International Journal of Electrical Electronics Computers & Mechanical Engineering (IJEECM)

ISSN: XXXX-XXXX

www.ijeecm.org Volume 01 Issue 1 ǁ Nov. 2015 ǁ

IJEECM journal of Electronics and Communication Engineering (ijeecm-jec)

Iris and Fingerprint based security control system

Thota Ramarao, Y Veera raju
VNR college of Engineering and Technology, Nidubrolu.

Abstract—**Authentication is a fundamental issue to any trustoriented computing system and also a critical part in many security protocols. In addition, uthentication also serves as the first step for many other security purposes, such as key management and secure group communication [5]. Passwords or smartcards have been the most widely used authentication methods due to easy implementation and replacement; however, memorizing a password or carrying a smartcard, or managing multiple passwords/smartcards for different systems (one for each system),is a significant overhead to users. In addition, they are artificially associated with users and cannot truly identify individuals Performing authentication is notoriously difficult. Biometrics has been widely used and adopted as a promising authentication [8] method due to its advantages over some existing methods, particularly, its resistance to losses incurred by theft of passwords and smart cards. However, biometrics introduces its own challenges, Such as being irreplaceable once compromised. Biometrics can be used to prevent unauthorized access to ATMs, cellular phones, smart cards, desktop PCs, workstations, and computer networks. It can be used during transactions conducted via telephone and internet (electronic commerce and electronic banking).**

Keywords: *ARM7 (LPC2148), RFID, Fingerprint module, pc cam, keypad, buzzer, DC motor*

1. INTRODUCTION

Biometrics refers to the automatic identification of a person based on his/her physiological or behavioural characteristics. This method of identification [1] is preferred over traditional methods involving passwords and PIN numbers for various reasons: the person to be identified is required to be physically present at the point-of identification; identification based on biometric techniques obviates the need to remember a password or carry a token. With the increased use of computers as vehicles of information technology, it is necessary to restrict access to sensitive/personal data. By replacing PINs, biometric techniques can potentially prevent unauthorized access [8] to or fraudulent use of ATMs, cellular phones, smart cards, desktop PCs, workstations, and computer networks. PINs and passwords may be forgotten, and token based methods of identification like passports and driver's licenses may be forged, stolen, or lost. Thus biometric systems of identification are enjoying a renewed interest. Various types of biometric systems [1] are being used for realtime identification; the most popular are based on fingerprint matching and Iris recognition. A biometric system is essentially a pattern recognition [4] system which makes a personal identification by determining the authenticity of a specific physiological or behavioural characteristics possessed by the user. Depending on the context, a biometric system can be either a verification (authentication) [8] system or an identification system. The current security model for verification of identity, protection of information and authentication to access data or services is based on using a token or password, tied to and thereby representing an individual to either authenticate identity or allow access to information [Annet al, 2007]. This token may be password or shared secret (something you know), an identity card (something you have) or biometric (something you are). In all this cases, the details of the token are held by a third party whose functions is to authorizes and at times allow the transaction to proceed if the details of an individual's token match those stored in a database. By using biometrics it is possible to establish an identity based on who you are, rather than by what you possess, such as an ID card, or what you remember, such as a password. In some applications, biometrics may be used to supplement ID cards and passwords thereby imparting an additional level of security. Such an arrangement is often called a dual-factor authentication scheme.

1.1 EXISTING SYSTEM

Present in existing system the person who ever wants to access his things or take his amount from Bank Lockers or ATM,s first of all he has to show his id card in front of the card accessing machine. If the card is valid then he wants to enter the password in a particular machine. If the password is correct then only the locker system will be opened otherwise it will not be opened, or he can draw the amount from particular machine. So likewise the person can access his things from bank lockers. Not only in banking systems for suppose in military areas also only authorized persons have to enter in that secure area that means the area will be restricted. The authorized persons those who wants enter in that restricted area first of all he wants to show his id card in front of the card accessing machine, if it is valid card then the person will have to enter the password, if it is valid password then only the door will be opened otherwise it will not be opened.

1.2 DISADVANTAGES OF EXISTING SYSTEM

- If the person's ID card and the password are stolen by his colleagues or family members then the things will be stolen in the existing system.
- By using authorized person's identity card some other person will enter in that particular authorized areas or in military.

2. PROPOSED SYSTEM STRUCTURE AND PROTOTYPE DESIGN

Figure 1: Block Diagram of the Project

A. Analysis

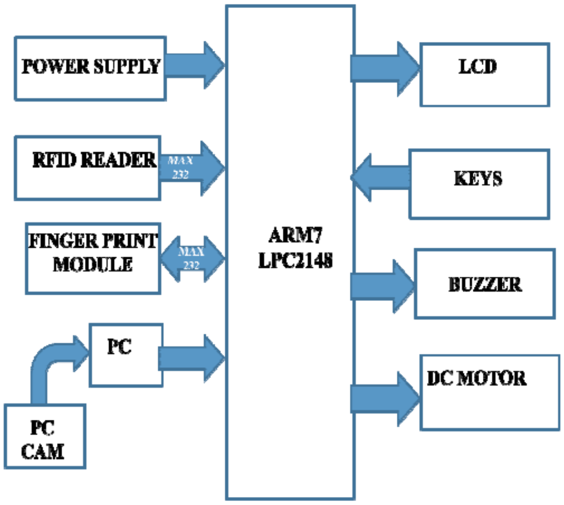


Figure 1: Block Diagram of the Project

A. Analysis of hardware Structure

1) ARM7TDMI: ARM architecture is based on reduced *Instruction Set Computer* (RISC) Principles. The RISC instruction set, and related decode mechanism are much simpler than those of Complex Instruction Set Computer (CISC) designs. This simplicity gives:

- A high instruction throughput
- An excellent real-time interrupt response
- A small, cost-effective, processor macro cell.

Microcontroller: A Micro controller consists of a powerful CPU tightly coupled with memory RAM, ROM or EPROM), various I / O features such as Serial ports, Parallel Ports, Timer/Counters, Interrupt Controller, interfaces-Analog to Digital Converter (ADC), Digital to Analog Converter (ADC), everything integrated onto a single Silicon Chip.

2) RFID reader Module: This is used to automatically identify the products tagged within the communication range of the reader, which will be able to provide the accurate consignments and real-time automatically manifest, and improve movable asset management accuracy and efficiency.

3) Finger Print Scanner: A fingerprint sensor is an electronic device used to capture a digital image of the fingerprint pattern. The captured image is called a live scan. This live scan is digitally processed to create a biometric template (a collection of extracted features) which is stored and used for matching.

4) Keys Section: With the help of these keys the users can enrol their Finger prints and they w

*5) PC Cam Section***:** This section is basically meant to capture the IRIS of the persons and to transfer this captured IRIS for Processing.

6) IRIS: In this we are using the Iris recognition technique. Iris recognition analyses the features that exist in the colored tissue surrounding the pupil, which has 250 points used for comparison, including rings, furrows, and freckles. Iris recognition uses a regular video camera system and can be done from further away than a retinal scan. It create an accurate enough measurement that can be used for Identification purposes, not just verification.

7) Buzzer: This is the output device which we are using to indicate the unauthorized person.

8) LOCKER SYSTEM: Here we are demo motor as the Locker for the authorized persons in the Locker system mode.

B. Building the Prototype System Initially the users will enrol their finge images that will be saved in the data base. block diagram of the project.

Step1: The person who ever want his amount from Bank Lockers to show his id card in front of the card accessing reader. If it is a valid one then it goes to second step. Otherwise Buzzer will be on and display like invalid person again it displays to show your RFID card.

Step2: In this step the user have to enter the correct password, if the user will entered the wrong password it will not moves the next step and the buzzer will be on, if the password accessing is continuously failed for three times means then process will move to the initial condition i.e. RFID tag showing step. If the user will then controller asks for a fingerprint access.

Step3: In this step the controller asks for a figure print access, if the finger print accessing is failed then buzzer will be on and the process will move to the first step i.e. showing step. If fingerprint access is matched with stored fingerprint or authorized person finger pr next step (Iris recognition).

Step4: In this step the person who ever want access that particular accessories or things firs in front of the PC camera at that time it will capture the image of eye and comparing with previous eye image in that same way we can access some others with matching or compare of eye image and this whole process will done wi matlab code. If it matches the locker system will be opened, user will access that Particular accessories or thing process goes on. If captured image does not matched with first taken image then the controller gives a halt to the pr demonstrating a DC fingerprints and eye Figure1 shows the he wants to access his things or take or ATM,s first of all he wants machine i.e . it gives the error message r ntered the correct password RFID tag print then it moves to first he/she has to place his eye with the help of things and the process and moves to the initial step1 at the same time buzzer will be on Figure2 shows the complete system operation flow

B. System operation flow

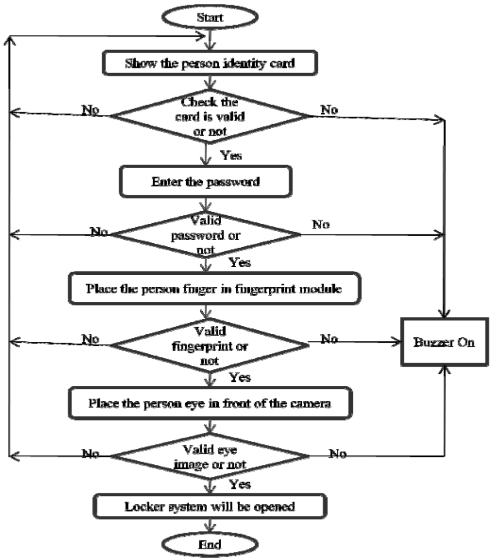

Start
Show the person identity card
Check the card is valid or not
No
No
Yes
Enter the password
Valid password or not
No
No
Yes
Place the person finger in fingerprint module
Valid fingerprint or not
No
No
Buzzer On
Yes
Place the person eye in front of the camera
Valid eye image or not
No
No
Yes
Locker system will be opened
End

Figure 2: The system operation flow

In this paper, we study user profile matching with privacy-preservation in mobile social networks (MSNs) and introduce a family of novel profile matching protocols. We first propose an explicit Comparison-based Profile Matching protocol (eCPM) which runs between two parties, an initiator and a responder. The eCPM enables the initiator to obtain the comparison-based matching result about a specified attribute in their profiles, while preventing their attribute values from disclosure.We then propose an implicit Comparison-based Profile Matching protocol (iCPM) which allows the initiator to directly obtain some messages instead of the comparison result from the responder. The messages unrelated to user profile can be divided into multiple categories by the responder. The initiator implicitly chooses the interested category which is unknown to the responder. Two messages in each category are prepared by the responder, and only one message can be obtained by the initiator according to the comparison result on a single attribute. We further generalize the iCPM to an implicit Predicate-based Profile Matching protocol (iPPM) which allows complex comparison criteria spanning multiple attributes. The anonymity analysis shows all these protocols achieve the confidentiality of user profiles. In addition, the eCPM reveals the comparison result to the initiator and provides only conditional anonymity; the iCPM and the iPPM do not reveal the result at all and provide full anonymity. We analyze the communication overhead and the anonymity strength of the protocols. We then present an enhanced version of the eCPM, called eCPM+, by combining the eCPM with a novel prediction-based adaptive pseudonym change strategy. The performance of the

eCPM and the eCPM+ are comparatively studied through extensive trace-based simulations. Simulation results demonstrate that the eCPM+ achieves significantly higher anonymity strength with slightly larger number of pseudonyms than the eCPM.

