A Strong proxy Signature Scheme with Partial Delegation

by

Pushpendra K. Mudgil-109CS0119 and Chandni Murmu-109CS0164

A thesis submitted in partial fulfillment for the
Degree of Bachelor of Technology in Computer Science and Engineering

under the guidance of

Prof. Sujata Mohanty
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA-769008

13th May 2013

Dedicated to the loved ones...

Certificate

This is to certify that the project entitled **"A STRONG PROXY SIGNATURE SCHEME WITH PARTIAL DELEGATION"** submitted by **Pushpendra Mudgil and Chandni Murmu** is an authentic work carried out by them under my supervision and guidance for the partial fulfillment of the requirements for the award of **Bachelor of Technoloy Degree** in **Computer Science and Engineering** at **National Institute of Technology Rourkela**.

To the best of my knowledge, the content of the project has not been submitted to any other Institute/University for the award of any Degree.

Date: **Prof. Sujata Mohanty**

Rourkela (National Institute of Technology, Rourkela)

Declaration of Authorship

We, Pushpendra Mudgil and Chandni Murmu, declare that this thesis titled, 'A STRONG PROXY SIGNATURE SCHEME WITH PARTIAL DELEGATION' and the work presented in it are our own. We confirm that:

- This work was done completely while in candidature for a B-Tech degree at this Institute.
- Where any portion of this thesis has previously been submitted for a degree or any other qualification at this Institute or any other University, this has been clearly stated in the references.
- Where we have consulted the published work of others, this is always clearly mentioned.
- Where we have quoted from the work of others, the source is also mentioned. This thesis is completely our own work with the exception of such quotations.
- We have acknowledged each and every main sources of help.

Signed: _____

Date: _____

Acknowledgements

"Tell me and I forget, teach me and I may remember, involve me and I learn."- Benjamin Franklin

We humbly express our gratitude to those who have contributed in the completion of this thesis. This thesis is an outcome of inspiring guidance of our advisor Prof. Sujata Mohanty.

Our batch mates have given us a lot of support and enthusiasm to grow intellectually and personally. We thank all the members of the Department of Computer Science and Engineering and the Institute who helped us in different ways in the completion of my work.

Our family for their love, encouragement and support.

And, we also thank God for everything.. . .

... Pushpendra Mudgil and Chandni Murmu

"-How long do you want these messages to remain secret?
+I want them to remain secret for as long as men are capable of evil."

Neal Stephenson, Cryptonomicon

Abstract

Proxy signature scheme is an extension of digital signature scheme first introduced by Mambo et al. in 1996, which allows a signer to delegate the signing capability to a designated person, called a proxy signer. There are three types of delegation, namely, full delegation, partial delegation, and delegation by warrant. In early proxy signature schemes, the identity of the proxy signer can be revealed by any trusted authority if needed. However, a secured proxy signature scheme must satisfy various properties, such as, verifiability, strong unforgeability, nonrepudiation, privacy, and strong identifiability.

In this thesis, we propose a strong proxy signature scheme based on two computationally hard assumptions, namely, Discrete Logarithmic Problem (DLP) and Computational Diffie-Helmann (CDH) problem, which satisfies all the security properties of a standard proxy signature scheme. The property 'strong' refers to the fact that only a designated person can only verify the authenticity of the proxy signature. No one, not even the original signer can verify the signature. The proposed scheme is based on partial delegation, in which a new proxy signing key is generated by the secret key of original signer. Also we compared the performance of the proposed scheme in terms of signature length, computational overhead and execution time with a popular scheme and found that our scheme has less computational overhead and of less signature length. Moreover, our scheme is proved to be secure against some active attacks.

The proposed scheme has wide applications in areas such as e-voting, e-commerce, secure transaction and e-cash.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A proxy signature permits a delegator to give partial signing rights to other parties called proxy signers. In other words, Proxy signature is a digital signature where an original signer delegates her signing power to a proxy signer, and then the proxy signer signs the message on behalf of the original signer. For example, a company's manager wants to go for a long trip. She would need an agent called a proxy agent, to whom she would assign her signing capability, and after the delgation,i.e. power assignment, the proxy agent would sign the documents on behalf of the manager. It has been 18 years since the notion of proxy signature was first introduced. However, the cryptographic treatment on proxy signature was introduced by Mambo et al. in 1996 [1][2].

1.1 Properties of proxy signature

Proxy signature is popular and is used widely because of its security properties. The security properties of proxy signature are [2]:

- **Verifiability:** From a proxy signature a verifier can be convinced of the original signers agreement on the signed message.
- **Strong unforgeability:** A valid proxy signature can only be generated by the designated proxy signer.
- **Strong non-repudiation:** A proxy signer cannot deny a valid proxy signature he/she generates.

