Real-Time, Byzantine-Tolerant Information Dissemination in Unreliable and Untrustworthy Distributed Systems

Kai Han*, Guanhong Pei*, Binoy Ravindran*, E. D. Jensen‡

*ECE Dept., Virginia Tech
Blacksburg, VA 24061, USA
{khan05,somehi,binoy}@vt.edu

‡The MITRE Corporation
Bedford, MA 01730, USA
jensen@mitre.org

***Abstract*—In unreliable and untrustworthy systems, information dissemination may suffer network failures and attacks from Byzantine nodes which are controlled by traitors or adversaries, and can perform destructive behaviors. Typically, Byzantine nodes together or individually "swallow" messages, or fake disseminated information. In this paper, we present an authentication-free, gossip-based real-time information dissemination mechanism called RT-LASIRC, in which "healthy" nodes utilize Byzantine features to defend against Byzantine attacks. We show that RT-LASIRC is robust against blackhole and message-faking attacks. Our experimental studies verify RT-LASIRC's effectiveness.**

I. Introduction

In a decentralized, network-based system, information dissemination (i.e., one node spreads important information to the entire system) may suffer attacks from malicious nodes. Attacks where a traitor or an adversary has full control of an authenticated node, and can perform destructive behaviors to disrupt the system are referred to as Byzantine attacks [1]. A node showing Byzantine behaviors is called a Byzantine node. Byzantine nodes are more difficult to deal with than other attackers [2], [3].

We consider distributed systems where authentication mechanisms (including any kind of encryption) are unable to defend against Byzantine nodes. A "healthy" node cannot trust its peers, because it does not know whether another node is a friend, a traitor, or an adversary. In the rest of the paper, we use the terms "Byzantine nodes" and "Byzantine attackers", interchangeably.

Furthermore, we consider systems that use *unreliable networks* (e.g., those without a fixed network infrastructure, including mobile, ad hoc and wireless networks) with dynamically uncertain properties. These uncertain properties, which are application- and network-induced, include arbitrary node failures, transient and permanent network failures, and varying packet drop behaviors. Example such systems that motivate our work include US DoD's Network-Centric Warfare system [4].

Gossip-based protocols offer a scalable and reliable message dissemination design paradigm for large-scale, unreliable systems. It "fights" non-determinism (i.e., unpredictable message losses and node failures) with non-determinism (i.e., randomly selecting sending targets)—duplicated messages guarantee propagation speed. However, gossip cannot protect against Byzantine attacks—if attackers fake information, gossip "helps" to disseminate the false information instead of the correct one.

In this paper, we present Real-Time LASIRC (RT-LASIRC), a gossip-based, real-time, and Byzantine-tolerant information dissemination mechanism. RT-LASIRC features gossip's same robustness in large-scale, unreliable distributed systems. In addition, it also provides a set of mechanisms to detect and defend possible Byzantine attacks in gossip-based message dissemination processes.

Particularly, RT-LASIRC does not gain help from any authentication mechanism. This justification stems from a number of reasons. Protection by means of authentication (e.g., cryptographic signatures) might be voided if the corruption comes from a internal traitor, might be impossible if data is generated by low powered nodes, e.g., sensors, or might simply be too costly to employ. Furthermore, different from previous authentication-free information dissemination research [1]–[3], RT-LASIRC does not assume any limitation on the number of attackers, and timely disseminates information within required deadlines.

RT-LASIRC evolves from our former work "LASIRC" mechanism [5], but it has major modifications. First, RT-LASIRC focuses on disseminating information throughout the entire system, while LASIRC is used for point-to-point message propagation. Second, RT-LASIRC has real-time properties, while LASIRC is a non-real-time mechanism. Third, the key component of LASIRC—Byzantine node detector—is redesigned in

the RT-LASIRC mechanism.

The rest of the paper is organized as follows: In Section II, we discuss possible Byzantine attacks in gossip-based information dissemination. We then present our Byzantine node detector in Section III. Sections IV describes and analyzes the RT-LASIRC mechanism. In Section V, we illustrate experimental results. We conclude the paper in Section VI.