Index Terms—Mobile social network, profile matching, privacy preservation, homomorphic encryption, oblivious transfer.

I. INTRODUCTION

Social networking makes digital communication technologies sharpening tools for extending the social circle of people. It has already become an important integral part of our daily lives, enabling us to contact our friends and families on time As reported by ComScore [1], social networking sites such as Facebook and Twitter have reached 82 percent of the world's online population, representing 1.2 billion users around the world. In the meantime, fueled by the pervasive adoption of advanced handheld devices and the ubiquitous connections of Bluetooth/WiFi/GSM/LTE networks, the use of Mobile Social Networking (MSNs) has surged. In the MSNs, users are able to not only surf the Internet but also communicate with peers in close vicinity using short-range wireless communications [2]–[6]. Due to its geographical nature, the MSNs support many promising and novel applications [7]–[12]. For example, through Bluetooth communications, PeopleNet [7] enables efficient information search among neighboring mobile phones; a message-relay approach is suggested in [8] to facilitate carpool and ride sharing in a local region. Realizing the potential benefits brought by the MSNs, recent research efforts have been put on how to improve the effectiveness and efficiency of the communications among the MSN users [9], [11], [12]. They developed specialized data routing and forwarding protocols associated with the social features exhibited from the behavior of users, such as, social friendship [9], social selfishness [11], and social morality [12]. It is encouraging that the traditional solutions can be further extended to solve the MSN problems by considering the unique social features.

Privacy preservation is a significant research issue in social networking. Since more personalized information is shared with the public, violating the privacy of a target user become much easier [13]–[17]. Research efforts [13], [14], [17] have been put on identity presentation and privacy concerns in social networking sites. Gross and Acquisti [13] argued that users are putting themselves at risk both offline (e.g., stalking) and online (e.g., identity theft) based on a behavior analysis of more than 4,000 students who have joined a popular social networking site. Stutzman [14] presented a quantitative analysis of identity information disclosure in social network communities and subjective opinions from students regarding identity protection and information disclosure. When the social networking platforms are extended into the mobile environment users require more extensive privacy-preservation because they are unfamiliar with the neighbors in close vicinity who may eavesdrop, store, and correlate their personal information at different time periods and locations. Once the personal information is correlated to the location information, the behavior of users will be completely disclosed to the public. Chen and Rahman [17] surveyed various mobile Social Networking Applications (SNAs), such as, neighborhood exploring applications, mobile-specific SNAs, and content-sharing applications, all of which provide no feedback or control mechanisms to users and may cause inappropriate location and identity

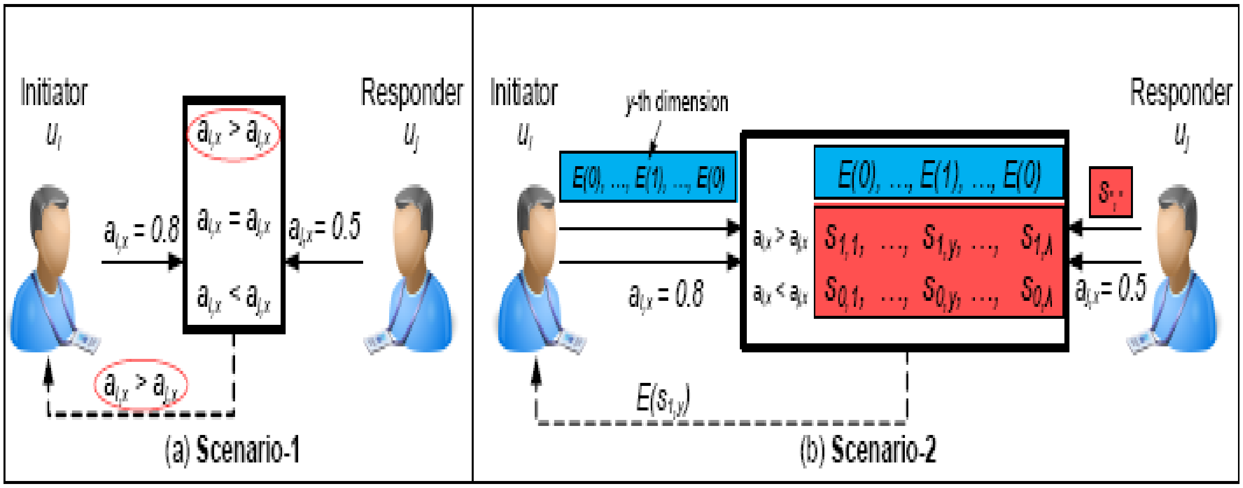


Fig. 1: Two considered scenarios: (a) Attribute value *ai;x* and attribute value *aj;x* will not be disclosed to *uj* and *ui*, respectively. The initiator obtains the comparison result at the end of the protocol. (b) *ai;x* and *aj;x* will not be disclosed to *uj* and *ui*, respectively. In addition, category *Ty* will not be disclosed to *uj* , and the comparison result will not be disclosed to any of *ui* and *uj* . The initiator obtains either *s*1*;y* or *s*0*;y* depending on the comparison result between *ai;x* and *aj;x*.

information disclosure. To overcome the privacy violation in MSNs, many privacy enhancing techniques have been adopted into the MSN applications [4], [12], [17]–[23]. For example, when two users encounter in the MSNs, privacy-preservin profile matching acts as a critical initial step to help users, especially strangers, initialize conversation with each other in a distributed and privacy-preserving manner. Many research efforts on the privacy preserving profile matching [20]–[23] have been carried out. The common goal of these works is to enable the handshake between two encountered users if both users satisfy each other's requirement while eliminating the unnecessary information disclosure if they are not. The original idea is from [18], where an agent of the Central Intelligence Agency (CIA) wants to authenticate herself to a server, but does not want to reveal her CIA credentials unless the server is a genuine CIA outlet. In the meantime, the server does not want to reveal its CIA credentials to anyone but CIA agents.

In the MSNs, we consider a generalized function to support information exchange by using profile matching as a metric. Following the previous example, we consider two CIA agents with two different priority levels in the CIA system, *A* with a low priority *lA* and *B* with a high priority *lB*. They know each other as a CIA agent. However, they do not want to reveal their priority levels to each other. *B* wants to share some messages to *A*. The messages are not related to user profile, and they are divided into multiple categories, e.g., the messages related to different regions (New York or Beijing) in different years (2011 or 2012). *B* shares one message of a specified category *T* at a time. The category *T* is chosen by *A*, but the choice is unknown to *B*. For each category, *B* prepares two self-defined messages, e.g., a low-confidential message for the CIA agent at a lower level and a high-confidential message for the agent at a higher level. Because *lA* < *lB*, *A* eventually obtains the low-confidential message without knowing that it is a lowconfidential one. In the meantime, *B* does not know which message A receives. The above function offers both A and B the highest anonymity since neither the comparison result between lA and lB is disclosed to A or B nor the category T of A's interest is disclosed to B. In the following, we refer to A as the initiator ui, B as the responder uj , the attribute used in the comparison (i.e., priority level) as ax, and the category T of A's interest as Ty. The attribute values of ui and uj on the attribute ax are denoted by $ai;x$ and $aj;x$, respectively. We first formally describe two scenarios from the above examples.

Scenario-1: The initiator wants to know the comparison result, i.e., whether it has a value larger, equal, or smaller than the responder on a specified attribute. For example, as shown in Fig. 1 (a), the initiator ui expects to know if $ai;x > aj;x$, $ai;x = aj;x$, or $ai;x < aj;x$.

Scenario-2: The initiator expects that the responder shares one message related to the category of its interest, which is however kept unknown to the responder. In the meantime, the responder wants to share with the initiator one message which is determined by the comparison result of their attribute values. For example, as shown in Fig. 1 (b), both ui and uj know that ax is used in the comparison and the categories of messages are $T1, \cdots , T\_$. The initiator ui first generates a (0, 1)-vector where the y-th dimension value is 1 and other dimension values are 0. Then, ui encrypts the vector with its own public key and sends the ciphertexts $(E(0), \cdots ,E(1), \cdots ,E(0))$ to the responder uj . The ciphertexts imply ui's interested category Ty, but uj is unable to know Ty since $E(0)$ and $E(1)$ are non-distinguishable without a decryption key. Ui also provides its attribute value $ai;x$ in an encrypted form so that uj is unable to obtain $ai;x$. On the other hand, uj prepares λ pairs of messages, each pair $(s1;h, s0;h)$ relating to one category $Th(1 \le h \le \lambda)$. Uj executes a calculation over the ciphertexts and sends the result to ui. Finally, ui obtains $E(s1;y)$ if $ai;x > aj;x$ or $E(s0;y)$ if $ai;x < aj;x$, and obtains $s1;y$ or $s0;y$ by the decryption.

A. Problem Statement

In the literature, there are many privacy-preserving profile matching protocols [10], [20]–[23]. They aim to determine the overall similarity of two profiles rather than their relation in specific attributes. They commonly check whether the proximity measure of the two profiles is larger, equal, or smaller than a pre-defined threshold value. The proximity measurement can be the size of the intersection of two sets or the distance of two vectors where sets and vectors are used to represent profiles. They do not consider the larger, equal, or smaller relations of the attribute values as the matching metrics. Moreover, the profile matching results are revealed to the participating users in certain conditions, and behavior linkage happens when the matching results are distinctive.

Consider users adopt the multiple-pseudonym technique [24], [25], i.e., users achieve high anonymity by frequently changing the unlinkable pseudonyms in the communication. As shown in Fig. 2, users uk and uj both change their pseudonyms at time t and $t'(> t)$. Since the matching result between uk and ui is non-unique value 0.7, ui is unable to link uk's behavior. However, ui is likely to know that user uj stays in its neighborhood because the matching result remains to be 0.1 which is much distinctive from other matching results. In addition, if 0.1 is unique among all possible matching results of users, they would easily recognize each other by executing the matching protocols though their profiles are not disclosed. Hence, the privacy protection of users is related to both their profiles and their profile matching results. Considering a user has v possible instances of the profile, we classify the anonymity of profile matching into three classes, nonanonymity, conditional anonymity, and full anonymity, based on the following definition.

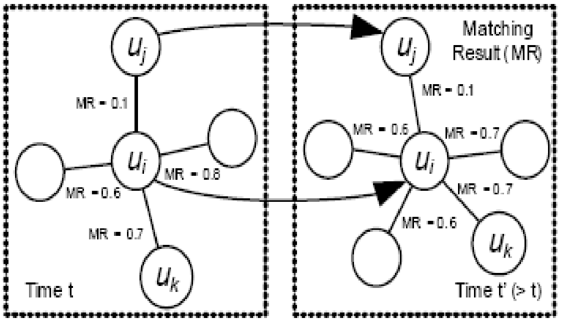


Fig. 2: Behavior linkage

Definition 1 (Non-Anonymity). A profile matching protocol provides non-anonymity if after executing multiple runs of the protocol with any user, the probability of correctly guessing the profile of the user is equal to 1.

