WWW Robustness Quiz

Answer the questions by filling in the blanks in reference to the internet:

The internet is a scale-free network. We can infer that it has a

high degree of tolerance towards random failures and a

low degree of tolerance against attacks.

The most successful attacks target the nodes that are the

most connected.

Node Connectedness Quiz

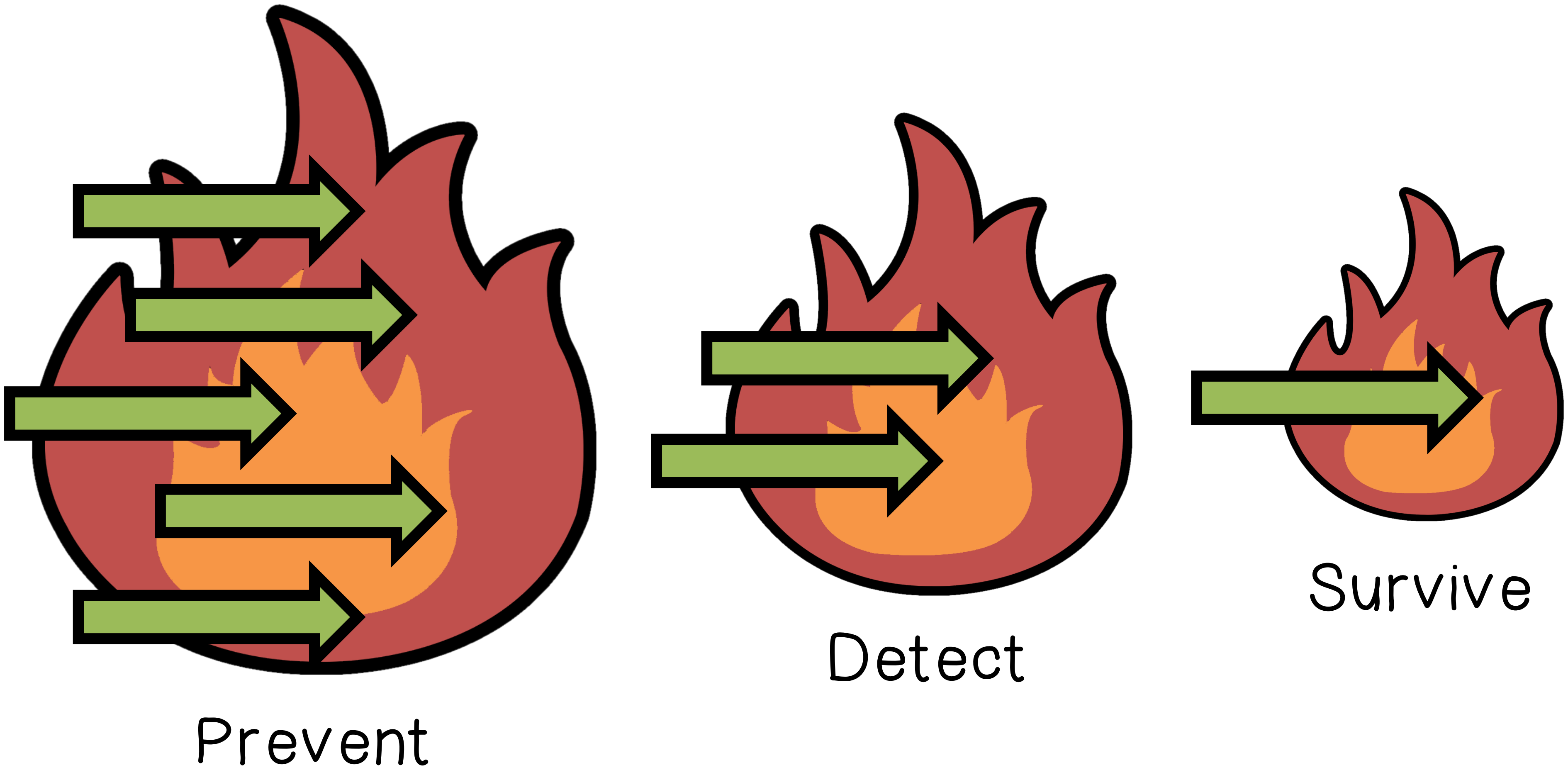
Match the method of determining node connectedness to its definition. Answer choices are: Temporal Closeness, Average Node Degree, Node Persistence

Average Node Degree : nodes with the largest number of nodes connected to them

Node Persistence : during a snapshots of internet traffic, these nodes nodes are the ones most likely to appear.

Temporal Closeness : nodes that interact with the largest number of nodes

Defense in Depth



Defense in Depth

What should tolerate attacks?

Our security-critical assets:

