Addis Ababa University
Addis Ababa Institute of Technology
School of Electrical and Computer Engineering

Data-Based Access Control for Social Networks

A thesis submitted to the School of Electrical and Computer Engineering in partial fulfillment of the requirements for the Degree of Master of Science in Computer Engineering

By Frehiwot Alemu Tiruneh

September, 2015

Addis Ababa University
Addis Ababa Institute of Technology
School of Electrical and Computer Engineering

Data-Based Access Control for Social Networks

By
Frehiwot Alemu Tiruneh

Advisor: Abraham Lamesgin

Declaration

I,the undersigned, certify that research work titled " Data-Based Access Control for Social Networks " is my own work. The work has not been presented elsewhere for assessment. Where material has been used from other sources, it has been properly acknowledged.

Frehiwot Alemu Tiruneh signature ——————

Date of submission: September, 2015

place: Addis Ababa

This thesis has been submitted for examination with my approval as a university advisor.

Advisor: Abraham Lamesgin signature ——————————

Addis Ababa University
Addis Ababa Institute of Technology
School of Electrical and Computer Engineering

Data-Based Access Control for Social Networks

By

Frehiwot Alemu Tiruneh

Approval By Board of Examiners

Dr. Yalemzewd Negash	_____
Dean, School of Electrical and Computer Engineering	signature
Abraham Lamesgin	_____
Advisor	signature
_____	_____
Internal Examiner	signature
_____	_____
External Examiner	signature

Abstract

Social networking has become a popular way to be in contact with each other. In social networking, people tend to share a wide range of information with other users of the networking site. Here, security of personal information has become a most critical issue. One of the important issues in online social network is that how user privacy is protected because online social network providers have full control over users' data. The online social network providers typically store users' information permanently. Meanwhile, the trend in information security is moving the security perimeter as close to the data as possible. We want to move the perimeter closer to the data, but do this without being able to derive who is accessing which data. An efficient privacy protection mechanism is important for online social networking sites that can be used to protect the privacy of online users' data from third parties. An access control mechanism shifts the control over data sharing back to the users by providing them with flexible and dynamic access policies. Hence, instead of relying on credentials given by a person trying to access information, there is a need to protect the data using only the data itself. In this context where decryption of data is made possible by already knowing some part of the data. This thesis work discusses the implementation of data based access control in social networking sites. That is, personal information is made available only to those who already have some of this information. We defined and analyzed types of data based access control methods (direct, indirect and order-invariant data based access control methods). An effort is made to design suitable policy

in order to apply them to social networking sites. We implemented our solution in a prototype platform for social networking sites using a Java based prototype and My Structured Query Language (MySQL) database. Our experimental results verify the effectiveness of indirect data based access control method over social networking sites. This mechanism provides enhanced security features from both eavesdrop attacks and provider attacks. Moreover, we present a performance study of the implemented prototype.

Key words: Data Based Accessed Control, Social Network Security, Access Control Policies.

Acknowledgment

First and foremost I would like to thank the Almighty God for giving me the strength through long and challenging process of the thesis.

I would like to express my sincere gratitude to my Advisor Mr. Abraham Lamesgin for his help, the overall support, guidance, suggestions and encouragement throughout the thesis work.

Furthermore, I would like to thank my colleagues, Helen Arefaynea and Dinkisa Aga, who were around me by giving ideas, encouraging and commenting on the work throughout the thesis.

At the end I would like to express my very profound gratitude to my beloved husband Muluken who spent sleepless nights with and was always my support in the moments when there was no one to answer my queries.

Contents

Abstract i
Acknowledgment iii
Contents iv
List of Acronyms v
List of Figures viii
List of Tables ix

1 Introduction **1**
1.1 Background Information 1
1.2 Statement of Problem 3
1.3 Objectives 3
1.4 Methodology 4
1.5 Outline of the Thesis 5

2 Literature Review **6**
2.1 Literature Survey 6
2.2 Cryptography Security 11
2.3 Secret Sharing Scheme 13
2.3.1 Share Distribution 14
2.3.2 Key Reconstruction 15
2.4 Data-based Access Control 15
2.5 Hash Function 18
2.5.1 Types of Hash Function 19

2.6 Cryptography Encryption 20
2.7 Symmetric Key Cryptography 21

3 Design of the Work **25**

3.1 Introduction 25
3.2 Modeling 25
3.3 Profile Structure 27
3.4 Algorithm Selection 28
3.4.1 Hash Algorithm 28
3.4.2 Secret Key Algorithm 29
3.5 Examples using Indirect Access Control Method 29

4 Implementation **32**

5 Results and Discussion **41**

5.1 Introduction 41
5.2 Evaluation 42
5.2.1 Confidentiality 42
5.2.2 Scalability 43
5.2.3 Flexibility 44

6 Conclusions and Recommendations **46**

6.1 Conclusion 46
6.2 Recommendation 48
References 50
Appendix 53

List of Acronyms

3DES	Triple Data Encryption Standard
AES	Advanced Encryption Standard
CPU	Central Processing Unit
DAC	Discretionary Access Control
DBAC	Data Based Access Control
DES	Data Encryption Standard
GUI	Graphical User Interface
IDE	Integrated Development Environment
IP	Internet Protocol
JDK	Java Development Kit
MAC	Mandatory Access Control
MySQL	My Structured Query Language
NIST	National Institute of Standard and Technology
NOYB	None of Your Business
OSN	Online Social Network
PDAC	Personal Data Access Control
RBAC	Role-Based Access Control
RDBMS	Relational Data Base Management System
SDBM	Sentential Database Manager
SQL	Structured Query Language
TCP	Transport Control Protocol

URL	Uniform Resource Locator

List of Figures

2.1 Symmetric key cryptography 22

3.1 System Model 26

4.1 Login Interface 34

4.2 Profile structure for new user 35

4.3 Steps to search a person 38

4.4 Searching Bob's basic information 39

4.5 Entering first secret key 39

4.6 Bob's basic information 40

5.1 Scalability of prototype 43

5.2 Performance result for different number of key of prototype 44

6.1 Snapshot of profiles in the database 53

List of Tables

4.1 User data items and the corresponding hash value in basic information category 36

5.1 Response time to search a person in three different access control methods 45

Chapter 1

Introduction

1.1 Background Information

A social networking website is an online platform that allows users to create a public profile and interact with other users on the website. The first social networking site SixDegrees.com was launched in 1997 [1]. Six Degrees allowed users to create profiles, list and message their friends and traverse friend's listings, thus fitting the definition above. Even though there were millions of users, users did not have many direct friends and Six Degrees did not offer much functionality besides messaging. The website finally shut down in 2000 [1]. During and after this period other websites started adding Online Social Networks (OSN) features to their existing content, essentially becoming OSNs, with various degrees of success. In the years that followed, new OSNs started from scratch and began to offer functionality beyond simply listing and browsing friends.

Facebook, Twitter, MySpace and LinkedIn are among most popular social networking sites according to traffic rank. The main goal of online social network site is to make available an information space, where each social network participant

can publish and share information as well as meet other people for a variety of purposes.

When registering to a social network, the user will be given an account and a profile, where he or she is able to modify personal information, specify relationships with other users and even manage personal resourses, such as comments and photos. So there is a need to protect sensitive information from unauthorized people. The traditional access control implemented in social networks is based on the relationship between each user. A member from a social network can decide which personal information is accessible by marking a given item as public or private. Although this approach has the advantage of being easy to implement, lacks flexibility. The relationship settings do not allow users to specify their own access control requirements, which might lead to a too loose or too restrictive access control policy [2]. Moreover, social networks store information remotely, rather than on a user's personal computer. This means the privacy policy may change at any time without a user's permission. Due to this content that was posted with restrictive privacy settings may become visible.

Another issue of traditional access control is the confidentiality of users' information. Access to profiles can be restricted by access control policies, so only permitted users will be able to see those profile data. However, the server has full access to all personal information. This means that if the security of the server is compromised, so is the confidentiality of personal information.

Some security issues of the social networking site, Facebook, have been discovered. Facebook had allowed users to deactivate their accounts, but had not actually allowed them to remove the account content from its servers. A Facebook representative explained to a student from the University of British Columbia that users had to clear their own accounts by manually deleting all of the content including wall posts, friends, and groups. A New York Times article noted the issue and

also raised concern that emails and other private user data remain indefinitely on Facebook's servers [3].

1.2 Statement of Problem

Sharing personal information on social networking sites can prove to be a privacy risk, however, it is precisely this functionality that defines social networks. Generally there are two main classes of privacy threats: those that involve disclosure to other "users" (registered or not), and those that originate from the online social network service provider's side. The main difference between these parties is the type of information they can access. A user or outsider can generally only view public information. The OSN provider can usually view all data in the system; including private uploads, browsing behavior, etc. Trust plays a big role in the relationship between a user and the service provider.

Therefore, it is important to improve the access control mechanisms in order to provide better security to social network users. Hence, instead of relying on credentials given by a person trying to access information, there is a need to protect the data using only the data itself, the approach is called data-based access control [4], where decryption of data is made possible by already knowing some part of the data. In this research work, it is intended to use data based access control for social networking sites.

1.3 Objectives

General Objective

The general objective of this thesis work is to design and implement data-based access control for social networking sites.

Specific Objectives The specific objectives are to:

- Study the concept of data-based access control.
- Selecting proper hash function based on literature.
- Comparison of different encryption methods and choose one which is more appropriate.
- Design and implement the data-based access control prototype on social networking site.
- Interpretation of the output which is gained from the prototype.

1.4 Methodology

- **Literature survey**: A thorough study on literature regarding security, social network sites, data-based access control paradigm and other resources related to the topic were conducted.
- **Modeling**: Based on the existing literature an interface of the social network sites and data-based access control paradigm for social network sites was modeled.
- **Implementation**: A data-based access control paradigm on the interface using Java Development Kit (JDK) language and MySQL Workbench was implemented.
- **Interpretation and conclusion**: Based on the output obtained from the implementation results were interpreted and conclusion was drawn based on the results obtained.

