

# PRIVACY PRESERVATION IN ONLINE TRANSACTIONS USING PRIVATE AND SUBSCRIPTION CREDENTIALS

A DISSERTATION

*Submitted in partial fulfillment of the  
requirements for the award of the degree*

*of*

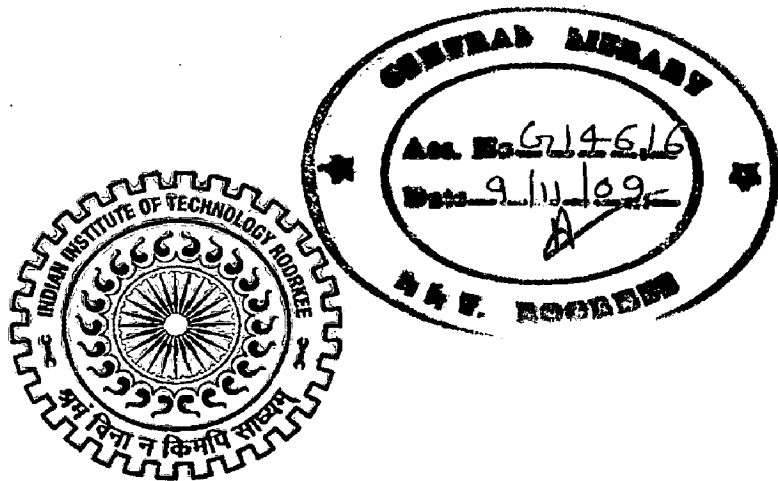
**MASTER OF TECHNOLOGY**

*in*

**INFORMATION TECHNOLOGY**

By

**ATHAVALE ADITI YOGANAND**



**DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING  
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE**

**ROORKEE -247 667 (INDIA)**

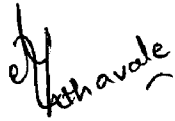
**JUNE, 2009**

## Candidate's Declaration

I hereby declare that the work being presented in the dissertation report titled "**Privacy Preservation in Online Transactions using Private and Subscription Credentials**" in partial fulfillment of the requirement for the award of the degree of Master of Technology in Information Technology, submitted in the Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, is an authentic record of my own work carried out under the guidance of Dr. Kuldip Singh, Professor, in Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee. I have not submitted the matter embodied in this dissertation report for the award of any other degree.

Dated: 16-6-2009

Place: IIT Roorkee

  
(Athavale Aditi Yoganand)


---

## Certificate

This is to certify that above statements made by the candidate are correct to the best of my knowledge and belief.

Dated: 16-6-2009

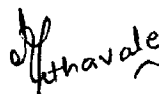
Place: IIT Roorkee.

  
**Dr. Kuldip Singh,**  
Professor,  
Department of Electronics  
and Computer Engineering,  
IIT Roorkee, Roorkee,  
247 667 (India).

## ACKNOWLEDGEMENTS

I am thankful to Indian Institute of Technology Roorkee for giving me this opportunity. It is my privilege to express thanks and my profound gratitude to my supervisor Dr. Kuldip Singh, Professor for his invaluable guidance and constant encouragement throughout the dissertation. I was able to complete this dissertation in time due to the constant motivation and support received from him.

I am also grateful to Mr. Sandeep Sood for helping me to clarify some basic and important concepts explored in the dissertation work. I am also thankful to all my friends who helped me directly and indirectly in completing this dissertation. Most importantly, I would like to extend my deepest appreciation to my family for their love, encouragement and moral support.



**(ATHAVALE ADITI YOGANAND)**

# ABSTRACT

---

In today's electronic society a large number of transactions are performed online. In order to perform these transactions, users share their personal data with number of organizations. There are chances that the data is transferred to other organizations, misused or misinterpreted. The data, in turn, is compiled to build detailed profiles of users. These profiles then can be used against the user in subsequent transactions. Thus there is a need to preserve the privacy of the user at the service provider's side.

Private credentials technology has been a leading privacy preserving technology to provide user authentication as well as authorization while keeping the user privacy intact. But still there are chances of linkability between show protocols of the same credential. Due to this and other requirements, a concept of subscription credentials has been developed and implemented in this dissertation work. Subscription credentials are specialized credentials built on the top of private credentials and are used for subscription based services. The constructs used for developing subscription credentials are based on that used for private credentials. Thus the subscription credentials are compatible with the private credentials. These credentials have the advantages such as verifier can encode attributes into the credential and less linkability as compared to private credentials.

Subscription credentials have been constructed using the elliptic curve arithmetic. Elliptic curve arithmetic has the property that it provides equal security in smaller key sizes as compared to that provided by its RSA or discrete logarithm based counterparts. Subscription credentials are based ECDLP i.e. Elliptic Curve Discrete Logarithm Problem. ECDLP is supposed to be harder than DLP i.e. Discrete Logarithm Problem. Thus the scheme presented here provides more security.

# Table of Contents

CANDIDATE'S DECLARATION	i
ACKNOWLEDEMENTS	ii
ABSTRACT	iii
TABLE OF CONTENTS	iv
LIST OF FIGURES	vii
LIST OF TABLES	viii
LIST OF ACRONYMS	ix
<b>CHAPTER 1 INTRODUCTION</b>	<b>1</b>
1.1 Introduction and Motivation	1
1.2 Statement of the Problem	2
1.3 Organization of the Report	2
<b>CHAPTER 2 PRELIMINARIES</b>	<b>4</b>
2.1 Authorization and Authentication in Online Transactions	4
2.2 Notion of Privacy	4
2.3 Requirements of Privacy Preserving Technologies	5
2.4 Credentials	6
2.5 DL Problem and DLREP function	7
2.6 Elliptic Curve Cryptography	8
2.7 ECDL Problem and ECDLREP function	12
2.8 Attacks on Discrete Logs and Elliptic Curve Discrete Logs	12
<b>CHAPTER 3 LITERATURE REVIEW AND RELATED WORK</b>	<b>14</b>
3.1 Untraceable Mails and Digital Pseudonyms	14
3.2 Pseudonym Systems	15
3.3 Pseudonym Technology for E-services	16
3.4 Prime Project	17

3.5	Privacy Enhancing Credentials	18
3.6	Private Credentials	19
3.7	Proxy Blind Signature Scheme Based on DLP and ECDLP	20
3.8	Research Gaps	21
<b>CHAPTER 4 PROTOCOLS FOR PRIVACY PRESERVATION</b>		<b>23</b>
4.1	Private Credential Protocols in Elliptic Curve Cryptography	23
4.2	Subscription Credentials	27
4.3	Subscription Credentials Issue Protocol	29
4.4	Subscription Credentials Show and Update Protocol	31
<b>CHAPTER 5 IMPLEMENTATION</b>		<b>35</b>
5.1	Generation of Group Parameters	35
5.2	Selection of Elliptic Curves	35
5.3	Java Implementation	36
	5.3.1 Platform	36
	5.3.2 Socket Programming	37
	5.3.3 Special Classes	38
	5.3.4 Package Structure	39
<b>CHAPTER 6 RESULTS AND ANALYSIS</b>		<b>41</b>
6.1	Privacy Preservation Analysis	41
6.2	Security against Known Attacks	42
<b>CHAPTER 7 CONCLUSION AND FUTURE WORK</b>		<b>44</b>
7.1	Conclusion	44
7.2	Future Work	45
<b>REFERENCES</b>		<b>46</b>
<b>PUBLICATIONS</b>		<b>49</b>

<b>APPENDIX</b>	<b>50</b>
1 Private Credential's Protocols	51
2 Binary Scalar Multiplication Method used in the Implementation	52
3 ECC Parameter by Certicom	52

## List of Figures

Fig. 2.1	$E(-1,0) : y^2 = x^3 - x$	8
Fig. 2.2	$E(1,1) : y^2 = x^3 + x + 1$	9
Fig. 2.3	The Elliptic Curve $E_{23}(1,1)$	10
Fig. 3.1	Mixer Network Proposed by Chaum	14
Fig. 3.2	Chaum Pedersen E-wallet Architecture	16
Fig. 3.3	Blind Signature $(r', s')$ on message $m$	21
Fig. 4.1	Private Credential Issue protocol (in ECC)	24
Fig. 4.2	Private Credential Show Protocol (in ECC)	26
Fig. 4.3	Subscription Credential Issue Protocol (in ECC)	30
Fig. 4.4	Subscription Credential Issue Protocol	31
Fig. 4.5	Subscription Credential Show-and-Update Protocol (in ECC)	32
Fig. 4.6	Subscription Credential Show-and-Update Protocol	34
Fig. 5.1	Java Package Structure	39



## List of Tables

Table 2.1	Points on Elliptic Curve $E_{23}(1,1)$ .....	9
Table 2.2	Correspondence between $Z_p^*$ and $E(Z_p)$ .....	11
Table 2.3	Equivalent Key Sizes in RSA and ECC.....	11
Table 5.1	Roles of User, CA, and Service Provider in Client–Server Architecture.....	37

## **List of Acronyms**

<b>Acronym</b>	<b>Full Form</b>
CA	Certification Authority
DLP	Discrete Logarithm Problem
DLREP	Discrete Logarithm Representation
DSA	Digital Signature Algorithm
ECC	Elliptic Curve Cryptography
ECDLP	Elliptic Curve Discrete Logarithm Problem
ECDLREP	Elliptic Curve Discrete Logarithm Representation
IOI	Item of Interest
PKCS	Public Key Cryptography Standards
PRIME	Privacy and Identity Management for Everyone
SHA	Secure Hash Algorithm
SSL	Secure Socket Layer
TCP	Transmission Control Protocol
TLS	Transport Layer Security

# CHAPTER 1

## INTRODUCTION

---

### 1.1 Introduction and Motivation

These days, e-literacy is increasing at an exploding rate. We use Internet for different types of actions. Internet is being utilized in business, education sector, games industry and for a seamless access to information. Internet has changed our way of life, with number of online actions being performed per day. We come across numerous websites, different online organizations while using Internet. These organizations provide different online services, or access to their valuable online resources. To avail this access, one transacts with these organizations. For doing so, one might be asked to fill in some information about oneself. Sometimes, the registration with the concerned organization is also needed to get ahead with the transaction. This information can be more than that required to perform a certain transaction.

In the name of providing more personalized, efficient and appropriate service to a user, organizations store huge amounts of data related to user in their databases. The data persists even if the user has finished his transaction. For business purposes, organizations share this data with each other. The organizations link the data on the basis of the identity of users and form detailed profiles of users [1]. The data includes information about oneself in the form of attributes. It also includes the information about various actions that a user performs on the Internet. Such data is used for advertising, for targeting users with specific and closed range of choices. The data might even be used for some illegal purposes. Further, the data is often used to provide discriminated services to the users.

To prevent such activities, it has become extremely important for users to keep their personal information private and secure. The meaning of privacy is multifaceted. Sometimes privacy of data means that the data can not be accessed by certain entities in a system. These entities include intruders, third party organizations and sometimes, the organizations themselves. That means privacy

issues need to be tackled at various levels. We require confidentiality of the message at the network level, so that any active or passive intruders should not get access to our information. We require privacy of our data even at the end of service provides from whom we are availing the services. In this dissertation report, privacy of users points to the later type.

To prevent the transfer of data or to maintain confidentiality of the data, encryption techniques are used. Basics of cryptography that have been used here mainly belong to the field of abstract algebra and number theory. Constructs used in the presented work are based on discrete logarithms as well as on elliptic curve arithmetic. Using these constructs, various features of privacy such as anonymity, authentication, unlinkability etc. can be implemented effectively.

## **1.2 Statement of the Problem**

For preserving privacy of users in online transactions, a scheme of subscription credentials, is proposed and implemented in this dissertation work. The objective of the presented work is to provide various features of privacy to the user by means of subscription credentials.

## **1.3 Organization of the Report**

This dissertation report proposes protocols of subscription credentials which are used to enhance the privacy of the user in online transactions. The organization of the report is as follows:

Chapter 2 covers the preliminaries which form a basis of the proposed work. It includes understanding of basic concepts like authentication and authorization, their necessity in online transactions, elaboration of the term 'privacy' in the context of the work, and the requirements of the privacy preserving techniques. It also includes some mathematical fundamentals of discrete logarithm problem, basics of elliptic curve cryptography and methods to solve discrete logarithm problems.

Chapter 3 discusses some of the important work in this field. The work discussed in this section is ordered in such a way that various concepts are unveiled from a generic sense to a detailed and specific work highlighting the underlying concepts.

Chapter 4 discusses redefinition of an existing protocol of private credentials in terms of elliptic curve cryptography. Then the concept of subscription credential is presented. Then subscription credentials' issue protocol and subscription credentials' show-and-update protocol are proposed. While discussing these two protocols, computations based on elliptic curve arithmetic are presented first, followed by their discrete logarithm based analogs.

Chapter 5 presents a brief description of the implementation of the above mentioned protocols. It covers details like how the elliptic curves are chosen, how the group parameters for discrete logarithm based schemes are chosen.

Chapter 6 discusses how the proposed scheme of subscription credential satisfies the privacy related properties. It then presents the results of the execution of the protocols. It then shows how the scheme is resistant to attacks based on mathematical properties of the protocol.

Chapter 7 concludes the dissertation work and gives suggestions for future work.

## CHAPTER 2

### PRELIMINARIES

---

#### 2.1 Authorization and Authentication in Online Transactions

Authentication is described as the process of verifying a claim which is made regarding an identity, or a set of attributes which relates to an identity. Traditionally authentication was mainly required in organizations to meet their own security needs. With the widespread use of personal computers and Internet, it has become an extremely necessary part of online transactions. There are different ways to authenticate oneself to computer systems. One can either tell the computer what he or she knows. An example of this type of authentication is a password based system. Another way to authenticate is to show something that one has; for example, a certificate or a smart card. Otherwise one can authenticate oneself by means of a physical trait i.e. fingerprint, or voice etc [2]. First two types of authentication techniques are more prevalent in online transactions.

Authorization on the other hand is the process of verifying if one is allowed to perform the operations that he or she is requesting to do. Even if authentication and authorization are different concepts, most of the times authentication forms a prerequisite for authorization. Authorization is ensured by storing records of all the privileged users per service or by listing all the allowable services per user [2].

With the widespread availability of the Internet, organizations provide access to their resources and services to internal as well as external users. These users have varying needs and permissions to access these resources and services. Thus authentication and authorization techniques are required to facilitate these users the appropriate access of online resources.

#### 2.2 Notion of Privacy

According to the definition given by Goldberg [3], privacy refers to the ability of individuals to control the collection, retention, and distribution of information about themselves. This information is of varying forms such as age, address, birth

date, birthplace, nationality, email address, IP address, and so on[2]. Privacy does not mean that this information shall never be revealed to anyone, but it maintains that user should be aware of its use other than as specified by service requirements.

