A
Dissertation
On

EFFICIENT VERIFIABLE SECRET SHARING SCHEME USING RSA CRYPTOGRAPHY

Submitted in Partial Fulfillment of the Requirement for the

Award of the Degree of

MASTER OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING
by

Vikas Kumar
2K13/CSE/31

Under the guidance of

Prof. O.P. Verma
(Head of Department, Computer Science and Engineering)

2013-2016
Department of Computer Science and Engineering

Delhi Technological University
Shahbad Daulatpur, Main Bawana Road, Delhi-110042

AUGUST 2016

DECLARATION

I hereby declare that the Major Project-II (CO-801) work entitled **“Efficient Verifiable Secret Sharing Scheme Using RSA Cryptography”** which is being submitted at Delhi Technological University, in partial fulfillment of requirements for the award of the degree of Master of Technology (Computer Science and Engineering) is a bonafide work carried out by me. To the best of my knowledge, the project work has not been submitted to any university or institution for the award of any degree.

Vikas Kumar
University Roll No.: 2K13/CSE/31)
M.Tech. (Computer Science & Engineering)
Department of Computer Science and Engineering

ACKNOWLEDGEMENT

I take this opportunity to express my gratitude to all those who have been instrumental in the successful completion of this project. First and foremost I would like to thank the **Lord Almighty** for showering his blessing in all endeavors.

I express my sincere thanks and a deep sense of gratitude to my project guide and Head of Department **Prof. O.P. Verma**, Department of Computer Science and Engineering, Delhi Technological University, for his valuable motivation, encouragement, and guidance to complete this project work successfully. I consider myself fortunate for having the opportunity to learn and work under his supervision and guidance.

I owe my sincere thanks to **Mr. Nitin Jain**, a doctoral student in the Department of Computer Science and Engineering, Delhi Technological University, who helped me with his insight regarding the problematic areas and introduced me to the secret sharing schemes.

Last but not the least I am grateful to all of my college teachers, librarian, staff members, family and friends for giving their valuable suggestions, encouragement, support and blessings for completing my project successfully.

Vikas Kumar
University Roll No.: 2K13/CSE/31)
M.Tech. (Computer Science & Engineering)
Department of Computer Science and Engineering

Department of Computer Science and Engineering
DELHI TECHNOLOGICAL UNIVERSITY
Shahabad Daulatpur, Main Bawana Road,
Delhi-110042

CERTIFICATE

This is to certify that the project work entitled **"Efficient Verifiable Secret Sharing Scheme Using RSA Cryptography"** is a bonafide record of work done by **Vikas Kumar, Roll No.: 2K13/CSE/31** at Delhi Technological University for partial fulfillment of the requirement for the award of the degree Master of Technology in Computer Science and Engineering. This project is a record of the candidate work carried out by him under my supervision and has not been submitted elsewhere, either in part or full, for the award of any other degree or diploma to the best of my knowledge and belief.

Prof. O. P. Verma
Head of Department
Department of Computer Science and Engineering
Delhi Technological University

ABSTRACT

Authenticity is one of the key aspects of security. As a cryptographic technique that guarantees information authenticity, secret sharing has been an active research field for many years. With the efforts of pioneering researchers, secret sharing has reached the level of practical application use in today's environment, which is designed to protect a secret piece of information among a group of users in such a way that only certain subsets of users can jointly reconstruct the secret whereas other subsets of users can ideally not obtain any information about the secret. Splitting the secret ensures that no has the complete the secret. Also, as all shares are of the same length, it makes guessing impossible.

A well-known principle in the world is "reduced trust”, i.e. in order to maintain any secret, lesser the power or knowledge an entity have, greater are the chances of keeping that secret. This philosophy is implemented in the digital world as well. Thus the idea of verifiability was introduced in the secret sharing as well. Although there are several other means to check the verifiability of the shares, in this project, the pioneer secret sharing scheme authored by Adi Shamir is studied for verifiable secret sharing (VSS) using RSA. In this verifiable secret sharing scheme (VSSS), the secret is shared among the set of participants after signing the shares with their private key, such that only specific subsets of the participants, after verifying their shares with the public keys, can successfully recover the secret at a later time using their private keys. Both Shamir’s secret sharing scheme and RSA are unconditionally secure for both the secret as well the shares; together they complement each other in the VSSS.

Abbreviations

MSSS	Multi Secret Sharing Scheme
PKC	Public Key Cryptosystem
PKI	Public Key Infrastructure
PVSS	Publicly Verifiable Secret Sharing
RSA	Rivest-Shamir-Adleman Cryptosystem
SS	Secret Sharing
SSS	Secret Sharing Scheme
SSSS	Shamir Secret Sharing Scheme
TCVSS	Threshold Changeable Verifiable Secret Sharing
VMSS	Verifiable Multi-secret Sharing
VSS	Verifiable Secret Sharing
VSSS	Verifiable Secret Sharing Scheme

List of Figures

Page No.

Table of Contents

Chapter 1
Introduction

1.1 Introduction to Secret Sharing

The security of operations taking place over a computer network is very important. It is necessary to protect such actions against bad users who may try to misuse the system. There are many schemes and cryptographic tools to solve this problem. Secret sharing schemes (SSS) act as a tool for safekeeping of highly sensitive information such as encryption keys etc. as this information can neither be lost nor exposed. Traditional cryptographic methods are not suited for achieving high levels of reliability and confidentiality simultaneously. That's so because there is always a dilemma in up keeping of the encryption key - whether to keep a single copy at one location (maximum secrecy) or to keep multiple copies at different locations (greater reliability). Increasing the reliability of the key copies lowers the secrecy/confidentiality because keeping multiple copies adds the attack vectors. Also out of many copies, there is always a risk for copy leaks. Secret sharing schemes offer a solution to this dilemma by providing high levels of confidentiality and reliability by providing the shares that can be distributed to servers using a threshold mechanism. The key can be reconstructed whenever there is a need. Secret sharing (SS) can also be used for sensor networks by sending the data as shares over the links to make the job of an eavesdropper harder.

1.2 Essentials of Secret Sharing

Secret sharing is a technique for sharing a secret among a group such that each shareholder has an allocated share of the secret. The secret can be reconstructed whenever a distinct predefined number of shares are combined together whereas standalone any individual share does not yield anything to any member of the group. The access structure has the predetermined groups of shareholders that are enabled to reconstruct the secret.

A secret sharing group consists of a dealer and shareholders. The dealer splits the secret into shares depending upon the number of shareholders **n**. A share is provided to each shareholder within the group such that any group of threshold **t** or more shareholders can successfully reconstruct the secret. However any group of fewer than **t** shareholders cannot. This is bounded on all shareholder group members and is the primary condition for the secret sharing. This is known as (t, n) threshold secret sharing scheme which is sometimes is also referred as an (n, t) threshold secret sharing scheme.

The two guarantees required from secret sharing schemes are:

1) Recoverability: Secret can be reconstructed with any threshold **t** number of shares.
2) Secrecy: Absolutely nothing could be learned about the secret with any threshold <**t**.

Adi Shamir[1] and George Blakley[2] independently introduced the concept of secret sharing and its safeguarding in 1979.

1.3 Various Secret Sharing Schemes

In secret sharing, the dealer is assumed to be an honest entity. Shamir's SS scheme is also based on this assumption. But anyone from the shareholder's group can be affected by an adversary. Or worst the dealer may turn dishonest. Overall there was no mechanism to check or verify what is shared between the shareholders within the group, as they were all just the receivers. Thus there was a need to incorporate the notion of verifiability in the secret sharing schemes. There are several types of VSS schemes which focus on different aspects of secret sharing.

1.3.1 Verifiable Secret Sharing (VSS)

When some additional feature is included in a secret sharing scheme to allow the shareholders to verify the consistency of their shares then it is called verifiable secret sharing. VSS was introduced to counter the malicious dealer and shareholders who can send false shares to the other shareholders. A few VSS schemes can detect the cheater while a few can also identify the cheater. The concept of introducing verifiability in the secret sharing scheme i.e. verifiable secret sharing (VSS) was given by Benny Chor, Silvio Micali, Shafi Goldwasser and Baruch Awerbuch in 1985.

1.3.2 Publicly Verifiable Secret Sharing (PVSS)

In this scheme, anyone from the secret sharing group can perform the verification of the shares without gaining any information about the secret or any share. It is assumed that there is no private communication between the dealer and the shareholders and all communication is public as in public key infrastructure (PKI) so as to ensure that verifiability of the shares[19]. Though most PVSS schemes do reveal some information regarding the shares in their verification procedure and do not achieve the strict honest verifier zero knowledge policy[18]. This scheme must satisfy the following properties:

- **Correctness**: If the dealer and the shareholders act honestly, each and every qualified threshold subset of shareholders can reconstruct the secret during the reconstruction process.

- **Privacy**: The adversary cannot get any information about the secret if the dealer is acting honestly.

- **Verifiability**: There exist a unique number of honest shareholders in any qualified threshold of shareholders with at least t honest shareholders that can reconstruct the secret even if the dishonest dealer goes undetected during verification. The participants cannot cheat without being detected while reconstruction.

1.3.3 Verifiable Multi-secret Sharing (VMSS)

In a verifiable (k, t, n) multi-secret sharing scheme (VMSS) a dealer shares **k** secrets among **n** shareholders with **t** threshold[20]. For this, the dealer needs to use the (t, n) secret sharing scheme **k** times, if **k** secrets are to be shared among **n** shareholders but it would result in low efficiency. To solve the problem, Jackson et al. [21] extended the (t, n) secret sharing scheme to the multi-secret sharing scheme and named it (k, t, n) multi-secret sharing scheme (MSS), which is both robust (**k** secrets can be recovered when **t** or more shareholders are pooled) and confidential (no secrets can be recovered when **t-1** or fewer shareholders are pooled). For VMSS, the shares provided by the dealer or the shareholder must be verifiable. VMSS can be used in many applications such as access control.

1.3.4 Threshold Changeable Verifiable Secret Sharing (TCVSS)

Threshold Changeable Verifiable Secret Sharing scheme addresses the problem of changing (mostly by increasing) the threshold **t** of a secret-sharing scheme after the share distribution phase without any communication between the shareholders and the dealer. Some solutions to this problem require a non-standard scheme designed specifically for this purpose while some need to have communication between the shareholders or the dealer. Whereas some solutions increase the threshold parameter **t** of the existing Shamir SS schemes even if they were setup without considering threshold change in future. One such solution is based on lattice reduction algorithms on lattice-based list decoding of Reed-Solomon codes with noise bounded in Lee norm[22]. A better solution would always allow anytime dealer-free threshold changeability.

1.3.5 Verifiable Secret Sharing with Dealer Leakage Resilience

This is relatively a new concept in verifiable secret sharing. While VSS captures a particular malicious behavior of the dealer a few other dishonest strategies exist like a leakage/dishonest dealer that subliminally hide information in the valid shares on the basis of an implemented strategy by or with the adversary prior to the VSS execution. In this scheme, it assumed that a dealer is subliminally sending information invalid shares, that provides an advantage to the attacker for secret reconstruction. This may seem strange as it can be argued that the dealer knows the secret and can simply send it over a separate communication channel which is outside the VSS settings it will lead to dealer's discard. To keep its malicious behavior from everyone except the attacker, the dealer's action must remain unidentified to everyone else. Thus the concept of Dealer-Leakage Resilience Verifiable Secret Sharing (DLR-VSS) was introduced[17] to counter both the verifiability of shares and the dealer-leakage resilience.

