exec}}, q_{\text{send}}, q_{\text{reve}}, q_{\text{corr}}, q_{\text{test}})$, its advantage in breaking the* AKE *security of 2R3PAKE is bounded by:*

$$\begin{aligned}\mathsf{Adv}^{\text{ake}}_{\text{2R3PAKE}}(t, Q) \leq\ & 2 \cdot \mathsf{Adv}^{\text{ake}}_{\text{2PAKE}}(t', Q') \\ & + 2 \cdot q_{\text{send}} \cdot \mathsf{Adv}^{\text{suf-cma}}_{\Sigma}(t', 2, 2) \\ & + 2 \cdot q_{\text{send}} \cdot \mathsf{Adv}^{\text{ind-seav}}_{\Omega}(t') \\ & + 2 \cdot \mathsf{Adv}^{\text{ddh}}_{\mathbb{G}}(t'),\end{aligned}$$

where $Q' = (2q_{\text{exec}}, q_{\text{send}}, q_{\text{send}}, q_{\text{corr}}, 2q_{\text{exec}} + q_{\text{send}})$ and t' is the maximum time required to perform the experiment $\mathbf{Exp}_0$ *involving an adversary who attacks 2R3PAKE with time complexity* t.

Proof. Let $\mathcal{A}$ be a PPT adversary who attacks the AKE security of 2R3PAKE with time complexity t and query complexity $Q = (q_{\text{exec}}, q_{\text{send}}, q_{\text{reve}}, q_{\text{corr}}, q_{\text{test}})$. We prove the theorem by making a series of modifications to the experiment $\mathbf{Exp}_0$, bounding the difference in $\mathcal{A}$'s success probability between two consecutive (modified) experiments, and ending up with an experiment in which $\mathcal{A}$ has a success probability of 1/2 (i.e., $\mathcal{A}$ has no advantage). By Succ_i, we denote the event that $\mathcal{A}$ correctly guesses the hidden bit b in experiment $\mathbf{Exp}_i$.

Before presenting the first modified experiment, we define the notion of a *clean* instance.

Definition 3. We say an instance Π^i_U is unclean if $\mathcal{A}$ has queried $\mathsf{Corrupt}(U')$ for some $U' \in pid^i_U$. Otherwise, we say it is clean.

Experiment $\mathbf{Exp}_1$. We modify the experiment by replacing each different 2ℓ-bit key (established by an execution of 2PAKE) with a random key drawn uniformly from $\{0,1\}^{2\ell}$ for all clean instances. The difference in $\mathcal{A}$'s success probability between $\mathbf{Exp}_0$ and $\mathbf{Exp}_1$ is bounded by:

Claim 1. $\left|\Pr_{\text{2R3PAKE},\mathcal{A}}[\mathsf{Succ}_1] - \Pr_{\text{2R3PAKE},\mathcal{A}}[\mathsf{Succ}_0]\right| \leq \mathsf{Adv}^{\text{ake}}_{\text{2PAKE}}(t', Q')$.

Proof. We prove the claim by constructing an adversary $\mathcal{A}'$ who attacks the AKE security of 2PAKE with advantage equal to $\left|\Pr_{\text{2R3PAKE},\mathcal{A}}[\mathsf{Succ}_1] - \Pr_{\text{2R3PAKE},\mathcal{A}}[\mathsf{Succ}_0]\right|$. Let k^i_U denotes the 2ℓ-bit key held by instance Π^i_U.

$\mathcal{A}'$ chooses a random bit $b \in \{0,1\}$ and invokes the adversary $\mathcal{A}$. $\mathcal{A}'$ then simulates the oracles for $\mathcal{A}$ as follows:

$\mathsf{Execute}$ queries. When an $\mathsf{Execute}(\Pi^i_A, \Pi^j_B, \Pi^k_S)$ query is asked, $\mathcal{A}'$ first checks if A, B or S was previously corrupted.

