Privacy-Preserving Friend Search over Online Social Networks

Huang Lin⋆, Sherman S. M. Chow‡, Dongsheng Xing†, Yuguang Fang⋆ and Zhenfu Cao†
⋆ University of Florida
‡ University of Waterloo
† Shanghai Jiao Tong University

***Abstract*—Friendships or social contacts represent an important attribute characterizing one's social position and significantly impact one's daily life. Over online social networks (OSNs), users may opt to hide their social circle, membership or connections to certain individuals or groups for privacy concern. On the other hand, this prohibits a major benefit of OSNs – building social connections. In order to enable OSN users to search for contacts they interested and leverage friends-of-friends relationship to grow their social network, we study the following privacy-preserving profiles searching (PPPS) problem: user P_1 wants to seek for contacts possessing a certain set of attributes from the contacts of P_2, while the contacts of P_2 remain hidden from P_1 and the criteria of P_1 is unknown to P_2 unless P_2 indeed having such contacts.**

While the PPPS problem can be solved with the help of oblivious transfer with hidden access control (OT-HAC) which in turn can be built by anonymous identity-based encryption (IBE) with blind key extraction (BKE) protocol, the designs of existing systems are often complicated and the efficiency are not satisfactory. A simple and efficient approach is especially important for P_2 since he is playing a helping role in the protocol.

In this paper, we propose efficient BKE protocols, for an anonymous IBE and an anonymous hierarchical IBE attributed to Ducas in CT-RSA '10. Our protocol for IBE is conceptually simpler and more efficient than an existing proposal by Camenisch *et al.* in PKC '09. Our protocol for HIBE is the first of its kind in the literature to the best of our knowledge. When compared with the OT-HAC system proposed by Camenisch *et al.* in PKC '11, our protocol is again conceptually simpler, supports predicates defined by vectors from a large-domain instead of bit-vectors, and allows retrieval of multiple items in one invocation. Finally, we demonstrate their practicality by our performance analysis on prototypes implementation.

I. Introduction

In our modern society, one's social contacts or associations or friendship generally play an important role in his/her life. One usually actively develops certain close relationships with others through direct personal interactions or friends' references in order to expand one's social circle. With the emergence of online social networks (OSNs), certain social relationship starts to change and even friendship has to be redefined because friendship in cyberspace, namely, *virtual friendship*, tends to be different from that in the real world. Virtual friends may be formed due to common interests although they may not know or trust each other as in real world. The virtual friendship tends to be formed either through personal interactions over the OSNs or references from virtual friends with less required trust relationship, and thus the so formed social networks consists of virtual friends with certain similarity in terms of certain attributes such as common interests. As a result, sometimes one could be easily identified by simply observing the social circles he/she is involved with. To some extent, an individual over OSNs can be easily identified by his/her social relationships, which may reveals certain private information about an individual. For example, an individual appearing on the friend list of an AIDS doctor may be suspected to be a patient, a person appearing on some suspected terrorist groups may be suspected to be a terrorist with high probability. With the ever increasing popularity of OSNs, how to effectively protect people's privacy has been raised as an important issue because people may share sensitive data without aware of the potential leak in OSNs. It is important to notice that a person's social contact structure has already been utilized to de-anonymize one's identity lately [4], [32], [33] or invade one's privacy [15]. In order to protect the privacy of one's social contacts, it has been suggested [3], [24] that one's social contacts should be encrypted rather than just be published openly.

Most of current OSNs have already provided some kinds of mechanism to protect one's social contacts. In Facebook or LinkedIn, users can choose not to reveal their social contacts to the strangers. Take a step further, it may be more desirable or flexible if users can choose which part of their social contacts can be revealed to whom. Users may classify their social circles into several sub-cirques according to the intimacy extent. For examples, one might consider his family members and old friends as the closest friends, his classmates or colleagues in the second outer social cirque, and those he has just met recently in the most outer cirque. Since "trust" has not yet fully developed with newly met friends, one might want to hide new friends from the old friends too.

One's friends may be familiar with his/her friends in the same cirque rather than those in different cirques. For example, one's classmates are familiar with each other but they might not know each other's family members at all. So a "natural" privacy setting is that the friendship status among the same cirque can be "public" within the cirque, but should be hidden from people not in the same cirque. This would hinder one of the major benefits of social networking which is for the expansion of social network. An analog is that a new member of many secret societies such as Skull and Bones at Yale University must be introduced by the senior member of the club. Here, we observe that friendship between A and B and friendship between B and C can play important role in the

friendship establishment between A and C. A might be a junior looking for advice from a trustworthy senior, and B might be a person who owns a certain private social resource (which might not be known to A at all since A and these private friends might not be in the same cirque as previously discussed). A may not want to reveal whom he/she is searching for (a person with HIV may not want to reveal that he/she is searching for an AIDS doctor). It would be ideal for A if there is a mechanism to guarantee that B has the requested friend A is searching for before A reveals to B whom A is really searching for.

We abstract the above privacy-preserving profiles searching (PPPS) problem as follows: P_2 has a list of encrypted profiles (of his/her friends). P_1, who is searching for a certain type of profile, contacts P_2 and checks whether P_2 has the profile P_1 is searching for, and both are only willing to reveal the minimum information to each other during this search process. In other words, P_1 is only willing to reveal his/her target profile when he/she is guaranteed that P_2 indeed has one that P_1 is searching for. P_2 will only send to P_1 the target profile when P_2 is willing to help. P_2 wishes not to reveal any information on his/her private friend list to P_1 if he/she decides not help or does not have the profile on his/her friend list. The privacy requirement is somehow counterintuitive: since without P_2 telling P_1 whether he/she has the target friend in advance, it is generally impossible for P_1 to send private information to P_2 based on the fact that they do not know each other practically. In this paper, we provide a mechanism that can enable P_1 to reveal his/her target profile type to P_2 only when P_2 does have the target profile and is willing to help while guaranteeing that P_1 does not gain any extra information on P_2's friend list during this process.

A. Oblivious Transfers

The PPPS problem is closely related to the notion of oblivious transfer with hidden access control (OT-HAC) [7]. An OT protocol enables a receiver to obtain one of many pieces of information (or an item) from a sender, but the sender remains oblivious as to which piece has been transferred. The basic notion of OT considers all items are of the same "kind" such that the receiver specifies the interested item by an index that may not have anything to do with the individual characteristics of each item to be obtained. One may also associate some attributes to each item so that the receiver can look up the list and locate the index of the item to be retrieved. The access control part of an OT protocol can enforce the requirement that only the items that the receiver is legitimate to get can possibly be transferred via the OT protocol. The rules here are defined upon the attributes describing each items.

The list of attributes, which determines the access control policy, should be known to the receiver in the aforementioned scenario. OT-HAC is an OT protocol such that the access control policy is hidden from the receiver. Now we are ready to discuss the similarities between OT-HAC and the PPPS problem. P_1 and P_2 can act as a receiver and a sender of an OT-HAC invocation respectively. The items being transferred is P_2's friends list, and the access control policy is the attributes describing P_2's friends, which should be hidden from P_1.

There are also a number of major differences. To make our discussion concrete, here we consider the OT-HAC system proposed by Camenisch *et al.* [7]. First, our PPPS scenario does not need the access control mechanism, The "permission" of a receiver is merely dependent by his/her own interests. On the other hand, a receiver of an OT-HAC system always have some kind of credential to be verified. Consequently, we do not need to deal with the credential revocation issue that was addressed in [7]. Second, their system aims to restrict the receiver to get at most one item per protocol invocation, while it is more natural to allow the receiver to get all matching profiles in one shot. Finally, there is a difference which may not be related to the functionalities provided by cryptographic means. In the OSN scenario, we believe that the one who introduces friends to others should have the right to know and the final say whether or who should be introduced. Even though it is not the case and with OT the "sharing" of profiles can be done in an oblivious way. It may be quite obvious shortly afterward after the protocol (since they will become friends in the OSN) anyway.

