Montclair State University
Montclair State University Digital Commons

Theses, Dissertations and Culminating Projects

5-2019

A Privacy-Aware Framework for Friend Recommendations in Online Social Networks

Mona Fahad Alkanhal
Montclair State University

Abstract

Online social networks (OSN), such as Facebook, Twitter, and LinkedIn, have revolutionized the way how people share information and stay connected with family and friends. Along this direction, user's privacy has been a significant concern to all users in the social networks. In this thesis, we propose a privacy-aware framework that allows users to outsource their encrypted profile data to a cloud environment. In order to achieve better security and efficiency, our framework utilizes a hybrid approach that consists of Paillier's encryption scheme and AES. Furthermore, we develop a privacy-aware friend recommendation protocol that recommends new friends to social network users without compromising their data. The proposed protocol adopts a collaborative analysis between the online social network provider and a cloud to increase the security in the suggested approach. Moreover, to increase the efficiency of the proposed protocol we utilize common-neighbors metric and universal hash functions. We compared our protocol with the existing work and demonstrate that our protocol is more efficient and achieves better security. We also conducted a set of experiments to evaluate the performance of our protocol and demonstrate its practicality.

MONTCLAIR STATE UNIVERSITY

A Privacy-Aware Framework for Friend Recommendations in Online Social Networks

By

Mona Alkanhal

A Master's Thesis Submitted to the Faculty of

Montclair State University

In Partial Fulfillment of the Requirements

For the Degree of

Master of Science

May 2019

College Science and Mathematics

Department Computer Science

Thesis Committee:

[redacted]

Dr. Bharath Kumar Samanthula
Thesis Sponsor

[redacted]

Dr. Boxiang Dong
Committee Member

[redacted]

Dr. Jiayin Wang
Committee Member

A PRIVACY-AWARE FRAMEWORK FOR FRIEND RECOMMENDATIONS IN ONLINE SOCIAL NETWORKS

A THESIS

Submitted in partial fulfillment of the requirements

for the degree of Master of Science

by

MONA FAHAD ALKANHAL

Montclair State University

Montclair, NJ

May 2019

Acknowledgements

First and foremost, I would like to express the deepest appreciation and thanks to my advisor Dr. Bharath K. Samanthula, you have been a great mentor for me. I am very grateful to Dr. Samanthula for motivating me and clarifying my confusions. I also would like to thank you for encouraging my research and for allowing me to grow as skilled in doing research. Your advice on both research as well as on completing my master program have been invaluable.

Most importantly, I would like to thank my family members for all their support and love over the years. I am fortunate to have such a beautiful mother who supported me in all my pursuits. She has been a constant source of support and encouragement and has made an untold number of sacrifices for the entire family.

I owe thanks to a very special person, my husband, Abdulaziz for his support and understanding during my pursuit of Master's degree that made the completion of this thesis possible. I greatly value his contribution and deeply appreciate his belief in me. I appreciate my little boy Ibrahim, who has been the light of my life for the last year and who has given me the extra strength and motivation to get things done.

Some special words of gratitude go to my friend, Abeer Alsaegh. Who has always been a major source of support when things would get a bit discouraging.

Last but not least, I would like to give special thanks from the deep of my heart to my late father Dr.Fahad Alkanhal, who always believed in my ability to be successful not only in the academic arena but in the whole of my life. You are gone but your belief in me has made this journey possible. Thank you all!

TABLE OF CONTENTS

Page

LIST OF ILLUSTRATIONS

LIST OF TABLES

Table Page

1. INTRODUCTION

1.1. BACKGROUND AND MOTIVATION

Over the past decade, online social networks (OSN) have become an interesting topic in the research community due to its importance not only in the social space but even in many fields such as business, marketing, and politics [1]. OSN became ubiquitous these days due to their simplicity and rapidity [2]. OSN has transformed the public discourse in the community and speed up the distribution of information among people [3].

Online social networks (OSN) mainly focus on sharing information between users to create new social relationships between individuals who share similar interests. Also, OSN provides many other functionalities that make users' lives easier, for example, messaging functionalities such as the "wall" feature where a user can create his/her own messages as well as upload any other type of media such as web links or photos.

As reported in [4], people like to establish relationships with like-minded individuals, a phenomenon referred to as homophily. To facilitate this, OSN provides an interesting functionality called the "friend" recommendation, an application that falls under the concept of interpersonal acquaintance across the world where each user stays in his/her location. As reported in[5], the friend recommendation application is considered as the first service in OSN for creating relations between users by recommending new friends based on diverse metrics such as hobbies and geographical locations. Moreover, the friend recommendation feature enables users to expand their social connections and share information, while keeping the user updated on new developments based on his/her own interests. There are many metrics that the friend recommendation application depends on for recommendations. For example, the “People You May Know” feature in Facebook uses the mutual friend strategy to recommend new friends [6]. In this feature, friend A can be recommended as a new friend to B if both A and B have some common friends. In contrast, the content-based algorithm focuses on user profile information such as hobbies and education. So, A can be recommended as a new friend to B based on how similar their profiles are. In this research, we restrict our discussion to the friend recommendation based on a common-neighbors score whereby a new friend is recommended to a user who is two-hop away and based on the number of mutual friends they have . In an instance, as shown in Figure 1.1, $John$ can be recommended as a new friend to $Jacob$ because of two main reasons. First, $John$ is two-hop away from $Jacob$. Second, $Jacob$ and $John$ have mutual friends $Mary$ and $Michael$.

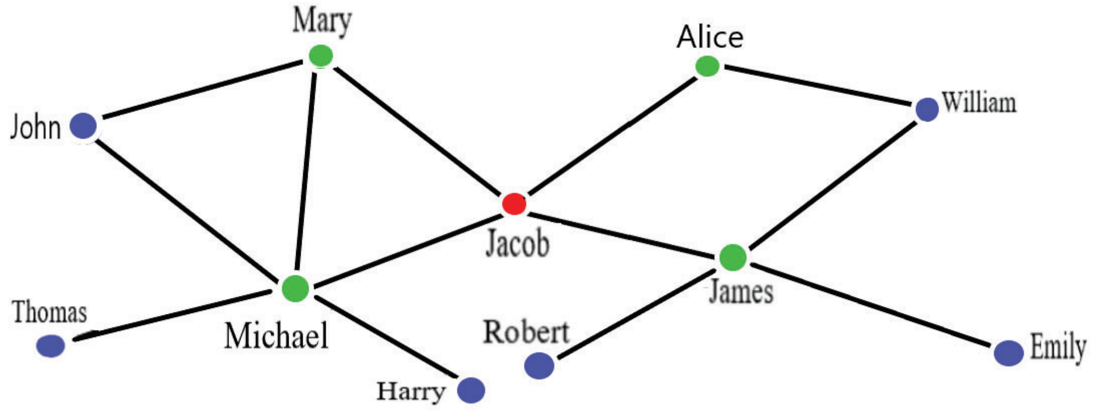


Figure 1.1: Example of two-hop for user $Jacob$ in a social network

One of the most important factors that influences OSN is the privacy of user data. Since the user does not have full control over his/her data, users’ data might be compromised at different levels. Since user data is handled by the online social network provider (OSNP), the user data should be protected even from the OSNP [3]. Nevertheless, as we mentioned before the friend recommendation feature is one of the significant functionalities that influences the privacy of users in OSN. In March 2018, it was reported that Facebook violated the privacy of its users’ data by allowing

Cambridge Analytica, that was working for one of the political parties, to access its users' data without their permissions [7]. Due to the requirement of the user data privacy and the significance of friend recommendations in OSN, there is a strong need to improve the preservation of privacy in the friend recommendation approach in social networks. In this thesis, we propose a privacy-aware model in OSN where users outsource their data to a Cloud environment in a hybrid approach that utilizes Paillier's encryption scheme and AES. Under this framework, we develop a privacy-aware friend recommendation protocol that recommends new friends to users without compromising their privacy [8].