1.5 Outline of the Thesis

The rest of the thesis is organized in to six chapters including this section. In Chapter 2 relevant literature was reviewed. Moreover, briefly presents the current situation of access control in information systems and introduces the concept of data-based access control. We confer and present the proposed model of DBAC in social networks in Chapter 3. In addition, based on literature, some comparisons of algorithms are discussed. We elaborate the implementation of the proposed model in chapter 4. Chapter 5 deals with performance evaluation tests and result analysis. Finally, in Chapter 6, we offer conclusions and suggest future work that can consolidate this work.

Chapter 2

Literature Review

2.1 Literature Survey

Most of the research done on securing social network sites has mainly focused on safeguarding the relationship between users. We review some of the works related to our study.

Several studies conducted in the past few years have identified the need for solutions to address the problem of information leakage in social networks. The Discretionary Access Control (DAC) [5] has been one of the cornerstones of computer systems protection since 1970's. In DAC, the owners of data objects have the exclusive privilege of deciding who gets what type of access to their data objects. One of the major issues with this "owner-centric" DAC is the propagation of access rights. The literature [6] on DAC has many approaches for addressing the access propagation problem using ideas such as copy flags and grant options.

The Mandatory Access Control (MAC) [7] is another protection scheme proposed first in the 1970's that aims to enforce lattice based information flow constraints to create high assurance information systems with confidentiality, integrity, and

aggregation concerns. A lattice is used to define the levels of security that an object may have and that a subject may have access to. The subject is only allowed to access an object if the security level of the subject is greater than or equal to that of the object. Object owners do not make access decisions. Access is granted only if a subject has the necessary clearance in order to access an object. Security levels are hierarchically ordered and higher security levels are said to dominate lower security levels. A subject gains access if its security clearance dominates the security level of the object to be accessed. In practice, covert channels remain a major impediment for implementing MAC. As a result, MAC-based schemes have found limited applicability in highly legislated environments such as the military.

Role-Based Access Control (RBAC) [8] has in the past arisen as the most widely discussed alternative to DAC and MAC. The basic idea of role-based access control is to create well defined abstractions called roles and assign permissions. It provides easier management of access rights by breaking down the task of user authorizations into two parts: one part involves assigning access rights to a role and the other part involves assigning roles to users. This allows for simple assigning and revocation of access rights. RBAC also lets for a hierarchy of roles, making it an ideal fit for many applications with roles that can be highly specialized. However, context in the activation, deactivation, and management of roles are not fully considered. Also, in collaborative environments, a user in an instance of a role might need a specific permission on only one instance of an object and RBAC does not have the ability to specify control that is fine-grained enough to deal with this case.

In [9], Carminati et al. present an access control model for web-based social networks, where policies are expressed ;as constraints on the type, depth, and trust level of existing relationships. The model uses a graph representation of the social network with nodes and the edges of the graph representing, respectively, the social network users and the relationships that exist between them. For the

relation-type the authors propose to extend the graph representation to support relationship types, direct or indirect. The depth of a relationship between two users corresponds to the shortest path among all possible paths with direct or indirect relations of the given type. The trust level between two users a and b for a given relationship type is defined as how trustworthy user a considers user b with respect to the given relationship. Finally, a relationship is represented by a tuple, where its components denote the users participating in it, along with the type, depth, and trust level of the relationship itself.

In their proposed model, each owner defines the access rules according to his or her preferences, while each requester is in charge of verifying to the owner whether he or she is authorized to access a given object. When a node requests access to a resource, the owner releases the access rules to the requesting node. In order to access the resource, the requesting node must provide the owner with a proof showing that it actually satisfies the policies. Proofs are generated using a reasoner, which allows the owner to locally verify whether the resource should be released or not. To prevent the requesting node from maliciously falsifying the proof, the authors propose the use of certificates. Whenever two users establish a new relation they create and sign a certificate stating there exist a direct relation with a certain trust level. Because generating proofs can become incredibly complex, especially when proving assertions regarding trust, special Central Nodes are introduced to manage the certificates and to provide reasoner functionality for the user generated requests.

Villegas et al. [10] proposed an access control scheme for protecting personal data, such as data posted on the social network (e.g., profile information such as address, date-of-birth, etc.) or personal data stored on online storage systems. In either case, the objective of the access control scheme is to use trust measures derived from the social network to control the data sharing activities. The authors tried to compute a "trusted distance" measure between users that is partly based on hop

distance on the social network and an affine distance derived from empirical data. The confidentiality specifications divide the people based on the trusted distance, the social network is divided into three zones: accept, attest, and deny. Requests from users in the accept zone (closest to the data origin) are accepted unconditionally while the requests from the deny zone (furthest from the data origin) are rejected outright. Requests from the attest zone need additional authorization to get access. Based on which zone a user is mapped into, her/his requests to access the data objects of another user be accepted, further processed, or rejected by Personal Data Access Control (PDAC). The requests further processed by the PDAC undergo another round of evaluation by a set of attesters designated by the owner of the data object. Finally they simulate using data from the myspace.com social network.

Some other research work has relied on cryptography techniques to protect user's private information. NOYB [11] encrypts personal information using a pseudo-random substitution which replaces personal information with a pseudo-randomly selected substitution taken from a public dictionary. NOYB assumes a shared secret key that is known to the social network user circle of friends and friends of friends. The authors argue that encrypted messages on a profile are easy to spot by the social network provider. Their approach uses substitution of "information atoms" (e.g. age, or name-and-gender) to hide information. NOYB substitutes the atom with a pseudo randomly chosen atom of another user. For example, suppose that (Alice, F, 20) are name, sex, and age of one user which are partitioned into two atoms (Alice, F) and (20). Each atom will be replaced by other atoms say (Bob, M), and (28). One public dictionary holds the atoms of the same type. Entries of dictionary are atom's index, atom's content, and atom's frequency. To prevent polluting dictionaries by adversaries, authors suggested keeping dictionaries by a group of trusted users who accept updates only from authorized members. While online service receives and displays fake data, authorized users receive a key and a nonce implemented in an out of band communication channel (does not exchange

keys via online service channel) to retrieve original atom from fake atom. The number of channels that are used for this scheme is high. Also outside users have no way to distinguish between users that are hiding their information and users that are not. This makes profiles meaningless to such users, and could lead to cases of mistaken identity.

Matthew Lucas et al. [12] propose to encrypt certain parts of user's profile using public key cryptography. Keys are derived from user's supplied passwords in order to maintain access flexibility. Their application is social network site dependent and relies on the social network site own servers for key management. A user's private key is encrypted with a password and also stored on the server. They present a Facebook application to protect private data by storing it in Facebook in encrypted form. The decryption algorithm is implemented in JavaScript and retrieved from Facebook. It does not provide complete privacy and is vulnerable to active attacks by the provider. Their scheme requires the social network user to select one by one each user's to which a message should be encrypted. However the access control is managed by server.It depends on trustworthy of both servers and OSN. Therefore a potential problem with this approach is the resulting low entropy of the keys.

As the literature survey reveals, there is numerous research being conducted to ensure data confidentiality and to increase the flexibility of the access control, with more emphasis on the flexibility. Since OSNs contain great amounts of useful and interesting data for a number of users, they form an attractive target for attackers.

Most works strained to address the issue of user related privacy threats using different access control approaches. The provider designs and configures the systems underlying the OSN. It has full access to any user related data, including message logs. Such setting can enable employers of the provider to access even tamper with users' information.

In this research, we tried to protect the social network sites both from the user and OSN provider side by protecting the data using the data itself. Accordingly, key management is carried out by the user. To our knowledge, provider related privacy threat has not been mentioned in existing literature.

2.2 Cryptography Security

Due to the recent development of computers and computer networks, huge amount of digital data can easily be transmitted or stored. However, transmitted data in networks or stored data in computers may easily be eavesdropped in and exploited or substituted by malicious users if unless protective measures are in place. Encryption is a way of securely storing or transmitting data.

There are two types of cryptographic security: computational security, and information-theoretic security. Computational security relies on the assumed difficulty of computational problems, e.g., the problems of integer factorization, computing discrete logarithms, etc. The RSA cryptosystem [13] proposed by Rivest et al. in 1978 typical is one cryptographic tool that rely on computational security. We note that the success of computational security relies on certain assumptions about the attacker's computational resources, and hence the security level of an existing cryptographic system with computational security will decrease as computer technology improves. Moreover, while it is assumed that the underlying computational problems used in some practical cryptographic systems cannot be solved in polynomial time, this remains unproven. There might be as-yet unknown but highly efficient algorithms that can break some cryptographic systems in polynomial time. Therefore, computational security faces the risk of being broken by progress in the devising of algorithms or by progress in computer technology.

In contrast, information-theoretic security relies on the theoretical impossibility of

breaking it, which is derived purely from information theory. Information theory has several applications in cryptography. First, it allows proving unconditional security of cryptographic systems. Second, it allows proving impossibility and lower bound results on the achievability of unconditional security. Third, it is a key tool in reduction proofs showing that breaking a cryptographic system is as hard as breaking an underlying cryptographic primitive. (e.g. a one–way function or pseudo-random function).

Since information-theoretic security does not rely on any unproven assumptions about the difficulty of computational problems, it can hold even if an attacker has unlimited computational resources. Thus, the security level of a system with information-theoretic security is independent of the progress of computer technology. One of the most important problems in cryptography is the transmission of a secret message between two legitimate users (the sender Alice and the receiver Bob) over an insecure communication channel such that an enemy (Eve) with access to the channel is unable to get useful information about the message being sent.

Information-theoretic security gives us the strongest definition of security [14]. For instance, Eve has perfect access to the insecure channel, i.e. she receives an exact copy of the cryptogram C, where C is obtained by Alice as a function of the plaintext M and a secret key K, shared by Alice and Bob. i.e. Eve gains no knowledge about M by knowing C. Notice that in this definition of a secure cipher system, no assumption about eavesdrop computational power is made therefore makes the information-theoretic security more desirable in cryptography than computational security. Moreover, since cryptographic tools with information-theoretic security are not vulnerable to future developments in quantum computing, they are attracting significant attention as a kind of post-quantum cryptography. For these reasons, information-theoretic security is sometimes called unconditional security. Secret sharing schemes are well known cryptographic tools that have information-

theoretic security [15].