- **Non-designated:** The warrant issued by the original signer does not specify who the proxy signer is. It is also transferable among proxy signers.
- **Strong identifiability:** From a proxy signature, any verifier can determine the identity of the proxy signer.
- **Proxy privacy:** No one can determine the identity of the proxy signer only from the proxy signature.
- **Privacy revocation:** Once needed, a trusted authority can reveal the proxy signers identity of the proxy signature.

1.2 Categories of proxy signature

Proxy signature has been classified into three broad categories. They are [2][3]:

- full delegation
- partial delegation
- delegation with warrant

1.2.1 Full Delegation

In proxy signature with full delegation, an original signer gives her private key to a proxy signer and the proxy signer using original signers private key signs document. The drawback of proxy signature with full delegation is that the original signer and proxy signer are very difficult to distinguish from each other.

1.2.2 Partial Delegation

In partial delegation proxy signature, the original signer derives a proxy key from her private key and hands it over to the proxy signer as a delegation capability. In proxy signature with partial delegation, the proxy signer can misuse the delegation capability, because partial delegation cannot restrict the proxy signers signing capability.

1.2.3 Delegation with warrant

The drawbacks of full delegation and partial delegation are eliminated by partial delegation with warrant. A warrant explicitly states the identity of signers, period of delegation and the qualification of messages on which the proxy signer can sign, etc. In other words, the warrant is used to certify that the proxy signer is really authorized by the original signer.

1.3 Related Work

The concept of proxy signatures was first proposed by Mambo et al. in 1996 [1]. He said that a proxy signature scheme allows a signer to delegate the signing capability to a designated person and the designated person was called a proxy signer. Lee et al. constructed a strong non-designated proxy signature scheme in 2001 [2]. The concept used in non-designated proxy signature scheme was that the original signer does not specify his/her proxy signer in proxy key issuing phase. Anyone can construct original signers proxy signing key if he/she owns the warrant and some secret parameters issued by the original signer. Then, it can be used by he/she to sign messages on behalf of the original signer. In the non-designated proxy signature scheme, the warrant and secret parameters are transferable among the proxy signers.

In 2002, Shum and Wei presented an enhancement to the Lee et al.s scheme. In their scheme they have tried to hide the identity of the proxy signer. The identity of the proxy signer cannot be determined by anyone from the proxy signature only. However, a trusted authority can reveal the proxy signers identity if required [2]. In 2005, Narn-Yih Lee and Ming-Feng Lee, showed that the ShumWei scheme cannot keep the property of the strong unforgeability[2], i.e both original signer and proxy signature can generate valid proxy signatures. In 2006, Huang et al. proposed the first proxy signature scheme in the standard model and following them other schemes, such as, the Yu et al.'s designated verifier proxy signature scheme were proposed [4].

In 2007, Kemal Bicakci presented a simple alternative approach that eliminates public-key cryptography in key generation, offers certainty and simplicity in the dispute resolution and avoids swallow attacks. They also introduce the concept of 1-out-of-n threshold traceable one-time signatures as an efficiency improvement [8]. In 2009, Liu Zhen-hua1, Hu Yu-pu, Zhang Xiang-song and Ma Hua gave a

security model of proxy signature schemes with fast revocation is formalized [7]. In 2011, Ying Sun, Chunxiang Xu, Yong Yu, Yi Mu proposed a new construction of proxy signature which is strongly unforgeable in the standard model with the computational DiffieHellman assumption in bilinear groups [6]. In 2012, Zhang Jian-hong, Xu Yu-wei, Cui Yuan-bo and Chen Zhi-peng have suggested a novel short proxy signature scheme [5].

1.4 Motivation

Unforgeability means that only the designated proxy signer can generate a valid proxy signature.In our literary survey we found that the property of non-forgeability was not satisfied in terms of security. Also, the length of proxy signature is large and has high communicational overhead.This motivated us to design a secure proxy scheme which would overcome this drawback which was found in many existing papers. It was also observed in our literary survey that a malicious original signer is able to generate a valid proxy signature by himself/herself without delegating the signing capability to any proxy signer.

1.5 Objective

The objective of our scheme is to design a Strong proxy signature scheme with partial delegation holding properties such as verifiability, non-repudiation, non-designated, proxy privacy and aims to achieve low computation and communication overhead and short signature length. We will be emphasizing more on overcoming the security flaw which was seen in many schemes [2]. The objective behind the project is also to produce a strong signature of short length with less computational overhead. Here, the property 'strong' refers to the fact that only a designated person can verify the authenticity of the proxy signature. No one, not even the original signer can verify the signature.