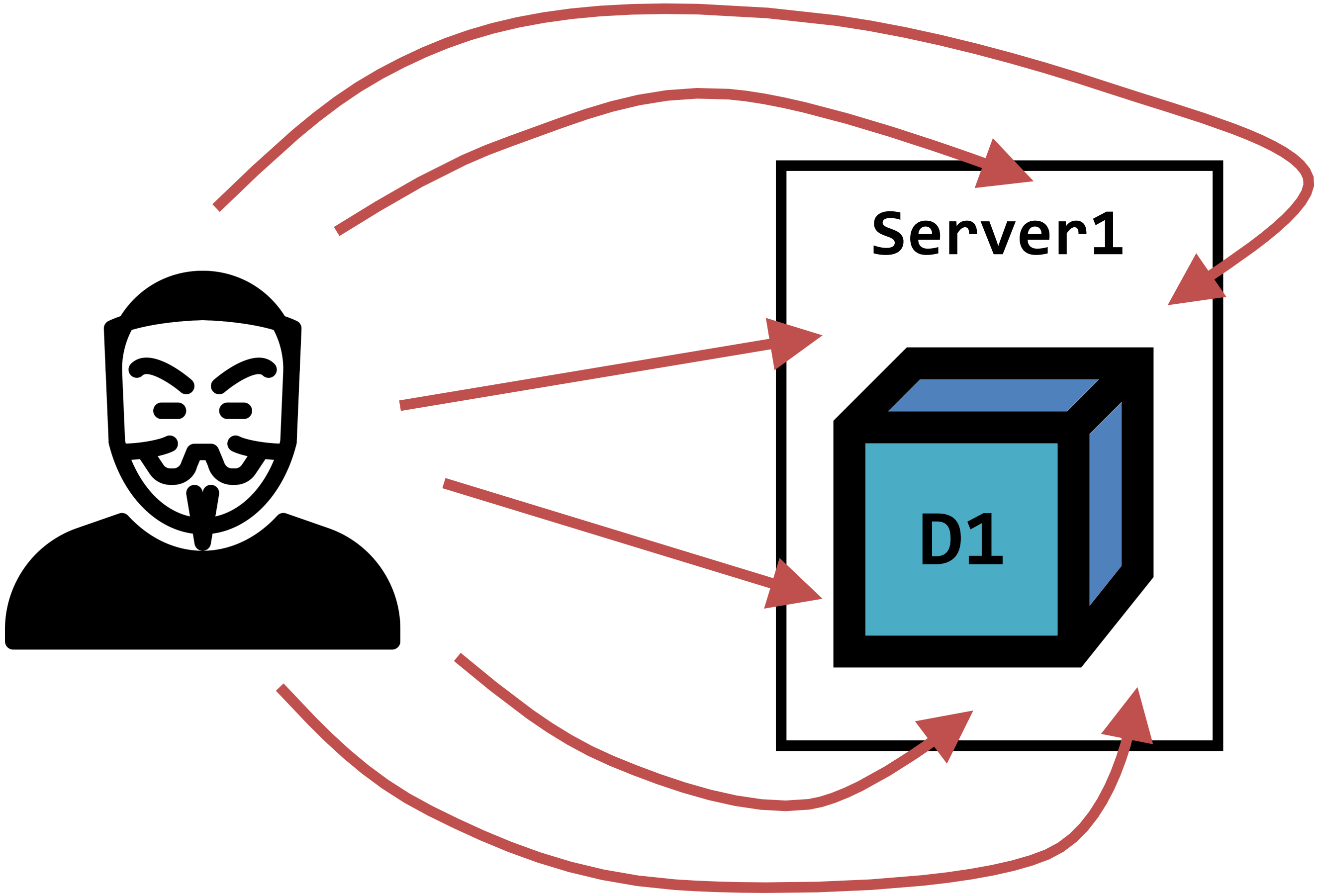
- Data (confidentiality, integrity, and availability)
- System services (availability and integrity)

Defense in Depth

Typically, data is stored in *one* place, e.g., a dedicated computer/server

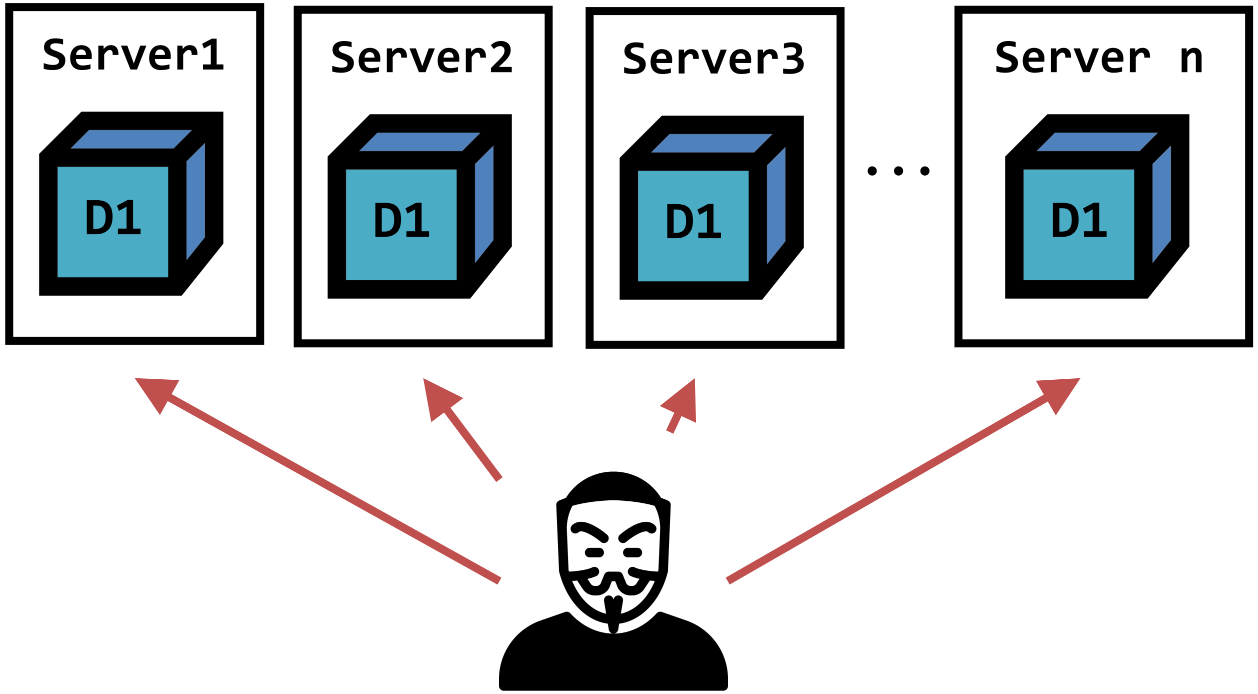
Compromise of the server leads to loss of confidentiality, integrity, and availability

Defense in Depth



Defense in Depth

What if we *replicate* the data and store the copies in multiple, say *n*, servers?