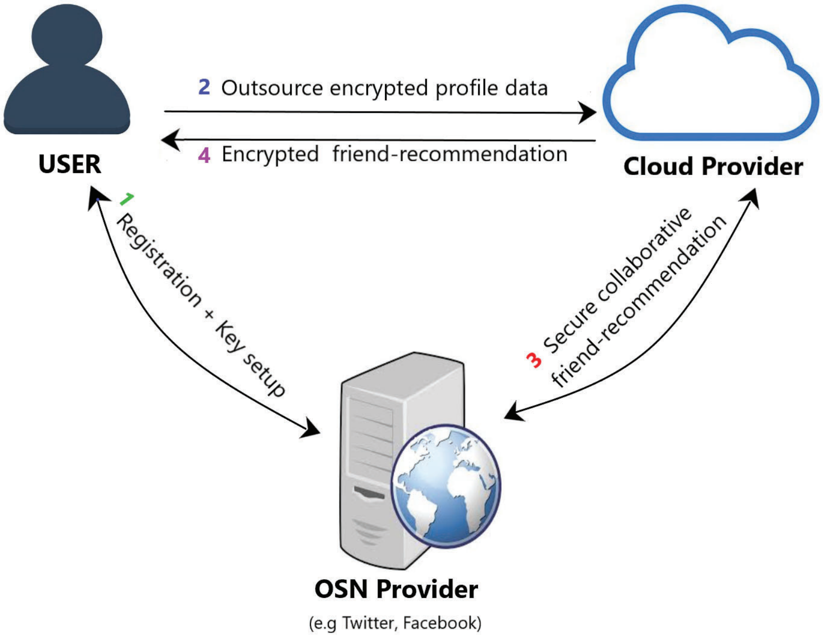

2 Outsource encrypted profile data
4 Encrypted friend-recommendation
USER
Cloud Provider
1 Registration + Key setup
3 Secure collaborative friend-recommendation
OSN Provider
(e.g Twitter, Facebook)

Figure 1.2: Privacy-Aware Friend Recommendation Protocol

1.2. PROBLEM STATEMENT

In our problem setting, as shown in Figure 1.2, we utilize a decentralized architecture that involves three parties: the user, cloud provider, and online social network provider. They are described in the following list:

- *The User:*

  The role of the online social network user is to encrypt his/her data and outsource the encrypted data to the Cloud. The user expects to obtain functionalities from the cloud.

- *The Cloud provider:*

  The Cloud provider assumes the responsibility of the storage task for the user's data. All the user's data is stored in a secure form which has been encrypted by the user. Moreover, the Cloud will serve as the buffer between the user and the online social network provider to improve the privacy-preservation of the user's data.

- *Online Social Network provider:*

  Has the responsibility for providing the functionality to the user. In our model, the online social network provider (e.g., Facebook) can present social network functionalities (e.g., our friend recommendation feature) with respect to the privacy of the user. We preserve the privacy of user's data in the friend recommendation by performing a distributed collaborative analysis between the Cloud and the OSNP while obviating the user's involvement in each step. We utilize a hybrid encryption scheme that contains Paillier's encryption scheme and AES. We also utilize a three rounds permutation function to increase the security of the user's data. We refer to the suggested model as the Privacy-Aware Friend Recommendation (PAFR). The main problem is how the friend-recommendation can be performed in a privacy-preserving manner with high security as each user's friend-list is considered private data. However, for any given user $u_i \in U$ the PAFR should satisfy the following requirements:

– The user's profile data of u_i is never revealed to any party. Specifically, $FRL(u_i)$ is only known to u_i.

– Likewise, $\forall\ R \in FRL(u_i)$, $FRL(R)$ will not be revealed to any user other than R.

– $\forall\ X \in L \rightarrow \varphi(u_i, X) \geq t$.

– At the end of PAFR, L can be accessed only by u_i.

Given a set of n users $U = (u_1, u_2, ...u_n)$, $FRL(u_i)$ denotes the friend-list of the user u_i, and X is a new friend that is recommended to u_i. Let $\varphi(u_i, X)$ denote the common-neighbors score (more details in Section 3) between two users (u_i and X). Based on the common-neighbors score, X can be recommended to u_i if $\varphi(u_i, X) \geq t$, where t denotes the threshold that is chosen by the OSNP. Thus, L is the final recommended list.

As shown in Figure 1.2, there are four main steps for performing PAFR. Step 1 includes a key setup process which is for sharing Paillier's public-key pk between OSNP and the user, and the registration process between the OSNP and the user. In Step 2, each user can outsource to the Cloud his/her encrypted profile data and his/her friend list in a matrix format (more details in Section 4) that is created based on the user's friend-list (which is encrypted by using the OSNP's pk_s). In Step 3, by using the encrypted matrix that the user has outsourced to cloud, the friend-recommendation protocol can be performed in a secure collaborative operation between the cloud and the OSNP. This process can be executed for a set of users in parallel. In Step 4, the recommended friend-list L will be shown to the user when she/he is online. The PAFR is formally defined as follows:

$$\text{PAFR}(u, FRL(u_1), FRL(u_2), ...FRL(u_n), t) \rightarrow L$$

1.3. CONTRIBUTION

In this thesis, we propose a privacy-aware friend-recommendation protocol that employs a hybrid encryption approach that utilizes two encryption schemes: AES and Paillier to increase the security of the suggested model and to preserve the privacy of user data in the OSN. The major contributions of this paper can be outlined as follows:

- *Security* : Compared to the previous approach [6], user's profile data is stored in an encrypted format in the Cloud. The PAFR algorithm does not release any contents or profile data to the Cloud or to the OSNP (more details in Section 4).
- *Accuracy* : Similar to existing work, the suggested protocol achieves a high accuracy (more details in Section 5).
- *Efficiency* : In the proposed model, we utilize optimized Paillier's encryption. Thus, our experiments show that our protocol is efficient (more details in Section 5).
- *Offline User Support* : Once the user has outsourced his/her encrypted profile data to the Cloud, he/she does not have to be involved in any operation in the collaborative-analysis that is performed between the Cloud and the OSNP.

1.4. ORGANIZATION

The remaining of the thesis is organized as follows: Section 2 summarizes the recent related work. We discuss existing background techniques in Section 3. Section 4 describes the suggested model in detail and the complexity analysis for the proposed protocol. We also analyze the security of the suggested model and compare it with the existing work in Section 4. Section 5 shows a comparison of performance between PAFR with existing work and demonstrates the implementation details of the suggested model. Finally, we conclude with the future work in Section 6.

2. THE RELATED WORK

2.1. FRIEND RECOMMENDATION IN OSN

The friend recommendation application falls under the concept of interpersonal acquaintance around the world while each user stays in his/her location. As reported in [7], the friend recommendation considered the first service in OSN for creating relations between users by recommending new friends based on diverse metrics such as hobbies and geographical locations. Moreover, the friend recommendation feature enables users to expand their social network, as well as develop new interests. The friend recommendation application depends on many metrics for recommending new friends. For example, the "People You May Know" feature on Facebook uses friend-to-friend strategies to recommend new friends [8]. In this feature, friend A can be recommended to friend B if both A and B have the same friend D. In contrast, a content-based algorithm leverages user profile information.