- If so, $\mathcal{A}'$ answers the $\mathsf{Execute}$ query as in experiment $\mathbf{Exp}_0$.
- Otherwise, $\mathcal{A}'$ answers the query using its own oracles. $\mathcal{A}'$ first asks two queries $\mathsf{Execute}(\Pi^i_A, \Pi^k_S)$ and $\mathsf{Execute}(\Pi^j_B, \Pi^{k'}_S)$. Let $\mathsf{T}_{\text{2PAKE}}$ and $\mathsf{T}'_{\text{2PAKE}}$ be two transcripts returned in response to the $\mathsf{Execute}$ queries. Next, $\mathcal{A}'$ makes the queries $\mathsf{Test}(\Pi^i_A)$ and $\mathsf{Test}(\Pi^j_B)$, and receives in return two keys $\overline{k}^i_A$ and $\overline{k}^j_B$ (either real or random). $\mathcal{A}'$ then generates the rest of the protocol messages, using $\overline{k}^i_A$ and $\overline{k}^j_B$ as k^i_A and k^j_B, respectively. Finally, $\mathcal{A}'$ returns these messages together with $\mathsf{T}_{\text{2PAKE}}$ and $\mathsf{T}'_{\text{2PAKE}}$ after ordering them properly.

Send queries. For each $\mathsf{Send}(\Pi_U^i, m)$ query, $\mathcal{A}'$ checks if m is a message for initiating a new session (of 2R3PAKE), or the Send query belongs to an execution of 2PAKE.

1. If both are untrue, $\mathcal{A}'$ responds to the query as in experiment $\mathbf{Exp}_0$.
2. Otherwise, $\mathcal{A}'$ answers it by making the same query to its own Send oracle. If the query prompts Π_U^i to accept, then $\mathcal{A}'$ checks if Π_U^i is clean.
   (a) If so, $\mathcal{A}'$ makes a $\mathsf{Test}(\Pi_U^i)$ query (unless the partner of Π_U^i has already been tested) and sets k_U^i to be the output of this Test query.
   (b) Otherwise, $\mathcal{A}'$ makes a $\mathsf{Reveal}(\Pi_U^i)$ query and sets k_U^i to be the output of this Reveal query.

Reveal queries. $\mathcal{A}'$ responds to the queries as per protocol specification.

$\mathsf{Corrupt}$ queries. $\mathcal{A}'$ answers these queries using its own $\mathsf{Corrupt}$ oracle.

Test queries. $\mathcal{A}'$ responds to these queries based on the ranomdly chosen bit b at the beginning of the simulation. $\mathcal{A}'$ will return the real session key if $b = 1$, and a random key chosen uniformly at random from $\mathbb{G}$ if $b = 0$.

At some point in time, $\mathcal{A}$ will terminate and output its guess b'. When this happens, $\mathcal{A}'$ outputs 1 if $b = b'$, and 0 otherwise.

From the simulation, it is clear that:

- The probability that $\mathcal{A}'$ outputs 1 when its Test oracle returns real session keys is equal to the probability that $\mathcal{A}$ correctly guesses the bit b in experiment $\mathbf{Exp}_0$.
- The probability that $\mathcal{A}'$ outputs 1 when its Test oracle returns random keys is equal to the probability that $\mathcal{A}$ correctly guesses the bit b in experiment $\mathbf{Exp}_1$.

That is, $\mathsf{Adv}^{\mathrm{ake}}_{\mathrm{2PAKE}}(\mathcal{A}') = \left|\Pr_{\mathrm{2R3PAKE},\mathcal{A}}[\mathsf{Succ}_1] - \Pr_{\mathrm{2R3PAKE},\mathcal{A}}[\mathsf{Succ}_0]\right|$. Since $\mathcal{A}'$ has at most time complexity t' and query complexity $Q' = (2q_{\mathrm{exec}}, q_{\mathrm{send}}, q_{\mathrm{send}}, q_{\mathrm{corr}}, 2q_{\mathrm{exec}} + q_{\mathrm{send}})$, it follows, by definition, that $\mathsf{Adv}^{\mathrm{ake}}_{\mathrm{2PAKE}}(\mathcal{A}') \leq \mathsf{Adv}^{\mathrm{ake}}_{\mathrm{2PAKE}}(t', Q')$. This completes the proof of Claim 1. $\square$

Experiment $\mathbf{Exp}_2$. This experiment is different from $\mathbf{Exp}_1$ only in that it is aborted and the adversary does not succeed if the following event Forge occurs.

Forge**:** The event that the adversary $\mathcal{A}$ makes a Send query of the form $\mathsf{Send}(\Pi_U^i, V\|msg)$ for uncorrupted U and V such that msg contains a MAC forgery.

