# A Cloud-based Architecture for Mitigating Privacy Issues in Online Social Networks

Mustafa Kaiiali[1], Auwal Iliyasu[2], Ahmad Wazan[3], Adib Habbal[4], and Yusuf Muhammad[5]
[1]Centre for Secure Information Technologies, Queen's University Belfast, UK
[2]The Department of Computer Engineering, Kano State Polytechnic, Nigeria
[3]Département Informatique, Institut de Recherche en Informatique de Toulouse, France
[4]InterNetWorks Research Lab, School of Computing, Universiti Utara Malaysia, Malaysia
[5]The Department of Computer Science, Saadatu Rimi College of Education, Nigeria

**Abstract:** *Online social media networks have revolutionized the way information is shared across our societies and around the world. Information is now delivered for free to a large audience within a short period of time. Anyone can publish news and information and become a content creator over the internet. However, along with these benefits is the privacy issue that raises a serious concern due to incidences of privacy breaches in Online Social Networks (OSNs). Various projects have been developed to protect users' privacy in OSNs. This paper discusses those projects and analysestheir pros and cons. Then it proposes a new cloud-based model to shield up OSNs users against unauthorized disclosure of their private data. The model supports both trusted (private) as well as untrusted (3rd party) clouds. An efficiency analysis is provided at the end to show that the proposed model offers a lot of improvements over existing ones.*

**Keywords:** *Online social network, cloud computing, user's privacy, access control, broadcast encryption.*

*Received June 17, 2016; accepted February 27, 2017*

## 1. Introduction

Online Social Networks (OSNs) have become part of our personal and professional lives. The use of social networking media is increasingly becoming more and more popular. This can be seen from the rate at which social networking sites are expandingtheir users' base.

Organizations have now realized the benefits of establishing a business platform that incorporates the interest of their customers. OSN serves as a customer service tool, as it offers an opportunity to find new customers. It also allows companies to connect and interact with their customers and promote new brands. By utilizing OSNs effectively, companies can reach out to discontented customers directly within their own social media environment to find innovative ways of improving the products or services they have to offer.

OSN has become a powerful tool for use in politics. It has changed the way political campaigns are run. People now use such venues to publicize their political views and to garner support. It has also improved the state of democracy by providing a platform through which people can communicate about a common issue. Candidates and Office holders can now communicate more effectively with a larger audience interactively.

OSNs have changed the way relationships are developed, information is shared, and how people communicate with their families and friends. The new way of communication and information sharing attracted large users to OSNs. Moreover, users may willingly reveal their personal information online.

A survey conducted by consumer national report research centre in 2010 showed that about 40% of OSN users disclose their private data online [12]. The large amount of users' private data maintained by the social networks providers makes them attractive targets for cyber-attacks [13]. This poses new risks related to users' privacy. For example, users' personal information could be gained by an attackerand then used for malicious activities such as scams and identity theft [9].

In general, users are entrusting their private data to multiple social networks without having guarantees on the way in which their data is being held or processed. Twitter was a victim of a successful attack in which information including users' names, email addresses, session tokens and encrypted/salted passwords were compromised [4].

Users' privacy in OSN can be susceptible to insider attacks as well; for example, OSN employees can know which profiles you have visited [7]. Some OSNs may sell their users' data to 3rd party companies for commercial purposes [16]. In addition, government may have a direct access to users' private information through collaboration with OSN providers, as revealed in 2013 [2].

Other potential breaches exist. As an example, a court order could force OSN to reveal information. Alternatively, an accidental release of private data due to a programming error may occur. In mid-2013, a security bug in Facebook causes the exposure of 6

million users' personal information to their contacts [15].

OSN providers have implemented measures that enhance users' privacy, e.g., Facebook can now prevent search engines from searching into users' timelines. This prevents users' profiles from being looked up by someone outside the social network [23]. However, the threat of insider attacks remains as potent as ever. This is due to the centralized architecture of the social network that uses a central storage for users' data. Decentralized OSN looks likea good solution, but it creates other challenges as discussed in the next section.

Several projects such as [11, 16, 17] have been developed to preserve OSN users' privacy. These projects protect users' private data from OSN and other unauthorized people mainly by encrypting the private data and storing it on a 3rd party server. This raises several challenges such as the cost of encryption/decryption operations and friendship revocation. Alternative approaches for decentralized OSN have also been proposed in [1, 19]. These approaches shift OSN from the conventional centralized paradigm, which was based on Client-Server communication model, into Peer-to-Peer system paradigm (P2P). According to P2P principle, a user can manage his/her own data and share it with friends without a need to have central servers. Although the idea is attractive, but still its inability to resolve the challenge of maintaining continuous availability over distributed peers diminishes it. Furthermore, centralized OSN have already established huge subscriber base. For example, Facebook and Twitter have subscriber bases of 1,871,000,000 and 317,000,000 respectively as of January 2017 [22]. Thus convincing people to leave such widely accepted platforms is a huge challenge on its own.

This paper proposes a new communication model for OSN built to explore the benefits of cloud computing paradigm where computing resources are provided as services using internet technologies to multiple users [14, 25]. In this model, users' private data is stored in a cloud storage accessible over the internet.

The proposed paradigm mitigates the discussed privacy issues in current OSNs in a different and more efficient way than what has been done in [11, 16, 17]. It categorizes users' data into public and private. The public data is published normally over OSN, while the private data is stored securely in a cloud storage to be protected against unauthorized access. A link to the private data will be published over the regular OSN with a security challenge to ensure that only authorized users can access the data. When compared to the model described in [19], the proposed model has not abandoned the widely accepted centralized OSN, rather it gets integrated with it.

