Distributing Trust on the Internet

— Position Paper for S.O.S. Workshop 2004 —

Christian Cachin

IBM Research
Zurich Research Laboratory
CH-8803 Rüschlikon, Switzerland
cca@zurich.ibm.com

March 20, 2004

1 Introduction

Trustworthiness and dependability are more important than ever for online services. A service that is often unavailable is simply not useful and clients will stop using it. Even worse is a service that has been subverted by an attacker because it may give a wrong answer.

Hardware faults are no longer the prime source of system outages; existing systems address reliability successfully. Today, the causes lie in the absence of software robustness and in the ease with which many systems can be abused, given some malevolence. Preventing such problems and creating truly survivable systems requires new methods, which are beyond simple extensions of known principles.

One established way to enhance the availability of a service is to use *redundancy*: distribute the service among several replicas and use dependable replication algorithms to keep the replicas synchronized. A number of *group communication systems* support this approach [12]. Most of them have been designed for environments with random or "benign" faults.

The last few years have seen the development of a class of new group communication systems that also tolerate malicious attacks. These systems provide secure and survivable distributed services using the paradigm of active replication. Notable among them are three systems with a shared focus on service replication, on suitability for wide-area networks by employing an asynchronous model, and on using cryptography to tolerate malicious insiders. These systems are BFT, COCA, and SINTRA. All three can cope with up to a fraction of their member nodes behaving in arbitrary, i.e., *Byzantine*, fashion.

The BFT system, by Castro and Liskov [10], provides state-machine replication for an (almost) asynchronous network, where a fraction of the replicas may fail. It has been tested with several real-world services.

COCA, the Cornell Online Certification Authority developed by Zhou, Schneider, and van Renesse [13], provides a distributed service to issue certificates containing a digital signature; it addresses also the long-term maintenance of a shared cryptographic key that is used to issue the certificates.

SINTRA [6] is the work of Kursawe, Poritz, Shoup, and Strobl together with the author. SINTRA works in a fully asynchronous model and implements randomized Byzantine agreement [5] and atomic broadcast [4]. The key feature of SINTRA is its use of new cryptographic protocols, in particular, protocols for *threshold cryptography* where the secret key of a cryptosystem is distributed and unknown to any single entity. An important aspect is also the development of appropriate formal models that allow to capture the negligible (but non-zero) failure probabilities inherent in any cryptographic approach [2].

2 State-of-the-Art

SINTRA [6] is based on the *state-machine replication* method, where a request to a service is processed by all replicas and the client infers the result from a majority of the received answers. This work is the first to address a fully asynchronous network [1] and to tolerate up to one third of faulty replicas that may exhibit Byzantine faults (which is the theoretical optimum).

The core component of SINTRA is a protocol [5] for the classical (binary) *Byzantine agreement* problem. It makes use of cryptography, specifically, protocols for threshold signatures and coin-tossing, and is both practical and theoretically optimal.

The asynchronous *atomic broadcast* of SINTRA [4] builds on a protocol for *multi-valued Byzantine agreement* [4], which itself uses binary Byzantine agreement. The atomic broadcast protocol proceeds in global rounds and uses one multi-valued agreement per round to agree on a batch of broadcast messages.

A faster *"optimistic" asynchronous atomic broadcast* [11] is also provided by SINTRA; it is similar to the atomic broadcast in BFT [10]. Under normal circumstances, this protocol runs in an optimistic mode, where the ordering of the broadcast messages is determined by a leader. This mode has low message and computational complexities. If the leader is not responsive in the view of sufficiently many other replicas, a new leader is chosen and some broadcast messages are ordered using Byzantine agreement, before the optimistic mode resumes.

Both of SINTRA's atomic broadcast protocols guarantee safety and liveness without making *any* timing assumptions or using any type of failure detector (in contrast to BFT).

Asynchronous atomic broadcast and threshold cryptography have recently been applied to the design and implementation of a secure distributed domain name service for maintaining DNS zone data [7]. The new service is able to provide fault tolerance and security even in the presence of a fraction of corrupted name servers, avoiding any single point of failure, and solves the problem of storing zone secrets online without leaking them to a corrupted server, while still supporting secure dynamic updates.

3 Challenges

All three service replication methods mentioned above work in a static group model and require manual setup. The systems themselves do not provide mechanisms to change the membership of the group in a secure way. Usually, a trusted "dealer" is needed to perform this operation. The degree of trust placed in the dealer varies; SINTRA probably makes the biggest assumptions about the dealer among the three because the dealer is used to select and distribute several threshold-cryptographic keys.