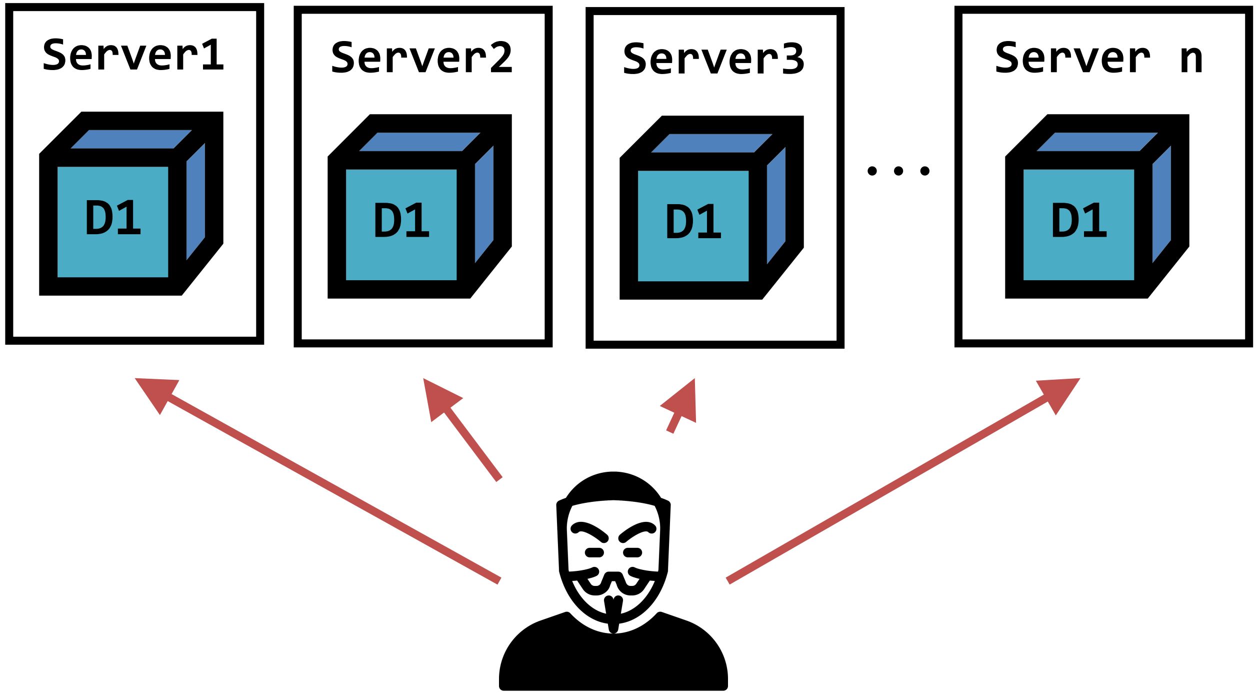


Confidentiality protection is weaker:

- Attacker has more chances: *n* targets vs. 1

Defense in Depth

What if we *replicate* the data and store the copies in multiple, say *n*, servers?



Integrity and availability are better protected:

- Attacker needs to compromise majority of the *n* servers

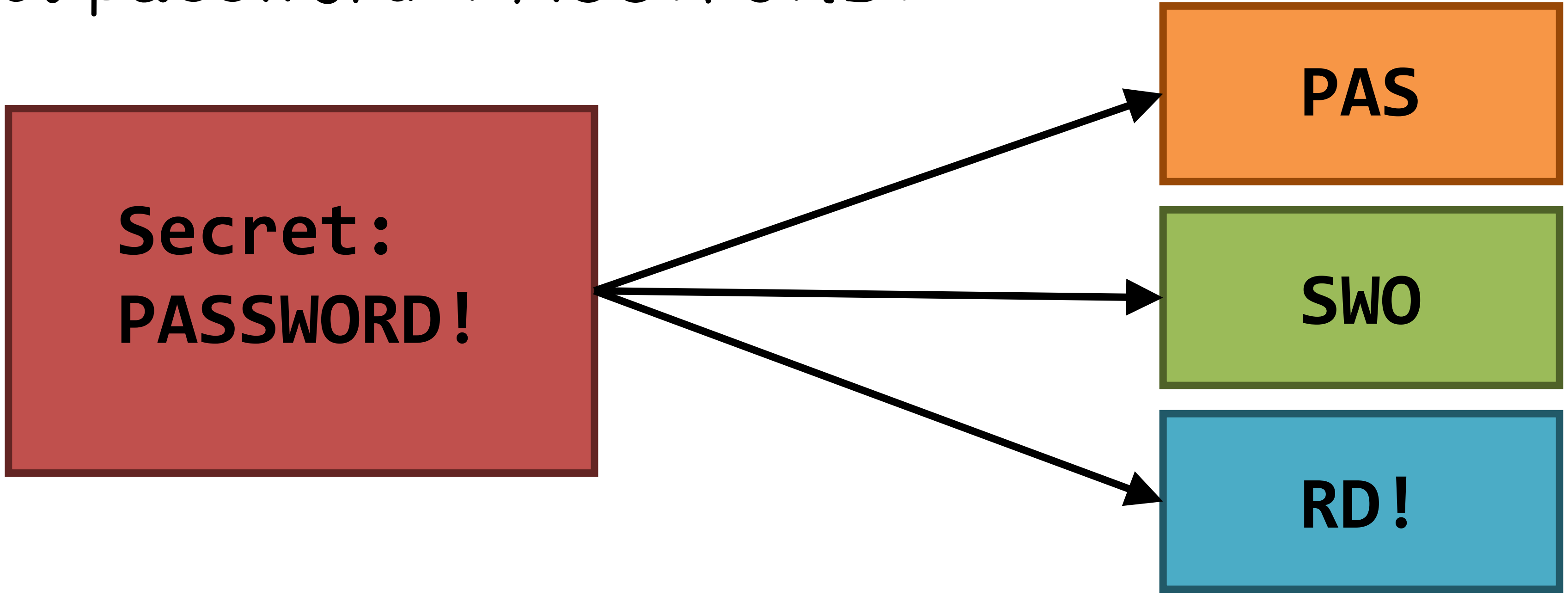
Defense in Depth

Can we have better confidentiality protection?