Furthermore, the architecture is designed to support two different scenarios. The first one is named as "trusted cloud storage scenario" where the cloud storage can be as simple as a mobile device SD card, a hard disk partition on a PC or a lun allocated on an organization's private cloud. The second one deals with the untrusted cloud storage scenario, which can be any 3rd party storage service provider such as Dropbox, Google Drive, etc., Therefore, data encryption is required to guard the data from the curiosity of the cloud service provider [18, 21]. Identity Based Broadcast Encryption (IBBE) technique is adopted in this paper for the first time to achieve scalable access control of OSN users' private data stored in public clouds.

The paper is organized as follows: section 2 discusses the related work and analyses their pros and cons. Section 3 briefly reviews the IBBE schema. The proposed Cloud-based Online Social Network is illustrated in section 4 in two different scenarios (trusted and untrusted cloud service providers). Section 5 discusses the implementation aspects of the proposed model. Finally, the efficiency analysis with comparison to other projects is presented in section 6.

## 2. Related Works

### 2.1. FlyByNight

FlyByNight is implemented as a Facebook application. It acts as a broker between application providers and end users [16]. Private information transmitted through Facebook are encrypted and decrypted in the client-side. OSN is only used to maintain friend relationship.

To use the application, users generates public/private key pair. The private key is encrypted with password and stored in the keys database in the FlyByNight server. The architecture (depicted in Fi gure 1) uses public key cryptography for one-to-one communication while proxy cryptography handles one-to-many communication in order to reduce the client-side computation and storage requirements. In one-to-one communication, a private message is encrypted with the recipient public key then tagged with his/her ID number. Facebook passes the encrypted message to be stored in FlyByNight server.

While in one-to-many communication, a proxy encryption technique is used. This technique enables a 3rd party (proxy) to transform a cipher text generated under one key into a cipher text that can be decrypted by another key without knowing either the content of the message or any of the keys.

To use the proxy encryption for group communications in FlyByNight, the user has to create a group associated with a key pair. To add a friend to the group, user creates a new key pair and a proxy key for this friend. The proxy key is stored in FlyByNight while the key pair is sent securely to the new friend encrypted using his/her public key.

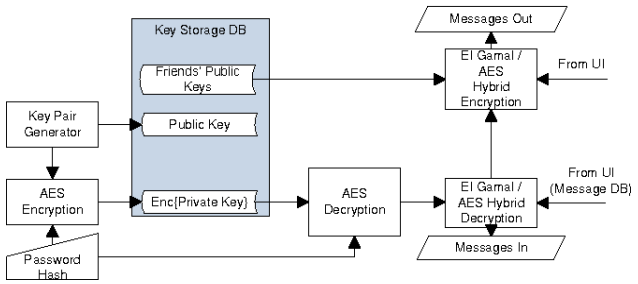


Figure 1. FlyByNight architecture [16].

If the user wants to publish a message to a group of friends, he/she has to encrypt the message using the group public key and stores the encrypted message in FlyByNight server. If a friend in the group wants to read this message, he/she asks FlyByNight to transform it to a new message encrypted using his/her own public key with the help of his/her proxy key stored at FlyByNight server. He/she then uses his/her private key to finally decrypt the message. Below is a description of El-Gamal [8] based proxy encryption technique used in FlyByNight.

Given an El-Gamal cryptosystem with global elements ($q$) and ($g$) where ($q$) is a prime number and ($g$) is its primitive root. User A generates private/public group key pair ($x$, $g^x$). To add user B to the group, user A generates another private/public key pair ($b$, $g^b$) for user B. To send a message ($m$) to the group, user A ecrypts it under the group public key ($g^x$) producing the cypher text ($C_1$, $C_2$) as shown in Equation (1):

$$C_1 = g^k, C_2 = m \cdot (g^x)^k : k \text{ is a random number} \quad (1)$$

This message can be decrypted by the group private key ($x$) as shown in Equation (2):

$$\frac{C_2}{C_1^x} = \frac{m \cdot g^{xk}}{g^{kx}} = m \quad (2)$$

However, a proxy key that enables a message encryped under the group's public key to be decrypted by B's private key is generated using Equation (3):

$$k_p = x - b. \quad (3)$$

The proxy key ($k_p$) is used to transform any message ($C_1$, $C_2$) encrypted by ($g^x$) into another message ($C_3$, $C_4$) that can be decrypted by ($b$) using Equation (4):

$$C_3 = C_1, C_4 = \frac{C_2}{C_1^{k_p}} = \frac{m \cdot g^{xk}}{g^{k(x-b)}} = m \cdot g^{kb} \quad (4)$$

FlyByNight has the following drawbacks:

- Revoking a friendship from a group requires re-computation of new key parameters that results in communication overhead.
- Users can connect only with one group at a time.
- It increases the burden over FlyByNight server to transform the encrypted message to another form using proxy key.
- Images are not protected.

## 2.2. Facecloak

FaceCloak is implemented as a web browser extension. It protects the user's published data from unauthorized users as well as from OSN providers [17]. When a user first interacts with the application, FaceCloak generates three keys: a master key, a personal index key and an access key. The master and personal index keys are shared between the user and his/her friends, while the access key is kept locally.

