Towards Efficient and Robust BFT Protocols

Lucas Perronne, Sara Bouchenak

Towards Efficient and Robust BFT Protocols

Lucas Perronne
Univ. Grenoble Alpes, LIG
Grenoble, France
Lucas.Perronne@imag.fr

Sara Bouchenak
Univ. Lyon, INSA Lyon, LIRIS
Lyon, France
Sara.Bouchenak@insa-lyon.fr

***Abstract*—Byzantine Fault Tolerant (BFT) protocols rely on state machine replication to handle arbitrary behaviors. Significant efforts have been recently made to strengthen these protocols in order to minimize the performance degradation in presence of faulty components. In this paper, we focus on the potential damages that could be introduced from the client side of such protocols. In order to deal with this specific kind of threats, BFT protocols rely on request authentication to provide access control. Nevertheless, byzantine clients may benefit from the weakness of the underlying authentications mechanisms in order to tamper with the performance of replicated systems. We describe the main reliability issues that can be introduced by faulty clients in current BFT protocols, and we argue against the systematic usage of digital signatures. Finally, we propose a new policy in order to avoid the overhead due to systematic signatures verifications.**

I. Introduction

With the expansion of cloud computing, concerns such as availability, liveness and security are attracting more interest. In such environments, hosted systems are replicated on different servers to prevent data loss and provide various guarantees to the customers. In order to provide an enhanced level of reliability, BFT protocols rely on State Machine Replication to handle unpredictable events, termed as arbitrary or byzantine faults. By definition, these protocols assure (i) *liveness* - eventual execution of correct client requests; and (ii) *safety* - consistency across replicas.

The Byzantine generals problem was first described by Lamport et. al. [6], stating that that a Byzantine agreement requires $3f + 1$ replicas to handle up to f arbitrary faults under partial synchrony i.e., given a known fixed upper bound on message delivery time. In the last decade, many improvements have been proposed to the state-of-the-art protocols. We can broadly categorize these contributions in two groups: protocols optimizing the performance under fault-free settings [2], and protocols minimizing the performance degradation introduced by faulty components [1], [5]. In this paper, we target both of these concerns at the same time, i.e., improving robustness to malicious components, while maintaining a high level of performance in fault-free settings. To do so, we remove the systematic usage of digital signatures, which is the main performance bottleneck of robust protocols compared to optimistic ones.

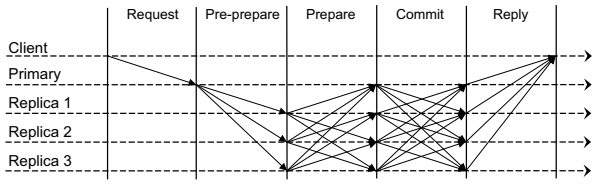

Fig. 1: Communication pattern of the PBFT protocol

II. Background of BFT

A. Overview of BFT protocols

In order to handle contention (concurrent arrival of client requests), Leader-based BFT protocols like PBFT [4], cf. figure 1, rely on a dedicated replica, called primary [4], [1], [5], [3]. First, the primary assigns sequence numbers to incoming requests – pre-prepare in figure 1. In a second round, a consensus involving all replicas is performed, in order to reach an agreement on the proposed sequence numbers – prepare in figure 1. Finally, when the agreement is achieved, all replicas are ensured to execute the same requests in the same order - commit in figure 1. If no agreement is obtained in time, a *View-change* is triggered by correct backups (non-primary replicas) to replace the faulty primary, e.g., crashed or unresponsive primary.

B. The Problem of MAC Attacks and its Solution

If client requests authentication is performed using Messages Authentication Codes (MACs), a vector of size $3f+1$ is computed by the clients, and appended to their requests. To ensure that a client is effectively the author of an incoming request, each replica authenticates that request by verifying its dedicated MAC among the $3f+1$ elements. *View-changes* can be intentionally triggered by faulty clients if these clients fill their authenticators with a corrupted MAC for the primary, and at least $f+1$ correct MACs among the $3f$ MACs dedicated to backups. As soon as a correct backup receives a request it can authenticate, it expects an agreement to be eventually obtained, however, the request which *appears* correct for the backup itself will not be successfully authenticated by the primary. The primary will not propose a sequence number for that request, $f + 1$ backups will eventually conclude that the primary is unresponsive, a *View-change* procedure will be triggered, and the primary will be replaced. Such denial-of-service attacks

TABLE I: Peak throughput measured for various protocols when running the 0/0 micro-benchmark [4] in closed loop with 100 clients.

