

**PRIVACY PRESERVING DISTRIBUTED PROFILE MATCHING IN MOBILE SOCIAL NETWORK****Rachid Chergui**

USTHB, Faculty of mathematics, P.B. 32, El Alia, 16111, Bab Ezzouar, Algeria

---

**ABSTRACT:** *In this document, a privacy-preserving distributed profile matching protocol is proposed in a particular network context called mobile social network. Such networks are often deployed in more or less hostile environments, requiring rigorous security mechanisms. In the same time, energy and computational resources are limited as these heterogeneous networks are frequently constituted by wireless components like tablets or mobile phones. This is why a new encryption algorithm having a high level of security while preserving resources is proposed in this paper. The approach is based on elliptic curve cryptography, more specifically on an almost completely homomorphic cryptosystem over a supersingular elliptic curve, leading to a secure and efficient preservation of privacy in distributed profile matching.*

**KEYWORDS:** Privacy Preserving, Profile Matching, Mobile Social Network

---

**INTRODUCTION**

Social networking websites, like Facebook [6] with its 900 million active users or Google+ [7], are of widespread use in our connected and globalized world. A major trend of these social networks is to attempt to provide instant and real-time access to for users, whatever their location and the connected device they use. This sensible demand from users has led to the development of mobile social networking (MSN) software like Foursquare [9] and Gowalla [8], in which individuals with similar interests are connected together and converse with one another through either tablets or mobile phone. In that approach, mobile apps use existing social networks to create native communities and promote discovery, leading to an improvement of web-based social networks using mobile features and accessibility. Making new connections according to personal preferences is a crucial service in MSN, 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. Authors of [10] have presented FindU, a first privacy-preserving personal profile matching scheme, designed for mobile social networks. In FindU, 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. They speak about a Blind and Permute (BP) protocol. Several increasing levels of user privacy are defined, with decreasing amounts of exchanged profile information. Authors of this document propose to use a different encryption scheme into the BP algorithm. This new scheme can provide a similar level of security while reducing drastically the computation and communication costs, which is critical in the MSN context. In BP algorithm, encryption over ciphertexts is required. The original method proposed in [10] achieves this requirement using a cryptosystem [12] that needs a lot of resources, which is quite incompatible with the constraints related to MSNs. Contrarily, the scheme proposed here is based on elliptic curve cryptography [15], which leads

to smaller keys and cryptograms, low cost computations and shorter communication messages, reducing largely by doing so the batteries consumptions. The remainder of this document is organized as follows. In Section 2, related works in the field of privacy-preserving profile matching are proposed. Then, in Section 3, we give recall the FindU protocol with related definitions. We give the protocol BP in Section 4. We construct the homomorphism encryption in Section 5 and we use it in Section 6 with performance analysis in Section 7. Section 8 conclude this work.

## Related Works

The methods used in the field of privacy-preserving distributed profile matching are usually classified into three main categories according to the cryptographic tools they use. In protocols based on *oblivious polynomial evaluation*, client and a server compute the intersection of the sets corresponding to their profiles, such that the client gets the result while server learns nothing. Homomorphic encryption that allows operations over cipher texts is used to evaluate a polynomial that represents client's input obviously. This method has been originally proposed in [3], through the FNP scheme. Other examples lying in the same category can be found, for instance, in [4] and [5]. These methods are however impracticable in MSNs because they do not achieve linear computational complexity. Protocols based on *oblivious pseudorandom functions* consist of two parties that securely compute a pseudorandom function, where one of them holds the key while the other provides the input (set elements). The objective is a secure set intersection. Suppose two parties with private sets wish to learn the intersection set without revealing anything else. Let  $P_1$  and  $P_2$  be two parties that have input  $X$  and  $Y$  respectively and  $F$  a pseudorandom function, while  $k$  is a key for  $F$  belonging to  $P_1$ .  $P_2$  compute  $\{F_k(y)\}_{y \in Y}$  and  $P_1$  compute  $\{F_k(x)\}_{x \in X}$  and send the results to  $P_2$ . Thus,  $P_2$  compare which elements appear in both sets to learn the intersection [2]. The complexity of this method is smaller than the first. The last category consists of protocols based on so-called commutative encryption. An encryption scheme  $E_k(\cdot)$  is said to have the commutative property when, for all keys  $k_1$  and  $k_2$ , we have:  $E_{k_1}(E_{k_2}(x)) = E_{k_2}(E_{k_1}(x))$ . For instance, the well known RSA encryption scheme has this commutative property. The main idea when considering privacy-preserving profile matching is thus to use the commutative encryption as a keyed one-way hash function, to generate a mapping for each element  $x$  such that no party knows the key [1]. A commonly related disadvantage of this method is that it often provide a weaker security [10]. Authors of [10] have presented a privacy-preserving profile matching called FindU. FindU is a symmetric protocol, i.e., the output is known at the same time by all parties. The characteristics of this scheme is further detailed in the next section.