1.4 Goal, Scope and Objective of Research

The goal of research for this thesis is to study and extend the features of PKI to the secret sharing schemes. RSA is one best known, secure, most widely used as well accepted algorithm in PKI. Using the proven secure RSA, the notion of verifiability of shares in the secret sharing scheme is studied. Though the thesis is started with this goal, other known problems encountered with other VSS schemes as well as any solution for them will also be discussed. A few of the problems discussed in all VSS schemes are share verification, share validation, cheater detection, cheater identification, dealer leakage etc will also be tried for a solution using this scheme.

1.5 Thesis Structure

The thesis is divided into five chapters.

Chapter 1 describes the introduction to secret sharing. Here we discuss the essentials of secret sharing, various schemes of secret sharing etc. This chapter also focuses on the goal, scope, and objective of this study.

Chapter 2 describes the work related to the various verifiable secret sharing schemes including their challenges.

Chapter 3 describes the proposed work to overcome the challenges in verifiable secret schemes most importantly cheater identification and the cheated shareholder.

Chapter 4 contains details of tools and software used for implementing the VSS using RSA. The various modules used for share generation, share verification, share validation and secret reconstruction are discussed.

Chapter 5 describes the conclusions and future work.

Chapter 2
Related Work

2.1 Verifiable Secret Sharing Schemes

The verifiability plays an important role in the SS scheme. A verifiable secret sharing scheme (VSSS) enables all shareholders to verify whether their shares are t-consistent without revealing the secret and the corresponding shares. In SS involving multiple dealers, the property of verifiability is more desirable as these dealers are usually mutually distrusted. In Shamir's secret sharing scheme (SSSS) we assumed that the dealer is reliable, however, a misbehaving dealer can deal inconsistent shares to the participants, from which they will not be able to reconstruct a secret. To prevent such malicious behavior of the dealer, a methodology for consistent verification by the shareholders was needed. The problem of VSS is to convince shareholders that collectively their shares are, **t-consistent**, meaning that every subset of ***t*** shares out of ***n*** (dealer distributed) defines the **same** secret. In Shamir's SSS the distributed shares are said to be t-consistent when the interpolation of the points ((1, *f*(1)), (2, *f*(2)), (3, *f*(3)),..........,(n, *f*(n))) an at most ***t-1*** degree polynomial. If the shareholders would transfer their shares, they could easily confirm consistency however, this would contradict the purpose of the secret sharing scheme. In the existing VSSS, it is assumed that shareholders may be corrupted by an adversary.

The verifiable secret sharing schemes are broadly classified as follows:

- **Interactive proofs**
- **Non-Interactive proofs**

Both versions allow the validity of secret shares to be verified without their secret being revealed; a shareholder can obtain high confidence that it holds a valid share of the secret rather than a useless random number.

Both proofs uses the following property:

- If the sum of two polynomials is of degree at most ***t-1***, then either both are of degree at most ***t-1*** or both are of degree greater than ***t-1***.

A simple, but incorrect, solution might be:

- The Dealer chooses an additional random polynomial of degree ***t-1***: *P(x).*

- The Dealer will prove that the random polynomial *P(x)* is of degree ***t-1***.
- The Dealer will prove that the sum of the secret polynomial: *f(x)* with the random polynomial: *P(x)* is of degree ***t-1***.

However, the drawback here is that the Dealer reveals the random polynomial as well as the sum of the secret polynomial and random polynomial hence we can determine secret polynomial and reveal the secret.

2.1.1 VSS: Interactive Proof (Benaloh Scheme)

Benaloh presented a notion of **t-consistency** to convince whether shares are verifiable or not. There are two different interactive proofs for VSS. In the first scheme, it is assumed that the shareholders do not cheat. In the second scheme it is assumed that the shareholders cheat[6].

- **Trusted Shareholders, Untrusted Dealer**

<u>The scheme is as follows</u>:

1. The Dealer uses the Shamir's SSS for a secret ***s*** and creates *f(x)* where
$$f(x) = s \qquad \text{.......................} (1)$$
and distributes the shares: $f(1),........f(n)$; one for each shareholder.
2. The Dealer chooses many (say 100) random polynomial of degree ***t-1***: $P_1(x),.........,P_{100}(x)$.
The Dealer commits himself to the polynomials by distributing to every i^{th} shareholder 100 shares, one of each polynomial: $P_1(i),.........,P_{100}(i),\ i \in [1......n]$.
3. The shareholders choose 50 random indices from $[1.....100]$, say $\{j1\ ...j50\}$, and ask the Dealer to reveal the random polynomials: $P_{j1}(x),.........,P_{j50}(x)$.
4. The Dealer reveals the polynomials in the chosen subset. It is assumed that all the shareholders receive the **same** polynomials (Using the mechanisms by which reliable broadcast by a dealer may be enforced).
5. The shareholders verify that the distributed shares of the revealed polynomials define polynomials of degree ***t-1***. As for now, the Dealer proved, with high

probability, that **all** the **shares** of all the polynomials: $P_1(x),.........,P_{100}(x)$ define polynomials of degree **t-*1***.

6. The Dealer reveals the sum of the share with its polynomial indices as given in *equation 2* below

   $$f(x) + P_{j51}(x),........., f(x) + P_{j100}(x) \qquad .... (2)$$

   where $\{j51 ...... j100\}$ are the remaining indices.

7. The shareholders verify that the *equation 2* given above is of degree ***t-1*** and correspond to their own shares, e.g. the i^{th} shareholder verifies, using the homomorphic property, by calculating the equivalence of the sum of their corresponding shares (LHS of *equation 3*) with that of the revealed polynomial sum (RHS of *equation 3*) in the i^{th} coordinate (by substituting the i^{th} coordinate in the revealed polynomial's sum) for all $t \in \{j_{51}...... j_{100}\}$ as given below

   $$f(i) + P_t(i) = f(x) + P_t(x) \qquad .... (3)$$

But if a conflict occurs, we cannot determine who is cheating: the Dealer or one of the shareholders.

- **Untrusted Shareholders, Untrusted Dealer**

There is no assumption on the honesty of either the Dealer or the shareholders. The idea is simple: the Dealer will commit by encryption. Instead of delivering the shares (of the 100 random polynomials) to every shareholder, the Dealer will **encrypt** all the shares and then distribute them all. Two additional requirements that the scheme requires are:

1. The encryption algorithm should have the below homomorphic property:

   $$E(x + y) = E(x) * E(y) \qquad .... (4)$$

2. The Dealer should use a secure broadcast while publishing the encrypted shares.

<u>The scheme is as follows</u>:

1. The Dealer uses the Shamir's SSS for a secret ***s*** and creates *f(x)* using *equation 1* (similar to the previous trusted shareholder, untrusted dealer scheme) and distributes the **encrypted** shares.

2. The Dealer chooses many (say 100) random polynomials of degree *t-1*: $P_1(x),.........,P_{100}(x)$. The Dealer commits himself to the polynomials by publishing the $100*n$ encrypted shares, i.e. 100 encrypted shares for each polynomial:

   $E(P_1(1),.........,E(P_{100}(1))$

   ……………………

   $E(P_1(i),.........,E(P_{100}(i))$

   ……………………

   $E(P_1(n),.........,E(P_{100}(n))$

3. The shareholders choose 50 random indices from $[1.....100]$, say $\{j1\ ...j50\}$, and ask the Dealer to reveal the random polynomials: $P_{j1}(x),.........,P_{j50}(x)$.
4. The Dealer reveals the polynomials in the chosen subset.
5. The shareholders will verify

   $$g^{Pw(i)} = E(Pw(i)) \qquad \text{........................ } (5)$$

   for all $w \in [j1......j50]$ and for all $i \in [1......n]$.
6. The Dealer reveals the sum of the share with its polynomial indices as given in *equation 2*, where $\{j51\ ......\ j100\}$ are the remaining indices.
7. Each i^{th} (*i*[1......n]) shareholder, using the homomorphic property (as in *equation 4*), verifies

   $$g^{f(x)+Pt(x)} = E(f(i)) \ *E(Pt(i)) \qquad \text{........................ } (6)$$

in the i^{th} coordinate, for all threshold $t \in \{j_{51}......j_{100}\}$.

Disadvantage of Interactive Proofs

On the basis of the above discussion, a few disadvantages of interactive proofs in VSS are:

1. Such an interactive proof asserts a proof only to the participants of the scheme, and that too only at the moment it is held.

2. These proofs have no meaning for a shareholder who is not online and does not participate in the random selections. As a result, these proofs are not valid to a third party.

2.1.2 VSS: Non-Interactive Proof (Feldman Scheme)

Contrary to the previous schemes, in a Non-Interactive Proof scheme[14], only the dealer is allowed to send messages, in particular, the shareholders cannot talk with each other or with the dealer when verifying a share. The basic idea is that the dealer sends extra information to each participant during the distribution and each participant verifies that his/her secret share is consistent with this extra information.

There is an additional requirement that the scheme requires:

- The homomorphic property both with respect to addition and multiplication shall be a part of the encryption algorithm:

  $$E(x + y) = E(x) \oplus E(y) \qquad \text{........................ } (7)$$

  $$E(x * y) = E(x) \otimes E(y) \qquad \text{........................ } (8)$$

  An example of an encryption algorithm that complies with this property is the Diffie-Hellman encryption.

<u>**The scheme is as follows**</u>:

1. The Dealer uses the Shamir's SSS for a secret **s** and creates *f(x)* using *equation 1* (similar to the previous trusted shareholder, untrusted dealer scheme), where $f(0)$ is the secret **s** (i.e. a_0):

   $$f(x) = a_0 + a_1 x + \text{ .... } + a_{t-1} x^{t-1} \qquad \text{....................... } (9)$$

   and $f(1), \text{.........} f(n)$ and are the shares for each individual shareholder. *A*dditionally, all the ***t*** coefficients are published by the dealer after encryption: $E(a_0), \text{.........}, E(a_{t-1})$.

2. Each i^{th} , $i \in [1...n]$ shareholder can verify its share as:

   $$E(f(i)) = E(a_0) \oplus (E(a_1) \otimes E(i^1)) \oplus \text{.........} \oplus (E(a_{t-1}) \otimes E(i^{t-1})) \qquad \text{........ } (10)$$

This is possible due to the **homomorphic** and the **associative** properties for both addition and multiplication.

3. If the share verification is successful, then the i^{th} shareholder broadcasts a message regarding its share verification. If all the shares are verified then the dealing phase is said to be completed successfully. If the share verification for any k^{th} shareholder fails, then that k^{th} shareholder broadcasts an accusation against the dealer. Other honest shareholders can decide about the culprit - dealer or the accuser, and act as per their rules.