In this report, term privacy is perceived as the privacy of user's data at the service providers side. It does not include the privacy of the data when the data is transmitted over network. Thus the protocols proposed in this dissertation work assume that the channel over which data is transmitted are encrypted. Apart from this, the proposed work also assumes that the underlying channels are semi-anonymous; i.e. an organization with whom a user is transacting with, is authenticated to user and the user is unauthenticated at that point. A variation of Secure Socket Layer (SSL) protocol called as Transport Layer Security (TLS) provides such types of one way authentication[4].

### **2.3 Requirements of Privacy Preserving Technologies**

According to the concept of privacy as described in the previous section, privacy preserving technologies require to satisfy some properties. These properties are classified as privacy related properties and security related properties [1]. Later we also briefly discuss some advance properties of privacy preserving technologies.

Privacy Related Properties:

- I. Anonymity:- Anonymity is the state of being non-identifiable within a set of subjects [5]. It can also be stated as 'privacy of identity'.
- II. Unlinkability:- Unlinkability of two or more items of interest (examples of IOIs are subjects, messages, events, actions etc) means that these IOIs are no more or no less related after an attacker's observation than they are related with his a-priori knowledge [5].
- III. Property sharing resistance:- Users should not be able to share their certified attributes or certificates with other users. One way of achieving this, is associating the secret value of the user with these attributes or certificates.

Security Related Properties:

- I. Authentication:- The property has been described in section 2.1
- II. Unforgeability of Certificates:- The certificates that a user shows to get access to a resource can not be forged without a central authority or an approved issuer of those certificates.
- III. Security of User's Secret Key:- In the protocols the secret key of the user must not be revealed to anyone[6].

Some advance properties include selective disclosure, non-repudiation etc. Selective disclosure property[7] is user's ability to show only the selected attributes to the verifier. Non-repudiation is the property where user can not deny any of his action with service providers.

## **2.4 Credentials**

Credentials are nothing but certificates, issued as a proof of qualification, or authority given to a user possessing it. The issuer of the credentials is a trusted third party, an organization or the user himself. Credentials which we come across in our everyday life are degrees, passports, identity cards, passwords, keys etc.

Credentials used in the context of the presented work are attribute certificates except the fact that these credentials do not contain the identity of the holder. Such type of credentials are called as anonymous credentials [8]. They do have fields which help to prove their authenticity. Standard formats for attribute certificates are defined in [9].

Depending upon their usage, some credentials are made valid only for one time use. Example of one time use credentials is a doctor's prescription. Some credentials are valid forever. These are called as multiple show credentials. In some cases, credentials are valid for 'n' number of times. After using them 'n' times, their validity expires. They are n-times-show credentials.

To use credentials for authorization and authentication in a seamless manner, they should satisfy properties of unforgeability and non-transferability. Unforgeability



property means that the credentials can not be fraudulently formed without the participation of the issuer of the credential. Non-transferability of the credentials means that a credential can not be passed on to other users.

## 2.5 DL Problem and DLREP function

In abstract algebra, discrete logarithms are group-theoretic analogs of ordinary logarithms. Let  $G$  be a finite cyclic group of order  $n$ . Let  $\alpha$  be a generator of  $G$ , and let  $\beta \in G$ . The discrete logarithm of  $\beta$  to the base  $\alpha$ , denoted  $\log_{\alpha} \beta$ , is a unique integer  $x$ ,  $0 \leq x \leq n - 1$ , such that  $\beta = \alpha^x$  [10]. Considering the group  $G$  as the multiplicative group  $Z_p^*$  of order  $p$ , the discrete logarithm problem (DLP) is the following: given a prime  $p$ , a generator  $\alpha$  of  $Z_p^*$ , and an element  $\beta \in Z_p^*$ , find the integer  $x$ ,  $0 \leq x \leq p - 2$ , such that  $\alpha^x \equiv \beta \pmod{p}$  [10].

Various cryptographic applications like Diffie-Hellman Key Exchange Protocol, Digital Signature Algorithm are based on the intractability of the discrete logarithm problem.

Let  $G_{q,p}$  be a subgroup of  $Z_p^*$  of order  $q$ .  $q$  is a prime number such that  $q$  divides  $(p-1)$ . Let  $g_1, g_2, \dots, g_l$  be the generators of this group. Generalization of the DL function, is called as the DLREP function. It is defined as a function with respect to generator  $g_1, g_2, \dots, g_l$  such that

$$f(x_1, x_2, \dots, x_l) = g_1^{x_1} * g_2^{x_2} * \dots * g_l^{x_l} \pmod{p} \quad (2.1)$$

where  $(x_1, x_2, \dots, x_l)$  is a tuple whose elements are in  $Z_q$  [11]. This tuple is called DL-representation of the product  $h = g_1^{x_1} * g_2^{x_2} * \dots * g_l^{x_l}$  with respect to the generators  $g_1, g_2, \dots, g_l$ .

A slight variation of this function, which is used in [7,11] is

$$f(\alpha, x_1, x_2, \dots, x_l) = (g_1^{x_1} * g_2^{x_2} * \dots * g_l^{x_l})^{\alpha} \pmod{p} \quad (2.2)$$

This function acts as one of the basic constructs of private credential technology [7,11], as this function allows to disclose some of the  $x_i$ , thus satisfying the selective disclosure property. At the same time, the value of  $\alpha$ , can be used to prove authenticity of the private credentials. This function acts as the basis of the subscription credential protocols proposed in this report.

## 2.6 Elliptic Curve Cryptography

Majority of the cryptographic protocols use either RSA or Discrete Logarithm based systems. To cope up with the attacks arising out of ever increasing computational power, the key length for these protocols has been increased over past few years. This has put a burden on these cryptographic protocols. A competing technology – Elliptic Curve Cryptography(ECC) is attracting lot of attention these days as it offers equivalent security in smaller key size[12]. Basics of elliptic curve cryptography are covered in the following text.

Elliptic curve equation takes the form of

$$y^2 = x^3 + ax + b \tag{2.3}$$

Curves  $E(-1,0)$  and  $E(1,1)$  are shown in figure 2.1 and 2.2 res.

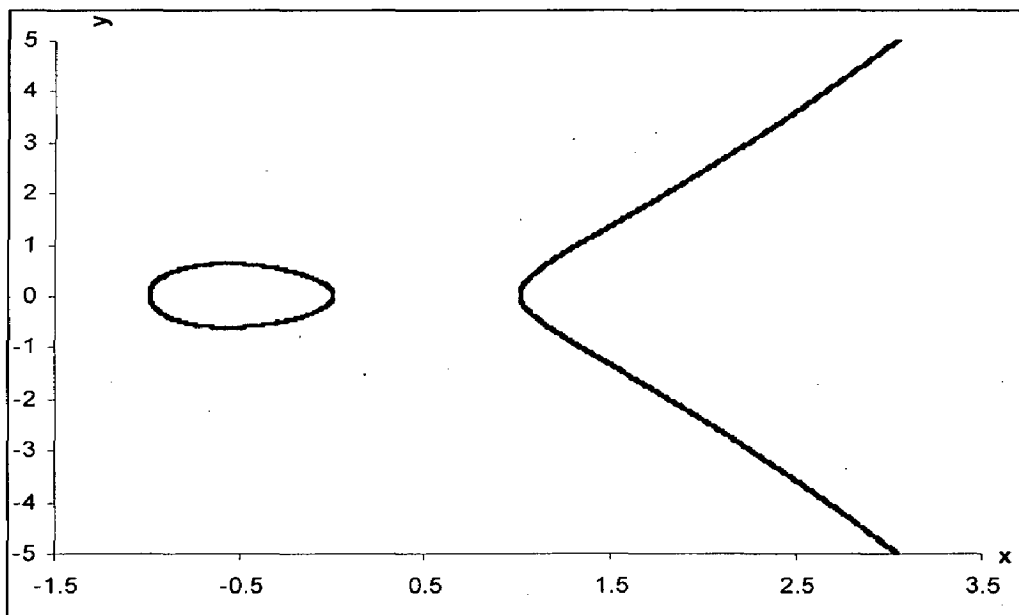


Figure 2.1:  $E(-1,0): y^2 = x^3 - x$

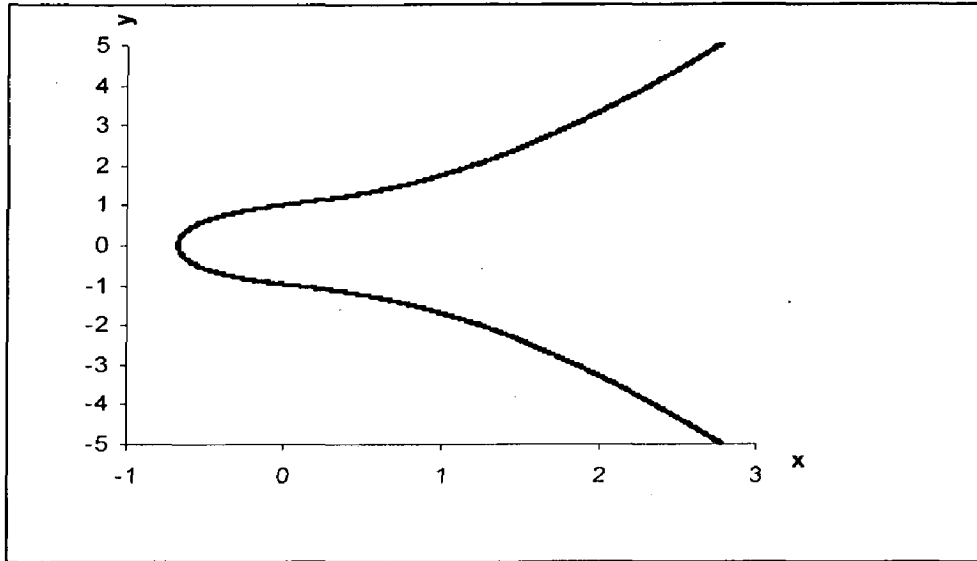


Figure 2.2:  $E(1,1): y^2 = x^3 + x + 1$

Consider a set of points  $E(a,b)$  which consists of points  $(x,y)$  that satisfy equation (2.3) for given  $a$  and  $b$ , together with a point  $O$  called as zero point. A group can be defined on this set provided following condition holds:

$$4a^3 + 27b^2 \neq 0 \quad (2.4)$$

The group operation is called as addition, which is denoted by '+'. The rules of addition are defined for adding two points of the group.

Elliptic curve cryptography uses curves in which the variables and coefficients are all restricted to elements of a finite field. Prime curves are the curves defined over  $\mathbb{Z}_p$  and binary curves are the curves which are defined over  $\text{GF}(2^n)$  [12]. According to [12], prime curves are best for software application.

Following table shows the points on elliptic curve  $E_{23}(1,1)$ .

Table 2.1 Points on elliptic curve  $E_{23}(1,1)$

(0,1)	(1,16)	(4,0)	(6,4)	(7,12)	(11,3)	(12,19)	(17,3)	(18,20)
(0,22)	(3,10)	(5,4)	(6,19)	(9,7)	(11,20)	(13,7)	(17,20)	(19,5)
(1,7)	(3,13)	(5,19)	(7,11)	(9,16)	(12,4)	(13,16)	(18,3)	(19,18)

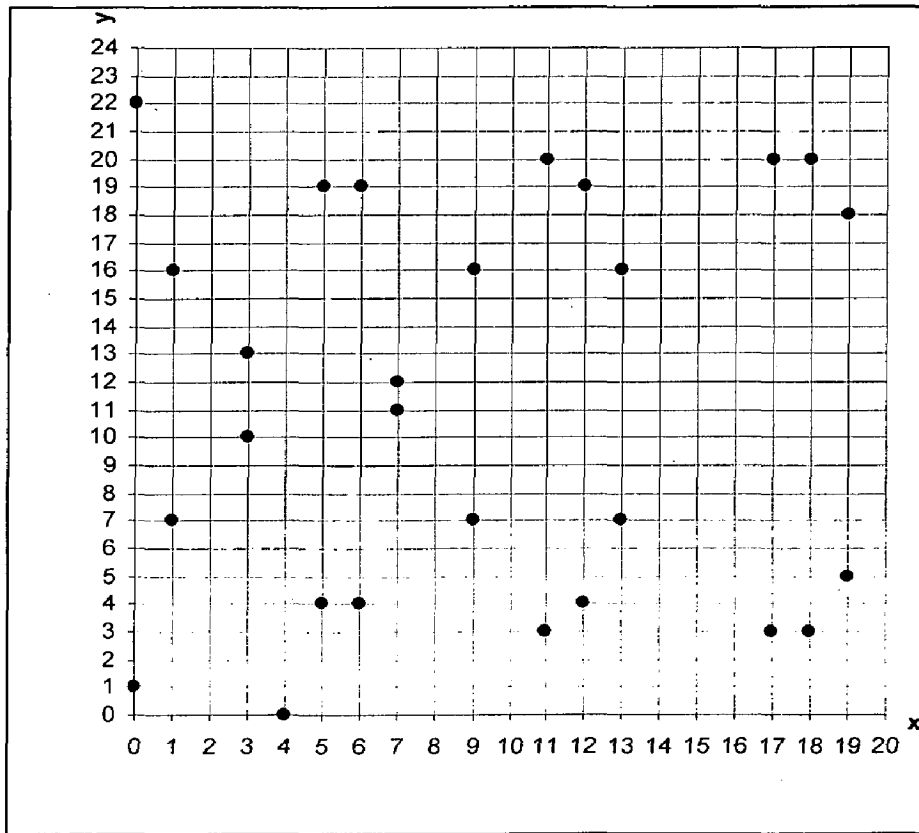


Figure 2.3 The elliptic curve  $E_{23}(1,1)$  [12]

Figure 2.3 shows the points from table 2.1 plotted in the x-y plane.

If  $P = (x_P, y_P)$ , then  $P + (x_P, -y_P) = O$ . Point  $(x_P, -y_P)$  is denoted as  $-P$ .

Let  $P = (x_P, y_P)$  and  $Q = (x_Q, y_Q)$  where  $P \neq -Q$ , then  $R = P + Q$ . Let  $R = (x_R, y_R)$ .

$$x_R = (\lambda^2 - x_P - x_Q) \bmod p \quad (2.5)$$

$$y_R = (\lambda (x_P - x_R) - y_P) \bmod p \quad (2.6)$$

where

$$\lambda = \begin{cases} \left\lfloor \frac{y_Q - y_P}{x_Q - x_P} \right\rfloor \bmod p & \text{if } P \neq Q \\ \left\lfloor \frac{3x_P^2 + a}{2y_P} \right\rfloor \bmod p & \text{if } P = Q \end{cases}$$

Multiplication operation is defined as the repeated addition.

