A MULTISIGNATURE SCHEME WITH DISTINGUISHED SIGNING AUTHORITIES

A Thesis submitted in partial fulfilment of
the requirements for the degree of

Bachelor of Technology

In

Computer Science & Engineering

By

Sriram Majeti (110CS0076)

Under The Guidance of

Prof. Sujata Mohanty

Department of Computer Science & Engineering
National Institute Of Technology
Rourkela-769008

National Institute Of Technology
Rourkela

CERTIFICATE

This is to certify that the thesis entitled, **“A Multisignature Scheme with Distinguished Signing Authorities”** submitted by **Sriram Majeti (110CS0076)** in the partial fulfilment of the requirements for the award of **Bachelor of Technology Degree** in **Computer Science and Engineering** at National Institute of Technology, Rourkela is an authentic work carried out by him under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University/Institute for the award of any Degree or Diploma.

Date:

Prof. Sujata Mohanty
Dept. of Computer Science &
Engineering
National Institute of Technology
Rourkela-769008

National Institute Of Technology
Rourkela

DECLARATION

I, **Sriram Majeti**, hereby declare that this thesis is my own and has been generated by me as the result of my own original research. I confirm that, wherever contributions of others are involved, every effort is made to indicate this clearly, with due reference to the literature, and acknowledgement of collaborative research. The text of the whole thesis is generated by me and my supervisor is not responsible if any dispute may arise. Furthermore, this thesis contains no material that has been submitted previously, in whole or in part, for the award of any other academic degree.

Sriram Majeti
(110CS0076)
Bachelor of technology,
Dept. of Computer Science & Engineering,
NIT Rourkela.

National Institute Of Technology
Rourkela

ACKNOWLEDGEMENT

I am indebted to my guide **Prof. Sujata Mohanty** for giving me an opportunity to work under her guidance. Like a true mentor, she motivated and inspired me through the entire duration of my work.

I also express my sincere gratitude to **Prof. Santanu Kumar Rath**, Head of the Department, Computer Science and Engineering, for providing valuable departmental facilities.

Sriram Majeti
(110CS0076)
Bachelor of technology,
Dept. of Computer Science & Engineering,
NIT Rourkela.

ABSTRACT

Multisignature is an extension of digital signature, where a group of signers jointly produce a valid signature on a message. Distinguished Signing authorities were first introduced by L.Harn. In Distinguished signing authorities, each of the signers is responsible for only one part of the message rather than the whole message itself. Along with the group of signers, we also have a trusted clerk who verifies the individual signatures and generates the final multisignature.

We extensively studied the two existing schemes proposed by Harn and Hwang. Although Hwnag's scheme overcame all the drawbacks of Harn's scheme, it has drawbacks like high computational cost and communication overhead. So, we proposed a modification over Hwang's scheme taking into consideration the signature length, computational complexity of signature generation and the cost of signature verification.

The performance evaluation of the proposed scheme was theoretically compared to that of the Hwang's scheme. Both the schemes are implemented to calculate the signature length and the execution time required for different phases in the multisignature scheme. Hence these implementation results were useful to verify the performance evaluation.

CONTENTS

List of figures

List of tables

1. OBJECTIVE

To design a new multisignature scheme with distinguished signing authorities based upon computationally hard assumption like discrete logarithmic problem (DLP). Our scheme will mainly focus on the following features:

i. Basic security criteria of Multisignature scheme

ii. Signature Length must be minimum

iii. Cost of signature generation and verification must be low.

iv. Less communicational overhead.

The focus of our research is on features such as signature length, execution time and communication costs as compared to existing schemes.

2. INTRODUCTION

Digital signature schemes help us to validate the authenticity of a message using the signer's public key. Only the signer can produce a valid signature using his private key. Multisignature is a special kind of digital signature scheme in which a group of signers can jointly produce a compact signature on a message. Such a system has many potential uses, for example – Contract signing, Co-signing of a document, etc. There may be situations in which the message can be divided into different parts and each signer is responsible only for a part of the message. In such scenarios, we can use multisignature scheme with distinguished signing authorities.

Multisignature schemes with distinguished signing authorities were first proposed by Harn in based on the discrete logarithmic problem [1]. In Harn's scheme, no evidence can be used to prove the signers distinguished authority. The reason is that, all individual signatures and multisignatures are generated on the same hash digest of all the partial contents. Moreover, Li et al.'s also showed that this scheme is not secure against their attack in [2]. Any insider using this attack, can produce a valid group signature for any message. Later, Hwang was able to overcome the above two problems in [3]. However, Hwang's scheme involves high computational cost and large number of data transmission.

In this thesis, we propose a novel multisignature scheme with distinguished signing authorities based upon DLP which is improvement over Hwang's scheme. The main modifications done are communication between the group of signers and the clerk, the role of the clerk and the calculation of commitment value and individual signatures during the multisignature generation phase.

The thesis is organised as follows. In Chapter 3, we discussed the literature covering all the significant research in this area. The proposed multisignature scheme is discussed in Chapter 4. The implementation details of the above scheme are given in Chapter 5. The results and observations are analysed in Chapter 6. Finally, we conclude in Chapter 7.

CHAPTER – 3
(LITERATURE SURVEY)

3. LITERATURE SURVEY

3.1 Cryptography

Cryptography is the science of writing in secret codes. It mainly involves two processes – encryption and decryption. The ordinary text is called plain text while the plain text after encryption is called cipher text. Encryption is the process of converting plain text into cipher text using a secret key. Decryption is the reverse of encryption. Modern day cryptography uses these three distinct techniques – Symmetric key cryptography, Asymmetric key cryptography and Hashing given in [7, 8].

3.1.1 Symmetric key cryptography:

In symmetric key cryptography, both the encryption and decryption use the same secret key. A single secret key is shared by both the sender and the receiver .The sender sends the plain text by converting the plain text into cipher text using the secret key. The receiver after receiving the cipher text uses the same secret key for decryption.

3.1.2 Asymmetric key cryptography:

In asymmetric key cryptography, we have two types of keys – public key and private key. To send a message, the sender uses the public key of the receiver to encrypt the plain text. Then the receiver uses his secret key to decipher the message.

3.1.3 Hashing:

In hashing, a fixed-length message digest is created out of a variable length message using a cryptographic hash function. These are mainly used in digital signature schemes. Hashing helps in providing check values for message integrity.

3.2 Digital Signature

Digital signature is used for finding the authenticity and validity of documents or messages. It consists of two mechanisms – signing and verification. The signer calculates the signature based on the message/document using his private key. While the verification can be done by anyone using the public key of the signer. Generally the signature is generated on the message digest created using one way cryptographic hash functions like SHA -1. Elgamal proposed a digital signature scheme based on discrete logarithm problem in [4].