Definition 2 (Conditional Anonymity). A profile matching protocol achieves conditional anonymity if after executing multiple runs of the protocol with some user, the probability of correctly guessing the profile of the user is larger than 1 _ .

Definition 3 (Full Anonymity). A profile matching protocol achieves full anonymity if after executing multiple runs of the protocol with any user, the probability of correctly guessing the profile of the user is always 1_ .

The profile matching protocols [21]–[23] allow the users to obtain the profile matching results which contain partial profile information. Further, the profile matching results may cause behavior linkage in certain conditions such that the revealed profile information will be correlated to break user anonymity. By cross-checking the profile matching results with the profile set, some possible instances may be excluded, and then the probability of correctly guessing the profile of the target user must be larger than 1 _ . Thus, the previous works [21] [23] only provide conditional anonymity. In this paper, we aim to design the profile matching protocols with conditional anonymity and full anonymity. We propose an explicit Comparison based Profile Matching protocol (eCPM) with conditional anonymity, an implicit Comparison-based Profile Matching protocol (iCPM) and an implicit Predicate-based Profile Matching protocol (iPPM) both with full anonymity.

TABLE I: Frequently used notations

$u_1, \cdots, u_N$	N users
u_i and u_j	The initiator and the responder
$a_1, \cdots, a_w$	The attributes in user profiles
a_x	The attribute used in comparison
$a_{i,x}$ and $a_{j,x}$	Two attribute values of u_i and u_j
$[1, l]$	The range of attribute values
$T_1, \cdots, T_\lambda$	The categories of messages
T_y	The category of the initiator u_i's interest
pid_i and pid_j	The pseudonyms of u_i and u_j
pk_i and pk_j	The public keys of u_i and u_j
$cert_{pid_i}$	The certificate on (pid_i, pk_i)
$s_{1,y}$ and $s_{0,y}$	Two messages in the category T_y
Π	The predicate set by the responder u_j
A	The attribute set of Π, $\lvert A \rvert = n$
$\bar{t}$	The threshold value of Π

B. Network Model

We consider a homogenous MSN composed of *N* mobile users. Users have equal wireless communication range, and the communication is bi-directional. The multi-pseudonym technique [24], [25] is adopted to preserve user identity and location privacy, i.e., users use pseudonyms rather than their real identities during communication and change their pseudonyms periodically. A Trusted Central Authority (TCA) is used for bootstrapping but not involved in user communication. During the bootstrapping, the TCA generates profiles, pseudonyms and associated certificates for individual users.

Similar to previous works [22], [23], each user is assumed to have *w* attributes, and its profile is a *w*-dimension vector spanning all these attributes. An integer value between 1 and l is assigned to each dimension, representing the priority level, knowledge level, or capability of users on the corresponding attribute. Therefore, a user *ui* will have a profile *pi* = (*ai;*1, · · · , *ai;w*) where $ai;h \in \mathbb{Z}$, $1 \leq ai;h \leq l$ and $1 \leq h \leq w$. For the messages to be shared, they are divided into λ categories *Th* for $1 \leq h \leq \lambda$. A non-exhaustive list of notations to be used throughout the rest of the paper can be found in Table 1.

Malicious users exist in the network. They are curious about the personal information of others, such as unique identities, location and profiles. Personal information revealed in profile matching imposes direct privacy threats to users. Fractions of such information may be aggregated by colluding users, and the behavior of the target user may be linked [15], [16]. To prevent privacy violation completely, personal information, and even profile matching results must not be disclosed. Protocol-dependent techniques are needed for preventing behavior linkage.

C. Our Contributions

In this paper, we propose a family of novel protocols eCPM, iCPM, and iPPM to solve the considered profile matching problems based on the above network model. These protocols rely on the homomorphic encryption to protect the content of user profiles from disclosure. They provide increasing levels of anonymity (from conditional to full). Our contributions are summarized below.

Firstly, we propose the eCPM for Scenario-1. For a specified attribute, the eCPM allows the initiator to know the comparison result, i.e., whether it has a larger, equal, or smaller value than the responder on the attribute. Due to the exposure of the comparison result, user profile will be leaked and linked in some conditions. We provide a numerical analysis on the conditional anonymity of the eCPM. We study the anonymity risk level in relation to the pseudonym change for the consecutive eCPM runs.

Secondly, we propose the iCPM for Scenario-2. In this protocol, the responder prepares multiple categories of messages where two messages are generated for each category. The initiator can obtain only one message related to one category for each run. During the protocol, the responder is unable to know the category of the initiator's interest. To receive which message in the category is dependent on the comparison result on a specified attribute. The responder does not know which message the initiator receives, while the initiator cannot derive the comparison result from the\ received message. We provide an analysis of the effectiveness of the iCPM, and show that the iCPM achieves full anonymity.

Thirdly, we extend the iCPM to obtain the iPPM, which has the same anonymity property as the iCPM. The iPPM allows the comparisons of multiple attributes for profile matching. The responder defines a predicate, similar to [26], which is a logical expression made of multiple comparisons between its own attribute values and the initiator's attribute values. The initiator receives one message from the responder corresponding to the specified category. To receive which message in the category is dependent on whether the initiator's attribute values satisfy the predicate or not. We provide an analysis of the effectiveness of the iPPM.

Fourthly, we improve the eCPM by combining it with an adaptive pseudonym change strategy and obtain a new variant eCPM+. In the eCPM+, each user measures its current neighborhood status periodically and predicts its future neighborhood status using an Autoregressive Moving Average

(ARMA) model. Based on the aggregate neighborhood status since last pseudonym change, it periodically estimates its anonymity risk level and changes its pseudonym when the level is too high. The extensive trace-based simulation shows that eCPM+ achieves significantly higher anonymity strength with slightly larger number of used pseudonyms than the eCPM.

The remainder of this paper is organized as follows. We review existing profile matching protocols in Sec. II and introduce some fundamental techniques in Sec. III. The three protocols eCPM, iCPM, and iPPM are presented respectively in Sec. IV-VI, along with the effectiveness discussion. Their communication overhead and anonymity strength are analyzed in Sec. VII. We present the eCPM+ in Sec. VIII and evaluate its performance in Sec. IX. Finally, we conclude the paper in Sec. X.

II. RELATED WORK

Mobile social networks as emerging social communication platforms [27]–[29] have attracted great attention recently, and their mobile applications have been developed and implemented pervasively. In mobile social networking applications, profile matching acts as a critical initial step to help users, especially strangers, initialize conversation with each other in a distributed manner. Yang et al. [30] introduced a distributed mobile communication system, called E-SmallTalker, which facilitates social networking in physical proximity. ESmallTalker automatically discovers and suggests common topics between users for easy conversation. Lu et al. [20] studied e-healthcare cases by proposing a symptom matching scheme for mobile health social networks. They considered that such matching scheme is valuable to the patients who have the same symptom to exchange their experiences, mutual support, and inspiration with each other.

In general, the profile matching can be categorized based on the formats of profiles and the types of matching operations. A well-known profile matching is the FNP scheme [19], where a client and a server compute their intersection set such that the client gets the result while the server learns nothing. Later, Kissner et al. [31] implemented profile matching with more operations including set intersection, union, cardinality and over-threshold operations. On the other hand, Ye et al. [32] further extended the FNP scheme to a distributed private matching scheme and Dachman-Soled et al. [33] aimed at reducing the protocol complexity. All the above solutions to the set intersection rely on homomorphic encryption operation. In the meantime, other works [34], [35] employed an oblivious pseudo random function to build their profile matching protocols, where communication and computational efficiency is improved. Li et al. [21] implemented profile matching according to three increasing privacy levels: i) revealing the common attribute set of the two users; ii) revealing the size of the common attribute set; and iii) revealing the size rank of the common attribute sets between a user and its neighbors. They considered an honest-but-curious (HBC) adversary model, which assumes that users try to learn more information than allowed by inferring from the profile matching results, but honestly following the protocol. They applied secure multiparty computation, the Shamir secret sharing scheme, and the homomorphic encryption scheme to achieve the confidentiality of user profiles.

In another category of profile matching [22], [23], [36], profiles can be represented as vectors, and matching operation can be inner product or distance. Such profile matching is a special instance of the secure two-party computation, which was initially introduced by Yao [37] and later generalized to the secure multi-party computation by Goldreich et al. [38]. Specifically, we introduce two recent works in this category. Dong et al. [23] considered user profile consisting of attribute values and measured the proximity of two user profiles using dot product *fdot*(*u, v*). An existing dot product protocol [39] is improved to enable verifiable secure computation. The improved protocol only reveals whether the dot product is above or below a given threshold. The threshold value is selected by the potential anonymity risk of their protocols; an adversary may adaptively adjust the threshold value to quickly narrow down the value range of the victim profile. Thus, it is required that the threshold value must be larger than a pre-defined lower bound (a system parameter) to guarantee user anonymity. The same problem exists in other works [21], [22]. Furthermore, Dong et al. [23] required users to make a commitment about their profiles to ensure the profile consistency, but profile forgery attack may still take place during the commitment phase. In the same category, Zhang et al. [22] set the matching operation *fdis*(*u, v*) of two *d*-dimension user profiles *u* and *v* as the calculation of the following distances: i) Manhattan distance, i.e., $fdis(u, v) = l\_(u, v) = (\Sigma d\ 1\ |vi - ui|\_)\ 1\ \_$; or ii) Max distance, i.e., $fdis(u, v) = lmax(u, v) = \max\{|v1 - u1|, \cdots, |vd - ud|\}$. The distance is compared with a predefined threshold τ to determine whether *u* and *v* match. Then, three increasing privacy levels are defined as: i) one of *u* and *v* learns *fdis*(*u, v*), and the other only learns *fdis*; ii) one o them learns *fdis*(*u, v*), and the other learns nothing; and iii) one of them learns whether $fdis(u, v) < \tau$, and the other learns nothing.

The proposed profile matching protocols are novel since the comparison of attribute values is considered as the matching operation. The intuitive idea is inspired by the famous Yao's millionaires' problem [37] and its solution [40]. Similar to other works [21]–[23], we propose three different protocols with different anonymity levels. For the eCPM with conditional anonymity, we provide detailed anonymity analysis and show the relation between pseudonym change and anonymity variation. For the iCPM and the iPPM with full anonymity, we show that the use of these protocols does not affect user anonymity level and users are able to completely preserve their privacy. In this section, we introduce homomorphic encryption and Autoregressive Moving Average (ARMA) model that will be used in our proposed profile matching protocols.