Then we have:

Claim 2. $\left|\Pr_{\mathrm{2R3PAKE},\mathcal{A}}[\mathsf{Succ}_2] - \Pr_{\mathrm{2R3PAKE},\mathcal{A}}[\mathsf{Succ}_1]\right| \leq q_{\mathrm{send}} \cdot \mathsf{Adv}^{\mathrm{suf-cma}}_{\Sigma}(t', 2, 2)$.

PROOF. Assuming that the event Forge occurs, we construct an algorithm $\mathcal{F}$ who outputs, with a non-negligible probability, a forgery against the MAC scheme Σ. The algorithm $\mathcal{F}$ is given oracle access to $\mathsf{Mac}_k(\cdot)$ and $\mathsf{Ver}_k(\cdot)$. The goal of $\mathcal{F}$ is to produce a message/tag pair (m, σ) such that: (1) $\mathsf{Ver}_k(m, \sigma) = 1$ and (2) σ was not previously output by the $\mathsf{Mac}_k(\cdot)$ oracle on input m.

Let n be the number of all different MAC keys established via a Send query made by $\mathcal{A}$. Clearly, $n \leq q_{\mathrm{send}}$. $\mathcal{F}$ begins by choosing a random $\alpha \in \{1, \ldots, n\}$. Let k_α^{mac} denote the α^{th} key among all the n MAC keys, and Send_α be a Send query that should be answered and/or verified using k_α^{mac}. $\mathcal{F}$ invokes $\mathcal{A}$ as a subroutine and handles the oracle calls of $\mathcal{A}$ as in experiment

$\mathbf{Exp}_1$ except that: it answers all Send_α queries by accessing its MAC generation and verification oracles. As a result, the α^{th} MAC key k_α^{mac} is never used during the simulation. If Forge occurs against an instance who holds k_α^{mac}, $\mathcal{F}$ halts and outputs the message/tag pair generated by $\mathcal{A}$ as its forgery. Otherwise, $\mathcal{F}$ halts and outputs a failure indication.

If the guess α is correct, then the simulation is perfect and $\mathcal{F}$ achieves its goal. Namely, $\mathsf{Adv}_\Sigma^{\mathrm{suf-cma}}(\mathcal{F}) = \Pr[\mathsf{Forge}]/n$. Since $n \leq q_{\text{send}}$, we get $\Pr[\mathsf{Forge}] \leq q_{\text{send}} \cdot \mathsf{Adv}_\Sigma^{\mathrm{suf-cma}}(\mathcal{F})$. As $\mathcal{F}$ has at most time complexity t' and makes at most two queries to $\mathsf{Mac}_k(\cdot)$ and $\mathsf{Ver}_k(\cdot)$, it follows, by definition, that $\mathsf{Adv}_\Sigma^{\mathrm{suf-cma}}(\mathcal{F}) \leq \mathsf{Adv}_\Sigma^{\mathrm{suf-cma}}(t', 2, 2)$. This completes the proof of Claim 2. $\square$

Experiment $\mathbf{Exp}_3$. We further modify the experiment so that $\mathsf{Execute}$ and Send oracles are simulated as in "the $\mathbf{Exp}_3$ modification" described below.

<u>The $\mathbf{Exp}_3$ modification</u>

When $\mathcal{A}$ asks an $\mathsf{Execute}$ or Send query, the simulator answers it exactly as in experiment $\mathbf{Exp}_2$ except that it modifies the way of generating the ephemeral public values (denoted as X and Y in the protocol) as follows:

- The simulator chooses two random $v_1, v_2 \in \mathbb{Z}_q^*$ and computes $V_1 = g^{v_1}$ and $V_2 = g^{v_2}$.
- For each instance Π_C^i, the simulator chooses a random $r \in \mathbb{Z}_q^*$, computes
$$R = \begin{cases} {V_1}^r & \text{if } C \text{ appears first in } pid_C^i \\ {V_2}^r & \text{if } C \text{ appears second in } pid_C^i, \end{cases}$$
and uses R as the ephemeral public value (i.e., as X or Y) of Π_C^i.