II. Byzantine Attacks

A. Gossip Rationale

A node initiates a gossip process by starting a series of synchronous gossip rounds. During each round, nodes holding information randomly select a set of neighbors to inform, without requiring any confirmation regarding message reception. The number of gossip rounds (or R), and the number of selected neighbors (i.e., the "fan-out" number, or F) are determined by the original sender [6]. Though robust to network uncertainties, gossip cannot protect against Byzantine attacks—if attackers fake information, the receiving victim nodes "help" to disseminate the false information to the entire system. Figure 1 shows the gossip protocol used in RT-LASIRC.

Algorithm 1: Gossip Emission (GOSSIP())

```
1 On gossiping a message msg:
2   while R != 0 do
3       Every gossip round r, randomly select F_r target nodes;
4       for each m ∈ [1, ..., F_r ] do
5           SEND(target_i, msg);
6       R = R − 1;
```

B. Message Structure

A message contains the original sender's (the information dissemination initiator) identifier (ID), the selected target node identifiers (ID), the gossip round number R, the fan out number (F_r), and information for dissemination, as shown in Table I.

TABLE I
Dissemination Message Structure

Original ID, Target IDs, R, F_r (Gossip Parameters)
Information for dissemination

C. Byzantine Attack Types

An individual Byzantine node may "swallow" messages—i.e., when it receives a message, it does not gossip the message contents to other nodes. Such behavior is called a *Black Hole* attack. In addition, two or more attackers may collude together to form a larger "*Black Hole*". We call these two attacks *Black Hole Class (BHC)* attacks.

A Byzantine attacker may also fake information, and gossips this false information into the system, causing the victim receivers continue to disseminate incorrect messages. Such attack is called an *Message-Faking (MF)* attack.

An MF attack is more harmful than a BHC one. Unlike BHC attackers that only "swallow" messages, an MF attacker directly propagates incorrect information. Since BHC attacks can be generally regarded as message losses/node failures, and gossip is robust to such network uncertainties, BHC attackers may not cause serious results. While MF attackers may contaminate the entire system by disseminating false messages. Figure 1 shows BHC and MF attacks in the gossip-based message dissemination.

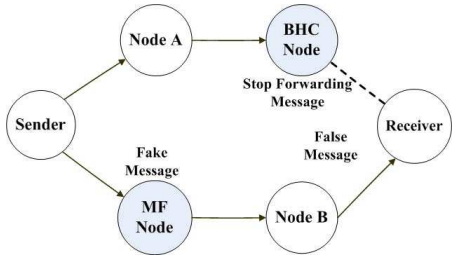

Fig. 1. Byzantine Attacks in Gossip-based Message Dissemination

III. Byzantine Attacker Detection

Because gossip protocols are robust to BHC attacks, in this section, we focus on MF Attacker Detectors (MFADs).

The idea of MFAD design comes from the feature of MF attackers—they receive correct information, but gossip incorrect one. Each node is equipped with an MFAD. To detect MF attackers hiding in the system, a non-attacker node ("healthy node") is supposed to initiate its own MFAD first, sets the gossip round R to 1, and broadcasts its information to the entire system. Since $R = 1$, an attacker must broadcast its fake messages to all other nodes in one gossip round. Therefore, the initial MFAD can easily identify an MF attacker if it receives that attacker's message. For those activated MFADs located on other "healthy" nodes, since they receive the original broadcasted message from the initial "healthy" node, they can also identify MF attackers by comparing every received message with the original one. The faked information can be arbitrary, instead of the simple YES/NO answer in LASIRC mechanism. MFADs are described in Algorithms 2 and 3, respectively.

If an activated MFAD does not receive the original message from the initial MFAD, it cannot identify any MF attacker in the later process. If an MFAD does not hear from another node, it cannot regard that node

Algorithm 2: Inititial MF Attacker Detector

```
Node i puts its own information in disseminated messages;
Node i sets R = 1;
At the first gossip round: Broadcast messages to the system;
After the second gossip round: Check information in every
  received message;
if the information is changed then
    Identify the message sender as an MF attacker;
```

Algorithm 3: Activated MF Attacker Detector

```
At the first gossip round: Receive messages from the initial
  MFAD;
At the second gossip round: Broadcast messages once;
After the second gossip round: Compare information in every
  received message with the one in the original message;
if the information is changed then
    Identify the message sender as an MF attacker;
```

as an MF attacker. Therefore, the effectiveness of an MFAD depends on the message loss ratio.