1.6 Thesis Outline

This thesis is organized as follows: Chapter 2 discusses the preliminaries, the proposed scheme is discussed in Chapter 3, Chapter 4 shows the implementation of the proposed scheme in which we will discuss the security analysis, comparative

performance evaluation and results of implementation. Finally, we conclude with Chapter 5 and give few future directions of our work.

Chapter 2

Preliminaries

We will be discussing few of the preliminaries which we have used thoughout our project work.

2.1 Discrete Logarithmic Problem (DLP)

The multiplicative subgroup of any finite field GF(q) is cyclic where q is a prime power, and the elements g $\in GF(q)$ that generate this subgroup are referred to as primitive elements[9]. When a primitive element g $\in GF(q)$ and any u $\in GF(q)^* = GF(q) - \{0\}$ is given, the discrete logarithm of u with respect to g is that integer k, $0 \leq k \leq (q-1)$, for which

$$u = g^k \tag{2.1.1}$$

It will be written as k = $\log_g u$. The discrete logarithm of u is sometimes called as the index of u. Finding the value of k is very difficult [9].

Besides the intrinsic interest that the problem of computing discrete logarithms has, discrete logarithm is of considerable importance in cryptography. An efficient algorithm for discrete logarithms would make a large number of authentication and key-exchange systems insecure.

There are many proposed algorithms for computing discrete logarithms which are known today. Among them index-calculus algorithm is the most powerful general purpose algorithm.

2.2 SHA-1 Hash Approach

SHA1 is an abbreviated form of Secure Hashing Algorithm. SHA-1 is a hashing algorithm designed and constructed by the United States National Security Agency and published by NIST. It is the improved version of the original SHA-0 and was first published in 1995. Although SHA-1 will soon be replaced by the newer and potentially more secure SHA-2 family of hashing functions, currently the most widely used SHA hash function is SHA-1. It is currently being used in a large number of applications, including TLS, SSL, SSH and PGP.

The output of SHA-1 is a 160 bit digest of any sized file or input. In structure it is similar to the previous MD4 and MD5 hash functions; in fact it shares some of the initial hash values. It uses a block size of 512 bit and has a maximum message size of 264 - 1 bits. By implementing SHA-1, we can compare implementations of cryptographic functions with specifications. If we ever need to verify that an existing implementation of a cryptographic function is secure this could be useful. The performance of the code can be optimized by running time profiles [10].

2.3 Integer Factorization Problem

There exist a variety of factorizing algorithms such as trial division, Fermat factorization, Pollard rho factorization, Brent's factorization method, Pollard p - 1 factorization, etc. **Fundamental Theorem of Arithmetic :** The fundamental theorem of arithmetic states that every positive integer can be written uniquely as a product of primes, when the primes in the product are written in non-decreasing order [14],i.e the fundamental theorem of arithmetic means that any composite integer can be factored.

If two large prime numbers are given, there are fast algorithms for multiplying them together. However, it is difficult to find the prime factors if one is given the product of two large primes. The apparent difficulty of factoring large integers forms the basis of some modern cryptographic algorithms. The RSA encryption algorithm [11], and the Blum Blum Shub cryptographic pseudorandom number generator [12] both rely on the difficulty of factoring large integers. If it were possible to factor products of large prime numbers quickly, these algorithms would be insecure.

The SSL encryption used for TCP/IP connections over the World Wide Web depends on the security of the RSA algorithm [13]. Hence if one could factor large integers quickly, "secured" Internet sites would no longer remain secure. It is unknown whether factoring is in the complexity class P in computational complexity theory. In technical terms, this means that there is no known algorithm for answering the question whether integer N have a factor less than integer s in a number of steps that is O(P(n)), where n is the number of digits in N, and P(n) is a polynomial function. Above all, no one has ever proved that such an algorithm exists, or does not exist. In layman's terms, one can simply ask the question what is the fastest algorithm for factoring large numbers. This is an important open question in mathematics.