There are quite a few cryptographic building blocks involved in the OT-HAC system of Camenisch *et al.* [7], namely, anonymous credential, which is in the form of a verifiable random function in their scheme, non-interactive zero-knowledge/witness-indistinguishable proof (Groth-Sahai proof system), and their re-randomization and extension. Each data item in their scheme is encrypted with an ElGamal-like encryption using the output of verifiable random function as the random padding (which means the encryption can only be done in a symmetric-key manner). Moreover, each of them is accompanied with a proof of ciphertext well-formness The receiver is required to re-randomize and extend this proof related to his/her credential during the OT. All of these help to ensure revocation can be done before each OT but also contribute to the complexities of their protocol. Finally, despite of the use of all these relatively-heavy cryptographic machinery, the predicate supported is testing whether a user have access to all categories specified in the access control policy.

B. Blind Identity-Based Encryption

Another efficient approach to construct OT is to utilize blind identity-based encryption The concept of blind IBE was introduced by Green and Hohenberger [19], which is an IBE equipped with a blind key extraction (BKE) protocol. BKE of an IBE aims to guarantee that a user can obtain the decryption key for an identity without letting the key issuer learning the identity. Consider that the data item in an OT application is encrypted by IBE under different identities (which represents a simple access control policy based on a single-field equality-checking), BKE helps the receiver to get one and only one private decryption key of the IBE system which fulfills the security requirement of the sender. The receiver's choice is also protected since the identity being requested is not leaked via the BKE protocol.

Unfortunately, all of the IBE schemes considered in their work fail to offer anonymity. This means that the access control policy of the encrypted items are not hidden. The first (and still the only one to the best of our knowledge) blind anonymous IBE, i.e., an anonymous IBE equipped with a BKE protocol, was proposed by by Camenisch *et al*. [8]. As acknowledged in their paper, the key structure is a bit complicated due to the design of the underlying anonymous IBE, which results in a BKE protocol which is not that efficient. On the other hand, Chow [10] considered a form of BKE protocol for another anonymous IBE with a simpler key structure. However, the blindness requirement is weaker than the selective-failure blindness required by the OT application. Recently, Lu and Tsudik [27] also proposed a blind key extraction protocol. However, they did not provide any evidence whether the underlying encryption scheme is anonymous.

II. Related Work

A. Searchable Encryption and Anonymous IBE

One may also rephrase our PPPS problem as a problem about searching among encrypted data. Public key searchable encryption is one such scheme which allows a data owner (i.e., the private hey holder) to delegate a searching trapdoor to an untrusted server for "searching" among the ciphertexts, without delegating the power of decryption. A simple form of searching is equality checking of keywords. Public-key encryption with keyword search (PEKS) can be built by using an identity based encryption (IBE) scheme with anonymity (or anonymous IBE) [1], [6]. A high-level correspondences between these two notions are as follows. Identity strings in IBE play the role of keywords in PEKS. A identity-based secret key is then a trapdoor for a particular keyword. To associate a keyword with a ciphertext, an encryptor builds an encrypted index which is an IBE encryption of a fixed string (say all 0's) under the keyword as the recipient identity. To test whether an encrypted index matches with the keyword associated with a trapdoor, trial decryption is done to see if decryption returns the fixed string. Since one should not be able to tell the keyword associated with an encrypted index without getting the corresponding trapdoor, that is why anonymity of IBE is crucial for the PEKS application. Many IBE schemes, such as [31], do not provide anonymity.

B. Private Similarities Discovery

There has been a large body of research on private social network dedicated to the designs or applications of private similarities discovery between two participants. [11], [13], [16], [18], [25] For examples, the trust implied by the social connections in social-network has been leveraged as a basis of sybil-resilient access control [28], [30]. Similarities here could refer to common interests, common friends, or common profile, etc. These similarities would be known to both participants after they run the private similarities discovery protocol. The privacy requirement of these protocols guarantees no extra information but the similarities is revealed to the participants. One commonly adopted underlying technique is private set intersection, such as the one by Freedman, Nissim and Pinkas [17]. A PSI scheme is a two-party protocol between a client C (Initiator) with an input set $X = \{x_1, \cdots, x_c\}$ and a server S with an input set $Y = \{y_1, \cdots, y_s\}$. At the end of the protocol, C learns the intersection of X and Y (i.e., $X \bigcap Y$) while S learns nothing.

PSI has been further generalized into several variants, e.g. enabling the initiator to obtain function on either the intersection or the union of the two input sets [23] or hiding the size of both inputs [2]. On the other hand, several recent results [12], [20]–[22], [34] have been focusing on improving the efficiency of PSI protocol that are secure in the malicious model.

Recently, two common friend discovery protocols based on PSI have been proposed [13], [25]. However, a major concern of these protocols are for privately providing the common contacts or friend profiles for both parties, which is quite different from our focus. Also, their security proof are either relying on random oracle or in the semi-honest mode. Comparably, our proposed protocols are proven secure against the malicious adversary under the standard model. While there exists PSI protocol (e.g., [21]) that is proven secure under the malicious adversary model without assuming random oracle, it requires a number of oblivious transfer invocations dependent on the input length (it will correspond to the length of id in our construction), which would lead to a much more communication overload compared with our scheme.

III. System model

A. System Overview

We first give a high level introduction to our system. P_1 is the initiating party who will send the "looking for friends" request to P_2 and starts the whole process. The input of P_1 would be a specification of his target friend type. The input of P_2 to the protocol is a private friend list L. P_1's goal is to get to be introduced to those in P_2's private friends list who match his requirement at the end of the protocol.

Each friend in list L will be assigned with a random index by P_2 at the beginning of the protocol. Each index will be encrypted under an attribute (or a set of attributes) using anonymous (hierarchical) IBE. The ciphertext will be published by P_2. The attributes here are basically a list of keywords of the friends' profile, such as "occupation: dentist", "age: 30", etc.

The key ingredient of our system is a blind key extraction protocol for the anonymous IBE system. P_2 will use his master key of the IBE as an input. After an invocation of the protocol, P_1 will receive m independently generated private key corresponding to the attribute he is interested while P_2 gets nothing.

A more detailed description of the key steps involved in our system are given in Fig. 1.

P_1 then uses the private key he receives to decrypt all the ciphertexts published by P_2, and returns the decryption results to P_2. Since all the information P_1 receives during the decryption are random indexes (the decryption is successful if a match between the specification of the private key and

the attribute set labeled with the ciphertext happens) or a random element (the decryption fails if no match happens). In either cases, P_1 only gets random information which reveals no information on list L. After receiving the decryption result, P_2 can decide whether he will introduce P_1 to the matching private friend(s). Suppose none of P_2's friends meets P_1's requirement, then all P_2 gets from P_1 are just m independent decryption results due to the unique randomness in each of P_1's m private keys. Per P_2's decision, P_1 will get the chance to know the correct person.

B. Network model

We assume the existence of a social network such as Facebook on top of the communication network. Our protocol assumes an underlying secure channel between two parties, which could be established easily when each of them owns a public/private key pairs, but does not require the coordination by a trusted third party. All parties carry out the matching protocol in a completely distributed fashion.

C. Adversary model

We are concerned about the attack launched by insider (i.e., the participants of the protocol, which are P_1 and P_2 in our description) since apparently it is much more powerful than those outsider attacks with much less useful information. Roughly speaking, we want to ensure that the attackers gain no extra useful information except those allowed by the protocol. In other words, P_1 only get to be introduced to the target friend when P_2 agrees to do the favor. Otherwise, P_1 will gain no extra information on P_2's private friend list. He will not even be able to distinguish the difference between the situation that P_2 cannot help or that P_2 is simply unwilling to help. P_2 only knows P_1's requirement when P_2 has the ability to help P_1, otherwise the only information P_2 gets is that P_2 does not have any friend P_1 wants.

As usual, the adversary for the proposed protocols can be divided into semi-honest (or honest-but-curious) and malicious adversaries. The semi-honest adversaries are assumed to follow their prescribed actions in the protocol. The malicious adversaries may behave arbitrarily during the protocol run. However, this does not prevent them from arbitrarily choosing private inputs before running the protocol or aborting the protocol prematurely. In other words, the participants in the proposed protocol could still act maliciously by claiming fake target friend type (for P_1) or fake private friend list L (for P_2). Indeed, these attacks can be weakened to some extent by adding some extra authentication mechanisms. However, this would imply a central authority, which is not desirable in our scenario.