Note that it is also possible to verify that $f\left(0\right) = s$*, if the secret* ***s*** *is known.*

The VSS enables the honest shareholders to identify dishonest shareholders if a minority of shareholders has been corrupted. Though, the honest shareholders can still participate in secret reconstruction. Thus VSS is a fundamental cryptography tool in secure multi-party computation and Byzantine agreement. There are several papers that address the optimal round complexity of VSS [5], multi-secrets VSS, the Byzantine agreement against the adversary etc.

A strong VSS can ensure that:

- all the shares are t-consistent;
- all the shares satisfy the security requirements of a secret sharing scheme.

The scheme was originally proposed by Pedersen[7] in 1998. This study of (n,t,n) SSS is information-theoretically secure similar to the Shamir's (t,n) SSS and thus satisfies the definition of a strong VSS.

Disadvantage of Non-Interactive Proofs

On the basis of the above discussion, a few disadvantages of non-interactive proofs in VSS are given below:

1. Since this type of VSS can tolerate corrupted shareholders in the secret reconstruction, the verification process is a bit complicated.
2. If the number of corrupt shareholders are in majority, then the secret reconstruction will fail.

3. It is very expensive (i.e., a private channel for exchange information between every pair of shareholders).

2.2 Problem with Existing Schemes

Verifiable secret sharing (VSS) is an important primitive based on a very old property of homomorphic commitments where a dealer shares a secret among **n** shareholders where it is assumed that at most **t** of them are being controlled by an adversary. But the homomorphic nature of the commitments usually offers the weak guarantees because homomorphism is not inherent to commitments. Therefore the utility of the homomorphic commitments in VSS needs to be re-analyzed and thus they are not a necessity for VSS in any type of communication setting - synchronous or asynchronous. While almost all VSS schemes can verify the shares and fairly detect the presence of a cheater in the shareholder's group, which is also the prime property of VSS scheme. However the problematic areas for some VSS schemes are:

- Identification of the cheater
- Identification of the cheated member (in non-interactive VSS schemes)
- Cheating by the dealer

Chapter 3
Proposed Work

3.1 Problem Statement

Verifiable secret sharing is based on the property of homomorphic commitments, either with respect to addition or multiplication or with respect to both addition & multiplication. But the homomorphic nature of the commitments offers the weak guarantees because homomorphism is not inherent to commitments. Therefore the emphasis on the homomorphic commitments in VSS needs to be re-analyzed and whether they are a necessity for VSS in any type of communication setting - synchronous or asynchronous. While all VSS schemes can verify the shares and fairly detect the presence of a cheater in the shareholder's group, essentially a VSS scheme shall provides all or at least some of the following facilities depending upon their trade-off factors for efficiency or application area or simply the lack of it:

- Share Verification
- Share Validation
- Cheater Detection
- Cheater Identification
- Dealer Dishonesty
- Dealer-Leakage resilience

Any VSS which implements all of the above is considered a good VSS scheme irrespective of the complexity or implementation challenges.

3.2 Proposed Solution

In this study, an approach for getting an efficient verifiable secret sharing scheme based on Shamir's (*t*, *n*) SSS using RSA is studied and implemented. RSA is used for transmitting the shares over a proactive network while the MD5 is used for share validation. Share validation will help in the cheater identification which is the main problematic area in many VSS schemes. Share validation will also lead us to the actual defaulter and the affected member. This implementation of verifiable secret sharing scheme using RSA consists of the following modules:

- Share generation from secret using Shamir's (*t*, *n*) SSS
- Share encryption
- Share decryption and verification
- Share validation and Secret reconstruction

Share generation using Shamir's (t,n) SSS:

The dealer from the shareholder's group uses the Shamir's (*t*, *n*) SS scheme to split the secret into shares. The dealer generates and publishes the hash values of all the generated shares using the MD5 algorithm. The dealer is committed to hashing and publishing the hash values of the generated shares before encrypting them.

Share encryption:

After hashing all of the generated shares, the shares are encrypted before distribution among the shareholders. The dealer uses the public key of the respective recipient shareholder to encrypt its piece of share and then distribute the shares over the proactive network.

Share verification and decryption:

As the shareholder will always receive its share encrypted with its public key, the shareholder can decrypt its received share using its private key or secret key. All the shareholders in the group are committed for hashing their decrypted share and publishing the hash values of their respective shares just after decryption. This commitment is binding on all the shareholders within the group. This commitment is also a counter measure for discouraging the man-in-the-middle attack from within the shareholder's group. The shareholders can validate their received shares after decryption by matching the obtained hash value with the hash published by the dealer. Whenever there is a request/call for shares for reconstruction, the shareholder will provide its verified share by encrypting its share with the public key of the requestor shareholder. The ***t*-consistent** verified shares will be used for share validation and secret reconstruction.

Share validation and Secret reconstruction:

The share validation is performed with the help of hashing. The received encrypted shares are decrypted using the private key of the receiver shareholder that is provided over the proactive network. After the successful decryption of the received shares, the reconstructing shareholder is also committed to calculate the hash value of all the decrypted shares and compares the calculated hash values with the hash values published by the dealer and individual shareholders in the share generation and share verification phases. If all the decrypted shares are validated then the secret is reconstructed using the Shamir SSS. However, if the validation process fails then the corresponding share is discarded and the adversary is traced using the failed hash values.

3.3 Proposed Architecture

The proposed architecture as well as the functioning of the proposed architecture of VSSS using RSA is explained below:

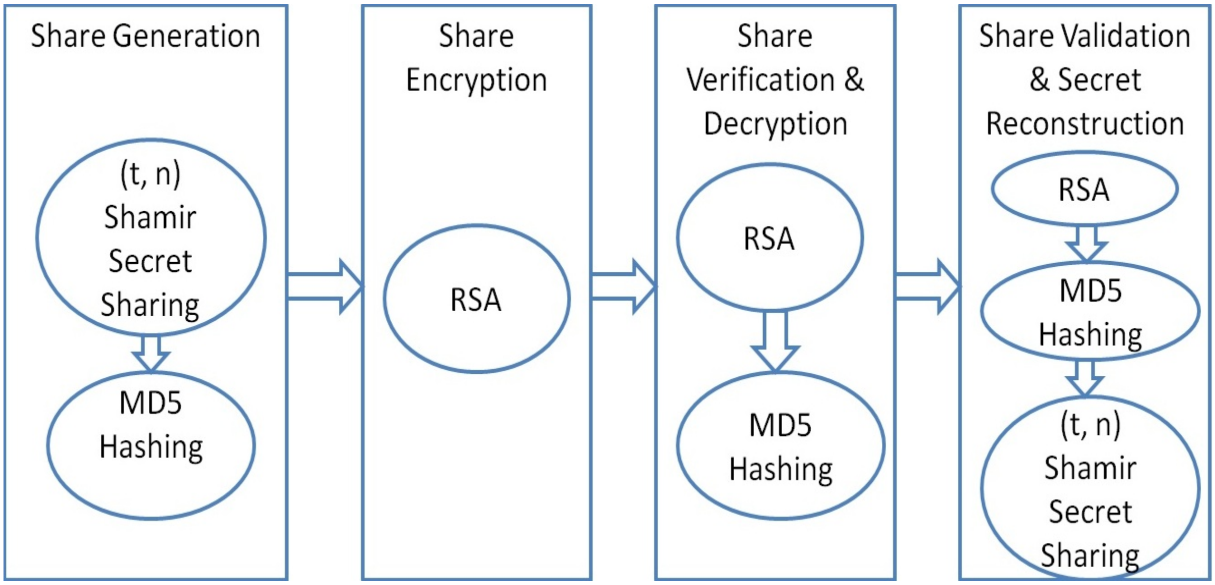

Share Generation
(t, n) Shamir Secret Sharing
MD5 Hashing
Share Encryption
RSA
Share Verification & Decryption
RSA
MD5 Hashing
Share Validation & Secret Reconstruction
RSA
MD5 Hashing
(t, n) Shamir Secret Sharing

Figure 1: Architecture of Proposed System of VSSS using RSA

The functioning of the architecture of the VSS using RSA is explained below.

The dealer uses Shamir's (*t*, *n*) SS scheme to split the secret into shares. The dealer also computes and publishes the hash values of all the generated shares using the MD5 algorithm as the dealer is committed for hashing the generated shares and publishing their hash values before encryption. This will constitute the share generation phase.

Then dealer uses the public key of the respective shareholder to encrypt its piece of share and distribute them after completing the hash operation. This will constitute the share encryption phase.

The shareholder decrypts its piece of encrypted share using its private key. At this phase, all the shareholders are committed for hashing and publishing the computed hash values of their decrypted shares immediately after share decryption. The shareholders can validate their shares by matching the computed hash value with the dealer published hash. Whenever there is a request/call for shares for secret reconstruction, the shareholder will provide its verified share by encrypting it with the public key of the requestor shareholder. This will constitute the share verification and decryption phase.

The share validation is performed with the help of hashing. The received encrypted shares are decrypted using the private key of the receiver shareholder that is provided over the proactive network. After the successful decryption of the received shares, the reconstructing shareholder is also committed to calculate the hash value of all the decrypted shares and compares the calculated hash values with the hash values published by the dealer as well as individual shareholders in the share generation and share verification phase respectively. If all the decrypted shares are validated then the secret is reconstructed using the Shamir SSS. However, if the validation process fails then the corresponding share is discarded and the adversary is traced using the failed hash values. This will constitute the share validation and secret reconstruction phase.

3.4 Flow Diagram

The flow diagram of the proposed VSSS using RSA system is explained below:

Share Generation

- ➢ Dealer generates the shares of the secret (SJ).
- ➢ Dealer hashes all the shares and publish their hash values.

Share Encryption

- ➢ Dealer encrypts the shares with the public key ($PubK_n$) of the respective shareholder (SH_n).

Share Decryption, Verification and Secret Reconstruction

- All shareholders decrypt the received encrypted share using their private key (PrvK_n).
- All shareholders perform the hashing of all decrypted shares and publish their result.
- All shareholders validate their shares by comparing their computed hash with the hash values published by the dealer.
- While secret reconstruction, the shareholder reconstructing the secret will receive the shares encrypted with its public key from the share provider shareholders. The receiver will decrypt the received shares using its private key and perform the hashing of all decrypted shares and compare their computed hash with the hash values published by the dealer as well by the individual shareholders. The validated shares are used for secret reconstruction (SJ).