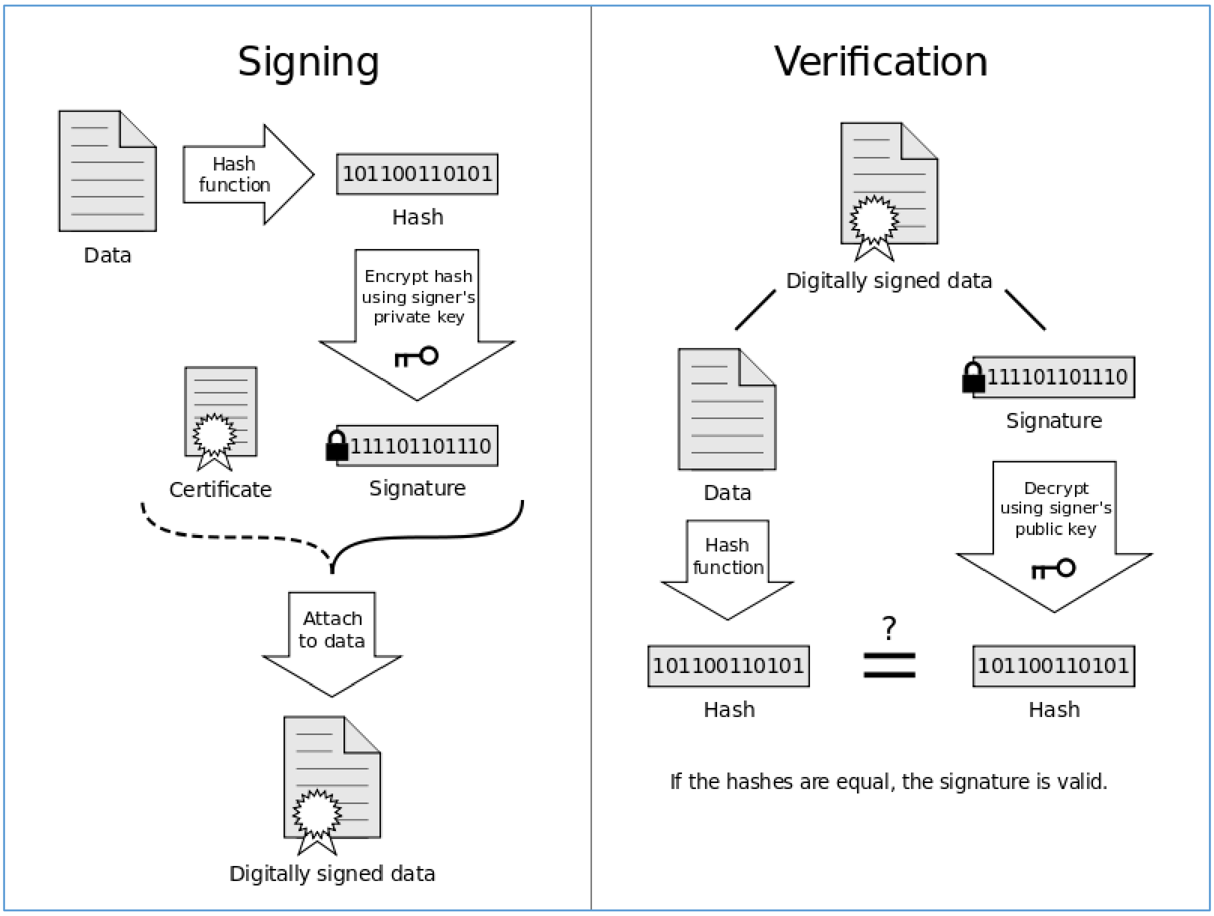


FIGURE – 1: Digital Signature scheme

The digital signature schemes are based upon computationally hard problems like discrete logarithmic problem (DLP), Decision Diffie-Hellman assumption (DDHA), Integer factorization problem (IFP) etc.

Depending on the extra features that can be incorporated into digital signature schemes, they are divided into different categories like:

i. Proxy signature
ii. Blind signature
iii. Ring signature
iv. Group signature
v. Multisignature

3.2.1 Discrete Logarithmic Problem (DLP):

Discrete logarithms are logarithms defined with regard to multiplicative cyclic groups as explained in [10, 16]. If G is a multiplicative cyclic group and g is a generator of G, then from the definition of cyclic groups, we know every element h in G can be written as g^x for some x. The discrete logarithm to the base g of h in the group G is defined to be x.

For example, consider group – Z_5^*,

The generator is 2.

Discrete logarithm of 1 is 4. $(2^4 \equiv 1 \bmod 5)$

The discrete logarithm problem is defined as:

Given a group G, a generator g of the group and an element h of G, to find the discrete logarithm to the base g of h in the group G.

3.2.2 Elgamal Digital Signature based on DLP:

Here we will discuss a basic digital signature based on DLP proposed by Elgamal in [4].

Let p be a large prime number, g be the primitive element, y public key and m be the document to be signed. The signer uses his secret key in order to produce a valid signature s on the given message m. Any outsider should be able to verify the validity of the signature using the public key y of the signer.

Signing procedure:

i. Choose a random number k between 0 and p-1 such that gcd(k,p-1) = 1

ii. Compute $r = g^k \mod p$

iii. S can be found out by solving the equation $m = xr + ks \mod(p-1)$

Verification Procedure:

p, m, r, s, y, g are made public.

By substituting the signature (r, s) along with these values in $g^m = y^r r^s \mod p$, we can find out the authenticity of the digital signature and the message pair.

Usage of DLP in above scheme:

Here, we have made g, p and y public. If an attacker tries to find out the secret key of the signer, he has to solve $y = g^x \mod p$. Finding x is a hard problem based on discrete logarithm. Hence securing the public key and prevents any attacker to find the secret key from it.

3.3 Multisignature scheme

3.3.1 Steps involved in a multisignature scheme:

Formally a multisignature scheme consists of four algorithms – Setup, Key generation, Signature generation and Signature verification as described in [11].

i. Setup :

   Parameters – Setup (1^k)

   A central authority, on input of the security parameter K, runs the algorithm setup to produce global information parameters. Algorithm setup is probabilistic.

ii. Key Generation:

   (SK, PK) - Key generation

   It is executed by each signer on input parameters, generate their respective secret key (SK) and public key (PK). This algorithm is also probabilistic.

iii. Signature Generation:

   The signing algorithm must be a probabilistic algorithm which, given a message m, the global information parameters, and a list of signers L along with their public and secret keys produce a multisignature S. The multisignature can be interactive or non-interactive.

iv. Signature Verification:

   {0, 1} – Verify (parameters, m, L, S) verifies if S is a valid signature on the given message m with respect to L. This algorithm is deterministic.

3.3.2 Distinguished Signing authorities:

The concept of distinguished signing authorities arises when the signers who are generating the multisignature are responsible only for a part of the whole message. It is more useful when the signers don’t need to sign the whole message. Let us understand this using an example similar to the one given in [1].