A. Homomorphic Encryption

There are several existing homomorphic encryption schemes that support different operations such as addition and multiplication on ciphertexts, e.g. [41], [42]. By using these schemes, a user is able to process the encrypted plaintext without knowing the secret keys. Due to this property, homomorphic encryption schemes are widely used in data aggregation and computation specifically for privacy-sensitive content [43]. We review the homomorphic encryption scheme [42] that serves a building block of our proposed profile matching protocols. A central authority runs a generator *G* which outputs $\langle p, q, R, Rq, Rp, \chi\rangle$ as system public parameters:

- $p < q$ are two primes s.t. $q \equiv 1 \bmod 4$ and $p \gg l$;
- Rings $R = Z[x]/\langle x2+1\rangle$, $Rq = R/qR = Zq[x]/\langle x2+1\rangle$;
- Message space $Rp = Zp[x]/\langle x2 + 1\rangle$;
- A discrete Gaussian error distribution $\chi = DZn;\_$ with standard deviation σ.

Suppose user *ui* has a public/private key pair (*pki, ski*) such that $pki = \{ai, bi\}$, with $ai = -(bis + pe)$, $bi \in Rq$ and $s, e \leftarrow \chi$, and $ski = s$. Let *bi;*1 and *bi;*2 be two messages encrypted by *ui*.

• **Encryption** $Epki$ ($bi;1$): $ci;1 = (c0, c1) = (aiut + pgt + bi;1, biut + pft)$, where ut, ft, gt are samples from χ.

• **Decryption $Dski$ ($ci;1$):** If denoting $ci;1 = (c0, \cdots, c\_1)$, $bi;1 = (\Sigma\_1\ k{=}0\ cksk) \bmod p$.

Consider the two ciphertexts $ci;1 = E(bi;1) = (c0, \cdots, c\_1)$ and $ci;2 = E(bi;2) = (c'0, \cdots, c'\_2)$.

• **Addition:** Let $\alpha = max(\alpha 1, \alpha 2)$. If $\alpha 1 < \alpha$, let $c\_1{+}1 = \cdots = c\_ = 0$; If $\alpha 2 < \alpha$, let $c'\_2{+}1 = \cdots = c'\_ = 0$. Thus, we have $E(bi;1 + bi;2) = (c0 \pm c'0, \cdots, c\_ \pm c'\_)$.

• **Multiplication:** Let v be a symbolic variable and compute $(\Sigma\_1\ k{=}0\ ckvk) \cdot (\Sigma\_2\ k{=}0\ c'kvk) = \hat{}c\_1{+}\_2v\_1{+}\_2 + \cdots + \hat{}c1v + \hat{}c0$. Thus, we have $E(bi;1 \times bi;2) = (\hat{}c0, \cdots, \hat{}c\_1{+}\_2)$.

B. Autoregressive Moving Average (ARMA) Model

Autoregressive model (AR) is a classic tool for understanding and predicting a time series data [44]. It estimates the current term zk of the series by a linear weighted sum of previous p terms (i.e., observations) in the series. The model order p is generally much smaller than the length of the series. AR is often combined with Moving-Average model (MA) to obtain complex ARMA model for generally improved accuracy. While AR depends on the previous terms of a time series data, MA describes the current value of the series using a linear weighted sum of white Gaussian noise or random shocks of its prior q values. As a straightforward combination of AR and MA, ARMA model is notated as ARMA(p, q) and written as

$$zk = c + p\Sigma\ i{=}1\ \phi izk{-}i + q\Sigma\ j{=}1\ \theta j\epsilon k{-}j + \epsilon k,$$

where c is a constant standing for the mean of the series, ϕi the autoregression coefficients, θi the moving-average coefficients, and ϵk the zero-mean white Gaussian noise error. For simplicity, the constant c is often omitted. Deriving ARMA(p, q) involves determining the coefficients ϕi for $i \in [1 \cdots p]$ and ϵ for $j \in [1..q]$ that give a good prediction. The model can be updated as new samples arrive so as to ensure accuracy, or it may be recomputed only when the prediction is too far from the true measurement. ARMA modeling has been applied to solve various problems, e.g. routing [45], and it will be used in Sec. VIII to enable pre-adaptive pseudonym change.

IV. EXPLICIT COMPARISON-BASED APPROACH

In this section, we present the explicit Comparison-based Profile Matching protocol, i.e., eCPM. This protocol allows two users to compare their attribute values on a specified attribute without disclosing the values to each other. But, the protocol reveals the comparison result to the initiator, and therefore offers conditional anonymity.

A. Bootstrapping

The protocol has a fundamental bootstrapping phase, where the TCA generates all system parameters, user pseudonyms, and keying materials. Specifically, the TCA runs G to generate $\langle p, q, R, Rq, Rp, \chi\rangle$ for initiating the homomorphic encryption (see Sec. III-A). The TCA generates a pair of public and private keys ($pkTCA$, $skTCA$) for itself. The public key $pkTCA$ is open to all users; the private key $skTCA$ is a secret which will be used to issue certificates for user pseudonyms and keying materials, as shown below. The TCA generates disjoint sets of pseudonyms ($pidi$) and disjoint sets of homomorphic public keys (pki) for users (ui). For every $pidi$ and pki of ui, the TCA generates the corresponding secret keys $pski$ and ski. In correspondence to each pseudonym $pidi$, it assigns a certificate $certpidi$ to ui, which can be used to confirm the validity of $pidi$. Generally, the TCA uses $skTCA$ to generate a signature on $pidi$ and pki. TheTCA outputs $certpidi$ as a tuple (pki, $SignskTCA$($pidi$, pki)). The homomorphic secret key ski is delivered to ui together with $pski$; pki is tied to $pidi$ and varies as the change of pseudonyms.

B. Protocol Steps

Consider user ui with a neighboring user uj . Denote by $pidi$ the current pseudonym of ui and by $pidj$ that of uj . Recall that ax is an attribute, $ai;x$ and $aj;x$ the values of ui and uj on ax, and l the largest attribute value. Suppose that ui as an initiator starts profile matching on ax with a responder uj . Let $pski$ and $pskj$ be the secret keys corresponding to $pidi$ and $pidj$, respectively. The protocol is executed as follows.

Step 1. ui calculates $di = Epki\ (ai;x)$, and sends a 5-tuple ($pidi$, $certpidi$, ax, di, $Signpski\ (ax, di)$) to uj .

Step 2. After receiving the 5-tuple, uj opens the certificate $certpidi$ and obtains the homomorphic public key pki and ma signature. It checks $certpidi$ to verify that (pki, $pidi$) are generated by the TCA, and it checks the signature to validate (ax, di). If any check is failed, uj stops; otherwise, uj proceeds as follows. It uses pki to encrypt its own attribute value $aj;x$, i.e., $dj = Epki\ (aj;x)$; it chooses a random value $\varphi \in \mathbb{Z}p$ such that $1 \leq \varphi < \lfloor p/(2l) \rfloor$ and $m|\varphi$ for any integer $m \in [1, l-1]$ (φ can be chosen dependent on uj 's anonymity requirement). By the homomorphic property, it calculates $Epki\ (ai;x - aj;x)$ and $d'j = Epki\ (\varphi(ai;x - aj;x))$; it finally sends a 5-tuple ($pidj$, $certpidj$, ax, $d'j$, $Signpskj\ (ax, d'j)$) to ui.

Step 3. After receiving the 5-tuple, ui opens the certificate $certpidj$ and checks the signature to make sure the validity of $pidj$ and (ax, $d'j$). If the check is successful, ui uses ski to decrypt $d'j$ and obtains the comparison result $c = \varphi(ai;x - aj;x)$. ui knows $ai;x > aj;x$ if $0 < c \leq p-1\ 2$, $ai;x = aj;x$ if $c = 0$, or $ai;x < aj;x$ otherwise.

C. Effectiveness Discussion

The effectiveness of the eCPM is guaranteed by the following theorems.

Theorem 1 (Correctness). *In the eCPM, the initiator ui is able to obtain the correct comparison result with the responder uj on a specified attribute ax.*

Proof: Recall $p \gg l$ and $1 \leq \varphi < \lfloor p/(2l) \rfloor$. As $1 \leq ai;x,\ aj;x \leq l$, we have $-l < ai;x - aj;x < l$. If $ai;x > aj;x$, we have $0 < \varphi(ai;x - aj;x) < \lfloor p/(2l) \rfloor \times l \leq p/2$. Because p is a prime and $\varphi(ai;x - aj;x)$ is an integer, we have $0 < \varphi(ai;x - aj;x) \leq (p - 1)/2$. In case of $ai;x < aj;x$, we may similarly derive $(p + 1)/2 \leq \varphi(ai;x - aj;x) < p$. Thus, by comparing $\varphi(ai;x-aj;x)$ with 0, $(p-1)/2$ and $(p+1)/2$, ui is able to know whether $ai;x > aj;x$, $ai;x = aj;x$, or $ai;x < aj;x$.

Theorem 2 (Anonymity). *The eCPM does not disclose the attribute values of participating users. Proof:* The initiator ui who starts the protocol for attribute ax encrypts its attribute value $ai;x$ using its homomorphic public key pki. Thus, the responder uj is unable to know any information about $ai;x$. On the other side, the responder uj does not transmit its attribute value $aj;x$, but returns $\varphi(ai;x - aj;x)$ to ui, where φ is a random factor added for anonymity. Since $m|\varphi$ for $1 \leq m \leq l - 1$, we have $m|(\varphi(ai;x - aj;x))$. Thus, $(ai;x - aj;x)$ can be many value between $-(l - 1)$ and $l - 1$ from ui's view, and the exact value of $aj;x$ is thus protected.

Theorem 3 (Non-forgeability). *The eCPM discourages profile forgery attack at the cost of involving the TCA for signature verification and data decryption. Proof:* Consider two users ui and uj running the eCPM with each other on attribute ax. Their public keys pki and pkj used for homomorphic encryption are generated by the TCA, and the TCA has full knowledge of the corresponding private keys ski and skj

. In addition, their attribute values are generated by the TCA and recorded in the TCA’s local repository, and the TCA can retrieve any attribute value of users (e.g. *ai;x* or *aj;x*) anytime when necessary. After the two users finish the protocol, *ui* will have *Signpskj* (*d'j*), and *uj* will have *Signpski* (*di*). If *ui*(*uj*) uses the forged profile in the protocol, *uj*(*ui*) can cooperate with the TCA to trace such malicious attack. Specifically, *uj*(*ui*) can send *Signpski* (*di*) (*Signpskj* (*d'j*)) to the TCA. the TCA will be able to check if the signatures are valid and the encrypted values are consistent with *ai;x* and *aj;x*. Thus, any profile forgery attack can be detected with the help from the TCA, and such attacks will be discouraged.

V. IMPLICIT COMPARISON-BASED APPROACH

`In this section, we propose the implicit Comparison-based Profile Matching (iCPM) by adopting the oblivious transfer cryptographic technique [40]. We consider users have distinct values for any given attribute. As shown in Fig. 3, the iCPM consists of three main steps. In the first step, *ui* chooses an interested category *Ty* by setting *y*-th element to 1 and other elements to 0 in a *λ* length vector *Vi*. *ui* then encrypt the vector by using the homomorphic encryption and sends the encrypted vector to *uj* . Thus, *uj* is unable to know *Ty* but still can process on the ciphertext. In the second step, *uj* computes the ciphertexts with input of self-defined messages (*s*1*;h*, *s*0*;h*) for 1 ≤ *h* ≤ *λ*, two encrypted vectors (*mi*, *di*), and its own attribute value *aj;x*. In the last step, *ui* decrypts the ciphertext and obtain *s*1*;y* if *ai;x* > *aj;x* or *s*0*;y* if *ai;x* < *aj;x*.