In cyberspace, individuals can make new friends easily by communicating with each other using online social networks (OSNs). Similar to what people usually do in real life, OSN users always try to expand their social circles in order to satisfy various social demands, e.g., business, leisure, and academia. In such cases, OSN users may ask for help from their existing friends to obtain useful feedback and valuable recommendations, and further, establish new connections with the friends of their friends (FoFs).

2.2. PRIVACY-PRESERVING FRIEND-RECOMMENDATION IN OSN

In this section, we review some existing work on privacy-preserving friend recommendation in OSN and outline their theses as well as compare them to our suggested model.

2.2.1. Caching technique in OSN for recommending new friends . Nilizadeh et al. [9] proposed a model that preserves the privacy of users' in OSNs to allow the users in social networks to control their own data. Additionally, to protect the confidentiality and integrity of user data, Nilizadeh et al. [9] proposed a decentralized architecture for social networks, referred to as Cachet. The decentralized architecture in [9] consists of a set of distributed untrusted nodes that store user data to ensure availability. The social contacts in the suggested model in [9] act as caches to save the recent updates of social networks and to decrease overhead communication in the network.

2.2.2. Trust relationship method for performing friend-recommendation. Cutillo et al. [10] proposed a model termed as a Safebook which is a type of OSN that applies a decentralized architecture while relying on peer-to-peer architecture to prevent privacy violations that might be accrued due to the centralized architectures. Additionally, Cutillo et al. [10] proposed a set of nodes which are present around the target user in order to store the user's data.

2.2.3. A competent friend-recommendation model. Samanthula et al. [6] suggested a model call $PPFR_h$ which is a friend-recommendation model based on a homomorphic encryption scheme. $PPFR_h$ applied a privacy-preserving friend recommendation feature that utilizes a randomization process. $PPFR_h$ relied on the common-neighbors score for computing the proximity between users in order to make the friend-recommendation. Also, Samanthula et al. [6]

applied the universal hash function to convert the user's ID to an integer form to enhance performance.

There are some drawbacks in [10] and [9] models that our model solves. The user's profile data is stored in another user's hardware in a peer-to-peer fashion, so that if this user is not available then the data cannot be retrieved. [6] utilized the homomorphic encryption scheme to enhance the privacy in the suggested model and involved the target user in order to generate the recommended friend list. Additionally, the efficiency in [6] depends on the size of the network. Thus, the scalability issue can be realized in a large network. In our model, we use the decentralized architecture by involving the Cloud to improve the storage process and to ensure the availability of users' data while ensuring the privacy-preserving of this data.

3. PRELIMINARIES

In this section, we present some concepts that will be used in the proposed solution. These are universal hash function, additive homomorphic encryption scheme, and the common-neighbors score.

Friend List of each user in the network
$FRL(Jacob) = \{Mary, Michael, Alice, James\}$
$Two - hop\ users = \{John, William, Emily, Robert, Harry, Thomas\}$
$FRL(John) = \{Mary, Michael\}$
$FRL(William) = \{Alice, James\}$
$FRL(Emily) = \{James\}$
$FRL(Robert) = \{James\}$
$FRL(Hary) = \{Michael\}$
$FRL(Thomas) = \{Michael\}$

Common neighbor scale for $Jacob$
$FRL(Jacob) \cap FRL(John) = \{Michael, Mary\}$
$FRL(Jacob) \cap FRL(William) = \{Alice, James\}$
$FRL(Jacob) \cap FRL(Emily) = \{James\}$
$FRL(Jacob) \cap FRL(Hary) = \{Michael\}$
$FRL(Jacob) \cap FRL(Robert) = \{James\}$
$FRL(Jacob) \cap FRL(Thomas) = \{Michael\}$
$\varphi(Jacob, John) = \varphi(Jacob, William) = 2$
$\varphi(Jacob, Emily) = \varphi(Jacob, Hary) = 1$
$\varphi(Jacob, Robert) = \varphi(Jacob, Thomas) = 1$

Table 3.1: Friend List of *Jacob* and the common-neighbors score based on Figure 1.1

3.1. UNIVERSAL HASH FUNCTION

To minimize the size of set F to be a set V this can be performed using the universal hash function [11]. Assume that $F = \{0, 1, 2, .., y - 1\}$, and $V =$

$\{0, 1, 2....., m-1\}$ (where $y > m$). Let h symbolize the hash function for given a positive integer $j \in F$ as follows:

$$h_{a,b}(j) = ((a \cdot j + b) \bmod p) \bmod m)$$

Let $\mathbb{Z}_p^* = \{1, ..., p-1\}$ and $\mathbb{Z}_p = \{0, 1, ..., p-1\}$. Assume p is a prime number that $\geq y$, and a, b are chosen randomly from $\mathbb{Z}_p^*$ and $\mathbb{Z}_p$, respectively. Thus, the probability of collision between $h(j)$ and $h(i)$ is $\frac{1}{m}$ where $h(j) - h(i) \bmod m$ is consistently assigned in V, $\forall\ j, i \in F$. The main idea behind using this process is to map each user's ID to integers.

3.2. ADDITIVE HOMOMORPHIC ENCRYPTION SCHEME

There are many types of homomorphic encryption scheme, however due to the efficiency of the additive homomorphic encryption scheme we utilize it for the proposed algorthim [6]. Let *Enc* and *Dec* denote the encryption and decryption of the additive homomorphic scheme. Also, assume *pk* and *sk* show the public-key and private-key respectively. Moreover, consider P_1 and P_2 are plaintexts $\in \mathbb{Z}_N$. There are some significant properties of the additive homomorphic encryption scheme[12], which are as follows:

- It is an additive function: $Enc_{pk}(P_1) \cdot Enc_{pk}(P_2) = Enc_{pk}(P_1 + P_2)$
- Suppose constant $X \in \mathbb{Z}_N$ and $Enc_{pk}(P_1)$:
  $Enc_{pk}(P_1\ )^X = Enc_{pk}(P_1 \cdot X)$
- For any set of cipher-texts C, there will not be any leakage of the plaintexts or any additional information to an attacker.

3.3. THE COMMON-NEIGHBORS SCORE

This method is simply for recommending a new user B to another user R. Suppose R and B are two-hop away in a given social network [13]. Combine the neighbors score between R and B is defined as the number of mutual friends between R and B. To simplify, let φ denote the common-neighbors score :

$$\varphi(B, R) = |FRL(R) \cap FRL(B)|$$

Example1. As shown in Figure 1.1, The main user is $Jacob$ who wishes to make new friends. The direct users of $Jacob$ are $\{Mary, Michael, Alice, James\}$. Assume threshold $t = 2$; the common-neighbors score and the friend list of each user in the network are shown in Table 3.1. Since $\varphi(Jacob, John) = 2$ and $\varphi(Jacob, William) = 2$. Thus, $(William, John)$ are recommended as new friends to $Jacob$.

pk_u	Paillier's public-key of any user u
pk_s	Paillier's public-key for OSNP
sk	Paillier's secrect key
pk_{ku}	AES private-key for any user u
U	A set of users $u_1, ..., u_n$ in OSN
u	For a single user
M'	A set of encrypted matrices M
$FRL(u)$	The friend list for user u
M_u	Un-encrypted matrix for user u
M''	A set of aggregated matrices
$(\pi_{c,1}, \pi_{c,2})$	The random permutation functions known to Cloud
π_s	The random permutation function known to OSNP
t	Threshold value for friend recommendation
PAFR	Privacy-Aware Friend-Recommendation

