SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

February 2020, Vol. 63 122101:1–122101:15
https://doi.org/10.1007/s11432-019-9907-x

# Intersection-policy private mutual authentication from authorized private set intersection

Yamin WEN[1,2], Fangguo ZHANG[3,4], Huaxiong WANG[5], Yinbin MIAO[2,6] & Zheng GONG[7*]

[1]*School of Statistics and Mathematics, Guangdong University of Finance and Economics, Guangzhou* 510320, *China;*
[2]*State Key Laboratory of Cryptology, P. O. Box* 5159, *Beijing* 100878, *China;*
[3]*School of Data and Computer Science, Sun Yat-sen University, Guangzhou* 510006, *China;*
[4]*Guangdong Key Laboratory of Information Security, Sun Yat-sen University, Guangzhou* 510006, *China;*
[5]*School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore* 637371, *Singapore;*
[6]*Department of Cyber Engineering, Xidian University, Xi'an* 710071, *China;*
[7]*School of Computer, South China Normal University, Guangzhou* 510631, *China*

Received 25 March 2019/Accepted 28 May 2019/Published online 16 January 2020

**Abstract** Private mutual authentication (PMA) enables two-way anonymous authentication between two users certified by the same trusted group authority. Most existing PMA schemes focus on acquiring a relatively onefold authentication policy that ensures affiliation-hiding or designated single-attribute matching. However, in practice, users are typically provided with multiple attributes. In addition to the affiliation-hiding requirement, how to effectively achieve a more flexible authentication policy for multi-attribute applications remains a challenging issue. The intersection policy for authentication is also required when the attribute intersection is not an empty set or its cardinality is no less than a threshold value. To solve the above problems, we first propose an optimal authorized private set intersection protocol with forward security based on identity-based encryption and then design a new PMA protocol with intersection-policy called IP-PMA, which provides a simple solution for secret handshakes between two members (holding multiple attributes) from the same organization. Formal security analyses proved that our two proposed protocols are secure in the random oracle model. Empirical tests demonstrated that the IP-PMA protocol is optimized with linear complexity and may be more suitable for resource-constrained applications.

**Keywords** private mutual authentication, secret handshake, intersection policy, authorized private set intersection, multi-attribute matching.

**Citation** Wen Y M, Zhang F G, Wang H X, et al. Intersection-policy private mutual authentication from authorized private set intersection. Sci China Inf Sci, 2020, 63(2): 122101, https://doi.org/10.1007/s11432-019-9907-x

## 1 Introduction

The design of privacy-preserving cryptographic protocols mainly relies on the level of user protection. A typical anonymous credential mechanism allows the verifier to only identify the prover's affiliation without knowing his/her real identity. However, for security-critical applications, affiliations should not be exposed to outsiders for full privacy concerns. Thus, the private mutual authentication (PMA) protocol (i.e., unlinkable secret handshake scheme) [1] is generally used to solve such affiliation-hiding problem. The anonymous credential system typically acts as a unidirectional authentication component to further protect individual identities, and the secret handshake mechanism serves as a bidirectional private authentication component to guarantee that two participants are affiliated by the same group. Such two-way protocols ensure the mutual authenticity of the communication parties that either participant leaks no

* Corresponding author (email: cis.gong@gmail.com)

 

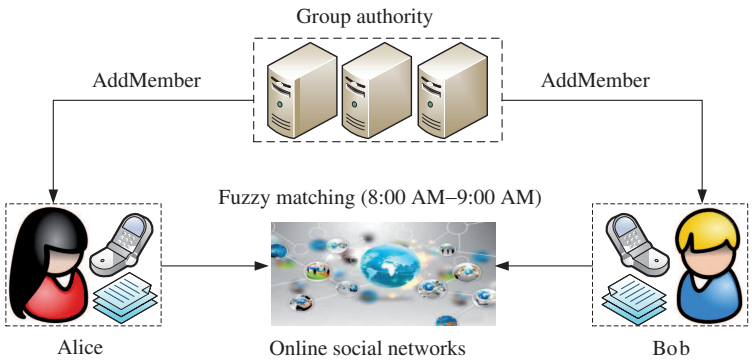


**Figure 1** (Color online) Illustration of the PMA scenario.

sensitive information to the opposite party who does not satisfy the authentication policy. Furthermore, a trusted third-party for each independent group called group authority (GA) is required that signs and distributes group credentials for its members. Thus, certified group members from the same group can perform a successful PMA protocol. Note that a group in this context refers to a social organization rather than the concept of a group in the number theory.

Most of the previous studies have focused on a simple single-attribute authentication policy, which only demands the same affiliation of two participants regardless of their respective status (such as job, age and salary). By cascading each descriptive role with one-time pseudonyms, the existing secret-handshake schemes can be easily extended to handle roles as described in [2]. The first fuzzy matching secret handshake scheme [3] was constructed from the fuzzy identity-based encryption (IBE) scheme [4], which significantly enriches authentication policies and provides approximate attribute-based matching. Although this scheme [3] enables two participants with several attributes to secretly authenticate each other, it also makes the different groups share the same group public/private keys supervised by the same GA. As the user memberships are typically issued from self-governed groups in real-world applications, the PMA protocol with fuzzy matching (PMA-FM) [5] extends the fuzzy matching to the truly multiple groups environments.

However, it is observed that both schemes [3,5] require two participants to implement a secure private set intersection (PSI) protocol [6] before achieving fuzzy matching and secret handshakes. Specifically, two partners should distinguish whether their attributes match and obtain the respective intersection in advance. The original description allows users to terminate the process immediately if their attributes cannot be matched during the secure set intersection, but leads eavesdroppers to recognize the result of the entire secret handshake protocol. Hence, participants must continue the protocol execution even if they use random values rather than their credentials. Additionally, the PSI protocol is only a building block in the above two schemes [3,5], and such a deployment still brings high computational and storage workloads because of the use of the PSI protocol and fuzzy IBE. To date, there are no more efficient secret handshake schemes that can provide similar multi-attribute approximate matching.

There are many important applications of PMA with multi-attribute matching. As stated in [3], searching for friends with common interests on online social networks (OSN) may be one of the most appealing applications. A user who registers with an OSN needs to acquire a set of attribute authorizations certified by the corresponding OSN administrator. By deploying the PMA protocol into the OSN platform, two OSN users can build a secret OSN connection when their profile lists are common to both users. Figure 1 illustrates the specific PMA scenario with multi-attribute matching. By performing OSN profile matching during the PMA protocol, information distinguished from the intersection computation is limited to what is common to both users. Indeed, it is necessary to share some sensitive information for particular occasions (such as for intelligence and law enforcement agencies [7], and health medical center [8]). Hence, PMA with multi-attribute matching can be used in widespread deployments (such as genomic testing [9], recommended systems [10], privacy-preserving data aggregation [11], and keyword searching [12–14]). Additionally, for resource-constrained applications, such as mobile wireless

networks [15], improving the efficiency of PMA with multi-attribute matching is essential.

Our contributions. To further improve the performance of PMA in a manner that is suitable for resource-limited users, in this paper, we focus on searching for an efficient design for the lightweight PMA protocol based on multi-attribute intersection. Because a PSI protocol is an essential tool to achieve the fuzzy matching policy as stated in the schemes [3, 5], we observe that an authorized PSI (APSI) protocol [8] can be directly applied to construct an efficient PMA with multiple attributes. The main contributions of this paper are summarized as follows in detail.

First, we propose a new forward-secure APSI protocol from IBE called IBE-APSI. Because of the optimal efficiency for multiple authorization settings, a practical APSI protocol based on IBE is selected to construct our PMA protocol. However, the APSI protocol that directly originates from IBE cannot guarantee forward security [8]. If the client's secret certificates were stolen by malicious adversaries, then the previous private intersection results of the client would be disclosed. Hence, an optimized IBE-APSI with forward security is presented first.

Second, we design a new efficient PMA protocol with an intersection policy, which is named as IP-PMA. Our IP-PMA is dependent on the above forward-secure IBE-APSI, and not only significantly reduces the computational and storage overhead but also achieves a flexible authentication policy. Specifically, the IP-PMA protocol helps two parties to discern each other if their attribute intersection is not empty or its cardinality is not less than a threshold value (e.g., $d$). Additionally, our proposed protocol enables participants to privately agree on a secret key for subsequent conversations, and avoids them to distinguish the elements that do not belong to the final intersection.

Third, the IP-PMA protocol is secure in the random oracle model (ROM) on the condition that the bilinear Diffie-Hellman (BDH) problem is intractable. Additionally, an empirical simulation demonstrates that the IP-PMA protocol is efficient in practical scenarios, and can be widely used in some resource-constrained environments.

Organization. The remainder of this paper is organized as follows. In Section 2, we review some previous related work on PMA. In Section 3, we review preliminaries related to our proposed schemes. We present the construction of forward-secure IBE-APSI in Section 4, and discuss its security and performance. In Section 5, we describe the concrete construction of IP-PMA in detail, along with its security analysis. In Section 6, we analyze the performance of IP-PMA in detail. Finally, we present the conclusion of this paper in Section 7.

## 2 Related work

In this section, we review some previous work related to PMA (that is unlinkable secret handshakes). The secret handshake mechanism that was introduced in [2], achieves simplicity and efficiency using pairing-based key agreements. Subsequently, many secret handshake schemes equipped with various cryptographic mechanisms (such as CA-oblivious encryption [16], ElGamal [17], RSA [18,19], and message recovery signature [20]) were proposed, which link the protocol instances performed by the same party as a result of public transferred pseudonyms. Although the one-time pseudonym solution can achieve unlinkability in practice, it inevitably incurs a large storage and computation overhead because of the single-usage of pseudonyms.