2.3 Secret Sharing Scheme

In the case of secure data storage, there are threats such as troubles of storage devices or attacks of destruction. In order to prevent such attacks, we must make as many copies of the secret as possible. But, if we have many copies of the secret, the secret tends to leak out, and hence, the number of the copies should be as small as possible. This contradictive requirement can be solved by a secret sharing scheme, which was proposed independently by Shamir [16] and Blakley [17] in 1979.

A secret sharing scheme is an important tool for distributed file systems protected against data leakage and destruction and for secure key management systems. The basic idea of secret sharing is that a dealer distributes a piece of information (called a share) about the secret to each participant such that qualified subsets of participants can recover the secret, but unqualified subsets of participants cannot obtain any information about the secret. If Alice and Bob want to exchange some information concealed from Eve, the challenge is to make sure that Eve is not able to understand that information, even if Eve can see the bits that are being transferred over the network. To sum up, the idea behind the scheme is sharing confidential information between n persons.

The scheme encrypts the secret information S into n pieces called shares where each share contains know information of secret S, but S can be decrypted by collecting several shares. This means that the secret sharing scheme is secure against both destruction and theft. We also note the scheme is unconditionally secure because the secret sharing scheme is not based on any assumption of computational difficulties like the factorization of integers or the calculation of discrete

logarithms. Hence, it is appropriate for long time data storage. Furthermore, a secret sharing scheme is expected to be used in the environment of ubiquitous networks to share secrets among many entities.

Shamir's threshold scheme, which is also called (k, n) threshold scheme, is based on polynomial interpolation ("Lagrange interpolation") to allow any k out of n participants and to recover the secret [18]. An informal definition of a threshold scheme is as stated below.

Definition 2.1:- Let k and n be positive integers $k \leq n$. A (k, n)-threshold scheme is a method of sharing a secret K among a set of n participants in such a way that any k participants can compute the value of the secret, but no group of $k-1$ or fewer can do so.

To give more explanation on the construction of the (k, n) threshold scheme work, let the set of participants be denoted by $P = \{P_1, P_2, \ldots, P_n\}$. The value of the secret K is chosen by the dealer, denoted D, who is a special participant not in P. When D wants to share the Secret K among the participants in P, D gives each participant some partial information, called a share. Sharing the secret,S, are distributed secretly, so no participant knows any other participant's share.

2.3.1 Share Distribution

First, D wants to share the secret K so that for this purpose D randomly chooses $k-1$ elements, denoted by the coefficient $a_0, a_1, a_2, ..., a_{k-1}$ where a_0 is the secret K. Secondly D computes $y_i = a(x_i)$,for $1 \leq$ i $\leq n$, where $a(x_i)$ can be constructed using Lagrange interpolation. Then D gives the share y_i to participant P_i. In short, the dealer constructs a random polynomial of degree at most $k-1$ in which the constant term is the secret K, i.e., $a_0 =$ K. Every participant obtains a point (x_i, y_i) from the polynomial.

2.3.2 Key Reconstruction

Any k participants can reconstruct the polynomial $a(x)$ and, hence, calculate the secret. But any group of $k-1$ participants cannot do so. This is basically done by means of polynomial interpolation. The function y_i is equal to the function $a(x_i)$, asserting that polynomial a(x) have at most $k-1$ degrees and, hence, can be written as

$$a(x) = a_0 + a_1x^1 + a_2x^2 \ldots + a_{k-1}x^{k-1} \tag{2.1}$$

where the coefficients $a_0, a_1, ..., a_{k-1}$ are unknown elements and $a_0 = K$ is the secret. Each participant knowing $y_i = a(x_i)$ can obtain linear equation in the k unknowns $a_1, a_2, \ldots, a_{k-1}$. So the group P has k linear equations. If the equations are all linearly independent, there will be a unique solution, and a_0 will be revealed as the key.

As we see from the above construction of secret sharing method coefficients are chosen randomly, except for the coefficient that represents the secret. The possibility of limiting the degree of the polynomial to k can makes this scheme transferable to other applications.

2.4 Data-based Access Control

When applied to data, Shamir's Secret Sharing method will have n data-items to fit into the polynomial. It is not possible to represent these as n points on a polynomial of degree $k-1$, because in general n points will not lie on a single polynomial of this degree. As a result Pieter amended Shamir's approach to introduce (k, n) threshold data-based access control paradigm [4]. He modifies the technique in to three schemes namely direct, in-order invariant and in-direct

scheme.

1. Direct data-based access control scheme

   In this scheme the data items can be recovered without any intermediate secret value that protects the data. The data item should first be changed into numbers using two way injective hash functions. Two way injective hash function is function that convert data in to numerals and also vice versa. If we have n number of data items (string) following conversion to numerals,can allow definition of a polynomial function with degree $n-1$. Furthermore, additional $n-k$ points should be published, to let someone who knows k of the data-items reconstruct the polynomial.

   Decryption (reconstruction of polynomial function from n points) can now be done by reconstructing the polynomial from the n–k public points plus k known points. Then recalculating the remaining points using Lagrange interpolation becomes trivial. The main drawback of this access control method is converting the encoded points back into data items. Reconstructing the data-items using direct data-based access control method, like Shamir's Scheme, needs additional key and modification. To overcome the downside of this method Pieter proposed another data-based access control technique called indirect data-based access control scheme.

2. Indirect data-based access control scheme

   In this scheme knowledge of enough data-items allows for the recovery of conventional encryption algorithm key, that can be used to decrypt the full data-set. In addition to data based access control it relies on standard encryption techniques. This means, $n+1$ points are necessary and sufficient to define and reconstruct a polynomial of degree n. Similar to the direct data based access control scheme data items should be converted in to numerals via a hash function. Then a polynomial f of degree n is defined using $n+1$ points, n data items and the secret key S. S is the key which is

obtained from the standard encryption algorithm. In order to recover the full polynomial including the key S one needs $n+1$–k additional points of the polynomial. Since the $(n+1-k)$ points on the polynomial made public represent additional points on the polynomial, these will allow anyone who knows k of the data-items to recover f from the $(n+1-k)+k = n+1$ points in his possession. S can be recovered from the point (0,S) and be used to decrypt all the data.

3. Order invariant data-based access control scheme

   For obtaining an order-invariant scheme, we apply the set difference construction of a secure sketch from [19]. For the direct scheme, assume again that we have an injective two-way function h mapping data-items (k number of data items) to numbers. Here, the secret values $h(D_i)$ are encoded as the x-coordinates for which the value of a polynomial f coincides with the value of a polynomial g, is a secret random polynomial function with degree $k-1$. This means function f is created from points $h(D_i)$ and $g(h(D_i))$. Since the values are encoded as x-coordinates as opposed to the y-coordinates in the previous scheme, order of the data items is unimportant.

   In [19], Reed-Solomon error correction is used to account for possibly incorrect set elements that are supplied in the decoding stage. Because we are not interested in error correction here, we can leave out this possibility. For decryption, one should input k values which are all correct.

   The encryption algorithm now takes two steps: The first step is selecting a secret polynomial g of degree $k-1$ at random and then compute and publish the unique monic polynomial (is a universal polynomial in which the leading coefficient, the nonzero coefficient of highest degree, is equal to 1) f of degree n by solving $f(h(D_i)) = g(h(D_i))$ for all D_i. Publish k as well. The polynomial function g could be resolved by solving $f(h(D_i)) = g(h(D_i))$ for all known items. Then the remaining values for which $f(h(x)) = g(h(x))$, can be calculated. In other words, given $k-1$ values for which $f(x) = g(x)$,

one learns nothing about the remaining root of $f(x) - g(x)$.

The order of the data item is the only difference between the direct and order-invariant direct method. In the latter the data items hash value will be assigned as x-coordinate so that items do not have any order.

2.5 Hash Function

A hash function is a function that takes a relatively arbitrary amount of input and produces an output of fixed size[20]. This process can be denoted as:

$$h = H(M) \tag{2.2}$$

where M is the input message and h is the hash, often called hash, hash value or message digest, generated by the hash algorithm H. Normally, the size of the hash h is fixed by the algorithm and most of the time a hash functions will produce unique output for a given input. The properties of some hash functions can greatly increase the security of a system network; when implemented correctly they can verify the integrity (means the property whereby data has not been altered in an unauthorized manner since the time it was created, transmitted, or stored by an authorized source) and source of a file, network packet, or any arbitrary data [21]. The main reason for creating a hash value of a message is that any accidental or intentional change to it will result in a completely different hash value. By comparing the hash values of a message before and after an event, such as downloading it, the integrity of the data can be validated. Generally, the hash of a message can be used to guarantee the integrity and authentication of a message and to "uniquely" represent the message.

Hash functions have a lot of features some are listed below.

- A hash function should be impossible for two different messages to ever produce the same message digest. Changing a single digit in one message will produce an entirely different message digest.
- It should be impossible to produce a message that has some desired or predefined output (target message digest).
- It should be impossible to reverse the results of a hash function. This is possible because a message digest could have been produced by an almost infinite number of messages.
- The hash algorithm itself does not need to be kept secret. It is made available to the public. Its security comes from its ability to produce one-way hashes.
- The resulting message digest is a fixed size. A hash of a short message will produce the same size digest as a hash of a full set of encyclopedias.

In most applications, hash functions have two properties. First, hash functions should be fast to compute. Secondly, it is often useful for hash values to be distributed uniformly, in the sense that every possible hash value is used about the same number of times, even when the original set of inputs contains groups of similar strings.

Even though there is no collision free hash function, today there are several hash functions currently in use. By examining the history and security available in each function, a user can determine which algorithm is best suited for their application.

2.5.1 Types of Hash Function

Various hash algorithms are employed for different purposes. One purpose of hash algorithm is converting strings to numerals.