A. Protocol Steps

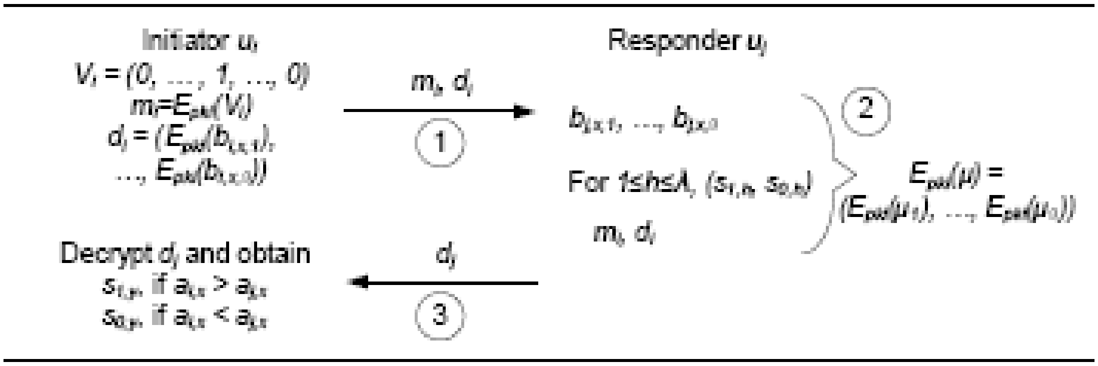

Initiator u_i
$V_i = (0, \ldots, 1, \ldots, 0)$
$m_i = E_{pk_i}(V_i)$
$d_i = (E_{pk_i}(b_{i,x,1}), \ldots, E_{pk_i}(b_{i,x,\lambda}))$
m_i, d_i
1
Responder u_j
$b_{j,x,1}, \ldots, b_{j,x,\lambda}$
For $1 \le h \le \lambda$, $(s_{1,h}, s_{0,h})$
m_i, d_i
2
$E_{pk_i}(\mu) = (E_{pk_i}(\mu_1), \ldots, E_{pk_i}(\mu_\lambda))$
Decrypt d_j and obtain
$s_{1,y}$, if $a_{i,x} > a_{j,x}$
$s_{0,y}$, if $a_{i,x} < a_{j,x}$
d_j
3

Fig. 3: The iCPM flow

In the iCPM, the responder *uj* prepares *λ* pairs of messages (*s*0*;h*, *s*1*;h*) for category *Th* (1 ≤ *h* ≤ *λ*) where *s*0*;h*, *s*1*;h* ∈ Z*p* and *s*0*;h*, *s*1*;h* ≤ (*p* − 1)⁄2. These messages are not related to *uj* ’s profile. The initiator *ui* first decides which category *Ty* it wants to receive messages related to. But *ui* does not disclose *Ty* to *uj* . Then, the responder *uj* shares either *s*0*;y* or *s*1*;y* to *ui* without knowing which one will be received by *ui*. When the protocol finishes, *ui* receives one of *s*0*;y* and *s*1*;y* with no clue about the comparison result. We elaborate the protocol steps below.

Step 1. ui generates a vector Vi = (v1, · · · , v_), where vy = 1 and vh = 0 for 1 ≤ h ≤ λ and h≠ y. This vector implies that ui is interested in the category Ty. Ui sets mi = Epki (Vi) = (Epki (v1), · · · ,Epki (v_)). It converts ai;x to binary bits ⟨bi;x;1, · · · , bi;x;_⟩, where θ = ⌈log l⌉, and sets di = (Epki (bi;x;1), · · · ,Epki (bi;x;_)). It sends a 6-tuple (pidi, certpidi , ax, di,mi, Signpski (ax, di,mi)) to uj .

Step 2. After receiving the 6-tuple, uj checks if (pidi, certpidi) are generated by the TCA and the signature is generated by ui. If both checks are successful, it knows that (ax, di,mi) is valid. uj proceeds as follows:

1) Convert aj;x to binary bits ⟨bj;x;1, · · · , bj;x;_⟩ and compute Epki (bj;x;t) for 1 ≤ t ≤ θ.

2) Compute e't = Epki (bi;x;t) − Epki (bj;x;t) = Epki (ζ't).

3) Compute e′′ t = (Epki (bi;x;t) − Epki (bj;x;t))2 = Epki (ζ′′ t).

4) Set γ0 = 0, and compute Epki (γt) as 2Epki (γt−1)+e′′ t , which implies γt = 2γt−1 + ζ′′ t .

5) Select a random rt ∈ Rp in the form of ax + b where a, b ∈ Zp, a≠ 0, and compute Epki (δt) as Epki (ζ't) + Epki (rt) × (Epki (γt) − Epki(1)), which implies δt = ζ't + rt(γt − 1).

6) Select a random rp ∈ Zp (rp≠ 0), and compute Epki (μt) as

_

Σh=1 ((s1;h + s0;h)Epki (1) + s1;hEpki (_t) □ s0;hEpki (_t)) _ (rp((Epki (vh))2 □ Epki (vh)) + Epki (vh)) + rp(_ Σh=1 Epki (vh) □ Epki (1)):

which implies μt = Σ_ h=1(s1;h(1 + δt) + s0;h(1 − δt))((v2 h − vh)rp + vh) + (Σ_ h=1 vh − 1)rp. Then, uj compiles Epki (μ) = (Epki (μ1), · · · ,Epki (μ_)), and makes a random permutation to obtain dj = P(Epki (μ)). It finally sends a 5-tuple (pidj , certpidj , ax, dj , Signpskj (ax, dj)) to ui. **Step 3**. ui checks the validity of the received 5 tuple. Then, it decrypts every ciphertext Epki (μt) in dj as follows: for Epki (μt) = (c0, · · · , c_), obtain μt by μt = (Σ_ h=0 chsh) mod p. If ai;x > aj;x, ui is able to find a plaintext μt ∈ Zp and μt = 2s1;y ≤ p − 1 and computes s1;y; if ai;x < aj;x, ui is able to find μt = 2s0;y and computes s0;y.

B.Effectiveness Discussion The correctness of the iCPM can be verified as follows. If ai;x > aj;x, then there must exist a position, say the t∗-th position, in the binary expressions of ai;x and aj;x such that bi;x;t∗ = 1, bj;x;t∗ = 0 and bi;x;t′ = bj;x;t′ for all t′ < t∗. Since γt = 2γt−1 + ζ′′ t , we have γt′ = 0, γt∗ = 1, and δt∗ = 1. For t′′ > t∗, we have γt′′ ≥ 2, and δt is a random value due to rt′′ . Since s0;y and s1;y are elements of Zp and rt is in the form of ax + b (a, b ∈ Zp, a≠ 0), ui can always determine the effective plaintext from others. The effective plaintext will be μt = Σ_ h=1(s1;h(1 + δt∗) + s0;h(1 − δt∗))((v2 h − vh)rp + vh) + (Σ_ h=1 vh − 1)rp. If the vector Vi from ui does not satisfy Σ_ h=1 vh = 1 or vh ∈ {0, 1}, ui cannot remove the random factor rp; if Vi satisfies the conditions, only s1;y and s0;y will be involved in the computation. Because δt∗ = 1, ui can obtain μt = 2s1;y ≤ p−1 and recovers s1;y. If ai;x < aj;x, we similarly have μt = 2s0;y and ui can obtain s0;y.

The confidentiality of user profiles is guaranteed by the homomorphic encryption. The comparison result δt is always in the encrypted format, and δt is not directly disclosed to ui. The revealed information is either s1;y or s0;y which is unrelated to user profiles. Therefore, the protocol transactions do not help in guessing the profiles, and the full anonymity is provided. In the meantime, vector *Vi* is always in an encrypted format so that *uj* is unable to know the interested category *Ty* of *ui*. In addition, *uj* ensures that only one of *s*1*;y* and *s*0*;y* will be revealed to *ui*. The non-forgeability property is similar to that of the eCPM. *Ui* will not lie as it makes signature *Signpski* (*ax, di*) and gives it to *uj* . The profile forgery attack will be detected if *uj* reports the signature to the TCA. Moreover, *uj* has no need to lie as it can achieve the same objective by simply modifying the contents of *s*1*;y* and *s*0*;y*.

VI. IMPLICIT PREDICATE-BASED APPROACH

Both the eCPM and the iCPM perform profile matching on a single attribute. For a matching involving multiple attributes, they have to be executed multiple times, each time on one attribute. In this section, we extend the iCPM to the multiattribute cases, without jeopardizing its anonymity property, and obtain an implicit Predicate-based Profile Matching protocol, i.e., iPPM. This protocol relies on a predicate which is a logical expression made of multiple comparisons spanning distinct attributes and thus supports sophisticated matching criteria within a single protocol run. As shown in Fig. 4, the iPPM is composed of three main steps. In the first step, different from the iCPM, *ui* sends to *uj n* encrypted vectors of its attribute values corresponding to the attributes in *A* where *A* (|*A*| = *n* ≤ *w*) is the attribute set of the predicate _. In the second step, *uj* sets 2*λ* polynomial functions *fsat;h*(*x*), *funsat;h*(*x*) for 1 ≤ *h* ≤ *λ*. *uj* then generates 2*λn* secret shares from *fsat;h*(*x*), *funsat;h*(*x*) by choosing 1 ≤ *h* ≤ *λ,* 1 ≤ *x* ≤ *n*, and arranges them in acertain structure according to the predicate _. For every 2*λ* secret shares with the same index *h*, similar to the step 2 of the iCPM, *uj* generates *θ* ciphertexts. *uj* obtains *nθ* ciphertexts at the end of the second step. In the third step, *ui* decrypts these *nθ* ciphertexts and finds *n* secret shares of *s*1*;y* and *s*0*;y*. *uj* finally can obtain *s*1*;y* or *s*0*;y* from the secret shares.

