Research on Privacy-preserving Matching Protocol with Threshold

Akihiko GONDA (1510018)
School of Information Science,
Japan Advanced Institute of Science and Technology
February 10, 2017

Keywords :
Privacy-preserving matching, Privacy, Homomorphic encryption, Mobile social networks

Abstract :

There is great interest in online friendship with the rapid expansion of mobile social networks due to the spread of smartphones and tablets. Social networking service sites such as Facebook and Twitter remain popular over the world because these are easy to use and anyone can contact friends. Meanwhile, social discovery is one of the services that many tech media such as VentureBeat and TechCrunch are expected to become the next big trend. Social discovery is one of the matching services for discovering new relationships on the mobile social networks using social preferences and personal information. We can easily form a group over the physical distance and interact with people who has the same interests as ours or who are paying attention. It is expected to create accidental discoveries and new values through interaction with them. However, it is dangerous to seek friends either using raw privacy information or meeting directly with strangers. On the other hand, we have to use more sensitive information to improve the accuracy of matching. In order to address these concerns over, privacy-preserving matching protocol is proposed as a method of safely matching using privacy information.

The privacy-preserving matching protocol is that one user secretly calculates intersection or similarity between his privacy information and the other's one, and then judges whether she becomes a friend or not. The privacy information used in privacy-preserving matching protocol are profile and interest. In particular, the interest is profile which you seek. This protocol is attracting the most attention as a technology to use the privacy information without leaking on the mobile social networks. For the purposes of this abstract, the term "server" will be taken to mean service center, which operates privacy-preserving matching service, and the term "user" will be taken to mean people using this service. Firstly, all users generate a pair of profile and interest and save it on their device. They manage their privacy information themselves instead of storing them in the server. Secondly, they encrypt profile and upload it to the service center for privacy-preserving matching. All users manage their own profile and interest, so the service center does not leak users' privacy information even if its servers are attacked. Thirdly, the server chooses two users randomly and both secretly calculates intersection or similarity between his/her profile and her/his interest with each other. Finally, the server judge whether the friend requirements are simultaneously satisfied or not. If both requirements are satisfied, the server will send them "match", but

otherwise it will send them “mismatch”. The friend requirement is, for example an ideal cardinality of the intersection, a threshold of cardinality of the intersection, a match-up of items such as limited to men, and so on. However, existing privacy-preserving matching protocols has some problems.

In many studies, it is possible to judge a case of unrequited love as a match or to keep track of which one is unrequited, since it is a matching protocol if one condition is satisfied. Many privacy-preserving matching protocols proposed by existing studies output “match” if at least one requirement is satisfied, so it is possible to judge one-sided love as “match” or to keep track of which is unrequited love. If the attacker can keep track of which is unrequited love, there is a risk of encouraging fraudulent such as forgery of profile or interest. In case that he is unrequited love, if he forges his own profile and the protocol outputs “match”, he can chase others’ interest. In case that an opponent is unrequited love, if he forges his own interest and the protocol outputs “match”, he can chase others’ profile. There are studies that solve this problem by proposing matching protocols that the server simultaneously judge whether both requirements are satisfied or not, but this requirement is not suitable for discovering new relationships. This is because the requirement is an ideal cardinality of the intersection between profile and interest and it is too strict as a friendship. The server outputs “mismatch” even if the real cardinality of the intersection passes beyond the ideal or does not reach. There are not only practical problems as above, but also functional problems.

The matching protocols with the highest privacy level normally output only “match” or “mismatch”, so if we enhance privacy, the protocols become poor in functionality. The matching which outputs those two is unsatisfactory as a service because users will have to worry about how to build relationships if they do not understand each other’s commons. We step through the process of becoming a friend with each other, so it is essential for matching to understand each other’s commons. It is not desirable for other information to be leaked out in the case of a mismatch, but in the case of a match, it is easier for those who understand the intersection to build up a relationship smoothly without worrying.

We proposed a privacy-preserving matching protocol that the friend requirements are both thresholds of intersection. In this protocol, both users encrypted with additive homomorphic public key cryptography, secretly calculate intersection of between profile and interest with each other, and the server judges whether or not the real cardinality of the intersection is larger than the threshold simultaneously. The server compares as above and output “match” if it simultaneously satisfies the requirements, and output “mismatch” otherwise. We also propose protocols that can calculate intersection only when they match further. Our adversary model considers semi-honest users and server. The semi-honest entities faithfully execute the protocols, but are curious to learn about other users’ profiles. The protocols are designed in order that privacy information and matching result such as “match” or “mismatch” cannot be inferred during execution.

In this research, we propose not only the privacy-preserving matching protocol, but also the correctness, privacy and efficiency of the protocol. In addition, we also measure the execution time of matching on a tablet. Smartphones and tablets are still inferior to laptops in processing capability, but in recent years 2 GB RAM or 3 GB RAM has been installed, so it is likely that new technologies that use public key cryptography, which have been avoided with inactive devices so far, are widely available. We implement our matching protocol in Android Studio 2.2, and we evaluate the efficiency on a tablet. All experiments are tested on Nexus 7 with 1.5GHz CPU, Qualcomm Snapdragon S4 processors, 2GB RAM, Android 6.0.1 OS.