1. djb2

   The algorithm was first reported by Professor Daniel J. Bernstein. The djb2 hash function simply starts with a seed, multiplies it by the magic number 33 (why it works better than many other constants, prime or not has never been adequately explained) and adds the current character to create a new hash value then repeats this for every character in the string [22]. The algorithm works shown in Appendix B.1.

2. SDBM

   This is the algorithm of choice which is used in the open source Sentential Database Manager (SDBM) project. It was found to do well in scrambling bits. The actual function is hash(i)= hash(i-1)∗r + str[i](Shown in Appendix B.2). The hash function seems to have a good over-all distribution for many different data sets [23].

3. Loselose

   The algorithm is the simplest hash function. Till a character in the string ends up, it simply add a constant number at each iteration. Therefore, to compute the iteration of the hash the algorithm only uses a single operation with very high collision.

2.6 Cryptography Encryption

Cryptography is a powerful tool used to protect information in computer systems by using mathematics to encrypt and decrypt data. It enables secure storage of sensitive information or transmission across insecure networks (like the Internet. Many cryptography algorithms are widely available and are used in information se-

curity. The two main characteristics that identify and differentiate one encryption algorithm from another are its ability to secure the protected data against attacks and its speed and efficiency in doing so. In cryptography original message is basically encoded in some non-readable format. This process is called encryption. The reverse process is called decryption, the only person who knows how to decode the message can get the original information. There are two main categories of cryptography depending on the type of security keys used to encrypt/decrypt the data. Thus on the basis of key used, cipher algorithms are classified as symmetric key algorithms, where the same key is used for encryption and decryption. In asymmetric key algorithm, encryption and decryption is done by two different keys. The two keys are private and public keys. Public key is used for encryption, so that the key is known by the public and private key is used for decryption, which is known only by the user. There is no need for distributing them prior to transmission. Asymmetric encryption techniques are almost 1000 times slower than Symmetric techniques, because they require more computational processing power [24]. Since our system needs the same key for both encryption and decryption we use symmetric key cryptography to take the secret key as one point in indirect data-based access control scheme.

2.7 Symmetric Key Cryptography

Symmetric key encryption is also called as single key cryptography due to its behavior of using same secret key, for both encryption and decryption. Users exchanging data keep this key to themselves. Message encrypted with a secret key can only be decrypted with the same secret key. The algorithm used for symmetric key encryption is called secret-key algorithm. Since secret-key algorithms are mostly used for encrypting the content of the message they are also called content-encryption algorithms.

The symmetric encryption scheme has five ingredients:

- **Plaintext:** This is the original intelligible message or data that is fed to the algorithm as input.
- **Encryption algorithm:** The encryption algorithm performs various substitutions and permutations on the plaintext.
- **Secret Key:** The secret key is also input to the encryption algorithm. The exact substitutions and permutations performed depend on the key used, and the algorithm will produce a different output depending on the specific key being used at the time.
- **Ciphertext:** This is the scrambled message produced as output. It depends on the plaintext and the key. The ciphertext is an apparently random stream of data, as it stands, is unintelligible.
- **Decryption Algorithm:** This is essentially the encryption algorithm run in reverse. It takes the ciphertext and the secret key and produces the original plaintext.

The process of symmetric key cryptography is shown in figure 2.1.

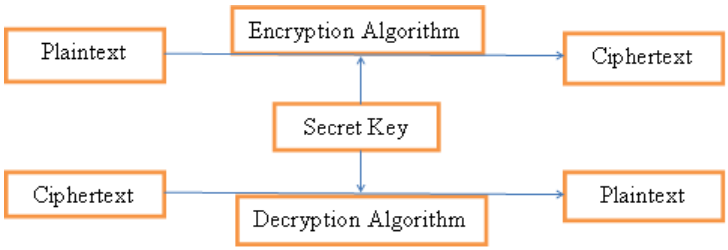


Figure 2.1: Symmetric key cryptography

The key should be distributed before transmission between entities. In this encryption process the receiver and the sender has to agree upon a single secret (shared) key. Given a message (message that can be read and understood without

any special measures called plaintext) and the key, encryption produces unintelligible data called ciphertext, which is about the same length as the plaintext was. Decryption is the reverse of encryption, and uses the same key as encryption. Following are some of popular secret key algorithms:

1. DES (Data Encryption Standard)

   DES was the first encryption standard to be recommended by National Institute of Standards and Technology (NIST). It is based on the IBM proposed algorithm called Lucifer. DES became a standard in 1974. Since that time, many attacks and methods were recorded that exploit weaknesses of DES, which made it an insecure block cipher.

2. 3DES(Triple Data Encryption Standard)

   As an enhancement of DES, the 3DES (Triple DES) encryption standard was proposed. In this standard the encryption method is similar to the one in original DES but applied 3 times to increase the encryption level. But it is a known fact that 3DES is slower than other block cipher methods.

3. AES (Advanced Encryption Standard)

   AES is the new encryption standard recommended by NIST to replace DES. It was originally called Rijndael (pronounced RainDoll). It was selected in 1997 after a competition to select the best encryption standard. AES standard algorithm can only accept a block size of 128 bits and a choice of three keys - 128, 192, 256 bits. Depending on which version is used, the name of the standard is modified to AES-128, AES-192 or AES-256 respectively. Like most encryption algorithms, AES is reversible. This means that almost the same steps are performed to complete both encryption and decryption in reverse order. Even though AES encryption is fast and flexible it needs more processing power. It can be implemented on various platforms especially in small devices. Brute force attack is the only effective attack known against it.

4. Blowfish

   It is one of the most common public domain encryption algorithms provided by Bruce Schneier [25]. Blowfish provides a good encryption rate in software and no effective crypt analysis of it has been found to date. It has a 64-bit block size and a variable key length from 32 up to 448 bits. The algorithm consists of two parts: a key-expansion part and a data- encryption part. Key expansion converts a key of at the most 448 bits into several sub key arrays totaling 4168 bytes. Data encryption occurs via a 16-round Feistel network [26]. All operations are XORs and additions on 32-bit words. The only additional operations are four indexed array data lookups per round. The operations carefully chosen for the algorithm were table lookup, modulus, addition and bitwise exclusive-or to minimize the time necessary to encrypt and decrypt data. Blowfish combined a bitwise exclusive-or process to be achieved on the left 32-bits before being changed by the function or propagated to the right 32-bits for the next round.

Chapter 3

Design of the Work

3.1 Introduction

In the previous chapter of this study, literature surveys and the concept of data based access control scheme have been discussed in detail. Under this chapter, modeling of the prototype, the structure of profile in the system and selection of an algorithm are given more attention Moreover, this section involves the actual development of a prototype.

3.2 Modeling

To protect profile of social network sites using data-based access control method we propose a model as shown below in figure 3.1.

In our system, users can enter their own profile information or request another user's profile information. When a user enters his/her own profile information, it is encrypted on the user's machine and sent to the server for storage. Thus, information which is stored in the server is in encrypted form. The profile infor-

mation which is stored in the server can be requested by any other user. However, that information is in encrypted form and the profile can be decrypted only by entering a sufficient number of profile attributes already known to the user. There attributes are requested by the server. As with encryption, decryption happens locally on the user's machine. The server therefore has no access at all to the details of the profile.To clarify further, let us take the following illustration.

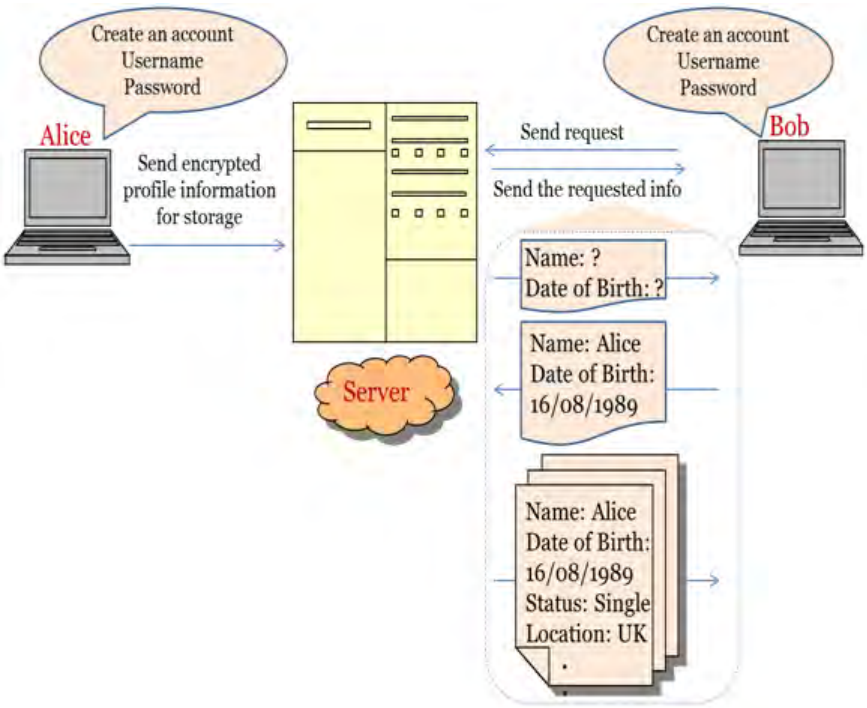


Figure 3.1: System Model

If Alice needs to register on the site, she fills her profile details and submits it to the server, where information is stored in encrypted form. Moreover, she sets which details to query of a user trying to access her profile. If someone needs to access the profile of Alice s/he should know sufficient amount of data which is required by Alice. For instance as you see from figure 3.1 when Bob sends a request to the server to see basic information of Alice the access policy on the server sends requirement questions to Bob. If he answers the questions, he can access all basic information. This is based on data-based access control policy.