The personal index key is used to encrypt the user's profile information. The access and master keys are used whenever users want to post messages. To place/modify a post, a user first has to provide his/her access key. The function of the access key is to prevent attacker, who may already be aware of the master key (because it is shared), from replacing the published posts. FaceCloak then directs the user to encrypt the post information using a symmetric key derived from his/her master key. The user also generates a Message Authentication Code (MAC) for each encrypted message to preserve message integrity. The encrypted message is transmitted over a TLS connection to a FaceCloak server. At the same time, FaceCloak generates fake information and send it to OSN site. The FaceCloak server also stores each encrypted message together with the fake information to serve as its index.

When an authorized friend wants to view the information posted, he/she queries the 3rd party server for the original information using the published fake message as its index. He/she then decrypts the original information obtained from the 3rd party server over a TLS connection and then replaces the fake information retrieved from the social networking site with it (Figure 2).

FaceCloak has the following drawbacks:

- Initialization steps have to be repeated whenever a user creates/revokes a friendship.
- All friends have the same level of data access. There is no proper fine-grained access control.
- Fake messages can cause extra load on the database system.

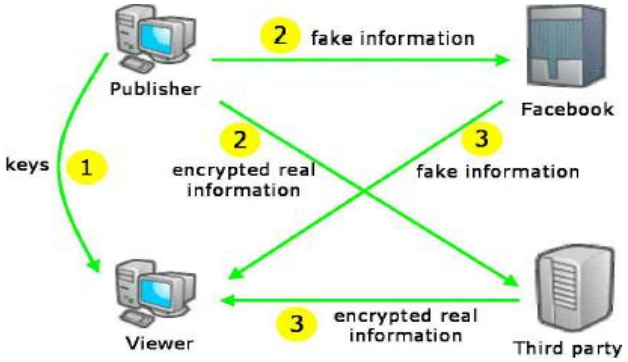


Figure 2. FaceCloak architecture [17].

### 2.3. NOYB

NOYB is a mechanism that uses encryption and transformation to preserve the privacy of users' private information [11]. User's private information is represented by a set of attributes, like name, gender, age etc. NOYB partitions these attributes into multiple atoms. For example, name and gender can be a single atom while age can be another one. A set of atoms represents a single class. A single dictionary stores all users' atoms that belong to the same class.

Each atom in the dictionary is associated with an index and each user knows his/her atoms' indexes. Indexes in the dictionary are then encrypted using a shared key. The resulting encrypted index looks like a real index of another user. This process helps to transform atoms between different users that yields to a fake and realistic data. Only users who know the shared key can reverse the substitution. To make the job harder on the attacker, NOYB adds to the dictionary an extra fake data for non-existing users. [11] proposed an illustration example, assume Alice's name, sex and age (Alice, F, 25) is partitioned into two atoms: (Alice, F) and (25). The first atom is substituted with (Bob, M) say, and the second with (28) say, from Bob and Charlie respectively based on the encrypted indices. Alice's friends can reverse the encryption to recover Alice's information, as they own the shared key. While Alice's atoms may similarly show up in other users' profiles so that an adversary cannot piece together her atoms.

The limitations of NOYB include:

- Users have no access control on their shared data.
- Each time a friendship is revoked, a new key has to be negotiated, which results in extra complexity.
- There is no well-defined key management.
- It only preserves privacy of users' profile not data.

### 2.4. My3

My3 is a P2P based OSN [19]. In My3, users manage their data and share it with friends without the use of a 3rd party. The system uses resources contributed by each user to store a Distributed Hash Table (DHT), which saves information that, enable users to track each other and exchange data. To increase the system availability, a user entrusts some of his/her friends called Trusted Proxy Set (TPS) to host and enforces access control over his/her data. TPS members for a given user are chosen based on geographical locations and online time. Thus user's other friends can access his/her data through the TPS members as long as their online time overlaps.

Some of the drawbacks of this approach include:

- Too much trust on the TPS members.
- The challenge to maintain continuous availability via distributed peers as users can only exchange data if and only if their online times overlap.

## 3. Identity Based Broadcast Encryption

Broadcast encryption schemas [5, 6, 10, 20] are cryptosystems that deliver encrypted messages over a broadcast channel so that only dynamically chosen subset of users can decrypt the message. The set of qualified users can be dynamically specified in each broadcast message. Users do not need to update their private keys for each broadcast. It is fully collusion resistant in that even if all users outside the dynamically chosen set collude, they cannot decrypt the message.

Broadcast encryption has many applications including DVD content protection and satellite TV subscription services [5]. However, it has never been employed before to mitigate the privacy issues of OSNs. Several broadcast encryption schemas based on bilinear pairings have been proposed [5, 6, 10, 20]. The one presented in [20] is adopted in this work as its number of users need not to be predeterminedat setup.

Hereby, a brief review about the IBBE schema is proposed. It is composed of four algorithms: Setup, Key-Extract, Encrypt and Decrypt. The intention of this section is not to discuss the mathematical foundations of IBBE but to briefly demonstrate the IBBE process so to be deployed into a cloud paradigm to effectively mitigate the privacy issues in OSN.

### 3.1. Setup (k, n)

It is executed by the manager, a person who wants to broadcast a message. It takes a security parameter (k) and the maximum number of receivers for a single broadcast (n). It generates the Master Key (MK), preserved by the manager, and the shared Public Key (PK).