Since the view of $\mathcal{A}$ is identical between $\mathbf{Exp}_2$ and $\mathbf{Exp}_3$, it is straightforward to see that:

Claim 3. $\Pr_{\text{2R3PAKE},\mathcal{A}}[\mathsf{Succ}_3] = \Pr_{\text{2R3PAKE},\mathcal{A}}[\mathsf{Succ}_2]$.

Experiment $\mathbf{Exp}_4$. In this experiment, each $\overline{X}$ and $\overline{Y}$ are computed as $\overline{X} = X$ and $\overline{Y} = Y$ (instead of as $\overline{X} = X^z$ and $\overline{Y} = Y^z$) if they are expected to be encrypted with a key held by a clean (server) instance. This is the only difference from $\mathbf{Exp}_3$.

Claim 4. $\left|\Pr_{\text{2R3PAKE},\mathcal{A}}[\mathsf{Succ}_4] - \Pr_{\text{2R3PAKE},\mathcal{A}}[\mathsf{Succ}_3]\right| \leq q_{\text{send}} \cdot \mathsf{Adv}_\Omega^{\mathrm{ind-seav}}(t')$.

Proof. We prove the claim by constructing a multiple eavesdropper $\mathcal{A}_{\text{meav}}$ who attacks the indistinguishability of Ω with advantage equal to $\left|\Pr_{\text{2R3PAKE},\mathcal{A}}[\mathsf{Succ}_4] - \Pr_{\text{2R3PAKE},\mathcal{A}}[\mathsf{Succ}_3]\right|$.

$\mathcal{A}_{\text{meav}}$ chooses a random bit $b \in \{0,1\}$ and invokes the adversary $\mathcal{A}$. $\mathcal{A}_{\text{meav}}$ then handles all the oracle queries of $\mathcal{A}$ as in experiment $\mathbf{Exp}_3$ except that it generates α_A and α_B for each clean server instance as follows:

> $\mathcal{A}_{\text{meav}}$ outputs $(X, \overline{X} = X^z)$ and $(Y, \overline{Y} = Y^z)$ as its own (two) plaintext-pairs (in the indistinguishability experiment $\mathbf{Exp}_\Omega^{\mathrm{ind-meav}}$), receives in return two ciphertexts c_1 and c_2, and sets $\alpha_A = c_2$ and $\alpha_B = c_1$. (Note, here, that c_1 and c_2 are encryptions of either X and Y or $\overline{X}$ and $\overline{Y}$.)

When $\mathcal{A}$ outputs its guess b', $\mathcal{A}_{\text{meav}}$ outputs 1 if $b = b'$, and 0 otherwise. It easily follows that:

- The probability that $\mathcal{A}_{\text{meav}}$ outputs 1 when the first plaintexts are encrypted in experiment $\mathbf{Exp}_{\Omega}^{\text{ind-meav}}$ is equal to the probability that $\mathcal{A}$ succeeds in experiment $\mathbf{Exp}_4$.

- The probability that $\mathcal{A}_{\text{meav}}$ outputs 1 when the second plaintexts are encrypted in experiment $\mathbf{Exp}_{\Omega}^{\text{ind-meav}}$ is equal to the probability that $\mathcal{A}$ succeeds in experiment $\mathbf{Exp}_3$.

Therefore, $\mathsf{Adv}_{\Omega}^{\text{ind-meav}}(\mathcal{A}_{\text{meav}}) = \left|\Pr_{\text{2R3PAKE},\mathcal{A}}[\mathsf{Succ}_4] - \Pr_{\text{2R3PAKE},\mathcal{A}}[\mathsf{Succ}_3]\right|$. Since $\mathcal{A}_{\text{meav}}$ eavesdrops at most q_{send} encryptions and has time complexity at most t', Claim 4 follows immediately from Lemma 1 of Section 4.1. □

Experiment $\mathbf{Exp}_5$. We now modify the way session keys are computed. For each clean instance and its partner instance, the shared session key is chosen uniformly at random from $\mathbb{G}$.

Claim 5. $\left|\Pr_{\text{2R3PAKE},\mathcal{A}}[\mathsf{Succ}_5] - \Pr_{\text{2R3PAKE},\mathcal{A}}[\mathsf{Succ}_4]\right| \leq \mathsf{Adv}_{\mathbb{G}}^{\text{ddh}}(t')$.