Table 2.2 shows the correspondence between the  $Z_p^*$  and  $E(Z_p)$  [13].

Table 2.2 Correspondence between  $Z_p^*$  and  $E(Z_p)$

Group	$Z_p^*$	$E(Z_p)$
Group elements	Integers $\{1,2,\dots,p-1\}$	Points $(x,y)$ on E and O
Group Operation	Multiplication modulo p	Addition of points
Notation	Elements: g,h Multiplication: g.h Inverse: $g^{-1}$ Division: g/h Exponentiation: $g^a$	Elements: P, Q Addition: $P + Q$ Negative: -P Subtraction: $P - Q$ Multiplication: a P

Larger the number of points on the curve, more secure it is. According to [12] the number of points N, in the set  $E_p(a,b)$  is bounded by,

$$p + 1 - 2\sqrt{p} \leq N \leq p + 1 + 2\sqrt{p} \quad (2.7)$$

Table 2.3 shows the key sizes comparison between RSA and ECC. The keys are said to be equivalent from the security point of view[14].

Table 2.3 Equivalent key sizes in RSA and ECC

RSA	ECC
1024	160
2048	224
3072	256
7680	384

Elliptic curve cryptography finds its usage in Elliptic Curve Diffie Hellman Key Exchange, Elliptic Curve Digital Signature Algorithm and so on. Recently, applications based on bilinear mappings on various elliptic curve groups are being developed. These include identity-based encryption, key agreement etc. Elliptic curve cryptography has now being widely accepted as a part of well known standards like IEEE P1363 and PKCS13[12].

## 2.7 ECDL Problem and ECDLREP function

Elliptic curve discrete logarithm is the elliptic curve analog of the discrete logarithm problem. The Elliptic Curve Discrete Logarithm (ECDL) Problem is stated as follows: Given  $P, Q \in E_p(a,b)$ , determine  $k$ , where  $k < p$  and  $Q = kP$ , [10].

Public key cryptography over elliptic curves is based on ECDL problem. The public key is a point on the curve and the private key is a random number. The public key is obtained by multiplying the private key with the generator point  $G$  on the curve[15].

Elliptic Curve Discrete Logarithm Representation (ECDLREP) function, is now stated as follows: Let  $P_1, P_2, \dots, P_l$  be the generators of the group  $E_q(a,b)$ . Then

$$f(x_1, x_2, \dots, x_l) = P_1 x_1 + P_2 x_2 + \dots + P_l x_l \quad (2.8)$$

A variation of this function, which is used in the subsequent work, is as follows:

$$f(\alpha, x_1, x_2, \dots, x_l) = \alpha (P_1 x_1 + P_2 x_2 + \dots + P_l x_l) \quad (2.9)$$

## 2.8 Attacks on Discrete Logs and Elliptic Curve Discrete Logs

Discrete logarithm problem (DLP) being the base of many cryptographic schemes, security of these scheme depends on the hardness of DLP. This section explains in brief, various methods to solve discrete logarithm problem. In all of the following methods, it is assumed that  $x$  which is the discrete logarithm of  $\beta$  to the base  $\alpha$  is to be found.

Baby step giant step is a version of exhaustive search method to find discrete logarithm with a memory-time trade off [10]. It calculates  $m = \sqrt{n}$  where  $n$  is the order of  $\alpha$ . To find  $x = \log_{\alpha} \beta$ ,  $x$  is expressed as  $(im + j)$  where  $0 \leq i, j < m$ . So,  $\alpha^x = \alpha^{im} \alpha^j$ . Thus  $\beta (\alpha^{-m})^i = \alpha^j$ . A table consisting of entries  $(j, \alpha^j)$  is constructed. And the value of  $x$  is calculated.

In Pollard rho method, a sequence of  $x_0, x_1, \dots$  is calculated and a collision  $x_i = x_{2i}$  is searched. Each  $x_i$  in the sequence is equal to  $\alpha^{a_i} \beta^{b_i}$ . So when  $x_i = x_{2i}$ ,  $\alpha^{a_i} \beta^{b_i} = \alpha^{2a_i} \beta^{2b_i}$ . Hence  $x$  is computed as  $(a_{2i} - a_i) (b_{2i} - b_i)^{-1} \pmod n$ . It takes same expected running time as baby step giant step with the advantage of negligible amount of storage space.

Pohling Hellman algorithm for solving discrete logarithm holds when  $n$  is a composite number. Being a composite number, it can be expressed as the multiplication of powers of prime numbers. If  $n = p_1^{e_1} * p_2^{e_2} * \dots * p_r^{e_r}$ . To find  $x = \log_{\alpha} \beta$ , the approach is to determine  $x_i = x \pmod{p_i^{e_i}}$  where  $1 \leq i \leq r$ . Then Gauss's theorem[10] is used to determine  $x \pmod n$ .

As per[10], The Index-calculus method is a very effective method for calculating discrete logarithms. The method is not applicable to all the groups. But whenever applicable, it solves the problem in sub-exponential time. The method chooses a subset (called as factor base) of  $G$ . The subset is chosen in such a manner that maximum elements of group  $G$  can be expressed using the products of elements of this subset. Then a database which consists of logarithms of elements of factor base is constructed and this database is further used to calculate the discrete logarithm of the required element of  $G$ .

## CHAPTER 3

# LITERATURE REVIEW AND RELATED WORK

---

### 3.1 Untraceable Mails and Digital Pseudonyms

The very first idea about preserving privacy of users in electronic mail system was put forth by David Chaum in [16]. The paper presents a technique in which subjects (users) in an communication system can maintain anonymity of their identity as well as the confidentiality of the message. The technique provides these features without the presence of a trusted authority in the system. The system of untraceable emails requires a computer termed as 'Mix' which processes messages before passing them on to the receiver. The sending of messages through a mix takes place in the following manner.

If the user wants to send a message  $M$  to receiver  $A$ , he attaches a random message  $R_0$  to  $M$ . He then encrypts it using the public key of  $A$  that is  $K_A$ . He then attaches another random string  $R_1$  as well as the address of the receiver  $A$  to the resulting message and encrypts it with the public key of the Mix. So the input to the Mix becomes  $K_M(R_1, K_A(R_0, M), A)$ . The Mix then decrypts the message, removes the random string  $R_1$  and sends  $K_A(R_0, M)$  to the address  $A$ . So the following expression denotes the input and output of the Mix:

$$K_M(R_1, K_A(R_0, M), A) \rightarrow K_A(R_0, M), A$$

Figure 3.1 pictorially shows the working of a Mix.

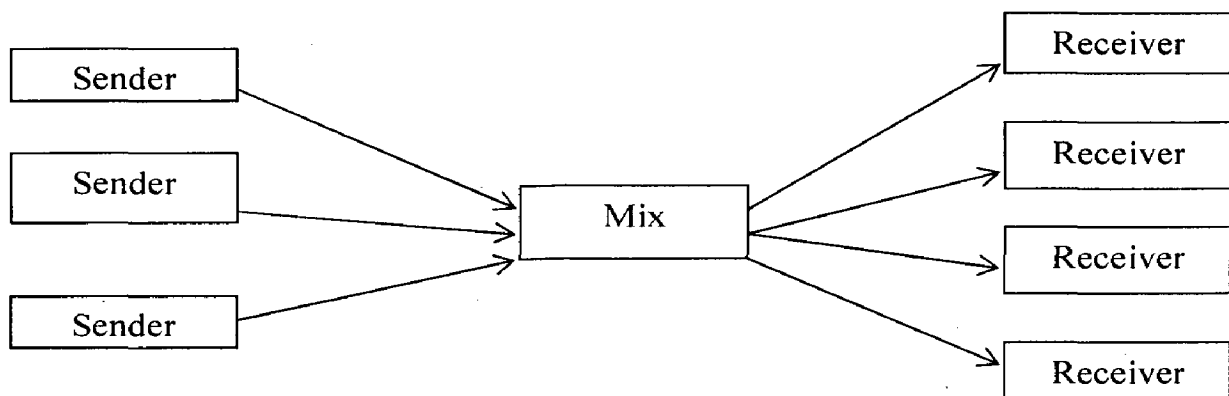


Figure 3.1 Mixer Network Proposed by Chaum



A number of mixes can also be used in a series to form a 'cascade'. It provides an advantage that any member of the cascade can provide the secrecy equivalent to that provided by the total cascade[16].

One time public key is used by the sender to create an encrypted address inside the message. This address is then used by the receiver of the message to respond to the sender anonymously.

This paper also suggests use of digital pseudonyms for the first time. It suggests that public keys are to be used as digital pseudonyms. It also states that the unlinkability of the pseudonyms with pseudonym holders is maintained with the help of a roster.

### **3.2 Pseudonym Systems**

Pseudonym systems[6] as suggested by Lysyanskaya et al. make use of digital pseudonyms as well as credentials to maintain the privacy of the user. The characteristic of this system is that the user maintains different pseudonyms with different organizations. Procedures such as master key generation of the user, key generation of the organization, pseudonym generation with an organization, communication with the organization, credential issue and credential transfer are defined.

Master key generation procedure helps user to generate a master key which he later uses in all the following procedures. Similarly each organization in the system also generates a key. Then the user creates a pseudonym with an organization through pseudonym generation procedure. In order to get a credential from an organization, it is necessary for the user to generate an pseudonym first. With the help of a pseudonym, the user authenticates himself and then gets issued a credential on this pseudonym. A user can also get a credential from an organization, if he shows a credential from other organization and proves that the two corresponding pseudonyms are issued to the same user (i.e. to himself).

The system ensures that each of the authenticated pseudonym belongs to a unique person. User's secret key can not be retrieved from his public key or from any of the above mentioned procedures. As per[6] two different pseudonyms, though issued to the same person, can not be linked to each other. In order to prevent sharing of the credential across users, the system ensures that a credential can not be transferred to another user without sharing the secret key of the user, which was used in the credential issue protocol. Further, the system also states that a credential can not be forged without organization's participation.

The system assumes the existence of one way functions. It's theoretical constructions are based on one way functions, bit commitments and zero knowledge proofs. A practical model based on discrete logarithm problem, Diffie Hellman problem is also provided.

### 3.3 Pseudonym Technology for E-services

Pseudonym Technology for E-services[1] discusses pseudonym technology from the privacy and security point of view. All the privacy and security related requirements of such technologies are thoroughly established. The paper first constructs a generalized pseudonym system architecture. Then the pseudonym technologies for services like e-cash, e-ticketing, e-voting etc. are reviewed taking into consideration the specific requirements of privacy in each of these systems. It introduces an application called as CAFÉ project based on e-wallet concept.

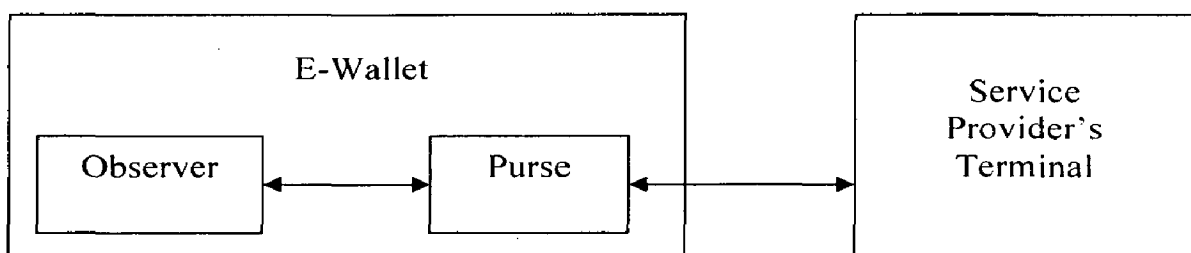


Figure 3.2 Chaum Pedersen E-wallet Architecture

Figure 3.2 shows Chaum- Pedersen's e-wallet architecture[1]. The concept of e-wallet suggests that organizations can store, read and update information in a tamper proof device issued to user i.e. the 'observer' shown in figure 3.2. This

device is issued by a bank and it contains e-coins issued by that bank. Purse is a device which is trusted by user. User puts the 'observer' device into the 'purse' device and then connects to the service providers terminal. The characteristic of the e-wallet system is that it facilitates offline payments. CAFÉ also uses cryptographic concepts like blind signature[17]. Apart from these features, it also maintains a mechanism for users' lost or stolen wallets.

### **3.4 Prime Project**

PRIME (Privacy and Identity Management for Europe) project [4] has implemented an integrated technical framework which processes personal data. The PRIME project anticipates a system in which people can use information services in a reliable way while keeping control over their personal details. Within the PRIME project, a system architecture has been proposed and a first prototype of the architecture has been implemented.

Some of the peculiar features that PRIME aims to provide are as follows:

- I. Privacy Negotiation:- It provides a provision for negotiating privacy between the user and his transaction partners. Together they come up with a policy which is agreed to both the parties.
- II. Anonymity Spectrum:- It offers different levels of anonymity to user according to the requirement of a transaction.
- III. Accountability:- A user can be made accountable to his actions even if he is anonymous in the transactions.

Apart from these characteristic features, it also provides identity management[18], data minimization etc. To provide these features, the architecture embodies various components like access control component, identity control component, obligation manager etc. The access control component ensures that access control policies are followed. The component facilitates attribute based access control system. The identity control component deals with credentials, communication with other parties and user interface. Obligation manager component triggers a specific workflow process defined by the obligation whenever appropriate triggering events occur[4].

Any transaction follows a sequence of two steps. In the first phase i.e. Negotiation phase, user first requests information about a service that he wants to avail. The service provider then comes up with the information and requirements of user's data. The identity control component at user's side then checks the necessary credentials of users and issues a list of user data and counter requirements. Once the service provider accepts the resulting agreement, the transaction proceeds to next step i.e. Contract Execution. In this step, service provider asks for the credentials of the users. User's identity component accesses the credentials and sends it back to the service provider. Service provider's identity component processes the data and stores the obligation concerning the data in the obligation manager. Thus PRIME handles the privacy and identity issues in an comprehensive fashion.

### 3.5 Privacy Enhancing Credentials

J. Nakazato et al. have presented a system based on elliptic curve cryptography which helps exchange a secret key between a user and a verifier for further anonymous communication [19]. The system is based on bilinear pairings. A pairing function is defined which maps values from an additive group to a value in a multiplicative group.

The pairing is denoted as  $e : G_1 * G_1 \rightarrow G_2$ , and it satisfies the following three properties:

- I. Bilinear:- Given any  $Q, R \in G_1$  and  $a, b \in Z_q$ , then  $e(aQ, bR) = e(Q,R)^{ab}$ .
- II. Non-degenerate:-  $e(P,P) \neq 1_{G_2}$  where  $P$  is the generator of  $G_1$
- III. Computable:- There is an efficient algorithm to compute  $e(Q,R)$  for any  $Q, R \in G_1$ .

The system makes use of public/private keys of user, verifiers and the server.