## The FindU Protocol

### Problem Definition

In mobile social networks, devices are wirelessly connected (using wireless interfaces such as bluetooth or wifi), thus resources are limited and a certain level of security is required. Authors of FindU algorithm suppose that the connexion is established under public key cryptosystem, where keys are distributed over parties securely. Then, when a party launches a matching, BP algorithm assure sharing a secret securely. Let us define these stages more precisely. The system consists of  $N$  users (parties) denoted as  $P_1, \dots, P_N$ , each possessing a portable device. We denote the initiation party (*initiator*) as  $P_1$ .  $P_1$  launches the matching process and its goal is to find one party that best *matches* with it, from the rest of the parties  $P_2, \dots, P_N$  that are called *candidates*. Each party  $P_i$ 's profile consists of a set of attributes  $S_i$ , which can be strings up to

a certain length.  $P_1$  defines a matching query to be a subset of  $S_1$  (in the following we use  $S_1$  to denote the query set unless specified). Also, we denote  $n = |S_1|$  and  $m = |S_i|, i > 1$ , assuming that each candidate has the same set length for the sake of simplicity. Let us now introduce the following definitions.

**Definition 1.** The match of the set  $S_i, i \in \{2, \dots, N\}$ , is by definition the cardinality of  $S_1 \cap S_i$ .

**Definition 2.** The best match  $P_i^*$  is defined as the party having the maximum intersection set size with  $P_1$ .

$P_1$  will first find out  $P_i^*$  via the proposed protocol. Then they will decide whether to connect with other based on their actual intersection set.

### Adversary Models

If a party obtains one or more (partial or full) attribute sets without the explicit consents from these users, we said he has achieved an *user profiling*. In that context, the two following levels of security can be defined [10].

- **Honest-but-Curious (HBC) adversary.** In this model, the attacker tries to learn more information than what is allowed, by inferring from the results while honestly following the protocol.
- **Malicious adversary.** The attacker tries here to learn more information than allowed by deviating from the protocol run.

### Design Goals

Here we intend to develop the design goals of FindU scheme. One of the main goals is to defend against profiling attack defined in the previous section. We let the user choose his level of security requirement that we discuss in the next section. By definition, the party  $P_1$  search among all parties the best that match with him, and at the end, the output of the algorithm will contain the intersection set between his set query at the profile set of all other parties. By launching FindU, and adversary may obtains all those informations. Thus, we let the user choose his privacy level. The main security goal is to thwart user profiling attack. Since the users may have different privacy requirement, and as it takes different amount of effort in protocol run to achieve them, we hereby define three levels of privacy where a higher level leaks less information to the adversary. Note that, by default, all of the following include letting  $P_1$  and the best match  $P_i^*$  learn the intersection set between them at the end of a protocol run.