Chapter 3

The proposed Strong proxy signature scheme based on partial delegation

The proposed scheme is a work undertaken to overcome the shortcomings of the scheme given by Narn-Yih Lee, Ming-Feng Lee (2005) [2]. Our proposed scheme focuses on the following:

- identifiability
- low computational and communicational overhead
- short signature length
- non-repudiation
- verifiability
- non-designated
- proxy privacy
- unforgeability

3.1 Layout of the proposed scheme

The proposed scheme consists of four phases. Namely,

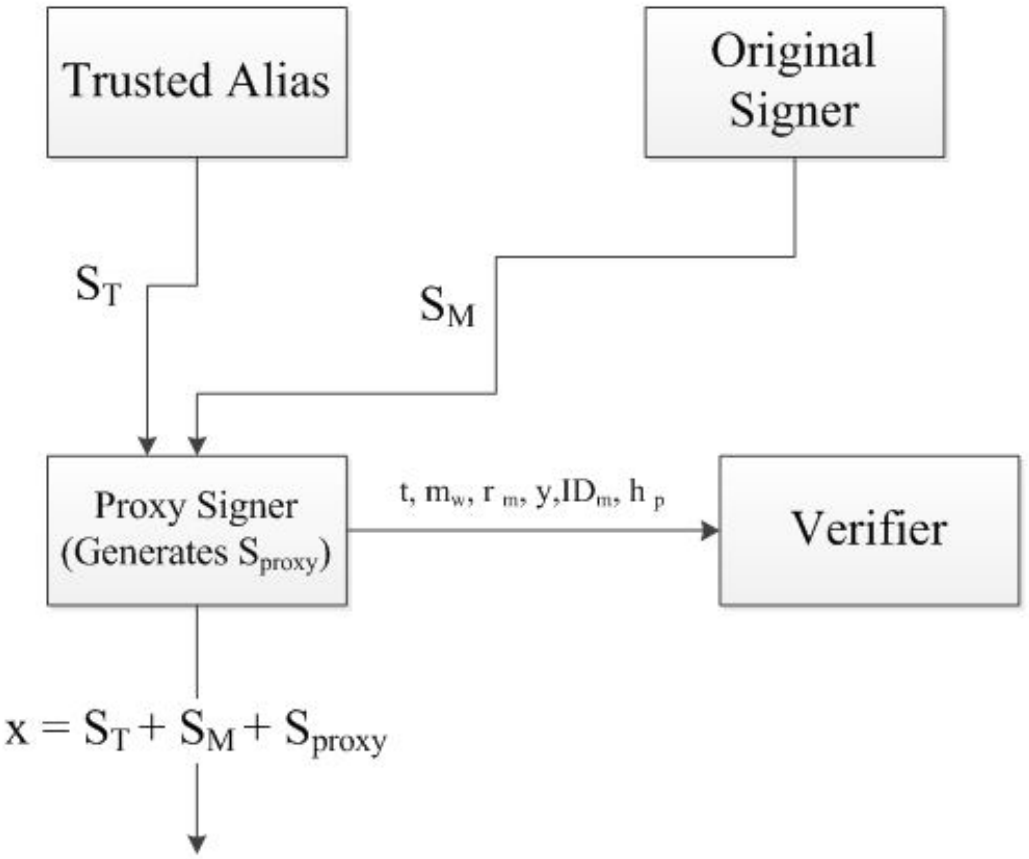

FIGURE 3.1: Layout of the proposed scheme

- Alias issuing phase
- Proxy key generation phase
- Signing phase
- Verification phase

Trusted Alias Issuing Authority T is responsible for issuing an alias for every proxy signer. M denotes an original signer, P denotes a proxy signer, respectively and V denotes a verifier. Figure 3.1 shows the various participants involed in this scheme. Some parameters used in this paper are showed as follows:

- p, q : large prime numbers, where q | $(p-1)$
- g : an element of order q in Z_p^*
- h(.) : a one-way hash function
- m : the signing message

- m_w : the warrant issued by original signer M
- S_T : key for proxy signer P, generated by Alias issuing authority T (in figure)
- S_M : key for proxy signer P, generated by original signer M (in figure)
- S_{proxy} : Secret key of the proxy signer P (in figure)
- x : key used by proxy signer P for proxy signature
- l : an integer in Z_q^*
- y_M : public key of original signer
- y_P : public key of proxy signer
- k : an integer in Z_q^*
- r : an integer in Z_p^*
- s : key generated by original signer for the proxy signer
- u : key generated by proxy signer for proxy signature
- t : hashed value

3.2 Alias issuing phase

T issues an alias h_P, a public parameter r_T and a secret key S_T to P and records the triplet (h_P, k_P, ID_P) into the database, where ID_P is the identity of P. P will check the validation of secret key S_T.