Naïve Secret Sharing Quiz

Cryptographic secret sharing involves giving each party a share of a secret.

Secret password: PASSWORD!



Naïve Secret Sharing Quiz

What is a major weakness of the Naive Secret Sharing scheme?

The major weakness of naive secret sharing is the more shares you have of the secret, the less work you have to do to guess the secret.

Hint: If a person knows a password is 9 characters long, how many possible combinations would have to be checked?

What if the person doing the guessing knew one share of the secret?

Secret Sharing

In cryptography, secret sharing refers to a method for distributing a *secret* amongst a group of participants, each of which is allocated a *share* of the secret

The secret can only be reconstructed when the shares are combined together; individual shares are of no use on their own

Secret Sharing

Benefits:

Gives tight control and removes single point vulnerability with regard to confidentiality

- Individual key shareholders (or a compromised party) cannot change/access the data

Each key share holder can be a *group* where each member stores a replica of the share

- Improve integrity and availability

Mathematical Definition

Goal is to divide some data *D* into n pieces $D_1, D_2, \ldots, D_n$ in such a way that:

- Knowledge of any k or more D pieces makes D easily computable
- Knowledge of any k -1 or fewer pieces leaves D completely undetermined (in the sense that all its possible values are equally likely)

Mathematical Definition

This scheme is called (k, n) threshold scheme

If k=n then all participants are required together to reconstruct the secret

Shamir's Secret Sharing

Invented by Adi Shamir in 1979

Suppose we want to use (k, n) threshold scheme to share our secret S where $k < n$

Shamir's Secret Sharing

Choose at random (k-1) coefficients a_1, a_2, a_3, ..., a_{k-1}, and let S be the a_0, that is, pick a random k-1 degree polynomial:

$$q(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{k-1} x^{k-1}$$

Shamir's Secret Sharing

Construct n points, or, (i, S_i=q(i)), where i=1, 2, ..., n

- These S_i's are called the *shares* of S
- All arithmetic done modulo a prime number p that is greater than both S and n
- Coefficients a_1, a_2, ..., and a_{k-1} of q(x) are randomly chosen from a uniform distribution over the integers in [0, p)

Shamir's Secret Sharing

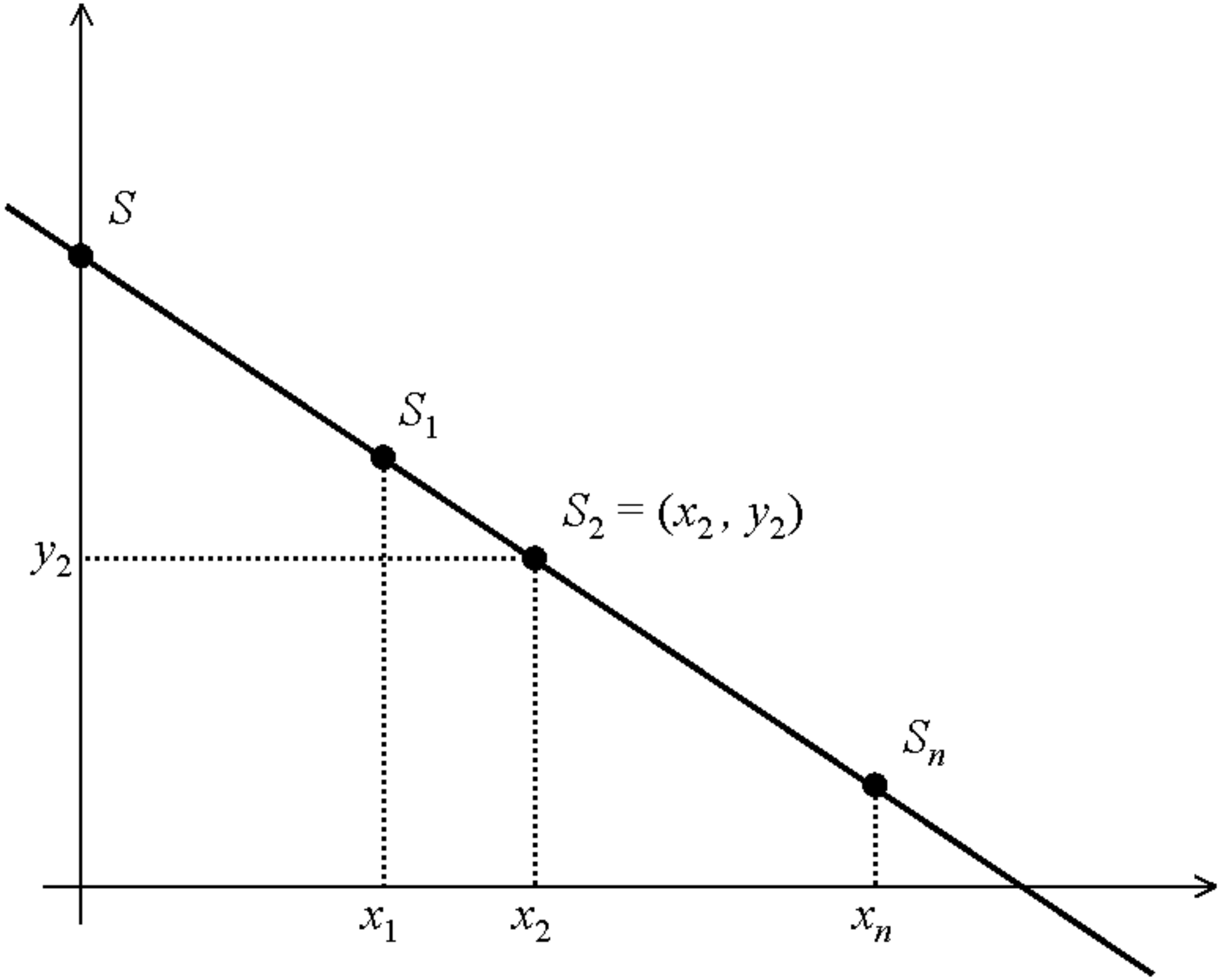
Given any subset of k of these pairs, or, points, (i, q(i)), we can find the coefficients of the polynomial q(x) by interpolation