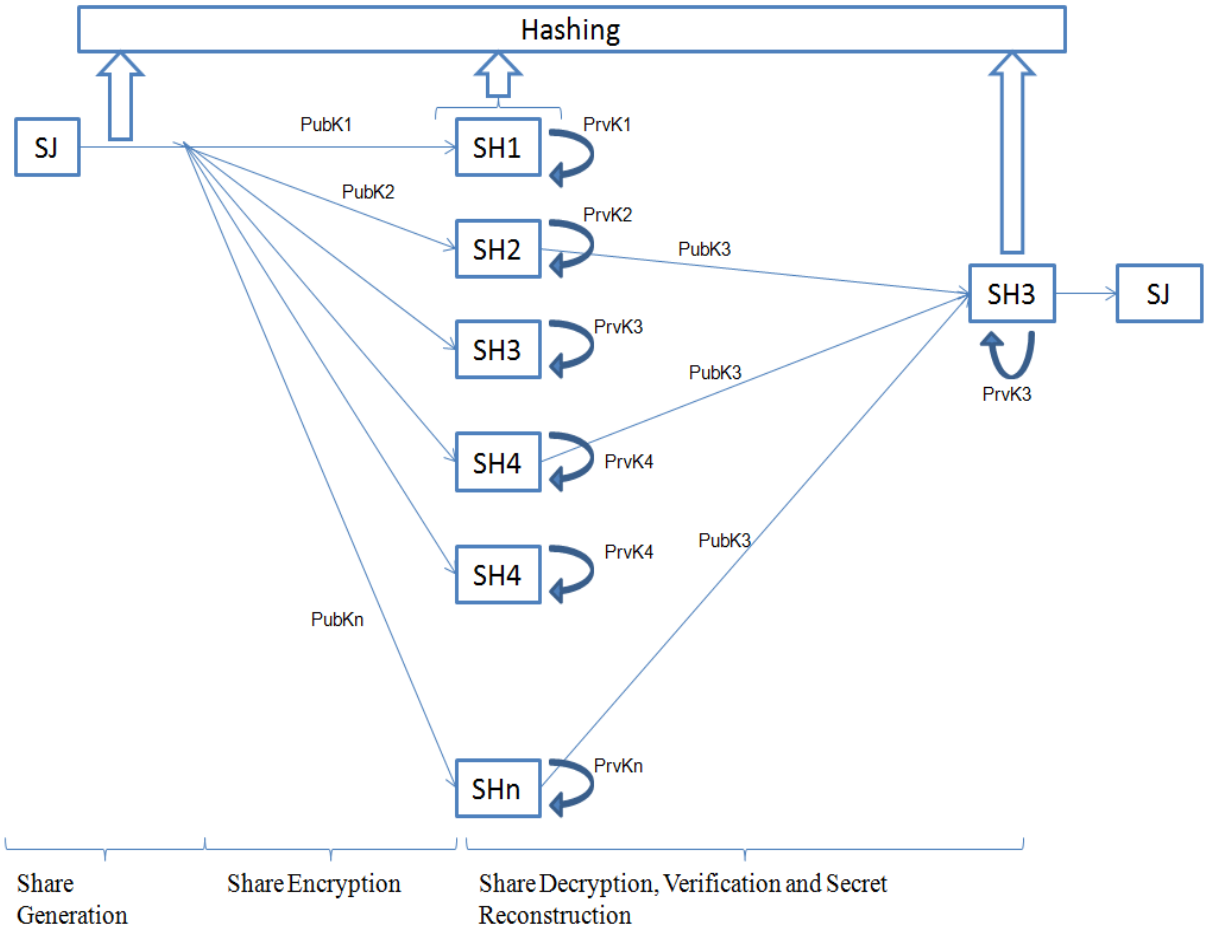


Figure 2: Flow Diagram of Proposed System

3.5 Algorithms Used

The following native algorithms are used in the implementation of proposed system. Their cumulative computational cost is also provided as asymptotic expression in Big-O notation, representing the worst case time complexity:

1. (t, n) Threshold Shamir Secret Sharing

   O(tn) = max {O(tn), O(t)}

   where O(tn) is the time complexity for the share generation & distribution and O(t) is the time complexity for secret reconstruction.

2. RSA Algorithm

   $O(log_2(N)^3)$ = max {$O(log_2(N)^2)$, $O(log_2(N)^3)$}

   where $O(log_2(N)^2)$ is the time complexity for encryption and $O(log_2(N)^3)$ is the time complexity for inverse operation & decryption.

3. MD5 Hashing Algorithm

   O(n) = max {O(n), O(1)}

   where O(n) is the time complexity for hashing operation and O(1) is the time complexity for hash matching or hash comparison.

Chapter 4
Implementation

4.1 Software Details of Implemented System

The proposed system implementation of verifiable secret sharing using RSA is emulated on a single computer using several tools. The details of the tools used for the proposed implementation of the system are as follows:

Operating System:	Windows 10
Database:	Microsoft SQL Server 2005
Programming Language:	C#.NET
Frameworks:	.NET 4.0

4.2 Pseudo Code

The notations used in the pseudo code are as follows:

- Secret (S)
- Shares of the secret (S_1.............S_n)
- Secret Splitting and Joining (SJ)
- Hashing (H)
- Shareholders (SH_n)
- Public Key ($PubK_n$)
- Private Key ($PrvK_n$)
- Encryption (E)
- Threshold (t)

The pseudo code for implementing the various phases of the proposed system is as follows:

I. Share Generation Phase

```
1. Use SJ to split S into $S_1$.............$S_{n.}$
```

2. H (S_1............S_n) and publish H for all S_1............$S_{n.}$

II. Share Encryption Phase

1. $E_{PubK1}(S_1)$......... $E_{PubKn}(S_n)$ for all S_1.........S_n and send to all SH_1......... SH_n.

III. Share Decryption, Verification and Secret Reconstruction Phase

Share Decryption

1. $E_{PrvK1}(E_{PubK1}(S_1))$......... $E_{PrvKn}(E_{PubKn}(S_n))$ to get (S_1............S_n) for all SH_1......... $SH_{n.}$
2. All SH_1......... SH_n performs H (S_1............S_n) and publish H for all S_1............$S_{n.}$

Share Verification

1. Compare H (S_1............S_n) from phase I and H (S_1............S_n) from phase III and all SH_1......... SH_n publishes them after comparison.

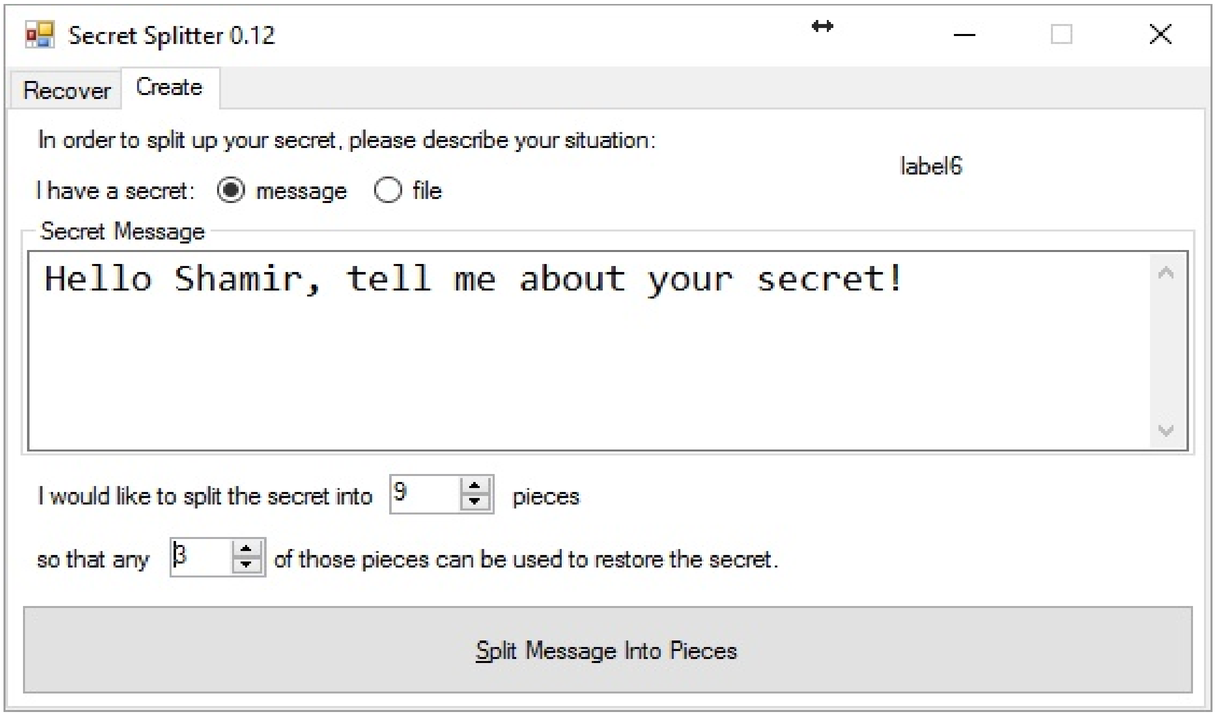
Secret Reconstruction

1. Any SH_n calls for **t** shares for secret reconstruction and receives E_{PubKn}(S_1.........S_n).
2. SH_n decrypts all received shares as E_{PrvKn}(E_{PubKn}(S_1.........S_n)).
3. SH_n performs H (S_1............S_n) and compare the computed values with H (S_1............S_n) from phase I and H (S_1............S_n) from phase III earlier.
4. Verified **t** (S_1............S_n) use SJ to get S.
5. Using SJ, non-verified **t** (S_1............S_n) traced for cheating.

4.3 Output of Working Model

In this section, the output of the implementation of the proposed system of VSSS using RSA is provided.

The below window is for secret splitting and secret reconstruction which is based on the Shamir's secret sharing scheme. Here we define the number of shareholders **n** and the threshold **t** value for secret reconstruction.



In order to split up your secret, please describe your situation:

I have a secret: ● message ○ file

Secret Message

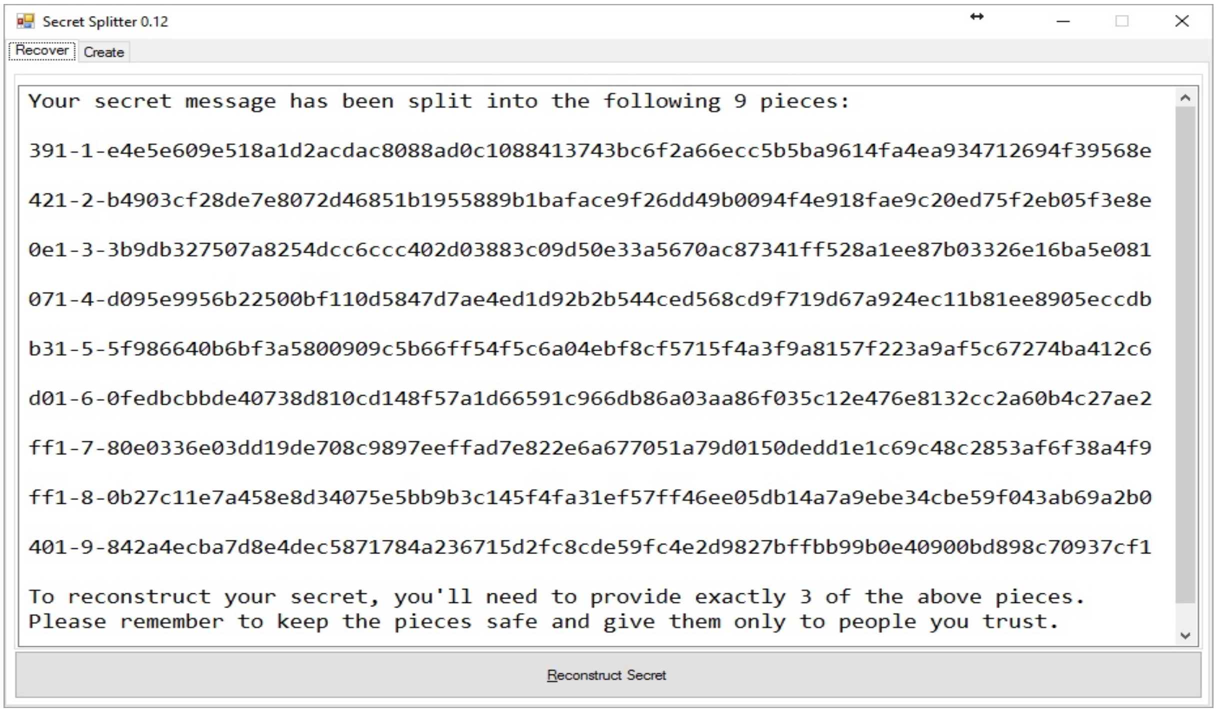
```
Hello Shamir, tell me about your secret!
```

I would like to split the secret into 9 pieces

so that any 3 of those pieces can be used to restore the secret.