Table 3.2: Common Notations

4. PROPOSED APPROACH, PAFR

4.1. GENERAL DESCRIPTION

A. OSN and Cloud services:

The general use of the OSN in the current societies is increasingly turning to be the modern trend. The online social networks have changed the way individual remains in touch with others such as family, relatives, friends and the approach that information is spread across communities without any boundaries [14]. The modern way of sharing information and communication gained the attention of a massive base of users to the OSNs. The enormous amount of private data preserved by the network providers have made such data an attractive target for cyber-attacks. Such a subject poses new risks directly related to the user data privacy. For instance, the known social media platform "Twitter" had previously been attacked in which the data including user email addresses, names, encrypted/salted passwords, and session tokens were all compromised [15]. It is clear that OSN has issues related to protection and privacy. Users are entrusting their private information to several social networks without having any guarantees that the method that their information is being processed will secure their private data. Consequently, OSNs are heading to what is known as the "Cloud"; where the social networks can be established to explore the enormous benefits of the paradigms of cloud computing whereby computing resources are offered as services through implementing internet technologies to many individuals [16]. In the Cloud-Based network, the user's private data (such as the data stored in social media networks where the users share with family and friends) will be kept in a trusted cloud storage, which is easily accessible.

B.Information's privacy:

One of the most important factors that influences OSN is the privacy of user data. Since the user does not have full control of his/her data, compromising the user's data might occur at different levels. As the users' data is handled by the OSNP, it should be protected even from the OSNP[3]. Many studies have been conducted on data manipulation by social networks that have access to user accounts and using such data without users' permission which is considered a direct violation of the individual's privacy.

Due to the requirement for user privacy and the importance of friend recommendations in OSN, there is a strong need to improve privacy-preserving function within the friend recommendation approach for online social networks. In this thesis, we propose a privacy-aware framework in OSN where users outsource their data to a Cloud environment in a hybrid approach that utilizes Paillier's encryption scheme and AES. Under this framework, we develop a privacy-aware friend recommendation protocol that recommends new friends to users without compromising their privacy[8]. Based on the components of the proposed model which we have explained earlier in Section 1, our proposed protocol is based on the following assumptions:

- User's profile data is considered as private information and only the user can see the data. In our protocol, the friend list is considered as private data and only the user can access his/her friend list.
- Both Cloud and OSNP act as semi-honest and they do not collude [17].
- The OSNP publishes its Paillier's public-key (pk_s) throughout network.
- Each user u shares his/her Paillier's public-key (pk_u) all over the network.

4.2. OUTSOURCING USER'S PROFILE DATA

```
Algorithm 1 PAFR
Require: $FRL$ for each user $u$ is considered as private data. (Note: $pk_s$ and $pk_u$ are
    known to every party whereas $sk_s$ known only to OSNP, $sk_u$ and $pk_{k_u}$ are known only
    to the user).
1: Data outsourcing: (for each user $u$)
    (a) Encrypts his/her profile data using AES private-key $Enc_{pk_{k_u}}(P_u)$
    (b) Creates matrix $M_u$ based on user's friend list and encrypts it using OSNP's Pillier's
        public-key $Enc_{pk_s}(M_u)$
    (c) Outsource $Enc_{pk_{k_u}}(P_u)$ and $Enc_{pk_s}(M_u)$ to Cloud.
2: Call $SCA$
```

For each user u, the profile data denoted by P_u is encrypted using the user's AES private-key pk_{k_u}. In order to perform the friend recommendation we encrypt the friend list of the user by using Paillier's encryption function and outsource it to the Cloud in a matrix format to help us achieve the friend recommendation. As discussed earlier, OSNP as a service provider publishes pk_s (Paillier's public-key) throughout the network. The reason behind using Paillier's encryption scheme in this step is for performing mathematical operations on encrypted data with high performance[12] and also to achieve the friend recommendation functionality without the need to reveal any user data to the OSNP or to the Cloud. The second step for data outsourcing process is the creation of matrix M_u based on the u friend-list[6]. Each u creates his/her own M_u with mx2 size (m is the number of rows) where M_u is assigned according to $FRL(u)$. For any given user u, we first compute the hash value then assign the user's ID in the first column and assign either 1 or 0 to the corresponding column. It depends on the first column's entry and whether it contains an ID, if so then the corresponding value is 1, otherwise it is 0. More specifically, the M_u is created by applying the universal hash function. To simplify,

