Chapter 7
Consistency Is Not Enough in Byzantine Fault Tolerance

Wenbing Zhao
Cleveland State University, USA

ABSTRACT

The use of good random numbers is crucial to the security of many mission-critical systems. However, when such systems are replicated for Byzantine fault tolerance, a serious issue arises (i.e., how do we preserve the integrity of the systems while ensuring strong replica consistency?). Despite the fact that there exists a large body of work on how to render replicas deterministic under the benign fault model, the solutions regarding the random number control are often overly simplistic without regard to the security requirement, and hence, they are not suitable for practical Byzantine fault tolerance. In this chapter, the authors present a novel integrity-preserving replica coordination algorithm for Byzantine fault tolerant systems. The central idea behind our CD-BFT algorithm is that all random numbers to be used by the replicas are collectively determined, based on the contributions made by a quorum of replicas, at least f+1 of which are not faulty.

INTRODUCTION

In Byzantine fault tolerance (BFT), a core concern is to ensure the consistency of replicas despite malicious attacks from the clients and compromised replicas (Zhao, 2014). This is accomplished by totally ordering incoming requests and by rendering the replica's operations deterministic (Zhang et al., 2011). In the presence of application non-determinism, such as the access of local clocks, replicas are rendered deterministic by forcing all non-faulty replicas to use the same values either supplied by the primary or computed deterministically. While this approach works well for some applications, such as a replicated file system, doing so could lead to the compromise of the service integrity for applications that rely on the use of random numbers.

For example, consider an Internet application that relies on the use of session-ids for stateful interactions between the server and its clients. As pointed out in (Dorrendorf, Gutterman, & Pinkas, 2007), if the session-id of an active session can be predicted, the client's session with the server could be hijacked,

DOI: 10.4018/978-1-5225-7359-3.ch007

which could lead to the leak of confidential information regarding the client, such as name, address, and the order history, or unauthorized orders (if the one-click option for placing orders is enabled). The session-id may be predicted by searching the limited entropy space if weak random bits are used in an application. For example, the authors of (Dorrendorf, Gutterman, & Pinkas, 2007) reverse-engineered a version of Tomcat (a popular Java Servlet Engine) and the related operations in a Window's based Java Virtual Machine. They could attack the system by performing about 251 searches in finding an active session-id.

Therefore, it is critical not to weaken the strength of the random bits essential for the integrity of their operations when replicating these systems for Byzantine fault tolerance. For a sound coordination algorithm, it is essential to enable each replica to access its own entropy source and maintain its independence in such operations. However, this desire is in conflict with the basic requirement for state machine replication (Schneider, 1990), which mandates that the replicas must be deterministic or rendered deterministic to maintain strong replica consistency. The conflicting requirements for security and replication must be reconciled to avoid the dilemma of either favoring security over high availability by not performing state machine replication of the systems, or trading security for high availability by removing the randomness of the systems in order to perform state machine replication.

In this chapter, we present a novel replica coordination algorithm, referred to as the Collective-Determination BFT algorithm, or CD-BFT algorithm in short, towards the reconciliation of the conflicting requirements for security and for strongly consistent replication. The central idea behind this algorithm is that all random numbers to be used by the replicas are collectively determined, and furthermore, the determination is based on the contributions made by a quorum of replicas, at least one of which is correct.

In the CD-BFT algorithm, the replicas first reach a Byzantine agreement on the set of contributions from replicas, and then apply a deterministic algorithm, such as the bitwise exclusive-or operation (Young & Yung, 2004), to compute the final random value. The freshness of the random numbers generated is application dependent. Our approach does not alter the freshness of the random numbers. If a pseudo-random number generator is used, it should be periodically reseeded from a good entropy source.

BACKGROUND

An arbitrary fault is often referred to as a Byzantine fault. The term was introduced in (Lamport, Shostak, & Pease, 1982) to highlight a specific faulty behavior that a Byzantine faulty process may disseminate conflicting information to other processes. For example, a compromised process might exhibit such Byzantine faulty behavior. Byzantine fault tolerance refers to the capability of tolerating Byzantine faults in a system. It can be achieved by introducing sufficient redundancy into the system and by using a sophisticated replica coordination algorithm that can cope with Byzantine faulty replicas and clients. A basic requirement for such an algorithm is to ensure that all server replicas agree on the total ordering of the requests received despite the existence of Byzantine faulty replicas and clients. Such an agreement is often referred to as Byzantine agreement (Lamport, Shostak, & Pease, 1982).

Recently, a number of efficient BFT algorithms (Castro & Liskov, 2002; Kotla et al., 2007; Yin et al., 2003) have been proposed. Our CD-BFT algorithm is derived from the PBFT algorithm and we use the same system model as that in (Castro & Liskov, 2002). The PBFT algorithm operates in an asynchronous distributed environment. The safety property of the algorithm, *i.e.,* all correct replicas agree on the total ordering of requests, is ensured without any assumption of synchrony. However, for the algorithm to