Figure 3: Secret splitting using Shamir's SSS

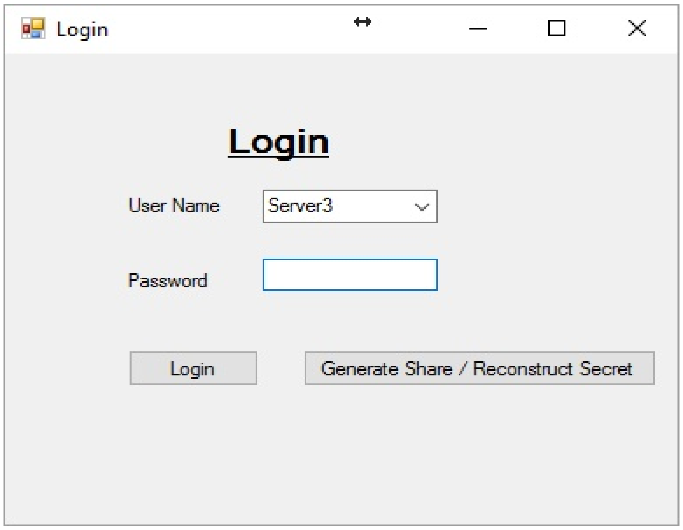
This window below shows the shares received after splitting the secret.



```
Your secret message has been split into the following 9 pieces:

391-1-e4e5e609e518a1d2acdac8088ad0c1088413743bc6f2a66ecc5b5ba9614fa4ea934712694f39568e

421-2-b4903cf28de7e8072d46851b1955889b1baface9f26dd49b0094f4e918fae9c20ed75f2eb05f3e8e

0e1-3-3b9db327507a8254dcc6ccc402d03883c09d50e33a5670ac87341ff528a1ee87b03326e16ba5e081

071-4-d095e9956b22500bf110d5847d7ae4ed1d92b2b544ced568cd9f719d67a924ec11b81ee8905eccdb

b31-5-5f986640b6bf3a5800909c5b66ff54f5c6a04ebf8cf5715f4a3f9a8157f223a9af5c67274ba412c6

d01-6-0fedbcbbde40738d810cd148f57a1d66591c966db86a03aa86f035c12e476e8132cc2a60b4c27ae2

ff1-7-80e0336e03dd19de708c9897eeffad7e822e6a677051a79d0150dedd1e1c69c48c2853af6f38a4f9

ff1-8-0b27c11e7a458e8d34075e5bb9b3c145f4fa31ef57ff46ee05db14a7a9ebe34cbe59f043ab69a2b0

401-9-842a4ecba7d8e4dec5871784a236715d2fc8cde59fc4e2d9827bffbb99b0e40900bd898c70937cf1

To reconstruct your secret, you'll need to provide exactly 3 of the above pieces.
Please remember to keep the pieces safe and give them only to people you trust.
```

Figure 4: Shares obtained after secret splitting

The window below is for individual login of the shareholders.

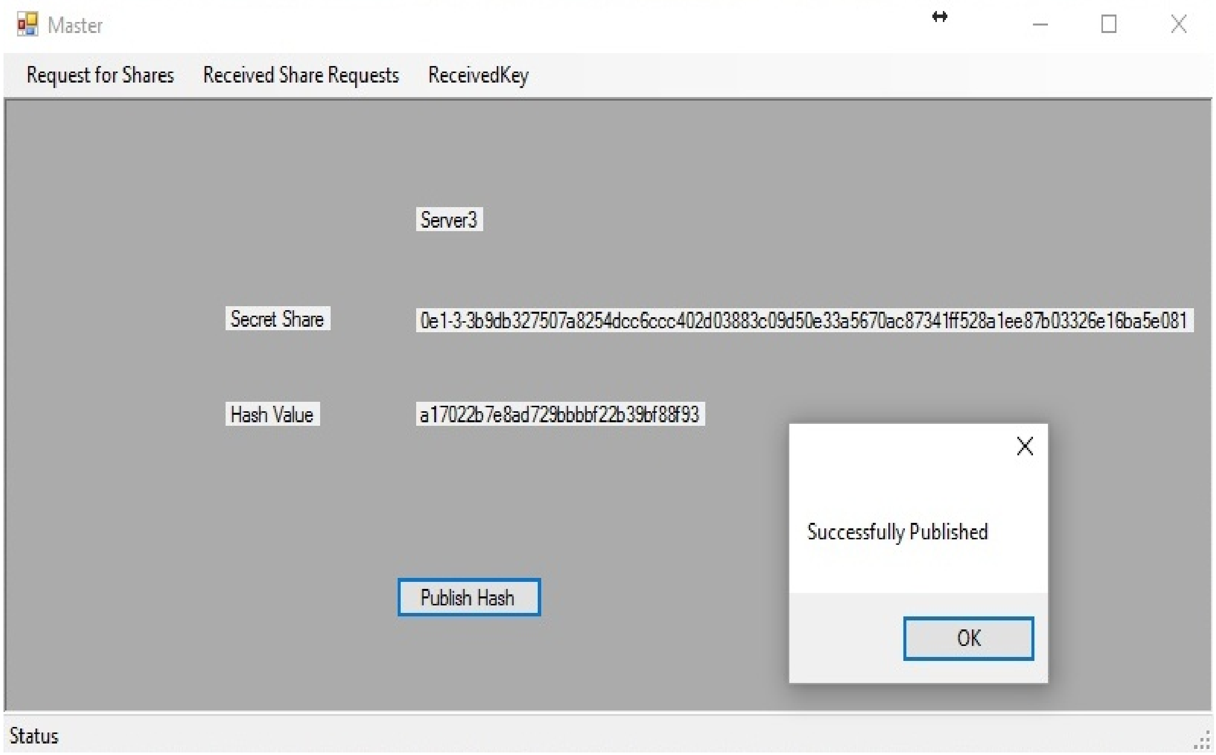


User Name Server3

Password

Figure 5: Individual shareholder login

Below is the individual shareholder's view of their secret. It decrypts the encrypted share provided by the dealer, computes the hash value of the obtained share and publishes the calculated hash.



Server3

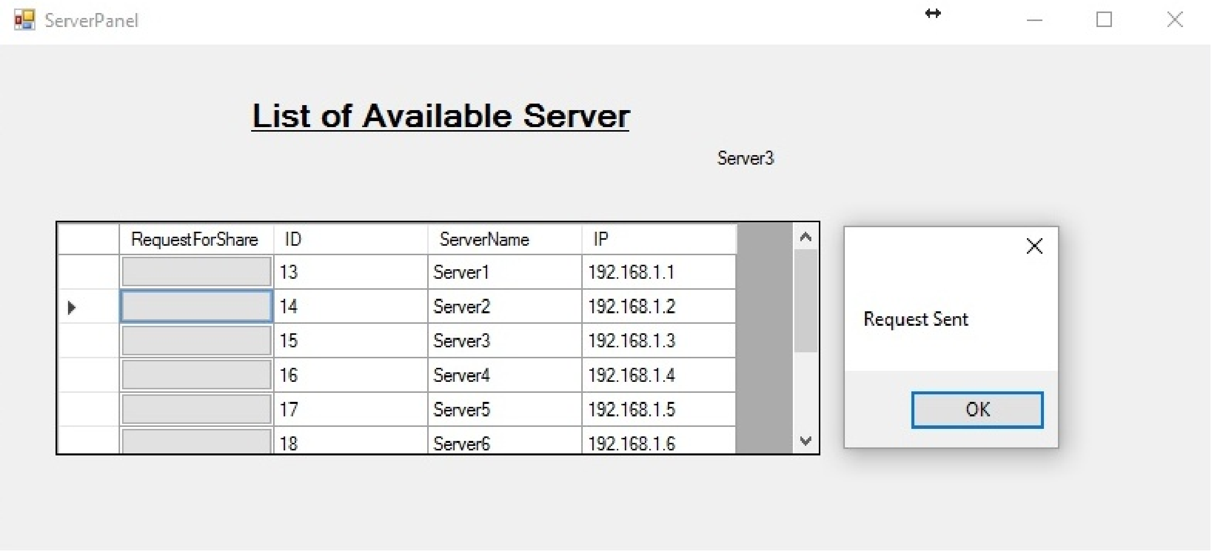
Secret Share 0e1-3-3b9db327507a8254dcc6ccc402d03883c09d50e33a5670ac87341ff528a1ee87b03326e16ba5e081

Hash Value a17022b7e8ad729bbbbf22b39bf88f93

Successfully Published

Figure 6: Individual shareholder can view his own share and performs hashing

Below is the window that shows all shareholders in the group from whom request for shares (for secret reconstruction) can be made.