$$M_u(h(FRL(u)[i])[0] = FRL(u)[i]$$

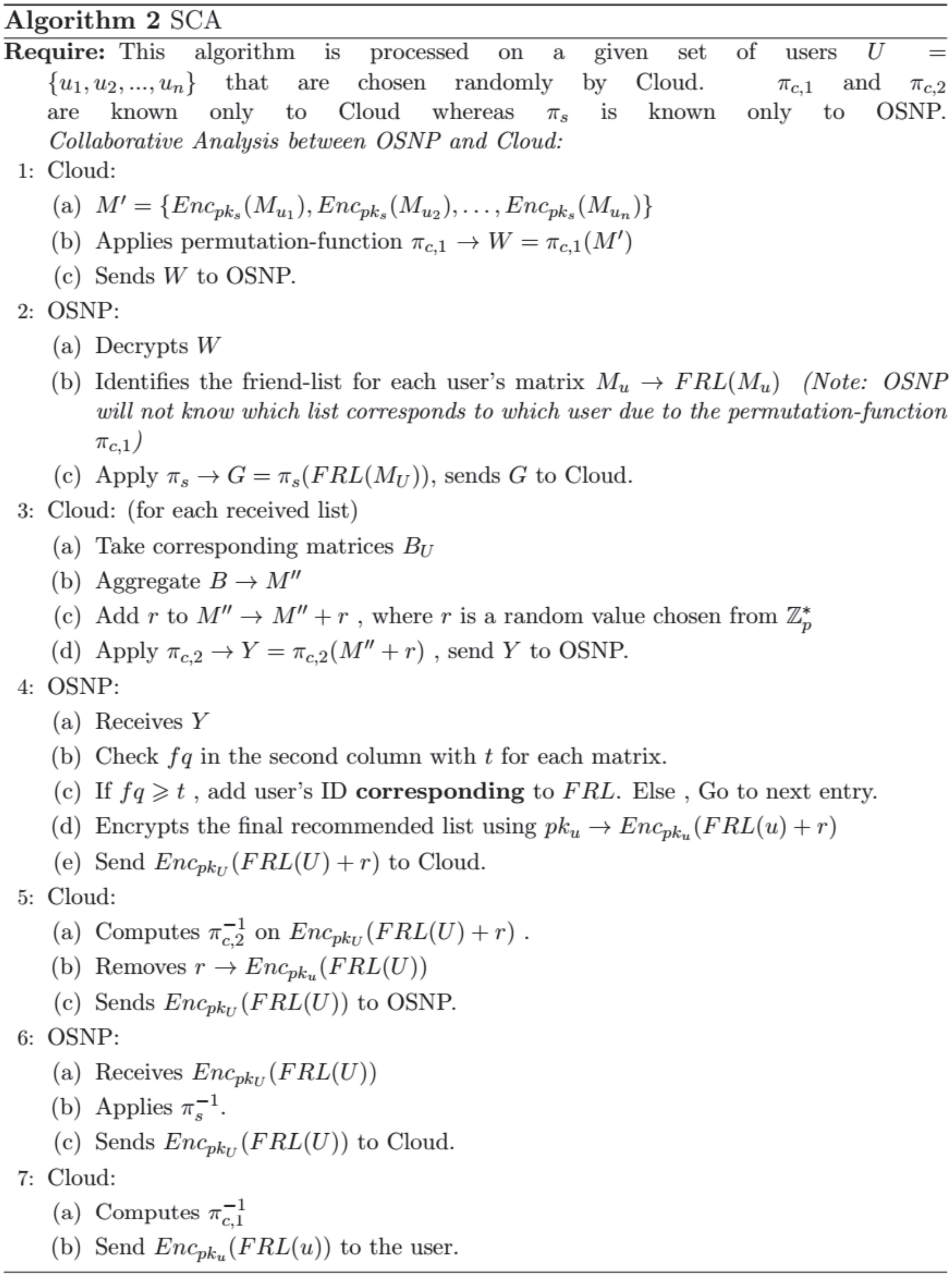
$$M_u(h(FRL(u)[i])[1] = 1$$

Where $FRL(u)[i]$ denotes the user ID of ith friend of u. After the creation process of M_u, by using pk_s each u encrypts M_u and outsources it to the Cloud with his/her encrypted profile data. More specifically,

$$\{Enc_{pk_{k_u}}(P_u), Enc_{pk_s}(M_u)\}$$

4.3. CLOUD BASED COLLABORATIVE COMPUTATION

This section explains the SCA algorithm that is termed Secure-Collaborative Analysis which is invoked after the data outsourcing step. In the SCA, the Cloud and the OSNP will jointly compute the new friend list for a given set of users U.



```
Algorithm 2 SCA
Require: This algorithm is processed on a given set of users $U = \{u_1, u_2, ..., u_n\}$ that are chosen randomly by Cloud. $\pi_{c,1}$ and $\pi_{c,2}$ are known only to Cloud whereas $\pi_s$ is known only to OSNP.
    Collaborative Analysis between OSNP and Cloud:
 1: Cloud:
    (a) $M' = \{Enc_{pk_s}(M_{u_1}), Enc_{pk_s}(M_{u_2}), \ldots, Enc_{pk_s}(M_{u_n})\}$
    (b) Applies permutation-function $\pi_{c,1} \rightarrow W = \pi_{c,1}(M')$
    (c) Sends $W$ to OSNP.
 2: OSNP:
    (a) Decrypts $W$
    (b) Identifies the friend-list for each user's matrix $M_u \rightarrow FRL(M_u)$ (Note: OSNP will not know which list corresponds to which user due to the permutation-function $\pi_{c,1}$)
    (c) Apply $\pi_s \rightarrow G = \pi_s(FRL(M_U))$, sends $G$ to Cloud.
 3: Cloud: (for each received list)
    (a) Take corresponding matrices $B_U$
    (b) Aggregate $B \rightarrow M''$
    (c) Add $r$ to $M'' \rightarrow M'' + r$ , where $r$ is a random value chosen from $\mathbb{Z}_p^*$
    (d) Apply $\pi_{c,2} \rightarrow Y = \pi_{c,2}(M'' + r)$ , send $Y$ to OSNP.
 4: OSNP:
    (a) Receives $Y$
    (b) Check $fq$ in the second column with $t$ for each matrix.
    (c) If $fq \geqslant t$ , add user's ID corresponding to $FRL$. Else , Go to next entry.
    (d) Encrypts the final recommended list using $pk_u \rightarrow Enc_{pk_u}(FRL(u) + r)$
    (e) Send $Enc_{pk_U}(FRL(U) + r)$ to Cloud.
 5: Cloud:
    (a) Computes $\pi_{c,2}^{-1}$ on $Enc_{pk_U}(FRL(U) + r)$ .
    (b) Removes $r \rightarrow Enc_{pk_u}(FRL(U))$
    (c) Sends $Enc_{pk_U}(FRL(U))$ to OSNP.
 6: OSNP:
    (a) Receives $Enc_{pk_U}(FRL(U))$
    (b) Applies $\pi_s^{-1}$.
    (c) Sends $Enc_{pk_U}(FRL(U))$ to Cloud.
 7: Cloud:
    (a) Computes $\pi_{c,1}^{-1}$
    (b) Send $Enc_{pk_u}(FRL(u))$ to the user.
```

CLOUD. As displayed in algorithm 2 (SCA), the Cloud knows two permutation functions $\pi_{c,1}$ and $\pi_{c,2}$. The Cloud performs a permutation function $\pi_{c,1}$ on the encrypted matrices (M'). The main goal for utilizing the permutation function is to prevent the OSNP from knowing which list or matrix corresponds to which user (as we assumed before the friend-list in our protocol is treated as private information) in order to guarantee the privacy of the user's data.

$$M' = \{Enc_{pk_s}(M_{u_1}), Enc_{pk_s}(M_{u_2}), ..., Enc_{pk_s}(M_{u_n})\}$$

$$W = \pi_{c,1}(M')$$

Then, the Cloud sends W to the OSNP.

OSNP. After receiving W, without knowing which matrix belongs to whom, the OSNP will decrypt each matrix $Enc_{pk_s}(M_u)$ and then determine the friend lists for the corresponding friends $FRL(M_u)$. Then, the OSNP sends the permuted friend lists G to the Cloud to allow the Cloud to aggregate the friends' matrices for each user. The important point here is that the OSNP performs the permutation-function π_s to anonymize the friend-list of each user u.

$$FRL(M_U) = \{FRL(M_{u_1}), FRL(M_{u_2}), ..., FRL(M_{u_n})\}$$

$$G = \pi_s(FRL(M_U))$$

CLOUD. For each received matrix, the Cloud extracts corresponding matrices B_U in order to aggregate these matrices (not the final matrix). The aggregated fq will show how many friends are shared between the ID in the first column and the target user. Before sending the aggregated matrices denoted by M'' to the OSNP, the Cloud chooses a random value r from $\mathbb{Z}_p^*$ to add it on each user's ID in the aggregated matrices. The idea behind this step is to hide from the OSNP the friend-list for each user. Finally, the Cloud will permute these aggregated matrices M'' using the second permutation $\pi_{c,2}$ and then sends the permuted matrices to the OSNP for comparison process. For a given user u, $FRL(u) = \{u_2, u_3, u_4\}$, $B_u = \{M_{u_2}, M_{u_3}, M_{u_4}\}$ and the aggregated matrix for u is

$$M''_u + r = \begin{pmatrix} u_2 + r_2 & 2 \\ 0 & 0 \\ u_3 + r_3 & 1 \\ 0 & 0 \\ u_4 + r_4 & 3 \end{pmatrix}$$

$$B_U = \{B_{u_1}, B_{u_2}, ..., B_{u_n}\}$$

$$M'' + r = \{M''_{u_1} + r_1, M''_{u_2} + r_2, ..., M''_{u_n} + r_n\}$$

$$Y = \pi_{c,2}(M'' + r)$$

OSNP. In this step, the OSNP executes comparison process to get the final recommended friend-list. The OSNP will compare fq in the second column of each user's matrix. If $fq \geq t$ then the OSNP adds the corresponding ID to the friend-list $FRL(u) + r$, otherwise; it will skip to the next entry. Then the OSNP encrypts the friend-list $FRL(u) + r$ using Paillier's public-key of the user pk_u. We use the Paillier's public-key of the user to prevent the Cloud from obtaining the friend-list (since Cloud knows r), then send the randomized encrypted friend-list $Enc_{pkU}(FRL(U) + r)$ back to the Cloud.