IV. Preliminaries

A. Definitions for Blind and Anonymous IBE

In a standard IBE scheme, the key extraction algorithm is run by the key generation center (KGC) to return a secret key $\mathsf{sk}_{\mathsf{id}}$ for a user's identity id. In a blind IBE, extracting the secret key is completed in a blinded manner. The blinding action obscures the identity from the KGC.

Preparation:

P_2 runs the `Setup` algorithm of AIBE scheme and publishes the output public parameters param and stores the master key msk.

For $j = 1 : m$
P_2 runs $\mathtt{Enc}(\mathsf{param}, \mathsf{id}_j, M_j)$ to generate all the respective ciphertext C_j, where id_j would be the attribute corresponding to user j. M_j would be a random index chosen by P_2 from the message field of the encryption algorithm.
Endfor
P_2 publishes $C = \{C_j\}_{j=1}^m$.

Initialization:

P_1 sends "looking for friend" requests to P_2.
For $j = 1 : m$
P_1 and P_2 runs `BlindExt`.
The protocol takes attribute id' specified by P_1, the master key msk submitted by P_2, and the pubic parameters param as input, and outputs the private key $d_{\mathsf{id}'}^{(j)}$ for user P_1.
Endfor

Finding Target Friends

For $j = 1 : m$
P_1 uses $d_{\mathsf{id}'}^{(j)}$ and run $\mathtt{Dec}(d_{\mathsf{id}'}^{(j)}, C_j)$ decrypt C_j
If $\mathsf{id}' == \mathsf{id}_j$
then the decryption is successful, P_1 gets the correct index $M_j' = M_j$.
Else
then P_1 gets a random message M_j'.
Endif
Endfor
P_1 returns all the messages $\{M_j'\}_{j=1}^m$ to P_2.

For $j = 1 : m$
P_2 checks whether $M_j' = M_j$ holds
If $M_j' == M_j$
then P_2 decides whether he introduces P_1 to user j or not
If the answer is yes,
then P_2 introduces him to user j.
Endif
Endif
Endfor

If $M_j' \neq M_j$ for all j or all the P_2's above decision is negative **then** P_2 tells P_1 that he does not have the required friend.
Endif

Fig. 1. General Framework

A blind IBE scheme consists of all the algorithms Π of an IBE scheme, and the protocol $\texttt{BlindExt}(P_1(\mathsf{id}), P_2(\mathsf{msk}))$ generates the secret decryption key $\mathsf{sk}_{\mathsf{id}}$ for P_1's input identity id in an interactive key issuing protocol between P_1 and P_2. P_1's output is a decryption key $\mathsf{sk}_{\mathsf{id}}$ and the output of P_2 is empty. Otherwise both parties output $\perp$.

The security requirements for blind IBE consist of two main properties: leak freeness and selective-failure blindness. leak freeness requires that BlindExt is a secure two-party computation (2PC) that does not leak any more information than AIBEExtract. Selective-failure blindness requires that the BlindExt protocol does not reveal any information on the user's input identity to a potentially malicious key issuer. Besides, the key issuer cannot cause the BlindExt protocol to selectively fail depending on the user's choice of identity.

***Definition** 1 (Leak Freeness):* A BlindExt protocol is leak free if, for all efficient adversaries $\mathcal{A}$, there exists an efficient simulator $\mathcal{S}$ such that for every value κ, no efficient distinguisher D can determine whether it is playing Game $Real$ or Game $Ideal$ with non-negligible advantage.

Game $Real$: Run AIBESetup(1^κ). As many times as D wants, he picks $\mathcal{A}$'s input state. $\mathcal{A}$ then runs $\texttt{BlindExt}(\mathcal{A}(params, state), \text{KGC}(\mathsf{param}, \mathsf{msk}))$ with the KGC. $\mathcal{A}$ returns the resulting view to D.

Game $Ideal$: Run AIBESetup(1^κ). As many times as D wants, he picks an initial input state. $\mathcal{S}$ obtains (param, $state$) and may choose values id to query an oracle OIBE-Extract that knows msk. The oracle returns key $\mathsf{sk}_{\mathsf{id}} \leftarrow$ AIBEExtract(param, msk, id), otherwise $\perp$. $\mathcal{S}$ returns a simulated view to D.

***Definition** 2 (Selective-Failure Blindness):* We say a BlindExt protocol is selective-failure blind if every adversary $\mathcal{A}$ has a negligible advantage in the Game $SF-Blind$

Game $SF-Blind$: $\mathcal{A}$ outputs param and a pair of identities id_0, id_1. A random bit $b \in \{0,1\}$ is chosen, and $\mathcal{A}$ is given black-box access to two oracles: U(param, id_b) and U(param, id_{1-b}). The U algorithms produce sk_b, sk_{1-b} respectively. If $\mathsf{sk}_b = \perp$ and $\mathsf{sk}_{1-b} = \perp$, $\mathcal{A}$ receives (sk_0, sk_1); if only $\mathsf{sk}_{1-b} = \perp$, ($\varepsilon$, $\perp$); if only $\mathsf{sk}_b = \perp$, ($\perp$, ε); and if $\mathsf{sk}_b = \mathsf{sk}_{1-b} = \perp$, $\mathcal{A}$ receives ($\perp$, $\perp$). Finally, $\mathcal{A}$ outputs his guess b. The advantage of $\mathcal{A}$ in this game is $|Pr[b' = b] - 1/2|$.

***Definition** 3 (Secure Blind Anonymous IBE):* A blind anonymous IBE scheme (Π, BlindExt) is secure if and only if: (1) Π is a secure anonymous IBE scheme, and (2) BlindExt is leak free and selective-failure blind.

B. Asymmetric Pairings

Let p be a prime and let $\mathbb{G}$, $\hat{\mathbb{G}}$, and $\hat{\mathbb{G}}_t$ be groups of order p. An asymmetric pairing is a map $\hat{e} : \mathbb{G} \times \hat{\mathbb{G}} \rightarrow \hat{\mathbb{G}}_t$ which is bilinear, non-degenerate and efficiently computable. Asymmetric here refers to the fact that the groups $\mathbb{G}$ and $\hat{\mathbb{G}}$ are not the same group.

The leak-freeness of the proposed scheme relies on the following inverse SXDH assumption, which is given below.

***Definition** 4:* Inverse SXDH Assumption. Let $a, b, c \leftarrow \mathbb{Z}_p$ be chosen randomly, $\mathbb{G}$, $\hat{\mathbb{G}}$ be a cyclic group. The generator of these group is g, $\hat{g}$. The Inverse SXDH Assumption is that no probabilistic polynomial time algorithm $\mathcal{B}$ given

$$g, \hat{g}, A = g^a, \hat{A} = \hat{g}^a, \hat{B} = \hat{g}^{\frac{1}{b}}, \hat{C} = \hat{g}^c$$

can decide whether $c = ab$ or $c \in \mathbb{Z}_p$ with greater than a negligible advantage.

We assume there is no efficiently computable homomorphism from $\hat{\mathbb{G}}$ to $\mathbb{G}$ here since this would obviously make the above problem tractable. The adversary only needs to map $\hat{C}$ to C, and check whether pairing $\hat{e}(C, \hat{B}) = \hat{e}(g, \hat{A})$ or not.

C. Zero-Knowledge Proofs about Discrete Log.

This paper uses well-known techniques for proving statements about discrete logarithms, such as proof of knowledge of a discrete logarithm modulo a prime [29]. These protocols are secure under the discrete logarithm assumption. When referring to the proofs, we will use the notation introduced by Camenisch and Stadler [9]. For instance, $Pok\{(x, s) : y = g^x h^s\}$ denotes a zero-knowledge proof of knowledge (ZKPoK) of exponents x, s such that $y = g^x h^s$ holds. All values in the parenthesis, in this example x, s denote quantities whose knowledge to be proven, while all other values are known to the verifier. We note that compared with the existing blind and anonymous IBE system [8], we use far more simple ZKPoK and use them less often, which is one of the reasons why our proposed constructions improve the efficiency.