### 3.2. Key-Extract (MK, $ID_i$)

It is run by the manager once for every receiver. It takes the MK and the identity string of a receiver ($ID_i$) as an input. It produces a unique private key for that receiver ($Pr_i$) as an output. $Pr_i$ is sent securely to the corresponding receiver.

### 3.3. Encrypt (PK, S, M)

Given the public key (PK), the set of receivers' identities (S), and the broadcast message (M). The algorithm outputs a symmetric key ($K_{IBBE}$) and a header (Hdr). The symmetric key is used to encrypt the broadcast message generating the ciphertext (C), while Hdr is later used to help a receiver in deriving the symmetric key if and only if his/her identity is listed in (S).

### 3.4. Decrypt (PK, $Pr_i$, S, Hdr, C)

This algorithm is run by a receiver. It takes as an input the PK, the user's private key ($Pr_i$), the set of receivers' identities (S), the Hdr and the message ciphertext (C). The algorithm can extract the symmetric key ($K_{IBBE}$) if the receiver's identity was included in the identities' set (S) used in the previous stage to generate $K_{IBBE}$, otherwise it generates an incorrect key value.

The next section shows how a Cloud-based Online Social Network architecture can be used to mitigate the user's privacy issues in OSNs. IBBE is deployed in the second scenario of the proposed model where the cloud storage is considered untrusted. The efficiency analysis presented in Section 6 shows that IBBE is a very effective method to be adopted for such scenarios.

## 4. Cloud-based Online Social Network

This section proposes the Cloud-based Online Social Network (COSN) in two different scenarios; trusted and untrusted cloud storage. The abstract concept of the first scenario was presented in [3]. The architecture protects users' privacy without abandoning the widely accepted centralized OSN platforms, which is kept to serve as a friends' management portal, where friendships/groups are created and managed.

COSN is designed to achieve the following goals:

- *Confidentiality*: Protecting users' private data from unauthorized access even from OSN providers.
- *Access Control*: Allowing users to define a flexible access control over their data.
- *Low Friendship Revocation Cost*: Allowing users to easily revoke a friendship without the need for renegotiating new security parameters.
- *Economic Use of the Internet Space*: Enabling users to share their data with their friends across various OSNs while storing their data in a single cloud storage rather than replicating it over various OSN servers. This can save the internet space.

### 4.1. Trusted Cloud Storage Scenario

In this scenario, the cloud storage is either provided by a trusted 3rd party or it can be a personal storage accessible over the internet. It embraces the following three services:

#### 4.1.1. Initialization Service (Executed Once)

This service allows the user to create a public/private key pair. The private key is stored locally; the public key is shared with user's friends.

#### 4.1.2. Data Upload Service

The user uploads new data to the cloud storage through the Data Management Portal (DMP) (Figure 3). DMP calls the OSN-Cloud Interface to fetch the user's list of OSN friends/groups. The user selects towhom he/she wants to grant the data access right. This creates an access-list stored at the DMP. A copy of this list is sent to OSN through the OSN-Cloud Interface along with a link to access the published data shared with the same list of members. Knowing only this link will not be sufficient to access the data.

#### 4.1.3. Data Access Service

When a client, a COSN user's friend, wants to access the uploaded data, he/she clicks on the published link. The link is prepared to trigger the client-side API to generate an Authenticator message that certifies the client as a legitimate friend. The client-side API communicates with the OSN-Cloud Interface to present the Authenticator as shown in Equation (5).

$$\text{Client} \xrightarrow{\text{Authenticator}} \text{OSN} - \text{Cloud Interface} \quad (5)$$

$\text{Authenticator} = E(Pr_{client}, [ID_{client} \,||\, N_1])$

- $Pr_{client}$: The client's private key used for digital signature.
- $N_1$: A random number used to prevent replay attacks.

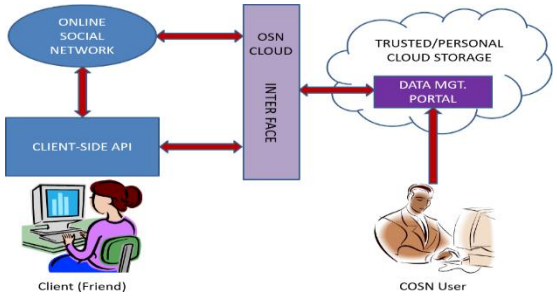


Figure 3. A high-level architecture for trusted cloud scenario.

The OSN-Cloud Interface decrypts the Authenticator message using the client's public key fetched from the keys database stored and managed by the DMP. Then it compares the client's ID against the data access-list stored at the DMP. If there is a match, the OSN-Cloud Interface generates a SessionNotifier message depicted in Equation (6). It encrypts a session key, the COSN user's ID, and a function of the received nonce with the client's public key. The output is further encrypted with the user's private key to assure to the client that he/she is communicating with the right entity. With the help of the negotiated session key, a secure channel will be established to transfer the data securely from the cloud storage to the client.

$$\text{OSN} - \text{Cloud Interface} \xrightarrow{\text{SessionNotfier}} \text{Client} \quad (6)$$

$\text{SessionNotifier} = E(Pr_{user}, E(Pu_{client}, (K_s, ID_{user}, F(N_1))))$

- $Pr_{user}$: The COSN user's private key used for digital signature.
- $Pu_{client}$: The client's public key, so that only the client can decrypt this message.
- $K_S$: The session key generated to establish a secure

channel for data transfer.
- $ID_{user}$: The COSN user's ID.
- $F(N_1)$: A function of the received nonce which indicates that this message is a reply to the received Authenticator message to mitigate replay attacks.