$$k_P \in_R Z_q^*$$

$$h_P = h(k_P, ID_P) \tag{3.2.1}$$

$$k_T \in_R Z_q^*, r_T = g^{k_T} \pmod{p} \tag{3.2.2}$$

$$S_T = x_T h(h_P, rT) + k_T \pmod{q} \tag{3.2.3}$$

record(h_P, k_P, ID_P) check

$$g^{S_T} = {y_T}^{h(h_P, r_T)} r_T \pmod{p} \tag{3.2.4}$$

3.3 Proxy key generation phase

This phase consists of three subphases. They are:

- Key generation phase
- Proxy delegation phase
- Proxy verification

3.3.1 Key generation phase

An original signer M chooses its private key $\mathrm{x}_M \in_R {Z_q}^*$ and publishes public key y_M which is computed as followed

$$\mathrm{y}_M = g^{x_M} \pmod{p} \tag{3.3.1}$$

$$\mathrm{k}_M \in_R Z_q^* \tag{3.3.2}$$

$$\mathrm{r}_M = g^{k_M} \pmod{p} \tag{3.3.3}$$

$$\mathrm{S}_M = x_M h(m_w, r_m, k_m) \pmod{q} \tag{3.3.4}$$

3.3.2 Proxy delegation phase

After generating all the necessary parameters, the original signer M communicates m_w, r_M, S_M to proxy signer P in a secured manner.

3.3.3 Proxy key verification

The proxy signer checks that

$$\mathrm{g}^{S_M} = y_M h(m_w, r_M, k_M) \pmod{p} \tag{3.3.5}$$

if the above condition is satisfied, the proxy signer accepts S_M, S_T and combines with S_{proxy} to form x as signing key.

3.4 Signing Phase

The proxy signer P computes the signing key x as:

$$\mathrm{x} = (\mathrm{S}_M + S_T + S_{proxy}) \pmod{q} \tag{3.4.1}$$

where:
$\mathrm{S}_{proxy} = \mathrm{x}_P h(m_w, r_m) \pmod{q}$, x_P is the private key of Proxy signer P.

To sign a message m, the proxy signer (P) performs the following operations:

- chooses l $\in_R Z_q^*$ and computes u as:

$$\mathrm{u} = \mathrm{g}^l \pmod{p} \tag{3.4.2}$$

- computes :

$$\mathrm{t} = \mathrm{h}(\mathrm{m}, \mathrm{m}_w, u^{x_P l^{-1}}.g^{xh(m_w, r_m)}) \pmod{p} \tag{3.4.3}$$

Proxy signature message is given by

$$(\mathrm{t}, \mathrm{m}_w, r_M, y, ID_M, h_P)$$

3.5 Proxy Verification

Any verifier obtaining the proxy signature $(\mathrm{t}, \mathrm{m}_w, r_M, y)$ can verify for the message m as per the following condition:

$$\mathrm{t} = \mathrm{h}(\mathrm{m}, \mathrm{m}_w, y_P.y^{h(m_w, r_M)}) \pmod{p} \tag{3.5.1}$$

$$\text{Let } \mathrm{j} = \mathrm{y}^{h(m_w, r_m)}$$

$$\mathrm{t} = \mathrm{h}(\mathrm{m}, \mathrm{m}_w, y_P.j) \pmod{p} \tag{3.5.2}$$

If the above condition is satisfied then, (t, m_w, r_M, y, ID_M, h_P) is assumed to be valid one, else it is rejected.

3.6 Correctness of the proxy signature

The correctness of the proxy signature can be checked as below:

$$\begin{aligned} u^{x_P l^{-1}}.g^{xh(m_w,r_M)} &= (g^l)^{x_P l^{-1}}.y^{h(m_w,r_M)} \\ &= (g^{x_P}).y^{h(m_w,r_M)} \\ &= y_P.y^{h(m_w,r_M)} \end{aligned} \tag{3.6.1}$$

where: $y = g^x \pmod p$

Chapter 4

Implementation

In this chapter we will be doing the security analysis of the proposed scheme. A brief comparison between the existing scheme and proposed scheme will be done and an overall implementation results will be displayed.

4.1 Security Analysis

As we have mentioned earlier, the proposed scheme will satisfy the security properties such as verifiability, non-repudiation, non-designated, strong identifiability and proxy privacy. We'll be analysing them here.

4.1.1 Verifiability

According to the property of verifiability, from a proxy signature a verifier can be convinced of the original signers agreement on the signed message. Satisfaction of this property in this paper can be justified by the fact that the original signer M is communicating m_w, r_M, S_M to proxy signer P in a secured manner. P uses m_w for his/her proxy signing purposes and send m_w along with other parameters to verifier for verification. From m_w the verifier comes to know that the original signer has agreed upon the signed message.