- **Privacy level 1 (PL-1).** When the protocol ends,  $P_1$  and each candidate  $P_i, 1 < i \leq N$ , mutually learn the intersection set between them, that is,  $I_{1,i} = S_1 \cap S_i$ . An adversary  $A$  should learn nothing beyond what can be derived from the above outputs and 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,  $P_1$  can obtain all candidates' intersection sets just in one protocol run, thus it reveals too much user information to the attacker, if he assume the role of  $P_1$ .

Therefore we define privacy level 2 in the following.

- **Privacy level 2 (PL-2).** When the protocol ends,  $P_1$  and each candidate  $P_i, 1 < i \leq N$ , mutually learn the size of their intersection set:  $m_{1,i} = |S_1 \cap S_i|$ . In addition, the best match  $P_{i^*}$  is allowed to know  $m_{1,i}$  values of other  $P_i$ s. The adversary  $A$  should learn nothing beyond what can be derived from the above outputs and its private inputs.
- **Privacy level 3 (PL-3).** At the end of the protocol,  $P_1$  and each  $P_i$  should only learn the ranks of each value  $m_{1,i}, 1 < i \leq N$ .  $A$  should learn nothing more than what can be derived from the outputs and its private inputs.

In PL-3, we can require that  $P_1$  only contacts the best match  $P_{i^*}$ , such that it only obtains the intersection set  $I_{1,i}$  with the best match. In this way,  $A$  will need at least  $N - 1$  protocol runs to know all other user's exact information, such that  $A$ 's profiling capability is much limited

Authors of FindU suggest that the protocol should be *lightweight and practical*, i.e., being enough efficient in computation and communication to be used in MSN. This is why we suggest to introduce homomorphism encryption into the FindU protocol. Readers are referred to [10] for a complete decryption of FindU. In order to achieve PL-2, authors introduce homomorphism encryption over cypher-text. For our part, to reduce largely the energy consumption, we suggest to use elliptic curve based encryption. The Blind and Permute Protocol (BP), part of the FindU system, is presented in the next section, whereas the proposed improvement is detailed in Section 5.

### Blind and Permute Protocol (BP)

The input to BP protocol is a sequence  $S = (s_1, \dots, s_n)$  of integer values that is componentwise additively split between  $A$  who has  $S^0 = (s_1^0, \dots, s_n^0)$  and  $B$  who has  $S^{00} = (s_1^{00}, \dots, s_n^{00})$ , such that  $S = S^0 + S^{00}$  [12], where  $+$  stands for the vectorial addition of integers. The output is a sequence  $S'$  obtained from  $S$  by:

1. permuting the entries of  $S$  according to a random permutation  $\pi$  that is known to neither  $A$  nor  $B$ ,
2. modifying the additive split of the entries of  $S$  so that neither  $A$  nor  $B$  can use their share of it to gain any information about  $\pi$ . We seek a protocol that does this in linear computation and communication complexity.

Observe that it suffices to give a protocol that does half of the job: It blinds and permutes for  $A$  according to a random permutation chosen by  $B$ . Then we can use such protocol a second time with the roles  $A$  and  $B$  reversed, resulting in a permutation that is the composition of two random permutations: one chosen by  $B$  and unknown to  $A$ , another chosen by  $A$  and unknown to  $B$ . The protocol where  $B$  chooses the permutation is given next.

1.  $A$  computes and sends  $E_A(s_1^0), \dots, E_A(s_n^0)$  to  $B$  (here  $E$  is the cryptosystem defined in [12] whose performance is compared to our scheme in section 7).
2.  $B$  selects  $n$  random numbers  $r_1, \dots, r_n$ , and for every  $i \in 1, \dots, n$  he computes  $E_A(-r_i)$  and multiplies it by the  $E_A(s_i^0)$  he received in the first step, thereby obtaining  $E_A(s_i^0 - r_i)$ .

