K. L. E. SOCIETY'S

B. V. Bhoomaraddi College of Engineering & Technology,

Vidyanagar, Hubli – 580031

(An Autonomous Institution)

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

Software Requirements Specification

for

Threshold Multi-Signature

Submitted by

Mr. XYZ1	**2BV**IS0****
Mr. XYZ2	**2BV**IS0****
Ms. XYZ3	**2BV**IS0****
Ms. XYZ4	**2BV**IS0****

2010 – 2011

CONTENTS

1. INTRODUCTION

1.1 Purpose

The purpose of this document is to present a detailed description of the Threshold Multi-Signature Scheme. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the Batch evaluator and the developers of the system and will be proposed to the Project Review Committee for its approval.

1.2 Product Scope

This software system will be a Secure Digital Signature Scheme, which can trace back the individual generator of the signature. This system will be designed to allow the verifier to check whether each signer of the signature is genuine.

More specifically, this system is designed to generate a Digital Signature which contains the identities of all the individual signers. In case of forgery, all the details of the signers are revealed. The signers of Signature do not remain anonymous as in the existing system.

This software can be further enhanced by using intranet for key and data exchange between two or more node(s).

1.3 Definitions, Acronyms and Abbreviations

TTP : Trusted Third Party

Digital Signature: A set of alphabetic or numeric characters used to authenticate a cryptographic message by ensuring that the sender cannot later disavow the

message, the receiver cannot forge the message or signature, and the receiver can prove to others that the contents of the message are genuine and originated with the sender.

1.4 References

[1]. IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.

[2]. Johann van der Merwe,Dawoud S. Dawoud, Member, IEEE, and Stephen McDonald, Member, IEEE**, "A Fully Distributed Proactively Secure Threshold- Multisignature Scheme"** IEEE transactions on parallel and distributed systems, VOL. 18, NO. 4, April 2006.

[3]. Qi Xie, Zhonghua Shen, Xiuyuan Yu. "**Threshold Signature Scheme Based on Modular Secret Sharing**". International Conference on computational Intelligence and security,2008.