Because the server has no access to the individual profile attributes, users can only

modify their own profiles by re-encrypting their entire profile and replacing their old profile with the newly encrypted one. If a user only wants to alter a single profile attribute, this means they have to supply their entire profile information all over again. They can of course request their own profile information from the server, but then, like any other user, they will have to provide enough information to first decrypt their own profile. In order to simplify this, it is possible for a user to store their own profile information in unencrypted form on their machine. This information can then be used as a basis whenever they want to update their profile. Furthermore, over the site the profile should have the same categories, means in social network site the profile structure for all users must be the same.

In order for this system to be usable, however, the server needs two pieces of information. The first is a way to uniquely identify user profiles, such that users can address the profile they want to upload or download. The second is a secret known only to the server and the user who wishes to update their profile. This secret serves to prevent malicious users from modifying or deleting existing profiles, while still allowing for update functionality.

3.3 Profile Structure

On social networking sites personal information is generally divided into different categories in the user's profile. For instance, information such as mobile phone number and email are usually in a category of contact information. Profiles of most social network sites consists different kinds of categories, like basic information, personal preferences, educational information, contact information, etc. In each category, the items of information are described in detail. In this paper we assume that the profile structure contains three categories that are:

- Basic information

- Educational information
- Likes and interest.

Each category has its own data items. Under the first category, Basic information, there are six data items that are name, sex, birthday, current city, home town and looking-for. The other category is educational information which contains information like university, high school, university class year, high school class year and field of study. The third category is likes and interest. This category consist different items, such as activities, interests, music, books and movies.

3.4 Algorithm Selection

3.4.1 Hash Algorithm

To select best hash algorithm for our work we consider three points. The first is minimal hash collisions and the other is excellent distribution with speed. Most interesting requirement was that the hash must be better than its competition. The competent hash algorithm doesn't have to be secure because for our work we need only to change strings in to numerals.

Loselose hash function is the one with high collision rather than SDBM and djb2. But since it uses only one operation (means it is one of additive has function) it has good speed. On the other hand djb2 hash algorithm has excellent distribution and speed than SDBM hash algorithm [27]. For coding up a hash function quickly, djb2 is usually a good candidate as it is easily implemented and has relatively good statistical properties. It has also excellent distribution, relatively minimum collision and speed on many different sets of keys and table sizes [28]. So that because of the above reasons in this paper we use djb2 hash algorithm to convert

strings in to numerals.

3.4.2 Secret Key Algorithm

As stated above, for indirect data-based access control scheme it needs standard encryption method. The generated key from the encryption algorithm will be taken as an additional point for the polynomial in order to be used as another secret for the scheme. Based on literature survey, Anjula Gupta and Navpreet Kaur Walia review and analyzed about cryptography algorithms to promote the performance of the encryption methods also to ensure the security proceedings. They conclude that blowfish has better performance and efficiency than all other ciphers [29]. Even though there is no significant difference between blowfish and DES symmetric encryptions, blowfish has better performance than other encryption algorithms by simulation results which makes it an excellent candidate as a standard encryption algorithm [30][31]. Mandal provided a fair comparison between four most common and used symmetric key algorithms: DES, 3DES, AES and Blowfish. A comparison had been made on the basis of these parameters: block size, key size, encryption/decryption time, CPU process time in the form of throughput and power consumption. These results showed that blowfish is more suitable than the rest [33]. Moreover blowfish has not any known security weak points so far, it can be considered as an excellent standard encryption algorithm [32]. Therefore, based on literature we use blowfish as our encryption method .

3.5 Examples using Indirect Access Control Method

To give an illustration about by what method polynomial function can be constructed and reconstructed and also how to get the secret key, we consider basic information category. As table 4.1 shows the hashed value for the corresponding

data items is done by using djb2 hash function. Values from the input data and the secret key will be: (1, 478), (2, 586), (3, 464), (4, 130), (5, 1064), (6, 765) and (0, 3661648797), where value at zero is found by converting encryption key in to numeral. Depending on user interest number of secret key and number of points which should be published can vary. In our example we select three numbers of attributes to be secret key i.e. Name, Sex and Birth Day.

Then by using Lagrange interpolation, the polynomial function $a(x)$ will be constructed as shown in equation 3.1:

$$a(x) = 5085613.559x^6 - 106797930.1x^5 + 889983114.7x^4 - 3737930431x^3 + 8259048636x^2 - 8971037322x + 3661648797 \quad (3.1)$$

Using the above polynomial function we publish $n+1-k$ points (where n is number of data items and k is number of secret key), that means in our case four points should be published at x =7, 8, 9 and 10. Those points are: (7, 3661629680.791), (8, 25631427357.896), (9, 102525764696.319) and (10, 307577404177).

Since our function is degree six we need seven points to reconstruct the polynomial. Here we have four points which were published before. So we need additional three points i.e. k points. For that reason, a user who can know k of the data items can reconstruct the polynomial function. Since we use Name, Sex and Birth Day as secret, users are expected to enter that information. Immediately the required data items will be hashed. Hence, points used to reconstruct polynomial are: (1,478), (2,586), (3, 464), (7, 3661629680.791), (8, 25631427357.896), (9,102525764696.319) and (10, 307577404177). In order that the new polynomial will be:

$$a(x) = 5085613.509x^6 - 106797928.4x^5 + 889983190.2x^4 - 3737930356x^3 + 8259048532x^2 - 8971037922x + 3661648991 \quad (3.2)$$

To minimize the digits of the number and to get more significant number we divide the secret by 1000. Reconstructing $a(x)$ can reassure getting the first key, k number of data items, and also points at , main secret key. So if the user get secret key, then can decrypt and see the rest of information.

Chapter 4

Implementation

In the previous chapter, we have seen an idea how to secure information using the data itself. Besides, we tried to model the general system and select different algorithms based on literature reviews. In this chapter, we present the implementation of system model which was discussed in previous chapter by first describing the requirement. Moreover this section involves development of prototype.

We use a 64-bit a win 7 platform on a machine with an Intel duo core Central Processing Unit (CPU) running at 1.73 GHz speed, and 1.50 GB RAM.

Our system, figure 3.1, has two parts, client side (front end) and server side (back end). The client side is used to connect the social network to the user via an interface. Therefore we install Netbeans IDE 6.7.1 software consequently we use Java as programming language to do the interface. We select this language because it is platform independent and also it is designed to allow developers to create programs quickly and easily, without having to deal with many of the low-level details of the program. One feature of the Java language that stands out is that it is designed to be a very high level programming language. Java handles memory management automatically using a garbage collector. Moreover it comes with a

great deal of packages and utilities to make development faster and easier. Besides seems to have the best built in support for Graphical User Interface (GUI) programming. We thought that Java was very well suited to GUI programming.

At the back end, the server is used for storage so that we can retrieve and store data. For this reason we need data base management system. Hence we use MySQL, server as a database to store users profile information. MySQL database server, which manages databases and tables, controls user access, processes Structured Query Language (SQL) queries. It is protected by password. All data items of user profile will be saved in the database table name called "PROFILE". Therefore each category has its own protection keys; a data in which it is selected by the owner of the profile is a key.

As stated above in chapter three to do prototype of social network sites there should be a profile structure thus this profile structure needs an interface. Accordingly in front end by creating a new package called "dbacforsns" we have done user interface. This interface contains all categories plus an account. As we stated in modeling portion to access the system two pieces of information are compulsory. The first is username which is used to identify who the user is and the other is a password, to authenticate an identity. If user does not have an account, the system helps someone to sign-up via "create an account" button as shown in figure 4.1. Generally, if we have an account the system should authenticate (is the mechanism whereby systems may securely identify their user's profile) otherwise creating new account and filling user's profile information are the two mandatory procedures to access the system.

Afterward before saving the profile users should select an information which can be used as a key, if another user know the selected information in one category, the secret key, then they can access all information in that category. The profile structures of the prototype looks as shown in figure 4.2. This means users can

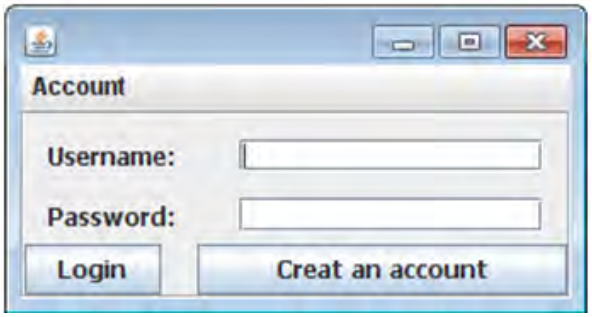


Figure 4.1: Login Interface

decide their own policy for allowing access to their profile information. They can do so by specifying for each category of data item how many items must be provided to access the rest of information in that category. Someone may need only two of the data item to be a secret key on the other hand others may construct the key from three of the data item. To sum-up the strength of the access policy is determined by owner of the profile. Using these users can set their own access policy. On the other hand, the two ends should communicate each other so that each data which is filled by the user will be saved in the back end.

We create new MySQL connection called "DBACMYSQL". It uses standard Transport Control Protocol/Internet Protocol (TCP/IP) method to connect to Relational Data Base Management System (RDBMS) with Hostname 127.0.0.1 (Name of the server host), Port number 3306 (TCP/IP port), Username (Name of the user to connect with) "root" and Password parameters. Then we create a new schema called "account". This schema contains tables.

Hereafter to connect a database within the IDE we register an appropriate JDBC driver called MySQL through Driver class com.MySQL.jdbc.Driver. Then we establish a database connection and this is represented by a database connection node. The details required to connect to a database are driver, database type, and username-password information. So our database Uniform Resource Locator (URL) looks like as follows: jdbc:mysql://localhost:3306/account.