$$Enc(FRL(U) + r) = \{Enc_{pk_{u_1}}(FRL(u_1) + r),$$

$$Enc_{pk_{u_2}}(FRL(u_2) + r), ..., Enc_{pk_{u_n}}(FRL(u_n) + r)\}$$

Three round inverse permutation-function π^{-1}. :

1. **Cloud**: Applies $\pi^{-1}_{c,2}$. Due to Paillier's property, Cloud eliminates r from each user's ID in the friend-list $Enc_{pk_U}(FRL(U)+r)$ to get the recommended users' IDs $Enc_{pk_U}(FRL(U))$. The most important point here is that the Cloud cannot see the friend list of the user since it is encrypted by Paillier's public-key for the user's pk_u.

$$Enc_{pk_U}(FRL(U)) = \{Enc_{pk_{u_1}}(FRL(u_1)),$$

$$Enc_{pk_{u_2}}(FRL(u_2)), ..., Enc_{pk_{u_n}}(FRL(u_n))\}$$

2. **OSNP**: Computes the inverse-permutation function π^{-1}_s, then sends $Enc_{pk_U}(FRL(U))$ to the Cloud.

3. **Cloud**: Computes $\pi^{-1}_{c,1}$ and then sends $Enc_{pk_u}(FRL(u))$ to the user.

4.4. COMPLEXITY ANALYSIS

The proposed protocol mainly relies on the collaborative operation between Cloud and OSNP. The computation cost differs for each party. On the Cloud side, the computation cost depends on three main operations. The first, is the additive homomorphic (involved during randomization operation) which depends on FRL and the hash domain size (m). Second, the permutation function which depends on the number of received FRL. The third operation is the encryption operation that is performed on a group of users lists which depends on m. Therefore, the computation complexity of the Cloud becomes restricted by $O(FRL \cdot m)$ number of encryption operations. On the other hand, the computation cost for the OSNP can be affected by the following operations. The OSNP performs the public key encryption operation which depends on m and the decryption operation which also depends on m. Also, shuffling users' IDs (involved in the permutation function) that depends on the number of received FRL. Therefore, the computation cost for the OSNP can be bounded by $O(U \cdot FRL \cdot m)$ number of decryption operations.

4.5. SECURITY ANALYSIS

In this thesis, we propose a privacy-aware model where the Cloud securely stores users data and communicates collaboratively with the OSNP without comprising the user's privacy. The suggested model adopts two algorithms: PAFR and SCA. They are as follows:

4.5.1. PAFR algorithm. According to Figure 1.2, the suggested protocol performs the registration and setup key between the user and the OSNP as an initial step. Since our model emphasizes that all data should be encrypted, each user outsources his/her encrypted profile data to the Cloud which is encrypted by using the AES pk_{k_u}. Simultaneously, each user creates his/her own matrix based on his/her friend list and outsources it to the Cloud after encryption using the Paillier public-key of the OSNP pk_s. The reason behind using Paillier's encryption scheme here is because of the property of Paillier's that enables proceeding on encrypted data with high performance [12].

4.5.2. SCA algorithm : The Collaborative analysis between OSNP and the Cloud . Once the Cloud receives the encrypted matrices, it will permute the received matrices using the first permutation function $\pi_{c,1}$ and then sends the permuted matrices to the OSNP. After receiving the permuted matrices and after their decryption by the OSNP, the OSNP identifies the friend list based on the received matrices. Then the OSNP applies its permutation function π_s. The Cloud extracts the corresponding matrices, then aggregates the corresponding matrices in order to have the friend-lists that contain two columns. The first column contains IDs of the users and the second column shows the aggregated frequency that shows how many friends are shared between the ID in the first column and the target user. After that, the Cloud randomizes the aggregated matrices by adding a random value r to each user ID and then the Cloud computes the second permutation function $\pi_{c,2}$. In this way, the OSNP cannot obtain the users IDs due to the randomization process. After all, OSNP compares the frequency fq of each user in the list with threshold t. If $fq \geq t$ then it adds the corresponding ID to the final friend list. Otherwise, it will check the next entry. Finally, the OSNP encrypts the final recommended friend list by using Paillier's public-key of the user.

We utilize the permutation function and the randomization operation to prevent any leakage of user's data between the Cloud and the OSNP since the friend-list for each user is consider as private information in our model.

Three rounds Permutation function.

- Due to Paillier's property, the Cloud eliminates the random value r from the friend list and then applies the inverse permutation function $\pi_{c,2}^{-1}$
- OSNP computes the inverse permutation function π_s^{-1}
- Cloud performs the inverse permutation $\pi_{c,1}^{-1}$

5. PERFORMANCE EVALUATION

5.1. COMPARISON WITH EXISTING WORK

	Common-neighbors scores	**Confidentiality of Outsourced Data**	**Support offline users**	**Scalability**	**High Accuracy**	**Round Complexity**
$PPFR_h$	×	✓	×	×	✓	$O(\|Fr(U)\|)$
PAFR	✓	✓	✓	✓	✓	3

Table 5.1: Comparison between PAFR and $PPFR_h$

In this section, we present the performance analysis of our PAFR protocol and compare it with the $PPFR_h$ protocol given in [6]. Based on Table 5.1, PAFR does not reveal the common-neighbors score to any party whereas $PPFR_h$ releases it to a third party T (e.g. network administrator). In terms of confidentiality, all data is encrypted in both $PPFR_h$ and PAFR included the profile data for users. Also, both protocols utilize the randomization process to maintain the privacy of the user's friend list as well as employ the permutation function π. Additionally, both PAFR and $PPFR_h$ consider the friend list as private information which is only known to the target user. However, for performing the friend-recommendation functionality, PAFR adopts a collaborative operation between the OSNP and the Cloud to generate a friend-list without any need to involve the user. Thus, the data can be pushed to the Cloud while the user stays offline. On the other hand, $PPFR_h$ emphasizes that the target user and his/her friends need to participate in the protocol to extract the friend-list. As a result, the user has to be online in order to obtain the friend-list as well as his/her direct friends. As shown in Table 5.1, PAFR is a scalable approach due to the collaborative analysis between the Cloud and the OSNP that works on a group of user list in parallel whereas in $PPFR_h$ a lot of users have to be involved and as a result it does not scale well. Additionally, similar to $PPFR_h$ our proposed protocol guarantees a high accuracy (e.g the accuracy is 94.1% when the domain hash size m = 7,000 and threshold t = 5). Finally, one of the factors that influences the performance in both protocol is the round complexity. In our proposed protocol PAFR always utilizes three rounds of computation which means the number of operations that are used by PAFR for recommending new friends is not affected by the number of the users. On the other hand, $PPFR_h$'s computation can be affected by the size of the friend list for each user because each friend in the list and the target user participate in order to generate the friend-list.

