Secure Group Communication in the presence of Byzantine faults

1. Problem abstract

In this document, we consider the problem of secure group consensus in the presence of Byzantine faults. Threshold cryptography scheme distributes n shares to participants such that *t* or more of them can uniquely determine the original secret. A dual threshold scheme (k, t, n) [3] considers k Byzantine failures such that at least t+1 honest participants concur in a threshold scheme. Traditionally, threshold schemes considered k = t+1 such that at least one honest parties took part in the threshold concurrence scheme. A strong threshold scheme will thus be able to detect and eliminate the misbehaving participant in the system. The problem here is to investigate the relation between threshold concurrence scheme developed against tolerance for malicious insiders Byzantine type attacks.

2. Background

Secure group communication framework and applications has been studied for numerous years now. Several secret sharing schemes and protocols are analyzed since the early literature by Shamir [1]. In his cryptographic study, Shamir considers the problem of sharing cryptographic keys among a group that intends to protect a secret such that it is easy to construct the key with a fewer trusted group members but never with just one. Shamir calls the scheme (t, n)-threshold scheme whenever at least *t*-trusted members together are able to recover the secret. Applications reliably choose t that is safe and convenience to use. In a similar treatise, Blakley [2] simultaneously introduced the notion of safeguarding the cryptographic keys from geometrical perspective. Shamir and Blakley addressed the same problem but from different perspectives.

3. Analysis framework

We intend the following in our analysis in the later sections.

- System model, Initial configuration and Transmission medium
- Failure model and Failure events
- Adversary lying models
- Factors to be studied: Efficiency, Integrity, Performance, Privacy, Scalability and Robustness.
- Comparison of related and their contributions

4. System Model

There are n processors (nodes) participating in secure group communication of which t processors are faulty. Fault processors may have been affected or taken control over by malicious external party so the behavior of these is unpredictable. We assume the malicious parties can take control over any of t processors in the system ranging from factors such as geographical neighborhood to ease of compromise.

We assume Byzantine type failure model with dynamic corruption model in which the adversaries choose the attacking hosts before initiating the attacks. Traditionally static corruption model is assumed explicitly or implicitly. However, more and more sources of attacks are intelligently and adaptively performed based on traffic patterns and control protocol between participants. Dynamic peer groups (like ad-hoc networks) cannot assume such adversary patterns especially when the trusted third party itself may be unavailable.

5. Adversary lying models

Distributed fault-tolerant models are required to considered a failure model that defines the number of hosts corrupted, the adversary corruption techniques such as static or dynamic, and how severe adversary effect is based on whether the adversary is active or passive.

In this, we consider the cryptographic ways that an adversary affects other users in the system.

Impersonation: An entity in a system is trusted by something that it possesses. Entity authentication is performed several ways based on: something known, something possessed or something inherent in the entity. An adversary passively monitoring the entity's identity can turn active by impersonating the original entity and obtaining unauthorized access to information. Known key attacks of the past keys may give sufficient clues about current secrets and thus impersonate

other users. Physical techniques such as biometrics, handwritten signature and fingerprints may prevent such impersonation.

Forgery: An adversary forges a signature of an user when he is able to submit a message that goes undetected as if the original player submitted the message. Non-repudiation schemes are required to prevent forgery. Forgery does not necessarily mean impersonation.

Unauthorized duplicate keys: It is possible for adversaries to crypt-analyze message and traffic patterns to determine the key sequence used in a group communication. Techniques such as known-plaintext attack can systematically determine the symmetric keys between participating members. In an interactive threshold secret sharing scheme, the host that finally determines the shared secret for the whole group is susceptible to passive attack. This results in an adversary duplicating the keys in launching active attacks.

Trusted third party attack: Online third-party distributes symmetric key information to participating entities for authentication. A challenge-response mechanism is initially exchanged to pre-authenticate. Pre-authenticated hosts are given symmetric keys for participating with other members. Authentication spoofing is one such 3rd party attack that involves dictionary-type attack in which the challenge and response are recorded between hosts and replayed whenever queried.

Non-Resilient protocol attack: In this case, the adversary is able to covertly initiate protocols between parties and influence their keys through alteration and deduct their final keys. In other cases, a legitimate party is deceived by identity of the party with which it shares the key and such protocols are not resilient against insider or outsider attacks.