2. OVERALL DESCRIPTION

2.1. Product Perspective

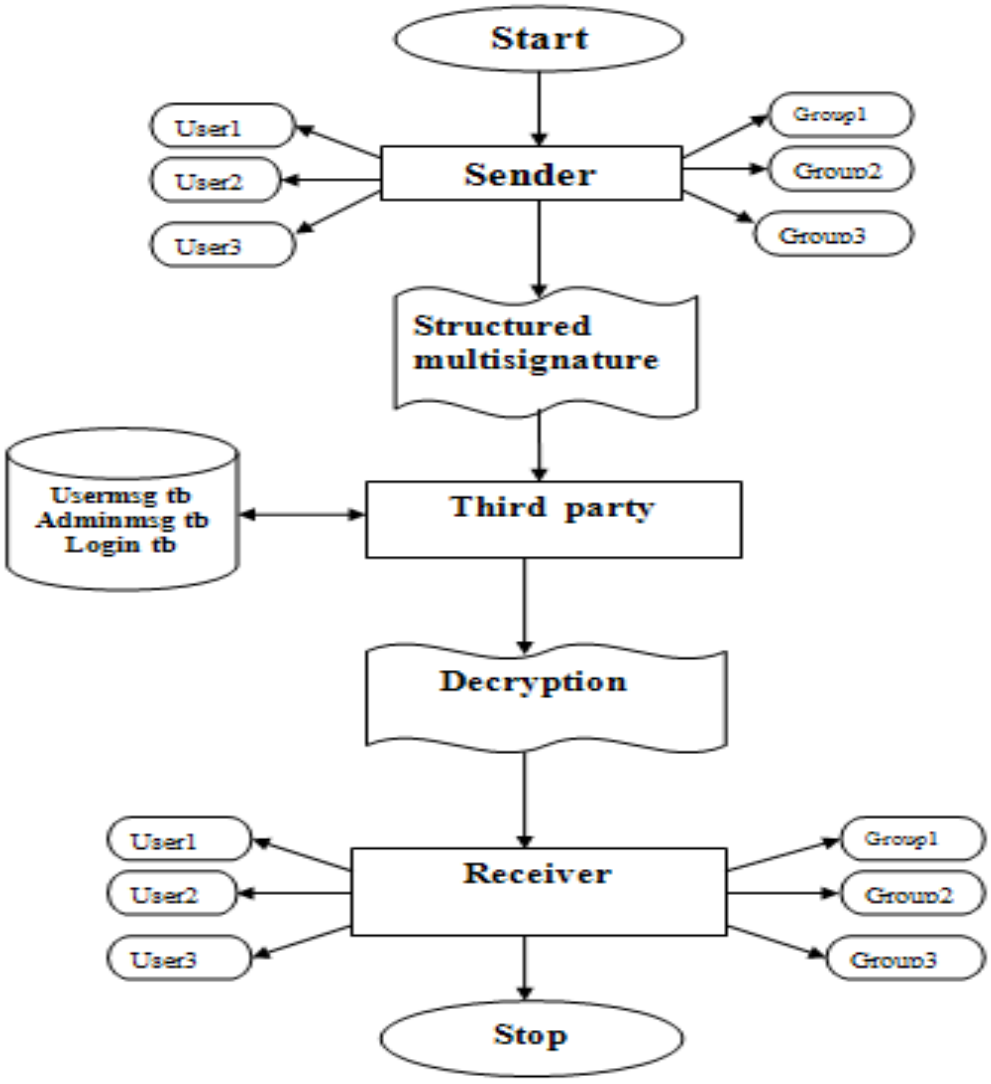

Figure 2.1: Model of the System

2.2 Product Functions

The main functions of product are as follows:

- Generation of Threshold Multi-Signature, which is a digital signature scheme.
- Verification of the Threshold Multi-Signature, whether all the people who have signed are valid.
- Management of the Keys with a TTP, i.e. sending the keys and generating them.

Encryption and Decryption of the signature to be sent and received respectively.

2.3 General Constraints

The application requires complex mathematical calculations such as exponential operations and Mod operations which becomes harder to implement as the value of the operands gets higher.

2.4 Assumptions and Dependencies

In this application, it has been assumed that the system does key exchange in a secure manner i.e., it does not suffer from the “Replay Attack”.

The System supports 32 bit arithmetic operations.

3. SPECIFIC REQUIREMENTS

< This section includes all the technical information needed to design the software. It is more specific than the previous section. It contains all the software requirements at a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements. Remember, the requirements should exhibit the following properties: correct, unambiguous, complete and consistent. >