An institute has a document consisting of physics, chemistry and maths sections. Each signer is responsible for verifying and signing the respective section belonging to the signer. The signer of maths section doesn’t need the knowledge of other two sections while signing. But the whole document is required which consists of all the sections. Anyone should be able to verify the authenticity and integrity of the document using the signature. For the sake of confidentiality, some verifiers may be restricted to access and verify only some sections of the document.

In the above case, multisignature schemes with distinguished signing authorities is required for efficient calculation of the signatures and have better confidentiality.

3.4 Analysis of Existing Schemes

3.4.1 Harn's multisignature scheme:

The concept of distinguished signing authorities was first proposed by Harn in [4]. He listed the properties associated with undistinguished and distinguished authorities.

Multisignature schemes with undistinguished signing authorities:

i. Multisignatures are generated by multiple group members with the knowledge of multiple private keys

ii. They can be verified easily by using the group public key without knowing each signer's public key

iii. It is computationally infeasible to generate the group signature without the co-operation of all group members.

Additional properties for distinguished signing authorities:

i. Each member has distinguished signing authority

ii. Partial content can be verified without revealing the whole document.

Drawbacks of Harn's schemes:

i. No evidence can be used to distinguish the signing authorities. The reason is that, all individual signatures and multisignatures are generated on the same hash digest of the hash digests of all the partial contents.

ii. Z.C.Li showed that this scheme is not secure against their attack in [2]. Any insider using this attack, can produce a valid group signature for any message.

3.4.2 Hwang's multisignature scheme:

Hwang was able to overcome all the drawbacks that were present in Harn's multisignature schemes in [3], in the following ways:

i. To overcome the insider forgery attack proposed in [2], the group's public key generation has been modified to use exponential operation instead of multiplications.
ii. A new step is included in the scheme called Evidence verification which helps us to distinguish the signing authorities. Now, signers can prove their ownership of their respective signatures and message pairs.

The following are the drawbacks we found out in Hwnag's scheme:

i. The multisignature generation phase requires lot of unnecessary data transmissions which lead to high communication overhead.
ii. Since, setup is run only a few times, the significant computational cost to be considered is during the signature generation and evidence verification phase. The computational complexity is high in case of the multisignature generation phase.

CHAPTER – 4

(PROPOSED MULTISINGATURE SCHEME)

4. PROPOSED MULTISIGNATURE SCHEME

4.1 Modification over Hwnag's scheme

The following are the major changes made in the existing Hwang's signature in order to propose our new scheme:

1. Role of the trusted clerk.
2. Communication between the signers and the clerk
3. Calculation of commitment value and individual signatures during the multisignature generation phase.

4.2 Proposed Multisignature scheme

Let the group of signers be {U_1, U_2, U_3… U_n} and the message be M = {m_1, m_2, m_3… m_n}. U_i is responsible for signing the partial content m_i for i=1, 2…n. A trusted clerk chooses a large prime p, a prime divisor q such that q | (p-1) and a one way hash function – h.

STEP -1 (Key generation):

Each signer selects a secret key x_i such that $1<x_i<q$. x_i is known only to U_i (secret key). Each signer U_i publishes their public key $y_i = \alpha^{x_i} \; (mod\, p)$. ɑ is the generator of cyclic group of order q Є Z_p^*. After everyone publishes their public key, clerk calculates group public key $Y = \prod_{i=1}^{n} {y_i}^{y_i} \; (mod\, p)$.

STEP - 2 (Multisignature generation phase):

i. Each signer U_i selects a random number k_i Є Zq* and computes $r_i = \alpha^{k_i} \; (mod\, p)$. He also computes the message digest using the hash function – h. Then each signer transmits (r_i, h (m_i)) to the clerk.

ii. Then the clerk calculates, the commitment value, R = $\prod_{i=1}^{n} r_{\mathrm{i}}^{(h(mi))}$ $(mod\, p)$ and the message digest of whole message m’ = h(h(m_1), h(m_2), …h(m_n),R) and sends (R and m’) to every signer.

iii. Each signer then computes their signatures $s_i = \left(y_i x_i m' + Rk_i h(m_i)\ \right)(mod\, q)$ and send them to clerk.

iv. Once the clerk receives all the signatures from the signers, he checks the authenticity of the individual signatures using the equation, $\alpha^{s_i} = ({y_i}^{m' y_i} * {r_i}^{Rh(m_i)}\ )(mod\ \ p)$.

v. If all the signatures are valid, then the clerk generates the multisignature, $S = (\sum_{i=1}^{n} s_i)\left(mod\, q\right)$. The multisignature for the message M is (R, S).

STEP - 3 (Multisignature Verification):

Check if the multisignature, (R, S) satisfies the equation, $\alpha^{S} = \left(Y^{m'} * R^{R}\right)\left(mod\, p\right)$. If it satisfies then the multisignature is valid and legit. Partial content can also be given like h(m_1)||h(m_2)||…||m_i||…||h(m_n) instead of giving the all the parts of the message - m_1||m_2||…||m_i||…||m_n .

STEP - 4 (Evidence Verification):

All individual signature (r_i, s_i) can be used as evidence to show that U_i is responsible for signing the part of the message – m_i. If (r_i, s_i) satisfy the equation, $\alpha^{s_i} = \left({y_i}^{m' y_i}\ *\ {r_i}^{Rh(m_i)}\right)(mod\, p)$ then evidence for U_i and m_i can be verified.

CHAPTER – 5

(IMPLEMENTATION OF PROPOSED SCHEME)

5. IMPLEMENTATION OF PROPOSED SCHEME

5.1 Details regarding implementation

- Language used : JAVA (DSA parameters, Big Integer class)
- Hash function : SHA -1 (creates 160-bit hash value)
- No. of signers considered are 2-10
- No. of bits used for p : 512 -1024 (only multiples of 64)
- No. of bits used for q : 160
- Permanent/Intermediate data is stored in CSV files
- Messages to be signed are present in a folder – “messages”
- 4 modules present – Setup, Generation, Verification, Evidence

5.2 Modules

5.2.1 Setup:

This module is responsible for the following:

i. Initialising the domain parameters –

p – Large prime number

q – Large prime number [q| (p-1)]

g – Generator

bl – Bit length

n – Number of signers

ii. Initialising the public and private keys for the n signers.