- k points uniquely determining a polynomial of degree k-1
- Once q(x) is determined, then evaluate $S=q(0)=a_0$, which is the secret
  - That is, given any k shares, we can reconstruct the secret

Shamir’s Scheme Example

Example 1: k = 2

even with a single share (point), the secret can still be any value in with equal probability



Shamir's Scheme Example

Determining $q(x)$

Given (x_i, y_i) where $y_i = q(x_i)$, use Lagrange Interpolation to compute $q(x)$

Shamir's Scheme Example 2

(3,5) Threshold Scheme

$$
\begin{aligned}
n &= 5 \\
k &= 3 \\
S &= 7 \\
a_0 &= S \\
a_1 &= 3 \\
a_2 &= 5 \\
p &= 11
\end{aligned}
$$

$$q(x) = 5x^2 + 3x + 7 \pmod{11}$$

$$
\begin{aligned}
S_1 &= q(1) = 5(1)^2 + 3(1) + 7 \pmod{11} \equiv 4 \\
S_2 &= q(2) = 5(2)^2 + 3(2) + 7 \pmod{11} \equiv 0 \\
S_3 &= q(3) = 5(3)^2 + 3(3) + 7 \pmod{11} \equiv 6 \\
S_4 &= q(4) = 5(4)^2 + 3(4) + 7 \pmod{11} \equiv 2 \\
S_5 &= q(5) = 5(5)^2 + 3(5) + 7 \pmod{11} \equiv 4
\end{aligned}
$$

Shamir's Scheme Example 2

(3,5) Threshold Scheme

Suppose people with shares $S_1 = 4$, $S_2 = 0$, $S_5 = 4$, decide to reconstruct the secret:

$$q(x) = \sum_{i=1}^{k} y_i \prod_{j=1, j \neq i}^{k} \frac{x - x_j}{x_i - x_j}$$

Shamir's Scheme Example 2

(3,5) Threshold Scheme

Suppose people with shares $S_1 = 4$, $S_2 = 0$, $S_5 = 4$, decide to reconstruct the secret:

$$q(x) = [4\frac{(x-2)(x-5)}{(1-2)(1-5)} + 0\frac{(x-1)(x-5)}{(2-1)(2-5)} + 4\frac{(x-1)(x-2)}{(5-1)(5-2)}] \pmod{11}$$

$$q(x) = [(x-2)(x-5) + 4(x-1)(x-2)] \pmod{11} = 5x^2 + 3x + 7 \pmod{11}$$

$$S = q(0) = 7$$

Shamir's Scheme Summation

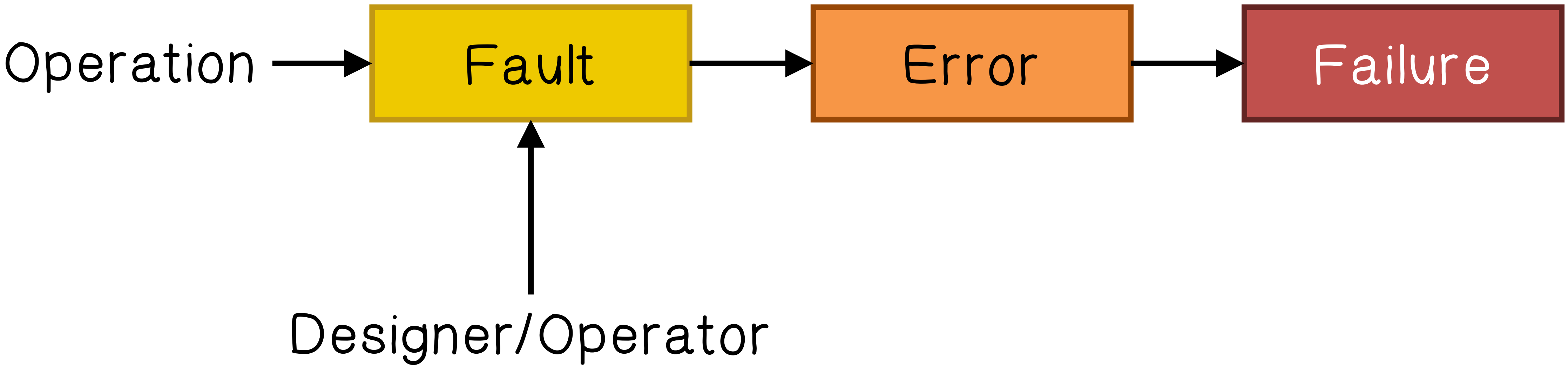
Using $n = 2k - 1$ requires adversaries to acquire more than $\left\lfloor \frac{n}{2} \right\rfloor = k - 1$ shares

Shamir's Scheme Summation

- Add or delete shares without affecting others
- Easy to create new shares without changing secret
- Easy to create hierarchical schemes
- Information theoretic security

Practical Byzantine Fault Tolerance

A fault is the cause of an error that leads to a system failure:

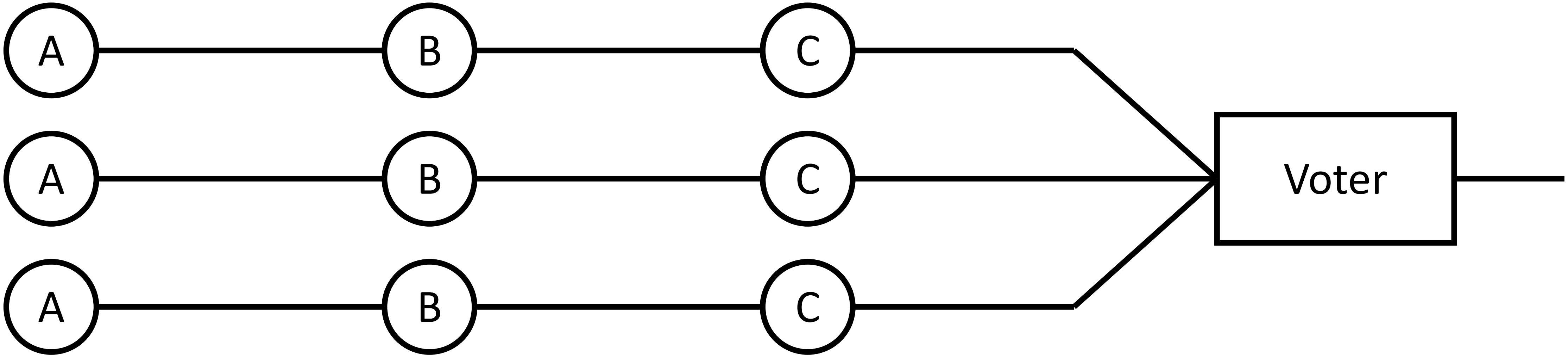


Practical Byzantine Fault Tolerance

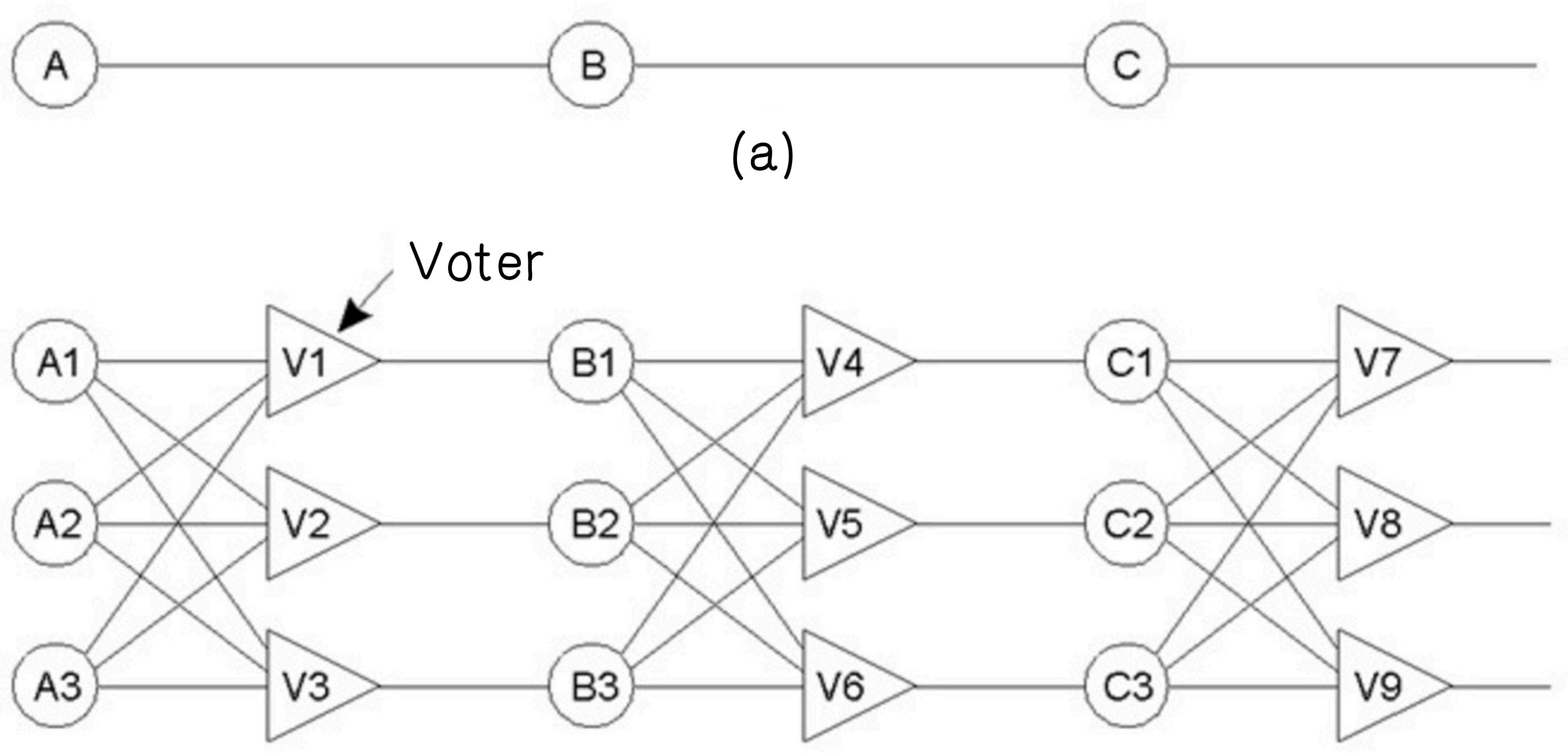
Fault tolerance can be achieved through failure masking