The entities in the proposed system are an authority, a server, a user, and verifiers. An authority is responsible for distributing the public/private key pairs. User is the one who wants to receive the service from service providers (verifiers in this case). Server acts as a trusted third party which is responsible for issuing a credential to

the user. The credential is then used for generating a ticket for a particular verifier. The ticket generated from such a credential is valid only for the communication between the user and the designated verifier. If the ticket is valid, the verifier executes a protocol to exchange keys with the user. This protocol generates the same session key at the end of user and that of verifier, which is further used for establishing a secure and authenticated channel. The system satisfies the security requirements such as unlinkability, unforgeability, limitability and non-transferability. Non-transferability in this protocols, refers to two cases:-

- I. A verifier who receives a ticket from an user, should not forge a valid ticket for another verifier.
- II. A user who is issued a credential for generating ticket, can not transfer the credential to another user without revealing the secret key.

### **3.6 Private Credentials**

The digital certificates used for authentication in these days enable the linkability of the different actions of a user. It leads to privacy eroding troubles such as traceability, discrimination, loss of user control over his data, erroneous user profiles and so on. Hence Zero knowledge Systems, Inc. has come up with an idea of private credentials[7].

Apart from the properties such as unlinkability, anonymity , user control, private credentials also support an attractive feature called as selective disclosure of the attributes. Also, private credential holders can demonstrate logical or arithmetic relations of the attributes of the credentials. The credentials can be refreshed without knowing the attribute it contains. Different CAs can certify different attributes of the same credential [7].

Two protocols namely issue protocol and show protocol are designed for the use of private credentials[11]. The constructs of the protocol are based on the DLREP function which has been explained thoroughly in section 2.5. One more important construct is that of proving knowledge of a DL-Representation of a value.

Suppose a user wants to prove a DL-representation  $(x_1, x_2, \dots, x_l)$  of  $h$  to another user, say Victor. Here,  $h = g_1^{x_1} * g_2^{x_2} * \dots * g_l^{x_l}$ . For proving knowledge of this representation

- I. The user chooses  $l$  random elements  $w_1, w_2, \dots, w_l$  in  $Z_q^*$ , calculates  $a = \text{SHA1}(g_1^{w_1} * g_2^{w_2} * \dots * g_l^{w_l})$
- II. The user then calculates  $c = \text{SHA1}(a, M) \bmod q$ .
- III. The user also computes  $r_i = c x_i + w_i$  for  $i = 0, 1, \dots, l$  and transmits  $a, r_i$  ( $1 \leq i \leq l$ ) to Victor.
- IV. Victor checks if  $a \stackrel{?}{=} \text{SHA1}(g_1^{r_1} * g_2^{r_2} * \dots * g_l^{r_l} * h^{-c})$

The concept of restrictive blind signature has been also used in the protocol. The protocol has been given in Appendix[1].

### 3.7 Proxy Blind Signature Scheme Based on DLP and ECDLP

In [20], a proxy signature method is proposed in which, as the name suggests, the authority of the signer is handed over to a proxy signer. Thus the proxy signer signs the message on behalf of the original signer. The concept of Blind Signature as suggested by Schnorr [21] is used in this scheme.

Blind signatures is a concept proposed by David Chaum in which the signer signs the blinded message[17]. It provides the unlinkability, which prevents the signer to link the signed message to the original message. Figure 3.3 shows the Schnorr's blind signature which requires 3 rounds of interaction between the signer and the user. Here  $p$  and  $q$  are prime numbers such that  $q$  is a factor of  $(p-1)$ . 'g' is the generator of the group.

The paper[20] presents the proxy signature scheme in DLP as well as in ECDLP. The scheme satisfies the properties of non-repudiation, unforgeability, verifiability and unlinkability. It also satisfies the property of distinguish-ability i.e. the proxy signature is distinguishable from the normal signature. It also claims that the ECDLP being harder than DLP, the signature in ECDLP has stronger security property. Further it requires less data size as the key size in ECDLP is much smaller than that in DLP.

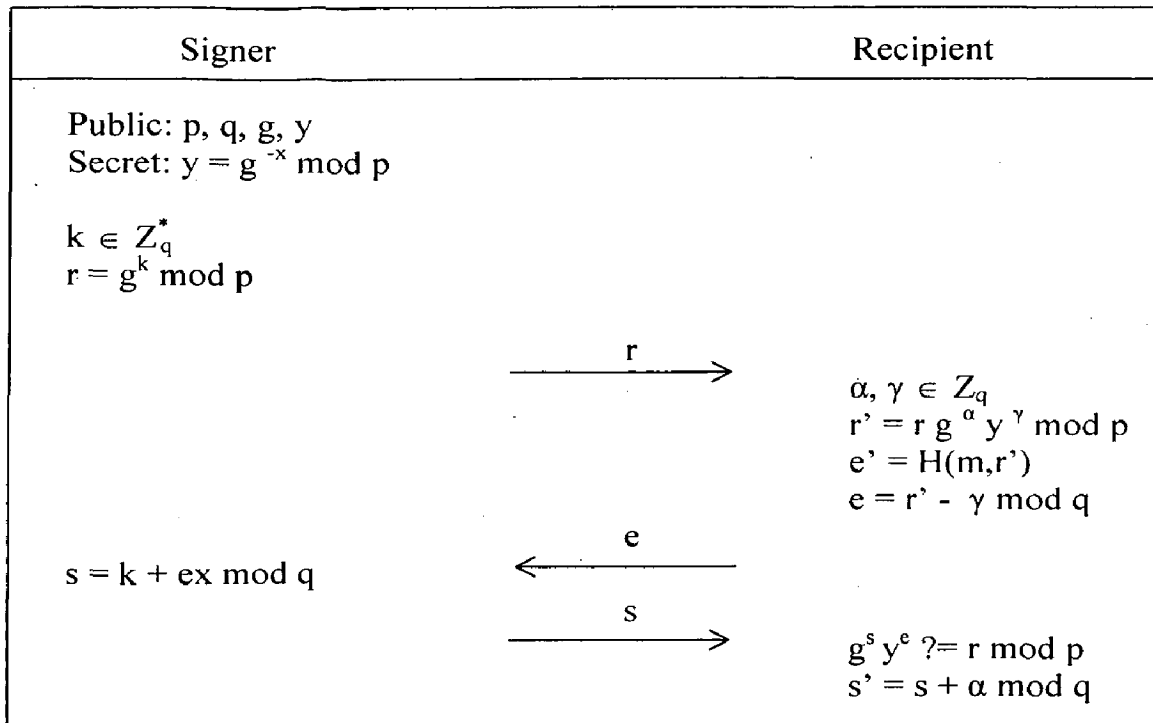


Figure 3.3 Blind Signature  $(r', s')$  on message  $m$

### 3.8 Research Gaps

Unlinkability of user's various actions with the service provider is one of the important feature of privacy preserving technologies. Unlinkability means that inability to link users attributes with the identity of the user. It also means the inability to link various protocols related to a particular credential. Various privacy preserving techniques which are mentioned in this chapter, fulfill the unlinkability property to different extents. Techniques like private credentials claim that even though the issue and show protocols of the same credential are linkable to each other, the linking of the attributes can not be traced back to the identity of the credential holder. In such cases, even though the identity of the individuals is not revealed, one the possible privacy threat namely discrimination of service is still valid for this anonymous profile.

Most of the privacy preserving technologies use various types of cryptographic techniques. These technique are based on the hard problems such as integer factorization problem, discrete logarithm problem, diffie hellman problem etc. All of these require very large key sizes to ensure proper security. Elliptic curve

cryptography which provides equivalent security in smaller key size has been rarely used in the privacy preserving technologies.

The work proposed in this report tries to address these research gaps.



## CHAPTER 4

# PROTOCOLS FOR PRIVACY PRESERVATION

---

### 4.1 Private Credential Protocols in Elliptic Curve Cryptography

As described in section 2.4, private credentials are an efficient means of achieving privacy. They support features of privacy like anonymity, unlinkability, selective disclosure and so on. Private credential protocols which are based on DLREP function, are given in Appendix[1]. This section redefines the protocols in terms of elliptic curve cryptography so as to deliver these features in an efficient manner.

In these protocols,  $E_q(a,b)$  is chosen as the elliptic curve where  $q$  is a prime.  $P$  is a point on the curve whose order  $n$  is a very large prime.

In this system, CA issues the private credentials to the users and user shows these credentials to the verifiers. Each CA has a public part and a private part. A private part consists of  $(y_0, y_2, \dots, y_{l-1})$  where  $y_i$  ( $0 \leq i \leq l-1$ ) are randomly selected from  $Z_n^*$ . It then publishes the value of the public part as  $(H_0, P_1, P_2, \dots, P_{l-1})$  where  $H_0 = y_0P$ ,  $P_1 = y_1P$ ,  $P_2 = y_2P$ ,  $\dots$ ,  $P_{l-1} = y_{l-1}P$

Figure 4.1 describes private credential issue protocol in terms of elliptic curve cryptography. ' $\alpha$ ' is the secret value of with the user. The protocol uses the variant of ECDLREP function, which is explained in section 2.7.

User chooses two random values  $a$  and  $b$  from  $Z_n^*$  and calculates  $H$ ,  $H'$  and  $B$ . He then sends the attributes  $(x_1, x_2, \dots, x_{l-1})$  that he wants to include in the private credential to CA. CA validates these attributes and if they are valid, it chooses a random element  $t$  from  $Z_n^*$  and sends  $T = tP$  to the user. User calculates  $R = B + T$ ,  $u' = \text{SHA1}(x(H') \parallel x(R)) \bmod n$  and  $u = u' - a \bmod n$  and sends  $u$  to CA. CA calculates  $v$  from  $u$  and sends it back to the user. After receiving  $v$  from CA, user calculates  $v' = (v + b) \alpha^{-1} \bmod n$ . Here  $(u', v')$  form the signature on  $H'$ . User verifies the validity of the signature by checking the equality of  $u'$  and  $\text{SHA1}(x(H') \parallel x(u'P + v'H')) \bmod n$ .

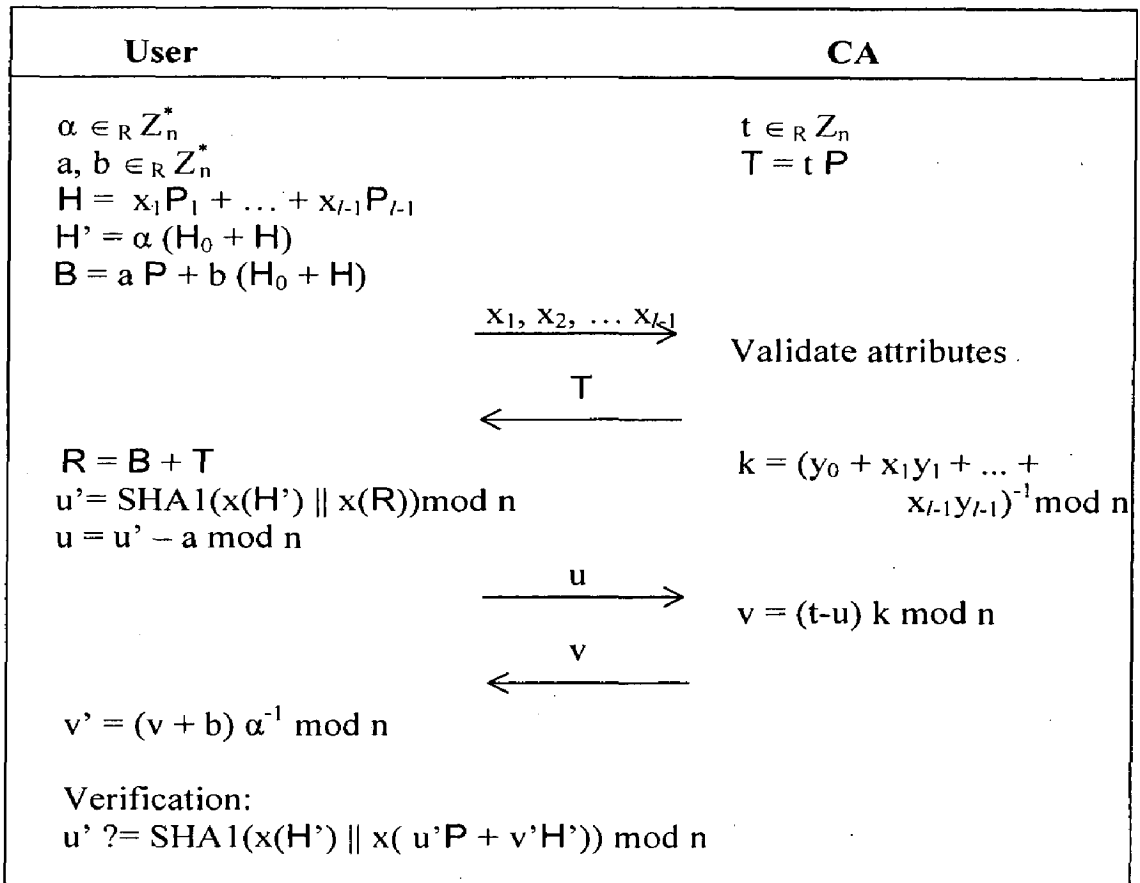


Figure 4.1 Private Credential Issue protocol (in ECC)

The equation for verification is

$$u' \stackrel{?}{=} \text{SHA1}(x(H') \parallel x(u' P + v' H')) \bmod n \quad (4.1)$$

$$\begin{aligned} \text{LHS} &= u' \\ &= \text{SHA1}(x(H') \parallel x(R)) \bmod n \end{aligned}$$

Hence equation (4.1) reduces to  $R \stackrel{?}{=} u' P + v' H'$

$$\begin{aligned} \text{LHS} &= R \\ &= B + T \\ &= a P + b (H_0 + H) + t P \end{aligned}$$

$$\begin{aligned} \text{RHS} &= u' P + v' H' \\ &= u' P + v' \alpha (H_0 + H) \\ &= u' P + (v + b) \alpha^{-1} \alpha (H_0 + H) \\ &= u' P + (v + b) (H_0 + H) \\ &= u' P + (v + b) (y_0 P + (x_1 y_1 + x_2 y_2 + \dots + x_{l-1} y_{l-1}) P) \end{aligned}$$

$$\begin{aligned}
&= u' P + ((t-u) k + b) (y_0 + x_1 y_1 + x_2 y_2 + \dots + x_{l-1} y_{l-1}) P \\
&= u' P + (t-u) P + (b) (y_0 + x_1 y_1 + x_2 y_2 + \dots + x_{l-1} y_{l-1}) P \\
&= u' P + (t-u) P + (b) (y_0 + x_1 y_1 + x_2 y_2 + \dots + x_{l-1} y_{l-1}) P \\
&= u' P + (t-u' + a) P + (b) (y_0 + x_1 y_1 + x_2 y_2 + \dots + x_{l-1} y_{l-1}) P \\
&= t P + a P + (b) (y_0 + x_1 y_1 + x_2 y_2 + \dots + x_{l-1} y_{l-1}) P \\
&= t P + a P + (b) (H_0 + H) \\
&= \text{RHS}
\end{aligned}$$

### Private Credentials Show Protocol

In this protocol, user proves the possession of proof of a private credential to the service provider or verifier[7]. This credential consists of two parts ( $H'$ ,  $u'$ ,  $v'$ ) which is sent to the verifier and a private part which is,  $(\alpha, x_1, x_2, \dots, x_{l-1})$ . After the completion of protocol, the verifier must be convinced that the user indeed possesses a valid private credential. As per the requirements of the verifier, the user should be able to disclose only those certified attributes that are relevant to the verifier.

