Construction of a Secret Sharing Scheme with Multiple Extra Functionalities

CPSC 449 Honours Thesis Proposal

John Johnson
Honours Student
Department of Computer Science
University of British Columbia

Supervisor : Dr. Clara Wood

October 2003

1. Introduction

There are circumstances where an action is required to be executed by a group of people. For example, to transfer money from a bank a manager and a clerk need to cooperate. A ballistic missile should only be launched if three officers authorize the action.

In communications networks that require security, it is important that secrets be protected by more than one key. Furthermore, a system of several keys that can be combined in multiple ways may allow for the recovery of a unique secret regardless of how they are combined. Schemes that have a group of participants that can recover a secret are known as Secret Sharing Schemes.

2. Overview of Area

The idea of secret sharing is to start with a secret, divide it into pieces called shares, which are then distributed amongst users by the dealer. Only certain groups (authorized subsets of participants) can reconstruct the original secret. More formally a Secret Sharing Scheme (SSS) is a method whereby n pieces of information called shares or shadows are assigned to a secret key K in such a way that
1. the secret key can be reconstructed from certain authorized groups of shares and
2. the secret key cannot be reconstructed from unauthorized groups of shares.

2.1 Threshold Schemes

First threshold schemes were independently invented by both Adi Shamir[3] and George Blackley[1] in 1979. Shamir [3], states that threshold schemes can be very helpful in the management of cryptographic keys. To protect data, we would encrypt it. However to protect the encryption key we need a different method. The most secure key management scheme keeps the key in a single place. This sort of scheme may not always be appropriate, for instance in a single case of misfortune the key may be rendered inaccessible. An obvious solution to this may be to make multiple copies of the key. This however also increases the risk associated in keeping multiple keys secret. By using Shamir's [3] threshold scheme concept we can get a very robust key management scheme.
Threshold schemes are ideally suited to situations where a group of mutually suspicious individuals with conflicting interests must cooperate [3].

We will now use the definition outlined in [5] to describe what a threshold secret sharing scheme is.
Definition 2.1. Let k and n be positive integers, $k \leq n$. A $(k,n)-$threshold scheme is a method of sharing a key K among a set of n players (denoted by P), in such a way that any k participants can compute the value of K, but no group of $k-1$ participants can do so.
The value of K is chosen by a special participant which is referred to by [5] as the dealer. When D wants to share the key K among the participants in P, gives each participant some partial information referred to earlier as a share. The shares should be distributed

secretly, so no participant knows the share given to any other participant. At some later time, a subset of participants $B \subseteq P$ will pool their shares in an attempt to compute the key K. Alternatively they could give their shares to a trusted authority which will perform the computation on their behalf. If $|B| \geq k$, then they should be able to compute the value of K as a function of the shares they collectively hold. Furthermore if $|B| < k$, then they should determine nothing about the value of K.

2.2 General SSS and Access Structures

In the outline of threshold schemes, we wanted k out of n participants to be able to determine the key. A more general situation is to specify exactly which subsets of participants should be able to determine the key and those that should not [5].
Let's denote Γ as being a set of subsets of P, and the subsets in Γ as being the subset of participants that should be able to compute the key. Then Γ is denoted as being the access structure and the subsets in Γ are called authorized subsets. Furthermore if we let K be the set of keys and S be the share set, we use the dealer D to share a key $k \in K$ by giving each player a share $s_i \in S$. Some time later a subset of players might attempt to determine K from the shares they collectively hold.

Definition 2.2 (Stinson, [5]). A perfect secret sharing scheme using the general access structure Γ, is a method of sharing a key K among a set of n participants such that P is the set of all participants, in such a way that the following two properties are fulfilled:
• If an authorized subset of participants $B \subseteq P$ pool their shares, so that they can determine the value of K.
• If an unauthorized subset of participants $C \subseteq P$ pool their shares, then they can determine nothing about the value of K

We notice that a (k,n)-threshold scheme creates the access structure $\{B \subseteq P \mid |B| \geq t\}$
This structure is referred to by Stinson [5] as the threshold access structure.

It possible to create a SSS for any access structure as long as this access structure satisfies monotone property:

- If B is an authorized subset $B \in \Gamma$ and $B \subseteq C \subseteq P$ then $C \in \Gamma$

In other words a superset of an authorized set is again an authorized set [2].