ServerPanel

List of Available Server

Server3

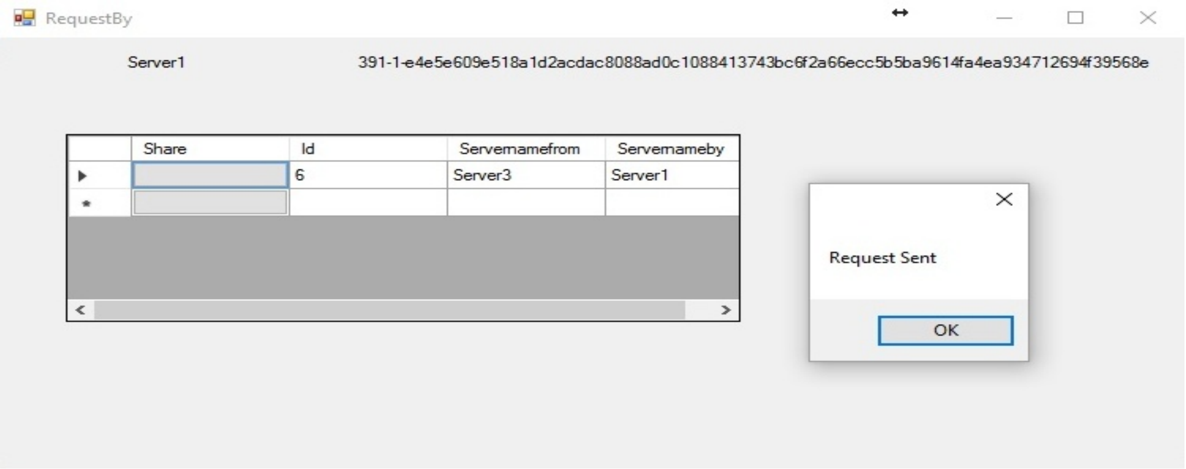
	RequestForShare	ID	ServerName	IP
		13	Server1	192.168.1.1
▶		14	Server2	192.168.1.2
		15	Server3	192.168.1.3
		16	Server4	192.168.1.4
		17	Server5	192.168.1.5
		18	Server6	192.168.1.6

Request Sent

Figure 7: Individual shareholder requesting for other shareholder's shares

* *Request for shares was sent to Server1, Server2 and Server9 from Server3.*

Below are the windows (figure 8, figure 9 and figure 10 respectively) show all requests for shares (requests received from other shareholders for secret reconstruction) received by any particular shareholder. It keeps the request for shares as requested by other shareholders for their secret reconstruction. The requested shareholder approves the individual request for a share using this window. E.g., the 3 figures below show the request for share that they have received from Server 1, Server2 and Server 9 respectively.



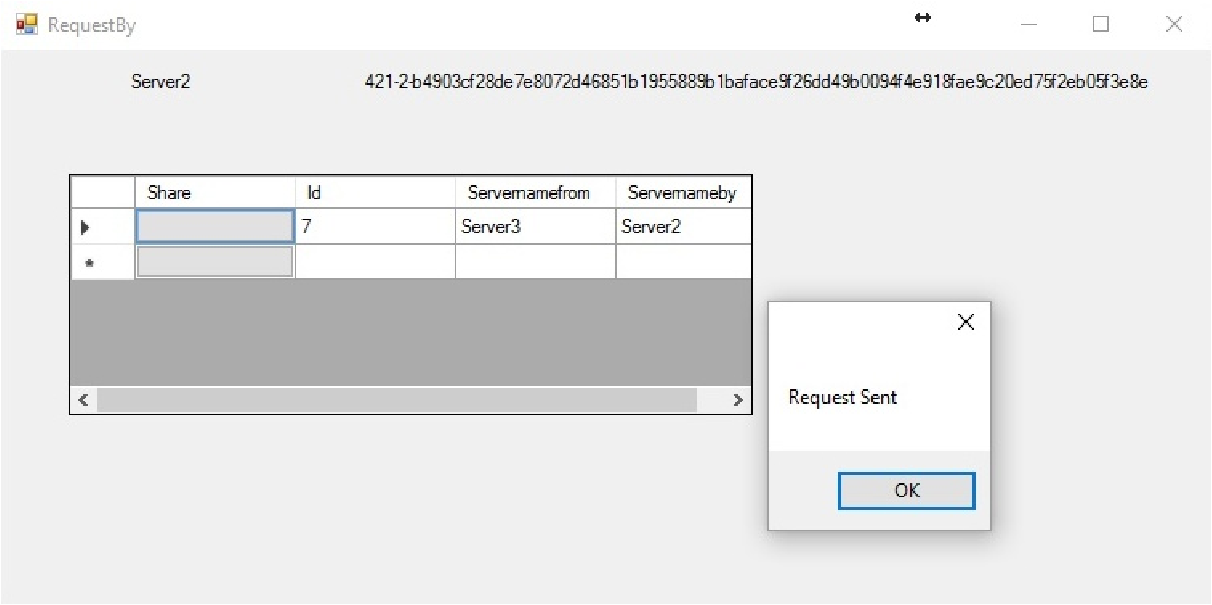
RequestBy

Server1 391-1-e4e5e609e518a1d2acdac8088ad0c1088413743bc6f2a66ecc5b5ba9614fa4ea934712694f39568e

	Share	Id	Servernamefrom	Servernameby
▶		6	Server3	Server1
*				

Request Sent

Figure 8: Requested shareholder approving the share's request 1

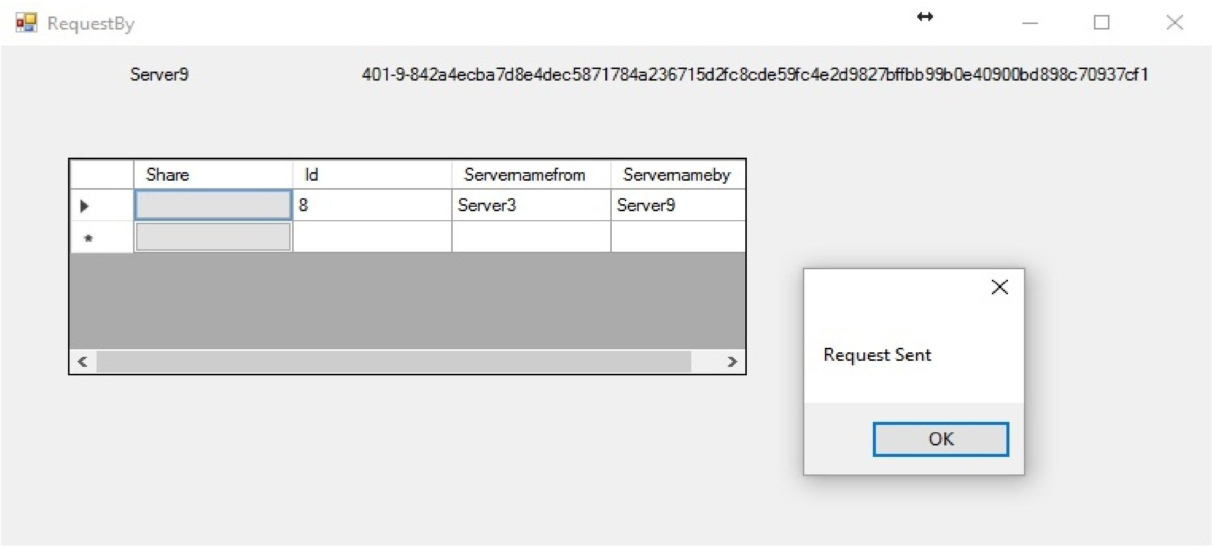


Server2 421-2-b4903cf28de7e8072d46851b1955889b1baface9f26dd49b0094f4e918fae9c20ed75f2eb05f3e8e

	Share	Id	Servernamefrom	Servernameby
▶		7	Server3	Server2
*				

Request Sent

Figure 9: Requested shareholder approving the share's request 2



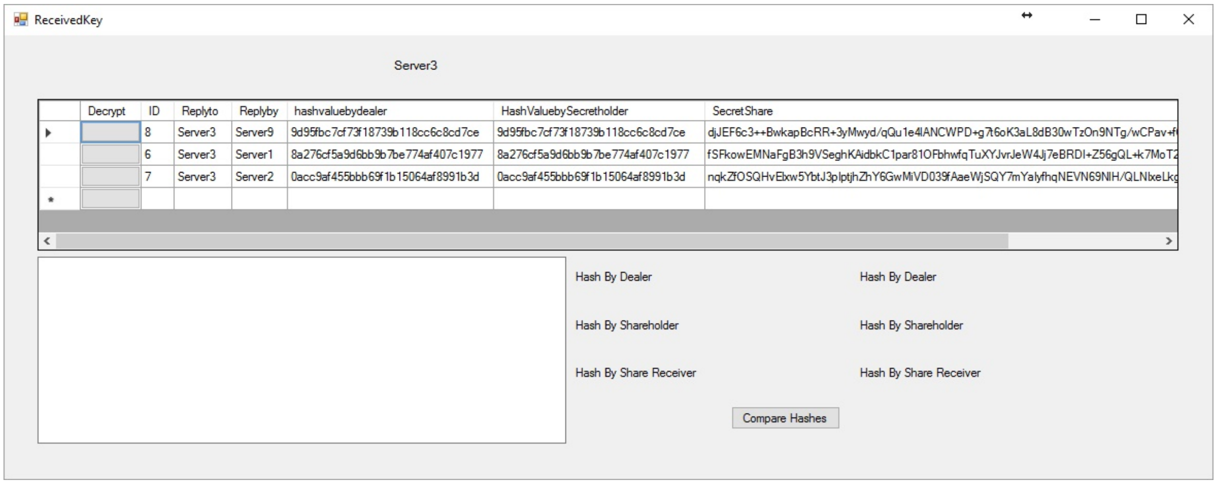
Server9 401-9-842a4ecba7d8e4dec5871784a236715d2fc8cde59fc4e2d9827bffbb99b0e40900bd898c70937cf1

	Share	Id	Servernamefrom	Servernameby
▶		8	Server3	Server9
*				

Request Sent

Figure 10: Requested shareholder approving the share's request 3

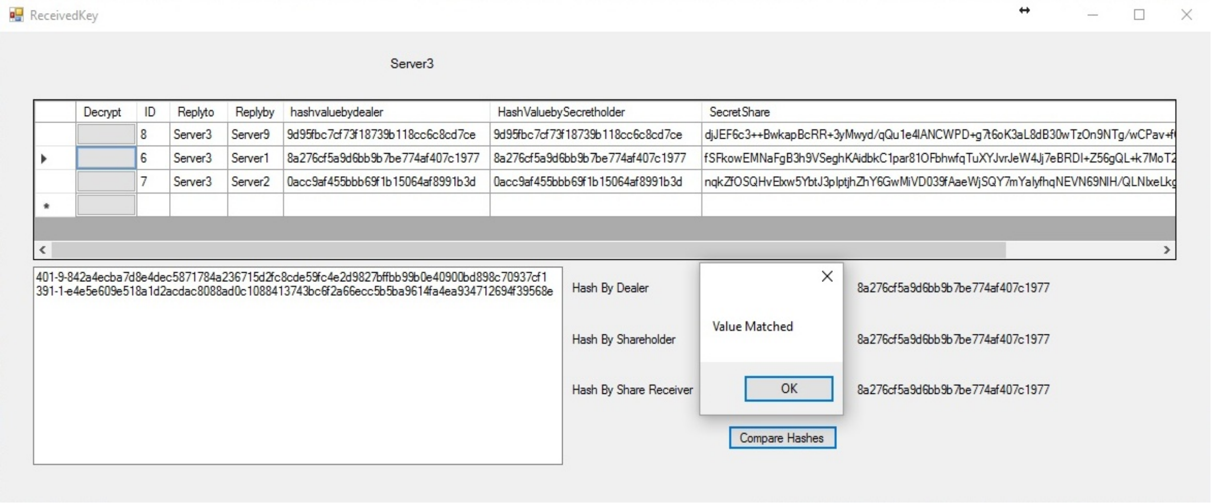
The windows below (figure 11 and figure 12) are the requestor shareholder's view, which shows all the shares provided by other shareholders for secret reconstruction. Here, the received shares are decrypted, hashed and also compared with the hash values published by the dealer and the hash values published by the individual shareholders. This part performs the validation of the shares. It can detect the cheater from the shareholder's group.