Input: No. of signers, Bit Length

Output: 3 CSV files

1. parameter.csv – p, q, g, bl, n
2. privatekey.csv – private keys of all the signers (x_i)
3. publickey.csv – public keys of all the signers and the group public key (y_i and Y)

$$y_i = g^{x_i} \ (mod\, p) \qquad Y = \prod_{i=1}^{n} {y_i}^{y_i} \ (mod\, p)$$

5.2.2 Generation:

This module is responsible for the following:

1. Obtain domain parameters from “parameter.csv” file.
2. Obtain key pairs (x_i,y_i) from “privatekey.csv”, “publickey.csv” files respectively.
3. Calculation of hash values of m_i and M’.
4. Storing the values of individual hash values - $h(m_i)$ in “hash.csv” file.
5. Individual commitment values – r_i are calculated and stored in “commitment.cvs” file. Total commitment value – R is formulated from $h(m_i)$ and r_i.
6. Y, R, M’ are stored in “output.csv” file.

$$r_i = g^{k_i} \ (mod\, p)$$

$$R = \prod_{i=1}^{n} \ {r_i}^{(h(m_i))} \ (mod\, p)$$

$$M' = h(\, h(m_1), h(m_2), \ldots\ldots\ldots h(m_n), R)$$

7. Using values y_i, x_i, M’, k_i, $h(m_i)$, we calculate individual signatures – s_i

$$s_i = \left(y_i x_i m' + R k_i h(m_i)\,\right) \ mod\; q$$

viii. Verification of individual signature authenticity.

$$g^{s}{}_{i} = \left(y_i^{m' y_i} \; * \; r_i^{Rh(mi)}\right) \; mod \;\; p$$

After verifying, store s_i in “signature.csv” file.

ix. Calculation of the final multisignature S and append it to the file “output.csv”.

$$\mathrm{S} = \left(\sum_{i=1}^{n} si\right) \; mod \; q$$

Input:

i. 3 CSV files generated in step-1 – parameter.csv, publickey.csv, privatekey.csv

ii. Set of message stored in the folder – “messages”

Output:

i. 4 CSV files - hash.csv, commitment.csv, signature.csv, output.csv

ii. Final multisignature – S

5.2.3 Verification:

This module is responsible for the following:

i. Obtain domain parameters from “parameter.csv” file

ii. S, M’, R, Y are obtained from “output.csv” file.

iii. Check the authenticity of the multisignature using $g^{S} = \left(Y^{M'} \; * \; R^{R}\right) mod \;\; p$

Input: 2 CSV files – parameter.csv, output.csv.

Output:

i. Validity and authenticity of the multisignature.

ii. Final multisignature – (R,S) and its length.

5.2.4 Evidence:

This module is responsible for the following:

i. Obtain domain parameters from “parameter.csv” file

ii. M’, R are obtained from “output.csv” file.

iii. y_i, s_i, h(m_i), r_i are obtained from the CSV files – publickey.csv, signature.csv, hash.csv and commitment.csv respectively.

iv. Check validity and authenticity of individual signatures using the following formula

$$g^{si} = \left(yi^{m'yi} * ri^{Rh(mi)}\right) mod\ p$$

Input:

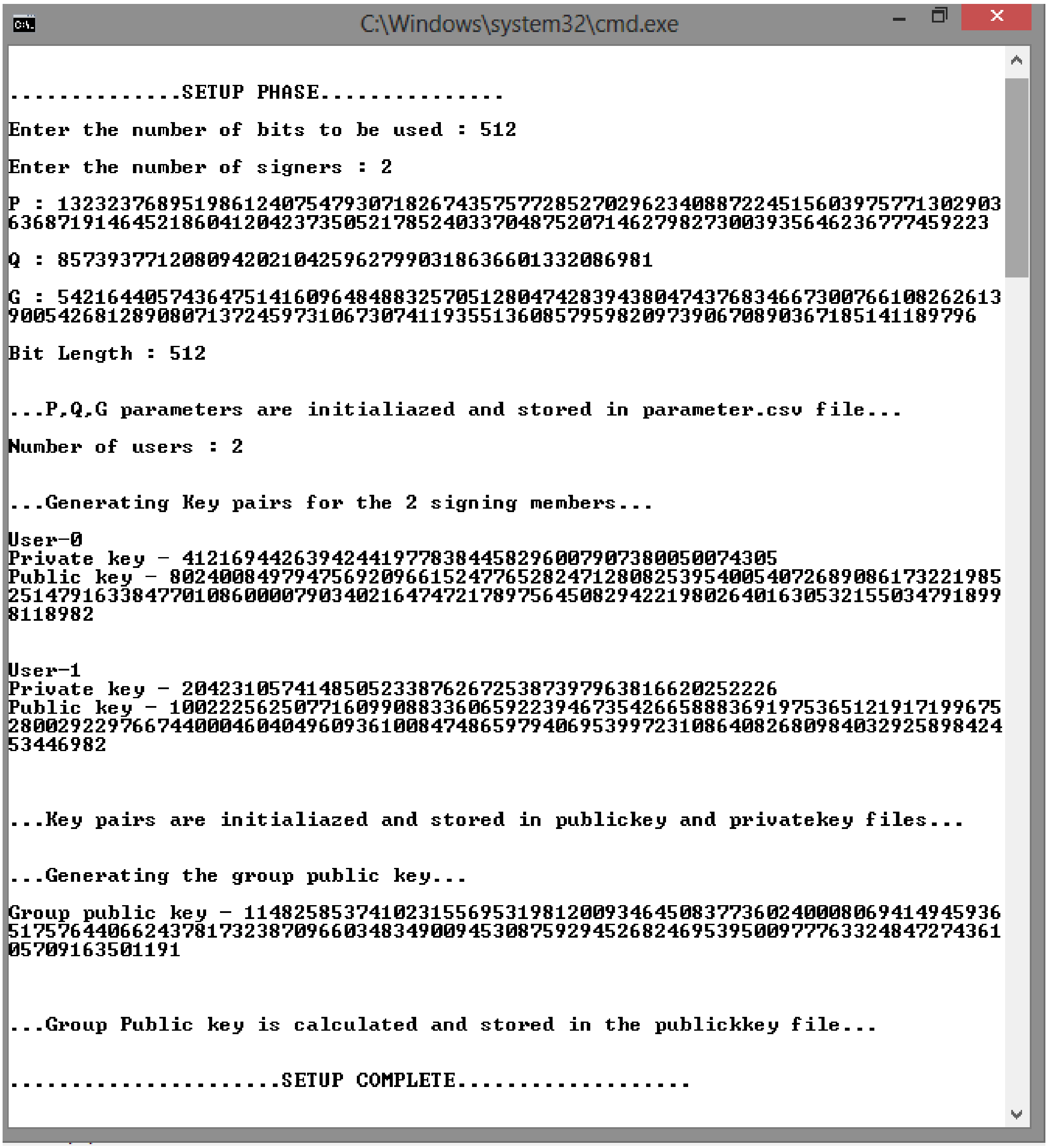
i. 6 CSV files – parameter.csv, output.csv, publickey.csv, signature.csv, hash.csv and commitment.csv

ii. Number of the user for (r_i,s_i) and hash of the message – h(m_j)

Output: Validity and authenticity of the individual signatures.