Suppose  $l = 5$  and user knows a ECDL-representation of  $(\alpha, x_1, x_2, x_3, x_4)$  and the verifier needs him to reveal attributes  $x_1 (= 10)$ ,  $x_2 (= 12)$ . Then,

$$H' = \alpha (10 P_1 + 12 P_2 + x_3 P_3 + x_4 P_4 + H_0)$$

$$\text{i.e. } -(10 P_1 + 12 P_2 + H_0) = x_3 P_3 + x_4 P_4 - H'/\alpha$$

The user proves to verifier that he knows a ECDL representation  $(x_3, x_4, -1/\alpha)$  of  $-(10 P_1 + 12 P_2 + H_0)$ . Generalization of the representation shown above, requires to define the set called as *show*, which is a subset of indices  $\{1, 2, \dots, l-1\}$  and its complement as *hide*. So, in terms of the generalized terms, user proves to verifier that he knows a ECDL representation of the number  $-(H_0 + \sum_{i \in \text{show}} x_i P_i)$  where  $i \in \text{show}$  with respect to the generators  $(\{P_i\}_{i \in \text{hide}}, H')$  [11].

Message  $M$  contains a nonce to assure the freshness of the message. The nonce can be a random value or a timestamp. It should also contain the message  $m$  on which user desires a digital signature. It needs to be agreed in advance that exactly which information is to be included in  $M$ .

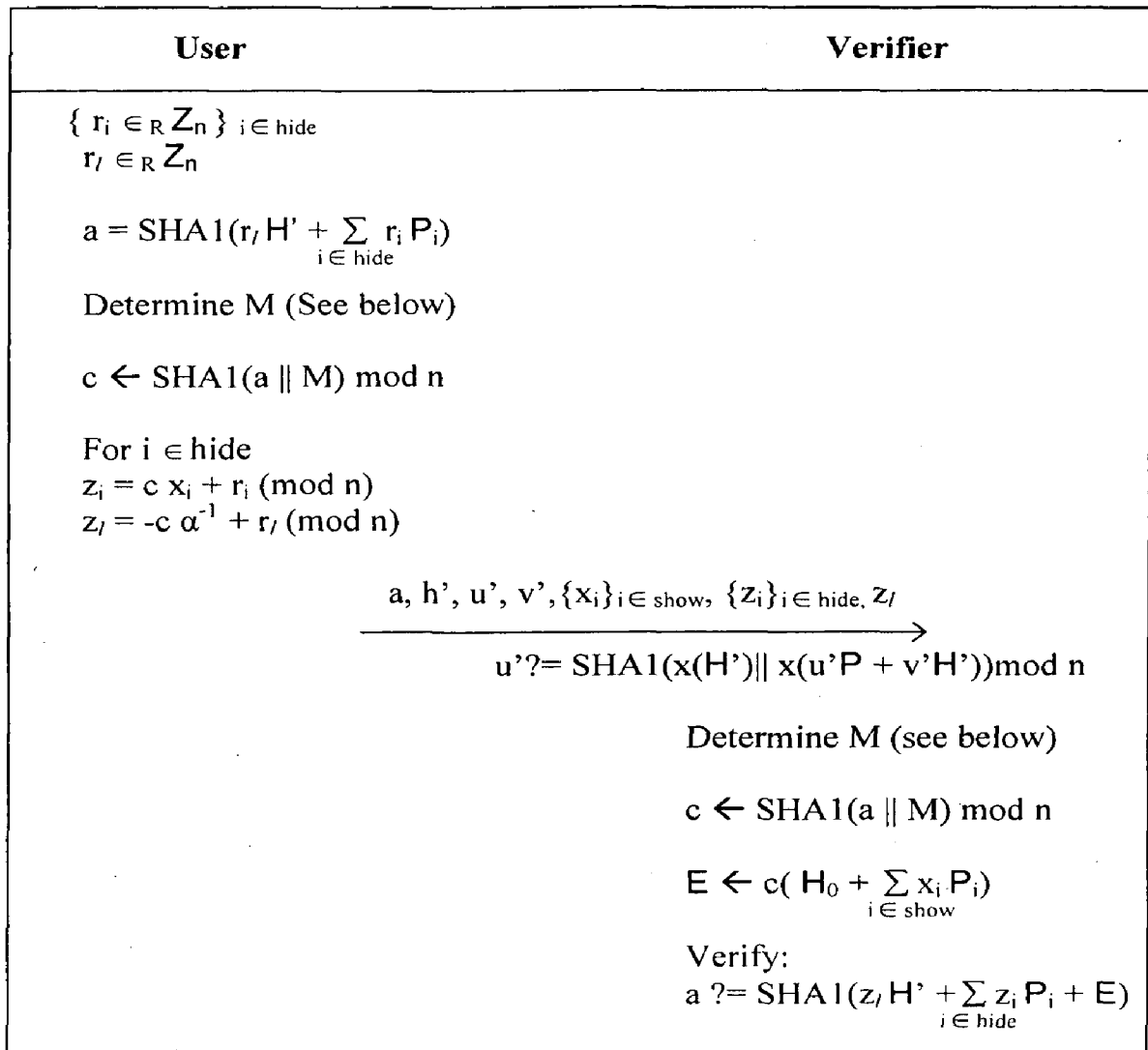


Figure 4.2 Private Credential Show Protocol (in ECC)

Verifier checks the validity of credential by verifying the equation. User then produces proof of knowledge of a particular ECDL-representation as explained above.

Verifier checks this by verifying the equation

$$a \stackrel{?}{=} \text{SHA1}(z_l H' + \sum_{i \in \text{hide}} z_i P_i + E) \tag{4.2}$$

$$\begin{aligned} \text{LHS} &= a \\ &= \text{SHA1}(r_l H' + \sum_{i \in \text{hide}} r_i P_i) \end{aligned}$$

$$\text{RHS} = \text{SHA1}(z_l H' + \sum_{i \in \text{hide}} z_i P_i + E)$$

Hence equation (4.2) reduces to

$$r_l H' + \sum_{i \in \text{hide}} r_i P_i \stackrel{?}{=} z_l H' + \sum_{i \in \text{hide}} z_i P_i + E$$

$$\text{LHS} = r_l H' + \sum_{i \in \text{hide}} r_i P_i$$

$$\begin{aligned} \text{RHS} &= z_l H' + \sum_{i \in \text{hide}} z_i P_i + E \\ &= (-c \alpha^{-1} + r_l) H' + \sum_{i \in \text{hide}} z_i P_i + E \\ &= r_l H' - c \alpha^{-1} H' + \sum_{i \in \text{hide}} z_i P_i + E \\ &= r_l H' - c \alpha^{-1} H' + \sum_{i \in \text{hide}} (c x_i + r_i) P_i + E \\ &= r_l H' - c \alpha^{-1} H' + c \sum_{i \in \text{hide}} x_i P_i + \sum_{i \in \text{hide}} r_i P_i + E \\ &= r_l H' + \sum_{i \in \text{hide}} r_i P_i + c (-\alpha^{-1} H' + \sum_{i \in \text{hide}} x_i P_i) + E \\ &= r_l H' + \sum_{i \in \text{hide}} r_i P_i + c (-\alpha^{-1} H' + \sum_{i \in \text{hide}} x_i P_i) + c(H_0 + \sum_{i \in \text{show}} x_i P_i) \\ &= r_l H' + \sum_{i \in \text{hide}} r_i P_i + c (-\alpha^{-1} (\alpha(H_0 + H))) + \sum_{i \in \text{hide}} x_i P_i + H_0 + \sum_{i \in \text{show}} x_i P_i \\ &= r_l H' + \sum_{i \in \text{hide}} r_i P_i + c (-H - H_0) + \sum_{i \in \text{hide}} x_i P_i + H_0 + \sum_{i \in \text{show}} x_i P_i \\ &= r_l H' + \sum_{i \in \text{hide}} r_i P_i + c (-H + \sum_{i \in \text{hide}} x_i P_i + \sum_{i \in \text{show}} x_i P_i) \\ &= r_l H' + \sum_{i \in \text{hide}} r_i P_i \\ &= \text{LHS} \end{aligned}$$

## 4.2 Subscription Credentials

A user has two types of relationships with the online organizations. One is for one-time transaction. E.g. When a user purchases a product from the organization, he never transacts with the same organization before or after the transaction. Another type of relationship is when the user subscribes for a service with the organization. In this, he is generally asked to register for the service. In the later stages, he can transact with the organization till the subscription expires.

Privacy preservation is comparatively easier in the former case, as one time credentials can be used for showing the necessary attributes of the user. For such type of transactions, private credentials[7] can be effectively used.

Some organizations offer services which are subscription based. E.g. a video online library may have a subscription worth a thousand dollars and a user may use it as and when he requires it. Every time he accesses the website, amount remaining in his account gets reduced depending on what all contents he has accessed in that session. For such types of relationships, the concept of subscription credentials has been presented here.

Various types of electronic payment systems are opted by service providers. One of the prevalent method amongst them is 'stored value payment system'. Such system enables users to make instant online payments to service providers or other individuals based on value stored in a digital account. These online systems rely on the value stored in a user's bank, or credit card account etc. Subscription credentials support such type of payment systems in a privacy preserving manner.

Using private credentials, user shows the necessary attributes to the service provider. On the basis of these attributes, subscription credential is issued to the user by the service provider. For the following transactions, this credential is shown to the service provider. Thus, the subscription credential is issued by as well as shown to the same organization. This credential is valid up-to a certain period or till a validating condition holds true. For this period, all the transactions hold the same session id. This session id is encoded in the subscription credential. The 'account balance' in the account of the subscription holder is also a field in the credential. Each time the user uses the service after showing the credential, the amount gets reduced. The new amount is encoded each time the user uses this credential. The 'access level' is the field which is used for the organization's authorization and access control policy. The organization also encodes an identifier which is peculiar to the organization's identity.

Thus the fields of the subscription credential are as follows:

***SubCred (SessionId, AccessLevel, ExpiryDate, AcntBal, OrgId)***

If the user wants to access the service and his subscription is over, he requires to undergo the issuing and showing protocol once again. Each time he acquires a new subscription credential a new session id is given to him.

### 4.3 Subscription Credential Issue Protocol

As explained in the previous section, subscription credentials are issued by the service provider. This protocol assumes that prior to the execution, the user has shown his credentials (preferably private credentials) to the service provider; and that the service provider has checked the attribute values. Hence in this protocol the message transfer starts from the service provider side. Depending on the attributes that the user possesses, the service provider decides the attributes values of subscription credential.

An elliptic curve  $E_q(a,b)$  is chosen. Let  $P$  be a point on the curve with a large order  $n$ , where  $n$  a prime. Service provider chooses  $y_1, y_2, \dots, y_l$  such that  $y_i \in \mathbb{Z}_n^*$  ( $1 \leq i \leq l$ ). This is service provider's secret part. He then publishes values  $P_1 = y_1P, P_2 = y_2P, \dots, P_l = y_lP$ .

Figure 4.3 shows the subscription credential issue protocol in terms of elliptic curve arithmetic.

Service provider chooses a random value  $k$  from  $\mathbb{Z}_n$  and performs a scalar multiplication  $S = kP$ . He also computes all the attributes of the subscription credential and sends all attributes ( $x_i$ 's) and  $S$  to the user. User calculates the  $H$  value as  $H = x_1P_1 + \dots + x_{l-1}P_{l-1}$  and  $H' = \alpha H$ . He then sends  $H'$  to the service provider. He also calculates value of  $M$  by adding  $H'$  and  $\alpha S$ . Service provider calculates  $u = \text{SHA1}(x(H'+T)) \bmod n$  and values of  $u'$  and  $R$ .  $(u', R)$  form the subscription credential for attributes  $x_1, x_2, \dots, x_l$ . To check the validity of the credential user verifies the equation  $u'H' = M + \alpha R$ . To verify the equation user needs his secret value  $\alpha$ .

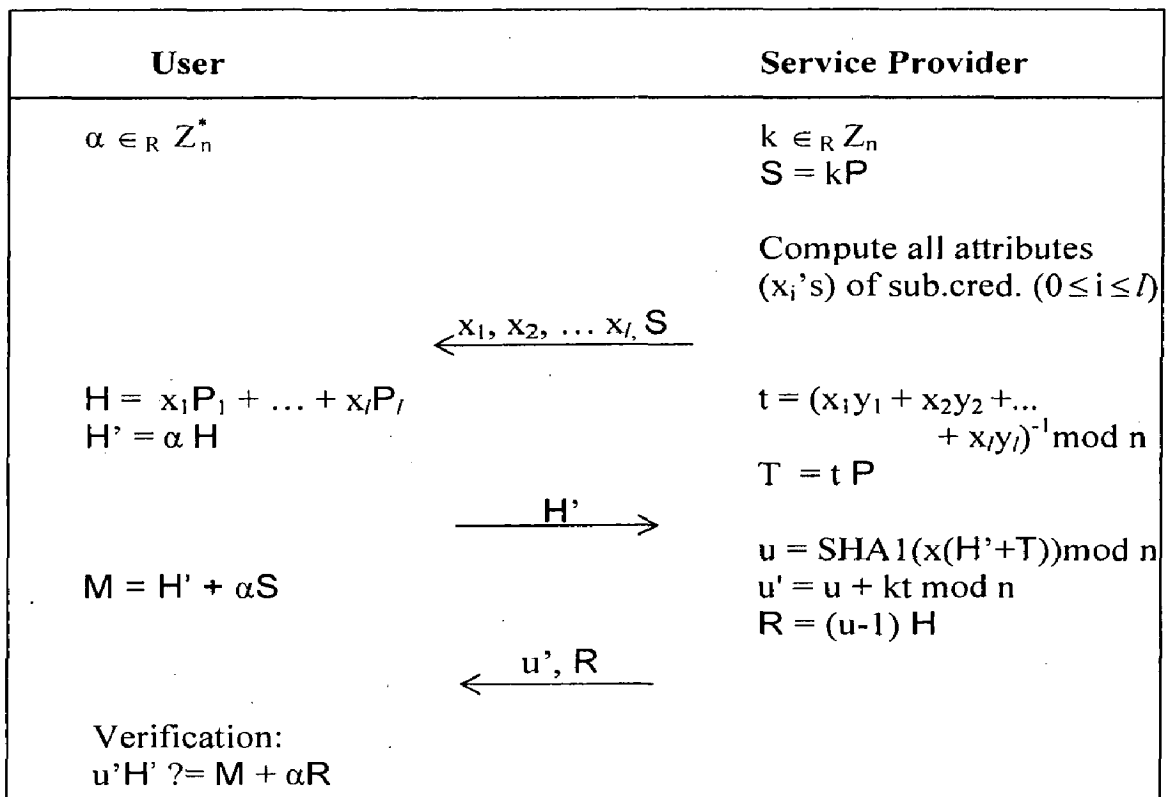


Figure 4.3 Subscription Credential Issue Protocol (in ECC)

$$\begin{aligned}
\text{LHS} &= u' H' \\
&= (u + kt) H' \\
&= u H' + kt H' \\
&= u H' + k\alpha H \\
&= u H' + k\alpha (x_1y_1 + x_2y_2 + \dots + x_ly_l)^{-1} H \\
&= u H' + k\alpha (x_1y_1 + x_2y_2 + \dots + x_ly_l)^{-1} H \\
&= u H' + \alpha k P \\
&= u H' + \alpha S
\end{aligned}$$

$$\begin{aligned}
\text{RHS} &= M + \alpha R \\
&= \alpha S + H' + \alpha R \\
&= \alpha S + H' + \alpha (u-1) H \\
&= \alpha S + \alpha H + \alpha (u-1) H \\
&= \alpha S + u H' \\
&= \text{LHS}
\end{aligned}$$

In this manner, equation is verified.