- By redundancy

Practical Byzantine Fault Tolerance



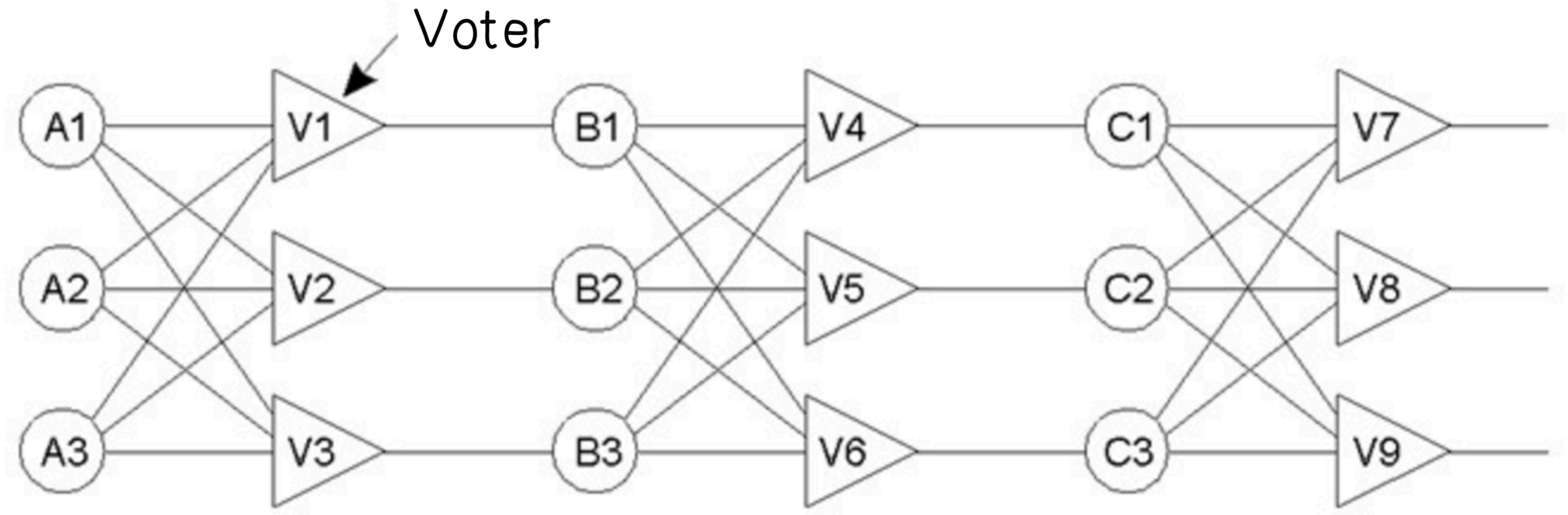
Practical Byzantine Fault Tolerance



(a)

(b)

Redundancy of System Services



- A group of services, each being identical
- A message sent to the group is delivered to all the copies/members of the group
- If some of the processes fail, it is assumed that at least one of the others will still be able to function correctly.

Redundancy of System Services

The redundant system needs to have:

- A set of non faulty services reaching a consensus, even in the presence of some corrupted or faulty ones
- This consensus is the correct service that the system will provide per a request

Redundancy Quiz

Fill in the blanks using the following answers: Maintainability, Safety, Availability, Reliability

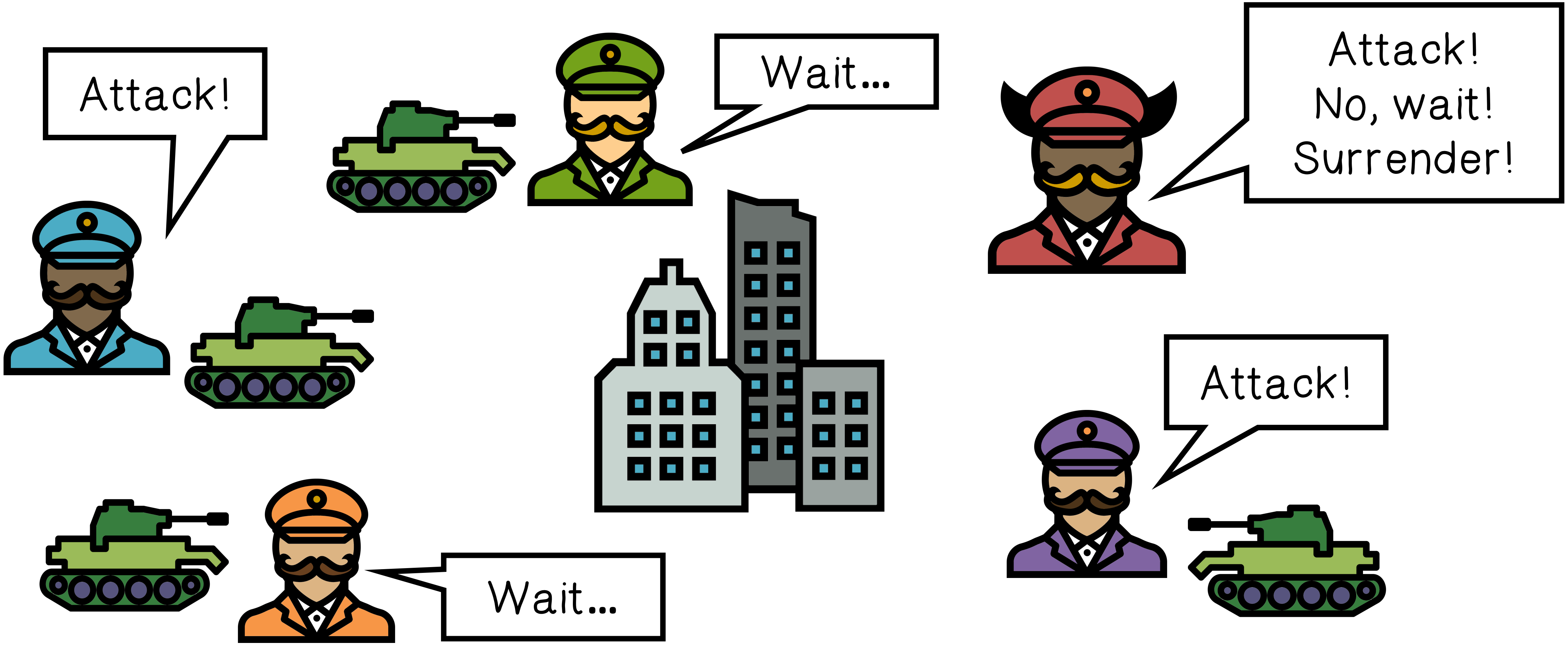
Availability : probability the system operates correctly at any given moment