3.1 Functional Requirements

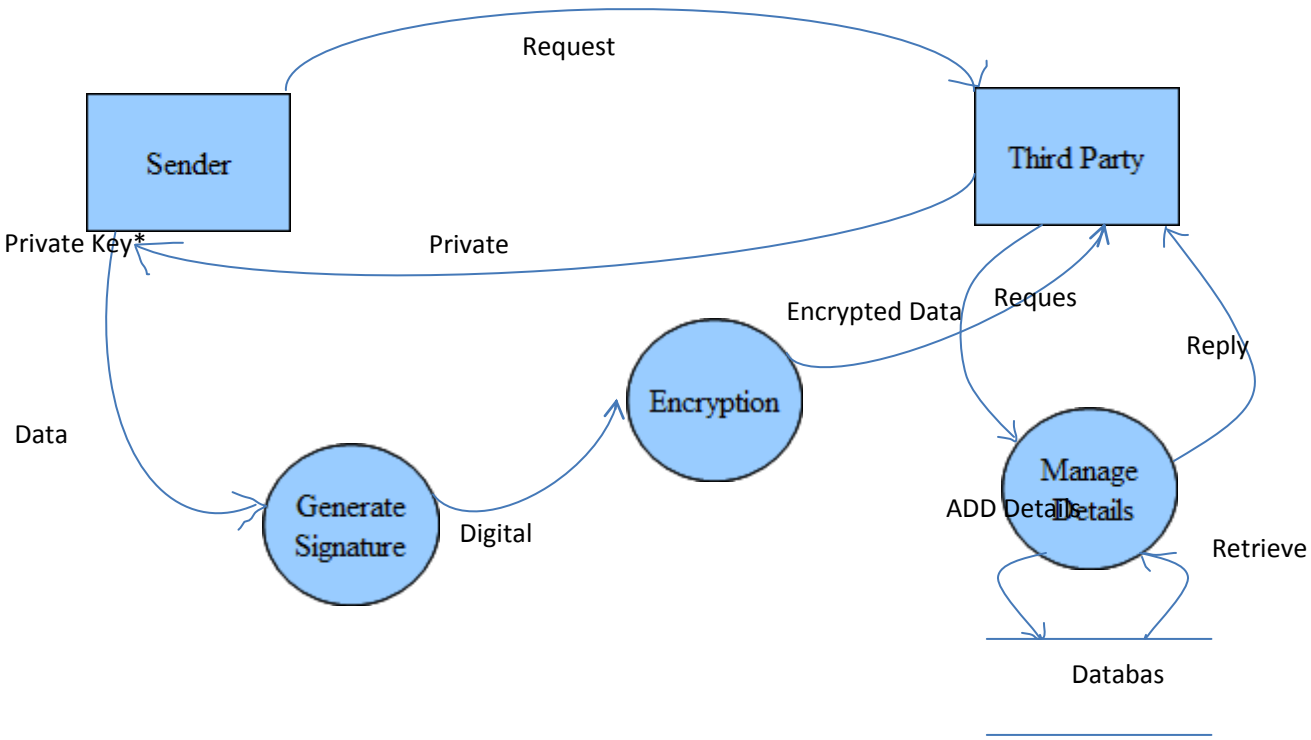

Figure 3.1: DFD of Sender Module

3.1.1 Process specification:

3.1.1.1. Signature generation

Valid input: It takes Data and the Private key.

Output: Encrypted data using the signature generation algorithm.

- It takes Data and the private key from the user and generates the signature.

3.1.1.2. Encryption

Valid Input: Digital Signature.

Output: Encrypted Data.

- It takes the digital signature as input and then encrypt it using the encryption algorithm
- Sends the encrypted data to the Trusted Third Party.

3.1.1.3. Manage Details

Valid Input: The Database

Output: The query given by the Third Party

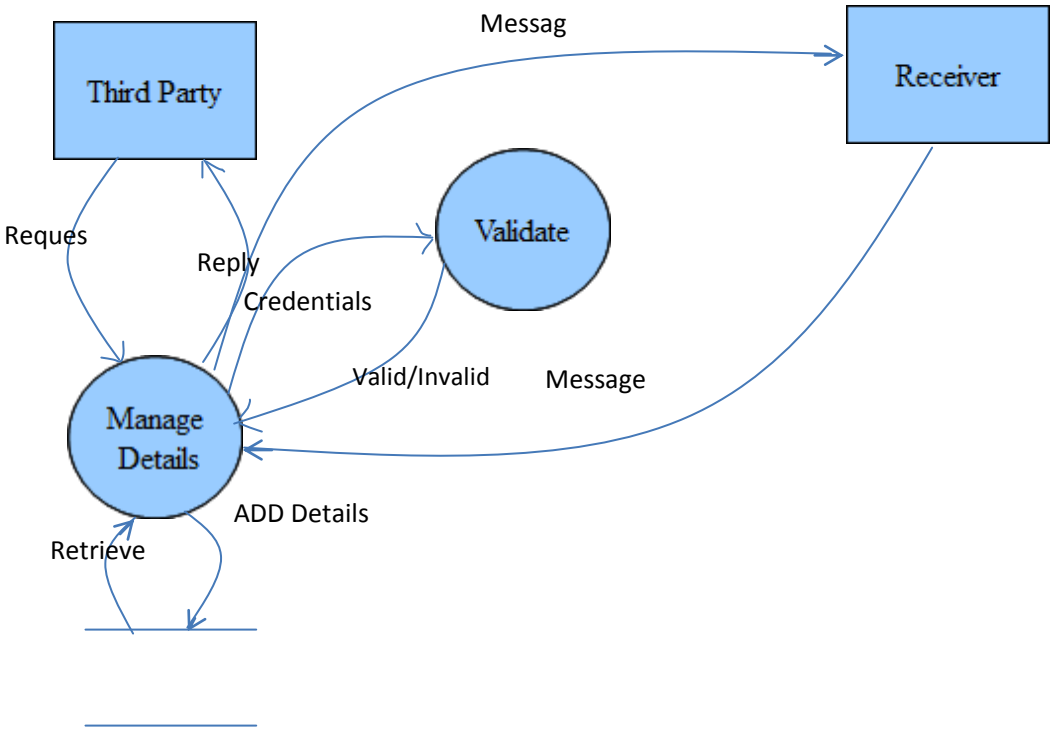

Figure: 3.2 DFD of Receiver Module

3.1.2.1. Validate

Valid input: It takes Data and the Public key.