If an illegitimate friend gets the published link, he/she will not be able to generate the Authenticator message of a legitimate friend so he/she will not be able to access the data. Furthermore, if he/she was able to capture an Authenticator message of a legitimate friend, he/she will not be able to decrypt the SessionNotifier message to get the session key through which data is transferred securely. Moreover, an intruder will not be able to spoof the COSN user's identity to deceive the client with fake data, as he/she cannot digitally sign the SessionNotifier message on behalf of the COSN user.

## 4.2. Untrusted Cloud Storage Scenario

In this scenario, user's private data has to be further protected from the curiosity of the cloud service provider. To achieve this additional requirement, data has to be stored encrypted in the cloud. This raises a set of challenges:

- Proper key management and distribution mechanism is needed.
- Encryption and decryption overhead while uploading and accessing the data.
- Friendship revocation overhead while decrypting and re-encrypting the published data using different security parameters.

Many proposals have been made to address all of these challenges. However, all of the proposed models suffer from certain limitations as discussed earlier in section 2. This section introduces a cloud-based technique that adopts an IBBE schema to come up with a model capable of addressing all of the pre-mentioned challenges efficiently. The architecture, depicted in Figure 4, is equipped with the following services:

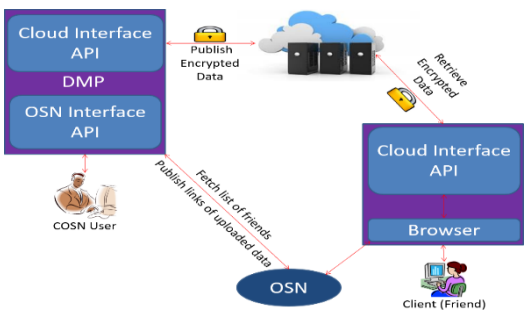


Figure 4. A high-level architecture for untrusted cloud scenario.

### 4.2.1. Setup and Key Distribution Service

Being the manager of the IBBE schema, the COSN user executes the IBBE'sSetup and Key-Extract algorithms (refer to section 3). The Setup algorithm initializes the schema'ssecurity parameters such as the bilinear groups, generators, and cryptographic hash functions. Then it outputs the master and public keys. While the Key-Extract algorithm enables the COSN user to create a unique private key for each friend of him/her using their OSN identities (obtained through the DMP-OSN Interface) and the master key. The private keys are then sent to the friends in an out-of-band mechanism such as email, SMS, etc. This service is executed for one time only.

### 4.2.2. Data Upload Service

This service is executed whenever the COSN user wants to publish data. The user selects the identities of whom he/she wants to grant access to the data, in other words, he/she creates an access-list (S). Then he/she executes the IBBE'sEncrypt algorithm to generate a Header (Hdr) and a symmetric key ($K_{IBBE}$) used to encrypt the data.

The access-list (S) and the header (Hdr) are labelled as a broadcast-header. The encrypted data together along with the public key and the broadcast-header are published on a 3rd party cloud storage through the DMP-Cloud Interface.

Finally, DMP publishes a link to the encrypted data on the user's OSN personal page through the DMP-OSN Interface. The link post is published along with a data preview (title, type and size of the data) and with the same rights of the access-list (S).

### 4.2.3. Data Access Service

This service is executed whenever a client (a friend) wants to access the published data. He/she clicks on the published link. This triggers the client-side API to establish a connection with the cloud storage whose address is obtained from the link. The client gets the encrypted data together with the public key and the broadcast-header (S and Hdr) from the cloud.

Then the client-side API runs the Decrypt algorithm of the IBBE schema and generates $K_{IBBE}$ out of the public key, the broadcast-header, and the client's private key. It then uses the key to decrypt the data.

### 4.2.4. Friendship Revocation Service

COSNhas a very flexible friendship revocation mechanism. Unlike other projects, discussed earlier in Section 2, COSNdoes not require to rerun the "Setup and Key Distribution" service for every friendship revocation.

The set of authorized friends are dynamically specified in IBBE for each shared data by just including their identities in the access-list (S). Therefore, a friendship can be revokedby simply removing the corresponding friend's identity from the access-list (S) associated with the newly shared data while maintaining all other security parameters that have been negotiated by the "Setup and Key

Distribution" service unchanged. According to IBBE, a revoked friend who is aware of the previously exchanged security parameters will never be able to decrypt a new IBBE message as far as his/her identity is excluded from the corresponding access-list (S).

The proposed model has the following drawbacks:

- It cannot prevent a revoked friend from accessing old data previously published with an access-list that includes his/her identity.
- It cannot allow a new friend to access old data as it has been published without his/her identity.

However, these drawbacks can be considered as intended features in many scenarios. When we revoke a friend, we usually do not bother about his/her accessibility to previously published data as he/she had already accessed them before while being a friend. Moreover, we are not always aware of all previously published data so we may not like the new friends to access all previously published data.

Nevertheless, a new friend can access previously published data if it had been published with an access-list containing the group identity to whichthe new friend belongs rather than containing the individual users' identities.

## 5. COSN Implementation Aspects

This section covers the implementation aspects of the proposed model. It gives a high-level design overview and describes various components involved.

### 5.1. Trusted Cloud Storage Scenario

Figure 5 depicts a high-level system setup of this scenario built upon an Android platform as a case study. It shows various system components involved and highlights the interactions between them. Typically, "DMP F.E." and "DMP B.E." perform the "DMP" and "OSN-Cloud Interface" roles discussed earlier in Section 4 respectively.