Reliability : ability to run correctly for a long interval of time

Safety : failure to operate correctly does not lead to catastrophic events

Maintainability : ability to easily repair a failed system

Byzantine Generals Problem



Byzantine Generals Problem

A commanding general must send an order to his n-1 lieutenant generals such that:

- All loyal lieutenants obey the same order
- If the commanding general is loyal, then every loyal lieutenant obeys the order he sends

System Models

Assumptions:

Asynchronous distributed system where nodes are connected by a network

- The network may fail to deliver messages, delay, duplicate or deliver them out of order

Byzantine failure model

- Faulty nodes may behave arbitrarily
- Independent node failures

The adversary cannot delay correct nodes indefinitely and cannot subvert the cryptographic techniques

System Properties

3f+1 is minimum number of replicas that allow an asynchronous system to provide safety and liveness

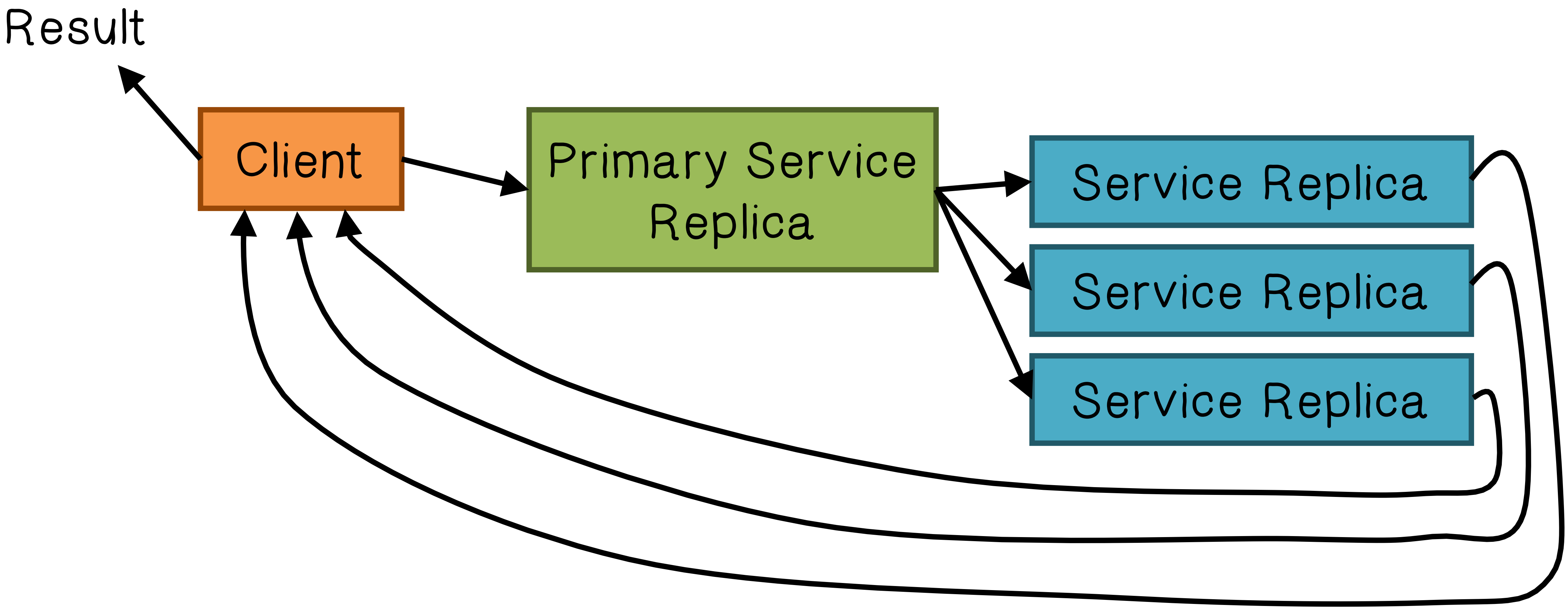
- where f is number of faulty replicas

System Properties

n= 3f+1 replicas are needed because it must be possible to proceed after communicating with n-f replicas since f replicas might be faulty and not responding

- But the f replicas that did not respond may be non-faulty and therefore f of those responded may be faulty
- n-2f > f therefore n > 3f

System Algorithm



Attack Tolerance

We cannot apply *fault* tolerance techniques directly to achieve *attack* tolerance

Redundancy in fault tolerance means using *replicas* of a system

- All replicas run the same program(s)
  - The single/same attack compromises *all* replicas

Attack Tolerance Through Diversification

Each instance has a different implementation

- Across all layers of the stack:
  - Different network and application protocols, programming languages, operating systems, etc.

Each uses different security protection mechanisms, or on different part of the program

- Efficiency – reduce security overhead
- Help identify attacks

Attack Tolerance Through Moving Target

Take the diversification concept one step further...

- Dynamically change network and system configuration

Many instances of the systems and network services

- Implemented differently
- Composed on-the-fly