Server3

	Decrypt	ID	Replyto	Replyby	hashvaluebydealer	HashValuebySecretholder	SecretShare
▶		8	Server3	Server9	9d95fbc7cf73f18739b118cc6c8cd7ce	9d95fbc7cf73f18739b118cc6c8cd7ce	djJEF6c3++BwkapBcRR+3yMwyd/qQu1e4lANCWPD+g7t6oK3aL8dB30wTzOn9NTg/wCPav+f
		6	Server3	Server1	8a276cf5a9d6bb9b7be774af407c1977	8a276cf5a9d6bb9b7be774af407c1977	fSFkowEMNaFgB3h9VSeghKAidbkC1par81OFbhwfqTuXYJvrJeW4Jj7eBRDI+Z56gQL+k7MoT2
		7	Server3	Server2	0acc9af455bbb69f1b15064af8991b3d	0acc9af455bbb69f1b15064af8991b3d	nqkZfOSQHvEkw5YbtJ3pIptjhZhY6GwMiVD039fAaeWjSQY7mYalyfhqNEVN69NIH/QLNlxeLkg
*							

Figure 11: Requested shareholders provided their encrypted shares



Server3

	Decrypt	ID	Replyto	Replyby	hashvaluebydealer	HashValuebySecretholder	SecretShare
		8	Server3	Server9	9d95fbc7cf73f18739b118cc6c8cd7ce	9d95fbc7cf73f18739b118cc6c8cd7ce	djJEF6c3++BwkapBcRR+3yMwyd/qQu1e4lANCWPD+g7t6oK3aL8dB30wTzOn9NTg/wCPav+f
▶		6	Server3	Server1	8a276cf5a9d6bb9b7be774af407c1977	8a276cf5a9d6bb9b7be774af407c1977	fSFkowEMNaFgB3h9VSeghKAidbkC1par81OFbhwfqTuXYJvrJeW4Jj7eBRDI+Z56gQL+k7MoT2
		7	Server3	Server2	0acc9af455bbb69f1b15064af8991b3d	0acc9af455bbb69f1b15064af8991b3d	nqkZfOSQHvEkw5YbtJ3pIptjhZhY6GwMiVD039fAaeWjSQY7mYalyfhqNEVN69NIH/QLNlxeLkg
*							

```
401-9-842a4ecba7d8e4dec5871784a236715d2fc8cde59fc4e2d9827bffbb99b0e40900bd898c70937cf1
391-1-e4e5e609e518a1d2acdac8088ad0c1088413743bc6f2a66ecc5b5ba9614fa4ea934712694f39568e
```

Hash By Dealer: 8a276cf5a9d6bb9b7be774af407c1977

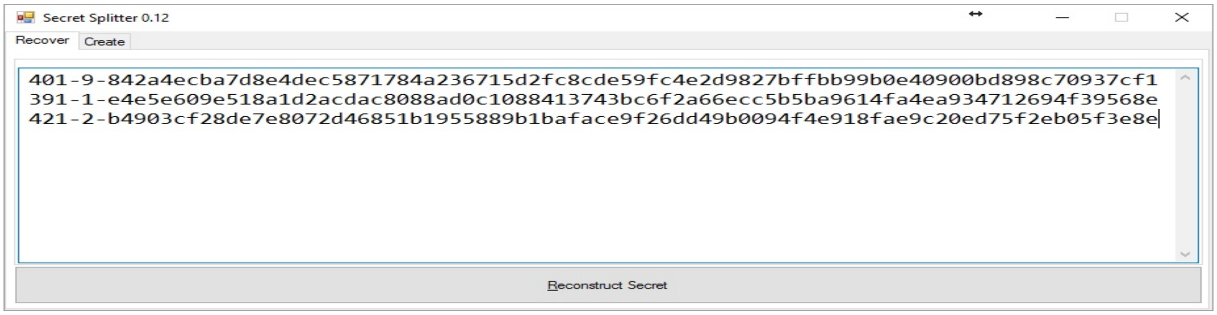
Hash By Shareholder: 8a276cf5a9d6bb9b7be774af407c1977

Hash By Share Receiver: 8a276cf5a9d6bb9b7be774af407c1977

Value Matched

Figure 12: Received encrypted shares are decrypted, hashed and hash compared

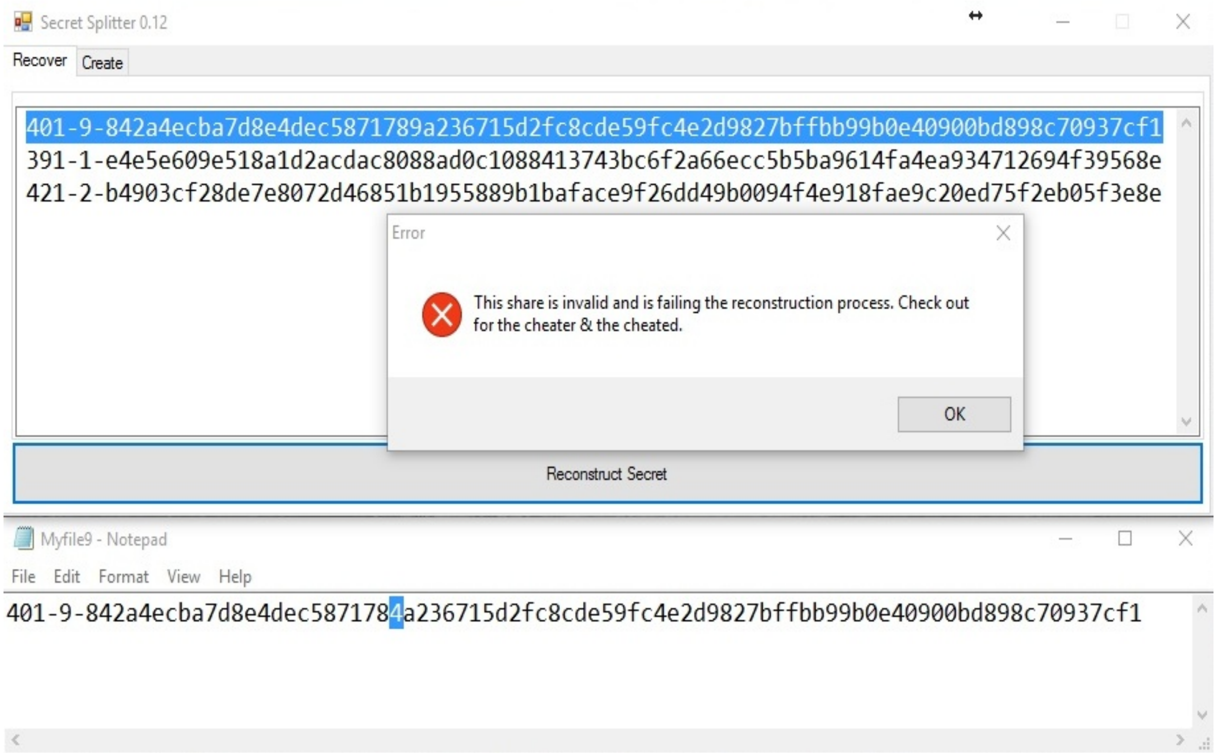
The below window is for secret reconstruction based on the Shamir's secret sharing scheme using validated **t** threshold shares for secret reconstruction which can be feed in any order.



```
401-9-842a4ecba7d8e4dec5871784a236715d2fc8cde59fc4e2d9827bffbb99b0e40900bd898c70937cf1
391-1-e4e5e609e518a1d2acdac8088ad0c1088413743bc6f2a66ecc5b5ba9614fa4ea934712694f39568e
421-2-b4903cf28de7e8072d46851b1955889b1baface9f26dd49b0094f4e918fae9c20ed75f2eb05f3e8e
```

Figure 13: Decrypted and validated shares are used for secret reconstruction

The below window shows that which share is failing the secret reconstruction process. The faulty share can be traced till the dealer because of the hashing.



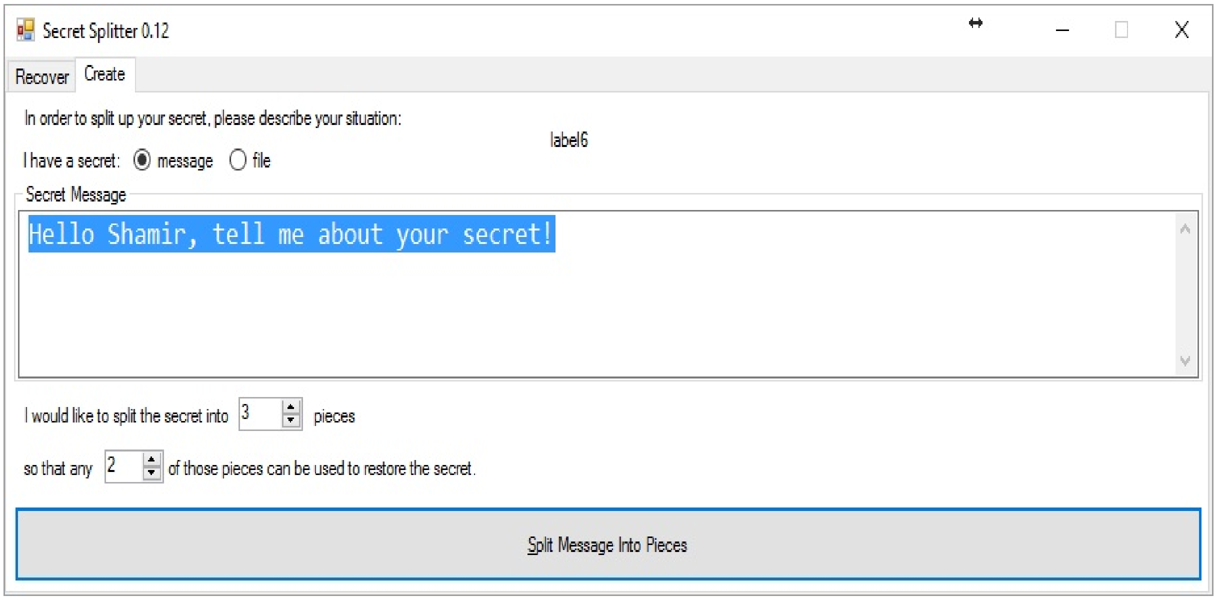
```
401-9-842a4ecba7d8e4dec5871789a236715d2fc8cde59fc4e2d9827bffbb99b0e40900bd898c70937cf1
391-1-e4e5e609e518a1d2acdac8088ad0c1088413743bc6f2a66ecc5b5ba9614fa4ea934712694f39568e
421-2-b4903cf28de7e8072d46851b1955889b1baface9f26dd49b0094f4e918fae9c20ed75f2eb05f3e8e
```

This share is invalid and is failing the reconstruction process. Check out for the cheater & the cheated.

```
401-9-842a4ecba7d8e4dec5871784a236715d2fc8cde59fc4e2d9827bffbb99b0e40900bd898c70937cf1
```

Figure 14: Cheater and cheated identification on the basis of invalid share

The window below shows the successful secret reconstruction using the validated shares.