Hence, many unlinkable secret handshake schemes [21–24] with reusable credentials have been further explored, but still cannot support revocation and traceability. To overcome the flaws in the above solutions, secret handshake schemes that offer certificate revocation or traceability have been proposed using various cryptographic solutions (such as broadcast encryption [25], conditional oblivious transfer [1], and group signature [26, 27]). Because of the tremendous number of computations and storage overhead caused by group key management [28], most of these solutions cannot be applied in a broad range of practical applications.

Ateniese et al. [3] first established the dynamic and fuzzy matching model, in which members could designate the matching policy that the opposite party should satisfy. Inspired by this dynamic matching

model, many variants of secret handshakes have been constructed, such as dynamic controlled matching [29], federated secret handshakes [30] and dynamic expressive matching [31]. However, the groups in [3] differentiated only by group name are in fact managed by the same upper GA, which distributes the same group public/private keys to those groups. Hence, Wen et al. designed a new dynamic matching scheme [32] and fuzzy matching scheme [5], which allow members from completely different groups. In recent years, new secret handshake schemes have been proposed for different practical requirements (such as symptom matching within mobile healthcare social networks [33,34], $k$-time authenticated secret handshake protocol [35], and deniable secret handshake model [36]). Considering non-interactive secret handshakes, a novel cryptographic primitive called matchmaking encryption was presented by Ateniese et al. [37]. Except for the aforementioned fuzzy matching schemes [3,5], there are no other efficient PMA protocols that support multi-attribute intersection matching.

## 3 Preliminaries

In this section, we first review the definitions of bilinear maps [38] and corresponding complexity assumptions, then present the definitions and security requirements of the APSI protocol and PMA protocol.

### 3.1 Bilinear maps and complexity assumption

**Definition 1** (Bilinear maps [38]). Let $\mathbb{G}$ and $\mathbb{G}_T$ be two cyclic groups of prime order $q$, $P$ be a generator of $\mathbb{G}$, and $\hat{e}: \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a bilinear map that has the following three properties.

- Bilinear: For all $P, Q \in \mathbb{G}$ and $a, b \in \mathbb{Z}_q$, if $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$, then the map $\hat{e}$ is bilinear.
- Non-degenerate: If $P$ generates $\mathbb{G}$, then $\hat{e}(P, P) \neq 1$ becomes a generator of $\mathbb{G}_T$.
- Computable: There exists an algorithm to efficiently compute $\hat{e}(P, Q)$ for all $P, Q \in \mathbb{G}$.

In cryptographic applications of bilinear maps, the BDH problem [38] is commonly assumed to be intractable. The definition of the BDH assumption is reviewed as follows.

**Definition 2** (BDH assumption [38]). Let $(\mathbb{G}, \mathbb{G}_T, \hat{e}, P)$ be public bilinear parameters. Given the tuple $(P, aP, bP, cP)$ for some $a, b, c \in \mathbb{Z}_q^*$, the goal of the BDH problem is to compute $\hat{e}(P, P)^{abc} \in \mathbb{G}_T$. The probability of $\mathcal{A}$ in successfully solving the BDH problem is defined by (1), where $\kappa$ denotes the security parameter, $\mathcal{G}$ is a BDH parameter generator.

$$\mathrm{Adv}_{\mathcal{G},\mathcal{A}}(\kappa) = \Pr[\mathcal{A}(\mathbb{G}, \mathbb{G}_T, \hat{e}, P, aP, bP, cP) = \hat{e}(P, P)^{abc}]. \tag{1}$$

We say that the BDH assumption holds if $\mathrm{Adv}_{\mathcal{G},\mathcal{A}} \leqslant \frac{1}{\mathrm{poly}(L)}$ is negligible for any polynomial function $\mathrm{poly}(\cdot)$ and sufficiently large $L$. In other words, there exist no algorithms that can solve the BDH problem with a non-negligible advantage.

### 3.2 Authorized private set intersection

The APSI protocol is an authorized variant of PSI, which requires that the client's records should be signed by a trusted certificate authority (CA). The definition of the APSI protocol involves CA, the client ($\mathsf{C}$) with input $C = \{(c_1, \sigma_1), \ldots, (c_v, \sigma_v)\}$ and the server ($\mathsf{S}$) with input $S = \{s_1, \ldots, s_w\}$, where $v, w$ represent the total number of elements in the corresponding sets, respectively. The definition of the APSI protocol is described as follows:

- Setup($1^\kappa$): Given the security parameter $\kappa$, the public parameters `params` including the public/secret key pair $(\mathrm{PK}_{\mathrm{CA}}, \mathrm{SK}_{\mathrm{CA}})$ of CA are output.
- Authorize($\mathrm{SK}_{\mathrm{CA}}$, $\mathrm{ID}_i$): CA issues a signature $\sigma_i$ for each element $c_i$ in the set $C$, so that $\mathsf{C}$ can acquire a set of valid authorizations for intersection computation. Note that the authorizations (e.g., $\sigma_i = \mathrm{Sign}_{\mathrm{SK}_{\mathrm{CA}}}(c_i)$) are generated by an existentially unforgeable signature algorithm Sign, which can pass the verification by utilizing $\mathrm{PK}_{\mathrm{CA}}$.

• Interaction(C, S): C and S participate this interactive protocol with their respective data set $C$ and $S$, where the authorizations will be obliviously verified by S. As a result, only elements $(c_i, \sigma_i) \in C$ satisfying the verifications (e.g., $c_i = s_j \wedge \mathsf{Vrfy}_{\mathrm{PK_{CA}}}(\sigma_i, c_i) = 1\}$) can be recovered by C, where Vrfy is CA's verification algorithm. Finally, C gets the verified intersection, while S obtains nothing.

For simplicity, we informally explain the following security requirements that should be satisfied by a secure APSI protocol.

• Correctness: If C and S honestly execute an APSI protocol, then C with valid authorizations can definitely acquire a correct intersection in which its elements match with those in $S$. Otherwise, the intersection may be an empty set if C does not hold any correct authorization.

• Client privacy: During the interactive APSI protocol, a malicious S cannot acquire any information about the elements of C except for the size of $C$. Thus, S cannot distinguish which elements in $C$ are consistent with $S$ at the end of the APSI protocol.

• Server privacy: In addition to the upper bound on the size of $S$, the elements that do not belong to the intersection cannot be leaked to C. Moreover, C without correct authorizations also cannot obtain the intersection correctly.

• Unlinkability: This requirement means that any two executions of APSI protocols initiated by the same participant (C or S) cannot be distinguished by the other side.

### 3.3 Definition and security requirements of the PMA protocol

Based on the definitions of secret handshakes [2, 3], the definition of the PMA protocol is described as follows, which involves four basic probabilistic polynomial-time algorithms.

• Setup($1^\kappa$): Given the high-enough security parameter $\kappa$, this algorithm outputs the public parameters `params`.

• CreateGroup(G): Taken as input `params`, GA creates the public/secret key pair $(\mathrm{gpk}_G, \mathrm{gsk}_G)$ for a group G.

• AddMember(U,G): When a user (e.g., U) applies to become a membership of G, the AddMember algorithm issues a set of legitimate attribute credentials for U. Assuming that each member has $n$ attributes $\mathtt{Att_U} = \{u_1, \ldots, u_n\}$, GA needs to verify the claimed identity and attributes before outputting the corresponding attribute credentials $\sigma_U = \{\sigma_{u_1}, \ldots, \sigma_{u_n}\}$ for $\mathtt{Att_U}$. Consequently, U becomes a legitimate member of G by being provided with $\sigma_U$.

• PMA(A,B): Two anonymous users (e.g., (A,B)) execute this protocol to secretly distinguish each other and transmit intelligence. The successful mutual authentication involves (A,B)'s secrets $\sigma_A, \sigma_B$ and `params`. According to the matching outcome of connotative authentication strategies, each party respectively outputs either "1" or "0" to imply whether the mutual authentication succeeds or not. If the output is "1", a correct key is applied to subsequent secure conversations. Otherwise, a random key is used in the following vacuous communications.

The PMA protocol derives from secret handshakes and provides policy-enhanced mutual authentication. The security properties of the PMA protocol are shown as follows:

• Correcteness: If two participants follow the PMA protocol correctly, they can succeed in establishing a session key when their attribute sets are authorized by the same group and have a common intersection.

• Impersonator resistance (IR): Any adversary who does not obtain the membership credentials of claimed attributes, even if observing or participating in an on-going PMA protocol, is unable to pretend to be a valid group member to perform a successful mutual authentication. Thus, a correct session key cannot be agreed between an impersonator and a legitimate member.

• Detector resistance (DR): It is infeasible for an eavesdropper or an active adversary to detect any useful information about the participants in the PMA protocol, by monitoring or activating a PMA protocol with an honest member. Specifically, non-authorized users cannot identify the group affiliations or attributes associated with some legal partner.

• Unlinkability: Multiple instances of the PMA protocol that involve with the same group member, cannot be differentiated or linked by adversaries.