However, practicality and usability demand that group changes be carried out automatically.

Challenge 1: Develop a secure asynchronous group membership system that tolerates Byzantine faults (specification, protocols, and prototype systems).

Such a system has to make sure that no cryptographic keys leak to an adversary. Some work toward this goal is already available with proactive cryptosystems [8], which periodically refresh the shared secret key in order to protect it against a *mobile* adversary, who can move from one server to another and corrupt all servers during the lifetime of the system. The refresh protocols can also be used to redistribute a key in a new group. But most of this work has focused on synchronous networks with broadcast, and practical protocols for asynchronous networks have only recently been proposed [3].

Another problem inherent in the replication approach is scalability. None of the existing group communication systems that tolerate Byzantine faults and that have strong consistency can support 1000's of nodes that are connected by a wide-area network. This problem stems from the use of protocols in which every replica sends a message to every other replica, which creates $O(n^2)$ messages in a group of n nodes.

Challenge 2: Develop a secure and dependable group communication system that does not suffer from the $O(n^2)$ phenomenon (new models, protocols, and systems).

Recent progress toward more efficient threshold cryptographic protocols has been made in a protocol for distributed cryptographic key generation in large networks [9]. It is the first threshold cryptographic protocol where every replica needs only *polylog*(n) messages.

References

[1] C. Cachin, "Distributing trust on the Internet," in *Proc. International Conference on Dependable Systems and Networks (DSN-2001)*, pp. 183–192, 2001.

[2] C. Cachin, "Modeling complexity in secure distributed computing," in *Future Directions in Distributed Computing* (A. Schiper, A. A. Shvartsman, H. Weatherspoon, and B. Y. Zhao, eds.), vol. 2584 of *Lecture Notes in Computer Science*, pp. 57–61, Springer, 2003.

[3] C. Cachin, K. Kursawe, A. Lysyanskaya, and R. Strobl, "Asynchronous verifiable secret sharing and proactive cryptosystems," in *Proc. 9th ACM Conference on Computer and Communications Security (CCS)*, pp. 88–97, 2002.

[4] C. Cachin, K. Kursawe, F. Petzold, and V. Shoup, "Secure and efficient asynchronous broadcast protocols (extended abstract)," in *Advances in Cryptology: CRYPTO 2001* (J. Kilian, ed.), vol. 2139 of *Lecture Notes in Computer Science*, pp. 524–541, Springer, 2001.

[5] C. Cachin, K. Kursawe, and V. Shoup, "Random oracles in Constantinople: Practical asynchronous Byzantine agreement using cryptography," in *Proc. 19th ACM Symposium on Principles of Distributed Computing (PODC)*, pp. 123–132, 2000.

[6] C. Cachin and J. A. Poritz, "Secure intrusion-tolerant replication on the Internet," in *Proc. Intl. Conference on Dependable Systems and Networks (DSN-2002)*, pp. 167–176, June 2002.

[7] C. Cachin and A. Samar, "Secure distributed DNS." To appear in *Proc. Intl. Conference on Dependable Systems and Networks (DSN-2004), Florence, Italy*. Preliminary version available as IBM Research Report, RZ 3509, Oct. 2003.

[8] R. Canetti, R. Gennaro, A. Herzberg, and D. Naor, "Proactive security: Long-term protection against breakins," *RSA Laboratories' CryptoBytes*, vol. 3, no. 1, 1997.

[9] J. Canny and S. Sorkin, "Practical large-scale distributed key generation," in *Advances in Cryptology: Eurocrypt 2004* (C. Cachin and J. Camenisch, eds.), vol. 3027 of *Lecture Notes in Computer Science*, pp. 139–154, Springer, 2004.

[10] M. Castro and B. Liskov, "Practical Byzantine fault tolerance and proactive recovery," *ACM Transactions on Computer Systems*, vol. 20, pp. 398–461, Nov. 2002.

[11] K. Kursawe and V. Shoup, "Optimistic asynchronous atomic broadcast." Cryptology ePrint Archive, Report 2001/022, Mar. 2001. `http://eprint.iacr.org/`.

[12] D. Powell (Guest Ed.), "Group communication," *Communications of the ACM*, vol. 39, pp. 50–97, Apr. 1996.

[13] L. Zhou, F. B. Schneider, and R. van Renesse, "COCA: A secure distributed online certification authority," *ACM Transactions on Computer Systems*, vol. 20, no. 4, pp. 329–368, 2002.