We can partially overcome the communication difficulty by keeping the initial MFAD broadcasting after the first round. In this way, nodes which do not receive original messages have more chances to receive correct information, and thus effectively enhance the success ratio of MF attacker detection.

IV. RT-LASIRC Mechanism

MFADs cannot exhaust attackers if message loss ratio is larger than zero, which is common in unreliable networks. Since gossip is robust to message losses and node failures, it is relatively easy to deal with hiding BHC attackers. However, gossip cannot handle hiding MF attackers, which is more dangerous. Therefore, it is necessary to design a gossip-based information dissemination mechanism to defend against MF attacks.

A. Message-Faking Attack Model

We introduce definitions for nodes participating in message dissemination:

Definition 1 (Healthy Node): A node that is not an MF attacker.

Definition 2 (Host (H)): A node that has received one or more messages.

Definition 3 (Launcher (L)): The initiating sender of the messages.

Definition 4 (Attacker (A)): An MF attacker that tries to spread a false message.

Definition 5 (Susceptible (S)): A healthy node that has not received any message.

Definition 6 (Infective (I)): A healthy host that believes false information.

Definition 7 (Recovered (R)): A healthy host that knows its received information is correct, or always sends correct messages.

Definition 8 (Consumer (C)): Every healthy node at the end of the information dissemination process.

We introduce definitions for disseminated messages:

Definition 9 (Agent): A fake message from an attacker.

Definition 10 (Virus): A fake message that possibly turns a susceptible into an infective.

Definition 11 (Antibiotic): An agent that turns a susceptible/infective into a recovered.

Definition 12 (Berry): A correct message.

Definition 13 (Vaccine): A berry that possibly turns a susceptible into a recovered.

Note that we assume that every healthy node ONLY checks its first received message, discarding all of the following messages with the same message ID, unless the message is an antibiotic, or contains a change defined in the message definitions.

A host is said to be immune to an attacker, if its MFAD has identified that attacker. A host gets immunized when its first received message is a vaccine.

B. Real-Time Gossip

We describe the protocol by first introducing the necessary definitions.

Definition 14: Gossip Round r: Denotes the r^{th} gossip time interval, at the beginning of which nodes send out messages. All messages are considered to arrive at their destination nodes when the round r ends. We assume that the message delay follows a non-negative distribution, e.g., the Gamma distribution [7]. Many distributions have infinite tails, and therefore, to determine the length of a gossip round, users need to decide a termination time point t_{end}, after which message arrivals can be ignored. This is done by determining a threshold on the message arrival ratio, which is referred to as Θ. For instance, if $\Theta = 98\%$, we can determine the relative t_{end} in a given distribution.

Definition 15: I_r : Denotes the number of informed nodes at the end of gossip round r.

Definition 16: U_r : Denotes the number of uninformed nodes at the end of gossip round r.

Definition 17: F_r : The number of messages a node sends out at the beginning of gossip round r.

Let N be the total number of nodes in the system. As a way similar to [7], we compute the expected number of uninformed nodes at the end of gossip round r:

$$U_r = U_{r-1} \times (1 - \frac{F_r}{N-1})^{I_{r-1}} \tag{1}$$

When $F_r \ll N - 1$, we have:

$$U_r = U_{r-1} \times \exp{(\frac{-F_r \times I_{r-1}}{N-1})} \tag{2}$$

The fan out and the number of messages issued during gossip round r (M_r), are shown in Equations 3

and 4, respectively:

$$F_r = \frac{N-1}{I_{r-1}} \times \ln(\frac{U_{r-1}}{U_r}) \quad (3)$$

$$M_r = F_r \times I_{r-1} = (N-1) \times \ln(\frac{U_{r-1}}{U_r}) \quad (4)$$

Different from gossip protocols with fixed fan out number at each round, here, F_r can be adjusted by users.

Lemma 1: The number of messages issued during all gossip rounds is $\Theta(N \log N)$.

Proof: From Equation 4, we have

$$\Sigma_{r=1}^{r=R} M_r = (N-1) \times \Sigma_{r=1}^{r=R} \ln(\frac{U_{r-1}}{U_r}) \quad (5)$$

where R is the total number of gossip rounds.