At the end of this protocol, user stores all the attributes of the credential, the credential  $(u', R)$  and the value of  $M$ . On the other hand, service provider stores the  $x_1$  attribute which is supposed to be the session id. This session id is later required for retrieving the records for the show and update protocol. Apart from that, service provider also stores the value of credential  $(u', R)$  for that session id. Value of  $H'$  is also stored at service provider's side.

Figure 4.4 shows the discrete-logarithm based counterpart of the subscription credential issue protocol.

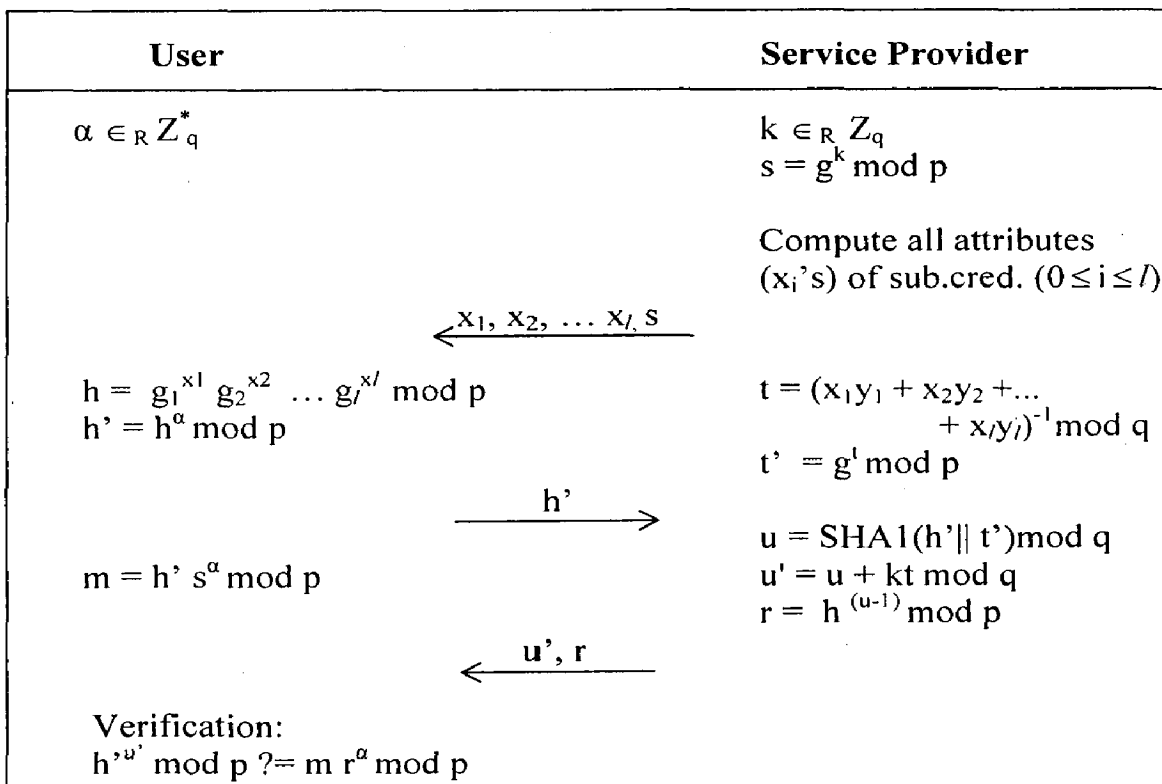


Figure 4.4. Subscription Credential Issue Protocol

#### 4.4 Subscription Credential Show and Update Protocol

In the show-and-update protocol of subscription credentials, user shows the attributes of the subscription credential to the service provider. He also shows the credential which is issued by the same service provider and proves its authenticity; i.e. he proves that he is actual user to whom this subscription credential is issued. In this protocol service provider verifies the authenticity of the credential, and renews some of the attributes of the credential. As this is the combined protocol for

showing the possession of credential as well as updating the credential, it is a two round protocol. In the first round of the protocol, user shows the attributes. He also sends some additional computed values, so that the credential is renewed. Then after verifying the authenticity of the credential, the service provider sends the updated credential back to the user. This protocol is shown in figure 4.5.

For simplicity, it is assumed that user knows that attribute  $x_2$  will be updated. If there are more than one attribute that will be updated in this protocol, generalization of this protocol is easily possible.

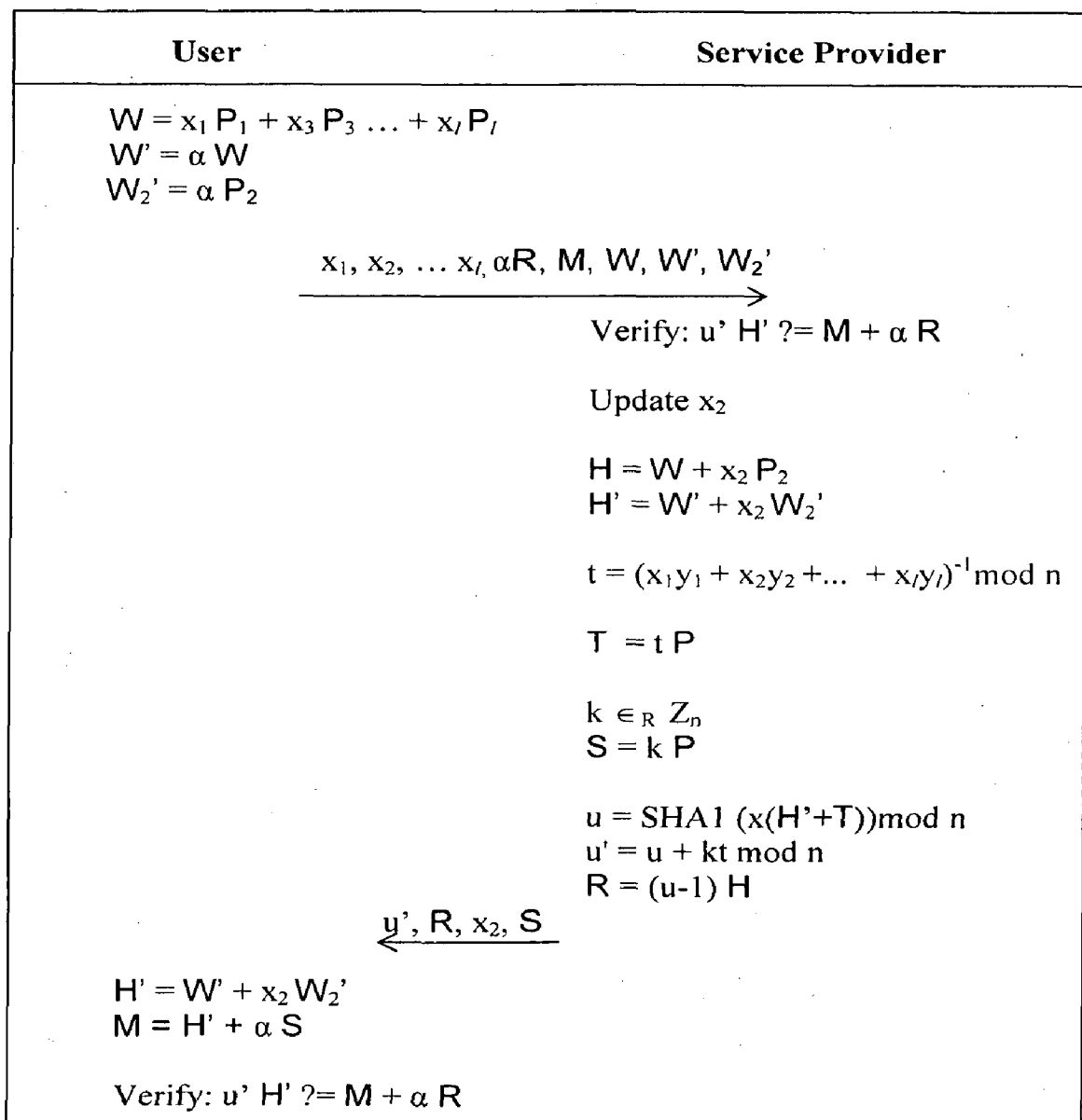


Figure 4.5 Subscription Credential Show-and-Update Protocol (in ECC)

User calculates  $W = x_1P_1 + x_3P_3 \dots + x_lP_l$  i.e. He calculates component analogous to H in the issue protocol except the difference of the factor  $x_2P_2$ . Note that here  $x_2$  is the attribute that is to be updated. Similarly he calculates  $W' = \alpha W$  which is similar to the value of H' and  $W_2' = \alpha P_2$ . He then sends all the attributes of the subscription credential, along with all W values (namely W, W' and  $W_2'$ ) to the service provider. He also sends the values of M and  $\alpha R$ . After receiving the values of all the attributes, service provider retrieves the data i.e. (u', H' and R) by matching the session id attribute of the subscription credential. To check the authenticity of the credential, service provider verifies the same equation  $u'H' \stackrel{?}{=} M + \alpha R$ .

Once the equation is verified, service provider updates the value of the attribute  $x_2$ . To compute values of H, he adds the value  $x_2P_2$  to W. Similarly H' is calculated as  $H' = W' + x_2W_2'$ . With the latest values of  $x_2$ , H and H', he further calculates values of T, S, u, u' and R in the same manner as calculated in the issue protocol. He then sends the updated credential (u',R) along with latest value of  $x_2$  and S. User computes  $M = H' + \alpha S$  and H'. To validate the credential value user checks the equation  $u'H' \stackrel{?}{=} M + \alpha R$ . Note that this protocol being a show-and-update protocol, updated credential is issued in the same format as issued in the issue protocol. The equation that is checked at service provider's side at the start of the protocol is same as the equation which is checked by the user at the end of the protocol.

Figure .4.6 shows the discrete-logarithm based counterpart of the subscription credential show and update protocol.

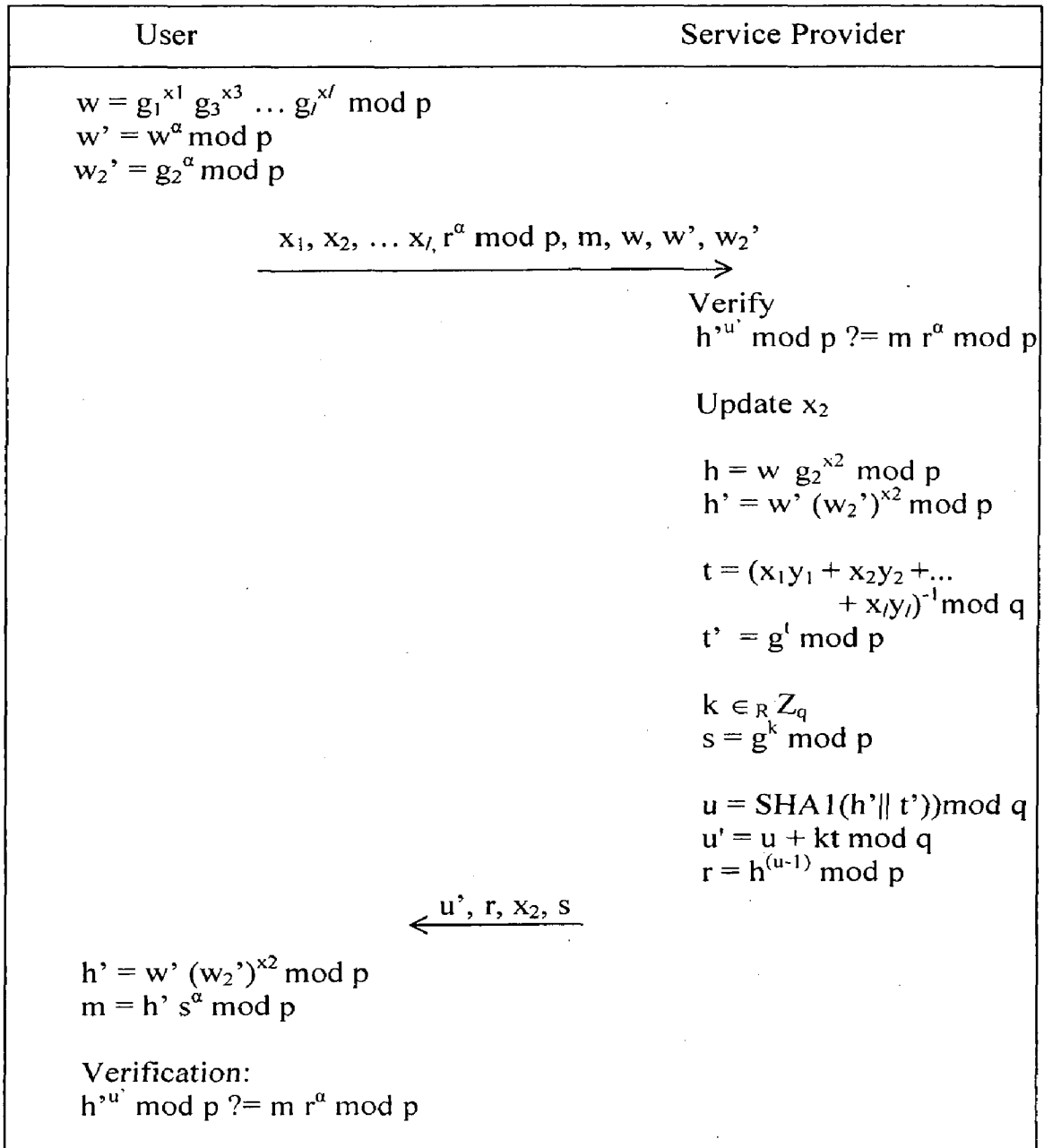


Figure 4.6 Subscription Credential Show-and-Update Protocol

## CHAPTER 5

# IMPLEMENTATION

---

### 5.1 Generation of Group Parameters

Discrete logarithm is defined in a set called as Group. In order to develop the protocols based on discrete logarithms, appropriate group parameters need to be generated. The protocols presented in the report are implemented in discrete logarithms as well as in terms of elliptic curve cryptography. The settings for the group are as follows.

