How to identify all cheaters in Pinch's scheme

Chan Yeob Yeun*

Information Security Group,
Royal Holloway, University of London,
Egham, Surrey TW20 0EX, UK
Email : c.yeun@rhbnc.ac.uk
Tel: +44 1784 443113
Fax: +44 1784 430766

Chris J. Mitchell

Information Security Group,
Royal Holloway, University of London,
Egham, Surrey TW20 0EX, UK
Email : c.mitchell@rhbnc.ac.uk

Summary: A modified version of the Pinch multiple secret sharing protocol is proposed, which identifies *all* cheaters, regardless of their number, improving on previous results by Pinch and Ghodosi *et al.*

Keywords: Online secret sharing; Identify all cheaters; Discrete logarithm problem; Digital signature

*The author is supported by a Research Studentship and Maintenance Award from RHUL.

1 Introduction

A secret sharing scheme is a protocol in which a dealer distributes a secret among a set of participants such that only specified subsets of them, defined by the access structure, can recover the secret at a later time.

Cachin [1] proposed a computationally secure scheme for online secret sharing with general access structures, where all the shares are as short as the secret. The scheme provides the ability to share multiple secrets and allows participants to be added dynamically, without having to redistribute new shares. These abilities are realised by storing additional authentic information at a publicly accessible location.

Pinch [2] pointed out that Cachin's scheme does not allow shares to be reused after the secret has been reconstructed without a further distributed evaluation protocol such as Goldreich *et al.* [3]. Pinch presented a modified protocol for computationally secure online seceret sharing, based on the intractability of the Diffie Hellman problem, where shares can be re-used.

Ghodosi *et al.* [4] pointed out that Pinch's scheme is vulnerable to cheating. They present a modified version of Pinch's protocol which detects cheating and prevents cheating assuming a majority of participants are honest, but does not protect a minority of participants of an authorised set against a majority colluding to falsely accuse the minority of cheating.

We propose an enhanced modified version of Pinch's secret sharing protocol which has the advantages over the original scheme, and its modification by Ghodosi *et al.*, that it detects cheating and enables the identification of *all* cheaters by an arbitrator, regardless of their number.

2 Pinch's Scheme

A secret sharing scheme is a protocol between a set of participants $\mathcal{P} = \{P_1, \ldots, P_n\}$ and a dealer D, where $D \notin \mathcal{P}$ is assumed. Certain subsets $X \in 2^{\mathcal{P}}$ are authorised to recover the secret K, initially known only to D, by combining shares S_i, each S_i being known only to P_i and D. The access structure Γ is then the set of minimal authorised sets, where an authorised set X is minimal if and only if $Y \subseteq X$ and Y authorised implies that $X = Y$.

In the remainder of this paper we work within the ring of integers modulo p, for some prime p. We suppose $p - 1$ has a large prime factor q, and we choose an element $g \in \mathbb{Z}_p$ of order q. The primes p and q must be chosen so that determining discrete logarithms to the base g modulo p is computationally infeasible. Most of our calculations involve working within the multiplicative cyclic group of order q generated by g. It is possible to describe the schemes in a more general group-theoretic framework, although we do not consider this here. We also use a one-way function f.

The basic protocol to share $K \in \mathbb{Z}_p$ works as follows:

The dealer D, who knows the secret K, randomly chooses secret shares $S_i < q$ for each participant P_i and transmits S_i over a secure channel to P_i. For each minimal trusted set $X \in \Gamma$ the dealer D randomly chooses g_X to be an element of multiplicative order q

(mod p), and computes

$$T_X = K - f(g_X^{\prod_{x \in X} S_x}) \bmod p$$

and posts the pair (g_X, T_X) on the notice board.

To recover the secret K, a minimal trusted set $X = \{P_1, \ldots, P_t\}$ of participants comes together and performs the following steps:

1. Participant P_1 reads g_X from the notice board and sends ${g_X}^{S_1} \bmod p$ to P_2.
2. Each subsequent participant P_i, for $1 < i < t$, receives ${g_X}^{S_1 S_2 \ldots S_{i-1}} \bmod p$, raises it to the power S_i and sends the result, which equals ${g_X}^{S_1 S_2 \ldots S_i} \bmod p$, to P_{i+1}.
3. The final participant P_t receives ${g_X}^{S_1 S_2 \cdots S_{t-1}} \bmod p$ and raises this value to the power S_t to form

$$V_X = {g_X}^{S_1 S_2 \cdots S_t} \bmod p = g_X^{\prod_{x \in X} S_x} \bmod p.$$

4. On behalf of group X, member P_t reads T_X from the notice board and can now reconstruct K as $K = T_X + f(V_X) \bmod p$.

If there are multiple secrets K_i to share, then it is possible to use the same shares S_i and one way function f, provided that each entry on the notice board has a fresh value g_X attached.

3 Analysis of the Protocol

3.1 How to detect cheating

Ghodosi *et al.* [4] describe a method for detecting cheating in the above protocols. Suppose in the initialisation phase of the scheme, the dealer D sends ${g_X}^{V_X} \bmod p$ to every authorised set X. Let the reconstruction protocol be the same as in the above scheme and let V'_X be the computed result. Every participant $x \in X$ can verify that ${g_X}^{V'_X} \equiv {g_X}^{V_X}$ (mod p). If the verification fails, then cheating has occurred in the protocol and thus the computed secret is not correct.