Proof. We prove the claim by constructing an algorithm $\mathcal{A}_{\text{ddh}}$ that solves the DDH problem in $\mathbb{G}$ with advantage equal to $\left|\Pr_{\text{2R3PAKE},\mathcal{A}}[\mathsf{Succ}_5] - \Pr_{\text{2R3PAKE},\mathcal{A}}[\mathsf{Succ}_4]\right|$.

On input a DDH-problem instance $(W_1 = g^{w_1}, W_2 = g^{w_2}, W_3) \in \mathbb{G}^3$, $\mathcal{A}_{\text{ddh}}$ chooses a random bit $b \in \{0, 1\}$, invokes the adversary $\mathcal{A}$, and simulates the oracles on its own. $\mathcal{A}_{\text{ddh}}$ handles all the queries of $\mathcal{A}$ as in experiment $\mathbf{Exp}_4$ except for the following:

- $\mathcal{A}_{\text{ddh}}$ uses W_1 and W_2 in place of V_1 and V_2 (see "the $\mathbf{Exp}_3$ modification").

- For each clean instance Π_C^i who sends $X = {W_1}^r$ and receives $Y = {W_2}^{r'}$, or vice versa, $\mathcal{A}_{\text{ddh}}$ sets the session key sk_C^i to be $W_3^{rr'}$.

Later, when $\mathcal{A}$ outputs its guess b', $\mathcal{A}_{\text{ddh}}$ outputs 1 if $b = b'$, and 0 otherwise.

The simulation above clearly shows that:

- The probability that $\mathcal{A}_{\text{ddh}}$ outputs 1 on a true Diffie-Hellman triple is equal to the probability that $\mathcal{A}$ correctly guesses the bit b in experiment $\mathbf{Exp}_4$.

- The probability that $\mathcal{A}_{\text{ddh}}$ outputs 1 on a random triple is equal to the probability that $\mathcal{A}$ correctly guesses the bit b in experiment $\mathbf{Exp}_5$.

Hence, $\mathsf{Adv}_{\mathbb{G}}^{\text{ddh}}(\mathcal{A}_{\text{ddh}}) = \left|\Pr_{\text{2R3PAKE},\mathcal{A}}[\mathsf{Succ}_5] - \Pr_{\text{2R3PAKE},\mathcal{A}}[\mathsf{Succ}_4]\right|$. From this and since $\mathsf{Adv}_{\mathbb{G}}^{\text{ddh}}(\mathcal{A}_{\text{ddh}}) \leq \mathsf{Adv}_{\mathbb{G}}^{\text{ddh}}(t')$, we obtain the inequality of Claim 5. □

In experiment $\mathbf{Exp}_5$, the session keys of all fresh instances are chosen uniformly at random from $\mathbb{G}$ and thus the adversary $\mathcal{A}$ obtains no information on the bit b chosen by the Test oracle. Therefore, it follows that $\Pr[\mathsf{Succ}_5] = 1/2$. This result combined with Claims 1–5 yields the statement of Theorem 1. ■

5. Concluding remarks

In this paper, we have proposed an efficient and secure 3-party password-only authenticated key exchange protocol that requires only two communication rounds. We have rigorously proved the security of the protocol in a widely accepted adversary model. Since our proof of security requires no idealizing assumptions, our proposed protocol would be considered equivalent to be provably secure in the standard model as long as the building blocks are also instantiated with schemes proven secure in the standard model. For a more efficient implementation of our proposed protocol, Steps 3 & 6 (see the protocol description in Section 4.2) can be omitted if security against undetectable online dictionary attacks is not required. This simplified protocol would still be AKE-secure in the sense of Definition 2 (i.e. Theorem 1 also holds for the simplified protocol). We finally note that it seems impossible to design an AKE-secure one-round key exchange protocol in the password-only 3-party setting, although we are unable to prove this statement formally.

References

[1] C. Hervey, P. van Oorschot, A research agenda acknowledging the persistence of passwords, IEEE Security & Privacy 10(1): 28–36, 2012.

[2] W. Wang, L. Hu, Efficient and provably secure generic construction of three-party password-based authenticated key exchange protocols, In Proceedings of INDOCRYPT 2006, LNCS 4329: 118–132, 2006.