## 4 Forward-secure IBE-APSI protocol

Privacy-preserving policy-based information transfer (PPIT) protocols introduced by Cristofaro et al. [39] allow a client (or requester, C) to retrieve private information from a server (or data owner, S) if C holds the corresponding signature issued by a trusted CA. As stated in [39], APSI protocols can be derived from the constructions of PPIT protocols. Studies on linear-complexity APSI protocols based on RSA-PPIT have been conducted [8, 40]. Other PPIT protocols based on Schnorr and IBE are not used to design APSI schemes with linear complexity because Schnorr-PPIT cannot provide client unlinkability and forward security, and IBE-PPIT cannot guarantee forward security. Therefore, in this section, we present an efficient (linear-complexity) and forward-secure APSI protocol based on IBE-PPIT.

### 4.1 Construction of IBE-APSI

First, we present the construction of forward-secure APSI from IBE-PPIT, which is called IBE-APSI. Then we analyze the security of the IBE-APSI protocol in Subsection 4.2.

Setup($1^\kappa$): Given security parameter $\kappa$, the Setup($1^\kappa$) algorithm first outputs global public parameters $(q, P, \mathbb{G}, \mathbb{G}_T, \hat{e})$, then selects $s \in \mathbb{Z}_q^*$ and computes $Q = sP$, and finally chooses two cryptographic hash functions $H_1 : \{0,1\}^* \to \mathbb{G}$, $H_2 : \mathbb{G}_T \to \{0,1\}^\kappa$. Note that the public/secret key pair of the CA is $(\mathrm{PK}_{\mathrm{CA}}, \mathrm{SK}_{\mathrm{CA}})$, where $\mathrm{PK}_{\mathrm{CA}} = sP$, $\mathrm{SK}_{\mathrm{CA}} = s$.

Authorize($\mathrm{SK}_{\mathrm{CA}}, c_i$): If C wants to obtain some matched information from S through an APSI protocol, then S needs to authenticate whether each identifier in $C$ is signed by the CA. This algorithm aims to issue an authorization/certificate on each identifier ($c_i$) of C. To generate the authorization for each element ($c_i$), the CA calculates a signature $\sigma_i = s \cdot \mathrm{hc}_i$, where $\mathrm{hc}_i = H_1(c_i)$. It is clear that the authorization on $c_i$ can be verified by checking $\hat{e}(P, \sigma_i) \stackrel{?}{=} \hat{e}(Q, \mathrm{hc}_i)$.

Interaction(C, S): This protocol involves the private inputs $\{(c_1, \sigma_1), \ldots, (c_v, \sigma_v)\}$, $\{s_1, \ldots, s_w\}$ of C and S, respectively. When executing the interaction in a time period, C and S can encode the current time period (for example, tt/dd/mm/yy) as a regular bit string $\mathrm{TS} = \{0,1\}^*$ in a uniform manner. The protocol is described as follows:

- $\mathsf{C} \to \mathsf{S} : \{U_c\}$: C first selects a random number $r_c \leftarrow_R \mathbb{Z}_q^*$ and sends $U_c = r_c \cdot P$ to S.
- $\mathsf{S} \to \mathsf{C} : \{U_s, T\}$: For the current interaction, S first generates time-stamp $\mathrm{TS}_s$ and calculates $\mathrm{ht}_s = H_1(\mathrm{TS}_s)$. Regarding the data identifiers $\{s_1, \ldots, s_w\}$, corresponding hash values $\{\mathrm{hs}_1, \ldots, \mathrm{hs}_w\}$ are pre-computed offline before the subsequent interaction, where $\mathrm{hs}_j = H_1(s_j), j \in [1, \ldots, w]$. Then, S chooses $r_s \leftarrow_R \mathbb{Z}_q^*$ and computes $U_s = r_s \cdot P$. Finally, S calculates $t_j = H_2(\hat{e}(Q, \mathrm{hs}_j)^{r_s} \cdot \hat{e}(U_c, \mathrm{ht}_s)^{r_s})$ for every $j \in [1, \ldots, w]$ and sends $T = \{t_1, t_2, \ldots, t_w\}$ to C.

Additionally, C can obtain the corresponding time-stamp tag $\mathrm{ht}_c = H_1(\mathrm{TS}_c)$ when it initiates the protocol. Then C computes a series of hash values $t'_i = H_2(\hat{e}(U_s, \sigma_i + r_c \cdot \mathrm{ht}_c))$ using each authorization $\sigma_i$ ($i \in [1, \ldots, v]$). C collects $T' = \{t'_1, \ldots, t'_v\}$ and acquires the intersection $I_C = T \cap T'$. Thus, C can distinguish real identifiers that match the data signals of S through element indices in the intersection.

**Correctness.** By testing the following computation equations, we can prove that the proposed IBE-APSI protocol is correct. If each element of C is signed correctly by CA and the elements are indeed equal (e.g., $c_i = s_j$) in the same time period ($\mathrm{TS}_c = \mathrm{TS}_s, \mathrm{ht}_c = \mathrm{ht}_s$), then C can obtain the acquired intersection results. We illustrate correctness by considering one element as an example in the following verification:

$$\begin{aligned} t'_i &= H_2(\hat{e}(U_s, \sigma_i + r_c \cdot \mathrm{ht}_c)) = H_2(\hat{e}(r_s \cdot P, s \cdot \mathrm{hc}_i) \cdot \hat{e}(r_s \cdot P, r_c \cdot \mathrm{ht}_c)); \\ t_j &= H_2(\hat{e}(Q, \mathrm{hs}_j)^{r_s} \cdot \hat{e}(U_c, \mathrm{ht}_s)^{r_s}) = H_2(\hat{e}(r_s \cdot P, s \cdot \mathrm{hc}_i) \cdot \hat{e}(r_s \cdot P, r_c \cdot \mathrm{ht}_c)). \end{aligned} \tag{2}$$

From the above deduction, we can see that $t'_i = t_j$ holds on the condition that $\mathrm{hc}_i = \mathrm{hs}_j$ and $\mathrm{ht}_c = \mathrm{ht}_s$. In other words, C can identify the corresponding intersection by computing $T \cap T'$. Accordingly, the linear IBE-APSI satisfies the correctness property.

### 4.2 Security analysis

**Lemma 1.** The IBE-APSI protocol is secure under the BDH assumption in the ROM.

*Proof.* We briefly demonstrate that the proposed IBE-APSI satisfies the security requirements of the APSI protocol. The proofs of client privacy, server privacy and unlinkability are described as follows.

**Client privacy:** It requires that IBE-APSI should not reveal any information related to $\mathsf{C}$. We prove this by analyzing the transcripts in the specific IBE-APSI protocol. According to the transmitted data, $\mathsf{C}$ only sends a value $U_c$ to $\mathsf{S}$. Note that $U_c = r_c \cdot P(r_c \leftarrow_R \mathbb{Z}_q)$ is uniformly selected from the group $\mathbb{G}$ at random and $U_c$ is completely unrelated to $C$. Hence, $\mathsf{S}$ or other adversaries cannot identify any element or authorization of $C$.

**Server privacy:** Regarding the privacy of $\mathsf{S}$ in IBE-APSI, we need to construct experiment $\mathsf{Exp}_{\mathrm{SP}}$ between adversary $\mathcal{A}$ and challenger $\mathcal{B}$. Assuming that adversary $\mathcal{A}$ breaks the server privacy with non-negligible advantage $\epsilon$ in experiment $\mathsf{Exp}_{\mathrm{SP}}$, $\mathcal{B}$ uses $\mathcal{A}$ to solve the BDH problem with non-negligible probability, which is similar to an IBE scheme [38]. First, $\mathcal{B}$ is given BDH parameters $(q, \mathbb{G}, \mathbb{G}_T, \hat{e})$ generated by $\mathcal{G}$ and BDH problem instance $(P, P_1, P_2, P_3) = (P, aP, bP, cP)$. Then the goal of $\mathcal{B}$ is to obtain result $D = \hat{e}(P, P)^{abc} \in \mathbb{G}_T$. Experiment $\mathsf{Exp}_{\mathrm{SP}}$ is described as follows.

Init: $\mathcal{B}$ simulates Setup to produce parameters $\mathtt{params} = (q, P, \mathbb{G}, \mathbb{G}_T, \hat{e}, H_1, H_2, Q)$, where $Q = P_1$ is set as the public key of the CA. Particularly, $H_1$ and $H_2$ are random oracles, and $\mathtt{params}$ is sent to $\mathcal{A}$.

Queries: Similar to the proof of a semantically secure IBE scheme, $\mathcal{A}$ who will attack the IBE-APSI can be trained by initiating some queries associated with Authorize algorithm and hash functions. When $\mathcal{A}$ invokes queries to $\mathcal{B}$, $\mathcal{B}$ will respond with some answers which are indistinguishable with the real attack.

• Hash query: If $\mathcal{A}$ queries random oracle $H_1$ related to identifiers $c_i$ at one time period TS, then $\mathcal{B}$ prepares a list of responses to answer those queries. $\mathcal{B}$ first invokes the coin tossing algorithm and obtains a random coin $\mathrm{coin} \leftarrow_R \{0, 1\}$. If $\mathrm{coin}_i = 0$, then $\mathcal{B}$ returns random hash value $H_1(c_i) = \gamma_i \cdot P, \gamma_i \leftarrow_R \mathbb{Z}_q^*$. Otherwise, $\mathcal{B}$ returns $H_1(c_i) = \gamma_i \cdot P_2$. Additionally, $\mathcal{A}$ queries $H_2$ on input $M_i \in_R \mathbb{G}_T$, then $\mathcal{B}$ selects $\mu_i \in_R \{0, 1\}^{\kappa}$ at random and responds with $\mu_i$ as the answer to query $H_2(M_i)$.