When a user save profile immediately each item will be (since it is in string form)

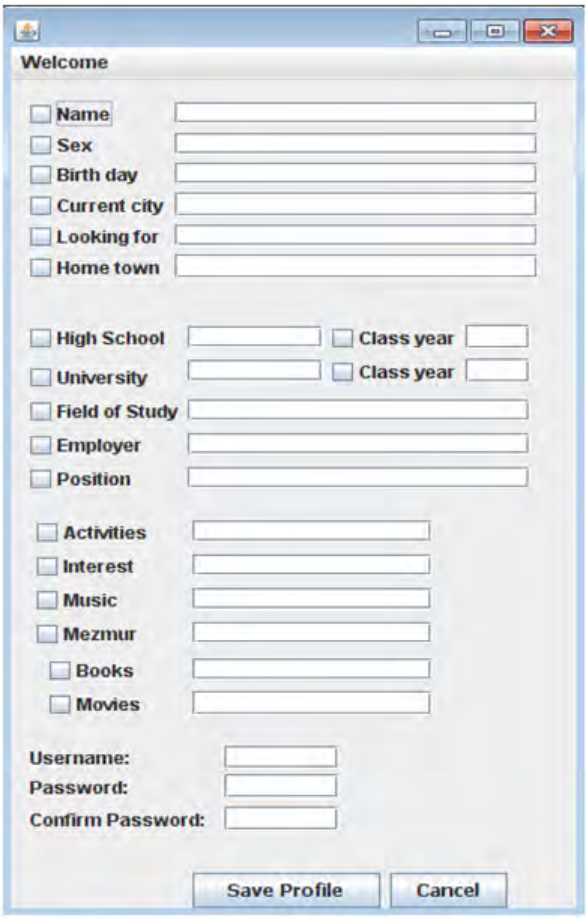


Figure 4.2: Profile structure for new user

converted to numerals by using djb2 hash function. For instance if we consider basic information of Alice, the hashed value for the data item looks like as shown in table 4.1. Depending on the access control method the hashed value has two purposes. The primary use is to construct the polynomial function and the other usage of this value is dependent on the access control method, it will be explained in detail below. Encryption and form of values which are stored in database are both dependent on type of access control method. In this thesis work we tried to implement and see the above three access control methods which are specified in chapter three. So that value stored and encryption which is implemented for the prototype will be explained below.

Direct data-based access control

The first access control scheme is direct data-based access control scheme. All

Table 4.1: User data items and the corresponding hash value in basic information category

Basic Information	User data	Hashed value
Name	Alice	478
Sex	Female	586
Birth Day	3/24/1996	464
Current city	AA	130
Home town	BahirDar	1064
Looking for	Networking	765

data except name of the owner are going to be hashed and stored in the profile table of the database because we need name of the user to search another account profile. Furthermore, in each category hashed value has been taken as y-coordinate orderly. By using those points we construct matrixes and calculate determinant of the matrix. Coefficients of the polynomial have been calculated then the polynomial function has been easily constructed using the coefficient. Therefore using the polynomial function we calculate and published $n - k$ extra points, where n is number of data items and k is points which is published in one category. The purpose of these extra points is for decryption means that if one knows k data items, the secret key, plus with these additional $n - k$ published points, the polynomial function can be reconstructed, because all points are in one polynomial function.

So that to decrypt and retrieve items in one category, first the hashed value of the required items displayed. However, due to limitation of one way hashed functions we cannot change the hashed value back to strings. Consequently with the intention of this the implementations of direct access control method for social networking sites are not successful. The reason is in social network sites through one category the data items are more than one so that retrieving an item is one challenge that we have faced during implementation of this access control method.

Order Invariant data-based access control

The second access control method is order invariant data based access control. Unlike data based access control method it is not concerned about the order of the data. In this method the hashed value of data items in one category are taken as x-coordinate. Moreover, the secret here is random polynomial function with degree $k-1$ so that using this function and x-coordinate value we get the corresponding y-coordinate values. The first and big challenge here is constructing random polynomial function. In addition to this, as we say above even though the difference between direct and indirect data based access control method is order of data item, the problem of two ways hash function here also exist. Therefore because of the above challenges we didn't farther go in implementation part of this scheme. To sum up, direct and order-invariant data based access control scheme have the same nature. Both of them needs two ways hash function and this is also the limitation of the two access control method. Thus they are not appropriate for social network sites.

Indirect data-based access control

The third access control method that we tried to implement for our system model is indirect data based access control method. For the first time to sign up, users click create login button and enter their profile. The profile in each category will be hashed and encrypt using djb2 and blowfish respectively. Hereafter the encrypted data item has been saved on profile table "profile" of the server database "account". Then using the secret key of the standard encryption, blowfish, plus hashed value of all data items in one category assigned to y-coordinate. Simultaneously the x-coordinate are assigned integer value from 1 to n number of data items and zero value to the encryption key consequently. Using those $n+1$ points, by calculating coefficients, n degree polynomial function constructed, where n is number of data items. So that after constructing polynomial functions since we need additional $n+1-k$ points we calculate and gain the corresponding y-coordinates. These points are points which should be published (stored).

In indirect data base access control method we have two secrets. One of the secret key, S, is initially used to encrypt the data item and this secret key also being taken as one point to create the second secret key by establishing polynomial function. From the polynomial function the constant value, function at zero, is the secret key, S. Therefore the other secret key is the main key for our model because unless we know this secret key we cannot rebuild polynomial function and at the same time we cannot get the first secret key, S, from the function at zero.

The next step is reconstructing n degree polynomial function. As we explain in chapter 3 the function is reconstructed from $n + 1$ points (k point from user input and the rest $n + 1 - k$ points are known, which is published before). If the user enters the exact k number of data items in one category which is asked by the owner of the data, then the polynomial is reconstructed. The constant value from the polynomial function is the secret key to decrypt the data. Entering k number of exact data items in a category is sufficient and necessary to see the rest information in that category because we can construct the polynomial function.

Using appropriate username one can login and search others profile by entering name of the people that we need. Figure 4.4 shows an interface to search a person. First users should select a category and should enter name of a person to be search.

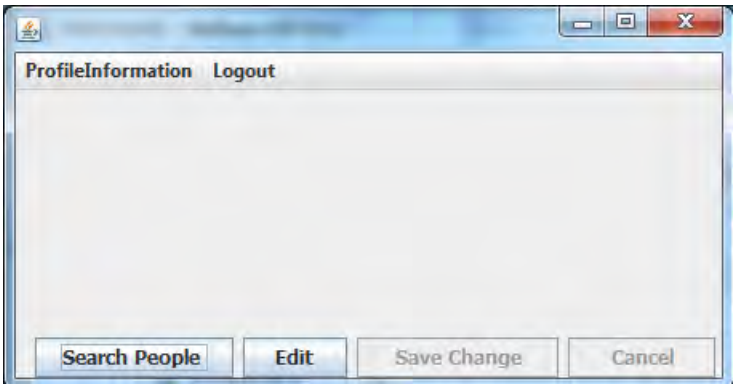


Figure 4.3: Steps to search a person

The next step is calculating the polynomial functions (offline). To do this necessary information should be fulfilled. As figure 4.5 show to know the basic information

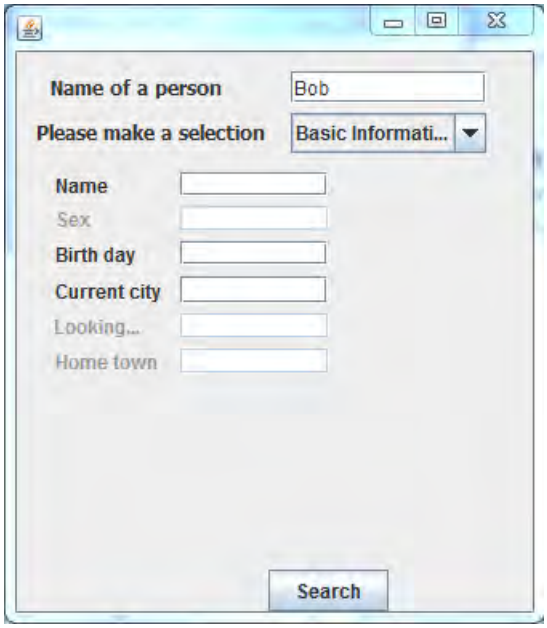


Figure 4.4: Searching Bob's basic information

of Bob one should know his birth day and current city information (i.e. to know basic information of Bob, birthday and current city are the secret keys and is mandatory). As the necessary information gained, immediately using djb2 hash function string inputs will be changed in to numerals. Then the polynomial function will be constructed. After inserting input data the whole thing will be done off-line.

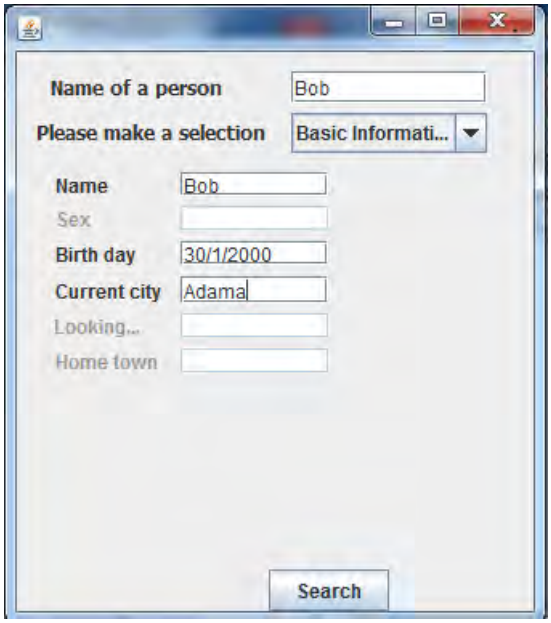


Figure 4.5: Entering first secret key

Finally if those information valid, one can retrieve the rest information of Bob as shown in figure 4:6. To sum up we can retrieve and see others information by entering sufficient number of profile attributes already known.