D. Review of Anonymous (H)IBE of Ducas

Our proposed system is based on blind and anonymous IBE scheme. our proposed blind anonymous IBE system is constructed upon Ducas' anonymous IBE system. Basically, the improvement of the blind and anonymous IBE system compared with the underlying anonymous IBE system is that the underlying **Extract** algorithm is replaced by a blind key extraction protocol. In this section, we briefly review Ducas' anonymous (H)IBE. We will show how the following algorithms are adapted as building blocks of our proposed system in the next section. We will show how to construct blind key extraction protocols in Sec. V.

Setup: To generate system parameters for an IBE, given bilinear groups ($\mathbb{G}$, $\hat{\mathbb{G}}$) with generators (g, $\hat{g}$), the setup algorithm first randomly picks $(\alpha, \beta, \gamma, \delta, \eta) \in \mathbb{Z}_p^5$, and sets: $\mathbb{G}_1 = g^\alpha$, $\mathbb{G}_2 = g^\beta$, $h = g^\gamma$, $f = g^\delta$, $t = g^\eta$, and their analogues: $\hat{g}_1 = \hat{g}^\alpha$, $\hat{g}_2 = \hat{g}^\beta$, $\hat{h} = \hat{g}^\gamma$, $\hat{f} = \hat{g}^\delta$, $\hat{t} = \hat{g}^\eta$. The public parameters param and the master secret msk are given by $\mathsf{param} = (g, g_1, h, f, t, \hat{g}, \hat{g}_2)$, $\hat{h} \in \hat{\mathbb{G}}^5 \times \hat{\mathbb{G}}^3$, $\mathsf{msk} = (\hat{g}_0) = \hat{g}^{\alpha\beta}$, $\hat{f}$, $\hat{t} \in \hat{\mathbb{G}}^3$.

$\texttt{Ext}(\mathsf{msk}, \mathsf{id})$: To extract a private key d_{id} for an identity $\mathsf{id} = I \in \mathbb{Z}_p^*$ picks random $r, R \in \mathbb{Z}_p$ and outputs $d_{\mathsf{id}} = \left(\hat{g}_0(\hat{h}^I \hat{f})^r \hat{t}^R, \hat{g}^r, \hat{g}^R\right) \in \hat{\mathbb{G}}^3$.

$\texttt{Enc}(\mathsf{param}, \mathsf{id}, M)$: To encrypt a message $M \in \hat{\mathbb{G}}_t$ under the identity $\mathsf{id} = I \in \mathbb{Z}_p^*$, pick a random $s \in \mathbb{Z}_p$ and output $C = \left(M \cdot \hat{e}(\mathbb{G}_1, \hat{g}_2)^s, g^s, (h^I f)^s, t^s\right) \in \hat{\mathbb{G}}_t \times \hat{\mathbb{G}}^3$

$\texttt{Dec}(d_{\mathsf{id}'}, C)$: To decrypt a ciphertext $C = (A, B, C_1, Z) \in \hat{\mathbb{G}}_t \times \hat{\mathbb{G}}^3$ using the private key $d_{\mathsf{id}'} = (d_0, d_1, d_2) \in \hat{\mathbb{G}}^3$, output $M = A \cdot \hat{e}(C_1, d_1) \cdot \hat{e}(Z, d_2) / \hat{e}(B, d_0) \in \mathbb{G}_t$.

The following is the HIBE scheme that is implied by the result of Ducas [14]. Firstly, the delegation ability is not imperative in our system, and hence we omit it from the below description (that means we will also omit the $Rerand$ algorithm given in [14]). A part of private key geared to delegation will also be omitted. Secondly, we note that after the part of private keys for delegation are omitted from the system, anonymity can be preserved in a simpler way as shown in the below description.

Setup: To generate system parameters for an HIBE system of depth ℓ, given bilinear groups ($\mathbb{G}$, $\hat{\mathbb{G}}$) with generators (g, $\hat{g}$), the setup algorithm first selects a random (α, β, γ, η)$\in \mathbb{Z}_p^4$, $\boldsymbol{\delta}\in \mathbb{Z}_p^\ell$, and sets: $\mathbb{G}_1 = g^\alpha$, $\mathbb{G}_2 = g^\beta$, $h = g^\gamma$, $\boldsymbol{f} = g^{\boldsymbol{\delta}}$, $t = g^\eta$, and their analogues: $\hat{g}_1 = \hat{g}^\alpha$, $\hat{g}_2 = \hat{g}^\beta$, $\hat{h} = \hat{g}^\gamma$, $\hat{\boldsymbol{f}} = \hat{g}^{\boldsymbol{\delta}}$, $\hat{t} = \hat{g}^\eta$. The public parameters param and the master secret msk are given by $\mathsf{param} = (g, \mathbb{G}_1, h, \boldsymbol{f}, t, \hat{g}, \hat{g}_2, \hat{h}) \in \hat{\mathbb{G}}^5 \times \hat{\mathbb{G}}^3$, $\mathsf{msk} = (\hat{g}_0) = \hat{g}^{\alpha\beta}$, $\hat{\boldsymbol{f}}$, $\hat{t} \in \hat{\mathbb{G}}^3$.

Ext(msk, $\mathbf{w}$): To extract a private key $d_{\mathbf{w}}$ for an identity $\mathbf{w}$ $=(w_1, \cdots, w_k)$, picks random $R \in \mathbb{Z}_p$, $\boldsymbol{r} \in \mathbb{Z}_p^k$ and outputs $d_{\mathbf{w}}=$

$$\left\{ d_0 = \hat{\mathbb{G}}_0 \prod_{i=1}^{k} \left(\hat{h}^{w_i} \hat{f}_i\right)^{r_i} \hat{t}^R, \{d_i = \hat{g}^{r_i}\}_{i=1}^k, d_{k+1} = \hat{g}^R \right\}$$

Enc(param, $\mathbf{v}$, M): To encrypt a message $M \in \hat{\mathbb{G}}_t$ under the public key $\mathbf{v}=(v_1, \cdots, v_k) \in \mathbb{Z}_p^*[k]$, pick a random $s \in \mathbb{Z}_p$ and output

$$C = \left(M \cdot \hat{e}(\mathbb{G}_1, \hat{g}_2)^s, g^s, \left\{ \left(\hat{h}^{v_i} \hat{f}\right)^s \right\}_{i=1}^k, t^s \right) \in \hat{\mathbb{G}}_t \times \hat{\mathbb{G}}^{2+k}$$

Dec($d_{\mathbf{w}}, C$): To decrypt a given ciphertext $C = (A, B, C_1, \cdots, C_k, Z) \in \hat{\mathbb{G}}_t \times \hat{\mathbb{G}}^{2+k}$ using the private key $d_{\mathbf{w}} = (d_0, d_1, \cdots, d_{k+1}) \in \hat{\mathbb{G}}^{k+2}$, output $M = A \cdot \hat{e}(C_1, d_1) \cdot \prod_{i=1}^k \hat{e}(C_i, d_i)/\hat{e}(B, d_0) \in \mathbb{G}_t$

V. Blind Key Extraction Protocol

The blind key extraction protocol for Ducas' anonymous IBE scheme is shown in Fig. 2.

A. Design of Our Protocol

User P_1 with the requested attribute id $= I$ and user P_2 with master key $\hat{g}_0, \hat{f}, \hat{t}$ (see Fig. 2) can jointly compute the private key for the attribute I. The private key is of the form $(\hat{g}_0 \left(\hat{h}^I \hat{f}\right)^{r+r_1} \hat{t}^{R+R_1}, \hat{g}^{r+r_1}, \hat{g}^{R+R_1})$, which means all the randomness $r+r_1$ and $R+R_1$ are jointly selected by both P_1 and P_2 without either of them knowing the other party's chosen random number. This is a standard practice used in current protocols [8].