• Authorize query: $\mathcal{A}$ requests queries for Authorize on one $c_i$ of its choice, and $\mathcal{B}$ simulates Authorize to return the corresponding authorization $\sigma_i$. To acquire the authorization, the $H_1$ query on $c_i$ is issued by $\mathcal{A}$, and corresponding response $H_1(c_i)$ is also maintained by $\mathcal{B}$. If $\mathrm{coin}_i = 1$, then $\mathcal{B}$ terminates the experiment. Note that $H_1(c_i) = \gamma_i \cdot P$ if $\mathrm{coin}_i = 0$ and authorization $\sigma_i = \gamma_i \cdot Q$, which matches the signature construction $\sigma_i = a \cdot H_1(c_i)$, where $a$ is the potential secret key of the CA.

Challenge: After being trained by running the above queries, $\mathcal{A}$ starts to interact with $\mathcal{B}$ to execute an attack.

• First, $\mathcal{A}$ selects two elements $c_0$ and $c_1$, which are different from other queried elements mentioned above and sends the challenging tuple $(c_0, c_1)$ to $\mathcal{B}$.

• $\mathcal{A}$, as the role of $\mathsf{C}$, executes the $\mathsf{Interaction}$ protocol and sends random message $U_c = r_c \cdot P, r_c \in_R \mathbb{Z}_q^*$ to $\mathcal{B}$, who plays the role of $\mathsf{S}$.

• $\mathcal{B}$ generates random bit $b \leftarrow_R \{0, 1\}$ and constructs the response values for identifier element $c_b$ at time period $\mathrm{TS}_s = \mathrm{TS}_c$. $\mathcal{B}$ invokes the simulation algorithm to obtain the hash values related to $c_b$ and $\mathrm{TS}_s = \mathrm{TS}_c$. For simplicity, $\mathcal{B}$ executes the one-time coin tossing algorithm to obtain the result of $\mathrm{coin}_b$. If $\mathrm{coin}_b = 1$, then $\mathcal{B}$ continues the experiment so that $H_1(c_b) = \gamma^* \cdot P_2, \gamma^* \leftarrow_R \mathbb{Z}_q^*$, $H_1(\mathrm{TS}_s) = H_1(\mathrm{TS}_c) = \gamma^* \cdot P$. Otherwise, $\mathcal{B}$ terminates the experiment.

• According to the $\mathsf{Interaction}$ protocol, $\mathcal{B}$ computes response values $U_s = {\gamma^*}^{-1} \cdot P_3$, $t_b \leftarrow_R \{0, 1\}^{\kappa}$ and returns $\{U_s, t_b\}$ to $\mathcal{A}$. We observe that $\mathcal{A}$ has to recover $t_b'$ by querying $M = \hat{e}(U_s, \sigma_b) \cdot \hat{e}(U_s, H_1(\mathrm{TS}_c))^{r_c} = D \cdot \hat{e}(U_c, P_3)$ on $H_2$, and further complete the matching for intersection computation, where $\sigma_b = a \cdot H_1(c_b) = \gamma^* \cdot abP$.

Guess: Finally, $\mathcal{A}$ wins the experiment if the guess that it outputs satisfies $b' = b, b' \in_R \{0, 1\}$.

Essentially, the process of private matching for $\mathsf{C}$ in the IBE-APSI protocol is derived from the decryption key. Similar to the security of the IBE scheme [38], the view of $\mathcal{A}$ in $\mathsf{Exp}_{\mathrm{SP}}$ is indistinguishable from a real attack if $\mathcal{B}$ does not abort during the simulation. Based on the previous analysis [38], the success

probability of $\mathsf{Exp}_{\mathrm{SP}}$ is at least $1/e(1+q_A)$ for all Authorize queries $q_A > 0$ on the condition that $\mathcal{B}$ does not abort.

According to the execution of $\mathsf{Exp}_{\mathrm{SP}}$, we can argue that if success probability $\epsilon \leqslant |\Pr[c = c'] - \frac{1}{2}|$ of $\mathcal{A}$ is non-negligible (that is, $\mathcal{A}$ calculates the correct authorization $\sigma_b$), then $\mathcal{B}$ applies $\mathcal{A}$ to solve the BDH problem (that is, $D = \hat{e}(P,P)^{abc} = M/\hat{e}(U_c, P_3)$) with at least $\epsilon'$. Suppose that $\mathcal{A}$ proposes at most $q_A > 0$ Authorize queries and $q_{H_2} > 0$ hash queries to $H_2$. Then, $\mathcal{B}$ can solve the BDH problem with at least $\epsilon' \geqslant \frac{2\epsilon}{e(1+q_A)\cdot q_{H_2}}$ probability advantage.

**Unlinkability:** Regarding the proposed IBE-APSI protocol, the data transferred from $\mathsf{C}$ (e.g., $U_c$) is generated randomly for each instance, which is uniformly and independently distributed in $\mathbb{G}^*$. Simultaneously, the data transcripts produced by $\mathsf{S}$ (e.g., $\{U_s, T\}$) are also calculated by different random values $r_s$ and $U_c$ for different interaction instances. Accordingly, it is infeasible for any adversary to distinguish two instances invoked by the same participant (client or server). That is, client and server unlinkability can be realized.

Therefore, we have proved that the IBE-APSI protocol is proved is a secure APSI protocol under the BDH assumption in the ROM.

**Remark 1.** The proposed IBE-APSI protocol also has linear complexity which only needs $O(w)$ communication overhead. Compared with other basic constructions from RSA and Schnorr with quadratic computation consumptions, the APSI based on IBE is the most efficient, particularly for settings which the client holds multiple authorizations [39]. The APSI protocol can be directly transformed from IBE-PPIT by enabling $\mathsf{S}$ to send ciphertexts related to the corresponding identifiers. However, such a construction cannot guarantee the forward security of transactions, where past parts of the information intersection of $\mathsf{C}$ are exposed if some authorizations are leaked or intercepted. By combining the blindness and timestamp techniques, the optimized IBE-APSI with forward security is presented first in this subsection. Specifically, when $\mathsf{C}$ recovers the corresponding tag $t_i'$ related to one identifier $c_i$, authorization $\sigma_i$ is completely blinded by random values generated from two participants for each instance. Thus, even if one authorization is captured by an adversary, the past transcripts linked with the revealed authorization remain unrecognized. Hence, the proposed IBE-APSI protocol is forward-secure through initiating the interactive key agreement. We notice that the IBE-APSI protocol is not mutual, which only enables the client to obtain the intersection results, whereas the functionality of APSI can be adapted to enforce the intersection policies. Therefore, we are primarily concerned with taking advantage of the forward-secure IBE-APSI protocol to construct a new efficient PMA protocol to achieve unlinkability and the mutual property (that is, both participants acquire intersection and bidirectional authentication). The detailed construction and security analysis of the new PMA protocol is presented in Section 5.

## 5 Intersection-policy PMA from IBE-APSI

In this section, the mutual authentication policy is reconsidered that aims to achieve multi-attribute intersection matching for two participants authorized by an independent GA. Based on the optimized IBE-APSI, an efficient intersection-policy PMA protocol denoted by IP-PMA, is first constructed in Subsection 5.1. Subsequently, the security of the proposed IP-PMA protocol is demonstrated in Subsection 5.2.

### 5.1 Construction of IP-PMA

The IP-PMA protocol derived from unlinkable secret handshakes comprises the following four algorithms.

Setup($1^\kappa$): Given security parameter $\kappa$, the algorithm outputs public parameters $\mathtt{params} = (q, P, \mathbb{G}, \mathbb{G}_T, \hat{e} : \mathbb{G}\times\mathbb{G} \to \mathbb{G}_T, H_1, H_2, d)$, where $P$ is a generator of subgroup $\mathbb{G}$ of prime order $q$, and $H_1 : \{0,1\}^* \to \mathbb{G}$ and $H_2 : \mathbb{G}_T \to \{0,1\}^\kappa$ are two cryptographic hash functions that are used to ensure data integrity and verifiability. Additionally, $d \geqslant 1$ is defined as a threshold value of the authentication policies, which is prescribed as the minimum cardinality of the attribute intersection between two protocol partners.

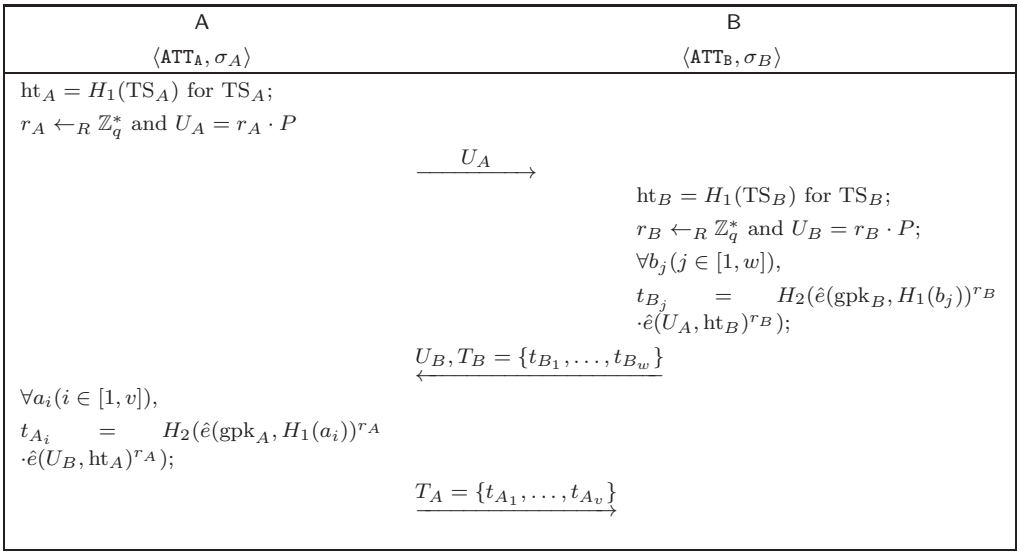


**Figure 2** IP-PMA protocol derived from IBE-APSI.

CreateGroup(G): To manage group G, GA first chooses $s \leftarrow_R \mathbb{Z}_q^*$ as its group secret key, and then generates its group public key $\mathrm{gsk} = s \cdot P$.