Protocols	Authenticators	Digital signatures
BFT-SMART [3]	41238	14718
COP [2]	50628	13414
Aardvaark [5]	52828	13848

can seriously tamper with the performance of protocols, especially if many clients compute a continuous load of corrupted authenticators. In order to forbid clients from triggering *View-changes*, client requests are now systematically authenticated with digital signatures by robust protocols, such as Aardvark or RBFT [5], [1]. In such settings, a single signature is generated per request, and is consistently verified by all replicas.

C. Problem Illustration

Replacing *Authenticators* with digital signatures introduces an extra computational cost for both clients and replicas. To underpin this statement, we ran a series of experiments involving three BFT protocols on which we performed request authentication either with MACs or digital signatures (figures I). We used the same cryptographic primitives as BFT-SMART for our experiments [3], which are Hmac:MD5 for MACs, and RSA:Sha1 for digital signatures.

III. Efficient and Robust BFT

A. Overall Approach

To avoid the extra computational cost of digital signatures over MACs while disabling the ability of byzantine clients to trigger *View-changes*, we designed new policies for *request transmissions* and *primary View-changes*. First, we only rely on digital signatures for request authentication when MAC-authenticated requests are unable to commit. Then, we do not trigger *View-changes* on backup replicas based on MAC-authenticated requests expectations.

B. Underlying Mechanisms

Disabling Mac Attacks. *View-changes* are not triggered on backups when an expected *MAC-authenticated* request does not commit. While such solution solves the problem of MAC attacks, it also removes the ability of correct backups to replace an unresponsive primary.

Replacing Unresponsive Primaries. Clients use digital signatures for requests *retransmission*. *View-changes* are triggered on backups when an expected *signed* request does not commit. If a primary is unresponsive, it will be replaced as soon as the client retransmits its request to the system. Nevertheless, a malicious primary may benefit from this solution in order to force the retransmission of all incoming MAC-authenticated requests into signed requests.

Dealing With Malicious Primaries. Backups monitor the ratio of successful *MAC-authenticated* requests per clients and primaries. *View-changes* are triggered on backups when too many clients fail to commit their *MAC-authenticated* requests.

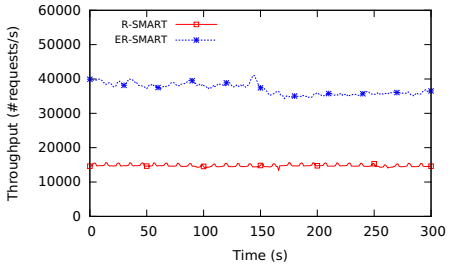

Fig. 2: Throughput evaluation when running the 0/0 micro-benchmark [4] in closed loop with 100 clients

C. Design Principles

Since we focus on *requests management* and *View-change* policies without modifying the core agreement itself, our approach can be integrated in most BFT protocols to increase reliability without the expensive overhead induced by systematic signature computations.

IV. Preliminary results

In figure 2, R-SMART refers to the robust BFT-SMART implementation, where digital signatures are enabled for requests authentication. ER-SMART refers to our modified efficient and robust BFT-SMART implementation. At time 150s, 10 clients start performing MAC-Attacks on ER-SMART. We observe that no view-change is triggered, while the throughput decreases from 39k to 36k, because only 90 clients keeps sending correct requests.

V. Conclusion

We propose in this paper a solution to benefit from an enhanced level of reliability, without relying on systematic digital signatures for request authentication. We also performed an experiment to show early results of our proposal on the BFT-SMART prototype.

References

[1] P.-L. Aublin, S. B. Mokhtar, and V. Quéma. RBFT: Redundant Byzantine Fault Tolerance. In *ICDCS*, 2013.

[2] J. Behl, T. Distler, and R. Kapitza. Consensus-oriented parallelization: How to earn your first million. In *Middleware*, 2015.

[3] A. Bessani, J. Sousa, and E. E. Alchieri. State machine replication for the masses with BFT-SMaRt. In *DSN*, 2014.

[4] M. Castro and B. Liskov. Practical Byzantine Fault Tolerance. In *OSDI*, pages 173–186, 1999.

[5] A. Clement, E. L. Wong, L. Alvisi, M. Dahlin, and M. Marchetti. Making Byzantine Fault Tolerant Systems Tolerate Byzantine Faults. In *NSDI*, 2009.

[6] L. Lamport, R. E. Shostak, and M. C. Pease. The Byzantine Generals Problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, 1982.