5.2. EXPERIMENTAL SETUP

In this section, we evaluate the computation cost of the user, cloud and OSNP in our PAFR protocol based on varying parameter values. We implemented the proposed protocol in Java using Intel(R) Core(TM) i5-7300 CPU @ 2.71GHz running Windows 10 Pro with 8.00GB memory.

m	**Standard Paillier**	**Optimized Paillier**
1000	14,216.5 ms	27.25 ms
3000	38,886.5 ms	72.25 ms
5000	65,236 ms	74.5 ms
7000	89,585 ms	91.25 ms
9000	168,448.75 ms	111.25 ms

Table 5.2: User Computation Time

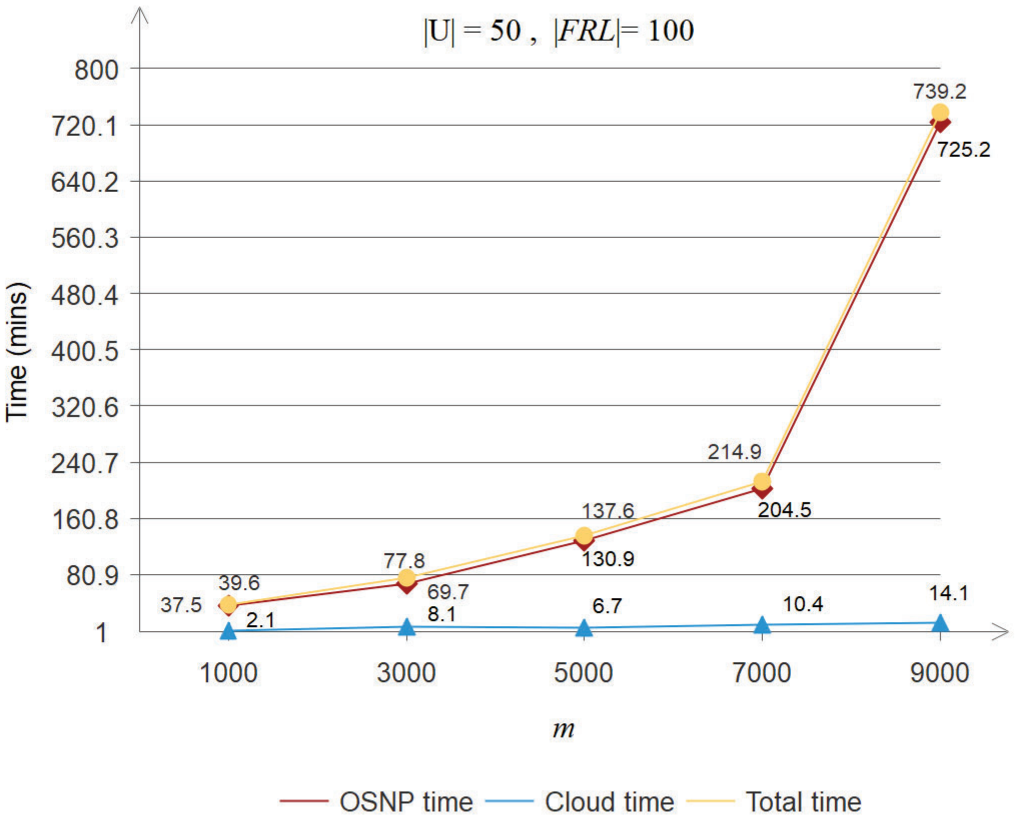


Figure 5.1: Computation time for Cloud and OSNP for varying m

5.3. EXPERIMENTAL RESULTS

5.3.1. User Complexity. As we indicated earlier in Section 5, PAFR supports offline user which means there is no need for the user to be online in order to perform the friend-recommendation application. Also, as shown in Table 5.1, PAFR is a scalable approach due to the collaborative-analysis between OSNP and Cloud that does not require the user to be involved in the process. Thus, we simulate the user computation time for the online situation using optimized Paillier's encryption as well as for the offline situation using standard Paillier. As shown in Table 5.2, we experiment with both online Paillier and offline Paillier on variant sizes of m. The average time for each value of m is directly

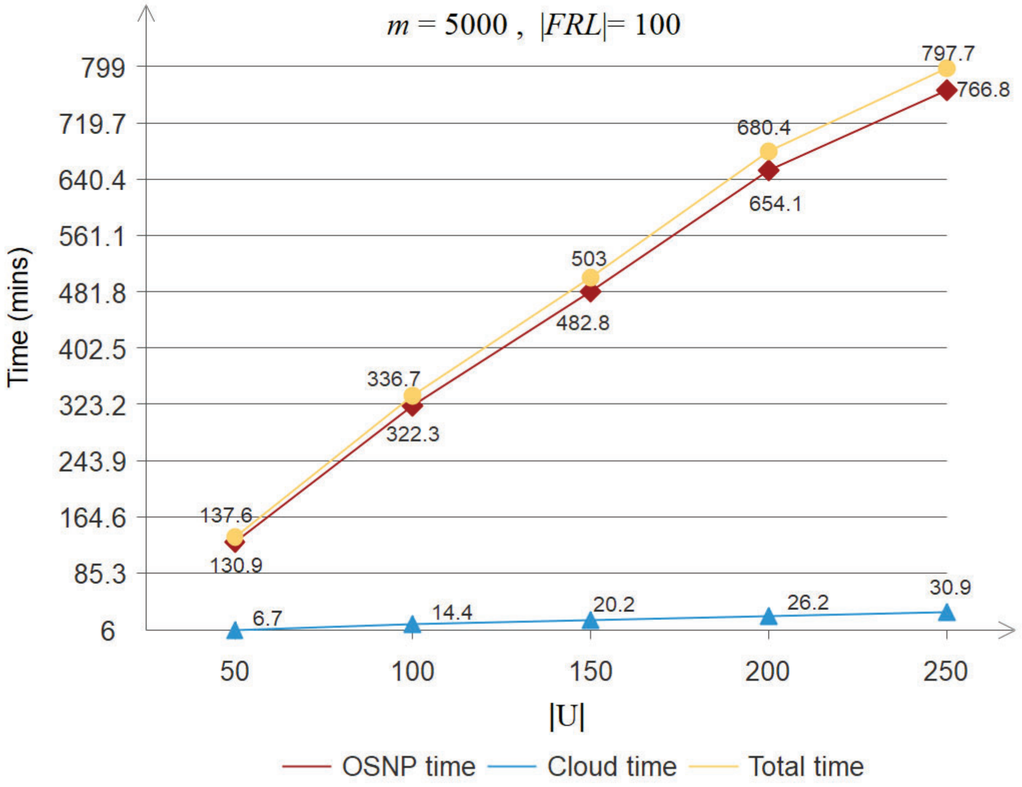


Figure 5.2: Computation time for Cloud and OSNP for varying set of users (U)

proportional to the m size. In addition, as shown in Table 5.2, online Paillier (optimized) takes minimal time compared with offline Paillier (standard). However, we deduce that PAFR can be performed whether the user is online or offline. Additionally, the user takes a few time to outsource his/her data and encrypting his/her friend list using Paillier's encryption.