6. Related work and contributions

Christian Cachin, Distributed Trust on the Internet, *Proceedings of DSN-2001*
Contribution: Distributed architecture for secure fault-tolerant services based on randomized byzantine agreements and atmoic broadcasts. Develops protocols for atomic and secure causal atomic broadcast; Byzantine agreement exploit concepts from threshold cryptography.

Model: (t, n)-threshold group communication; Byzantine group agreement; Trusted dealer to initialize secret state and distributed secret values; Asynchoronous communication links with no particular protocol timing assumptions; static number of servers; Uses a 2t+1 digitally signed responses to determine the outcome of the vote.

Failure model: Faulty processors, communication links and internal clocks are controlled by single adversary; static corruption model;

M. Reiter. Secure agreement protocols: Reliable and atomic group multicast in Rampart. In Proceedings of the 2nd ACM conference on computer and communication security, Nov. 1994.
Contribution: Develops theoretical notions for 3 new algorithms: echo-multicast, reliable-multicast and atomic multi-cast, all of them used for secure group communication with group membership changes. Corrupted parties and voting scheme is used as primaryconcurrence scheme $n > 2t/3$

Model: Static number of processes that can be corrupted; channels authenticated and protect the integrity (using cryptographic techniques such as signatures); sender and receiver can communicate even though corrupted parties exist - that is, no finite upper bounds on the message transmission times; no clock synchronization & timeout used in the protocols

Failure mode: Generic corruption model in which more than 2/3th of honest parties are required to ensure voting scheme of more than 2/3th concurrence in a group communication

M. Castro and B. Liskov. Practical Byzantine Fault Tolerance. In Proceedings of the Third Symposium on Operating Systems Design and Implementation (OSDI),New Orleans, LA, Feb. 1999.
Contribution: Asynchronous state machine replication protocol that tolerates regular byzantine faults. Each node has several replica (to the tolerable extent) and defines views to access the replica. The protocol is used to implement NFS operation with several back and performance of BFS (byzantine file system) as opposed to NFS is studied. Group membership changes are not considered

Model: Asynchronous distributed system with failure that fail delivery, delay, duplicate and deliver out of order;

Failure model: Adversary cannot delay messages arbitrarily long, is computationally bound compared to other honest nodes in the network and is unable to subvert cryptographic techniques, such as signature forging; Generic corruption model in which more than 2/3th of honest parties are required to ensure voting scheme of more than 2/3th concurrence in a group communication;

K.P. Kihlstrom, L. E.Moser and P. M. Melliar-Smith , ``The SecureRing Protocols for Securing Group Communication,'', Proceedings of the IEEE 31st Hawaii International Conference on System Sciences, Kona, Hawaii (January 1998).
Contribution: A reliable ordered message delivery and group membership protocol despite byzantine faults is provided. Studies the protocol performance when fault is detected; uses ring configuration to broadcast and maintain secure ordering of messages; Byzantine fault detection implemented.

Model: N distributed processes with a logical communication link imposed; asynchoronous protocol with no local clock synchronization

Failure model: For regular & membership configuration change, atleast 2n+1)/3 of total processors are required to compensate for the failure.

References

1. A.Shamir. How to share a secret. Communications of the ACM, 1979.

2. G.R. Blakley. Safeguarding cryptographic keys. In Proceedings of the National Computer Conference 1979, volume 48 of AFIPS Conference Proceedings, 1979.

3. Christian Cachin, Distributed Trust on the Internet, *Proceedings of DSN-2001*

4. M. Reiter. Secure agreement protocols: Reliable and atomic group multicast in Rampart. In Proceedings of the 2nd ACM conference on computer and communication security, Nov. 1994.

5. M. Castro and B. Liskov. Practical Byzantine Fault Tolerance. In Proceedings of the Third Symposium on Operating Systems Design and Implementation (OSDI),New Orleans, LA, Feb. 1999.

6. K.P. Kihlstrom, L. E.Moser and P. M. Melliar-Smith , ``The SecureRing Protocols for Securing Group Communication,'', Proceedings of the IEEE 31st Hawaii International Conference on System Sciences, Kona, Hawaii (January 1998).