AddMember(U, G): When verifying the identity of applicant U, GA calls AddMember to make the applicant an authenticated group member. Note that U typically holds multiple attributes $\mathtt{Att}_U = (u_1, \ldots, u_n)$, where each attribute is encoded to binary string ($u_i \in \{0,1\}^*$). For each attribute $u_i, i \in [1, n]$, GA computes $\sigma_{U_i} = s \cdot H_1(u_i)$ for U. Accordingly, U becomes the legitimate group member with multiple attribute credentials $\sigma_U = \{\sigma_{U_1}, \ldots, \sigma_{U_n}\}$.

PMA(A, B): In some real application scenarios, two anonymous users A and B would like to run this PMA algorithm to distinguish whether they satisfy the other side's authentication policy. PMA is the core algorithm of our IP-PMA, which supports a flexible intersection policy. If the other side belongs to the same designated group and holds common attributes, then the two partners can authenticate each other secretly without leaking their privacy and agree on a session key for subsequent transactions. At one time period TS, A, who holds $v$ attributes $\mathtt{Att}_\mathtt{A} = \{a_1, \ldots, a_v\}$, executes the PMA algorithm with secret input $\sigma_A$, and B, who holds $w$ attributes $\mathtt{Att}_\mathtt{B} = \{b_1, \ldots, b_w\}$, runs the PMA algorithm with its secret input $\sigma_B$. The detailed interactive PMA protocol is specified in Figure 2.

To confirm that the intersection policy between the two participants is attained, A first calculates $T'_A = \{t'_{A_1}, \ldots, t'_{A_v}\}$ for each attribute $a_i$ ($i \in [1, v]$), where $t'_{A_i} = H_2(\hat{e}(U_B, \sigma_{A_i} + r_A \cdot \mathrm{ht}_A))$. By matching intersection $I_A = T_B \cap T'_A$, A can distinguish those attributes that are equal to the other side's attributes and the cardinality of the corresponding intersection. If $|I_A| \geqslant d$, then A outputs "1" and computes subsequent session key $K_A = H_2(\hat{e}(U_B, \sum_i^{t_{A_i} \in I_A} \sigma_{A_i})^{r_A})$. Otherwise, A outputs "0" and generates random ephemeral key $K_A \leftarrow_R \{0,1\}^\kappa$ for subsequent interactions with B.

Meanwhile, B calculates $T'_B = \{t_{B_1}, \ldots, t_{B_w}\}$ for each attribute $b_j (j \in [1, w])$ after receiving $T_A$, where $t_{B_j} = H_2(\hat{e}(U_A, \sigma_{B_j} + r_B \cdot \mathrm{ht}_B))$. B also identifies those attributes that satisfy the intersection policy by computing intersection $I_B = T_A \cap T'_B$. If $|I_B| \geqslant d$, B outputs "1" and computes a subsequent session key $K_B = H_2(\hat{e}(U_A, \sum_j^{t_{B_j} \in I_B} \sigma_{B_j})^{r_B})$. Otherwise, B outputs "0" and similarly produces random ephemeral key $K_B \leftarrow_R \{0,1\}^\kappa$.

**Correctness.** If the authentication policies of A and B match, this means that both participants belong to the same group ($\mathrm{gpk}_A = s_A \cdot P = s_B \cdot P = \mathrm{gpk}_B$) and have common elements authorized by the same group manager. That is, both A and B can compute the intersection correctly. For an implementation of IP-PMA at one time period that is uniformly encoded by $\mathrm{TS}_A = \mathrm{TS}_B(\mathrm{ht}_A = \mathrm{ht}_B)$, the correctness of the intersection computation can be verified by considering attribute $a_i = b_j$ as an

example. The detailed explanation is as follows:

$$\begin{aligned} t'_{A_i} &= H_2(\hat{e}(U_B, \sigma_{A_i} + r_A \cdot \mathrm{ht}_A)) = H_2(\hat{e}(r_B \cdot P, s_A \cdot H_1(a_i)) \cdot \hat{e}(r_B \cdot P, r_A \cdot \mathrm{ht}_A)); \\ t_{B_j} &= H_2(\hat{e}(\mathrm{gpk}_B, H_1(b_j))^{r_B} \cdot \hat{e}(U_A, \mathrm{ht}_B)^{r_B}) = H_2(\hat{e}(r_B \cdot P, s_A \cdot H_1(a_i)) \cdot \hat{e}(r_B \cdot P, r_A \cdot \mathrm{ht}_A)). \end{aligned} \tag{3}$$

According to (3), $t'_{A_i} = t_{B_j}$ can be established if and only if attributes $a_i$ and $b_j$ match and are signed by the same GA ($s_A = s_B$) in the same time period $\mathrm{ht}_A = \mathrm{ht}_B$. Clearly, other identical authorized attributes can be disclosed by A by matching set $T'_A$ with set $T_B$. Furthermore, B also distinguishes common attributes (e.g., $t'_{B_j} = t_{A_i}$) by matching $T'_B$ with $T_A$ in a similar manner. For further affirmation and secret communication, A and B compute $K_A$ and $K_B$ according to intersection $I_A$ and $I_B$, respectively. The corresponding session keys are negotiated using (4).

Suppose that the participants (A and B) belong to the same group (that is, $\mathrm{gpk}_A = \mathrm{gpk}_B$) and have common authorized certificates. Then, they can acquire equal intersections, $I_A = I_B$, and compute the corresponding aggregated sum values, which are denoted by $\mathrm{IP}_A = \sum_i^{t_{A_i} \in I_A} H_1(a_i) = \sum_j^{t_{B_j} \in I_B} H_1(b_j) = \mathrm{IP}_B$, respectively. When setting $\mathrm{IP}_A = \mathrm{IP}_B$, session key $K_A = K_B$ is agreed for the subsequent secret two-party communications. Therefore, the PMA protocol between A and B is also successful. If the intersection policy is not satisfied, then session keys $K_A$ and $K_B$ are random values, which do not create meaningful conversations between A and B.

$$\begin{aligned} K_A &= H_2\left(\hat{e}\left(U_B, \sum_i^{t_{A_i} \in I_A} \sigma_{Ai}\right)^{r_A}\right) = H_2\left(\hat{e}\left(r_B \cdot P, s_A \cdot \left(\sum_i^{t_{A_i} \in I_A} H_1(a_i)\right)\right)^{r_A}\right) \\ &= H_2\left(\hat{e}\left(s_A \cdot P, \sum_i^{t_{A_i} \in I_A} H(a_i)\right)^{r_A \cdot r_B}\right) = H_2\left(\hat{e}\left(\mathrm{gpk}_A, \sum_i^{t_{A_i} \in I_A} H_1(a_i)\right)^{r_A \cdot r_B}\right), \\ K_B &= H_2\left(\hat{e}\left(U_A, \sum_j^{t_{B_j} \in I_B} \sigma_j\right)^{r_B}\right) = H_2\left(\hat{e}\left(r_A \cdot P, s_B \cdot \left(\sum_j^{t_{B_j} \in I_B} H_1(b_j)\right)\right)^{r_B}\right) \\ &= H_2\left(\hat{e}\left(s_B \cdot P, \sum_j^{t_{B_j} \in I_B} H_1(b_j)\right)^{r_B \cdot r_A}\right) = H_2\left(\hat{e}\left(\mathrm{gpk}_B, \sum_j^{t_{B_j} \in I_B} H_1(b_j)\right)^{r_B \cdot r_A}\right). \end{aligned} \tag{4}$$

### 5.2 Security analysis

**Theorem 1.** IP-PMA is a secure PMA protocol with multi-attribute intersection under the BDH assumption in the ROM.

*Proof.* As stated in Subsection 3.3, a secure PMA protocol needs to satisfy IR, DR and unlinkability, in addition to the correctness property. Through presenting the above verification equations, we have shown that our IP-PMA protocol is correct. To provide the formal security proof of the above properties, the corresponding attack experiments are modeled between adversary $\mathcal{A}$ and challenger $\mathcal{B}$. If $\mathcal{A}$ wins those attack experiments with a non-negligible probability, then $\mathcal{B}$ uses $\mathcal{A}$ to solve the BDH problem with a non-negligible probability. With the aim of enabling $\mathcal{A}$ to be trained in the adaptively chosen-message attack, we consider that $\mathcal{A}$ is a probabilistic polynomial-time algorithm that can access the following oracles $\mathcal{O} = \{\mathcal{O}_{\mathrm{CG}}, \mathcal{O}_{\mathrm{AM}}, \mathcal{O}_{\mathrm{PMA}}, \mathcal{O}_{H_1}, \mathcal{O}_{H_2}\}$. These oracles are simulated by $\mathcal{B}$ to execute different algorithms of the IP-PMA protocol (i.e, CreateGroup, AddMember, and PMA) and two full-domain hash functions which are considered as random oracles.