The only existing blind protocol for an anonymous IBE [8] adopts two-party protocol (although for simple modular arithmetic) to realize this goal. Our construction does not require such a two-party protocol. Most of our performance gain can be attributed to this technical novelty. Besides, our protocol uses less and simpler ZKPoK, which also contributes to the efficiency improvement.

We use the notation PoK to represent the ZKPoK of the respective secret values in the following blind key extraction protocols. The corresponding statement has been omitted due to simplicity of expression. Most of these ZKPoK can be easily realized by the existing standard techniques as introduced in Sec. IV-C. A possible exception is one proving the knowledge of an exponent with respect to a secret pre-image ($\hat{f}$, $\hat{t}$) of a public asymmetric pairing value ($\hat{e}(f, \hat{g})$, $\hat{e}(f, \hat{t})$), which we will describe in Appendix of the full version [26]. We note that these PoK can be done in an interactive manner, instead of relying a hash function to generate the challenge value which will make us resort to the random oracle model instead.

The proof of the following theorem can be found in the full version.

***Theorem** 1:* Our blind extraction protocol provides a leak free and selective failure blind extraction protocol for anonymous IBE scheme under the Inverse SXDH Assumption.

Proof: For a corrupt user, our proof consists of constructing the following simulators:

Sim$_1$: When $\mathcal{A}$ engages $\mathcal{S}$ in a BlindExt protocol, $\mathcal{S}$ behaves as follows. The simulator randomly chooses $\hat{X}_1, \hat{X}_2 \in (\hat{G})^2$, and send them to the user.

In the next message of the protocol, $\mathcal{A}$ must send to $\mathcal{S}$ a value $\hat{h}_1$ and prove the knowledge of values ρ_1, I. If the proof fails to verify, $\mathcal{S}$ aborts. Otherwise the simulator $\mathcal{S}$ extracts ρ_1, I. It also receives from $\mathcal{A}$ the following values: $\hat{h}_2 = \hat{g}^{\rho_4} \hat{X}_1^{r_1}, \hat{h}_3 = \hat{g}^{\rho_5} \hat{X}_2^{R_1}$. Similarly, it extracts r_1, R_1, ρ_4, ρ_5. Then it will submit I to the trusted party, who returns the valid secret key $\hat{g}_0 \left(\hat{h}^I \hat{f}\right)^{r'} \hat{t}^{R'}, \hat{g}^{r'}, \hat{g}^{R'}$ for this identity. The simulator delivers

- $\hat{g}_0 \left(\hat{h}^I \hat{f}\right)^{r'} \hat{t}^{R'} \hat{g}^{r'\rho_1} / \hat{h}_1^{r_1} \hat{X}_3^{\rho_4} \hat{X}_4^{\rho_5}$
  $= \hat{g}_0 \left(\hat{g}^{\rho_1} \hat{h}^I \hat{f}\right)^{r'-r_1} \hat{t}^{R'-R_1} \hat{f}^{r_1} \hat{t}^{R_1} \hat{g}^{\frac{\rho_4}{\rho_2} + \frac{\rho_5}{\rho_3}}$,
- $\hat{g}^{r'-r_1} = \hat{g}^r$,
- $\hat{g}^{R'-R_1} = \hat{g}^R$,
- $\hat{X}_3 = \hat{g}^{\frac{1}{\rho_2}}$,
- $\hat{X}_4 = \hat{g}^{\frac{1}{\rho_3}}$

to the user, where $\hat{X}_3, \hat{X}_4$ are chosen randomly from $(\hat{G})^2$. We note that the distribution of this secret key will be distributed identically with the actual private key received by the user since the randomness r', R' will be distributed identically with $r + r_1, R + R_1$ in the real protocol. And here we assume the randomness chosen by P_2 is r and R, which are unknown to the simulator.

Sim$_2$: This simulator is different from the first simulator in the sense that $\hat{X}_1, \hat{X}_3$ are generated as in the real protocol but $\hat{X}_2, \hat{X}_4$ are still distributed as in Sim$_1$, which are two independent random values. It proceeds without calling the trusted authority. It uses all the zero-knowledge simulators but the simulator for the last proof in the scheme.

Sim$_3$: This simulator is different from the second simulator in the sense that $\hat{X}_2, \hat{X}_4$ are also generated as in the real protocol.

Sim$_4$: This simulator is different from the third simulator in the sense that it uses all the zero-knowledge simulator.

The fourth simulator is clearly indistinguishable from the real protocol by the zero-knowledge property of the proof system.

P_2(param, msk) $\quad\quad$ P_1(param, id $= I$)

$\rho_2, \rho_3 \leftarrow (\mathbb{Z}_p)^2$

$\hat{X}_1 = \hat{f}^{\rho_2}, \hat{X}_2 = \hat{t}^{\rho_3}$

$\xrightarrow{\hat{X}_1, \hat{X}_2, PoK(\rho_2, \rho_3)}$

$\rho_1, \rho_4, \rho_5, r_1, R_1 \leftarrow (\mathbb{Z}_p)^5$

$\hat{h}_1 = \hat{g}^{\rho_1}\hat{h}^I, \hat{h}_2 = \hat{g}^{\rho_4}(\hat{X}_1)^{r_1}, \hat{h}_3 = \hat{g}^{\rho_5}(\hat{X}_2)^{R_1}$

$\xleftarrow{\hat{h}_1, \hat{h}_2, \hat{h}_3, PoK(\rho_1, I), PoK(\rho_4, r_1), PoK(\rho_5, R_1)}$

If the proof fails
to verify, abort;

$r, R \leftarrow (\mathbb{Z}_p)^2$

$$\begin{aligned} d_0 &= \hat{g}_0 \left(\hat{h}_1\hat{f}\right)^r \hat{t}^R (\hat{h}_2)^{\frac{1}{\rho_2}} (\hat{h}_3)^{\frac{1}{\rho_3}} \\ &= \hat{g}_0 \left(\hat{h}_1\hat{f}\right)^r \hat{t}^R \hat{f}^{r_1} \hat{g}^{\frac{\rho_4}{\rho_2}} \hat{t}^{R_1} \hat{g}^{\frac{\rho_5}{\rho_3}} \\ &= \hat{g}_0 \left(\hat{g}^{\rho_1}\hat{h}^I\hat{f}\right)^r \hat{t}^R \hat{f}^{r_1} \hat{t}^{R_1} \hat{g}^{\frac{\rho_4}{\rho_2}+\frac{\rho_5}{\rho_3}}, \\ d_1 &= \hat{g}^r, d_2 = \hat{g}^R, \hat{X}_3 = \hat{g}^{\frac{1}{\rho_2}}, \hat{X}_4 = \hat{g}^{\frac{1}{\rho_3}} \end{aligned}$$

$\xrightarrow{d_0, d_1, d_2, \hat{X}_3, \hat{X}_4}$

Compute

$$\begin{aligned} & \frac{d_0}{(d_1)^{\rho_1}(\hat{X}_3)^{\frac{1}{\rho_2}}(\hat{X}_4)^{\frac{1}{\rho_3}}} \hat{h}^{Ir_1} \\ = & \frac{\hat{g}_0\left(\hat{g}^{\rho_1}\hat{h}^I\hat{f}\right)^r \hat{t}^R \hat{f}^{r_1} \hat{t}^{R_1} \hat{g}^{\frac{\rho_4}{\rho_2}+\frac{\rho_5}{\rho_3}}}{\hat{g}^{r\rho_1}\hat{g}^{\frac{\rho_4}{\rho_2}+\frac{\rho_5}{\rho_3}}} \hat{h}^{Ir_1} \\ = & \hat{g}_0 \left(\hat{h}^I\hat{f}\right)^{r+r_1} \hat{t}^{R+R_1} \end{aligned}$$

Check whether

$$\begin{aligned} & \hat{e}\left(\hat{g}_0(\hat{h}^I\hat{f})^{r+r_1}\hat{t}^{R+R_1}, g\right) \\ & = \hat{e}\left(\hat{g}_1, \hat{g}_2\right)\hat{e}(h^I f, \hat{g}^{r+r_1})\hat{e}(t, \hat{g}^{R+R_1}) \end{aligned}$$

holds. If so, outputs

$\hat{g}_0 \left(\hat{h}^I\hat{f}\right)^{r+r_1} \hat{t}^{R+R_1}, \hat{g}^{r+r_1}, \hat{g}^{R+R_1}$.