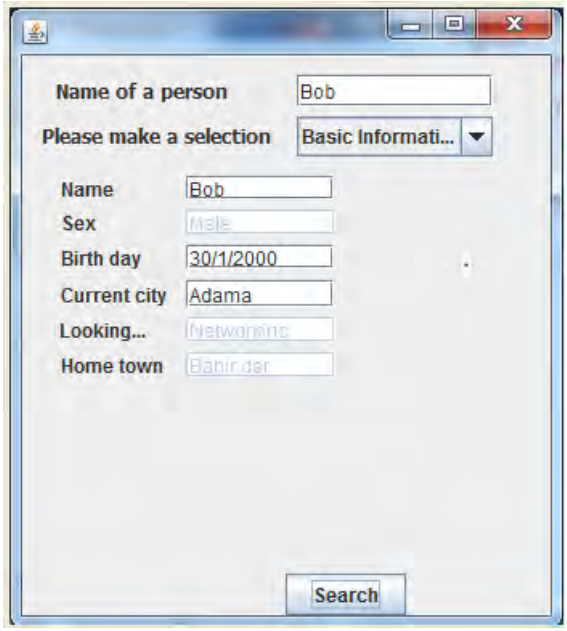


Figure 4.6: Bob's basic information

Chapter 5

Results and Discussion

5.1 Introduction

Testing and performance evaluation of the prototype data-based access control system is the final step that helps to measure whether the system achieves the proposed objectives or not. Our implementation runs the entities (such as clients and key server) locally. The database also runs at the same machine, and we use a local host-only network connection to send queries and retrieve the data.

To run the queries, we execute each query 10 times to get the average execution time and standard deviation. Executing a query involves sending a request to the database, fetching the result and to put the result on the screen. In order to determine how our solution scales, we split our dataset (of 1000 users) in four parts. We conduct the experiments with 100 users, with 250 users, with 500 users and with the entire dataset of 1000 users.

This chapter presents performance evaluation of the prototype system.

5.2 Evaluation

Securing users' privacy in OSNs requires the following properties: The first is a need to hide user's private information from anyone also including the OSN provider and other than those authorized by the user. Further there is a need to allow users to flexibly define their own access control policy. As we explain in chapter one the main objectives of this thesis work are allowing users to set and access their own access policy and the other is securing user's profile information from the provider side. So after implementing our system we set three parameters to evaluate our prototype. Those parameters are confidentiality, scalability and flexibility.

5.2.1 Confidentiality

When we store profile information of a user in a server, there is a need of security, an adversary should not be able to obtain the sensitive data. Profile information should only be accessible to people that have only access to the data. We evaluate the confidentiality of our system from two points of view. The first is from user side, when we save or modify our profile information; since it is encrypted from user machine any one cannot eavesdrop and read any data item. This means a user can only modify their profile by re-encrypting their entire profile and replacing their old profile with the newly encrypted one. On the other hand, user's who has a secret key for a category can retrieve and see data items only for this category. The other is from provider side, providers cannot retrieve and see any user's data from the server because every profile data in each category is encrypted, from the user machine, and recorded in the database. The Snapshot in Appendix A of sample profile in the database shows this result. To sum up providers cannot retrieve and read an information in the database.

5.2.2 Scalability

Scalability is one of the evaluation matrices considered in this paper. In order to determine the scalability, which is capability of the technique to handle large amounts of high dimensional data, of our prototype we use the above dataset classification with different username and password combination. During login the response time for the three cases is the same. In order to search another user someone should enter sufficient amount of data. Using those and adding some data from the database searching and decryption will be conduct. If the data is correct, the response time to search a user in a category is as shown in figure 5:1 below. The results show that the variation in response time between different dataset is relatively small. As the number of registered user increases, the variations of response time to search a person in four cases are almost indistinguishable. This indicates that as the number of users increase the duration to search a user is reasonably around the same.

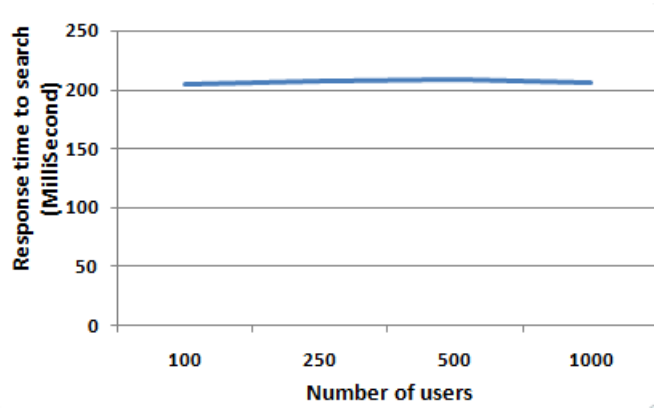


Figure 5.1: Scalability of prototype

On the other hand, the response times to decrypt information for different number of keys are shown in figure 5:2. As we explained in chapter three when we create an account we select different data items to be a secret key in each category. Such as, in the basic information category there are 6 data items so that depending on their interest during creation of an account users can choose their own data as a secret. The number of data which is decrypted depends on the number of data

which is used as a key. As the number of key increases, number of decrypting data will decrease.

As the figure below shows, the x-axis represent number of secret key which is selected by the user in a category. In order that a user can set his/her own number of secret key. For example, number 6 represent all data are used as a secret key (all data items in basic information category is private data). On the contrary, if someone does not set any secret key, all information in a category will be public. Besides this a user can set his/her own access control policy. Hence in basic information category there are five modes of key. As the result indicate the difference between five modes of key is hard to see by naked eye because it is relatively small. Moreover, as the results show decryption time do not have any relationship with the scale up of the user

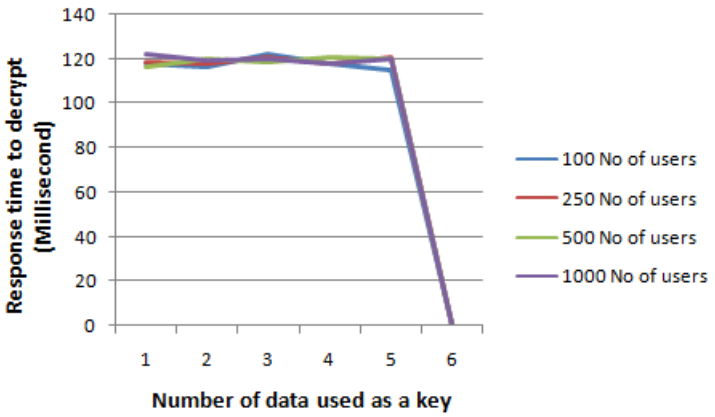


Figure 5.2: Performance result for different number of key of prototype

5.2.3 Flexibility

One final feature we have provided is that users can decide their own policy for allowing access to their profile information. They can do so by specifying for each category of information how many items must be provided to access the remainder of the information in that category.

The table below shows an overhead to search a person. We compared response

time of three different cases. From table below the second row represents a prototype without any access control; information in the database is stored in form of plaintext. Third row in the table correspond to the time that it takes to respond when we use only standard encryption (like nowadays access control method an encryption happen in the server side). The last row signify indirect data based access control; users can choose their own policy for allowing access to their profile information. Hence the security level of indirect data based access control method is multi layered. Since encryption happens in users machine, providers (in the server side) and also eavesdroppers do not have any information about the user.

Table 5.1: Response time to search a person in three different access control methods

Control Method	Response Time(milliseconds)
Without access control	82
Blowfish encryption	134
Indirect data based access control	216

Chapter 6

Conclusions and Recommendations

6.1 Conclusion

In this paper, we introduced data based access control polices in social networks. Instead of relying on credentials given by users for access to other users' information, access control is made possible by knowing a sufficient amount of information from others' profile. Different from traditional access control methods, the users' knowledge is the key to other users' profile, which is more difficult to be copied, stolen or forgotten.

The privacy protecting mechanism is fully customizable to satisfy users' privacy needs. In which each user was the administrator to personalize privacy settings and to define access rights. Therefore for better protecting the privacy of users, changing control from provider side to user side is essential. Finally, providing user the ability to flexibly define new access control mechanism to their profile information, in these mechanism OSN providers are getting limited trust and

users are given full control over their personal information. In summary, users can set their own access control policy dynamically.

In addition, we implemented a data-based access control system for access to users' profile on social networks. Among three different schemes of data based access control policy as we explain in the implementation part, due to limitation of two way hash function, only indirect data based access control method is suitable to social network sites. In this system, users can enter their own profile information and set access polices. The profile can be requested by another user and decrypted by entering a sufficient number of profile attributes already known.

Moreover, this access control method makes the social network sites security level to be multi layered. We plan to extend our work along several directions. A first extension concerns a practical mechanism in social network supporting this prototype. In addition to this, we plan to concentrate on DBAC polices that can be applied to information from mixed categories.

6.2 Recommendation

There are several possible refinements to this implementation we have considered but not yet implemented. These candidates for future work are discussed below. First, it would be possible for users to specify the structure of their own profile. This can be easily implemented but has the disadvantage that the structure of the profile would need to be available to everyone. For example a user might want to share their employment history, which can be retrieved only by those who already have some knowledge of this and is in-accessible to others, however any user can see the number of employment entries this user has in their profile, thereby leaking some information.

A second issue is that the underlying secret sharing technique only allows for a symmetric data access policy. A way of emulating an asymmetric policy using this system would be to assign integer weights to data items by storing profile attributes multiple times, optionally mangling them before encryption, in the same encrypted profile. This way it would even be possible to have a master attribute which decrypts every other attribute in the category. Just as with the feature described above, in this case information on the weights should be published, possibly leaking some information. Repetition of information could also possibly weaken the encryption scheme and allow for cryptographic attacks. This scheme does not directly allow for a weight of zero, i.e. have a piece of information revealed by decryption, but do not let this information itself grant access to further information, however, this functionality can also be provided by padding information with random data in the encryption step and transparently removing the padding after decryption.

Finally, we have decided to keep the different profile attributes disjoint. This means that for example employment information cannot grant access to personal information and vice verse. It would be possible to specify rules that would have

personal information grant access to employment information, but not the other way around. A policy such as this is not supported by the secret sharing technique, however this can be achieved by mixing information of different categories and using the emulation of weight based asymmetric policy described above.

Bibliography

[1] D. Boyd and N. B. Ellison. *Social network sites: Definition, history, and scholarship.*Journal of Computer-Mediated Communication, 13(1):210–230, 2007.