5.3 Screen shots

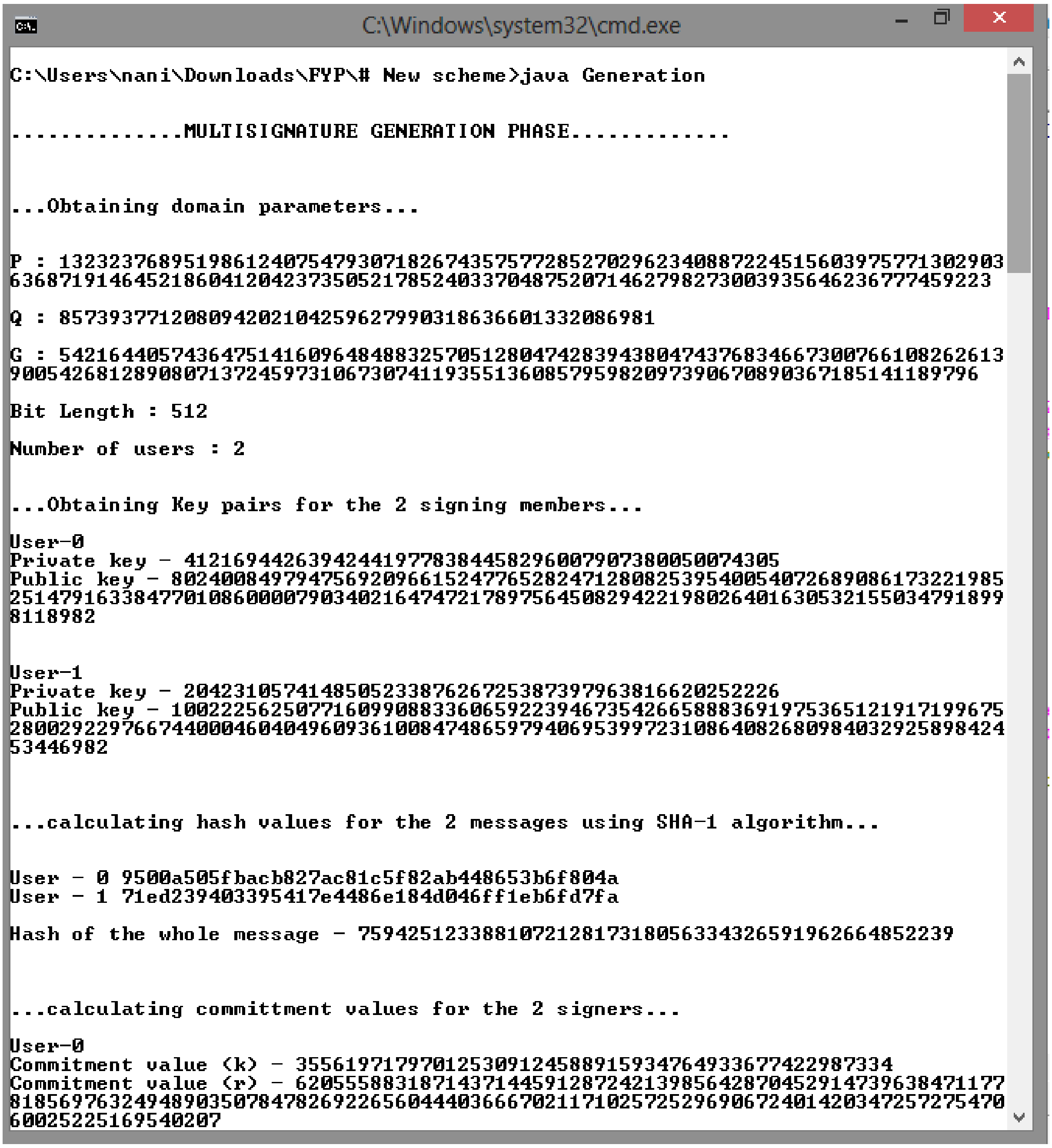
5.3.1 Setup phase:



```
..............SETUP PHASE...............

Enter the number of bits to be used : 512

Enter the number of signers : 2

P : 1323237689519861240754793071826743575772852702962340887224515603975771302903
6368719146452186041204237350521785240337048752071462798273003935646236777459223

Q : 857393771208094202104259627990318636601332086981

G : 5421644057436475141609648488325705128047428394380474376834667300766108262613
900542681289080713724597310673074119355136085795982097390670890367185141189796

Bit Length : 512


...P,Q,G parameters are initialiazed and stored in parameter.csv file...

Number of users : 2


...Generating Key pairs for the 2 signing members...

User-0
Private key - 412169442639424419778384458296007907380050074305
Public key - 8024008497947569209661524776528247128082539540054072689086173221985
25147916338477010860000790340216474721789756450829422198026401630532155034791899
8118982


User-1
Private key - 204231057414850523387626725387397963816620252226
Public key - 1002225625077160990883360659223946735426658883691975365121917199675
28002922976674400046040496093610084748659794069539972310864082680984032925898424
53446982



...Key pairs are initialiazed and stored in publickey and privatekey files...


...Generating the group public key...

Group public key - 1148258537410231556953198120093464508377360240008069414945936
51757644066243781732387096603483490094530875929452682469539500977763324847274361
05709163501191



...Group Public key is calculated and stored in the publickkey file...


......................SETUP COMPLETE..................
```