3.  $B$  generates a random permutation  $\pi_B$  and applies it to the sequence of  $E_A(s_i^0 - r_i)$ 's computed in the previous step, obtaining a sequence of the form  $E_A(v_1^0), \dots, E_A(v_n^0)$  that he sends to  $A$ . He also applies  $\pi_B$  to the sequence  $s_1^{00} + r_1, \dots, s_n^{00} + r_n$ , obtaining a sequence  $v_1^{00}, \dots, v_n^{00}$ . Note that the sequence  $v_1^0 + v_1^{00}, \dots, v_n^0 + v_n^{00}$  is a permuted version of  $S$  (permuted according to  $\pi_B$ ).
4.  $A$  decrypts the  $n$  items  $E_A(v_1^0), \dots, E_A(v_n^0)$  received from  $B$ , obtaining the sequence  $v_1^0, \dots, v_n^0$ .

In the FindU algorithm (advanced version), BP permit achieving PL-2 level of security.

### Homomorphism Encryption

We use elliptic curves based cryptography to construct homomorphism encryption function.

### Operation over Elliptic Curves

#### Addition and Multiplication

Elliptic curve cryptography (ECC) is an approach to public-key cryptography based on the algebraic structure of elliptic curve over finite fields [13]. Elliptic curves used in cryptography are typically defined over two types of finite fields: prime fields  $F_p$ , where  $p$  is a large prime number, and binary extension fields  $F_{2^m}$  [14]. In our paper, we focus on elliptic curves over  $F_p$ . Let  $p > 3$ , then an elliptic curve over  $F_p$  is defined by cubic equation  $y^2 = x^3 + ax + b$  as the set

$$\Sigma = \{(x, y) \in F_p \times F_p / y^2 \equiv x^3 + ax + b \pmod{p}\}$$

where  $a, b \in F_p$  are constants such that  $4a^3 + 27b^2 \not\equiv 0 \pmod{p}$ . An elliptic curve over  $F_p$  consists of the set of all pairs of affine coordinates  $(x, y)$  for  $x, y \in F_p$  that satisfy an equation of the above form and an infinity point  $O$ . The point addition and its special case, point doubling over  $\Sigma$ , is defined as follows (the arithmetic operations are defined in  $F_p$  [16]). Let  $P = (x_1, y_1)$  and  $Q = (x_2, y_2)$

be two points of  $\Sigma$ . Then:

$$P + Q = \begin{cases} O & \text{if } x_2 = x_1 \text{ and } y_2 = -y_1, \\ & \text{otherwise.} \end{cases}$$

$$(x_3, y_3)$$

where:

- $x_3 = \lambda^2 - x_1 - x_2$ ,
- $y_3 = \lambda \times (x_1 - x_3) - y_1$ ,
- $\lambda = \frac{(y_2 - y_1) \times (x_2 - x_1)^{-1}}{1} \quad \text{if } P \neq Q,$
- $\lambda = \frac{3x_1^2 + a}{2y_1} \quad \text{if } P = Q.$

Finally, we define  $P + Q = O + P = P, \forall P \in \Sigma$ , which leads to an abelian group  $(\sigma, +)$ . The multiplication  $n \times P$  means  $P + P + \dots + P$   $n$  times, and  $-P$  is the symmetric of  $P$  for the group law  $+$  defined above, for all  $P \in \Sigma$ .

### Public/Private Keys Generation with ECC

In this section we show how we can generate the public and private keys for encryption, following the cryptosystem proposed by Boneh et al. [15]. Let  $t > 0$  be an integer called “security parameter”. To generate public and private keys, first of all, two  $t$  – bits prime numbers must be computed. Therefore, a cryptographic pseudorandom generator can be used to obtain two vectors of  $t$  bits,  $q_1$  and  $q_2$ . Then, a Miller-Rabin test can be applied for testing the primality or not of  $q_1$  and  $q_2$ . We denote by  $n$  the product of  $q_1$  and  $q_2$ ,  $n = q_1 \times q_2$ , and by  $l$  the smallest positive integer such that  $p = l \times n - 1$ .  $l$  is a prime number while  $p = 2 \pmod{3}$ . In order to find the private and public keys, we define a group  $H$ , which presents the points of the super-singular elliptic curve  $y^2 = x^3 + 1$  defined over  $F_p$ . It consists of  $p + 1 = n \times l$  points, and thus has a subgroup of order  $n$ , we call it  $G$ . In another step, we compute  $g$  and  $u$  as two generators of  $G$  and  $h = q_2 \times u$ . Then, following [16], the public key will be presented by  $(n, G, g, h)$  and the private key by  $q_1$ .