However, this method should be carefully implemented to prevent attacks which exploit the arithmetic of exponents. Since we choose our generator g_X to have order q, we know that ${g_X}^{V'_X} \equiv {g_X}^{V_X} \pmod p$ if and only if $V'_X \equiv V_X \pmod q$. Hence, if a malicious participant could arrange for everyone to accept $V'_X = V_X + rq$ for some non-zero integer r, then cheating will not be detected.

For this reason, we propose an alternative way of detecting cheating. Suppose in the initialisation phase of the scheme, the dealer D publishes $h(K_i)$ on the notice board for every secret K_i that is being shared (where h is a one-way collision-resistance hash-function). Every participant, having reconstructed the secret (K'_i, say), can verify its validity by hashing it and comparing the resulting hash-code $h(K'_i)$ with the value on the notice board. If the verification fails, then cheating has occurred in the protocol and thus the computed secret is not correct.

Note that the second method requires less storage space on the notice board than the first method. In the first method D stores ${g_X}^{V_X} \bmod p$ on the notice board, and hence needs to store $|\Gamma|$ values for every secret. In the second method, D stores $h(K_i)$ on the notice board, and hence D only needs to store one hash-code for every secret. Thus, the second method is a better way of detecting cheating.

3.2 An Enhanced Protocol which identifies all cheaters

We now describe an enhanced version of the protocol, which will enable the identification (by the dealer) of all cheaters. As a pre-requisite to using the scheme, every participant must have an implementation of an agreed digital signature scheme, and must have selected a key pair for this signature scheme.

In addition, every participant must have a means of obtaining a verified copy of every other participant's public signature verification key. This could, for example, be provided by having a Trusted Third Party (e.g. the dealer, D) certify every participant's public key, and having every participant distribute their certificate with every signed message they send.

The modified protocol will operate exactly as described in section 2, with the exception of the following modifications. In Steps 1 and 2 of the protocol, participant P_i, as well as forwarding ${g_X}^{S_1 S_2 \cdots S_i} \bmod p$, also forwards a signature on a data string, signed using his or her private signature key. More specifically, if $s_{P_i}(Y)$ denotes the digital signature on data Y computed using the private signature key of P_i, then P_i computes and forwards the signature

$$s_{P_i}({g_X}^{S_1 S_2 \cdots S_i} \bmod p || X || g_X)$$

to the next participant P_{i+1} (where $||$ denotes concatenation of data items). Also, when participant P_i receives ${g_X}^{S_1 S_2 \ldots S_{i-1}} \bmod p$ and the signed string containing ${g_X}^{S_1 S_2 \ldots S_{i-1}} \bmod p$, P_i checks the signature before proceeding with the protocol.

If cheating is detected by the method described in the second scheme in section 3.1, then every participant sends to the dealer the signed data strings they received during execution of the protocol. The dealer D calculates $g_X^{S_1}, g_X^{S_1 S_2}, \ldots, g_X^{S_1 \cdots S_t}$ in sequence, checking that what D gets is what was submitted by $P_1, P_2, \ldots, P_t$. As soon as a calculated value $g_X^{S_1 \cdots S_i}$ does not equal the submitted value, D knows that P_i cheated. P_i cannot claim to have been framed, since D has P_i's signature on $s_{P_i}({g_X}^{S_1 S_2 \cdots S_i} \bmod p || X || g_X)$. Then D uses the cheater's submission to check P_{i+1}'s submission and so on (i.e. for every i, D verifies that the value signed by P_i raised to the power $S_{i+1} \bmod p$ is equal to the value signed by P_{i+1}). Thus, D will then be able to identify *all* the parties who sent incorrect values during the protocol.

This use of signatures will also protect a minority of the members of an authorised set against a majority colluding to falsely accuse the minority of cheating.

4 Conclusion

The enhanced protocol can be used in such a way that cheating by participants can be detected, in which case the participants in an authorised set X can request help from the dealer D, who can always uniquely identify the cheaters.

5 Acknowledgements

The authors are grateful to Fred Piper for his support, and to Mike Burmester and Karl Brincat for comments on an early draft of the paper.

References

[1] C. Cachin. "On-line secret sharing". In C. Boyd, editor, *Proceedings of the 5th IMA Conference on Cryptography and Coding*, pages 190–198. Springer-Verlag, 1995.

[2] R.G.E. Pinch. "Online multiple secret sharing". *Electronics Letters*, 32(12):1087–1088, 1996.

[3] O. Goldreich, S. Micali, and A. Widgerson. "How to play any mental game or a completeness theorem for protocols with honest majority". In *Proceedings of 19th ACM Symposium on the Theory of Computing*, pages 218–229, 1987.

[4] H. Ghodosi, J. Pieprzyk, G.R. Chaudhry, and J. Seberry. "How to prevent cheating in Pinch's scheme". *Electronics Letters*, 33(17):1453–1454, 1997.