4.1.2 Non-Repudiation

By the property of non-repudiation a proxy signer cannot deny a valid proxy signature he/she has generated. This is ensured by the unique key S_{proxy} of the

proxy signer P which is used in generating the signing key. The generated signing key is different for different proxy signers and can be generated by the proxy signer himself, hence, the proxy signer cannot deny a valid proxy signature that he/she generates.

4.1.3 Non-Designation

Non-designation property says that the warrant issued by the original signer should not specify who the proxy signer is and it is also transferable among proxy signers. Again, this property is maintained in the scheme when the original signer M is communicating only the parameters m_w, r_M, S_M to proxy signer P in a secured manner. Among these parameter there is no such thing which can specify the identity of proxy signer.

4.1.4 Strong Identifiability

According to this property any verifier can determine the identity of the proxy signer from a proxy signature. Equations (3.4.1) and (3.4.3) shows that S_{proxy}, which is the unique key of P, is blend with other parameter to get x which is used to generate t, which in turn is passed on to the verifier. The verifier can derive the identity of the proxy signer from t.

4.1.5 Strong Unforgeability

In this proxy signature scheme,proxy signing key x is computed by using $(\mathrm{S}_M, S_T, S_{proxy})$ where: $\mathrm{S}_{proxy} = \mathrm{x}_P h(m_w, r_M)$, x_P is the private key of Proxy signer P. Assuming that an original signer has access to secret key S_T and his own key S_M but he still needs to access S_{proxy} to compute proxy signing key x. Computing S_{proxy} is very difficult as it is known only to a proxy signer. Proxy key x and x_P are used in calculation of verification parameter t as given in the equation (3.4.3). So, if the value calculated by verifier does not match with the value of verification parameter calculated in proxy key generation phase, as in equation (3.5.1) then proxy signature would be regarded as invalid. In this way property of strong unforgeability is satisfied.

4.1.6 Proxy Privacy

Proxy privacy property provides that the proxy signature alone is not enough for anyone to determine the identity of the proxy signer. This is clearly seen in the equation (3.4.3). And also, the Proxy signature message is given by (t, m_w, r_M, y, ID_M, h_P).

4.2 Comparative Performance Evaluation

In this section a comparative evaluation of performance of the existing scheme and the proposed scheme will be done.

4.2.1 Snapshots

Existing Scheme: After implementing the existing scheme in JAVA (NetBeans IDE 6.9.1) the output obtained is as seen in the figure 4.1

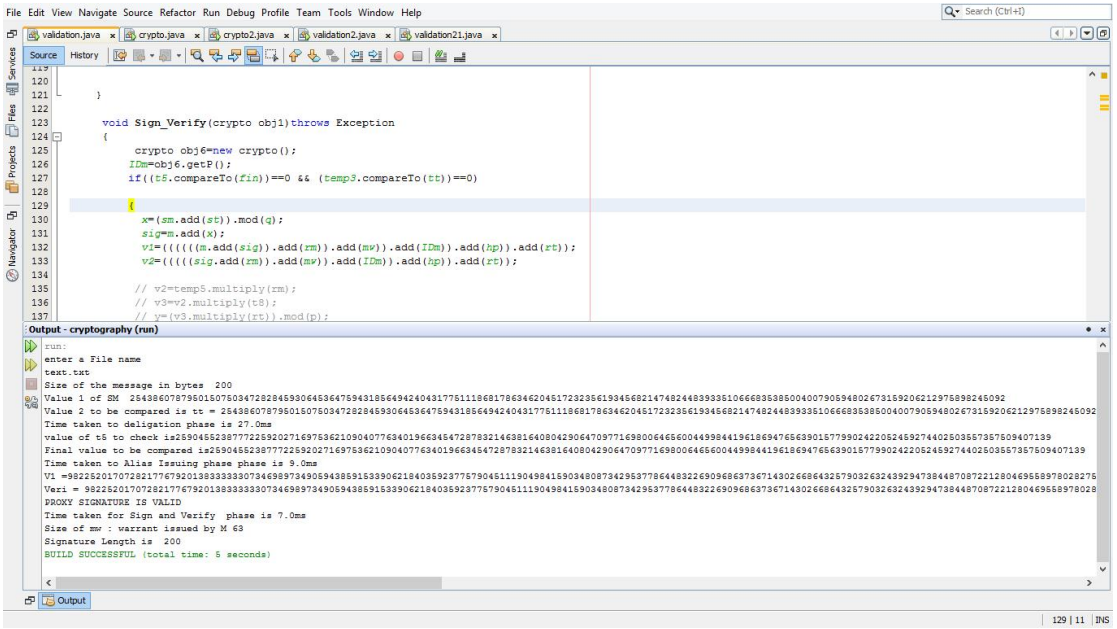

FIGURE 4.1: Snapshot of the existing scheme

Proposed Scheme: After implementing the proposed scheme in JAVA (NetBeans IDE 6.9.1) the output obtained is as seen in the figure 4.2