**Impersonator resistance**. If $\mathcal{A}$ wants to break the IR property, then $\mathcal{A}$ must forge some valid attribute credentials before executing a successful PMA protocol with the other side. The experiment for IR is denoted by $\mathtt{Exp}_{\mathrm{IR}}$, which includes multiple interactions between $\mathcal{A}$ and $\mathcal{B}$. Suppose that $\mathcal{B}$ is given a BDH instance $(P, P_1, P_2, P_3)$, then $\mathtt{Exp}_{\mathrm{IR}}$ comprises the following steps.

Init: $\mathcal{B}$ first initializes the system environment by calling the algorithm $\mathtt{params} \leftarrow \mathrm{Setup}(1^\kappa)$ and obtains public parameters $\mathtt{params} = (q, P, \mathbb{G}, \mathbb{G}_T, \hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T, H_1, H_2, d)$ from the given BDH

parameters generated by $\mathcal{G}$. Then, $\mathcal{A}$ selects independent group $\mathsf{G}^*$ and $\mathtt{ATT}^*$ as its attack target, and initially sets $\mathsf{Chosen} = (\mathsf{G}^*, \mathtt{ATT}^*)$.

Queries: $\mathcal{A}$ is allowed to query oracles $\mathcal{O}$ to corrupt some groups or group members. Specifically, $\mathcal{A}$ can adaptively query $\mathcal{O}_{\mathrm{CG}}$, $\mathcal{O}_{\mathrm{AM}}$, $\mathcal{O}_{\mathrm{PMA}}$, and hash functions, which are served by $\mathcal{B}$.

• CreateGroup query: $\mathcal{A}$ could query $\mathcal{O}_{\mathrm{CG}}$ to create groups that are denoted by set $\mathsf{Cor_G}$. $\mathcal{B}$ produces the group public key and group secret key for groups in $\mathsf{Cor_G}$ according to the CreateGroup algorithm. Note that the corrupted group set cannot include target group $\mathsf{G}^*$, that is, $\mathsf{Chosen} \cap \mathsf{Cor_G} = \emptyset$. For the target group $\mathsf{G}^*$, $\mathcal{B}$ generates the group key pairs using the BDH parameters, that is, the group public key of $\mathsf{G}^*$ is $P_1$, which implies that the group secret key is $a$.

• Hash query: If $\mathcal{A}$ queries hash oracle $\mathcal{O}_{H_1}$ on one attribute index $u_i \in_R \{0,1\}^*$, then $\mathcal{B}$ responds with hash results $h_i = H_1(u_i) \leftarrow_R \mathbb{G}$. Let $q_{H_1}$ be the total number of queries for hash oracle $H_1$, $\mathcal{B}$ needs to prepare the $q_{H_1}$ hash response set $\mathcal{H}_1 = (h_1, h_2, \ldots, h_{q_{H_1}})$, where $P_2$ is embedded in the hash values, by running a coin tossing algorithm similar to the hash query in the security proof of IBE-APSI. Additionally, for hash queries on $H_2$, $\mathcal{B}$ can consider the random values from $\{0,1\}^{\kappa}$ as the corresponding responses of $\mathcal{O}_{H_2}$.

• AddMember query: $\mathcal{A}$ repeatedly queries $\mathcal{O}_{\mathrm{AM}}$ to acquire the secrets of $\mathcal{U}$, which include the queried (corrupted) group members from $\mathsf{Cor_G}$. Hence, $\mathcal{A}$ claims that it has a series of valid attributes ($\mathtt{Att}_{\mathcal{A}}$) that are authorized by the CAs of groups in set $\mathsf{Cor_G}$. $\mathcal{B}$ mostly returns the corresponding credentials according to the AddMember algorithm. Equally, $\mathcal{A}$ cannot query to be a group member of $\mathsf{G}^*$. The AddMember algorithm of $\mathsf{G}^*$ is simulated by $\mathcal{B}$ in accordance with the Authorize algorithm, which is described as the security proof of IBE-APSI.

• PMA query: $\mathcal{A}$ can query $\mathcal{O}_{\mathrm{PMA}}$ to execute some PMA protocol as the role of one participant (e.g., $\mathsf{A}$), while $\mathcal{B}$ represents the other side (e.g., $\mathsf{B}$) and simulates the corresponding transcripts.

Challenge: $\mathcal{A}$ represents $U^* \notin \mathcal{U}$ with attribute set $\mathtt{Att}^*$ satisfying $U^* \in \mathsf{G}^*$ ($\mathcal{U} \cap \mathsf{G}^* = \emptyset$). Note that the attribute elements in $\mathtt{Att}^*$, which can be informed to $\mathcal{B}$, are also not queried for $\mathcal{O}_{\mathrm{AM}}$. Then, $\mathcal{A}$ randomly generates $U_{\mathcal{A}} = r_{\mathcal{A}} \cdot P$ and sends $U_{\mathcal{A}}$ to the other participant simulated by $\mathcal{B}$, who represents an honest group member of $\mathsf{G}^*$. Subsequently, $\mathcal{B}$ responds with $\{U_{\mathcal{B}}, T_{\mathcal{B}}\}$ using an attribute set $\mathtt{Att}_{\mathcal{B}}$, which intersects with $\mathtt{Att}^*$. Note that the transmitted transcripts $\{U_{\mathcal{B}}, T_{\mathcal{B}}\}$ are multiple encryptions based on $\mathtt{Att}^*$, which are also used by $\mathcal{B}$ to implant the BDH problem instance. Finally, $\mathcal{A}$ also sends matching token set $\{U_{\mathcal{A}}\}$ to $\mathcal{B}$ according to its set $\mathtt{Att}^*$ before completing the IP-PMA protocol.

Output: $\mathcal{A}$ attempts to acquire the intersection elements after receiving $\{U_{\mathcal{B}}, T_{\mathcal{B}}\}$, and then obtains the correct session key for subsequent conversations. If the PMA algorithm between $\mathcal{A}$ and $\mathcal{B}$ outputs "1", then $\mathcal{A}$ wins this experiment, and responds with "1", or "0" otherwise. In fact, $\mathcal{A}$ outputs "1" on the condition that it can deduce the corresponding decryption key (that is, the correct element authorization).

Accordingly, we analyze that if the successful probability of $\mathcal{A}$ is non-negligible $\epsilon$ (that is, $\mathcal{A}$ can compute the correct decryption key with non-negligible probability), then $\mathcal{B}$ can compute $D = \hat{e}(P,P)^{abc}$ with non-negligible probability $\epsilon' \geqslant \frac{2\epsilon}{e(1+q_A)\cdot q_{H_2}}$. The detailed reduction approach is similar to that for the server privacy of IBE-APSI in Lemma 1, which does not enable the client to obtain the corresponding intersection elements without legitimate authorizations.

**Detector resistance.** The experiment for DR is denoted by $\mathtt{Exp}_{\mathrm{DR}}$, which allows $\mathcal{A}$ to discover the difference between two instances of the PMA protocol by interacting with $\mathcal{B}$. Additionally, $\mathcal{A}$ can further distinguish which instance is executed with the true group member rather than simulator $\mathsf{R}$ with advantage $\epsilon$. Clearly, $\mathcal{A}$ can guess successfully with at least $\frac{1}{2}$ advantage. Thus, $\mathcal{A}$ can win the experiment $\mathtt{Exp}_{\mathrm{DR}}$ with at least $\frac{1}{2} + \epsilon$ advantage. Furthermore, $\mathcal{A}$ can be applied to solve the BDH problem with a non-negligible advantage by executing a security game $\mathtt{Exp}_{\mathrm{DR}}$ with $\mathcal{B}$.

Considering the similarity between $\mathtt{Exp}_{\mathrm{IR}}$ and $\mathtt{Exp}_{\mathrm{DR}}$, in the following, we explain the key difference in Output for simplicity. After being trained by accessing oracles $\mathcal{O}$, $\mathcal{A}$ needs to execute a PMA protocol with $\mathcal{B}$, who selects $b \leftarrow_R \{0,1\}$. If $b = 0$, then $\mathcal{B}$ executes a PMA protocol with $\mathcal{A}$ as proof of IR. If $b = 1$, then $\mathcal{B}$ executes a PMA protocol with $\mathcal{A}$ as $\mathsf{R}$, where $\mathsf{R}$ is a random simulator. Then, $\mathcal{B}$ sends $U_{\mathcal{B}} = \mathrm{SIM}(\mathtt{params}), T_{\mathcal{B}} = \mathrm{SIM}(\mathtt{params})$ to $\mathcal{A}$. From the transcripts of our proposed IP-PMA scheme, we can also see that $\{U_{\mathcal{B}}, T_{\mathcal{B}}\}$ sent by an honest group member are still uniformly distributed in $\mathbb{G}$ and

$\{0,1\}^{\kappa}$. In this way, the transcripts produced by simulator $\mathsf{R}$ are indistinguishable from an honest member simulated by $\mathcal{B}$. Thus, it is necessary for $\mathcal{A}$ to distinguish values $\{U_{\mathcal{B}}, T_{\mathcal{B}}\}$ computed by a real user from random values chosen by $\mathsf{R}$. However, this case only occurs if $\mathcal{A}$ can recover the correct $t'_{A_i}$ from $\{U_{\mathcal{B}}, T_{\mathcal{B}}\}$ using the correct decryption key, in which we can use $\mathcal{A}$ to solve the BDH problem. In the final step of $\mathtt{Exp}_{\mathrm{DR}}$, $\mathcal{A}$ needs to output its guess $b'$ related to $b$. Because of the complete randomness of transcripts sent from the participants, $\mathcal{A}$ cannot output the correct $b' = b$ with a non-negligible advantage over $\frac{1}{2}$. Therefore, based on the assumption of the intractability of the BDH problem, $\mathcal{A}$ cannot win the $\mathtt{Exp}_{\mathrm{DR}}$ to attack the DR property with a non-negligible advantage over $\frac{1}{2}$.