In order to split up your secret, please describe your situation:

I have a secret: ● message ○ file

Hello Shamir, tell me about your secret!

I would like to split the secret into 3 pieces

so that any 2 of those pieces can be used to restore the secret.

Figure 15: Reconstructed secret

4.4 Performance Analysis of Some VSS Schemes

	Share Generation	Share Verification	Secret Reconstruction
VSS based on CRT[10]	$O(nk)$	$O(\mathrm{k}n^2l^2)$	$O(t^2l^2)$
VSS based on Number Theory[11]	$O(n)$	$O(n^2l^2)$	$O(t^2l^2)$
VSS based on GCRT[12]	$O(n)$	$O(n^2l^2)$	$O(t^2l^2)$
VSS based on Finite Automaton Public Key Cryptosystem (PKC)[13]	$O(n^3)$	$O(n^2)$	$O(n^2)$
VSS based on RSA (This scheme)	$O(n^2)$	$O(n^3)$	$O(\mathrm{t}n^3) \approx O(n^4)$

Performance analysis of the computation cost of some VSS schemes

Note: ***n** is the number of shares, **t** is the threshold, **l** is the size of the share in bits, **k** is the number of verification secrets, and **N** is the size of modulus used in RSA algorithm.*

In VSS based on RSA, standalone SSSS is used for both share generation as well as of secret reconstruction. All the complexities are calculated in the worst case (Big-O notation). The computation cost of share generation in VSS using RSA is almost equal to other VSS schemes $O(n^2)$.

$$O(n^2) = \max \{O(n^2), O(nt)\}$$

where $O(n^2)$ is the time complexity for hashing **n** shares and $O(nt)$ is the time complexity for generating the shares of the secret.

The cost involved in share generation in VSS using RSA is only marginally more than the VSS based on CRT[10] because the number of verification secrets is less than the number of secrets itself but less than the VSS based on Finite Automaton PKC[13]. The computation cost for secret reconstruction is approximately equal to the other VSS schemes. When the share verification phase of VSS using RSA is compared with other VSS schemes, only a significant difference is found as all VSS schemes practically use the same amount of computational cost with a little difference coming from l, which is a finite value.

All RSA operations are essentially modular exponentiation - whether it is encryption, decryption, signing or verifying and these computations are performed by a series of modular multiplications. The RSA has the total computation cost of $O(log_2(N)^3)$, using the standard multiplication and division algorithms, where N is the size of the modulus used in RSA algorithm. The total cost of share verification phase in VSS using RSA is $O(n^3)$.

$O(n^3)$ = max $\{O(nlog_2(N)^3)$, $O(n^3)\}$

where $O(nlog_2(N)^3)$ is the time complexity for inverse operation & decryption for ***n*** shares and $O(n^3)$ is the time complexity for hashing ***n*** shares & hash matching ***n*** with the dealer's published hash values.

Though, by using other exponential algorithms such as *Karatsuba algorithm* for multiplication and *Barrett reduction* for Euclidian division and inversion process, time complexity for encryption & decryption processes can be further reduced in RSA to $O((log_2N)^{1.585})$ and $O((log_2N)^{2.585})$ respectively but it may not benefit the verification phase too much in VSS using RSA in particular.

The total cost of secret reconstruction phase in VSS using RSA is $O(\mathrm{t}n^3) \approx O(n^4)$, if a large amount of **t** shares are called for secret reconstruction.

$O(\mathrm{t}n^3)$ = max $\{O(tlog_2(N)^3), O(tn^3)\}$

where $O(\mathrm{t}log_2(N)^3)$ is the time complexity for inverse operation & decryption for ***t*** threshold shares and $O(tn^3)$ is the time complexity for hashing ***t*** shares & hash matching with ***n*** dealer's and shareholder's published hash values. Whereas the complexity of $O(tn^3)$ in the secret reconstruction phase can be explained as $O(tn)$ for hashing of all individual ***t*** threshold shares and $O(n^2)$ for hash matching with the dealer’s and all of the shareholder’s published hash values for all individual ***t*** threshold shares.

From the above study, when the overall scheme of VSS using RSA is considered (share generation, share verification and secret reconstruction), then this scheme seems to be almost equal to the other studied VSS schemes, as the total computation cost of most VSS schemes studied here is found to be quadratic and the computation cost of VSS using RSA also turned out to be quadratic. However if any other cost effective method is incorporated for hashing or share validation, then this difference will further reduce considerably. As RSA is a very resource intensive algorithm and cannot be used directly in any resource limited environment so its use can be justified for only some critical applications like in the cases of server's disaster recovery management where there is the need for restructuring any application, database etc. on the basis of secure cryptographic keys. Thus this scheme provides optimum results at enterprise level – where the limitations on the computing power and the network infrastructure are less than the consumer world. Also, as the share verification and secret reconstruction are an on-demand activity (required only when there is a need for reconstructing the secret by any shareholder), there will not any consistent overhead due our scheme. As cryptography is an ever evolving field, any improvement in the computational cost is always a welcome step.

On the basis of the above performance analysis as well as implementation of VSSS using RSA, the various advantages and limitations are given below:

Advantages:

1. The VSS using RSA can be implemented in a proactive network.
2. VSS using RSA does not depend on the homomorphic commitments.
3. A cheater can be detected and thus its share can be discarded.
4. Shares can be distributed securely between the shareholders.
5. VSS using RSA leverages the advantages of public key cryptography to the secret sharing schemes.
6. The secrets are safe from the adversaries.

Limitations:

1. Dealer-leakage resilience but it is a relatively new concept and need more insight for its incorporation in PKI.

2. As secret sharing is a closed group activity, it is based on several assumptions.

Chapter 5
Conclusions and Future Scope

In this thesis, an RSA-based verifiable secret sharing scheme is discussed. However, it is found to be efficient only in the computational and network rich environments. It both defines as well limits its application. The VSSS using RSA extends the basic definition of a (t, n) SSS and also provides a formal definition of the (n, t, n) SSS. This (n, t, n) SSS is information-theoretically secure, similar to the Shamir's (t, n) SSS. As all generated shares are of the same size, the adversary can never get to know whether it is viewing the actual secret or some part of it. During the verification process, any share not satisfying the security requirements of a (t, n) secret sharing scheme were omitted. It also helped in both aspects of cheater detection as well identification. It can accurately detect as well identify the cheater in the SS group.

In future, one can further explore the verification processes in secret sharing schemes using public key cryptography. I would also like to explore about the much more secure and robust VSSS like multi-secret sharing scheme (MSSS) with the extended capabilities of share verification and cheater detection & identification[8] etc. Also, it would be interesting to explore its application in dealer-leakage resilience[17]. In this thesis, it is found that PKI can be used in threshold SS mechanism with cheater detection and identification. As PKI using RSA is very resource intensive, it may not be favorable in some applications like embedded systems using wireless networks. By using lightweight asymmetric cryptography, the functionality of this VSSS can also be extended to the embedded systems. This can also be justified because of the fact that hardware implementations of PKC work much faster their software counterpart. The use of elliptic curve and self-pairing can be further explored to develop secret sharing schemes with more generalized access structure.

References

[1] Shamir, A., "How to share a secret", Nov 1979, Communications of the ACM Vol 22 No 11, pp. 612–613.

[2] Blakley, G. R., "Safeguarding Cryptographic Keys", 1979, AFIPS Conference Proceedings, Vol 48, pp. 313–317.

[3] Islam, S., Mahmudul Hasan, A. S. M., "Implementation of Shamir's Secret Sharing on Proactive Network", Sep 2013, International Journal of Applied Information Systems, Vol 6 No 2, pp. 17-22.

[4] Chor, B., Goldwasser, S., Micali, S., Awerbuch, B., "Verifiable Secret Sharing and Achieving Simultaneity In The Presence Of Faults", 1985, Massachusetts Institute of Technology, IEEE, pp.383-395.

[5] Fitzi, M., Garay, J., Gollakota, S., Rangan, C.P., Srinathan, K., "Round-Optimal and Efficient Verifiable Secret Sharing".

[6] Benaloh, J.C., "Secret sharing homomorphisms: keeping shares of a secret", Lecture Notes in Computer Science, vol. 263, Springer-Verlag, pp. 251–260.

[7] Pedersen, T. P., "Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing", 1998, Springer-Verlag, pp 129-140.

[8] Harn, L., "Generalized cheater detection and identification", 2014, Institution of Engineering and Technology Information Security, Vol. 8 Issue 3, pp. 171–178.

[9] Lin, C., Harn, L., "Unconditionally Secure Verifiable Secret Sharing Scheme", Sep 2012, Advances in Information Sciences and Service Sciences, Vol 4 No 17, pp. 514-518.

[10] Harn, L., Miao, F., Chang, C. C., "Verifiable secret sharing based on the Chinese remainder theorem", June 2014, Security and Communication Networks, Vol 7, No 6, pp. 950-957.

[11] Liu, Y., Harn, L., Chang, C. C., "A novel verifiable secret sharing mechanism using theory of numbers and a method for sharing secrets", May 2015, International Journal of Communication Systems, Vol 28, No 7, pp. 1282-1292.

[12] Yanjun, L., Chang C. C., "An Integratable Verifiable Secret Sharing Mechanism", July 2016, International Journal of Network Security, Vol 18, No 4, pp. 617-624.

[13] Saeidia, A., Zahedib, M. M., Amroodi, A. N., "A New Secret Sharing Based on finite automaton public key cryptosystem", August 2014, Journal of Theoretical Physics and Cryptography (JTPC), Vol 6, pp. 38-41.

[14] Feldman, P., "A Practical Scheme for Non-Interactive Verifiable Secret Sharing", 1987, Proceedings of the 28th Annual Symposium on Foundations of Computer Science (SFCS), pp. 427-437.

[15] https://github.com/moserware/SecretSplitter

[16] http://www.moserware.com/2011/11/life-death-and-splitting-secrets.html

[17] R. F. Olimid, "Dealer-Leakage Resilient Verifiable Secret Sharing", Sep 2014, Department of Computer Science, University of Bucharest, Romania, pp 1-10.

[18] Peng, K., Bao, F. "Efficient Publicly Verifiable Secret Sharing with Correctness, Soundness and ZK Privacy", 2009, LNCS 5932, Springer-Verlag pp. 118–132.

[19] Jhanwar, M. P., Venkateswarlu, A., Safavi-Naini, R., "Paillier-Based Publicly Verifiable (Non-interactive) Secret Sharing".

[20] Shao, J., "Efficient verifiable multi-secret sharing scheme based on hash function", Mar 2014, Elsevier Information Sciences, Vol 278, pp 104–109.

[21] Jackson, W., Martin, K., O’Keefe, C., "Multisecret Threshold Schemes", CRYPTO 1993, LNCS, Vol 773, pp 126–135.

[22] Steinfeld, R., Wang, H., Pieprzyk, J., "Lattice-Based Threshold-Changeability for Standard Shamir Secret-Sharing Schemes", Dept. of Computing, Macquarie University, North Ryde, Australia, pp 168-183.