Fig. 2. Our Blind Key Extraction Protocol for (Anonymous) IBE

For the indistinguishably of the first and second simulator, it can be reduced to the following assumption: Given $\hat{g}, A = g^a, \hat{A} = \hat{g}^a, \hat{B} = \hat{g}^{\frac{1}{b}}, \hat{C} = \hat{g}^c$, and we must decide whether $c = ab$ or $c \in \mathbb{Z}_p$. As a simulator, we set $f = A^\theta, \theta \in \mathbb{Z}_p$ (which implies that $\hat{f} = \hat{A}^\theta$) and choose the rest public parameters as in the real protocol. The simulator first computes $\hat{X}_1 = \hat{C}^{\theta\lambda}, \lambda \in \mathbb{Z}_p$ (implying that $\rho_2 = b\lambda$ if $c = ab$) and choose a random $\hat{X}_2 = \hat{t}^{\rho_3}$. We return them to the adversary. We receive the answer from the adversary as in the real protocol and extract $\rho_1, Ir_1, R_1, \rho_4, \rho_5$ from the corresponding proof. Finally, we set $\hat{X}_3 = \hat{g}^{\frac{1}{\rho_2}} = \hat{B}^{\frac{1}{\lambda}}$, randomly pick $\hat{X}_4$, and generate $d_0 = \hat{g}_0 \left(\hat{h}^I\hat{f}\right)^r \hat{t}^R \hat{g}^{\rho_1 r} \hat{f}^{r_1} \hat{t}^{R_1} \hat{X}_3^{\rho_4} \hat{X}_4^{\rho_5}$ together with $\hat{g}^r, \hat{g}^R, \hat{X}_3, \hat{X}_4$.

Assuming the underlying ZKPoK is secure, if $c = ab$, the values the adversary receives will be distributed as in Sim_2, otherwise they are distributed just as in Sim_1.

The indistinguishably of the second and third simulator can be established using an almost identical argument. Considering the following assumption: Given $\hat{g}, A = g^a, \hat{A} = \hat{g}^a, \hat{B} = \hat{g}^{\frac{1}{b}}, \hat{C} = \hat{g}^c$, and we must decide whether $c = ab$ or $c \in \mathbb{Z}_p$. We set $t = A^\theta, \theta \in \mathbb{Z}_p$, which implies that $\hat{t} = \hat{A}^\theta, \theta \in \mathbb{Z}_p$ and choose the rest public parameters as in the real protocol. The simulator first interacts with the adversary to extract ρ_1, I. Then we compute $\hat{X}_2 = \hat{C}^{\theta\lambda}, \lambda \in \mathbb{Z}_p$ and choose a random $\rho_3 \in \mathbb{Z}_p$ to generate. We return them to the adversary. We receive the answer from the adversary as in the real protocol and extract r_1, R_1, ρ_4, ρ_5 from the corresponding proof. Finally, we set $\hat{X}_4 = \hat{g}^{\frac{1}{\rho_3}} = \hat{B}^{\frac{1}{\lambda}}$, generate $\hat{X}_3 = \hat{g}^{\frac{1}{\rho_2}}$, and $\hat{g}_0(\hat{h}^I\hat{f})^r \hat{t}^R \hat{g}^{\rho_1 r} \hat{f}^{r_1} \hat{t}^{R_1} \hat{X}_3^{\rho_4} \hat{X}_4^{\rho_5}, \hat{g}^r, \hat{g}^R, \hat{X}_3, \hat{X}_4$.

Again, if $c = ab$, the values the adversary receives will be distributed as in Sim_3, otherwise they are distributed just as in Sim_2.

The indistinguishably between the fourth and third simulator is due to the zero-knowledge property of the proof system.

Now, for selective-failure blindness, we first observe that in this protocol, U sends to $\mathcal{A}$ a value $\hat{h}_1$ uniformly distributed in $\hat{G}$ and then performs a zero-knowledge proof of knowledge ρ_1, I. Suppose that $\mathcal{A}$ runs one or both of his oracles up to this point. Now, it is $\mathcal{A}$'s turn to speak, and at this point,

his views so far are computationally indistinguishable. Let's assume that $\mathcal{A}$ must now return two values $\hat{X}_1, \hat{X}_2 \in (\hat{G})^2$ to the first oracle. Suppose $\mathcal{A}$ chooses this pair using any strategy he wishes. Then, the oracle's response would be just as what U provides in the real protocol, which is $\hat{g}^{\rho_4}\hat{X}_1^{r_1}, \hat{g}^{\rho_5}\hat{X}_2^{R_1}$ and the zero-knowledge proof of knowledge $r_1, R_1\rho_4, \rho_5$. At this point $\mathcal{A}$'s view are still computationally indistinguishable. Now it's $\mathcal{A}$'s turn to provide $d_0, d_1, d_2, \hat{X}_3, \hat{X}_4$. $\mathcal{A}$ could choose these five values in any way he wishes. At the point $\mathcal{A}$ fixes on two values, he is able to predict the output sk_i of these oracles U(param, id_b) with non-negligible advantage as follows:

1) $\mathcal{A}$ does the proof of Step 4 internally with itself. If the proof fails, it records $\mathsf{sk}_0 = \perp$. Otherwise, the adversary temporarily records $\mathsf{sk}_0 = \mathtt{BlindExt}(\mathsf{param}, \mathsf{msk}, \mathsf{id}_0)$.
2) In turn, $\mathcal{A}$ generates different $d_0, d_1, d_2, \hat{X}_3, \hat{X}_4$ and executes a second proof of knowledge (again internally), now for the second oracle. It performs the same checks and recordings for sk_1 and id_1.
3) Finally the adversary predicts (sk_0, sk_1) if both $\mathsf{sk}_0 \neq \perp$ and $\mathsf{sk}_1 \neq \perp$. If both $\mathsf{sk}_0 = \perp$ and $\mathsf{sk}_1 = \perp$, output ($\perp$, $\perp$), if only $\mathsf{sk}_1 = \perp$, ($\perp$, ε), if only $\mathsf{sk}_0 = \perp$; and (ε, $\perp$).

This prediction is correct, because $\mathcal{A}$ is performing the same check as the honest U, and when both tests succeed, outputting a pair of valid secret keys obtained via BlindExtract (param, msk, id) as does U. But at a higher-level, note that if $\mathcal{A}$ is able to predict the final output of its oracles accurately, then $\mathcal{A}$'s advantage in distinguishing U(param, id_b) and U(param, id_{b-1}) is the same without this final output. Thus, all of $\mathcal{A}$'s advantage must come from distinguishing the earlier messages of the oracles. Since these oracles only send three uniformly random value $\hat{h}_1, \hat{h}_2, \hat{h}_3$, and then perform a ZKPoK about the representation of $\hat{h}_1, \hat{h}_2, \hat{h}_3$ with respect to the values known to the adversary. Therefore, the security of the underlying proof guarantees that $\mathcal{A}$ cannot distinguish between them with non-negligible probability. ■

B. Hierarchical Extension

Now we extend our blind key extraction protocol for the anonymous IBE scheme of Ducas to its hierarchical extension (Fig. 3) working with the hierarchical IBE we described in previous section.

In our context, we use the identity to describe the attribute of the profiles being queried. Utilizing HIBE we can support AND policy of multiple fields via the multiple identities in the identity-vector of HIBE. We notice that this is in the same spirit as the policy supported by the OT system of Camenisch *et al*. [7] which is also about AND policy. However, due to the nature of the underlying HIBE scheme we used, our privacy guarantee is weaker in the sense that privacy is only preserved for queries involving the same number of fields. The weakness can be easily seen from the fact that the length of the ciphertext is directly proportion to the number of fields specified during encryption. Nevertheless, we remark that our protocol can be used in other applications requiring a blind key extraction protocol for IBE, such as supporting conjunctive searches in privacy-preserving delegated forensic search with authorization [8], [10], which was posed as an open problem in [8].