A. Protocol Steps

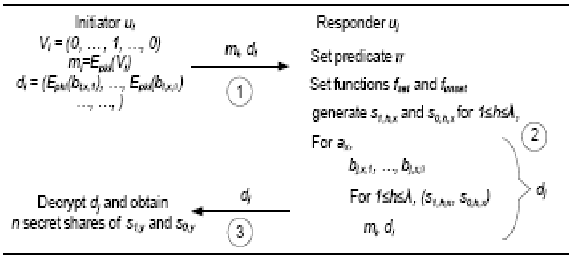

Initiator u_i
$V_i = (0, \dots, 1, \dots, 0)$
$m_i = E_{pki}(V_i)$
$d_i = (E_{pki}(b_{i,x,1}), \dots, E_{pki}(b_{i,x,l})$
$\dots, \dots,)$
m_i, d_i
1
Responder u_j
Set predicate π
Set functions f_{sat} and f_{unsat}
generate $s_{1,h,x}$ and $s_{0,h,x}$ for $1 \leq h \leq \lambda$,
For a_x,
$b_{j,x,1}, \dots, b_{j,x,l}$
For $1 \leq h \leq \lambda$, $(s_{1,h,x}, s_{0,h,x})$
m_i, d_i
2
d_j
Decrypt d_j and obtain
n secret shares of $s_{1,y}$ and $s_{0,y}$
d_j
3

Fig. 4: The iPPM flow

The iPPM is obtained by combining the iCPM with a secret sharing scheme [46] to support a predicate matching. The initiator *ui* sends its attribute values corresponding to the attributes in *A* to the responder *uj* . Without loss of generality, we assume *A* = *{a*1, · · · , *an}*. Then, *uj* defines a predicate _ = "_*t* of *{*(*ai;x, opt, aj;x*)|*ax* ∈ *A}*", where the comparison operator *opt* is either > or < and _*t* ≤ *n*. The predicate contains *n* number of requirements (i.e., comparisons), each for a distinct *ax*. The responder *uj* determines *λ* pairs of messages (*s*0*;h*, *s*1*;h*) for attributes *ah* (1 ≤ *h* ≤ *λ*). The initiator *ui* receives *s*1*;h* if at least _*t* of the *n* requirements are satisfied, or *s*0*;h* otherwise. Similar to the iCPM, *Ty* is determined by *ui* but unknown to *uj* . The threshold gate 1 ≤ _*t* ≤ *n* is chosen by *uj* . When *n* = 1, the iPPM reduces to the iCPM. The protocol steps are given below.

Step 1. *ui* generates a vector *Vi* = (*v*1, · · · , *v*_), where *vy* = 1 and *vh* = 0 for 1 ≤ *h* ≤ *λ* and *z* ≠ *y*, and sets *mi* = *Epki* (*Vi*) = (*Epki* (*v*1), · · · ,*Epki* (*v*_)). In addition, *ui* selects the attribute set *A* (|*A*| = *n*), and sends a 6-tuple (*pidi, certpidi* , *A, di,mi, Signpski* (*A, di,mi*)) to *uj* , where *di* contains *nθ* (*θ* = ⌈log *l*⌉) ciphertexts as the homomorphic encryption results of each bit of *ai;x* for *ax* ∈

A. Step 2. uj checks the validity of the received 6-tuple (similar to the

Step 2 of the iCPM). It creates a predicate _ and chooses the threshold gate _t. Using the secret sharing scheme [46], uj creates 2λ polynomials: fsat;h(v) = ρ_t−1;hv_t−1 + · · ·+ρ1;hv+s1;h and funsat;h(v) = ρ′n−_t;hvn−_t+· · ·+ρ′1;hv+ s0;h for 1 ≤ h ≤ λ, where ρ_t−1;h, · · · , ρ1;h, ρ′n−_t;h, · · · , ρ′1;h are random numbers from Z∗p. For each attribute ax ∈ A, it calculates the secret shares of s1;h;x and s0;h;x as follows (s1;h;x, s0;h;x ≤ (p − 1)/2 are required):

$$\begin{cases} s_{0,h,x} = 0||f_{unsat,h}(x), & \\ s_{1,h,x} = 1||f_{sat,h}(x), & \text{if ``} a_{i,x} > a_{j,x} \text{'' } \in \\ s_{0,h,x} = 1||f_{sat,h}(x), & \\ s_{1,h,x} = 0||f_{unsat,h}(x), & \text{if ``} a_{i,x} < a_{j,x} \text{'' } \in \end{cases}$$

Note that uj adds a prefix 0 or 1 to each secret share such that ui is able to differentiate the two sets of shared secrets, one for s1;h, the other for s0;h. uj runs the Step 2 of the iCPM n times, each time for a distinct attribute ax ∈ A and with (s1;h;x, s0;h;x) for (1 ≤ h ≤ λ) being input as s1;h and s0;h, respectively. uj then obtains dj including nθ ciphertexts. Finally, it sends a 6-tuple (pidj , certpidj , _t, A, dj , Signpskj (dj)) to ui.

Step 3. ui checks the validity of the received 6-tuple. Ui can obtain n secret shares, and each of these shares is either for s0;y or s1;y. It then classifies the n shares into two groups by looking at the starting bit (either '0' or '1'). Thus, if _ is satisfied, ui can obtain at least _t secret shares of s1;y and be able to recover s1;y; otherwise, it must obtain at least n−_t+1 secret shares of s0;y and can recover s0;y.

B.Effectiveness Discussion

The correctness of the iPPM is as follows. At Step 2, the responder uj executes the Step 2 of the iCPM n times, each time it effectively delivers only one secret share of either s0;y or s1;y to ui. When ui receives either _t shares of s1;y or n − _t + 1 shares of s0;y, it can recover either s1;y or s0;y. The interpolation function corresponding to the secret sharing scheme always guarantees the correctness. The anonymity and non-forgeability of the iPPM are achieved similar to those of the iCPM and the eCPM, respectively.

VII. PERFORMANCE ANALYSIS

In this section, we analytically study the performance of three proposed protocols eCPM, iCPM, and iPPM in terms of communication overhead and anonymity strength. When analyzing anonymity, we consider the case that users have distinct values for any given attribute. Non-distinct attribute values and comparison operations "≥" and "≤" will be considered in our future work.

A. Communication Overhead

Let |R| be the size of one ring element in Rq. In the eCPM, the initiator and the responder both need to send ciphertexts in size of 2|R|, and the communication overhead is thus subject only to the system parameter |R|. In order to achieve full anonymity, the iCPM constructs ciphertext in a sequence of operations. From Sec. III-A, we know |Enc(b)| = 2|R|. Thus, the communication overhead of the initiator is 2(θ + λ)|R| with θ = ⌈ log l⌉ . It can be seen that the initiator's communication overhead increases with system parameters (θ, λ). According to Sec. III-A an addition operation of homomorphic encryption does not increase the ciphertext size, while a multiplication with inputs of two ciphertexts of lengths a|R| and b|R| outputs a (a+b−1)|R|- length ciphertext. Thus, in the iCPM, the communication overhead of the responder increases to 6θ|R|. It is concluded that the communication overhead of the eCPM and the iCPM are constantly dependent on system parameters (θ, λ). The iPPM extends the iCPM by building complex predicates. From the protocol description, we observe that if a predicate includes $n \geq 1$ comparisons, the communication overhead of the iPPM would be approximately n times of that in the iCPM.

B. Anonymity

Suppose that user *ui* is currently using pseudonym *pidi* to execute profile matching with others.We consider an adversary aiming to break the *k-anonymity* of *ui*. *K* anonymity [47] is a classic concept for evaluating anonymity. It implies that a series of comparison results provide *k*-anonymity protection to a user if the user's behavior cannot be distinguished from at least $k - 1$ other users. We have the following definition:

Definition 4. *The k-anonymity risk level of a user is defined as the inverse of the* minimum number of distinct protocol runs (MNDPR) *that are required to break the user's k-anonymity.*

From this definition, the *k*-anonymity risk level reflects the difficulty that the adversary can break a user's *k* anonymity. In the iCPM and the iPPM, the profile matching initiator does not reveal its attribute values to the responder, and the responder has no clue about the comparison result and only reveals the self-defined messages which are not related to the profile. In this case, a user's *k* anonymity risk level can be minimum, i.e., no matter how many protocol runs are executed, its *k* anonymity risk level is always the lowest. Therefore, the iCPM and the iPPM both provide full anonymity (put users at minimum anonymity risk).

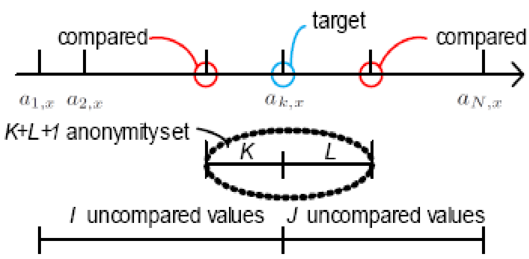
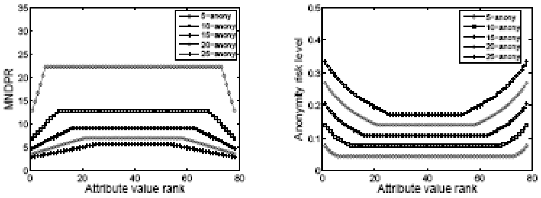


Fig. 5: Identifying the target from others

For the eCPM, it exposes the comparison results to users and thus obviously puts users at risk of the disclosure of attribute values. Because every eCPM run is executed for a particular attribute (which is specified by the initiator), any user *ui* has a *k*-anonymity risk level on its each individual attribute. When "=" case happens, users have higher anonymity level because they will be indistinguishable from other users with the same attribute values. In the following, we consider the worst case where users have distinctive attribute values on a single attribute. For a given attribute *ax*, we assume $a1;x > a2;x > \cdot \cdot \cdot > aN;x$, where *ai;x* is the value of *ui* on *ax*. In order to break *ui*'s *k*-anonymity on *ax*, the adversary has to make comparisons '$a\_;x > ai;x$' and '$ai;x > a\_;x$' for $\beta - \alpha - 1 < k$ so that the anonymity set of *ai;x* has a size smaller than *k*. Let *I* and *J* respectively be the numbers of larger and smaller values on *ax* among all the users that have not been compared to *ai;x*. Let $K \leq I$ and $L \leq J$ respectively be the number of such un-compared values in the *k*-anonymity set of *ai;x*. The relations among *I, J,K*, and *L* are shown in Fig. 5. Assuming the contact is uniformly random, we define a recursive function *f* as shown in Eqn. (1).

The above function *f*(*I, J,K,L*) returns the MNDPR with respect to a user's *k* anonymity on *ax* in the eCPM. Thus, the user's anonymity risk level in this case is defined as $L = 1/f(I, J,K,L)$. Since we assumed *a*1*;x*, · · · *aN;x* are sorted in a descending order, the index *I* actually reflects the rank of *ai;x* among the attribute values. Fig. 6 plots the MNDPR *f*(*I*, *J*,*K*,*L*) and the *k*-anonymity risk level *L* in terms of 78 users' attribute values where *k* = 5, 10, · · · , 25. It can be seen that a user with a median attribute value will have a lower *k*anonymity risk level than those with larger or smaller values. This is reasonable because the user with a median attribute value is less distinctive from other users.