Let  $p$  be a large prime number, for which  $p-1$  has a large prime factor  $q$ . In all the practical applications that use discrete logarithms,  $p$  is generally 1024 bits long.  $q$  is 160 bits long.  $Z_p^*$  denotes the set  $\{1, 2, \dots, p-1\}$ .  $G_{q,p}$  denotes a subgroup of  $Z_p^*$  of order  $q$ . All calculations involving elements of this subgroup are assumed to be mod  $q$ , and all calculations involving the field of exponents are assumed to be mod  $p$ . Let  $g$  denote the generator of the group; i.e. every element in the group can be generated from  $g$ .

This set up is similar to that used in DSA, i.e. Digital Signature Algorithm. Hence in order to generate parameters  $p$ ,  $q$  and  $g$  of required length, 'DSA' instance of 'KeyPairGenerator' class is used. This class provides methods to generate these parameters in an efficient manner.

### 5.2 Selection of Elliptic Curves

The security of a system which is based on elliptic curve depends on the underlying curve. There are two types of elliptic curves: Fixed and Random.

Generating a random curve is a very slow process. First we need to generate coefficients to make a valid elliptic curve, typically using a cryptographic random number generator. Then the number of points on the curve needs to be counted and factored. If one of the criteria is not satisfied, the values are discarded and we start again[22].

Using a published curve is the easiest and fastest method [22]. A possible concern in using such curve is that an attacker may be more motivated to attack a publicly known curve since more people would use it. The chance of collision is also increased. But according to [23], if 5 bits are added to the fixed curves, security equivalent to random curves is provided.

The curves specified in [24] use primes with fast modular reduction algorithms. To select the base point (Generator Point), a point with a large prime order  $n$  is chosen. The parameters in [24] also supplies a sample base point  $G = (G_x, G_y)$ .

Hence in this implementation, fixed curve as given in [24] is chosen as fixed curves. Though 5 bits for added security are not added in this implementation. The curve chosen for this implementation is given in Appendix[3].

### **5.3 Java Implementation**

The original protocol of private credentials[11] as well the proposed protocols of subscription credentials are implemented. Thus in all, there are four different protocols that have been implemented. They are as follows:

1. Private Credential Issue Protocol
2. Private Credential Show Protocol
3. Subscription Credential Issue Protocol
4. Subscription Credential Show and Update Protocol

Further all of these protocols are implemented in terms of discrete logarithms as well as in terms of elliptic curve cryptography. Thus in all, there are eight implementations.

#### **5.3.1 Platform**

The implementation has been done in Java language, on Windows XP platform. JDK 1.5 was used. Eclipse editor was used as an IDE. The reasons for choosing Java as the development language are twofold. As it will be discussed in the next section, the implementation requires client and server architecture. The attributes, credentials and intermediate computed values are to be passed between client and the server. Java supports socket programming. It has a vast variety of classes which

support socket programming and appropriate ones can be chosen to satisfy the program specific requirements.

Other reason is the support of cryptography. As all of these protocols are based on cryptographic constructs, it is necessary to have classes which support basic cryptographic operations. Java provides classes like BigInteger, DSAPrivateKey, DSAPublicKey, KeyPairGenerator etc. Some of the classes which are used repeatedly in the implementation are discussed briefly in section 5.3.3.

### 5.3.2 Socket Programming

The implementation of these protocols uses client server architecture. In the private credential issue protocol, user reveals the attributes to the Certification Authority and gets the credential from it. Hence the user acts as the client and CA acts as the server. Whereas in case of private credentials show protocol and subscription credential's protocols, user acts as the client and the service provider acts as the server. Table 5.1 clearly states the roles of user, CA and service provider in this architecture.

Table 5.1 Roles of User, CA, and Service Provider in Client –Server Architecture

Scheme	Private Credential		Subscription Credential	
	Issue	Show	Issue	Show-Update
Server	Certification Authority	Service Provider	Service Provider	Service Provider
Client	User	User	User	User

TCP provides a reliable, point-to-point communication channel between the client and the server. To communicate over TCP, a client program and a server program establish a connection with each other prior to the data transfer. Each program binds a socket to its end of connection. Then, the client and the server read from and write to the socket, bound to the TCP connection.

In order to pass the objects through the sockets, objects are needed to be serialized. Serialization is nothing but converting the objects into bit stream so that it can be transmitted over the network. Java provides an interface named as 'Serializable'. In order to pass an object across the connection, the class has to implement the Serializable interface. Then at the other end of the connection, the data stream is deserialized to convert it back to the object.

### 5.3.3 Special Classes

This section describes some of the Java classes that have been used in the implementation of the protocols. Some of the classes are in built-in classes that come with JDK, while some are developed particularly for this implementation.

**BigInteger.java :-** This is a built-in class in Java. This class is particularly designed for applications which require very large data sizes. It provides analogs to all of Java's primitive integer operators, and all relevant methods from `java.lang.Math`. Additionally, `BigInteger` provides operations for modular arithmetic, GCD calculation, primality testing, prime generation, bit manipulation, and a few other miscellaneous operations.

**Point.java :-** This is a user-defined class designed for this implementation. This class represents a point on the elliptic curve. As a point on the elliptic curve is defined by two large numbers  $x$  and  $y$ , this class contains two fields of the type `BigInteger`. Elliptic curve arithmetic like point addition, scalar multiplication are performed on the objects of the type `Point`.

**ECC.java:-** This class is also a user-defined class. Current versions of JDK (upto JDK 1.6) do not include any class which facilitates the arithmetic on an elliptic curve. Hence this class is developed. It provides methods for point addition, point doubling and scalar multiplication (i.e. multiplying a point with a `BigInteger`).

**Utils.java:** This is a user-defined class which provides generic methods for converting the hexadecimal numbers to decimal numbers.



Common.java: This class generates an instance of class ECC, which is used subsequently for all the protocols based on elliptic curve cryptography.

Apart from these classes, there are classes called as wrapper classes to transfer more than one objects in a single message.

### 5.3.4 Package Structure

Figure 5.1 shows the package structure that is used.

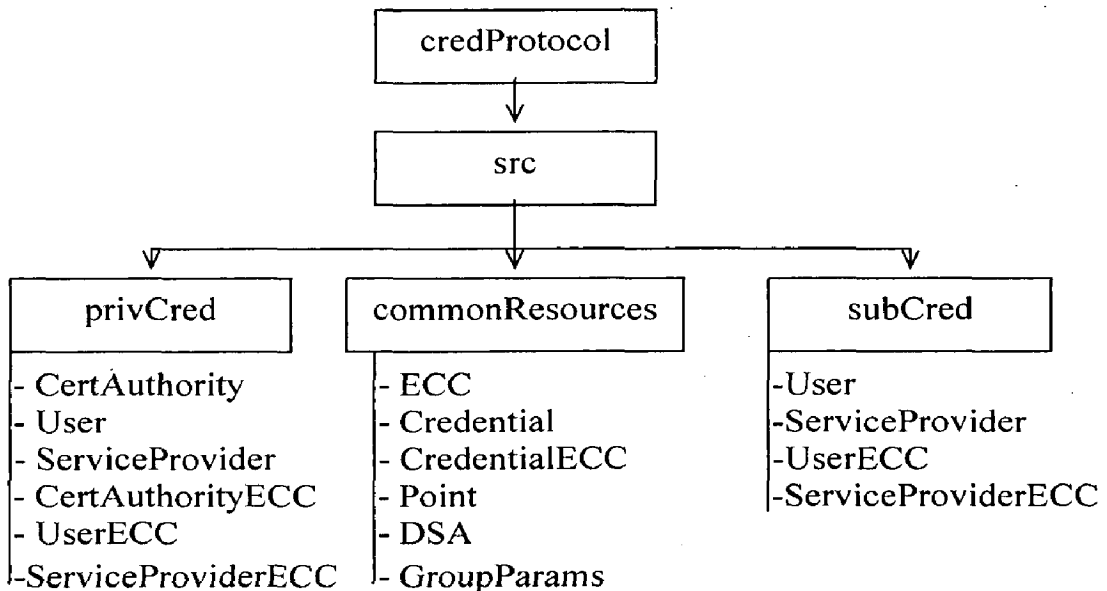


Figure 5.1 Java Package Structure

Apart from the classes shown here under the commonResources package, wrapper classes are also there in this package.

In the execution of private credentials' protocol, server i.e. Certification Authority Class is run. It instantiates its private part  $(y_1, y_2, \dots, y_l)$ , creates a server socket and waits for the client to approach it. The public part  $(P_1, P_2, \dots, P_l)$  is supposed to be published. This has been simulated through use of files. After starting the server, server writes the public values into a file. When client process (here 'User' class) is run, it reads those values from the file. Similarly, the records of users at service provider's side etc are written to and read from text files.

In case of subscription credential protocol, the server and the client remain same (service provider being the server and user being the client) in issue as well as show-and-update protocol. But two separate executions are required for running the issue and show-and-update protocol.

## CHAPTER 6

# RESULTS AND ANALYSIS

---

This section analyses the proposed scheme of subscription credentials from privacy and security point of view. It provides the property wise analysis of subscription credentials and explains how the presented scheme enhances the privacy and security of the private credentials.

### 6.1 Privacy Preservation Analysis

Proposed scheme of subscription credentials satisfies the following privacy related requirements.

- I. Anonymity:- The scheme assumes that prior to the execution of issue protocol, user has shown his attributes to the service provider using any private credentials. Once he shows those attributes, the subscription credential is issued to the user. Attributes of this credential are decided by the service provider itself. In all the later stages of the scheme (such as show and update protocol) user transacts with the service provider using by passing values such as attributes of the subscription credentials, values of  $H'$ , and the subscription credential etc. Hence, the user never reveals his identity to the service provider. So the anonymity of the user is never compromised in this scheme.
  
- II. Unlinkability:- Subscription credentials are valid till one of the following events occur: a) Their expiry date is passed. b) The account balance is zero. During this period the credential will have the same session id. Till this period the credential shows are linkable. After that if the user requires the service of the same service provider, he will be issued a new subscription credential bearing a new session id. So next times, the value of  $H'$  which is equal to  $(\alpha (x_1P_1 + \dots + x_lP_l))$  will be different. Even if the service provider stores the values of  $x_2, \dots, x_l$ , he will not be able to calculate  $H'$  as it is computed by user by multiplying  $H$  with secret value  $\alpha$ .

Further many users may have the same values of  $x_2, x_3, \dots, x_l$ . Thus even if the service provider stores the data about these individuals, mere values of these attributes can not be linked to a particular user.

- III. **Unforgeability of Credentials:-** Unforgeability of the subscription credential  $(u', R)$  may be attempted by a corrupt user. Here it is extremely difficult to forge a valid credential, as the attributes  $(x_1, x_2, \dots, x_l)$  of the subscription credentials are decided by the verifier himself. From the data of past transactions, if the user attempts to forge the credential, it will not be validated at the time of verification protocol. In the first step of verification protocol, the verifier checks the equation  $u'H' \stackrel{?}{=} M + \alpha R$ . The equation will not be verified as service provider will not have stored values  $(u', R, H')$  corresponding to values received from the user.
- IV. **Property Sharing Resistance:-** In the issuing protocol as shown in Fig 4.3, in order to verify the correctness of the credential, user requires the secret value  $\alpha$ . He can transfer the value  $\alpha R$  without revealing  $\alpha$  to other user. The value of  $\alpha R$  is not secret, as the accurate execution of the protocol will reveal its value. Because if the equation is valid, the value of  $\alpha R$  is same as value of  $(u'H' - M)$ .
- V. **Secrecy of User's Secret Value:-** The protocol is based on the generalized version of ECDLP. So it is very hard to know the value of  $\alpha$  even if the user transmits  $H'$  ( $=\alpha H$ ),  $W'$  ( $=\alpha W$ ) and  $W_2'$  ( $=\alpha P_2$ ) to the service provider. Thus the service provider will not know the value of user's secret alpha after executing the issue and show-and-update protocol with the user.

## 6.2 Security against Known Attacks

Section 2.8 described the methods that are used to solve the discrete logarithm problem in general. This section describes how the presented scheme of subscription credentials is resistant to attacks based on these methods.

- I. Pohling – Hellman Method:- The discrete logarithm and elliptic curve discrete logarithm problems are susceptible to this method if the 'n' is a composite number. In the proposed scheme, P is taken as a base point on the elliptic curve  $E_q(a,b)$  where order of P is a prime. So the proposed scheme is resistance to Pohling-Hellman method.
- II. Baby Step-Giant Step & Pollard-Rho Method:- Choosing a key of 160 bits makes it resistant to these types of attack [23].
- III. Index-calculus Method – This is the fastest possible general purpose algorithm to solve discrete logarithms[10]. But there is no known Index-calculus method for solving an ECDLP[23]. Thus the scheme presented here is immune to such kind of attack.

## CHAPTER 7

# CONCLUSION AND FUTURE WORK

---

### 7.1 Conclusion

After establishing the importance of privacy of user's personal data, various privacy preserving technologies are discussed in this report. One of the leading schemes among them is that of private credentials. Private credentials suffer from a specific type of linkability. Further an issuer can not encode attributes into the private credentials. To remove such disadvantages as well as to exploit all the exiting features of private credentials, a concept of subscription credentials has been proposed.

Subscription credentials are issued as well as verified by the same organization i.e. the service provider. Subscription credentials issue protocol assumes that the user has shown the necessary private credentials to the service provider. By maintaining the same basic set up such as nature of public and private part of the issuer of the credential, and same underlying mathematical properties, the scheme of subscription credentials is made compatible with that of private credentials.