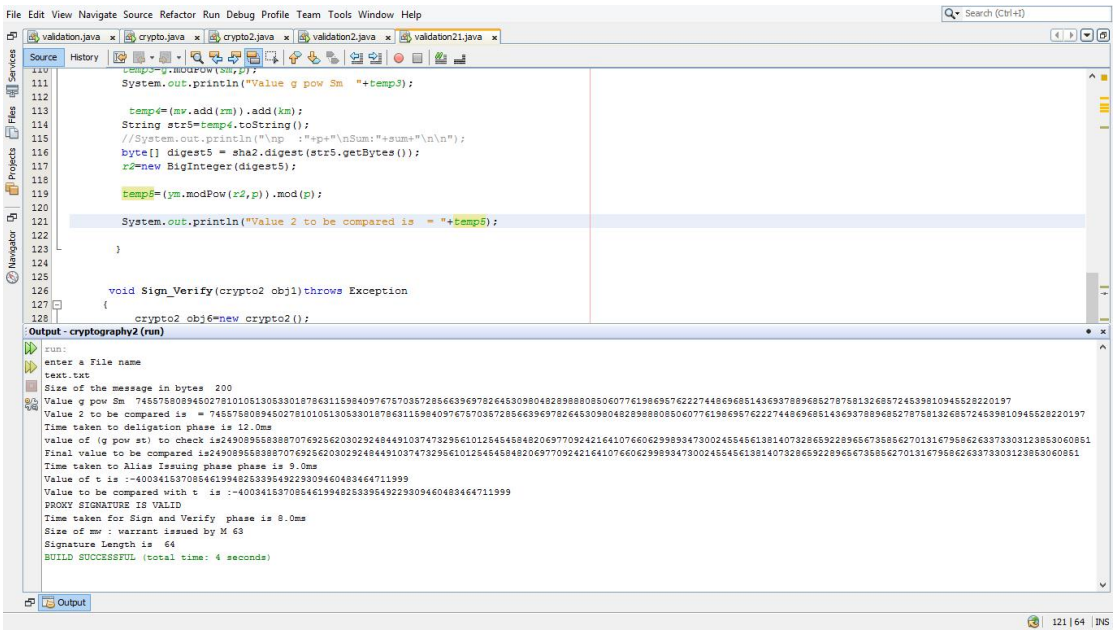

FIGURE 4.2: Snapshot of the proposed scheme

Sl.No.	Signature Length	Warrant Size	AI Phase	Deleg. Phase	S and V Phase
1	200 bytes	63 bytes	9.0 ms	11.0 ms	5.0 ms
2	200 bytes	63 bytes	9.0 ms	10.0 ms	4.0 ms
3	200 bytes	63 bytes	9.0 ms	11.0 ms	7.0 ms

TABLE 4.1: Results of Existing Scheme

Sl.No.	Signature Length	Warrant Size	AI Phase	Deleg. Phase	S and V Phase
1	64 bytes	63 bytes	8.0 ms	12.0 ms	10.0 ms
2	64 bytes	63 bytes	8.0 ms	11.0 ms	9.0 ms
3	64 bytes	63 bytes	8.0 ms	11.0 ms	9.0 ms

TABLE 4.2: Results of Proposed Scheme

4.2.2 Comparison of execution time

The execution time of alias issuing phase, delegation phase and sign and verify phase along with signature length and warrant length for the existing scheme is given in the table 4.1. The table shows the results for three tests.

The execution time of alias issuing phase, delegation phase and sign and verify phase along with signature length and warrant length for the proposed scheme is given in the table 4.2. The table shows the results for three tests.

Note: In the given tables AI Phase stands for Alias Issuing Phase, Deleg. Phase is for Delegation Phase and S and V Phase is for Sign and Verify Phase.

As visible from both the tables, the length of the proxy signature has been greatly reduced. Though, not much difference can be seen in the execution time of the delegation and sign and verify phase, communicational overhead has been taken care of by reducing the number of parameters needed to be communicated to the verifier without hampering the satisfaction of the security properties of the proxy signature.

