Find U: Privacy-Preserving Distributed Profile Matching in Secure Multi-party Computation (SMC) for Preventing Malicious Attacks

A. Mallareddy[1], J. Dharani[2], R. Yadgiri Rao[3]

1 Research Scholar (JNTUH), Department of Computer Science & Engineering,
Professor & HOD (CSE) Sri Indu Institute of Engineering & Technology,
Sheriguda (V), Ibrahimpatnam (M), RR Dist, India

2 M. Tech (CSE) , Department of Computer Science & Engineering,
Sri Indu Institute of Engineering & Technology, Sheriguda (V), Ibrahimpatnam (M), RR Dist

3 Associate Professor, Department of Computer Science & Engineering,
Sri Indu Institute of Engineering & Technology, Sheriguda (V), Ibrahimpatnam(M), RR Dist

E-mail: 1 mallareddyadudhodla@gmail.com 2 dharani.juluri@gmail.com
3. rvrrao@gmail.com

ABSTRACT

Making new connections according to personal preferences is a crucial service in mobile social networking, where the initiating user can find matching users within physical proximity of him/her. In existing systems for such services, usually all the users directly publish their complete profiles for others to search. However, in many applications, the users' personal profiles may contain sensitive information that they do not want to make public. In this paper, we propose Find U, the first privacy-preserving personal profile matching schemes for mobile social networks. In Find U, an initiating user can find from a group of users the one whose profile best matches with his/her; to limit the risk of privacy exposure, only necessary and minimal information about the private attributes of the participating users is exchanged. Matching user profiles using their physical proximity via mobile social networking is a critical thing. We propose Find U, the concept used to limit the privacy levels and also to find the best matching profiles. To realize the user privacy levels here we are using secure multiparty computation (SMC) techniques. We also propose protocols such as PSI, PCSI to prove their security proofs. We evaluate the efficiency of the protocols by adopting the total run time and energy consumption.

Index Terms-- Private profile matching, Shamir secret sharing algorithm, Secure multi-party computation, set inflation attack, Honest but curious model, Blind and permute model.

Introduction

With the proliferation of mobile devices, mobile social networks (MSNs) are becoming an inseparable part of our lives. Leveraging networked portable devices such as smart phones and PDAs as platforms, MSN not only enables people to use their existing online social networks (OSNs) at anywhere and anytime, but also introduces a myriad of mobility-oriented applications, such as location-based services and augmented reality. Among them, an important service is to make new social connections/friends within physical proximity based on the matching of personal profiles. For example, Magnet U [1] is a MSN application that matches one with nearby people for dating or friend-making based on common interests. In such an application, a user only needs to input some (query) attributes in her profile, and the system would automatically find the persons around with similar profiles. The scopes of these applications are very broad, since people can input anything as they want, such as hobbies, phone contacts and places they have been to. The latter can even be used to find “lost connections” [2] and “familiar strangers” [3]. However, such systems also raise a number of privacy concerns. Let us first examine a motivating scenario. In a hospital, patients may include their illness symptoms and medications in their personal profiles in order to find similar patients, for physical or mental support. In this scenario, an initiating user (initiator) may want to find out the patient having the maximum number of identical symptoms to her, while being