[3] J. Nam, J. Paik, H. Kang, U. Kim, D. Won, An off-line dictionary attack on a simple three-party key exchange protocol, IEEE Communications Letters 13(3): 205–207, 2009.

[4] N. Lo, K. Yeh, Cryptanalysis of two three-party encrypted key exchange protocols, Computer Standards & Interfaces 31(6): 1167–1174, 2009.

[5] C. Lin, T. Hwang, On 'a simple three-party password-based key exchange protocol', International Journal of Communication Systems 24(11): 1520–1532, 2011.

[6] S. Wu, Q. Pu, S. Wang, D. He, Cryptanalysis of a communication-efficient three-party password authenticated key exchange protocol, Information Sciences 215: 83–96, 2012.

[7] KKR. Choo, C. Boyd, Y. Hitchcock, Errors in computational complexity proofs for protocols, In Proceedings of ASIACRYPT 2005, LNCS 3788: 624–643, 2005.

[8] KKR. Choo, C. Boyd, Y. Hitchcock, Examining indistinguishability-based proof models for key establishment protocols, In Proceedings of ASIACRYPT 2005, LNCS 3788: 585–604, 2005.

[9] KKR. Choo, C. Boyd, Y. Hitchcock, The importance of proofs of security for key establishment protocols: Formal analysis of Jan-Chen, Yang-Shen-Shieh, Kim-Huh-Hwang-Lee, Lin-Sun-Hwang, and Yeh-Sun protocols, Computer Communications 29: 2788–2797, 2006.

[10] M. Bellare, P. Rogaway, Entity authentication and key distribution, In Proceedings of CRYPTO 1993, LNCS 773: 232–249, 1994.

[11] M. Abdalla, P. Fouque, D. Pointcheval, Password-based authenticated key exchange in the three-party setting, In Proceedings of PKC 2005, LNCS 3386: 65–84, 2005.

[12] M. Abdalla, P. Fouque, D. Pointcheval, Password-based authenticated key exchange in the three-party setting, IEE Proceedings-Information Security 153(1): 27–39, 2006.

[13] C. Lin, H. Sun, M. Steiner, T. Hwang, Three-party encrypted key exchange without server public-keys, IEEE Communications Letters 5(12): 497–499, 2001.

[14] T. Lee, T. Hwang, C. Lin, Enhanced three-party encrypted key exchange without server public keys, Computers & Security 23(7): 571–577, 2004.

[15] M. Abdalla, D. Pointcheval, Interactive Diffie-Hellman assumptions with applications to password-based authentication, In Proceedings of FC 2005, LNCS 3570: 341–356, 2005.

[16] H. Wen, T. Lee, T. Hwang, Provably secure three-party password-based authenticated key exchange protocol using Weil pairing, IEE Proceedings-Communications 152(2): 138–143, 2005.

[17] R. Lu, Z. Cao, Simple three-party key exchange protocol, Computers & Security 26(1): 94–97, 2007.

[18] H. Chung, W. Ku, Three weaknesses in a simple three-party key exchange protocol, Information Sciences 178(1): 220–229, 2008.

[19] H. Guo, Z. Li, Y. Mu, X. Zhang, Cryptanalysis of simple three-party key exchange protocol, Computers & Security 27(1): 16–21, 2008.

[20] H. Kim, J. Choi, Enhanced password-based simple three-party key exchange protocol, Computers and Electrical Engineering 35(1): 107–114, 2009.

[21] H. Huang, A simple three-party password-based key exchange protocol, International Journal of Communication Systems 22(7): 857–862, 2009.

[22] E. Dongna, Q. Cheng, C. Ma, Password authenticated key exchange based on RSA in the three-party settings, In Proceedings of ProvSec 2009, LNCS 5848: 168–182, 2009.

[23] T. Lee, T. Hwang, Simple password-based three-party authenticated key exchange without server public keys, Information Sciences 180(9): 1702–1714, 2010.

[24] W. Wang, L. Hu, Y. Li, How to construct secure and efficient three-party password-based authenticated key exchange protocols, In Proceedings of INSCRYPT 2010, LNCS 6584: 218–235, 2011.

[25] T. Chang, M. Hwang, W. Yang, A communication-efficient three-party password authenticated key exchange protocol, Information Sciences 181(1): 217–226, 2011.