The blind key extraction protocol for HIBE essentially involves multiple instances of the sub-protocol involved for each identity in our blind key extraction protocol for IBE. Following the strategy in proving the IBE one we can derive the following theorem.

***Theorem** 2:* Our blind extraction protocol provides a leak free and selective failure blind extraction protocol for HIBE scheme.

VI. Performance Analysis

We implemented prototypes of our proposed blind anonymous IBE system and the existing one by Camenisch *et al*. [8] in C. We used the Pairing-Based Cryptography (PBC) Library. The curve we used for the existing scheme is type F. A curve of such type has the form of $y^2 = x^3 + b$. The curve we used for our proposed scheme is type D (which corresponds to the same c159 curve used in, for example, [5]) due to the restriction on homomorphism. The curve of such type has the form of $y^2 = x^3 + ax + b$. The order of both curves is around 160 bits, as is $\mathbb{F}_q$, the base field. We choose an identity id of length 160 bits. For our experiments, we used a desktop machine with an Intel Celeron 530 1.73GHZ CPU and 1GB of RAM, running Linux/Ubuntu 6.10. All the timing reported below are averaged over 100 randomized runs. We note that all the implementation was completed on one desktop, and hence the below timing report does not represent the comparison between the communication cost of the implementation of two frameworks.

We measured the efficiency of three algorithms/protocol for the blind anonymous IBE system: the encryption, decryption algorithms and blind key extraction protocol. The efficiency of all the algorithms or protocol for the existing blind anonymous IBE [8] depends on n, the number of blocks composed of ℓ bit integers each identity $\mathsf{id} \in \{0,1\}^{\ell \times n}$ is chopped into. Our proposed system does not have such restriction. But their IBE system achieves adaptive security while our system does not. The key extraction protocol is the most time-consuming component in the whole system. Our proposed key extraction protocol is more efficient than the existing one. When the length of identity is 160 bits, it takes only 0.0793079s on average for our proposed scheme to finish while it takes 0.456886s for the existing one to complete. We believe that the efficiency gain should be more remarkable if the communication cost is also counted. The existing scheme already has one extra communication round without even counting the additional communication rounds due to ZKPoK protocol. This efficiency gain can mainly be attributed to three reasons. First, we do not use any generic 2PC protocol since it is both communication and computation costly. It takes two communication rounds and three complicated ZKPoK protocol runs to complete a run of the underlying 2PC protocol. Second, there are totally five ZKPoK runs with complicated statement in the existing scheme while there are only three ZKPoK protocol runs with much simpler statement in our proposed system. Third, we

P_2(param, msk) P_1(param, **w**)

For i=1: k
$\rho_i \leftarrow \mathbb{Z}_p$
$\hat{h}_i = \hat{g}^{\rho_i}\hat{h}^{w_i}$

$\xleftarrow{\{\hat{h}_i, PoK(\rho_i, w_i)\}_{i=1}^{k}}$

If the proof fails for any i
to verify, abort;
$\sigma_1, \cdots, \sigma_k, \sigma \leftarrow (\mathbb{Z}_p)^{k+1}$
$\hat{X}_1 = \hat{f}_1^{\ \sigma_1}, \cdots, \hat{X}_k = \hat{f}_k^{\ \sigma_k}, \hat{X} = \hat{t}^{\sigma}$

$\xrightarrow{\hat{X}, Pok(\sigma), \{\hat{X}_i, PoK(\sigma_i)\}_{i=1}^{k}}$

For i=1: k
$\mu_i, r_i \leftarrow \mathbb{Z}_p$,
$\hat{h}_{i+k} = \hat{g}^{\mu_i}\hat{f}_i^{\ \sigma_i r_i}$
$\mu, R \leftarrow (\mathbb{Z}_p)^2$
$\hat{h}_{2k+1} = \hat{g}^{\mu}\hat{t}^{\sigma R}$

$\xleftarrow{\{\hat{h}_i\}_{k+1}^{2k+1}, PoK(\mu, \mu_1, \cdots, \mu_k, r_1, \cdots, r_k, R)}$

For i=1:k
$\hat{h}_{i+k}^{\frac{1}{\sigma_i}} = \hat{g}^{\frac{\mu_i}{\sigma_i}}\hat{f}_i^{\ r_i}$,
$r_i' \in (\mathbb{Z}_p)^2$
$R' \in \mathbb{Z}_p$
$\hat{h}_{2k+1}^{\frac{1}{\sigma}} = \hat{g}^{\frac{\mu}{\sigma}}\hat{t}^{R}$
$d_0 = \hat{\mathbb{G}}_0 \prod_{i=1}^{k}\left(\hat{h_i}\hat{f}_i\right)^{r_i'}\hat{t}^{R'}\hat{t}^{R}\prod_{i=1}^{k}\hat{f}_i^{\ r_i}\hat{g}^{\frac{\mu_i}{\sigma i}}\hat{g}^{\frac{\mu}{\sigma}}$
$= \hat{\mathbb{G}}_0 \prod_{i=1}^{k}\left(\hat{g}^{\rho_i}\hat{h}^{w_i}\hat{f}_i\right)^{r_i'}\hat{t}^{R'}\hat{t}^{R}\prod_{i=1}^{k}\hat{f}_i^{\ r_i}\hat{g}^{\frac{\mu_i}{\sigma i}}\hat{g}^{\frac{\mu}{\sigma}}$
$, d_1 = \hat{g}^{r_1'}, d_2 = \hat{g}^{r_2'}, \cdots, d_k = \hat{g}^{r_k'},$
$d_{k+1} = \hat{g}^{R'}$
$\hat{Y}_1 = \hat{g}^{\frac{1}{\sigma_1}}, \cdots, \hat{Y}_k = \hat{g}^{\frac{1}{\sigma_k}}, \hat{Y} = \hat{g}^{\frac{1}{\sigma}}$

$\xrightarrow{d_0, d_1, \cdots, d_{k+1}, \hat{Y}_1, \cdots, \hat{Y}_k, \hat{Y}}$

$\frac{d_0 \prod_{i=1}^{k}\hat{h}^{w_i r_i}}{\prod_{i=1}^{k} d_i^{\rho_i}\hat{Y}_i^{\mu_i}\hat{Y}^{\mu}}$
$= \hat{\mathbb{G}}_0 \prod_{i=1}^{k}\left(\hat{h}^{w_i}\hat{f}_i\right)^{r_i'+r_i}\hat{t}^{R'+R}$
Check whether
$\hat{e}\left(\hat{g}_0 \prod_{i=1}^{k}(\hat{h}^{w_i}\hat{f}_i)^{r_i'+r_i}\hat{t}^{R'+R}, g\right)$
$= \hat{e}\left(\hat{g}_1, \hat{g}_2\right)\prod_{i=1}^{k}\hat{e}\left(h^{w_i}f_i, \hat{g}^{r_i'+r_i}\right)\hat{e}\left(t, \hat{g}^{R'+R}\right)$
holds.
If it holds, then outputs
$\hat{g_0}\prod_{i=1}^{k}\left(\hat{h}^{w_i}\hat{f}_i\right)^{r_i'+r_i}\hat{t}^{R'+R}, \{\hat{g}^{r_i+r_i'}\}_{i=1}^{k}, \hat{g}^{R'+R}.$

Fig. 3. Our Blind Key Extraction Protocol for (Anonymous) HIBE

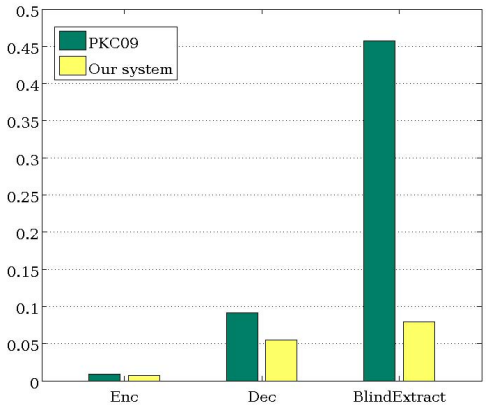

Fig. 4. Comparison of running time (s)

rely on SXDH assumption which is conjectured to hold for type D curve.

