Authenticated Algorithm for Byzantine Agreement

By: D. Dolev and H.R. Strong

Presented by: Gunjan Khanna
Motivation

- Presentation on Monday talked about Byzantine agreement in distributed systems
- Byzantine agreement modeling is important to model insider threats
- Byzantine Agreement defined can be achieved in $(n-1)$ phases.
- No. of processors required to tolerate $f$ failures are $3f + 1$
Byzantine Agreement is *Expensive*

- Order of Messages are $O(n^f)$ with $n$ nodes and $f$ failures
- $n-1$ rounds leads to a delay in agreement
- Authentication does limit the total number of processors to $f + 2$
- Total number of messages also depend upon the connectivity of graph
System Model

- Byzantine behavior for failed nodes.
  - Interactive Consistency or Unanimity
- Byzantine Agreement
  - (I) All correct processors agree on the same value
  - (II) If sender is correct, all processors agree on its value.
- Aim to prove the lower bound of $f+1$ rounds for any type of messages.
Assume a completely connected Graph
- Reliable Communication Channel

In case of an authentication algorithm
- No one can forge the signatures
- Transmitter appends the signature to the message

Lynch and Fisher provided a $t+1$ bound on the number of rounds but without any authenticated messages.
Terms used in Paper

- History
  - N processor history is a finite sequence of $n$ node phases and a phase 0
  - A phase is a directed graph with nodes corresponding to processors and labels on edges.
  - Label is the information sent by the processor

![Diagram of a directed graph with nodes labeled p, q, r, and s and directed edges connecting them]
Formal Terms

- Subhistory
  - History seen by a processor $p$
  - Only incoming edges present

- Agreement Algorithm

- Class of Histories

- Correctness Rule
Byzantine Agreement

- A processor is correct at phase $k$ according to the *correctness rule* it is correct at each of the previous phases $k-1$
  - *Correctness rule* operates on the subhistory of each process

- Byzantine Agreement can be achieved
  - (I) If for $p$ and $q$ correct processes $F_p H = F_q H$.
  - (II) If sender is correct and sends a value $v$ then for a correct process $p F_p H = v$. 
Theorem 1

- Byzantine agreement with authentication can be achieved for \( n \) processors with at most \( t \) faults in \( t+1 \) phases, \( n > t+1 \)

Proof:
- Each node signs the value at phase \( i \) and sends it to only the processes who have not signed it yet.
- \( F \) operates on the messages and throws all the incorrect messages.
- After \( t \) phases each message has \( t \) signatures on it.
- Hence each correct value has been seen by all the correct processes after \( t+1 \) phases.
Theorem 2

- Byzantine agreement cannot be achieved in $t$ or lower phases with $t$ failures.

Proof:

- Let’s assume that Byzantine agreement can be achieved in $t$ or fewer phases.
- Let $R$ be the correctness rule and $F$ be the decision function. Together they achieve the Byzantine agreement.
- Critical Sequence which contains the incorrect processes.
- Faulty Processes increase serially
Proof Cnt.d

- If $FpH'=FpH$ implies equivalent histories, $H$ and $H'$
- A node is hidden at phase $k$ if there are no outedges from it in any later phases.
- Show by induction that “If a node $r$ representing a processor at phase $k$ of history $H$ in $C$ then,”
  - There is a history equivalent to $H$ denoted by $H'$ through phase $k$ except for outedges of $r$ with $r$ correct and all other processes correct.
  - If all other processes are correct then there exist $H'$ with outedges of $r$ replaced, $r$ hidden and all other processes are correct.

- Case 1 $k=t$
Proof Cnt.d

(a) If $r$ is incorrect at phase $k$ then we correct the outedges of $r$ one at a time.
- For each individual change there is a processor which sees the same history as before
- Final $H'$ will have $r$ correct and all processes trivially correct.

(b) If $r$ is a process and all other processes are correct then removing outedges of $r$ will preserve the equivalence relation in $H$. 
Proof Cnt.d

- Assume (a) and (b) to be true for all phases after $k$
  - (i) Let $r$ be the incorrect edge at phase $k$.
    - Correct all nodes at phase $k$ hyp. (a)
    - While the incorrect outedges remain
      - Replace the position $k+1$ in $C$ with $s$ which is the target of the incorrect outedge.
      - Hide $s$ at $k+1$
      - Correct all nodes hyp. (b)
  - The final result $H'$ will have $r$ and all correct processes.
Proof Cnt.d

(ii) All correct nodes at phase $k$ and $r$ be a node

- Correct nodes at phase $k+1$ hyp. (a)
- Replace $k^{th}$ position with label of $r$.
- While the outedges of $r$ remain
  - Replace the position $k+1$ in $C$ with $s$ which is the target of the outedge $e$.
  - Hide $s$ at $k+1$ hyb. (b)
  - remove $e$
  - Correct all processes after phase $k$ hyp. (a)
- Hide $r$ at phase $k +1$. Hence the final result will have $r$ hidden at phase $k$ and all other processors correct after phase $k$. 
Algorithms using Authentication

- A label is a sequence of authentication or just a single authentication.
  - (Label a)p
- No process can alter the authentication nor can it pretend to have received an authentication.
- Main idea is to restrict the no. of messages.
Theorem 3

- Byzantine agreement can be reached with t failures within t+1 rounds with $O(e)=O(n^2)$ messages
  - Restrict the number of messages from each processor to 2.
  - Each processor orders the messages lexicographically in the order $(..(v)p_1)p_2...p_n)$
  - It relays the first two messages with distinct value
    - Only one message is relayed if values are the same
    - If two messages are relayed within t+1 phases then it decides “sender fault”
  - If no other process imports any other value then v is extracted.
Theorem 4

- Byzantine agreement can be reached for \( n \) processes using \( t+2 \) rounds and \( O(nt) \) messages when there are \( t \) failures to tolerate.
  - Chose \( t+1 \) processes to be relay processes.
  - Non relay processes send messages only to the relay processes
  - Messages = \( O(nt) \)
  - Why are \( t+2 \) phases required?
Theorem 5

- Assumptions
  - $t+1$ connected graph
  - $K$-Diameter of a graph is the least upper bound of the lengths of the $k$ disjoint paths

- If $d$ is the diameter of a $t+1$ connected graph then Byzantine agreement can be reached within $t+d$ phases.

- Proof: Trivially True from Theorem 1.
Theorem 6

Byzantine agreement can be reached in $t+1$ phases with $O(nt)$ messages in a complete network.

Proof

If $n < (2t+1)$ then it is true from Theorem 3.

Use only $2t+1$ processes as active

Correctness Rule is same as Theorem 3.

(Lexicographically correct)
Algorithm Cnt.d

- Passive processes do not send any message.
- They collect the signed messages and extract values with at least $t+1$ distinct signatures.
- If any correct active process extracts a value, then some correct active process extracts that value by phase $t$ and then the message it will relay will have $t+1$ active signatures.
- It can be extended for the case with $t-1$ phases.
Contributions

- Proposes several efficient algorithms for reaching Byzantine agreement
  - Reduces the number of message requirement
  - Gives an important lower bound on the number of phases required
- Gives bounds on an authenticated algorithm for a $t+1$ connected graph
Conclusions

- These results still hold
- One should look in terms of Probabilistic algorithms.
  - With some probability you mark a process faulty
- Lower bound on messages is not derived