### Encryption and Decryption

After the private/public keys generation, we proceed now to the encryption and decryption phases:

- Encryption: Assuming that our message space consists of integers in the set  $0, 1, \dots, T$ , where  $T < q_2$ , and  $m$  the (integer) message to encrypt. First, a random positive integer is picked from the interval  $[0, n-1]$ . Then, the cypher-text is defined by

$$C = m \times g + r \times h \in G,$$

in which  $+$  and  $\times$  refer to the additive and multiplication laws defined previously.

- Decryption: once the message  $C$  arrived to destination, to decrypt it, we use the private key  $q_1$  and the discrete logarithm of base  $q_1 \times g$  as follows:

$$m = \log_{q_1 \times g} q_1 \times C$$

### Homomorphic Properties

As we have mentioned before, our approach ensures easy encryption/decryption without any need of extra resources. This will be proved in the next section. Moreover, our approach supports homomorphic properties, which gives us the ability to execute operations on values even though they have been encrypted. Indeed, it allows  $N$  additions and one multiplication directly on cryptograms. As the product operation will not be used in the profile matching, we will not detail it in this section. Addition over cypher-texts are done as follows: let  $m_1$  and  $m_2$  be two messages and  $C_1, C_2$  their cypher-text respectively. Then the sum of  $C_1$  and  $C_2$ , let call  $C$ , is represented by  $C = C_1 + C_2 + r \times h$  where  $r$  is an integer randomly chosen in  $[0, n-1]$  and  $h = q_2 \times u$  as presented in the previous section. This sum operation guarantees that the decryption value of  $C$  is the sum  $m_1 + m_2$ .

### The modified version of BP Protocol

We rewrite the protocol BP with our novel cryptosystem with  $E$  meaning the novel algorithm.



1.  $A$  computes and sends  $E_A(s_1^0), \dots, E_A(s_n^0)$  to  $B$ .
2.  $B$  selects  $n$  random numbers  $r_1, \dots, r_n$ , and for every  $i \in 1, \dots, n$  he computes  $E_A(-r_i)$  and add it with the  $E_A(s_i^0)$  he received in the first step, thereby obtaining  $E_A(s_i^0 - r_i)$ .
3.  $B$  generates a random permutation  $\pi_B$  and applies it to the sequence of  $E_A(s_i^0 - r_i)$ 's computed in the previous step, obtaining a sequence of the form  $E_A(v_1^0), \dots, E_A(v_n^0)$  that he sends to  $A$ . He also applies  $\pi_B$  to the sequence  $s_1^{00} + r_1, \dots, s_n^{00} + r_n$ , obtaining a sequence  $v_1^{00}, \dots, v_n^{00}$ . Note that the sequence  $v_1^0 + v_1^{00}, \dots, v_n^0 + v_n^{00}$  is a permuted version of  $S$  (permuted according to  $\pi_B$ ).
4.  $A$  decrypts the  $n$  items  $E_A(v_1^0), \dots, E_A(v_n^0)$  received from  $B$ , obtaining the sequence  $v_1^0, \dots, v_n^0$ .