**Unlinkability.** During the PMA protocol, the participants only send a random value and series of identifier tokens that are also blinded by respective random factors. Informally, it is infeasible for any probabilistic polynomial adversary to differentiate two instances of PMA performed by the same party. Based on the proof of IR, we provide a sketch proof of unlinkability by defining the experiment for unlinkability (which is called $\mathtt{Exp}_{\mathrm{Unlink}}$). Compared with experiments $\mathtt{Exp}_{\mathrm{IR}}$ and $\mathtt{Exp}_{\mathrm{DR}}$, the main difference is in the challenge phase. $\mathcal{B}$ randomly selects $b \leftarrow_R \{0,1\}$, and then performs two mutual authentication protocols with $\mathcal{A}$. If $b = 0$, then $\mathcal{B}$ represents the same participant (e.g., $U^*$) to activate two mutual authentication instances denoted by PMA $(\mathcal{A}, U^*)$. If $b = 1$, then $\mathcal{B}$ represents different users $U^*$ and $V^*$ to execute two PMA instances with $\mathcal{A}$ (e.g., PMA$(\mathcal{A}, U^*)$, PMA $(\mathcal{A}, V^*)$). In the process of $\mathtt{Exp}_{\mathrm{Unlink}}$, $\mathcal{A}$ attempts to recognize whether the two sessions are performed by the same user $U^*$ and outputs final guess $b'$. Consequently, $\mathcal{A}$ wins the experiment on the condition that it can guess $b' = b$ with a non-negligible advantage over $\frac{1}{2}$.

Specifically, we notice that each transmitted token (e.g, $t_{A_i}$ or $t_{B_j}$) associated with one attribute needs to be calculated using the random factors of two participants. It is infeasible for any adversary to distinguish whether the two interactive instances PMA$(\mathcal{A}, U^*)$ and PMA$(\mathcal{A}, V^*)$ are related to the same group member. From the proof of Lemma 1 and IR, if $\mathcal{A}$ can win experiment $\mathtt{Exp}_{\mathrm{Unlink}}$ with a non-negligible advantage $\frac{1}{2} + \epsilon'$, then $\mathcal{B}$ also can apply $\mathcal{A}$ to solve the BDH problem. Therefore, based on the intractability of the BDH problem, the unlinkability of IP-PMA is achieved.

## 6 Performance evaluation of IP-PMA

In this section, we analyze the performance of IP-PMA regarding its computation and communication overhead by comparing it with the previous ABK07 scheme [3] and WZ14 scheme [5]. Most of the experiments were implemented on a personal computer (Intel Pentium-4 2.4 GHz CPU, 8 GB RAM). The related algorithms were implemented using the GMP library for modular operations, and OpenSSL to generate parameters and hash functions.

According to the empirical results in previous work, bilinear pairing and modular exponentiation are two of the most time-consuming operations that are considered. The PBC (pairing-based cryptography)[1)] cryptographic library is an efficient tool used to produce elliptic curve parameters and compute pairings. For clarity, $T_{\mathrm{p}}$ denotes the computation time for one bilinear pairing operation in the elliptic curve groups, which costs approximately 5.427 ms. $T_{\mathrm{e}}$ denotes the total time for one modular exponential operation, which costs approximately 2.42 ms. $T_{\mathrm{sm}}$ denotes the running time for computing a scalar multiplication operation, which costs approximately 2.165 ms. $T_{\mathrm{H}}$ denotes the total running time for one hash-to-point operation, which costs approximately 5.493 ms. First, we present the theoretical comparison in Table 1. For consistency, $n$ represents the number of attributes owned by each participant. Additionally, $v$ and $w$ in our IP-PMA are equal to $n$ (that is, $v = w = n$), and $d$ denotes the intersection threshold. Note that the core PMA phase only considers the computation overhead generated by one side (e.g., A) because of symmetry.

From Table 1, we notice that $2n+3$ modular exponentiation operations are required in the Setup phase in the ABK07 scheme [3]. In the ABK07 scheme, each claimed group/organization is essentially identified by attribute representations. The CreateGroup algorithm is invalid for the ABK07 scheme, where each

1) http://crypto.stanford.edu/pbc/.

**Table 1** Comparison of related PMA schemes

| | ABK07 [3] | WZ14 [5] | IP-PMA |
|---|---|---|---|
| Setup | $(2n+3)T_{\mathrm{e}}$ | – | – |
| CreateGroup | – | $T_{\mathrm{e}}$ | $T_{\mathrm{sm}}$ |
| AddMember | $n(n+6)T_{\mathrm{e}}$ | $n(n+6)T_{\mathrm{e}}$ | $n(T_{\mathrm{sm}}+T_{\mathrm{H}})$ |
| PMA | $(2d+1)T_{\mathrm{p}}$ $+(n(n+4)+d+2)T_{\mathrm{e}}$ | $(2d+2)T_{\mathrm{p}}$ $+(n(n+4)+4n+d+2)T_{\mathrm{e}}$ | $2T_{\mathrm{sm}}+(3n+1)T_{\mathrm{p}}$ $+(2n+1)T_{\mathrm{e}}+(n+1)T_{\mathrm{H}}$ |
| Communication complexity | $2(n+1)\lvert\mathbb{G}_1\rvert$ | $6n\lvert\mathbb{G}_1\rvert+2\lvert\mathbb{Z}_q\rvert$ | $2(\lvert\mathbb{G}\rvert+n\lvert\kappa\rvert)$ |
| Intersection computation | Need prepared | Need prepared | No need prepared |

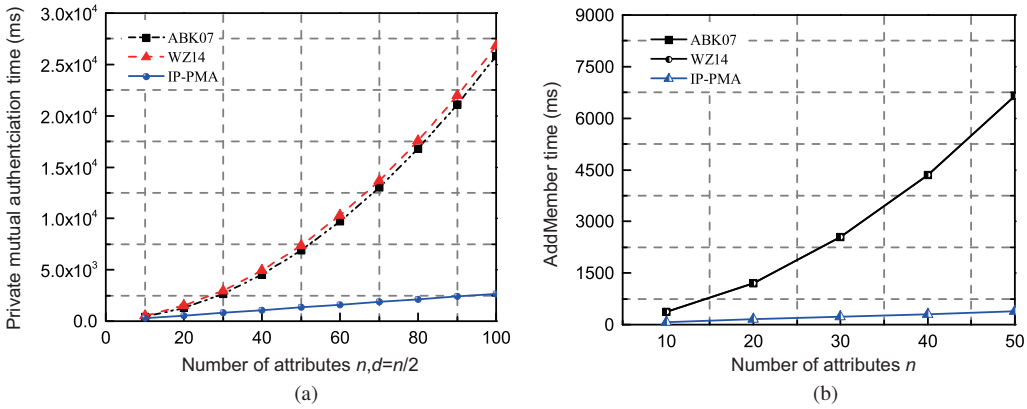


**Figure 3** (Color online) Performance trends of computation costs. (a) PMA; (b) AddMember.

group cannot be managed independently without its group key pair. Specifically, the group public/secret keys are issued in the Setup phase, where the same $n+2$ modular exponentiations are required because of different groups. Therefore, the notion of group in the ABK07 scheme cannot make a clear distinction. All members from different groups are indeed subordinate to the same group manager originated from the Setup phase. For the WZ14 scheme and our proposed IP-PMA, different groups are separate and have respective group public and private keys. Hence, the computation costs of WZ14 and IP-PMA schemes are reduced in the Setup phase, whereas the CreateGroup phase needs corresponding operations for the generation of a group public key.

Derived from the presented IBE-APSI, our proposed IP-PMA also realizes the intersection policies for private mutual authentications. For both the ABK07 and WZ14 schemes, the secure protocols for set intersection must be prepared and executed before the handshake protocol is run. Similar to black boxes, the corresponding computation and communication costs for PSI are not easy to measure in terms of the performance comparison that should actually be taken into account. As stated in the ABK07 scheme [3], the prepared set intersection protocol leaks the results of two-party secret handshakes. Even if the participants identify that their attributes cannot be matched after running the prescribed set intersection, they still have to continue the handshake protocol by sending random values to ensure the indistinguishability to eavesdroppers property. For our proposed protocol, the set intersection and PMA are simultaneously implemented by applying the IBE-APSI protocol.

According to the evaluations in Table 1, we notice that the computation costs of both the ABK07 and WZ14 schemes in the PMA phase are quadratic ($O(n^2)$). In more detail, we illustrate the increasing trends of the computation costs by varying the number of attributes in the three protocols in Figure 3. For simplicity, we set the threshold value to $d=n/2$. In Figure 3(a), the varying curves of the computation costs in the PMA phase demonstrate that our proposed protocol requires the minimum time consumption. For instance, when $n=50$, the time costs of PMA in the ABK07 and WZ14 schemes are approximately 6876 and 7365 ms, respectively, whereas that of IP-PMA is approximately 1348 ms. Additionally, the increasing trend of the computation overhead in the AddMember phase in Figure 3(b) also demonstrates that our proposed protocol achieves the best efficiency. Furthermore, we also evaluate the communication

overhead of the aforementioned three schemes by setting $n = 10, |\mathbb{G}| = |\mathbb{G}_1| = 1024$ bits, and $|\mathbb{Z}_q| = |\kappa| = 160$ bits. Thus, the experimental results show that our proposed protocol only requires approximately 0.64 K bytes bandwidth complexity, whereas the ABK07 and WZ14 schemes demand approximately 2.6 K bytes and 7.5 K bytes communication overhead, respectively. Therefore, the new IP-PMA protocol performed better when the number of elements was increased in each partner's attribute set, which is more suitable for resource-constrained applications.