[2] Barbara Carminati, Elena Ferrari, and Andrea Perego Dicom. *Enforcing access control in web-based social networks.* Transactions on Information and System Security,10 2009.

[3] Aspan Maria. *How sticky is membership on facebook? just try breaking free.*New York Times, 02 2008.

[4] Wolter Pieters and Qiang Tang. *Data is key: Introducing the data-based access control paradigm.* Data and Applications Security, 5645/2009:240-251, 2009.

[5] Lampson, B. W. *Protection.* 5th Princeton Symp. Information Science and Systems (Mar. 1971),pp. 437-443. Reprinted in Operating Systems Rev. 8,1 (Jan. 1974), 18-24.

[6] Benantar, M. *Access Control Systems: Security, Identity Management and Trust Models.* Springer, New York, NY, 2006.

[7] Bell, D. E., and LaPadula, L. J. *Secure Computer Systems: Mathematical Foundations.* Rep. MTR-2547, The MITRE Corporation, Bedford, MA,Mar. 1973.

[8] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. *Role-based access control models.*IEEE 14th IEEE Computer, vol. 29, no. 2, pp. 38–47, 1996.

[9] B. Carminati, E. Ferrari, and A. Perego. *Rule-based access control for social networks.* In On the Move to Meaningful Internet Systems 2006: OTM workshops 2006, pages 1734–1744, 2006.

[10] Wilfred Villegas, Bader Ali, and Muthucumaru Maheswaran. *An Access Control Scheme for Protecting Personal Data,*2010

[11] S. Guha, K. Tang, and P. Francis. *NOYB: Privacy in Online Social Networks.*In WOSN, 2008.

[12] M. M. Lucas and N. *FlyByNight: Mitigating the Privacy Risks of Social Networking.* In WPES, 2008.

[13] R.L.Rivest, A.Shamir,andL.Adleman. *A method for obtaining digital signatures and public key crypto systems.* Commun.of the ACM, 21:120-126, 1978.

[14] Rui A. Costa. *Information-Theoretic Security: an overview*

[15] Sennur Ulukus. *Information Theoretic Security* 2012 European School of Information Theory, April 2012, Antalya, Turkey

[16] A.Shamir. *How to Share a Secret.* Commun. ACM, vol.22, no.11, pp.612–613, 1979.

[17] G.R.Blakley. *Safeguarding Cryptographic Keys.* Proc. AFIPS, vol.48, pp.313–317, 1979.

[18] Jun Kurihara, Shinsaku Kiyomoto, Kazuhide Fukushima, and Toshiaki Tanaka. *A New (k, n)-Threshold Secret Sharing Scheme and Its Extension.* 11th Information Security Conference September 15th–18th, 2008 Taipei, Taiwan.

[19] Y. Dodis, L. Reyzin, and A. Smith. *Fuzzy extractors: How to generate strong keys from biometrics and other noisy data.* In EUROCRYPT '04, volume 3027 of LNCS, pages 523–540. Springer, 2004.

[20] Zhijie Shi, Chujiao Ma, Jordan Cote, and Bing Wang. *Hardware Implementation of Hash Functions.* Springer Science+Business Media, LLC 2012.

[21] John Edward Silva. *An Overview of Cryptographic Hash Functions and Their Uses.* January 15, 2003.

[22] Sylvain Collange, Yoginder Dandass, Marc Daumas, David Defour. *Using Graphics Processors for Parallelizing Hash-based Data Carving,* 2009.

[23] Arash Partow. *General Purpose Hash function Algorithm.* Available on, http://www.partow.net/programming/hashfunctions, accessed on 3/17/2016.

[24] Hardjono. *Security In Wireless LANS And MANS.* Artech House Publishers 2005. bibitem25 Ali Ahmad Milad, Hjh Zaiton Muda, Zul Azri Bin Muhamad Noh and Mustafa Almahdi Algaet *Comparative Study of Performance in Cryptography Algorithms (Blowfish and Skipjack).* Journal of Computer Science 8 (7): 1191-1197, 2012, ISSN 1549-3636

[25] Gurjeevan Singh, Ashwani Kumar and K. S. Sandha. *A Study of New Trends in Blowfish Algorithm.* International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622, Vol. 1, Issue 2, pp.321-326.

[26] *String Hash Function.* Available on, http://www.cse.yorku.ca/ oz/hash.html, accessed on 3/17/2016.

[27] Jenkins B. *A Hash Function for Hash Table LookUp.* Available on, http://burtleburtle.net/bob/hash/doobs.html, accessed on 3/18/2016.

[28] Anjula Gupta and Navpreet Kaur Walia. *Cryptography Algorithms: A Review.* ISSN: 2321-9939 , Volume 2, Issue 2 , 2014 IJEDR.

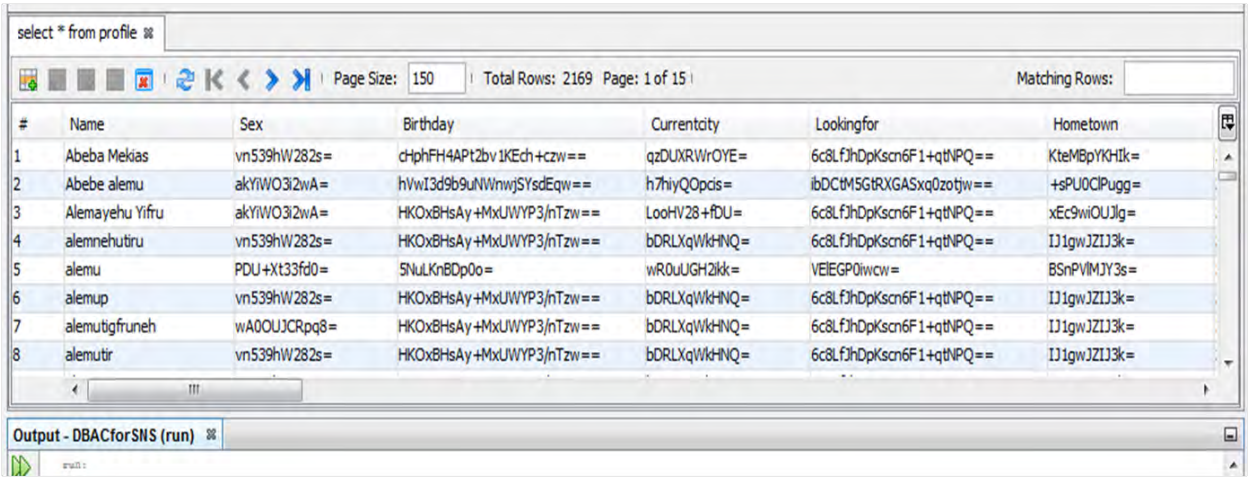
[29] D. S. Abdul. Elminaam, H. M. Abdul Kader and M. M. Hadhoud. *Performance Evaluation of Symmetric Encryption Algorithms.* Communications of the IBIMA ISSN: 1943-7765, Volume 8, 2009.

[30] Simar Preet Singh, and Raman Main. *Comparison of Data Encryption Algorithms.* IJCSC, Volume 2, No. 1, January-June 2011, pp. 125-127

[31] Pratap Chandra Mandal. *Evaluation of performance of the Symmetric Key Algorithms: DES, 3DES ,AES and Blowfish* Journal of Global Research in Computer Science Department of Computer Application, vol 3, pp 67-70, August 2012.

Appendix

Appendix A Decrypted data in the Database



#	Name	Sex	Birthday	Currentcity	Lookingfor	Hometown
1	Abeba Mekias	vn539hW282s=	cHphFH4APt2bv1KEch+czw==	qzDUXRWrOYE=	6c8LfJhDpKscn6F1+qtNPQ==	KteMBpYKHIk=
2	Abebe alemu	akYiWO3i2wA=	hVwI3d9b9uNWnwjSYsdEqw==	h7hiyQOpcis=	ibDCtM5GtRXGASxq0zotjw==	+sPU0ClPugg=
3	Alemayehu Yifru	akYiWO3i2wA=	HKOxBHsAy+MxUWYP3/nTzw==	LooHV28+fDU=	6c8LfJhDpKscn6F1+qtNPQ==	xEc9wiOUJlg=
4	alemnehutiru	vn539hW282s=	HKOxBHsAy+MxUWYP3/nTzw==	bDRLXqWkHNQ=	6c8LfJhDpKscn6F1+qtNPQ==	IJ1gwJZIJ3k=
5	alemu	PDU+Xt33fd0=	5NuLKnBDp0o=	wR0uUGH2ikk=	VElEGP0iwcw=	BSnPVlMJY3s=
6	alemup	vn539hW282s=	HKOxBHsAy+MxUWYP3/nTzw==	bDRLXqWkHNQ=	6c8LfJhDpKscn6F1+qtNPQ==	IJ1gwJZIJ3k=
7	alemutigfruneh	wA0OUJCRpq8=	HKOxBHsAy+MxUWYP3/nTzw==	bDRLXqWkHNQ=	6c8LfJhDpKscn6F1+qtNPQ==	IJ1gwJZIJ3k=
8	alemutir	vn539hW282s=	HKOxBHsAy+MxUWYP3/nTzw==	bDRLXqWkHNQ=	6c8LfJhDpKscn6F1+qtNPQ==	IJ1gwJZIJ3k=

Figure 6.1: Snapshot of profiles in the database

Appendix B Algorithms

1. djb2 algorithm

```
unsigned long
hash(unsigned char *str)
{
  unsigned long hash = 5381;
  int c;
```

```
 while (c = * str++)
 hash = ((hash << 5) + hash) + c; /* hash * 33 + c */
 return hash;
}
```

2. SDBM algorithm

```
static unsigned long
sdbm(str)
unsigned char * str;
 {
   unsigned long hash = 0;
   int c;
   while (c = *str++)
   hash = c + (hash << 6) + (hash << 16) - hash;
   return hash;
  }
```

3. Loselose algorithm

```
 unsigned long
 hash(unsigned char *str)
   {
unsigned int hash = 0;
int c;
while (c = *str++)
hash + = c;
return hash;
   }
```