VII. Conclusions

This paper provides a general framework for finding friends of interest in a privacy preserving social network. We show the connection between this problem and the blind anonymous IBE scheme. This paper provides an efficient blind anonymous IBE scheme to improve the efficiency of the solution. We further generalize the concept of blind anonymous IBE to blind hierarchical IBE.

References

[1] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi. Searchable encryption revisited: Consistency properties, relation to anonymous ibe, and extensions. *J. Cryptology*, 21(3):350–391, 2008.

[2] G. Ateniese, E. D. Cristofaro, and G. Tsudik. (if) size matters: Size-hiding private set intersection. In *Public Key Cryptography*, pages 156–173, 2011.

[3] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, and D. Starin. Persona: an online social network with user-defined privacy. In *SIGCOMM*, pages 135–146, 2009.

[4] L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda. All your contacts are belong to us: automated identity theft attacks on social networks. In *WWW*, pages 551–560, 2009.

[5] D. Boneh and X. Boyen. Short signatures without random oracles and the sdh assumption in bilinear groups. *J. Cryptology*, 21(2):149–177, 2008.

[6] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano. Public key encryption with keyword search. In *EUROCRYPT*, pages 506–522, 2004.

[7] J. Camenisch, M. Dubovitskaya, G. Neven, and G. M. Zaverucha. Oblivious transfer with hidden access control policies. In *Public Key Cryptography*, pages 192–209, 2011.

[8] J. Camenisch, M. Kohlweiss, A. Rial, and C. Sheedy. Blind and anonymous identity-based encryption and authorised private searches on public key encrypted data. In *Public Key Cryptography*, pages 196–214, 2009.

[9] J. Camenisch and M. Stadler. Efficient group signature schemes for large groups (extended abstract). In *CRYPTO*, pages 410–424, 1997.

[10] S. S. M. Chow. Removing escrow from identity-based encryption. In *Public Key Cryptography*, pages 256–276, 2009.

[11] L. P. Cox, A. Dalton, and V. Marupadi. Smokescreen: flexible privacy controls for presence-sharing. In *MobiSys*, pages 233–245, 2007.

[12] E. D. Cristofaro, J. Kim, and G. Tsudik. Linear-complexity private set intersection protocols secure in malicious model. In *ASIACRYPT*, pages 213–231, 2010.

[13] E. D. Cristofaro, M. Manulis, and B. Poettering. Private discovery of common social contacts. In *ACNS*, pages 147–165, 2011.

[14] L. Ducas. Anonymity from asymmetry: New constructions for anonymous hibe. In *CT-RSA*, pages 148–164, 2010.

[15] C. Dwork. A firm foundation for private data analysis. *Commun. ACM*, 54(1):86–95, 2011.

[16] M. J. Freedman and A. Nicolosi. Efficient private techniques for verifying social proximity. In *IPTPS*, 2007.

[17] M. J. Freedman, K. Nissim, and B. Pinkas. Efficient private matching and set intersection. In *EUROCRYPT*, pages 1–19, 2004.

[18] S. Garriss, M. Kaminsky, M. J. Freedman, B. Karp, D. Mazières, and H. Yu. Re: Reliable email. In *NSDI*, 2006.

[19] M. Green and S. Hohenberger. Blind identity-based encryption and simulatable oblivious transfer. In *ASIACRYPT*, pages 265–282, 2007.

[20] C. Hazay and Y. Lindell. Efficient protocols for set intersection and pattern matching with security against malicious and covert adversaries. In *TCC*, pages 155–175, 2008.

[21] C. Hazay and K. Nissim. Efficient set operations in the presence of malicious adversaries. In *Public Key Cryptography*, pages 312–331, 2010.

[22] S. Jarecki and X. Liu. Efficient oblivious pseudorandom function with applications to adaptive ot and secure computation of set intersection. In *TCC*, pages 577–594, 2009.

[23] L. Kissner and D. X. Song. Privacy-preserving set operations. In *CRYPTO*, pages 241–257, 2005.

[24] M. E. Kounavis, X. Kang, K. Grewal, M. Eszenyi, S. Gueron, and D. Durham. Encrypting the internet. In *SIGCOMM*, pages 135–146, 2010.

[25] M. Li, N. Cao, S. Yu, and W. Lou. Findu: Privacy-preserving personal profile matching in mobile social networks. In *Infocom*, pages 1–8, 2011.

[26] H. Lin, S. S. M. Chow, D. Xing, Y. Fang, and Z. Cao. Privacy preserving friend search over online social networkss. In *Eprint*, pages 1–11, 2011.

[27] Y. Lu and G. Tsudik. Enhancing data privacy in the cloud. In *5th IFIP WG 11.11 International Conference on Trust Management (IFIPTM)*, June, 2011.

[28] A. Mohaisen, N. Hopper, and Y. Kim. Keep your friends close: Incorporating trust into social network-based sybil defenses. In *INFOCOM*, 2011.

[29] C.-P. Schnorr. Efficient signature generation by smart cards. *J. Cryptology*, 4(3):161–174, 1991.

[30] N. Tran, J. Li, L. Subramanian, and S. S. M. Chow. Optimal sybil-resilient node admission control. In *INFOCOM*, 2011.

[31] B. Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT*, pages 114–127, 2005.

[32] G. Wondracek, T. Holz, E. Kirda, and C. Kruegel. A practical attack to de-anonymize social network users. In *IEEE Symposium on Security and Privacy*, pages 223–238, 2010.

[33] Y. Xie, F. Yu, and M. Abadi. De-anonymizing the internet using unreliable ids. In *SIGCOMM*, pages 75–86, 2009.

[34] Q. Ye, R. Steinfeld, J. Pieprzyk, and H. Wang. Efficient fuzzy matching and intersection on private datasets. In *ICISC*, pages 211–228, 2009.

Appendix

Given the public parameters $g, \in \mathbb{G}$, $G = \hat{e}(g, \hat{f})$ [1], to prove the knowledge of $\rho \in \mathbb{Z}_p$ such that $\hat{X} = \hat{f}^{\rho}$ where $\hat{X}$ is public and $\hat{f}$ is only known to the prover:

1) Pick $k \in_R \mathbb{Z}_p$ and send $a = G^k$.
2) Get back challenge $c \in \mathbb{Z}_p^*$.
3) Compute $t = k - c\rho \mod p$.
4) Accept if and only if both $a = G^t \hat{e}(g, \hat{X})^c$ hold.

[1] either it is generated honestly, or $g, f \in \mathbb{G}, \hat{g} \in \hat{\mathbb{G}}$ are all public and $\psi(g) = \hat{g}$, $\psi(f) = \hat{f}$

a) Correctness:

$$\begin{aligned} G^t\hat{e}(g,\hat{X})^c &= G^{k-c\rho}\hat{e}(g,\hat{X})^c \\ &= G^k\hat{e}(g,\hat{f}^\rho)^{-c}\hat{e}(g,\hat{X})^c \\ &= a \end{aligned}$$

b) Soundness: For two accepting proofs (a, c_1, t_1) and (a, c_2, t_2), we have

$$\begin{aligned} G^{t_1}\hat{e}(g,\hat{X})^{c_1} &= G^{t_2}\hat{e}(g,\hat{X})^{c_2} \\ G^{t_1}\hat{e}(g,\hat{X})^{c_1} &= G^{t_2}\hat{e}(g,\hat{X})^{c_2} \\ \hat{f}^{t_1}\hat{X}^{c_1} &= \hat{f}^{t_2}\hat{X}^{c_2} \\ \hat{f}^{t_1-t_2} &= \hat{X}^{c_2-c_1} \\ \rho &= \frac{t_1-t_2}{c_2-c_1} \end{aligned}$$

c) Honest-Verifier Zero-Knowledge: Pick $t \in \mathbb{Z}_p$ and $c \in \mathbb{Z}_p^*$, compute $a = G^t\hat{e}(g,\hat{X})^c$.