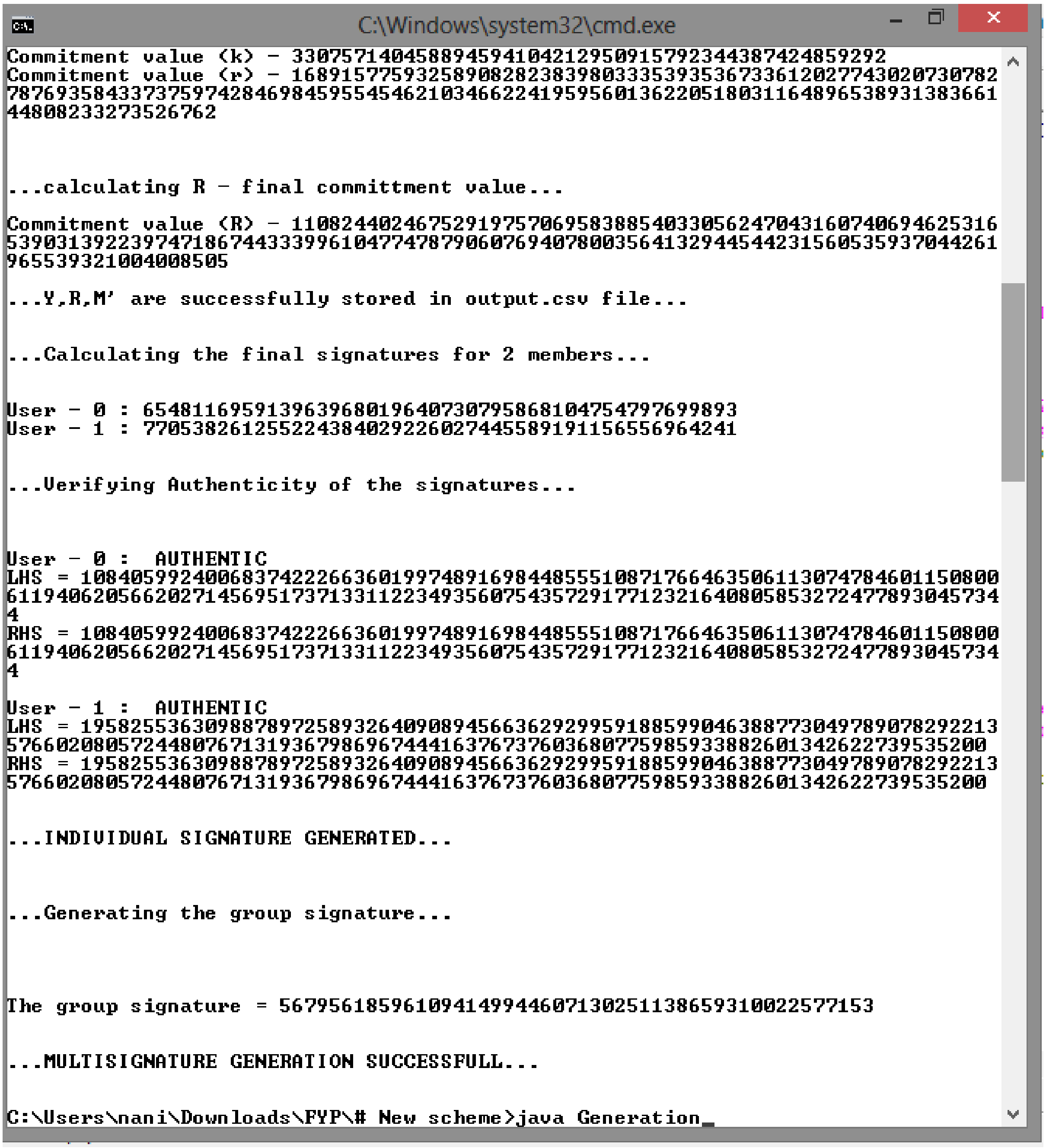
FIGURE - 2: Output of Setup phase

5.3.2 Signature Generation phase:



```
C:\Users\nani\Downloads\FYP\# New scheme>java Generation


..............MULTISIGNATURE GENERATION PHASE.............



...Obtaining domain parameters...


P : 1323237689519861240754793071826743575772852702962340887224515603975771302903
6368719146452186041204237350521785240337048752071462798273003935646236777459223

Q : 857393771208094202104259627990318636601332086981

G : 5421644057436475141609648488325705128047428394380474376834667300766108262613
900542681289080713724597310673074119355136085795982097390670890367185141189796

Bit Length : 512

Number of users : 2


...Obtaining Key pairs for the 2 signing members...

User-0
Private key - 412169442639424419778384458296007907380050074305
Public key - 8024008497947569209661524776528247128082539540054072689086173221985
25147916338477010860000790340216474721789756450829422198026401630532155034791899
8118982


User-1
Private key - 204231057414850523387626725387397963816620252226
Public key - 1002225625077160990883360659223946735426658883691975365121917199675
28002922976674400046040496093610084748659794069539972310864082680984032925898424
53446982



...calculating hash values for the 2 messages using SHA-1 algorithm...


User - 0 9500a505fbacb827ac81c5f82ab448653b6f804a
User - 1 71ed239403395417e4486e184d046ff1eb6fd7fa

Hash of the whole message - 759425123388107212817318056334326591962664852239



...calculating committment values for the 2 signers...

User-0
Commitment value (k) - 355619717970125309124588915934764933677422987334
Commitment value (r) - 620555883187143714459128724213985642870452914739638471177
81856976324948903507847826922656044403666702117102572529690672401420347257275470
60025225169540207
```

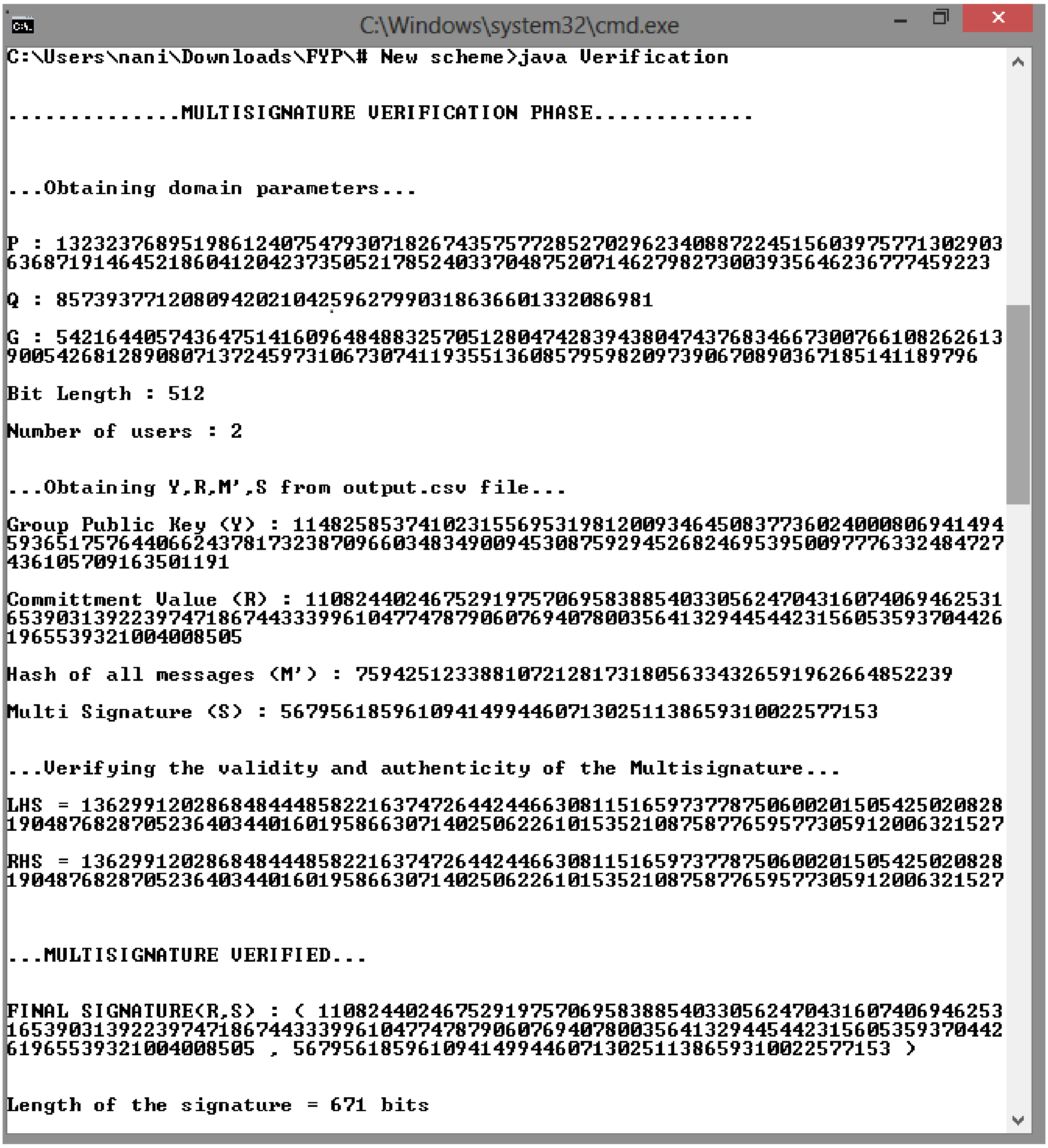
FIGURE - 3: Output of Generation phase_1



```
Commitment value (k) - 330757140458894594104212950915792344387424859292
Commitment value (r) - 1689157759325890828238398033353935367336120277430207307827876935843373759742846984595545462103466224195956013622051803116489653893138366144808233273526762



...calculating R - final committment value...

Commitment value (R) - 11082440246752919757069583885403305624704316074069462531653903139223974718674433399610477478790607694078003564132944544231560535937044261965539321004008505

...Y,R,M' are successfully stored in output.csv file...


...Calculating the final signatures for 2 members...


User - 0 : 654811695913963968019640730795868104754797699893
User - 1 : 770538261255224384029226027445589191156556964241


...Verifying Authenticity of the signatures...



User - 0 :  AUTHENTIC
LHS = 1084059924006837422266360199748916984485551087176646350611307478460115080061194062056620271456951737133112234935607543572917712321640805853272477893045734
4
RHS = 1084059924006837422266360199748916984485551087176646350611307478460115080061194062056620271456951737133112234935607543572917712321640805853272477893045734
4

User - 1 :  AUTHENTIC
LHS = 195825536309887897258932640908945663629299591885990463887730497890782922135766020805724480767131936798696744416376737603680775985933882601342622739535200
RHS = 195825536309887897258932640908945663629299591885990463887730497890782922135766020805724480767131936798696744416376737603680775985933882601342622739535200


...INDIVIDUAL SIGNATURE GENERATED...



...Generating the group signature...




The group signature = 567956185961094149944607130251138659310022577153


...MULTISIGNATURE GENERATION SUCCESSFULL...

C:\Users\nani\Downloads\FYP\# New scheme>java Generation_
```