5.3.2. Computation time for Cloud and OSNP. Based on Figures 5.1, Figure 5.2, and Figure 5.3 there are three significant factors that influence the computation time for Cloud and OSNP: the size of the user's matrix indicated by $|m|$, the size of the friend-list denoted by $|FRL|$, and the number of users U. As shown in Figure 5.2, minimal time is taken by the Cloud since it has the

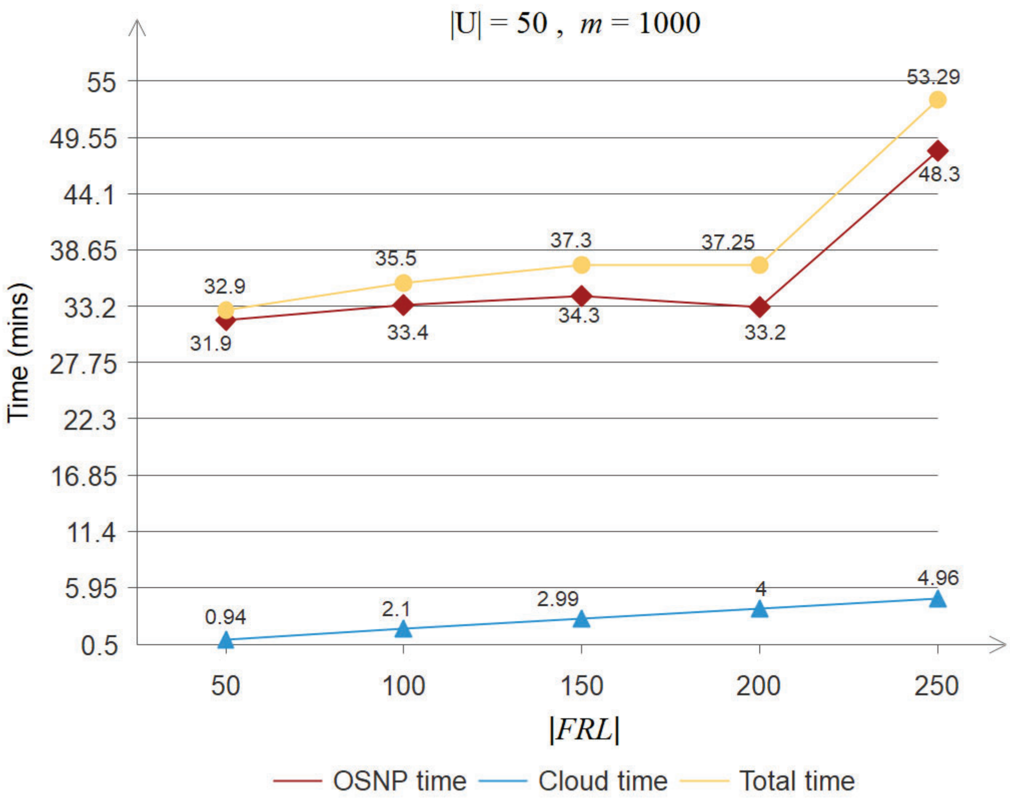


Figure 5.3: Computation time for Cloud and OSNP for varying sizes of friend lists

responsibility for storage and it does not perform any decryption function. On the other hand, the highest amount of time is the total time which is greatly affected by the OSNP. The reason why the OSNP takes a long time is because of the decryption function. Nonetheless, the role of the OSNP could be handed over to a second Cloud, which means that the OSNP will do nothing, and all operations will be performed by two Clouds. In the real-world, the two-cloud model can be played by two different cloud service providers, such as Amazon and Google. As friend-recommendation is not a real-time application, the computation time of PAFR is reasonable compared to the privacy guarantees achieved.

6. CONCLUSION

6.1. SUMMARY

Due to the importance and ubiquity of OSN, we have addressed one of the significant issues that influences the privacy and security of OSN users. In the existing OSNs, social networks users do not have full control over their data. Thus, the user data might be compromised at different levels. Additionally, the existing friend recommendation feature ,which is considered as the first service in OSN for creating relations between users by recommending new friends [5], cannot be performed when the users' friend lists remain private.

In this thesis, we proposed a privacy-preserving friend recommendation framework by utilizing a hybrid encryption scheme which consists of Paillier's encryption and AES. The challenge is to determine how the friend recommendation functionality can work while the friend lists of users are considered as private information. Thus, we utilized Paillier's encryption scheme to allow us to work on the encrypted data. Additionally, in the proposed protocol, each user encrypts his/her profile data using AES and outsources it to a Cloud environment in a matrix format. The suggested protocol (PAFR) consists of three main parties: The user, Cloud provider, and online social networks provider (OSNP). Our protocol is superior to existing work [6] both in terms of security and efficiency. We conducted the experimental evaluation to showcase the computation time of the proposed protocol based on different parameters.

6.2. FUTURE WORK

We outlined the future work for the proposed protocol that we have presented in this thesis as follows:

- *Security*: The proposed PAFR protocol is secure under the semi-honest model. We will investigate how the suggested model can be improved to achieve security against malicious adversaries, for example, if one of the OSN's user is malicious, the protocol should still work.
- *Performance* : Further experiments can be carried out, such as parallel implementation of the proposed protocol, to better assess the performance.
- *Accuracy* : In the proposed protocol, we compute the friend recommendations based on the common-neighbors method. We try to investigate alternative methods for recommendation to enhance accuracy. Additionally, recommending new friends by finding the similarities between the encrypted users profiles is also an interesting direction to pursue further study.

REFERENCES

[1] Carla Mooney. *Online Social Networking*. GREENHAVEN Publishing, 2009.

[2] Sitaram Asur and Bernardo A Huberman. Predicting the future with social media. In *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 01*, pages 492–499. IEEE Computer Society, 2010.

[3] Michael Beye, Arjan Jeckmans, Zekeriya Erkin, Pieter Hartel, Reginald Lagendijk, and Qiang Tang. Literature overview-privacy in online social networks. *Enschede, October*, 2010.

[4] Asim Ansari, Skander Essegaier, and Rajeev Kohli. Internet recommendation systems, 2000.

[5] Dagadu M Jadhavar and VR Chirchi. Friend recommendation system for online social networks. *International Journal of Computer Applications*, 153(12), 2016.

[6] Bharath K Samanthula, Lei Cen, Wei Jiang, and Luo Si. Privacy-preserving and efficient friend recommendation in online social networks. *Trans. Data Privacy*, 8(2):141–171, 2015.

[7] Katherine Bindley Deepa Seetharaman. Facebook controversy: What to know about cambridge analytica and your data, 2018.

[8] JMIT Radaur. A survey on friend recommendation system. 2016.

[9] Shirin Nilizadeh, Sonia Jahid, Prateek Mittal, Nikita Borisov, and Apu Kapadia. Cachet: a decentralized architecture for privacy preserving social networking with caching. In *Proceedings of the 8th international conference on Emerging networking experiments and technologies*, pages 337–348. ACM, 2012.

[10] Leucio Antonio Cutillo, Refik Molva, Thorsten Strufe, et al. Safebook: A privacy-preserving online social network leveraging on real-life trust. *IEEE Communications Magazine*, 47(12):94–101, 2009.

[11] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2009.

[12] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 223–238. Springer, 1999.

[13] Mark EJ Newman. Clustering and preferential attachment in growing networks. *Physical review E*, 64(2):025102, 2001.

[14] SI Auwal, SI Faisal, IM Yusuf, Halis Altun, Mustafa Kaiiali, and Ahmad Samer Wazan. Cloud-based online social network. In *2013 International Conference on Electronics, Computer and Computation (ICECCO)*, pages 289–292. IEEE, 2013.

[15] Wayne Jansen and Timothy Grance. Guidelines on security and privacy in cloud computing. *NIST Special Publication on Computer Security*, 2011.

[16] Matthew M Lucas and Nikita Borisov. Flybynight: mitigating the privacy risks of social networking. In *Proceedings of the 7th ACM workshop on Privacy in the electronic society*, pages 1–8. ACM, 2008.

[17] Wenliang Du and Mikhail J Atallah. Secure multi-party computation problems and their applications: a review and open problems. In *Proceedings of the 2001 workshop on New security paradigms*, pages 13–22. ACM, 2001.