Further, the protocol of subscription credentials is implemented using elliptic curve arithmetic. Elliptic curve cryptography provides equivalent security in smaller key size than its discrete logarithm or RSA based counterpart. Generators of the group are approximately equal in length to the key size. In the presented scheme, there are more than one generators. As the scheme is implemented using ECC, all of these will have smaller length, thus saving on memory. In addition to that, the scheme provides more security as it is based on ECDLP, and ECDLP is assumed to be harder than DLP.

Thus the presented scheme of subscription credentials using elliptic curve cryptography is enhanced from privacy perspective and at the same time it also provides increased security due to its underlying mathematical problem.

## **7.2 Future Work**

The proposed scheme of subscription credentials has a scope of improvement. One of the important operation over prime fields is that of scalar multiplication. This operation is considered to be equivalent to the modular exponentiation operation in discrete logarithm based systems. Time required for the binary scalar multiplication [Appendix 2] method used in this implementation depends upon the bit length of the multiplier. More sophisticated methods of scalar multiplication such as windows-w method using Non-Adjacent Form (NAF) are in place. These methods can be used to reduce the execution timing for subscription credentials.

## REFERENCES

- [1] R. Song, L. Korba, and G. Yee, "Pseudonym Technology for E-Services", In Privacy Protection for E-Services, Idea Group Inc., 2006.
- [2] Byron Braswell, "The Need for Authentication and Authorization", IBM Redbooks, 2003, <http://www.redbooks.ibm.com/abstracts/tips0266.html?Open> (Last accessed on 13 June 2009)
- [3] I. Goldberg, D. Wagner, and E. Brewer, "Privacy-Enhancing Technologies for the Internet", COMPCON'97, IEEE, 1997, pp. 103-109.
- [4] J. Camenisch, A. Shelat, D. Sommer, S. Fischer-Hubner, M. Hansen, H. Krasemann, G. Lacoste, R. Leenes, and J. Tseng, "Privacy and Identity Management for Everyone", DIM'05, ACM, 2005, pp. 20-27.
- [5] A. Pfitzmann and M. Hansen, "Anonymity, Unlinkability, Unobservability, Pseudonymity, and Identity Management – A Consolidated Proposal for Terminology", version v0.25, Dec. 6, 2005, [http://dud.inf.tu-dresden.de/Anon\\_Terminology.shtml](http://dud.inf.tu-dresden.de/Anon_Terminology.shtml) (Last accessed on 10 June 2009).
- [6] A. Lysyanskaya, R. Rivest, A. Sahai, and S. Wolf, "Pseudonym Systems", In Selected Areas in Cryptography, Springer Verlag, 1999, vol. 1758, Lecture Notes in Computer Science, pp. 184-199.
- [7] S. Brands, "Private Credentials", published by Zero Knowledge Systems Inc., November 2000.
- [8] J. Camenisch, and E. V. Herreweghen, "Design and Implementation of the *idemix* Anonymous Credential System", CCS'02, ACM, November 2002, pp. 21-30.



- [9] S. Farrell, "RFC 3281, An Internet Attribute Certificate Profile for Authorization" , April 2002, <http://www.ietf.org/rfc/rfc3281.txt> (Last accessed on 10 June 2009)
- [10] A. Menezes, P. Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, Published by CRC Press, June 1996.
- [11] A. Glenn, I. Goldberg, F. Legare, and A. Stiglic, "A Description of Protocols for Private Credentials", published by Zero Knowledge Systems Inc., June 2001.
- [12] W. Stallings, *Cryptography and Network Security – Principles and practise*, Published by Prentice Hall, 2006.
- [13] A. Jurisic, and A. Menezes, "Elliptic Curves and Cryptography", March 2005.
- [14] "SEC – 1: Elliptic Curve Cryptography", published by Certicom Corp., version 2.0, May 2009.
- [15] M. S. Anoop, "Elliptic Curve Cryptography: An Implementation Tutorial", published by Tata Elxsi Ltd, 2001.
- [16] D. Chaum, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms", *Communications of the ACM*, 1981, vol. 24, pp. 84-88.
- [17] D. Chaum, "Blind signatures for untraceable payments," *Advances in Cryptology- CRYPTO'82*, Springer Verlag, 1982, pp. 199- 203.
- [18] S. Gevers, K. Verslype, and B. Decker, "Enhancing Privacy in Identity Management Systems", *WPES, ACM*, October 2004, pp. 60-63.

- [19] J. Nakazato, L. Wang and A. Yamamura, "Privacy Enhancing Credentials", ASIAN 2007, Springer-Verlag, 2007, vol. 4846, Lecture Notes in Computer Science, pp. 55–61.
- [20] Z. Tan, Z. Liu, and C. Tang, "Digital Proxy Blind Signature Schemes based on DLP and ECDLP", MMRC, AMSS, December 2002, vol. 21, pp. 212-217.
- [21] M. Chang, I. Chen, I. Wu, and Y. Yeh, "Schnorr Blind Signature based on Elliptic Curve", Asian Journal of Information Technology, Grace Publications Network, 2003, vol. 2, pp. 130-134.
- [22] E. Yin, "Curve Selection in Elliptic Curve Cryptography", published by San Jose State University, 2005.
- [23] Y. Hitchcock, P. Montague, G. Carter, and E. Dawson, "The Security of Fixed versus Random Elliptic Curves in Cryptography", ACISP 2003, Springer Verlag, 2003, vol. 2727, Lecture Notes in Computer Science, pp. 55-66.
- [24] "SEC 2- Recommended Elliptic Curve Domain Parameters", published by Certicom Corp., version 1.0, September 2000.
- [25] K. D. Idoh, "Elliptic Curve Cryptography: Java Implementation", InfoSecCD Conference'04, ACM, 2004, pp. 86-93.

## **PUBLICATION:**

[1] Aditi Athavale, Kuldip Singh, Sandeep Sood, "Design of a Private Credentials Scheme based on Elliptic Curve Cryptography", CICSyN 2009, Indore, July 2009, (Accepted and to be published)

# APPENDIX

## 1. Private Credential Protocols

The following figure shows the private credential issue protocol as given in [11]

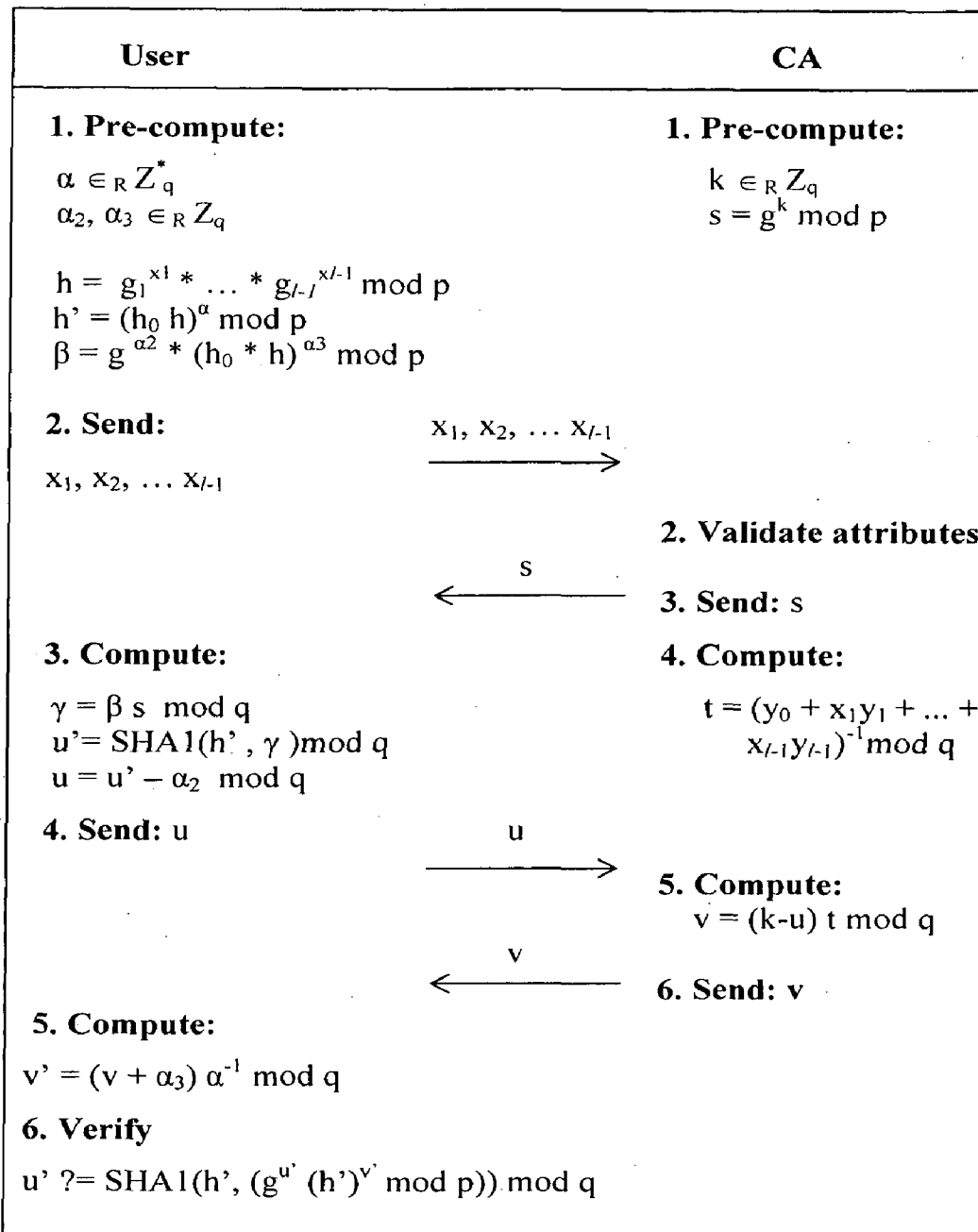


Figure 1: Private Credential Issue Protocol

The figure on the next page shows private credential show protocol.

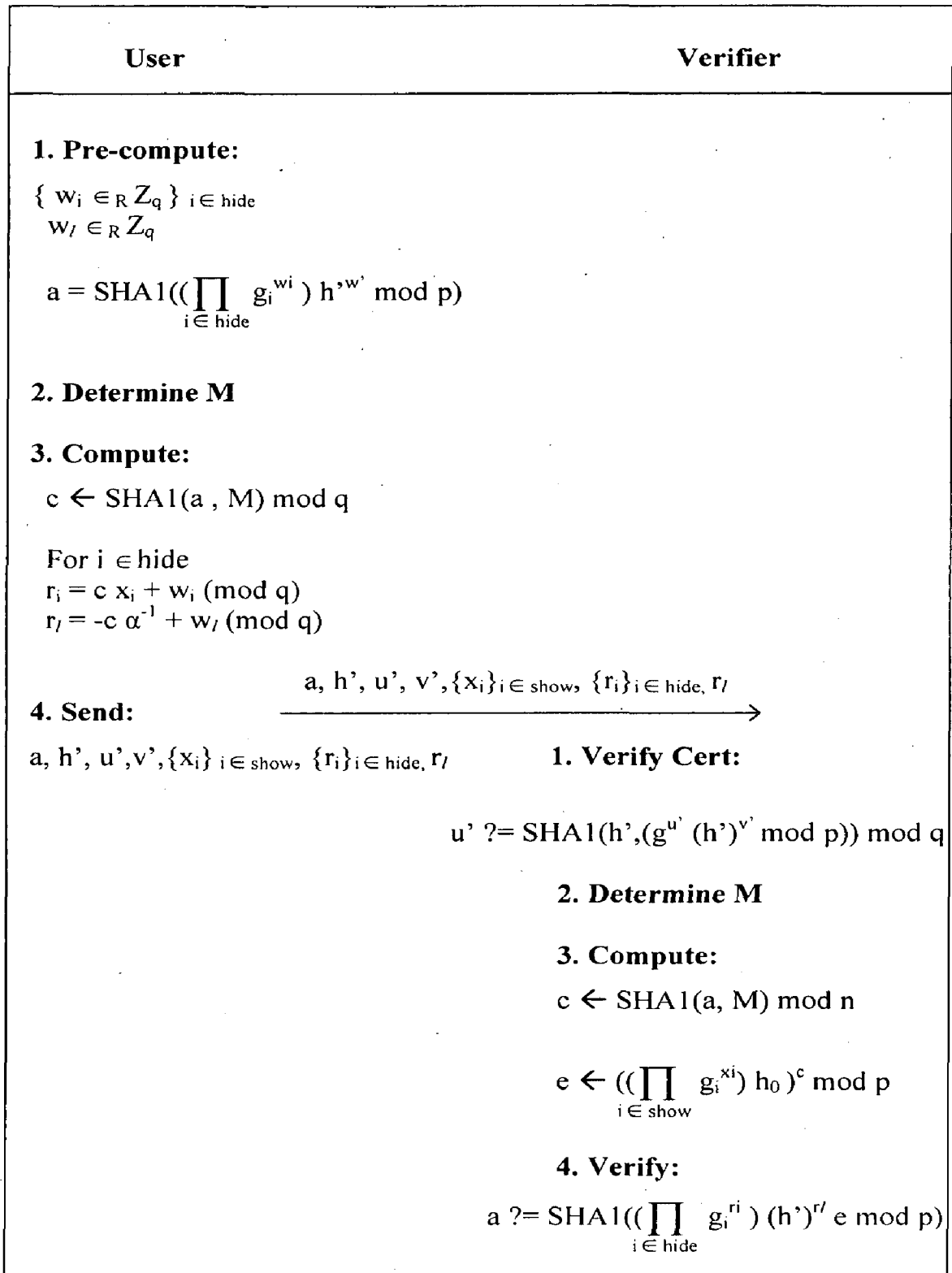


Figure II: Private Credential Show Protocol

## 2. Binary Scalar Multiplication Method used in the Implementation

This method is based on the binary representation of the multiplier  $k$ . It is expressed as

$$k = \sum_{j=0}^{i-1} k_j 2^j \quad \text{where } k_j = \{0,1\}$$

and

$$kP = \sum_{j=0}^{i-1} k_j 2^j P \quad \text{where } k_j = \{0,1\}$$

## 3. ECC Parameters by Certicom

Following are the ECC parameters for a 160-bit elliptic curve. These parameters are as given in ‘Standard for Efficient Cryptography’ by Certicom Corp. These parameters have been used in the implementation of ECC based protocols in this dissertation work.

$p =$  FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 7FFFFFFF

$a =$  FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 7FFFFFFC

$b =$  1C97BEFC 54BD7A8B 65ACF89F 81D4D4AD C565FA45

$n =$  1000000 00000000 0001F4C8 F927AED3 CA752257

$G_x =$  4A96B568 8EF57328 46646989 68C38BB9 13CBFC82

$G_y =$  23A62855 3168947D 59DCC912 04235137 7AC5FB32