4.3 Results of implementation

Results of implementing the proposed scheme can be seen from figure 4.2 and table 4.2. The objectives of low communicational overhead and short signature length have been achieved through the implementation of the proposed strong proxy signature scheme with partial delegation. The other objectives were achieved in the design itself.

Chapter 5

Conclusion

The proposed Proxy signature scheme with partial delegation satisfies following properties:

- verifiability
- non-repudiation
- non-designated
- Strong unforgeability

In proxy delegation phase, the m_w, r_M, S_M can be transferred among the proxy signers and hence this S_M secret key can be used by any proxy signer to compute proxy signing key x for signing messages. Thus property of non- designated is achieved.

Since proxy signature of message m involves the identity of original signer ID_M, so a verifier can be convinced that proxy signer is authorized by the original signer to sign the messages.

In this proxy signature scheme, proxy signing key x is computed by using (S_M, S_T, S_{proxy}) where: $S_{proxy} = x_P h(m_w, r_M)$, x_P is the private key of Proxy signer P. Assuming that an original signer has access to secret key S_T and his own key S_M but he still needs to access S_{proxy} to compute proxy signing key x. Computing S_{proxy} is very difficult as it is known only to a proxy signer. Proxy key x and x_P are used in calculation of verification parameter t as given in the equation (3.4.3). So, if the value calculated by verifier does not match with the value of verification parameter calculated in proxy key generation phase, as in equation

(3.5.1) then proxy signature would be regarded as invalid. In this way property of strong unforgeability is satisfied.

The length of the proxy signing key is larger than that one used in Shum and Wei scheme. Hence, it increases the security.

This project work also ensures that a signature of shorter length is obtained without much computational overhead. The signature length of the proposed scheme is 64 bytes and size of the warrant is 63 bytes. Signature length is reduced from 200 bytes in Shum and Wei scheme to 64 bytes in proposed proxy signature scheme. Reduced parameters and signature length provides security and reduces communication overhead, since less parameters have to be passed to the verifier.

The proposed scheme has a wide application in e-voting and e-cash system.

References

[1] M. Mambo, K. Usuda, E. Okamoto, Proxy signature: delegation of the power to sign the message, IEICE Trans. Fundamentals E79-A (9) (1996) PP. 1338 - 1353.

[2] Narn-Yih Lee, Ming-Feng Lee, The security of a strong proxy signature scheme with proxy signer privacy protection, Science Direct, Elsevier, Applied Mathematics and Computation 161 (2005) PP. 807 - 812

[3] Manik Lal Das, Ashutosh Saxena, and Deepak B Phatak , Algorithms and Approaches of Proxy Signature: A Survey, International Journal of Network Security Vol.9 No.3 (2009) PP.264 - 284.

[4] Huang X. Susilo, W. Mu, Y. Wu W., Proxy Signature Without Random Oracles, MSN 2006, LNCS, Vol. 4325, Springer-Verlag, Berlin (2006) PP. 473 - 484

[5] Zhang Jian-hong, Xu Yu-wei, Cui Yuan-bo, Chen Zhi-peng, Efficient short proxy signature scheme based on multi-linear map, Elsevier, Vol. 2 (2012) PP. 109 - 113

[6] Ying Sun, Chunxiang Xu, Yong Yu, Yi Mu, Strongly un-forgeable proxy signature scheme secure in the standard model, The Journal of Systems and Software, Elsevier, Vol. 84 (2011) PP.1471-1479

[7] Liu Zhen-hua, Hu Yu-pu, Zhang Xiang-song Ma Hua, Secure proxy signature scheme with fast revocation in the standard model, Science Direct, Elsevier, Vol. 16(4) (2009) PP. 116 - 124

[8] Kemal Bicakci, One-time proxy signatures, Informatics Institute, Computer Standards Interfaces, Elsevier, Vol. 29 (2007) PP. 499 - 505

[9] A. M. Odlyzko, Discrete logarithms in finite fields and their cryptographic significance, ATT Bell Laboratories, PP.1-2

[10] NIST paper:http://csrc.nist.gov/publications/fips/fips18/fips1803_final.pdf

[11] Wesstein, Eric W. "RSA Encryption." From Mathworld, an online encyclopedia. April, 2001. Available: http://mathworld.wolfram.com/RSAEncryption.html

[12] Junod, Pascal. "Cryptographic Secure Pseudo-Random Bits Generation: The Blum-Blum-Shub Generator." August 1999.

[13] Housley et al. "RFC 2459: Internet X.509 Public Key Infrastructure Certificate and CRL Profile." January, 1999.