[26] C. Lee, S. Chen, C. Chen, A computation-efficient three-party encrypted key exchange protocol, Applied Mathematics & Information Sciences 6(3): 573–579, 2012.

[27] J. Nam, Y. Lee, S. Kim, D. Won, Security weakness in a three-party pairing-based protocol for password authenticated key exchange, Information Sciences 177(6): 1364–1375, 2007.

[28] R. Phan, W. Yau, B. Goi, Cryptanalysis of simple three-party key exchange protocol (S-3PAKE), Information Sciences 178(13): 2849–2856, 2008.

[29] E. Yoon, K. Yoo, Cryptanalysis of a simple three-party password-based key exchange protocol, International Journal of Communication Systems 24(4): 532–542, 2011.

[30] H. Liang, J. Hu, S. Wu, Re-attack on a three-party password-based authenticated key exchange protocol, Mathematical and Computer Modelling 57(5–6): 1175–1183, 2013.

[31] H. Tsai, C. Chang, Provably secure three party encrypted key exchange scheme with explicit authentication, Information Sciences 238: 242–249, 2013.

[32] J. Nam, KKR. Choo, J. Paik, D. Won, On the security of a password-only authenticated three-party key exchange protocol, Cryptology ePrint Archive: Report 2013/540, 2013.

[33] J. Nam, KKR. Choo, J. Paik, D. Won, An offline dictionary attack against a three-party key exchange protocol, Cryptology ePrint Archive: Report 2013/666, 2013.

[34] M. Szydlo, A note on Chosen-Basis Decisional Diffie-Hellman assumptions, In Proceedings of FC 2006, LNCS 4107: 166–170, 2006.

[35] K. Yoneyama, Efficient and strongly secure password-based server aided key exchange, In Proceedings of INDOCRYPT 2008, LNCS 5365: 172–184, 2008.

[36] J. Zhao, D. Gu, Provably secure three-party password-based authenticated key exchange protocol, Information Sciences 184(1): 310–323, 2012.

[37] C. Lin, H. Sun, T. Hwang, Three-party encrypted key exchange: attacks and a solution, ACM SIGOPS Operating Systems Review 34(4): 12–20, 2000.

[38] C. Chang, Y. Chang, A novel three-party encrypted key exchange protocol, Computer Standards & Interfaces 26(5): 471–476, 2004.

[39] H. Chen, T. Chen, W. Lee, C. Chang, Security enhancement for a three-party encrypted key exchange protocol against undetectable on-line password guessing attacks, Computer Standards & Interfaces 30(1–2): 95–99, 2008.

[40] E. Yoon, K. Yoo, Improving the novel three-party encrypted key exchange protocol, Computer Standards & Interfaces 30(5): 309–314, 2008.

[41] H. Chien, T. Wu, Provably secure password-based three-party key exchange with optimal message steps, The Computer Journal 52(6): 646–655, 2009.

[42] D. Lou, H. Huang, Efficient three-party password-based key exchange scheme, International Journal of Communication Systems 24(4): 504–512, 2011.

[43] J. Yang, T. Cao, Provably secure three-party password authenticated key exchange protocol in the standard model, Journal of Systems and Software 85(2): 340–350, 2012.

[44] S. Wu, K. Chen, Q. Pu, Y. Zhu, Cryptanalysis and enhancements of efficient three-party password-based key exchange scheme, International Journal of Communication Systems 26(5): 674–686, 2013.

[45] M. Bellare, D. Pointcheval, P. Rogaway, Authenticated key exchange secure against dictionary attacks, In Proceedings of EUROCRYPT 2000, LNCS 1807: 139–155, 2000.

[46] M. Abdalla, D. Pointcheval, Simple password-based encrypted key exchange protocols, In Proceedings of CT-RSA 2005, LNCS 3376: 191–208, 2005.

[47] J. Katz, V. Vaikuntanathan, Round-optimal password-based authenticated key exchange, In Proceedings of TCC 2011, LNCS 6597: 293–310, 2011.

[48] KKR. Choo, A proof of revised Yahalom protocol in the Bellare and Rogaway (1993) model, The Computer Journal 50(5): 591–601, 2007.

[49] S. Goldwasser, S. Micali, Probabilistic encryption, Journal of Computer and System Sciences 28(2): 270–299, 1984.