(a) MNDPR per 78 users (b) Anonymity risk level

Fig. 6: Numerical results on user anonymity risk level

VIII. PERFORMANCE ENHANCEMENT

We have derived the maximum number of distinct Ecpm runs (MNDPR) before a user's *k*-anonymity is broken. This number is obtained under an assumption of uniformly random contact. However, in reality, users as social entities are likely to gather with others who have similar attribute values. This situation increases user anonymity risk level quickly when profile matching is executed frequently, and the *k*-anonymity can be broken within a much smaller number of the eCPM runs as a result. Recall that multi-pseudonym techniques are used to protect user identity and location privacy. Similar to previous work [24], [25], here we consider that pseudonyms themselves are unlinkable. In the eCPM, if a user does not change the pseudonym, the comparison result will be easily linked to break the *k*-anonymity. If a user changes pseudonym for each protocol run, information revealed by the protocol cannot be directly linked, and the user obtains highest anonymity. Nevertheless, it is desirable that the user changes pseudonym only when necessary, since pseudonyms are limited resources and have associated cost [24], [25] (e.g., communication cost for obtaining them from the TCA and computation cost for generating them on the TCA). Thus, user anonymity is tightly related with pseudonym change frequency.

$$f(I,J,K,L)=\begin{cases}0, & \text{if } K+L<k-1 \text{ or } I<K \text{ or } J<L;\\ \frac{I-K}{I+J}(f(I-1,J,K,L)+1)+\frac{J-L}{I+J}(f(I,J-1,K,L)+1)+ & \\ \frac{\sum_{z=1}^{K}(f(I-1,J,K-z,L)+1)}{I+J}+\frac{\sum_{z=1}^{L}(f(I,J-1,K,L-z)+1)}{I+J}, & \text{otherwise}\end{cases}$$

Our goal is to improve the anonymity strength of the eCPM by combining it with a pre-adaptive pseudonym change strategy which enables users to take necessary pseudonym change action before their *k*-anonymity is broken. The new verso of the eCPM is referred to as eCPM+. Before presenting the preadaptive strategy, we first propose a post-adaptive pseudonym change strategy, where users measure their anonymity risk levels periodically and change their pseudonym after their anonymity risk levels becomes larger than a pre-defined threshold value.

The post-adaptive strategy assumes that a user *uj* as responder runs the protocol on an attribute *ax* with an initiator *ui* (recognized by seeing the same pseudonym) only once, and refuses to participate any subsequent protocol running on the same *ax* with *ui*. However, if *ui* has changed its pseudonym since the last protocol running with *uj* , then *uj* will consider *ui* as a *new partner* and participate the protocol. Time is divided into slots of equal duration. The *neighborhood status* of *ui* on attribute *ax* in a time slot is characterized by a pair of values *NSi;x* = (*ni;x;s*, *ni;x;l*), respectively implying the number of new partners (identified in the time slot) with attribute values smaller than *ai;x* and the number of those with attribute values larger than *ai;x*. It varies over time due to user mobility and can be modeled as a time series data.

The centre of this strategy is the continuous measurement of user anonymity risk level based on neighborhood status. In the iCPM, attribute values are protected, and users obtain the matching results. For every attribute *ax*, user *ui* maintains the numbers *Ni;x;s* and *Ni;x;l* of discovered values that are smaller and larger than its own value *ai;x* since the last change of pseudonyms. These two numbers are respectively the sum of individual *ni;x;s* and the sum of *ni;x;l* corresponding to the past several time slots. Recall that *ai;x* is ranked the *i*-th largest among all *N* users in the network. Let *I* = *i*−1 and *J* = *N*−*i*. *ui* is not able to compute the accurate MNDPR because it does not have the information of the last two arguments of function *f*() (see Eqn. 1). The anonymity risk level of *ui* on *ax* may be estimated as *L* = 1/*f′*(*Ni;x;s*,*Ni;x;l*), where *f′*(*Ni;x;s*,*Ni;x;l*) approximates the MNDPR of *ui* regarding *ax* and is given as

$$\sum_{\substack{1\le\alpha\le I-N_{i,x,s}\\ 1\le\beta\le J-N_{i,x,l}}} \Pr[(\alpha,\beta)]\cdot f(I-N_{i,x,s}, J-N_{i,x,l}, \alpha,$$

For simplicity, we assume that the *Ni;x;s* values are randomly
distributed among the *I* − *α* users (0 ≤ *α* ≤ *I* − *Ni;x;s*) with larger values on *ax* than *ui* and the *Ni;x;l* values are randomly distributed among the *J* − *β* smaller-value users (0 ≤ *β* ≤ *J* − *Ni;x;l*). Thus, for *Ni;x;s* ≥ 1 and *Ni;x;l* ≥ 1, we have *f′*(*Ni;x;s*,*Ni;x;l*) as

$$\sum_{\substack{0\le\alpha\le I-N_{i,x,s}\\ 0\le\beta\le J-N_{i,x,l}}} \frac{\binom{I-\alpha-1}{N_{i,x,s}-1}\binom{J-\beta-1}{N_{i,x,l}-1}}{\binom{I}{N_{i,x,s}}\binom{J}{N_{i,x,l}}} f(I-N_{i,x,s}, J-$$

Ni;x;l)

For $N_{i,x,s}=0$ and $N_{i,x,l}\ge 1$, $f'(N_{i,x,s},N_{i,x,l})$ is

$$\sum_{0\le\beta\le J-N_{i,x,l}} \frac{\binom{J-\beta-1}{N_{i,x,l}-1}}{\binom{J}{N_{i,x,l}}}\cdot f(I, J-N_{i,x,l}, I, \beta).$$

For $N_{i,x,s}\ge 1$ and $N_{i,x,l}=0$, $f'(N_{i,x,s},N_{i,x,l})$ is

$$\sum_{0\le\alpha\le I-N_{i,x,s}} \frac{\binom{I-\alpha-1}{N_{i,x,s}-1}}{\binom{I}{N_{i,x,s}}}\cdot f(I-N_{i,x,s}, J, \alpha, J).$$

In the above computation, *ui* needs to know *N* and its value rank *i*. The information can be obtained from the TCA when *ui* registers to the TCA. If users are allowed to freely leave and enter the network, they will need to de-register/re-register themselves with the TCA when leaving/joining the network. In this case, (*N*, *t*) are changing, and the TCA has to be involved in the network operation in order to maintain latest network status and update users with the latest information.

The post-adaptive strategy also relies on *pseudonym lifetime* for making pseudonym change decisions. Suppose that user *ui* is currently using pseudonym *pidi*. The longer *pidi* has been used, the more private information of *ui* is leaked in case its anonymity has been broken. Hence, when *ui*'s anonymity risk level *Li* has stayed unchanged for a certain duration, called the lifetime of *pidi* and denoted by *τ* (*pidi*), *ui* changes its pseudonym for damage control. However, *τ* (*pidi*) should not be given as a constant value, but subject to *Li*. The higher *Li* is, the more possible the anonymity of *ui* is broken, and therefore the smaller *τ* (*pidi*) is. We define *τ* (*pidi*) = *ξ* MNDPR*i* *Li* , where MNDPR*i* is obtained by Eqn. 1 and *ξ* > 1 is the pseudonym lifetime factor.

For the pre-adaptive pseudonym change strategy, each user *ui* initializes an ARMA model for its neighborhood status on every attribute when entering the network. Since it has *w* attributes, the number of ARMA models to be initialized is *w*. At the end of each time slot, it measures its current neighborhood status on each attribute and updates the corresponding ARMA models. It takes the post-adaptive strategy for each attribute to determine whether to change its pseudonym. In case pseudonym change is not suggested, it proceeds to predict the neighborhood status on all the attributes in the following time slot using the ARMA models. If one of the predicted neighborhood status leads to an unacceptable anonymity risk level, it changes its pseudonym; otherwise, it does not. The pre-adaptive strategy strengths the post-adaptive strategy by one-step ahead prediction based decision making and generally enhances user anonymity.

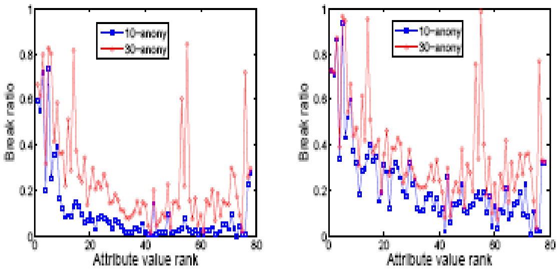
IX. PERFORMANCE EVALUATION

The eCPM+ addresses accumulative anonymity risk in multiple protocol runs and tunes itself automatically to maintain desired anonymity strength. Some previous works [21], [22] are concerned only with the anonymity risk brought by each individual protocol run, and some works [23] reduce anonymity risk by manually adjusting certain threshold values. Though they provide the conditional anonymity as the eCPM, they are not comparable to the eCPM and the eCPM+ because the anonymity protection of users is considered in terms of consecutive protocol runs. Therefore, in this section we evaluate the eCPM+ (which uses a pre-adaptive pseudonym change strategy) in comparison with two other eCPM variants, respectively employing a constant pseudonym change interval *z* (CONST-*z*) and a post-adaptive pseudonym change strategy (Post). ***A. Simulation Setup***

Our simulation study is based on the real trace [48] collected from 78 users attending a conference during a four-day period. A contact means that two users come close to each other and their attached Bluetooth devices detect each other. The users' Bluetooth devices run a discovery program every 120 seconds on average and logged about 128, 979 contacts. Each contact is characterized by two users, a start-time, and a duration. In CONST-*z*, we set the pseudonym change interval *z* from 1 to 40 (time slots); in the post-adaptive and pre adaptive strategies, we set pseudonym lifetime factor $\xi = 30$. In the pre-adaptive strategy, we use ARMA order (10, 5).

We use the contact data to generate user profiles. According to social community observations [49], users within the same social community often have common interests and are likely interconnected through strong social ties [11]. The stronger tie two users have, the more likely they contact frequently. Let *fi;j* denote the number of contacts of users *ui* and *uj* . We build a complete graph of users and weight each edge (*ui, uj*) by *fi;j* . By removing the edges with a weight smaller then 100, we obtain a graph *G* containing 78 vertices and 2863 edges. We find all maximal cliques in *G* using the Bron-Kerbosch algorithm [50]. A clique is a complete subgraph. A maximal clique is a clique that cannot be extended by including one more adjacent vertex. We obtain the 7550 maximal cliques *C*1, · · · ,*C*7550 that all contain ≥ 15 users.

Without loss of generality, we assume that these cliques are sorted in the descending order of the weight sum of their edges (the weight sum of *C*1 is the largest). We then construct communities in the following way. Scan the sequence of cliques from *C*1 to *C*7550. For a scanned clique *Ci*, find a clique *Cj* that has been previously scanned and identified as *core clique* and contains ≥ 80% vertices of *Ci*. If there are multiple such cliques, take the one with largest weight sum as *Cj* . If *Cj* is found, assign *Ci* with the same attribute as *Cj* ; otherwise, generate a new attribute, assign it to *Ci*, and mark *Ci* as a core clique. After the attribute generation and assignment, merge the cliques with the same attribute into a community. A community contains multiple users, and a user may belong to multiple communities. From the above settings, we generate 349 attributes and thus obtain 349 communities. We however concentrate on the first generated 100 attributes and their corresponding communities for simplicity. On average, each of these considered communities contains 28 users, and each user belongs to 38 communities.