The number of issued messages during all gossip rounds is $\Theta(N \log N)$. ■

Theorem 2: The number of issued messages is independent of R, I_r, U_r or F_r.

Proof: The result is directly derived from Lemma 1. ■

C. RT-LASIRC Mechanism

We now describe the RT-LASIRC mechanism. Algorithm 4 shows how a launcher works. A launcher is the initiating sender of disseminated messages. It holds the correct information, so it cannot be infected by an MF attacker.

Algorithm 4: Launcher

```
1 Initialize information in the message msg;
2 GOSSIP(msg);
```

When an MF attacker receives a message, it is activated. If the sender is another attacker, it follows the sender. Otherwise, it changes the information contained in the message. Algorithm 5 describes the MF attacker.

Algorithm 5: MF Attacker

```
1 On receiving a disseminated message msg:
2   if the sender is not an MF attacker then reverse the
    answer in msg;
3   GOSSIP(msg);
```

Algorithm 6 shows healthy node behaviors in the RT-LASIRC mechanism. A susceptible turns into a recovered if its first received message is a vaccine (from a recovered), or turns into an infective if that message is a virus (from an infective or an MF attacker).

Consumer behavior in the RT-LASIRC mechanism is shown in Algorithm 7. After gossip finishes, if a healthy node still cannot identify itself as a recovered (knowing the correct information), it will count the number of the same information in received messages.

Algorithm 6: Susceptible, Infective and Recovered

```
1  On receiving the first message msg:
2    GOSSIP(msg);
3  On receiving another message with the same message ID:
4    if the sender has sent message before then
5      if the contained information changes then
6        adopt information in new message; //Identify
         sender has changed from an infective to a
         recovered
7        reverse the answer in msg;
8        GOSSIP(msg); //Change from an infective to a
         recovered
9    if the sender is an identified MF attacker then
10     if the information in the first msg is the same as the
       one in this attacker's message then
11       discard its current information and adopt the one
         in msg;
12       GOSSIP(msg); //Change from an infective to a
         recovered
```

If the consumer is optimistic, it believes MF attackers occupy less than half of the total number of nodes. Then, it will select the information in most received messages. Otherwise, the consumer is pessimistic, and it will select the information in less received messages.

Algorithm 7: Consumer

```
1 After gossip finishes:
2   if the consumer has not identified itself as a recovered then
3     select the information in most(less) received messages;
      //Optimistic (Pessimistic) consumer
```

V. Experimental Studies

We simulated the RT-LASIRC mechanism in a 700-node system, in which every node is reachable to others. We considered a scenario where BHC attacks resulted in 15% of message loss during propagation, and compared RT-LASIRC's MFADs with our former LASIRC's.

Figure 2 shows the Detection Ratio (DR; the ratio of number of MF attackers detected by a node to the total number of MF attackers) along with the Message Loss Ratio (MLR). We observe that DR on both RT-LASRIC and LASIRC decrease when MLR increases, because MFAD depends on received messages; thus its performance degrades when MLR increases. In addition, we observe that RT-LASIRC's DR is higher than that of LASIRC's. This is because RT-LASIRC's initial MFAD continues broadcasting during the whole gossip process, and balances the lost messages with gossiping for more than once. We also observe that MFAD cannot detect all MF attackers if MLR is larger than 0. Therefore, we need RT-LASIRC mechanism to deal with hiding MF attackers that survive this detection.

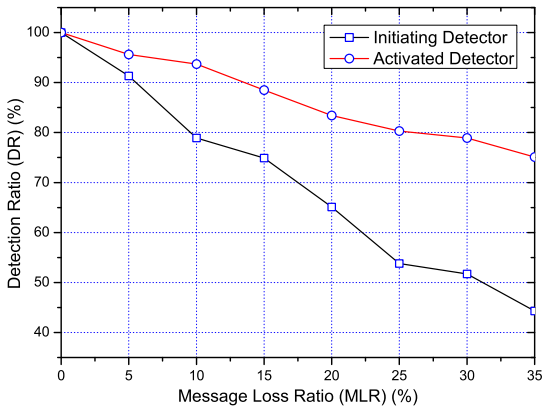

Fig. 2. LASIRC MF Attacker Detection

Figure 3 shows Infective Ratio (IR; the ratio of number of infectives to the number of healthy nodes) along with the total number of gossip round R and gossip round number r (Definition 14). This figure gives a general sense of RT-LASIRC's real-time properties. The required number of gossip rounds (R) is set from 2 to 10(the time period for RT-LASIRC to disseminate messages is from $2R$ to $10R$). We may observe that as R increases, IR sharply increases to nearly 0%, since an infective has more chance to become an recovered if it has more time (Algorithm 6). However, even when dissemination time is not sufficient, IR remains at a low value (14.87% when $R = 2$).