## 7 Conclusion

In this paper, we proposed an efficient IP-PMA protocol that can support a multi-attribute intersection policy. To achieve better efficiency and forward security, first, an IBE-APSI protocol was optimized. The IP-PMA protocol derived from the new forward-secure IBE-APSI protocol greatly reduced the computation time of two participating parties. Furthermore, our proposed protocol did not have to run an additional PSI protocol before executing a real PMA protocol. The new protocol integrates the intersection computation of the attributes of two participants with a secret handshake scheme to achieve multi-attribute matching. Challenging future research is to explore more flexible and widely used application requirements, such as supporting multiple GAs, multi-party intersection and private authentication. Additionally, it would be interesting to design efficient protocols adapted to big data analysis or data mining by combing PSI with machine learning.

**Acknowledgements** This work was supported by National Natural Science Foundation of China (Grant Nos. 61672550, 61572028, 61300204), National Key R&D Program of China (Grant No. 2017YFB0802503), National Cryptography Development Fund (Grant No. MMJJ20180206), National Social Science Foundation of China (Grant No. 14BXW031), Natural Science Foundation of Guangdong (Grant Nos. 2019A1515011797, 2016A030310027, 2014A030313609, 2018A030313954), Project of Science and Technology of Guangzhou (Grant No. 201802010044), State Scholarship Fund of China Scholarship Council (CSC) (Grant No. 201808440097), and Research Team of Big Data Audit from Guangdong University of Finance and Economics.

**References**

1 Jarecki S, Liu X M. Private mutual authentication and conditional oblivious transfer. In: Proceedings of the 29th Annual International Cryptology Conference, Santa Barbara, 2009. 90–107
2 Balfanz D, Durfee G, Shankar N, et al. Secret handshakes from pairing-based key agreements. In: Proceedings of IEEE Symposium on Security and Privacy, Berkeley, 2003. 180–196
3 Ateniese G, Blanton M, Kirsch J. Secret handshakes with dynamic and fuzzy matching. In: Proceedings of Network and Distributed System Security Symposium, 2007. 159–177
4 Sahai A, Waters B. Fuzzy identity-based encryption. In: Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques (Advances in Cryptology - EUROCRYPT), St. Petersburg, 2005. 457–473
5 Wen Y M, Gong Z. Private mutual authentications with fuzzy matching. Int J High Performance Syst Archit, 2014, 5: 3–12
6 Freedman M, Nissim K, Pinkas B. Efficient private matching and set intersection. In: Proceedings of the 23th International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT), Interlaken, 2004. 1–19
7 El Defrawy K, Faber S. Blindfolded data search via secure pattern matching. Computer, 2013, 46: 68–75
8 de Cristofaro E, Tsudik G. Practical private set intersection protocols with linear complexity. In: Proceedings of the 14th International Conference on Financial Cryptography and Data Security, Tenerife, 2010. 143–159
9 de Cristofaro E, Faber S, Gasti P, et al. Genodroid: are privacy-preserving genomic tests ready for prime time? In: Proceedings of the 11th Annual ACM Workshop on Privacy in the Electronic Society, Raleigh, 2012. 97–108
10 Baglioni E, Becchetti L, Bergamini L, et al. A lightweight privacy preserving SMS-based recommendation system for mobile users. Knowl Inf Syst, 2014, 40: 49–77
11 Guan Z T, Zhang Y, Zhu L H, et al. EFFECT: an efficient flexible privacy-preserving data aggregation scheme with authentication in smart grid. Sci China Inf Sci, 2019, 62: 032103
12 Miao Y B, Ma J F, Liu X M, et al. Practical attribute-based multi-keyword search scheme in mobile crowdsourcing. IEEE Internet Things J, 2018, 5: 3008–3018
13 Miao Y B, Ma J F, Liu X M, et al. Attribute-based keyword search over hierarchical data in cloud computing. IEEE Trans Serv Comput, 2017. doi: 10.1109/TSC.2017.2757467

14 Miao Y B, Ma J F, Liu X M, et al. Lightweight fine-grained search over encrypted data in fog computing. IEEE Trans Serv Comput, 2018. doi: 10.1109/TSC.2018.2823309

15 He D B, Wang D, Xie Q, et al. Anonymous handover authentication protocol for mobile wireless networks with conditional privacy preservation. Sci China Inf Sci, 2017, 60: 052104

16 Castelluccia C, Jarecki S, Tsudik G. Secret handshakes from CA-oblivious encryption. In: Proceedings of the 10th International Conference on the Theory and Application of Cryptology and Information Security, Jeju Island, 2004. 293–307

17 Zhou L, Susilo W, Mu Y. Three-round secret handshakes based on ElGamal and DSA. In: Proceedings of the 2nd International Conference on Information Security Practice and Experience, Hangzhou 2006. 332–342

18 Vergnaud D. RSA-based secret handshakes. In: Proceedings of International Workshop on Coding and Cryptography, Bergen, 2005. 252–274

19 Jarecki S, Kim J, Tsudik G. Beyond secret handshakes: affiliation-hiding authenticated key exchange. In: Proceedings of the Cryptographers' Track at the RSA Conference, San Francisco, 2008. 352–369

20 Wen Y M, Zhang F G, Xu L L. Secret handshakes from ID-based message recovery signatures: a new generic approach. Comput Electrical Eng, 2012, 38: 96–104

21 Wen Y M, Zhang F G, Xu L L. Unlinkable secret handshakes from message recovery signature. Chin J Electron, 2010, 19: 705–709

22 Huang H, Cao Z F. A novel and efficient unlinkable secret handshakes scheme. IEEE Commun Lett, 2009, 13: 363–365

23 Su R W. On the security of a novel and efficient unlinkable secret handshakes scheme. IEEE Commun Lett, 2009, 13: 712–713

24 Gu J, Xue Z. An improved efficient secret handshakes scheme with unlinkability. IEEE Commun Lett, 2011, 15: 486–490

25 Jarecki S, Liu X. Unlinkable secret handshakes and key-private group key management schemes. In: Proceedings of the 5th International Conference on Applied Cryptography and Network Security, Zhuhai, 2007. 270–287

26 Kawai Y, Yoneyama K, Ohta K. Secret handshake: strong anonymity definition and construction. In: Proceedings of the 5th International Conference on Information Security Practice and Experience, 2009. 219–229

27 Wen Y M, Zhang F G. A new revocable secret handshake scheme with backward unlinkability. In: Proceedings of the 10th European Workshop on Public Key Infrastructures, Services and Applications, Athens, 2010. 17–30

28 Jarecki S, Kim J, Tsudik G. Group secret handshakes or affiliation-hiding authenticated group key agreement. In: Proceedings of the Cryptographers' Track at the RSA Conference, San Francisco, 2007. 287–304

29 Sorniotti A, Molva R. A provably secure secret handshake with dynamic controlled matching. Comput Secur, 2010, 29: 619–627

30 Sorniotti A, Molva R. Federated secret handshakes with support for revocation. In: Proceedings of the 12th International Conference on Information and Communications Security, Barcelona, 2010. 218–234

31 Hou L, Lai J Z, Liu L X. Secret handshakes with dynamic express matching policy. In: Proceedings of the 21st Australasian Conference on Information Security and Privacy, 2016. 461–476

32 Wen Y M, Gong Z. A dynamic matching secret handshake scheme without random oracles. In: Proceedings of the 8th International Conference on Network and System Security, Xi'an, 2014. 409–420

33 Lu R X, Lin X D, Liang X H, et al. A secure handshake scheme with symptoms-matching for mhealthcare social network. Mobile Netw Appl, 2011, 16: 683–694

34 He D B, Kumar N, Wang H Q, et al. A provably-secure cross-domain handshake scheme with symptoms-matching for mobile healthcare social network. IEEE Trans Dependable Secure Comput, 2018, 15: 633–645

35 Tian Y G, Zhang S W, Yang G M, et al. Privacy-preserving k-time authenticated secret handshakes. In: Proceedings of the Australasian Conference on Information Security and Privacy (ACISP 2017), Auckland, 2017. 281–300

36 Tian Y G, Li Y J, Zhang Y H, et al. DSH: deniable secret handshake framework. In: Proceedings of the 14th International Conference on Information Security Practice and Experience (ISPEC 2018), Tokyo, 2018. 341–353

37 Ateniese G, Francati D, Nuñez D, et al. Match me if you can: matchmaking encryption and its applications. https://eprint.iacr.org/2018/1094

38 Boneh D, Franklin M. Identity-based encryption from the weil pairing. In: Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology - CRYPTO, Santa Barbara, 2001. 514–532

39 de Cristofaro E, Jarecki S, Kim J, et al. Privacy-preserving policy-based information transfer. In: Proceedings of the 9th International Symposium on Privacy Enhancing Technologies, Seattle, 2009. 164–184

40 de Cristofaro E, Kim J, Tsudik G. Linear-complexity private set intersection protocols secure in malicious model. In: Proceedings of the 16th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT), Singapore, 2010. 213–231