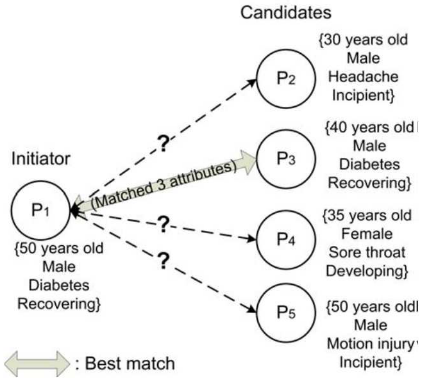

Fig. 1. Private profile matching in mobile social networks

Reluctant to disclose her sensitive illness information to the rest of the users, and the same for the users being matched with. If users’ private profiles are directly exchanged with each other, it will facilitate user profiling where that information can be easily collected by a nearby user, either in an active or passive way; and those user information may be exploited in unauthorized ways. For example, a salesman from a pharmacy may submit malicious matching queries to obtain statistics on patients’ medications for marketing purposes. To cope with User profiling in MSNs, it is essential to disclose minimal and necessary personal information to as few users as possible. In fact, the ideal situation is to let the initiator and its best matching user directly and privately find out and connect to each other, without knowing anything about other users’ profile attributes, while the rest of the users should also learn nothing about the two user’s matching attributes. The scenario is illustrated in Fig. 1, where the party P1 is the initiator and the others are called “candidates”. P1’s best matching user is P3, who shares the maximum number of symptoms with her. Since directly publishing all the profile attributes is undesirable, it is challenging to find out the matching users privately. One may think of simply turning off the cell phone or input very few attributes, but these would interfere with the system usability. Recently, Yang et. al. proposed E Small Talker [4], a practical system for matching people's interests before initiating a small-talk. However, E-Small Talker reveals the exact common attributes between the initiator and every other user, which could be more than necessary. Another difficulty of private matching under a MSN setting is the lack of a centralized authority. Lu et. al. [5] proposed a symptom matching scheme for mobile health social networks, assuming the existence of a semi-online central authority.

In this paper, we overcome the above challenges and make the following main contributions.

(1) We formulate the privacy preservation problem of profile matching in MSN. Three increasing levels of privacy are defined, where the information learnt by the initiator and each candidate includes: the intersection set between their profile attributes, the size of their intersection set, and the rank of their intersection set size, respectively.

(2) We propose two fully distributed privacy-preserving profile matching protocols. The basic ideas come from private set-intersection (PSI) techniques. However, solutions based on existing PSI schemes are less efficient. We leverage secure multi-party computation (SMC) based on polynomial secret sharing, and propose several key enhancements to improve the computation and communication efficiency. Also, users can choose personalized privacy levels when running the same matching instance.

(3) We provide thorough security analysis and performance evaluation for our schemes. Our schemes achieve several security properties not achieved by previous works, i.e., they are not only secure under the honest-but-curious (HBC) model but can also prevent several key malicious attacks. Meanwhile, they are shown to be more efficient under the settings of MSN.

II.RELATED WORK

Privacy preserving profile matching protocols, without relying on a client-server relationship nor any central server. We propose novel methods to reduce energy consumption and protocol run time, while achieving reasonable security levels. Specifically, we exploit the homomorphic properties of Shamir secret sharing to compute the intersection between user profiles privately, and due to the smaller computational domain of secret sharing, our protocols achieve higher performance and lower energy consumption for practical parameter settings of an MSN. Such a framework is also applicable to many scenarios beyond the motivating problems in this paper, for example, in patient matching in online healthcare social networks. **Algorithm:**

In this section, we first outline the idea of FindU, and then present two core designs for the PSI and PCSI protocols.

Finally we address practical issues including user discovery.

A. Overview

We present two protocols that aim at realizing one level of privacy requirement each. We start with the basic scheme realizing PSI under PL-1, which is based on secure polynomial evaluation using secret sharing. At a high level, for P1 and each Pi (2 ≤ i ≤ N), their inputs are shared among a subset Pi of 2t + 1 parties (the computing set) using (t, 2t + 1)-SS, based on which they cooperatively compute shares of the function Fi(xj) = Rij · fi(xj) + xj for each 1 ≤ j ≤ n,where fi(y) is the polynomial representing Pi's set, and Rij is a random number jointly generated by P1 and Pi but not known to any party. We have xj ∈ I1,i iff. Fi(xj) = xj .

The values of {Fi(xj)}1≤j≤n remain in secret-shared forms between P1 and Pi before their shares are revealed to each other. To reduce the communication complexity, we propose an enhancement that aggregates multiple multiplication and addition operations into one round during the secure polynomial evaluation computation.