FIGURE - 4: Output of Generation phase_2

5.3.3 Signature Verification phase:



```
C:\Users\nani\Downloads\FYP\# New scheme>java Verification


..............MULTISIGNATURE VERIFICATION PHASE.............


...Obtaining domain parameters...

P : 1323237689519861240754793071826743575772852702962340887224515603975771302903
6368719146452186041204237350521785240337048752071462798273003935646236777459223

Q : 857393771208094202104259627990318636601332086981

G : 5421644057436475141609648488325705128047428394380474376834667300766108262613
900542681289080713724597310673074119355136085795982097390670890367185141189796

Bit Length : 512

Number of users : 2


...Obtaining Y,R,M',S from output.csv file...

Group Public Key (Y) : 114825853741023155695319812009346450837736024000806941494
59365175764406624378173238709660348349009453087592945268246953950097776332484727
436105709163501191

Committment Value (R) : 11082440246752919757069583885403305624704316074069462531
65390313922397471867443339961047747879060769407800356413294454423156053593704426
1965539321004008505

Hash of all messages (M') : 759425123388107212817318056334326591962664852239

Multi Signature (S) : 567956185961094149944607130251138659310022577153


...Verifying the validity and authenticity of the Multisignature...

LHS = 13629912028684844485822163747264424466308115165973778750600201505425020828
19048768287052364034401601958663071402506226101535210875877659577305912006321527

RHS = 13629912028684844485822163747264424466308115165973778750600201505425020828
19048768287052364034401601958663071402506226101535210875877659577305912006321527



...MULTISIGNATURE VERIFIED...


FINAL SIGNATURE(R,S) : ( 1108244024675291975706958388540330562470431607406946253
16539031392239747186744333996104774787906076940780035641329445442315605359370442
61965539321004008505 , 567956185961094149944607130251138659310022577153 )


Length of the signature = 671 bits
```

FIGURE - 5: Output of Verification phase

CHAPTER – 6
(RESULTS AND OBSERVATIONS)

6. RESULTS AND OBSERVATIONS

6.1 Security Analysis

1. The proposed scheme is not vulnerable to the Li et al.'s attack. If an insider U_k wants to forge a signature, he has to publish his public key,

   $$y'_k = y_k\big((\prod_{i=1}^{n-1} y_i{}^{y_i})^{-1}\big)(mod\, p).$$

   The above problem is a very difficult problem to solve. This can't be done in polynomial time.

2. If any attacker tries to solve the following equation,

   $$\alpha^{s_i} = \big(y_i{}^{m' y_i} * r_i{}^{Rh(m_i)}\big)(mod\, p).$$

   This involves solving a DLP and a one-way hash function which are hard problems. So finding the secret keys of the signers is not possible, so nobody can forge the individual signatures or the multisignature.

6.2 Performance evaluation

1. The signature length of proposed scheme = |R|+|S| ≈ |p|+|q| which is the minimal possible length with primes - p, q. Both the proposed scheme and the Hwang's scheme have same length.

2. The verification cost for our scheme is same as to that of the Hwang's schemes.

3. Let the communication cost be measured by the total no. of transmissions during the multisignature generation process and the number of signers be *n*.

TABLE – 1: Comparison of communication cost

Name of the scheme	**Total number of transmission**
Proposed scheme	3n
Hwang's scheme	$n^2 + n$

4. Computational Cost

Let T_H denote the cost of hash function h, T_E denote the cost of exponential operation and T_M denote the cost of multiplication operation in modular arithmetic.

i. Proposed scheme:

a) Public group key generation = $nT_E + (n-1)T_M$

b) Multisignature generation =

$\{n\ (T_E + T_H)\} + \{\ (n-1)T_M + T_H\ \} + \{\ n(4T_M)\ \} + \{\ 3n(T_E + T_M\ )\ \}$

$= 4n\ T_E + (8n-1)\ T_M + (n+1)\ T_H$

c) Multisignature verification = $3T_E + T_M$

ii. Hwang’s scheme:

a) Public group key generation = $nT_E + (n-1)T_M$

b) Multisignature generation =

$\{n\ (T_E + T_H)\} + \{(n-1)T_M + T_H\} + \{n\ (4T_M + 2T_H)\} + \{3n(T_E + T_M)\}$

$= (n^2+4n)\ T_E + (n^2+6n)\ T_M + (n^2+3n)\ T_H$

c) Multisignature verification = $3\ T_E + T_M$

It can be observed that in public key generation and signature verification phase, the computational cost is equal in both the schemes. They only differ in their multisignature generation phase.