Output: Decrypt data using the Decryption algorithm.

- It takes Data and the public key from the TTP and validates the signature.

3.1.2.2. Manage Details

Valid Input: The Database

Output: The query given by the Third Party

- It takes the query given by the third party processes it and returns back result. The Modules Involved in carrying out these functions.

3.1.3. Sender Module

The sender Encrypts and sends message to the third party. The procedures are

3.1.3.1. Login Procedure

This Procedure is used to authorize the user. The user logs in using his id and password. Only the authorized users can send the message.

3.1.3.2. Encryption Procedure

This Procedure involves the encryption of the data to be sent. The sender requests for the key from the TTP and encrypts(signs) the data and sends it to third party.

3.1.3.3. Message Send Procedure

This procedure is used to send the message to the TTP.

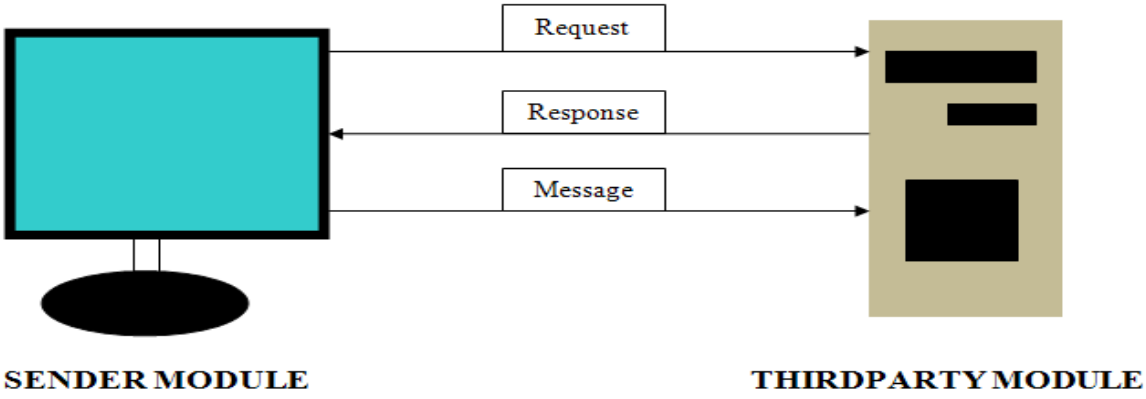

Figure 3.3: Pictorial Representation of Sender

3.1.4 Third Party Module

- Receive the request from both the sender as well as the receiver.
- Request from sender, the third party will authenticate the sender.
- Accepts the message from the sender and stores it in its database.
- Request from the receiver, the third party will authenticate the receiver and delivers the decrypted message.

The Procedure implemented are

3.1.4.1. Database Connect Procedure

Using this module the user data is stored in database. The ID of the user and the Keys to be distributed are stored in the database using this procedure.

3.1.4.2. Sender Procedure

This module manages the sender related queries and responses. The distribution of key for the sender is handled by this module

3.1.4.3. Receiver Procedure

This module manages the receiver related queries and responses. The TTP accepts Authorized receiver requests and delivers them the message.