#### 5.1.1. Cloud Storage

It is a personal trusted storage represented by a 32GB SD card of the user's personal mobile device and made accessible over the internet (via Wi-Fi or mobile data networks) through DMP F.E. web portal (discussed later in this section).

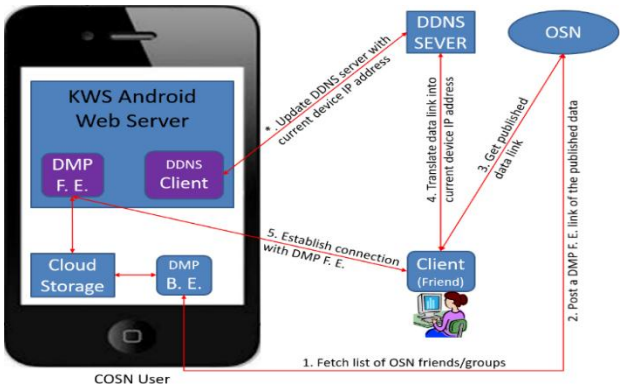


Figure 5. An Android based trusted cloud setup.

#### 5.1.2. Data Management Portal Back End: (DMP B.E.)

It is an Android APK that enables COSN users to publish their data on the cloud storage. Whenever a user is willing to post some data, DMP B.E. interacts with the user's OSN account to fetch his/her list of OSN friends/groups. The user selects to whom he/she wants to grant the data access. This creates an access-list stored along with the published data in the cloud storage. Then DMP B.E. publishes a link (accessible through DMP F.E. web portal) to the uploaded data as a post on the user's OSN personal page andshares it with every member in the access-list.

#### 5.1.3. Data Management Portal Front-End (DMP F.E.)

It is a web application hosted at KWS Android Web Server. It is responsible for handling clients' requests to access user's published data and enforces the access control policy on the shared data following the same protocol illustrated earlier in section 4.

#### 5.1.4. DDNS Client

It is a Dynamic DNS web agent. By default, KWS web server is shipped with a DDNS agent so it is not required to install a separate one. The DDNS client is responsible for updating the correspondingDDNS server with the IP address of the mobile device.

Each COSN user has to get a personal public domain name linked to his/her DMP F.E.IP address. Mobile devices are used to obtain different IP addresseswhile moving across various Wi-Fi or mobile datanetworks. Hence, the DDNS agent is needed to dynamically update the corresponding DDNS server with the newly obtained IP [24].

#### 5.1.5. KWS Android Web Server

It is a mobile web server that can be used to host websites on Android devices and to serve files over http/https. Currently, KWS can handle up to 999 concurrent connections if the device's hardware can support it [26]. This number is good enough to handle the concurrent friends' access of a person with

small/medium popularity. As the device's hardware is becoming more efficient, this number will be increased to be able to serve concurrent friends' requests of a popular person. Moreover, if we consider that DMP F.E. is meant justto handleprivate data access stored at the personal cloud storage while the public data is published normally on OSN, then this schema can be good enough even to host the personal pages of popular persons.

### 5.2. Untrusted Cloud Storage Scenario

Figure 4 illustrates the second setup of COSN where the cloud storage is provided by a 3rd party cloud service provider. Unlike the first setup, components such as KWS web server and DDNS agent are not needed since the cloud storage is hosted by a 3rd party cloud service provider that takes the responsibility to store the published data and serve clients requests.

Likewise, DMP in the second setup is a single component, unlike the first setup where DMP consists of a front-end web application (to serve the clients) and a back end application (to upload the data). DMP interacts directly with OSN and the cloud storage through its "OSN Interface API" and "Cloud Interface API" respectively. The "OSN Interface" communicates with OSN to fetch the list of friends/groups and lets the user define data access control. Then it sends a link of the published data to OSN after the data is uploaded into the cloud storage through the "Cloud Interface". The "Cloud Interface" itself provides services to execute the Setup, Key-Extract, and Encrypt algorithms of the IBBE schema to encrypt the data then publish it onto the cloud storage.

The client browses the user's OSN personal page. If he/she clicks on a published data link, this triggers the client-side "Cloud Interface API" to fetch the shared data from the cloud and execute the IBBE Decryptalgorithm to decrypt the data.

## 6. Efficiency Analysis of COSN

COSN offers many advantages when compared with earlier models such as FlyByNight, FaceCloak, NOYB, and My3 based on the following perspectives:

### 6.1. Scalability

FlyByNight and FaceCloak projects employed 3rd party servers to store users' private data. For these models to be scalable, FlyByNight and FaceCloak projects need to establish large data centres as big as the Facebook's ones in order to store the users' published data.

However, the first scenario of COSN offers a much more feasible solution, since each user has his/her own cloud storage in which he/she stores his/her data without the need to recourse to a 3rd party server.

Moreover, the second scenario of COSN offers each user the ability to host his/her data in different 3rd party cloud storage service providers such as Dropbox, Google Drive, etc. This solution is absolutely more available than constructing large data centres for FlyByNight or FaceCloak.

### 6.2. Centralized Data Management

COSN enables users to store and manage their data in one place, which is the cloud storage. This feature, which none of the previous projects (FlyByNight, FaceCloak, NOYB or My3) provides, enables simultaneous integration with multiple OSNs. Users can share their data with their friends on various OSNs while having the data stored in one place instead of replicating the same data over various OSN networks.