For PL-2, the advanced scheme achieves efficient PCSI. The main idea is that, the parties in Pi first compute the (t, 2t+1)-shares of the function Fi(xj) = Rij · fi(xj), 1 ≤ j ≤ n securely using the basic scheme, whereas xj ∈ I1,i iff. Rij · fi(xj) = 0. In order to blind from P1 the correspondence between its inputs {xj} (j ∈ {1, · · · , n}) and the outputs Fi(xj′) (j′ ∈ {1, · · · , n}), we employ a blind-and-permute (BP) method. To reduce the number of invocations of the BP protocol, we use share conversion to convert the (t, t + 1)-shares of {Fi(xj)}1≤j≤n (held by parties in the reconstruction set P′

i) into (1, 2)-shares shared between P1 and Pi, so that only one BP invocation is needed between P1 and each Pi.B. The Basic Scheme We first give two definitions that capture the idea to involve the minimum number of parties during computation.

Definition 3 (Computing set of Pi): A set of 2t+1 parties Pi ⊂ P, who help P1 and Pi to compute the shares of Fi(xj), 1 ≤ j ≤ n. Pi includes P1 and Pi, and the rest 2t−1 parties are chosen as Pi+1, Pi+2, · · · with indices wrapping around.

Definition 4 (Reconstruction set of Pi): A set of t+1 partiesP′ i ⊂ Pi, who will contribute the shares of Fi(xj), 1 ≤ j ≤ n to P1 and Pi for reconstruction, P′ i also includes P1 and Pi, and the rest t − 1 parties are chosen in the same way as in the computing set.

As input, each party has a set of attributes: P1 has S1 = {x1, x2, ..., xn} and Pi has Si = {yi1, yi2, ..., yim}, respectively, where each element is an encoded attribute in Fp. For example, a hash algorithm can be used for encoding.

Rather than publishing the sets as they are, each Pi first generates an m-degree polynomial based on Si as follows:

fi(y) = (y − yi1) · (y − yi2) · · · (y − yim) = Σm k=0 aikyk, (1) where {aik}0≤k≤m−1 are coefficients. We require aim ≡ 1 so that Pi cannot give an all-zero polynomial. The function to becomputed is: Fi(xj) = Rij · fi(xj) + xj for each 1 ≤ j ≤ n,where Rij = rijr′ ij , rij and r′ ij are random numbers generated by P1 and Pi, respectively. In this way, if Fi(xj) ∈ Si, xj ∈ I1,i with high probability, and if Fi(xj) /∈ S1 then xj /∈ I1,i. The basic scheme consists of three phases, describes one run between two parties - P1 and Pi. The whole protocol between P1 and P2, ..., PN consists of N− 1 instances of the two-party protocol, which can be parallelized/ aggregated to save time (details are shown in [1]). In

the data share distribution phase, P1 shares the 1 to m powers of each of its set elements, while Pi shares its private inputs among Pi's computing set. In addition, P1 and Pi also share their n random numbers, respectively.

In the computation phase, the parties in Pi participate in secure computation of the shares of {Fi(xj)}1≤j≤n. In particular, to evaluate fi(xj), a straightforward way is to compute m − 1 multiplications of aikxkj
, 1 ≤ k ≤ m − 1 by
invoking the SS-multiplication protocol m−1 times. However,
this will introduce too much communication cost. Therefore, we propose to aggregate those multiplications into one round. That is, each party Pl ∈ Pi first locally compute a product-sum of shares zijl =Σm−1k=1 [aik]l[xkj]l based on m − 1 pairs of local shares {[aik]l, [xkj]l}1≤k≤m−1.

Then, after computing zijl, each party Pl ∈ Pi proceeds in the same way as in SS-Mul. Specifically, each Pl shares the value zijl to others by choosing a t-degree random polynomial hl(x), and then locally computes the same linear combination (Σ2t+1k=1 λkhk(l)) of the received secondary shares to get its own share of the product-sum - [Σm−1k=1 aikxkj]l. We denote this variant of SS-Mul as SS-Mul-

Add, whose correctness follows from the homomorphic properties of SS-Add and SSMul.