TABLE – 2: Comparison of computational cost

Name of the scheme	Computational cost in Multisignature generation
Proposed scheme	$4n\ T_E + (8n-1)\ T_M + (n+1)\ T_H$
Hwang’s scheme	$(n^2+4n)\ T_E + (n^2+6n)\ T_M + (n^2+3n)\ T_H$

6.3 Empirical Results

We have implemented both our proposed scheme and Hwang's scheme. The different criteria compared are Length of signature (bits) and Execution time (micro sec) for multisignature generation phase.

6.3.1 Signature Length:

Number of signers (n)	PROPOSED SCHEME	HWANG'S SCHEME
2	670	671
3	671	669
4	670	669
5	668	671
6	667	666
7	669	668
8	669	671
9	670	671
10	668	667
AVERAGE	669.11	669.22

TABLE - 3: Signature length comparison

Length of p = 512 bits

Length of q = 160 bits

|p| + |q| = 512 +160 = 672 bits

Both the schemes have almost the same signature length and are equal to |p| + |q|.

6.3.2 Execution time for Multisignature Generation phase

Number of signers (n)	Proposed scheme (micro seconds) (I)	Hwang's scheme (micro seconds) (II)	Difference (micro seconds) (II) - (I)
2	73266	78867	5601
3	85944	98825	12881
4	98930	121358	22428
5	107690	144043	36353
6	119468	180815	61347
7	130947	211537	80590
8	144828	245240	100412
9	158492	270757	112265
10	166054	303091	137073

TABLE - 4: Multisignature generation phase execution time

- It is evident from the table that the execution time for generation phase is directly proportional to the number of signers.
- In every case, the execution time for the proposed scheme is always less than that of the existing scheme – Hwang's scheme.
- The difference between the execution time of the schemes increases notably when there are more number of signers.

CHAPTER – 7

(FUTURE SCOPE AND CONCLUSION)

7.1 FUTURE SCOPE

The Setup phase uses exponential operations in order to avoid the insider attack shown by Li et al.'s in [2]. We can try to reduce this computational cost by using less costly operation like multiplication.

We studied only the basic attacks possible. So, further possible attacks can be tested and improvements can be suggested to increase the security of the proposed scheme.

7.2 CONCLUSION

The proposed scheme has all the basic requirements of a multisignature scheme with distinguished signing authorities listed in [1]. Since the setup phase in Hwang's schemes in [3] is same as that of the proposed scheme, it is resistant to insider attack shown by Li et al.'s in [2]. Our scheme is based on a computationally hard assumption, i.e. Discrete Logarithmic Problem which makes it secure.

The length of the signature, public key generation and verification cost are same for the proposed scheme and Hwang's scheme. But, the communication cost and the multisignature generation cost has been drastically reduced in the proposed scheme when compared to Hwang's scheme. These have been substantiated by observation and results obtained upon implementing both schemes.

8. REFERENCES

[1] Harn, L., "Digital multisignature with distinguished signing authorities," Electronics Letters, vol.35, no.4, pp.294, 295, 18 Feb 1999.

[2] Li, Z. C.; Hui, L. C K; Chow, K.P.; Chong, C. F.; Tsang, W. W.; Chan, H. W., "Cryptanalysis of Harn digital multisignature scheme with distinguished signing authorities," Electronics Letters, vol.36, no.4, pp.314, 315, 17 Feb 2000.

[3] S. J. Hwang, M. S. Hwang, S. F. Tzeng, "A New Digital Multisignature Scheme With Distinguished Signing Authorities," J. Inform. Sci. Eng., 2003, 19(5):881-887.

[4] T. Elgamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," IEEE Trans. Inform. Theory. 1985, 31(4):469-472.

[5] L. H. Dung, N. H. Minh, "New Digital Multisignature Scheme with Distinguished Signing Responsibilities," Int. J. Compt. Sci. Network Security, 2010, 10(1):51-57.

[6] Minh Nguyen Hieu; Hung Dao Tuan, "New multisignature schemes with distinguished signing authorities," Advanced Technologies for Communications (ATC), 2012 International Conference on , vol., no., pp.283,288, 10-12 Oct. 2012

[7] William Stallings, "Cryptography and Network Security: Principles and Practice", Prentice Hall, 4th Edition, 2005.

[8] Behrouz A. Fourazan, Debdeep Mukhopadhyay, "Cryptography and Network Security", Tata McGraw Hill, 2nd Edition, 2010.

[9] "Efficient discrete logarithm based multi-signature scheme in the plain public key model", Designs, Codes and Cryptography, 2010, Volume 54, Number 2, Page 121, Changsha Ma, Jian Weng, Yingjiu Li, Robert Deng.

[10] http://www.doc.ic.ac.uk/~mrh/330tutor/ch06s02.html - Discrete Logarithm Problem (Internet source).

[11] "Multisignatures as Secure as the Diffie-Hellman Problem in the Plain Public-Key Model", Pairing-Based Cryptography – Pairing 2009 Lecture Notes in Computer Science Volume 5671, 2009, pp 35-51, Duc-Phong Le, Alexis Bonnecaze, Alban Gabillon.

[12] E. J. Yoon and K. Y. Yoo, "Cryptanalysis of Two Multisignature Schemes with Distinguished Signing Authorities," 2006 International Conference on Hybrid Information Technology - Vol2 (ICHIT'06), 2006, p.492-495.

[13] C. Popescu, "A Digital Multisignature Scheme with Distinguished Signing Responsibilities," Studies in Informatics and Control, 2003, 12(3):227-231.

[14] H. F. Huang, C. C. Chang, "Multisignatures with distinguished signing authorities for sequential & broadcasting architectures," Comput. Stand. Interfaces., 2005, 27(2): 169-176

[15] Itakura K., Nakamura K.: A public-key cryptosystem suitable for digital multisignatures. NEC Res. Dev., 71, 1–8 (1983).

[16] A. Odlyzko, "Discrete logarithms in finite fields and their cryptographic significance," Proc. Eurocrypt 84.

[17] L. Harn, "Group-oriented (t, n) threshold digital signature scheme and digital multisignature," IEE Proceedings: Computers and Digital Techniques, Vol. 141, 1994, pp.307-313.

[18] L. Harn and Y. Xu, "Design of generalised Elgamal type digital signature schemes based on discrete logarithm," Electronics Letters, Vol. 30, 1994, pp. 2025-2026.

[19] Goldwasser S., Micali S., and Rivest R.: A digital signature scheme secure against adaptive chosen message attacks. SIAM J. Comput. 17(2), 281–308 (1988).

[20] Goldwasser, S., S. Micali, and A. Yao, "Strong Signature Schemes," Proc. 15th Annual ACM Symposium on Theory of Computing, (Boston Massachusetts, April 1983), 431-439.