Lab 4

Byzantine Agreement

Design

- Assumed, first n servers (n = number of honest servers) are honest to maintain simplicity
- Remaining servers are Byzantines
- Maintained 3k+1 (where k = number of byzantine)

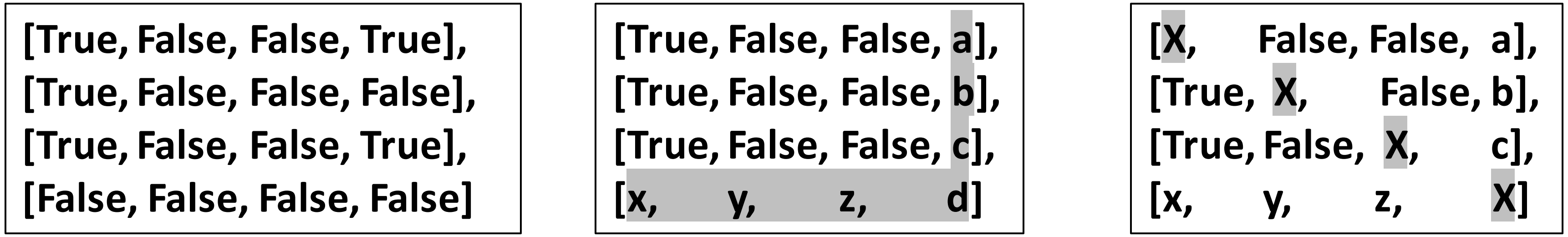
Design(2)

- Honest generals voted first
- Byzantine waited until all honest vote received. Then, **round1** votes delivered to honest servers
- After receiving all votes from others, honest servers started sending **round2** vectors to all other servers.
- After receiving all vectors from honest servers, Byzantine started sending **round2** vectors to honest servers
- Now all honest servers have all **vectors(2D array)** from other servers

Task 1 – 4 servers (N=4, k=1)

- 3 honest nodes and 1 Byzantine (N=4, k=1)
- To determine Byzantine, verified the first n (number of honest) values in each column are same or not. In this case column 4 does not match, so last server is Byzantine
- Changed The Byzantine values with random letters
- Crossed the diagonal values
- If any value has a majority, that value is put into the **result vector**
- If no value has a majority, the corresponding element of the **result vector** is marked UNKNOWN
- Final Attack : Calculated from max of **result vector**

Task 1 – Example(3)



```
[True, False, False, True],
[True, False, False, False],
[True, False, False, True],
[False, False, False, False]
```

```
[True, False, False, a],
[True, False, False, b],
[True, False, False, c],
[x,    y,     z,     d]
```

```
[X,    False, False, a],
[True, X,     False, b],
[True, False, X,     c],
[x,    y,     z,     X]
```

Result Vector = **[True, False, False, 'UNKNOWN']**

Attack = **False**

Task 1 – 4 servers (N=4, k=1)

- So above example demonstrate that agreement is reached.
- Byzantine general must respect the agreement protocol. If not, they can send the wrong SERVER_ID.

Task 2: 3 servers (N=3, k=1)

```
[True, False, True],
[True, False, False],
[False, False, False]
```

```
[True, False, 'n']
[True, False, 'x']
['l', 'n', 'j']
```

```
['X', False, 'n']
[True, 'X', 'x']
['l', 'n', 'X']
```

Click to addResult Vector = **['UNKNOWN', UNKNOWN , 'UNKNOWN']**

Attack = **UNKNOWN**

For 3 servers Byzantine can change the result. That's why there is a 3k+1 rule.

In their paper, Lamport et al. (1982) proved that in a system with k faulty processes, an agreement can be achieved only if 2k+1 correctly functioning processes are present, for a total of 3k+1.

Task 3 – A general solution?

- To handle more than one Byzantine general, We considered first n servers are honest.

- Coordination between the Byzantine generals can break the coordination algorithm. We can verify it by example:

    - Suppose we have total 7 servers (5 honest and 2 byzantine)
    - Honest servers sent **Attack, Attack, Attack, Retreat, Retreat**
    - Byzantine coordinated each other and both decided to send **Retreat**
    - So, result will change completely