### 6.3. Efficient Friendship Revocation Mechanism

Most of the earlier projects lack efficient friendship revocation mechanism as discussed in section 2. Creating/revoking a friendship requires key regeneration and redistribution. This is a costly process.

Table 1. Comparison between COSN and the reviewed approaches.

| | Required Encryption | Access Control Support | Trust on 3rdParty | Friendship Revocation Cost | Scalability | Availability |
|---|---|---|---|---|---|---|
| **FlyByNight** | Asymmetric | Available | Not Required | High | - | Available |
| **FaceCloak** | Symmetric | Partially | Not Required | High | - | Available |
| **NOYB** | Symmetric | Partially | Partially | High | - | Available |
| **My3** | Symmetric | Partially | Partially | High | Scalable | Partially |
| **COSN - Scenario 1** | Not Required | Available | Not Required | Low | Scalable | Available |
| **COSN - Scenario 2** | Symmetric | Available | Not Required | Low | Scalable | Available |

COSN offers much more efficient friendship creation/revocation mechanism. Friends are simply added by adding their identities to the dynamically created access-list. Moreover, users need not to distribute an encryption key for every published data. This key can be generated instantly by the friends themselves (using theIBBE schema) as long as their identities are included in the access-list. In addition, friendship revocation is simply accomplished by excluding a friend's identity from future created access-list without the need for regenerating and redistributing keys.

Table 1 compares the existing OSN privacy projects discussed earlier in section 2 with the proposed COSN scenarios based on various performance parameters.

## 7. Conclusions

OSNs have touched our lives in many positive ways. People now acquire more information, more knowledge and have better opportunity to interact using OSNs. However, the privacy issue is raising a serious concern. All user's published data on OSN has

to be protected against unauthorized friends' access and even against OSN providers. Many research works have been done to encounter the OSN privacy issues.

This paper presents a novel cloud-based model for mitigating privacy issues in OSN. In the proposed model, users' private data is protected from unauthorized friends' access as well as from the curiosity of the service provider. It offers two paradigms, the first one stores the data in a trusted cloud storage, while the second one adopts an IBBE schema to securely store the data on a 3rd party untrusted cloud storage.

The paper shows that the proposed model is more efficient when it comes to scalability, integration with multiple OSNs, dynamic access-list membership support, and efficient friendship revocation mechanism. We believe that our work sorts out many OSN privacy issues and also offers many other desirable features.

## Acknowledgement

We would like to express our sincere gratitude to Prof. Sakir Sezer, Dr. Suleiman Y. Yerima, and Dr. BooJoong Kang from the Centre for Secure Information Technologies (CSIT), ECIT, QUB, for the assistance they have made to this research work during the revision stage.

## References

[1] Albertini D. and Carminati B., "Relationship-Based Information Sharing in Cloud-Based Decentralized Social Networks," *in Proceedings of the 4th ACM Conference on Data and Application Security and Privacy*, Texas, pp. 297-304, 2014.

[2] APUZZO M., "What's the problem with PRISM?," https://www.yahoo.com/news/whats-problem-prism-203441280.html, Last Visited, 2013.

[3] Auwal S., Faisal S., Yusuf I., Altun H., Kaiiali M., and Wazan A., "Cloud-Based Online Social Network," *in Proceedings of the International Conference on Electronics, Computer and Computation*, Ankara, pp. 289-292, 2013.

[4] BBC Technology News, "Twitter: Hackers target 250,000 users," http://www.bbc.co.uk/news/technology-21304049, Last Visited, 2013.

[5] Boneh D., Gentry C., and Waters B., "Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys," *in Proceedings of Advances in Cryptology-CRYPTO*, California, pp. 258-275, 2005.

[6] Delerablée C., "Identity-Based Broadcast Encryption with Constant Size Ciphertexts and Private Keys," *in Proceedings of Advances in Cryptology-ASIACRYPT*, Kuching, pp. 200-215, 2007.

[7] Douglas N., "Facebook Employees Know What Profiles you Look at," GAWKER, http://gawker.com/315901/facebook-employees-know-what-profiles-you-look-at, Last Visited, 2007.

[8] Elgamal T., "A Public Key Cryptosystem and A Signature Scheme Based on Discrete Logarithms," *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469-472, 1985.

[9] Gan D. and Jenkins L., "Social Networking Privacy-Who's Stalking You?," *Future Internet*, vol. 7, no. 1, pp. 67-93, 2015.

[10] Gentry C. and Waters B., "Adaptive Security in Broadcast Encryption Systems," *Advances in Cryptology-EUROCRYPT*, Cologne, pp. 171-188, 2009.

[11] Guha S., Tang K., and Francis P., "NOYB: Privacy in Online Social Networks," *in Proceedings of the 1st workshop on Online Social Networks, ACM*, Seattle, pp. 49-54, 2008.

[12] Hajli N. and Lin X., "Exploring the Security of Information Sharing on Social Networking Sites: The Role of Perceived Control of Information," *Journal of Business Ethics*, vol. 133, no. 1, pp. 111-123, 2016.

[13] Jagatic T., Johnson N., Jakobsson M., and Menczer F., "Social Phishing," *Communications of the ACM*, vol. 50, no. 10, pp. 94-100, 2007.

[14] Jansen W. and Grance T., "SP 800-144. Guidelines on Security and Privacy in Public Cloud Computing," Technical Report, National Institute of Standards and Technology, 2011.