### Performance Analysis

The experimental results presented in [13] compare the performance comparison between RSA and ECC. For the same level of security, say level one, a device operating over RSA need a key of 472 bits while over ECC we need only a key of 46 bits. In [12], authors give a performance analysis between a cryptosystem based on Composite Degree Residuosity Classes CDRC, which is the scheme that is proposed in the BP algorithm. First, RSA is better then CDRC in term of computational complexity. CDRC offer a security level equivalent to  $Class[n]$  while RSA is equivalent to  $RSA[n, F_4]$  and we have [12]

$$RSA[n, F_4] \Rightarrow Class[n]$$

On the other hand, for the same key size, CDRC require 5120 elementary operations for encryption while RSA need only 17 operations. All those results prove the efficiency of ECC in term of performance.

### CONCLUSION AND FUTURE WORK

An homomorphic encryption scheme that enhances the performance of the FindU algorithm has been proposed in this document. Achieving the PL-3 security level is the main open problem not yet resolved. In future work, homomorphic encryption will be investigated in order to solve this issue.

### Appreciation

The research is partially supported by ATN laboratory of USTHB University.

### REFERENCES

- [1] Agrawal, Rakesh and Evfimievski, Alexandre and Srikant, Ramakrishnan, *Information sharing across private databases*, Proceedings of the 2003 ACM SIGMOD international conference on Management of data, no. 12, pp. 86–97, 2003.
- [2] Hazay, Carmit and Lindell, Yehuda, *Efficient protocols for set intersection and pattern matching with security against malicious and covert adversaries*, Proceedings of the 5th conference on Theory of cryptography, no. 21, pp. 155–175, 2008.

- [3] Michael J. Freedman, Kobbi Nissim, Benny Pinkas, *Efficient Private Matching and Set Intersection*, EUROCRYPT, no.3027 , pp. 1–19, 2004.
- [4] ,Kissner, Lea and Song, Dawn *Privacy-Preserving Set Operations*, Advances in Cryptology – CRYPTO 2005, no. 3621, pp. 241–257, 2005.
- [5] Ye, Qingsong and Wang, Huaxiong and Pieprzyk, Josef *Distributed Private Matching and Set Operations* , Information Security Practice and Experience, no. 4991, pp. 347–360, 2008.
- [6] Zuckerberg, Mark and Saverin, Eduardo and Moskovitz, Dustin and Hughes, Chris *Facebook* , 2012.
- [7] Brin, Sergey and Page, Larry *Google+* , 2012.
- [8] Williams, Josh and Raymond, Scott *Gowalla* , 2012.
- [9] Crowley, Dennis and Selvadurai, Naveen *Foursquare* ,2012.
- [10] Ming Li *User-Centric Security and Privacy Mechanisms in Untrusted Networking and Computing Environments*, Worcester Polytechnic Institute,2011.
- [11] Qi, Yinian and Atallah, Mikhail J *Efficient Privacy-Preserving k-Nearest Neighbor Search* , Proceedings of the 2008 The 28th International Conference on Distributed Computing Systems, no.9 , pp.311–319, 2008.
- [12] Paillier, Pascal, *Public-key cryptosystems based on composite degree residuosity classes* , Proceedings of the 17th international conference on Theory and application of cryptographic techniques, no.16 , pp. 223–233, 1999.
- [13] Bahi, Jacques and Guyeux, Christophe and Makhoul, Abdallah, *Secure Data Aggregation in Wireless Sensor Networks. Homomorphism versus Watermarking Approach* ADHOCNETS 2010, 2nd Int. Conf. on Ad Hoc Networks, no.49 , pp. 344–358, 2010.
- [14] R.C.C. Cheung and N.J. Telle and W. Luk and P.Y.K. Cheung, *Secure encrypted-data aggregation for wireless sensor networks* , no.13 , pp. 1048– 1059, 2005.
- [15] Boneh, Dan and Goh, Eu-Jin and Nissim, Kobbi, *Evaluating 2-DNF Formulas on Ciphertexts* , no.13 , pp. 325–341, 2005.
- [16] D. Hankerson and A. Menezes and S. Vanstone, *Guide to Elliptic Curve Cryptography* Springer, 2004.