(a) $z = 1$ (b) $z = 20$

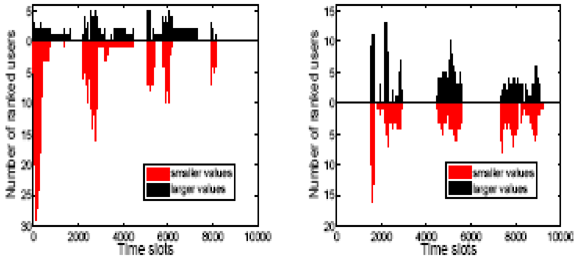
Fig. 7: Anonymity break period under the constant strategy

Afterwards, we assign values to each user in *G* for these 100 attributes. For an attribute *ax*, we find the corresponding community\ *Cx* and do the following. For each user in *Cx*, we compute the weight sum of its incidental edges in *Cx*; for each vertex outside *Cx*, we compute the weight sum of its incident edges to the vertices in *Cx*; then, we sort all the users in the decreasing order of their weight sums and assigned their values on *ax* with (78, 77, · · · , 1). This assignment method\ is reasonable because a large weight sum indicates a large interest in communicating with users in *Cx* and thus a strong background in the aspect represented by *ax*.

Our simulation spans 10, 000 time slots, each lasting 30 seconds, and focuses on a randomly selected attribute. Users can change their pseudonym at the beginning of each time slot. The pseudonym is *corrupted* in terms of *k*-anonymity (on the selected attribute) if there are less than *k*−1 other users in the network that will obtain the same matching results in the same protocol settings. A user experiences an anonymity break (on the selected attribute) if it is using a *corrupted* pseudonym.

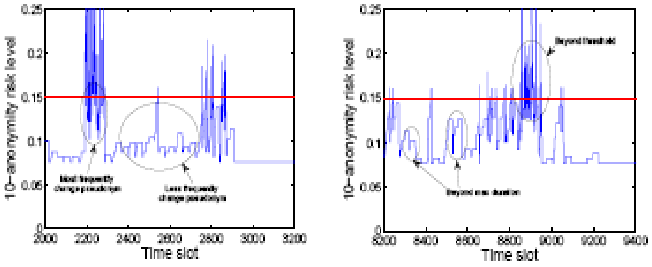
B. SimulationResults

Figure 7 sh ows the anonymity break period experienced by each user with the constant strategy being used. It can be seen that when *z* = 1, each user experiences the shortest anonymity break period at the cost of 10, 000 pseudonyms per user. Anonymity break is still possible in this extreme case because users may have multiple contacts within a single time slot while they are still using the same pseudonym. If a user has a more restrictive anonymity requirement (e.g., from 10-anonymity to 30-anonymity) or uses a larger pseudonym change interval (from 1 time slot to 20 time-slots), it will have more *corrupted* pseudonyms and thus suffer a longer period of anonymity break.



(a) The 7th user (b) The 32nd user

Fig. 8: Neighborhood status over time



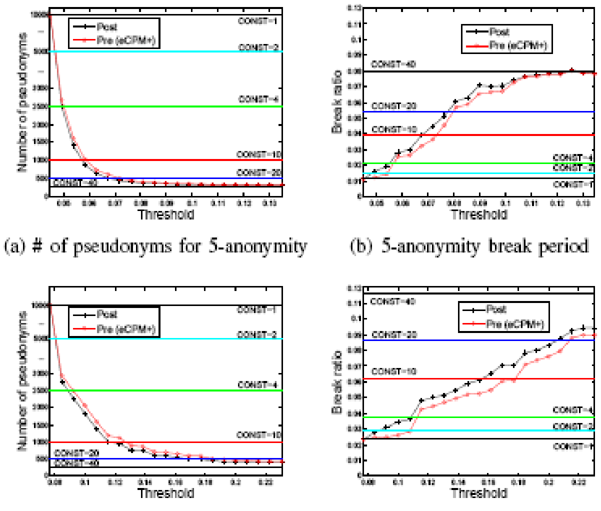
(a) Time period (2000, 3200) (b) Time period (8200, 9400)

Fig. 9: Anonymity risk level over time ($th = 0.15$)

The neighborhood status of a user on a given attribute is characterized by the number of neighbors with larger values and the number of neighbors with smaller values. We investigate the regularity of neighborhood status of individual users over time and justify the effectiveness of pre-adaptive strategy. To do so, we randomly choose two users, ranked respectively the 7th and the 32nd. Figure 8 shows their neighborhood status. The 7th user's neighborhood status exhibits regular change, i.e., the numbe of neighbors with larger values stays stable, and that of neighbors with smaller values decrease linearly over time. For the 32nd user, the number of users with larger values and the number of users with smaller values both decrease.

We choose the 32nd user, who in general has lower anonymity risk level than the 7th user, and show its 10-anonymity risk level in two consecutive time periods (2000, 3200) and (8200, 9400) with the post-adaptive strategy in Fig 9. The anonymity risk level threshold is *th* = 0.15. In the figure, the drop from a high risk level to a low risk level indicates a pseudonym change. Recall that a user changes its pseudonym not only when the anonymity risk level is beyond threshold *th* but also when its current pseudonym expires. This is reflected by the anonymity risk level drop happened below the threshold line in the figure. From Fig. 8, we can see that the pseudonym change frequency is high when the user encounters a large number of neighbors. This is reasonable as a large number of profile matching runs are executed in this case, and the user's anonymity risk level grows quickly. When the level is beyond a pre-defined threshold, the user changes its pseudonym.

Figure 10 shows the performance of the constant, the postadaptive and the pre-adaptive strategies respectively for 5-anonymity and 10-anonymity, in relation with threshold *th*. The results are obtained with respect to the 32nd user. For the constant strategy, multiple lines are plotted, respectively corresponding to *z* = *{*1, 2, 4, 10, 20, 40*}*. As *z* goes up, the user consumes a decreasingly number of pseudonyms and has an increasingly break ratio (the ratio of the number of time slots that the *k* anonymity of the 32nd user is broken to 10,000). It can be seen that the number of pseudonyms consumed by the post-adaptive and pre-adaptive strategies are much smaller than those of the constant strategy. For example, in the case of 5-anonymity and *th* = 0.0763, the post-adaptive strategy spends 369 pseudonyms and results in a 514 time slot anonymity break period.The con-stant strategy consumes 500(> 369) pseudonyms and has a 0.0540(> 0.0514) break ratio. The post-adaptive strategy outperforms the constant strategy in anonymity protection by using fewer pseudonyms to achieve smaller break ratio. Similar phenomena are observed for other *th* values and 10-anonymity scenario as well. In particular, we find that as expected, the pre-adaptive strategy leads to yet better anonymity performance than the post-adaptive one. Fig. 10 shows that in case of 5-anonymity and *th* = 0.0763, the pre-adaptive strategy consumes 449(> 369) pseudonyms and results in a 0.0445(< 0.0514) break ratio. The pre-adaptive strategy consumes slightly morepseudonyms, but achieves significantly shorter anonymity break period.



(a) # of pseudonyms for 5-anonymity (b) 5-anonymity break period

(c) # of pseudonyms for 10-anonymity (d) 10-anonymity break period

Fig. 10: Pseudonyms and break ratio (the 32nd user)

X. CONCLUSION

We have investigated a unique comparison-based profile matching problem in Mobile Social Networks (MSNs), and proposed novel protocols to solve it. The explicit Comparisonbased Profile Matching (eCPM) protocol provides conditional anonymity. It reveals the comparison result to the initiator. Consider the *k*-anonymity as a user requirement, we analyze the anonymity risk level in relation to the pseudonym change for consecutive eCPM runs. We have further introduced an enhanced version of the eCPM, i.e., eCPM+, by exploiting the prediction-based strategy and adopting the pre-adaptive pseudonym change. The effectiveness of the eCPM+ is validated through extensive simulations using real-trace data. We have also devised two protocols with full anonymity, i.e., implicit Comparison-based Profile Matching (iCPM) and implicit Predicate-based Profile Matching (iPPM). The iCPM handles profile matching based on a single comparison of an attribute while the iPPM is implemented with a logical expression made of multiple comparisons spanning multiple attributes. The iCPM and the iPPM both enable users to anonymously request for messages and respond to the requests according to the profile matching result, without disclosing any profile information.

In current version of the iCPM and the iPPM, we implement ">" and "<" operations for profile matching. One future work is to extend them to support more operations, such as "≥" and "≤". Another future work is to hide the predicate information in the iPPM. Currently, the responder needs to transmit the threshold value of the predicate to the initiator, which may reveal partial information of the responder's interest. Restricting the disclosure of such parameter will be of significance for advancing comparison-based family of profile matching protocols and warrants deep investigation.

REFERENCES

[1]"Comscore," http://www.comscoredatamine.com/.

[2] A. G. Miklas, K. K. Gollu, K. K. W. Chan, S. Saroiu, P. K. Gummadi, and E. de Lara, "Exploiting social interactions in mobile systems," in *Ubicomp*, 2007, pp. 409–428.

[3] S. Ioannidis, A. Chaintreau, and L. Massouli´e, "Optimal and scalable distribution of content updates over a mobile social network," in *Proc. IEEE INFOCOM*, 2009, pp. 1422–1430.

[4] R. Lu, X. Lin, and X. Shen, "Spring: A social-based privacy-preserving packet forwarding protocol for vehicular delay tolerant networks," in *Proc. IEEE INFOCOM*, 2010, pp. 632–640.

[5] W. He, Y. Huang, K. Nahrstedt, and B. Wu, "Message propagation in adhoc- based proximity mobile social networks," in *PERCOM workshops*, 2010, pp. 141–146.

[6] D. Niyato, P. Wang, W. Saad, and A. Hjørungnes, "Controlled coalitional games for cooperative mobile social networks," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 4, pp. 1812–1824, 2011.

[7] M. Motani, V. Srinivasan, and P. Nuggehalli, "Peoplenet: engineering a wireless virtual social network," in *MobiCom*, 2005, pp. 243–257.

[8] M. Brereton, P. Roe, M. Foth, J. M. Bunker, and L. Buys, "Designing participation in agile ridesharing with mobile social software," in *OZCHI*, 2009, pp. 257–260.

[9] E.Bulut and B.Szymanski, "Exploiting friendship relations for efficient routing in delay tolerant mobile social networks," *IEEE Transactions on Parallel and*

Distributed Systems, vol. 23, no. 12, pp. 2254–2265, 2012.

[10] Z. Yang, B. Zhang, J. Dai, A. C. Champion, D. Xuan, and D. Li,
"E-smalltalker: A distributed mobile system for social networking in physical proximity," in *ICDCS*, 2010, pp. 468–477.

[11] Q. Li, S. Zhu, and G. Cao, "Routing in socially selfish delay tolerant networks," in *Proc. IEEE INFOCOM*, 2010, pp. 857–865.

[12] X. Liang, X. Li, T. H. Luan, R. Lu, X. Lin, and X. Shen, "Moralitydriven data forwarding with privacy preservation in mobile social vnetworks," *IEEE Transactions on Vehicular Technology*, vol. 7, no. 61, pp. 3209–3222, 2012.

[13] R. Gross, A. Acquisti, and H. J. H. III, "Information revelation and vprivacy in online social networks," in *WPES*, 2005, pp. 71–80.