Short Paper: Mobile Proactive Secret Sharing*

David Schultz
Massachusetts Institute of Technology
das@csail.mit.edu

Barbara Liskov
Massachusetts Institute of Technology
liskov@csail.mit.edu

Moses Liskov
The College of William and Mary
mliskov@cs.wm.edu

ABSTRACT

MPSS is a new way to do proactive secret sharing in asynchronous networks. MPSS provides *mobility*: The group of nodes holding the shares of the secret can change at each resharing, which is essential in a long-lived system. MPSS additionally allows the number of tolerated faulty shareholders to change when the secret is moved so that the system can tolerate more (or fewer) corruptions; this allows reconfiguration on the fly to accommodate changes in the environment.

Categories and Subject Descriptors

C.2.4 [**Computer Communication Networks**]: Distributed Systems—*distributed applications*

General Terms

Security

1. INTRODUCTION

Malicious attacks are an increasing problem in distributed systems. If a node holds an important secret, that secret could be exposed by an attack in which an intruder gains control of that machine. An example of such a secret is the private key used by a certificate authority (such as Verisign) to sign its certificates.

Secret sharing allows a group of servers to possess *shares* of a secret, such that any $t + 1$ servers can collaborate to compute with the secret, but any t or fewer servers can learn nothing about the secret. Proactive secret sharing extends secret sharing to work in a long-lived system, in which nodes can become compromised over time, allowing the adversary to collect more than t shares and recover the secret. These schemes provide a share regeneration protocol, in which a new set of shares of the same secret is generated and the old shares discarded, rendering useless any collection of t or fewer old shares the adversary may have learned.

This paper describes a new resharing protocol called MPSS (for *Mobile Proactive Secret Sharing*). MPSS supports mobility: the secret shares can be moved to a new set of servers, which may overlap with the old set or be completely disjoint. Mobility is important because resharing requires the use of secure channels and when a node is corrupted its secret keys needed to run the resharing protocol may be compromised. MPSS allows such a server to be replaced with a new one. It protects the secret even when up to t servers in the old group and t servers in the new group are faulty. MPSS requires $n = 3t + 1$ shareholders, which is optimal for a protocol that works in an asynchronous network.

Additionally, MPSS allows the threshold t to change. This is desirable because the threshold has a meaning: it represents an assumption about how easily nodes can be corrupted. If current events dictate a reevaluation of this assumption (e.g., a newly discovered vulnerability in Windows), it is better to change t than to start over. To our knowledge ours is the first practical scheme for changing the threshold that works even when there is up to the threshold number of failures in each of the old and new groups.

The MPSS protocol is a substantial extension of the scheme of Herzberg et al. [2]. However, computing new shares is more involved in MPSS because we need to preserve the secret even when t in the old group and t nodes in the new group are faulty. Another difference is that MPSS works in an asynchronous network, where the non-arrival of a message does not imply that its sender is faulty.

MPSS includes an efficient protocol. It is designed to perform well for values of t that might occur in practice; analysis [3] indicates that this range is 1–10. The protocol requires $O(t^2)$ messages and $O(t^4)$ bytes of communication in total. Additionally, the protocol is designed to minimize latency and bandwidth utilization for the common case, when fewer than t failures occur.

Only two of the previous proactive secret sharing schemes are capable of moving the secret to a new group in a fully general setting. The scheme of Zhou et al [4] requires $O(\binom{n}{t})$ bytes of communication, and the scheme of Cachin et al [1] requires $O(t^4)$ bytes like MPSS, but $O(t^3)$ messages.

A full description of MPSS can be found in [3]. That paper also provides correctness arguments, performance analysis, and a more detailed discussion of related work.

*This research was supported by NSF ITR-6896772.

PODC'08, August 18–21, 2008, Toronto, Ontario, Canada.
ACM 978-1-59593-989-0/08/08.

2. REFERENCES

[1] C. Cachin, K. Kursawe, A. Lysyanskaya, and R. Strobl. Asynchronous verifiable secret sharing and proactive cryptosystems. In *Proc. CCS 2002*, pp. 88–97.

[2] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung. Proactive secret sharing, or how to cope with perpetual leakage. In *CRYPTO '95*, pp. 457–469.

[3] D. Schultz. Mobile proactive secret sharing. In *Master's Thesis, MIT*, 2007.

[4] L. Zhou, F. B. Schneider, and R. van Renesse. APSS: Proactive secret sharing in asynchronous systems. *TISSEC*, 8(3):259–286, Aug 2005.