[15] Kleinman A., "Facebook Bug Exposed Email Addresses, Phone Numbers of 6 Million Users," http://www.huffingtonpost.com/2013/06/21/facebook-bug_n_3480739.html, Last Visited, 2013.

[16] Lucas M. and Borisov N., "Flybynight: Mitigating the Privacy Risks of Social Networking," *in Proceedings of the 7th ACM workshop on Privacy in the Electronic Society*, Virginia, pp. 1-8, 2008.

[17] Luo W., Xie Q., and Hengartner U., "FaceCloak: An Architecture for User Privacy on Social Networking Sites," *in Proceedings of the 12th International Conference on Computational Science and Engineering*, Vancouver, pp. 26-33, 2009.

[18] Muhammad Y., Kaiiali M., Habbal A., Wazan A., and Ilyasu A., "A Secure Data Outsourcing Scheme Based on Asmuth-Bloom Secret Sharing," *Enterprise Information Systems*, vol. 10, no. 9, pp. 1001-1023, 2016.

[19] Narendula R., Papaioannou T., and Aberer K., "My3: A Highly-Available P2P-Based Online Social Network," *in Proceedings of IEEE*

*International Conference on Peer-to-Peer Computing*, Kyoto, pp. 166-167, 2011.
[20] Sakai R. and Furukawa J., "Identity-Based Broadcast Encryption," IACR Cryptology ePrint Archive, pp. 217, 2007.
[21] Sakthivel A., "Enhancing Cloud Security Based On Group Signature," *The International Arab Journal of Information Technology*, vol. 14, no. 6, pp. 923-929, 2017.
[22] The Statistics Portal, "Most famous social network sites worldwide," https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users, Last Visited, 2017.
[23] Tsukayama H., "Facebook privacy: Users should check these settings as new changes roll out," http://wapo.st/1gwtcqs, Last Visited, 2013.
[24] Vixie P., Thomson S., Rekhter Y., and Bound J., "Dynamic Updates in the Domain Name System (DNS UPDATE)," RFC 2136, April 1997, http://www.rfc-editor.org/info/rfc2136, Last Visited, 2017.
[25] Yuvaraj M., "Cloud Computing Software and Solutions for Libraries: A Comparative Study," *Journal of Electronic Resources in Medical Libraries*, vol. 12, no. 1, pp. 25-41, 2015.
[26] Zafar K., kWS-Android Web Server, https://kamranzafar.org, Last Visited, 2017.

**Mustafa Kaiiali** completed his B.E. degree in Computer Science at Aleppo University, Syria in 2003. Then he obtained his M.Tech and Ph.D. degrees from the Department of Computer and Information Sciences (DCIS), University of Hyderabad, India in 2008 and 2012, respectively. His areas of expertise are: Cloud Computing, Information Security, and Networking. He also passed the test of Cisco Certified Network Professional in Security in 2012. He has several publications in well-reputed journals and international conferences. Currently, he is with the Centre for Secure Information Technologies (CSIT), ECIT, Queen's University Belfast (QUB), United Kingdom as a Research Fellow in Cloud Security. Recently, he has been elevated by IEEE as a Senior Member.

**Auwal Iliyasu** is working as lecturer II in the Department of Computer Engineering, Kano State Polytechnic, Kano, Nigeria since 2010. He obtained his bachelor degree in Computer En-gineeering at Bayero University Kano Nigeria. After his under-graduate studies, he had his postgraduate studies (M.Sc) at Mevlana University, Konya, Turkey in the department of Com-puter Engineering. His research interest are Networking, Infor-mation Security, Cloud Computing and Internet of thing (IoT).

**Ahmad Wazan** is an Assist. Prof. at Paul Sabatier University, Toulouse, France. His research topics include trust management, PKIs, Acess Control and recently security re-quirement engineering issues. His research group has pro-posed the extension of the X.509 trust model by adding a new entity called, Trust Broker. The proposition is now included in the 2016 edition of X.509.

**Adib Habbal** (SM'15) is the head of InterNetWorks Research Platform at the School of Computing, Universiti Utara Malaysia (UUM). He also serves as Executive Council Member of Internet Society Malaysia Chapter. Dr. Habbal received his Ph.D. degree in Computer Science from UUM and he has more than ten years of experience in teaching and university lecturing. Dr. Habbal is the Internet Society (ISOC) Fellow alumni to the Inter-net Engineering Task Force (IETF). In 2013, he was selected as Asia-Pacific Advanced Network (APAN) Fellow to the APAN 35th and Techs in Paradise conference (TIP2013) held at the Uni-versity of Hawaii. In addition to being a speaker at a number of renowned research conferences and technical meetings, he also participates in various international fora such as IEEE meetings, ACM SIGCOMM meeting, the IETF, Internet2 Meeting, APNIC, and APAN. Dr. Habbal's current area of research focuses on Future Internet and 5G mobile networks.

**Yusuf Muhammad** is working as an assistant lecturer in the department of Computer Science, SaadatuRimi College of Education, Kano, Nigeria since 2014. He has completed his undergraduate studies (B.Eng Computer) at Bayero University, Kano, Nigeria. After his bachelor degree he had his postgradu-ate studies (M.Sc. Computer) at the department of Computer Engineering, Mevlana University, Konya, Turkey. His area of expertise are: Cloud Computing, Networking, Information Secu-rity and Database Systems. His current research work focuses on Cloud Computing. He has publications in IEEE proceedings and Enterprise Information Systems.