3.1.4.4. Decryption Procedure

This module does the decryption of the data sent by the sender

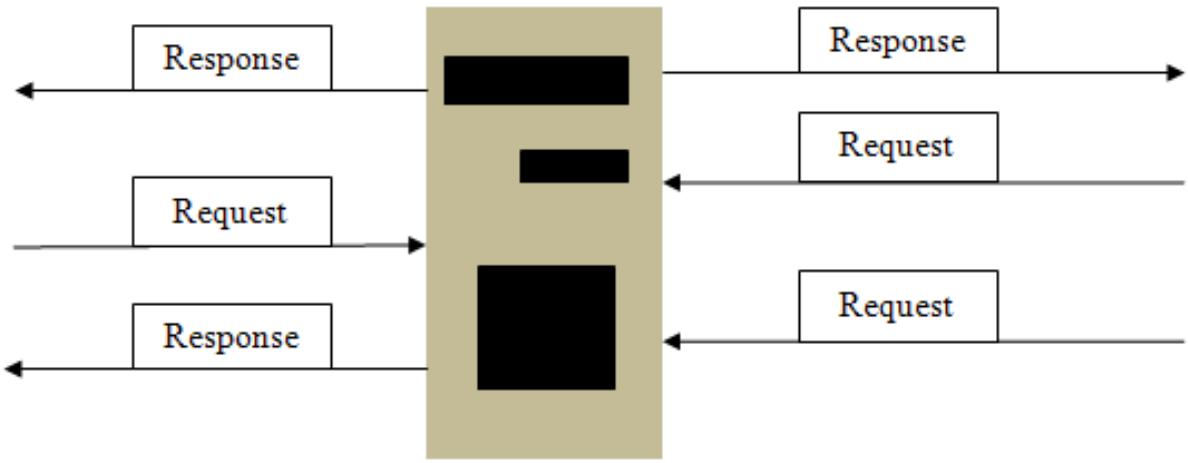

Figure 3.4: Pictorial Representation of TTP

3.1.5 Receiver Module

- Receiver/receiver group is authenticated.
- Check whether this particular receiver/receiver group has any message in the database.
- If any message is present, then third party will decrypt, deliver it to the receiver.

The Procedures for this module are

3.1.5.1. Log in Procedure

This procedure is to authorize the user. It checks the ID and Password of the receiver

3.2 External Interface Requirements

3.2.1 System Interfaces

- The User requests the private key of 16 bits from TTP
- On the request, TTP shall send the private key to the user
- The User on getting the private key encrypts the data/message using El-Gamal Algorithm
- The user then encrypts the signature using the DES algorithm
- and sends the data to the TTP
- The TTP stores the message along with the ID of the sender and the receiver

- The receiver has to authenticate himself, on authenticating, the receiver requests the message from the TTP
- The TTP checks whether it has any message meant to the requested receiver.
- The receiver on receiving the message decrypts and verifies the signature
- The receiver can find id of each Signer and the Identity can be revealed.

3.2.2 Hardware Interfaces

Optional
< Characteristics of the interface between the SW product and HW components of the system. Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used >

3.2.3 Software Interfaces

- The application is built using Netbeans IDE 8.0.
- In order to store the data, the application uses Oracle as well which gives us more security in the network.

3.2.4 Communication Interfaces

- Sockets for creating client and server.
- TCP Protocol for transfer of packets.

3.3 User Interface Requirements

The interface must be easy to understand. The user interface includes

- SCREEN FORMATS/ORGANIZATION: The introductory screen will be the first to be displayed which will allow the users to enter the user name and password. He will be logged in once he enters valid details.
- WINDOW FORMAT/ORGANIZATION: When the user chooses some other option, then the information pertaining to that choice will be displayed in a new window which ensures multiple windows to be visible on the screen and the users can switch between them.
- DATA FORMAT: The data entered by the users will be alpha numeric
- END MESSAGES: When there are some exceptions raising error like entering invalid details, then error messages will be displayed prompting the users to re-enter the details.

3.4 Database Requirements

- Every field information will be of type string
- The database will be accessed whenever the sender or the receiver makes a request for data
- The database will be accessed only by the TTP
- Data entities and relationships defined by the ER diagram below

3.5 Performance Requirements

- The System must trace the exact individuals 95% of the times.
- Any outsider must be able to verify the signature 98% of the times.
- The signature must be Unforgeable 100% of the times.

3.6 Software Quality Attributes

All Multi-signatures on an arbitrary message, generated by an honest authorized subset of group members, forming subgroup P, can be verified by any outsider V (with respect to the group). This implies that the group-oriented signature is publicly verifiable.

Only a threshold of‘t’ or more authorized group members are able to collaboratively generate a valid Multisignature. This property thus incorporates unforgeability.

Any outsider V can learn the identities of the individual signers belonging to P from the Multisignature on ‘m’ without interaction with any of the group members and/or a group manager. This implies that the signers are publicly traceable with public information. Traceability implies accountability, the individual signers participating in the Multisignature scheme can be held accountable for their contribution to the group oriented signature.

No colluding subset of group members can generate a valid Multisignature not satisfying the traceability property. Coalition-resistance subsumes framing-resistance, i.e., no subset of group members can sign on behalf of any other subset of group members.

An adversary in possession or control of the group secret key and/or the individual secret shares of any number of group members cannot generate a valid Multisignature and/or partial/ individual signatures. Thus, although the underlying threshold cryptosystem has been broken, the Multisignature signature scheme should not be breakable.

APPENDICES

This section is optional.
< Appendices may be included if any, either directly or by reference, to provide supporting details that could aid in the understanding of the Software Requirements Specifications.>