Since Fi(xj) = rijr′ij(ai0 +Σm−1k=1 aikxkj + xmj) + xj ,

Pl’s share of Fi(xj) can then be easily computed by invoking two more SS-Mul.

In the reconstruction phase, at least t + 1 shares of Fi(xj) are needed to reconstruct Fi(xj). To this end, the parties reveal their shares to P1 and Pi, who can obtain Fi(xj) by polynomial interpolation. P1 and Pi can test if Fi(xj) = xj , 1 ≤ j ≤ n and Fi(xj) = yj , 1 ≤ j ≤ m respectively, to determine their intersection set.

III. PROBLEM DEFINITION

A. System Model

Our system consists of N users (or parties) denoted as P1... PN, each possessing a portable device. We denote the initiating party (initiator) as P1. P1 launches the matching process and its goal is to find one party that best “matches” with it, from the rest of the parties P2,..., PN which are called candidates. Each party Pi’s profile consists of a set of attributes Si, which can be strings up to a certain length. P1 defines a matching query to be a subset of S1, and in the following we use S1 to denote the query unless specified. Also, we denote n = |S 1| and m = |S i|, i > 1, assuming each candidate has the same set size for simplicity.

There could be various definitions of “match”. In this paper, to keep it simple, we consider |S1 ∩ Si| > 0 as match (same with [4]). The best match, Pi□ is defined as the party having the maximum intersection set size with P1. P1 will first find out Pi□ via our protocols, and then they decide whether to connect with each other based on their actual intersection set. For the network, we assume devices communicate through wireless interfaces such as Bluetooth or WIFI. For simplicity, we assume every participating device is in the communication range of each other. In addition, we assume that a secure communication channel has been established between each pair of users, which can be done easily if each device has public/private key pair. Otherwise, we can use the group device pairing technique [6] to establish pair wise session keys. We do not assume the existence of a trusted third party during the protocol run; all parties carry out profile matching in a completely distributed way. They may cooperate with each other, i.e., when P1 runs the protocol with each Pi, a subset of the rest of parties would help them to compute their results.

B. Adversary Model

An outsider can eavesdrop the communication channel or modify, replay and inject messages; however it is not our main focus to prevent against active attacks from outsiders. From now on, we will deal with insiders who are participators of the matching protocol. An insider’s goal is to conduct user profiling, i.e., obtain as much personal profile information of other nearby users as possible. With a user’s attributes, a bad guy could correlate and identify that user via its MAC addresses or public keys. However, we cannot absolutely prevent user profiling, because at least the initiator and its best matching user will mutually learn the intersection set between them to make connections. Thus we focus on minimizing the amount of private information revealed in one protocol run. The parties could try to learn more information than allowed, by either inferring from the results but honestly following the protocol, or actively deviating from it. The former corresponds to the honest-but-curious (HBC) model, while the latter corresponds to the malicious model [7]. In this paper, the proposed protocols are proven secure under the HBC model; although not proven secure under the malicious model, we analyze a number of active attacks and show how they are secure against them. The adversary may act alone (be any single party) or several parties may collude. We assume that the size of a coalition is smaller than a threshold t, where t is a parameter. And we shall also assume N ≥ 2t + 1 for our proposed schemes.

C. Design Goals

1) Security Goals-- Our main security goal is to thwart user profiling attack. Since the users may have different privacy requirements and it takes different amount of efforts in protocol run to achieve them, we hereby define three levels of privacy where a higher level leaks less information to the adversaries. Note that, by default, all of the following include letting P1 and the best match Pi∗ learn the intersection set between them at the end of a protocol run. Definition

1 (Privacy Level 1 (PL-1))

When the protocol ends, P1 and each candidate Pi, 2 ≤ i ≤ N mutually learn the intersection set between them: I1,i = S1∩Si. An adversary A (whose behavior is defined in Sec. II-B) should learn nothing beyond what can be derived from the above outputs and its private inputs. If we assume the adversary has unbounded computing power, PL-1 actually corresponds to unconditional security for all the parties under the HBC model. Obviously, in PL-1, P1 can obtain all candidates' intersection sets just in one protocol run. Thus it reveals too much user information to the attacker if he assumes the role of P1. Therefore we define privacy level 2 in the following.