3. Outline of Proposed Research

So far I outlined fundamental properties and constructions of secret sharing. In many circumstances, secret sharing has to provide more flexibility and functionality. The requirements for different extra functionalities of SSS are often contradictory to each other which makes construction of a SSS with several additional features a challenge.

In my proposed research I will introduce a basic SSS and will incrementally add several extra functionalities to it. Then I will critically analyze the strengths and weaknesses introduced into the scheme by each of these features.

The basic model that I will use is a secret sharing scheme based upon (n,n) threshold sharing. Later it will be extended to general access structures. The implementation of (n,n) scheme will be done as follows:

> Let the secret be a vector of η numbers $S_\eta = \{s_1, s_2, \ldots, s_\eta\}$. The dealer chooses a modulus p, such that $p > \max(s_1, s_2, \ldots, s_\eta)$. Next the dealer generates randomly, uniformly and independently n-1 vectors S_η^i with elements in Z_p. The last share S_η^n is obtained by summing secret S_η with n-1 shares S_η^i in Z_p, $1 \le i \le n-1$.
> All n participants are given shares that are η-dimensional vectors S_η^j with elements in Z_p.
> To retrieve the secret, participants have to add their vectors component-wise in Z_p.

<u>The following list contains extra functionalities that will be added:</u>

1. Proactive functionalities

> It will be possible to perform the following actions without changing the secret:
> - periodically renew shares
> - enroll and disenroll shareholders
> - recover lost or corrupted shares
> - change of the access structure (say increment or decrement of the threshold parameter)

2. Publicly verifiable

> Verifiable secret sharing will allow participants to check whether shares they are given by the dealer are consistent with other shares and the secret [4].

3. General access structure

> Allowed access structures will be extended from (n,n)-threshold to any monotone structure [2].

4. *Limit dealer knowledge about the secret*

> Usually, the dealer has to know the secret in order to share it. This gives dealer advantage over ordinary shareholders. There are situations where such advantage can lead to abuse. Thus, it is preferable to either limit dealer's knowledge or completely eliminate dealer and share the secret automatically.

5. *Cheating prevention*

> Cheating is when a dishonest participant intentionally modifies shares in such a way that after the combiner announces the secret, the cheating participant is able to compute it, while other participants cannot. Cheating prevention is a technique which addresses this problem [6].

4. Proposed Timeline, Deadlines and Deliverables

Timeline

Phase	*Start Date*	*End Date*
Implement basic SSS	Oct 20	Nov 5
Shares renewal feature	Nov 6	Nov 16
Enroll/disenroll, change access structure features	Nov 17	Nov 27
Recover feature	Nov 28	Dec 15
Publicly verifiable	Dec 16	Jan 5
General access structure	Jan 6	Jan 20
Limit dealer knowledge about the secret	Jan 21	Feb 20
Cheater prevention	Feb 21	Mar 19
Compile Data, Finalize Project Report	Mar 20	Mar 27

Deliverables

Final Report - 25 to 35 pages

This paper is targeted toward people with minimal Secret Sharing cryptography knowledge. Detailed proofs of most results will be left out. However, important results will be noted and examples shown to illuminate their correctness.

Code - 1500 to 3000 lines

The code will be written in Java and executable on CS machines.

Maintenance Manual - 5 pages

This manual will be given as an appendix to the final report.

5. References

[1] Blakley, G. R., Safeguarding Cryptographic Keys, Proceedings AFIPS June 1979 National Computer Conference

[2] Itoh M., A. Saito and T. Nishizeki, "Multiple assignment scheme for sharing secret", Journal of Cryptology, vol.6, no.1, 1993, pp.15-20

[3] Shamir, A., How to Share a Secret, Communications of the ACM, vol.22, no.11, 1979

[4] Stadler, M., “Publicly verifiable secret sharing”, Lecture notes in Computer Science, 1997, 190-199

[5] Stinson, D.R., Cryptography: Theory and Practice, CRC Press, 1995.

[6] Tompa, M., and Woll, H. "How to share a secret with cheaters", Journal of Cryptology, Vol.1, No.2, 1988, pp.133-138.