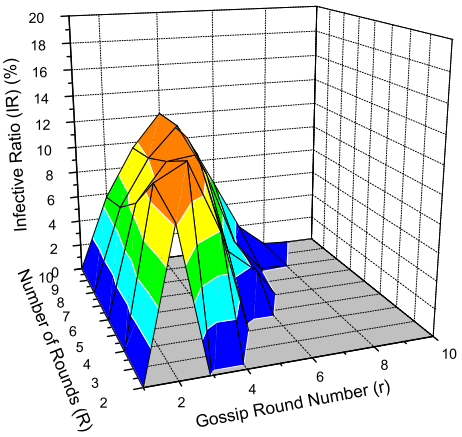

Fig. 3. Infective Ratio under 35% Attackers

We can make a clearer observation in Figure 4 ($R = 8$) that shows the cross-section of Figure 3. We observe that as r increases, IR keeps increasing till the fourth round ($IR = 12.17\%$), then it quickly decreases to nearly zero ($IR = 0.0001\%$ when $r = 8$). In early gossip rounds, IR increases because some healthy nodes first receive false messages and then turn into infectives. As gossip process precedes, the RT-LASIRC mechanism automatically turns infectives into recovereds according to Algorithm 6 and 7. This is because MFADs make nodes identify a number of MF attackers, and help them turn viruses into antibiotics during message dissemination.

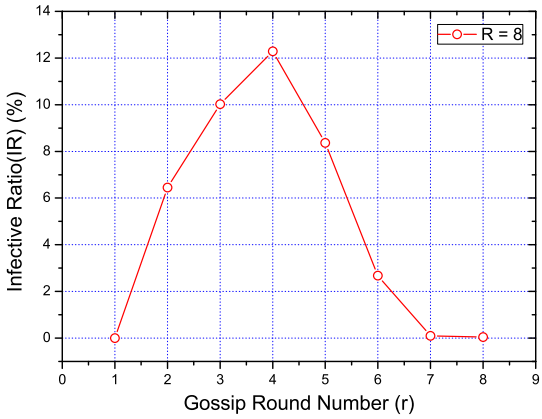

Fig. 4. Infective Ratio when $R = 8$

VI. Conclusions

We presented a real-time, Byzantine-tolerant information dissemination mechanism called RT-LASIRC. With MF attacker detectors, RT-LASIRC can detect Byzantine attackers before information dissemination begins. RT-LASIRC is robust to BHC attacks because of its gossip feature. In addition, RT-LASIRC protects against message-faking attacks during gossip processes. Our experimental studies verify the effectiveness of the RT-LASIRC mechanism.

References

[1] B. Awerbuch, Reza Curtmola, et al., "Mitigating byzantine attacks in ad hoc wireless networks," Tech. Rep. I, JHU CS Dept., March 2004.

[2] Allen Clement Harry C. Li et al., "Bar gossip," in *USENIX OSDI*, November 2006, pp. 191–204.

[3] Y. M. Minsky and F. B. Schneider, "Tolerating malicious gossip," *Distributed Computing*, vol. 16, no. 1, pp. 49–68, February 2003.

[4] CCRP, "Network centric warfare," http://www.dodccrp.org/html2/research_ncw.html, Last accessed, May 2006.

[5] K. Han et al., "Byzantine-tolerant point-to-point information propagation in untrustworthy and unreliable networks," in *NBiS*, March 2007.

[6] B. Pittel, "On spreading a rumor," *SIAM Journal on Applied Mathematics*, vol. 47, no. 1, pp. 213 – 223, February 1987.

[7] S. Verma and W. Ooi, "Controlling gossip protocol infection pattern using adaptive fanout," in *ICDCS*, 2005.