Definition 2 (Privacy Level 2 (PL-2))

When the protocol ends, P1 and each candidate Pi, 2 ≤ i ≤ N mutually learn the size of their intersection set: m1,i = |S1 ∩ Si|. In addition, the best match Pi□ is allowed to know the m1,i values with other Pis. The adversary A should learn nothing beyond what can be derived from the above outputs and its private inputs. In PL-2, except when m1,i = |S1| or |Si|, P1 and each Pi both will not learn exactly which attributes are in I1,i. The additional information for Pi□ is intended for it to learn whether itself is the best match under active attacks. In PL-2, the adversary needs to run the protocol multiple times to obtain the same amount of information with what he can obtain under PL-1 when he assumes the role of P1. However, PL-2 still allows A to guess which attributes are in the matching set with non-negligible probability, especially when the attribute sets are small.

Definition 3 (Privacy Level 3 (PL-3))

When the protocol ends, P1 and each Pi should only learn the ranks of each value m1,i, 2 ≤ i ≤ N. A should learn nothing more than what can be derived from the outputs and its private inputs. In PL-3, we can require that P1 only contacts the best match Pi□ , such that it only obtains the intersection set I1,i□ with the best match. If there is a tie, then the party with lowest ID is chosen as the best match. In this way, A will need at least N−1 protocol runs to learn all other user's exact profile attributes, and thus A's profiling capability is much limited.

2) Usability and Efficiency

For profile matching in MSN, it is desirable to involve as few human interactions as possible. In this paper, human user only needs to explicitly participate in the end of the protocol run, e.g., decide whether to connect when he/she becomes the best match. In addition, the system design should be lightweight and practical, i.e., being efficient Enough in computation and communication to be used in MSN. Finally, the users (especially the candidates) shall have the option to flexibly personalize their privacy levels.

D. Challenges

It is very challenging to achieve all the design goals simultaneously, especially if we desire high level of security But are unwilling to pay the high costs of computation and communication. Similar problems to ours can be found in the literature, namely private set intersection (PSI) and private cardinality of set intersection (PCSI) [7], and they are mostly tackled under the framework of Secure Multi-party Computation (SMC). The general SMC techniques [8] are often far from efficient. Researchers have proposed various customized solutions for those problems, but when applied to the ones defined here, they lead to high energy consumption and long protocol run time. In this paper, we explore novel methods with higher efficiency, while achieving reasonable security (resist a threshold number of colluders).

E. Relations to Existing Problems

In PL-1, each sub-protocol (between P1 and Pi) relates to the two-party PSI problem [7], [9], [10], while the PL-2 relates to two-party PCSI [7], [9], [11]. PL-3 is most related to the privacy-preserving nearest neighbor search problem [12], [13]. Unlike most existing problems in PSI and PCSI, we require the output of the sub-protocol between P1 and each Pi be secret-shared between them, so that the result can be revealed to both party at once to prevent cheating. This turns out to be an essential idea to minimize user profiling under malicious behavior. In addition, we define our security under the threshold cryptography model, which allows us to explore more

Efficient solutions. Finally, our problems are defined under the distributed setting, where there is no Client-server relationship nor any central party. Such framework is applicable to many scenarios beyond the motivating problems in this paper.

A. Overview

We present two protocols that aim at realizing one level of privacy requirement each. We start with the basic scheme realizing PL-1. We base our idea on the FNP scheme [7], but use secret sharing to compute polynomial evaluation securely. At a high level, for P1 and each Pi (2 ≤ i ≤ N), their inputs are shared among a subset Pi of 2t + 1 parties (the computing set) using (t, 2t + 1)-SS, based on which they cooperatively compute shares of the function Fi(xj) = Ri j · fi(xj) + xj for each 1 ≤ j ≤ n, where fi(y) is the polynomial representing Pi's set, and Ri j is a random number jointly generated by P1 and Pi but not known to any party. We have xj □ I1,I iff. Fi(xj) = xj. The values of {Fi(xj)}1≤j≤n remain in secret shared forms between P1 and Pi before their shares are revealed to each other, to provide verifiability. To reduce the communication complexity, we propose an enhancement to the secure polynomial evaluation computation.

For PL-2, the advanced scheme achieves efficient PCSI. The main idea is that, the parties in Pi first compute the (t,2t+1)-shares of the function Fi(xj) = Ri j · fi(xj), 1 ≤ j ≤ n securely using the basic scheme, whereas xj □ I1,i iff. Ri j · fi(xj) = 0. In order to blind from P1 the correspondence between its inputs {xj} (j □ {1, · · · , n}) and the outputs Fi(xj_) (j_ □ {1, · · · , n}), we employ a blind-and-permute (BP) method. To reduce the number of invocations of the BP protocol, we use share conversion to convert the (t, t+1)-shares of {Fi(xj)}1≤j≤n (held by parties in the reconstruction set P_ i) into (2, 2)-shares shared between P1 and Pi, so that only one BP invocation is needed between P1 and each Pi. The security of both the basic and advanced schemes is proven. Finally, we also discuss possible solutions to achieve PL-3, and leave practical solutions that achieve PL-3 as future work.

The SMC has been a problem that has attracted the attention of scholars and the industry for quite some time. Although a vast amount of work has been done upon the subject, the perpetual implementation of the endeavors has only yielded a perennial hornet's nest. Having said that, it should be acknowledged that to compute results upon data whose source is not known is not child's play; and the works undertaken until now have served a great purpose in enlightening the industry of the subtleties of this so-called SMC problem. Thus motivated with the intention of solving this SMC problem we proposed a new protocol Encrytpo_Random through which we had put forward what we perceived, to be the most appropriate and seemingly plausible solution to the SMC conundrum. The methodology followed was quite elementary yet very comprehensible. Encrytpo_Random worked on a two layer basis; it consisted of the parties (1st layer) who aspire to draw out a result collectively and being apprehensive of each-others intentions appoint an assumedly unbiased third party (2nd layer) to carry out the
Computation and announce the result.

In Extended Encrytpo_Random the domain of the 2nd layer has been extended from a single third-party to multiple third-parties, from whom a single entity is chosen at run time and given the responsibility of performing the required computation. A proposal sounds overtly hyperbolic without a thorough layout of the architecture to aptly implement it. Thus, here we also present a meticulously worked-out architecture to realize the protocols and also to showcase and answer the pertinent queries that are bound to arise in the minds of the audience. The modus-operandi of the protocol deters the bodies involved to exhibit any malicious conduct by presenting thoroughly planned impediments in the path of the transfer of data among themselves. The security of information of the parties is of utmost importance in any approach seeking to solve the SMC enigma. In our protocols we have taken adequate precautions so as to guarantee the security of data of the involved parties. Instead of sending the entire data blocks the parties break. Them into packets and randomly distribute amongst themselves, for a stipulated number of times. Provisions
Have been done so as to ensure that the parties do not get to know whose data packets they are forwarding, and in Stark contrast, the third party also doesn't have even a Lilliputian hint as to whose data packet a particular party is sending. This necessitates the need of a secure channel to transfer the data packets which have been dealt with in the deftly formed and apposite architecture. To further conceal the identity of the data packets we apply an encrypting function upon the data packets; these encrypting functions also reach to the third party through the same path and are used to decode the packets and rearrange them to form data blocks.

IV. IMPLEMENTATION

1. Security

Since the users may have different privacy requirements and it takes different amount of efforts to achieve them, we hereby (informally) define two levels of privacy where the higher level leaks less information to the adversaries.

2. Usability and Efficiency

For profile matching in MSN, it is desirable to involve as few human interactions as possible. In this paper, a human user only needs to explicitly participate in the end of the protocol run, e.g., decide whom to connect to based on the common interests. In addition, the system design should be *lightweight and practical*, i.e., being enough efficient in computation and communication to be used in MSN. Finally, different users (especially the candidates) shall have the option to flexibly *personalize their privacy levels*.

3. Shamir secret sharing scheme

Secret sharing schemes are multi-party protocols related to key establishment. The original motivation for secret sharing was the following. To safeguard cryptographic keys from loss, it is desirable to create backup copies. The greater the number of copies made, the greater the risk of security exposure; the smaller the number, the greater the risk that all are lost. Secret sharing schemes address this issue by allowing enhanced reliability without increased risk.

4. Preventing Malicious Attacks.

Our protocols in this paper are only proven secure in the HBC model; it would be interesting to make it secure under the stronger malicious model, i.e., to prevent an adversary from arbitrarily deviating from a protocol run. we showed that with an additional commitment round before final reconstruction (which adds little additional overhead), a specific type of “set inflation attack” can be easily prevented where a malicious user influences the final output in her favorable way by changing her shares after seeing others’.

V.CONCLUSION

Secure Multi-Party Computation is a well researched topic. Quite a few protocols already exist, and work is going-on on another handful. Through Extended Encrypto_Random we have endeavored to present a concept that emphasizes the need to keep the structure of the proposed solution to the problem very forthright so as to avoid ambiguities; at the same time ensuring the security of information by taking efficient and intricate measures. The data is first distributed and then sent forward; assuring that no party becomes victim to sabotage by other parties and also that, no party gets undue privilege, as the sole responsibility of the computation process is not vested upon a single entity. The encrypted nature of data further hinders any possibility of spiteful conduct. The possibility of collaborative malefic behavior by some party and the TTP has been completely curbed by concealing the identity of the TTP until runtime. Our protocol also reduces the complexities that are encountered in three and four layer protocols.

REFERENCES

[1] “Magnetu.” [Online]. Available: http://magnetu.com

[2] J. Manweiler, R. Scudellari, and L. P. Cox, “Smile: encounter-based trust for mobile social services,” in ACM CCS ’09, 2009, pp. 246–255.

[3] “Familiar strangers.” [Online]. Available: http://www.paulos.net/research/ intel/familiarstranger/index.htm

[4] Z. Yang, B. Zhang, J. Dai, A. Champion, D. Xuan, and D. Li, “E-smalltalker: A distributed mobile system for social networking in physical proximity,” inIEEE ICDCS ’10, June. 2010.

[5] R. Lu, X. Lin, X. Liang, and X. Shen, “A secure handshake scheme with symptoms-matching for mhealthcare social network,” Mobile Networks and Applications, pp. 1–12, 2010.

[6] M. Li, S. Yu, W. Lou, and K. Ren, “Group device pairing based secure sensor association and key management for body area networks,” in IEEE INFOCOM’10, Mar. 14-19 2010, pp. 1–9.

[7] M. Freedman, K. Nissim, and B. Pinkas, “Efficient private matching and set intersection,” in EUROCRYPT’04. Springer-Verlag, 2004, pp. 1–19.

[8] A. C. Yao, “Protocols for secure computations,” in SFCS ’82, 1982, pp. 160–164.

[9] Q. Ye, H. Wang, and J. Pieprzyk, “Distributed private matching and set operations,” in ISPEC’08, 2008, pp. 347–360.

[10] E. D. Cristofaro and G. Tsudik, “Practical private set intersection protocols with linear complexity,” in Financial Cryptography and Data Security ’10, 2010.

[11] L. Kissner and D. Song, “Privacy-preserving set operations,” in CRYPTO ’05,LNCS. Springer, 2005, pp. 241–257.