Introduction to Information Security

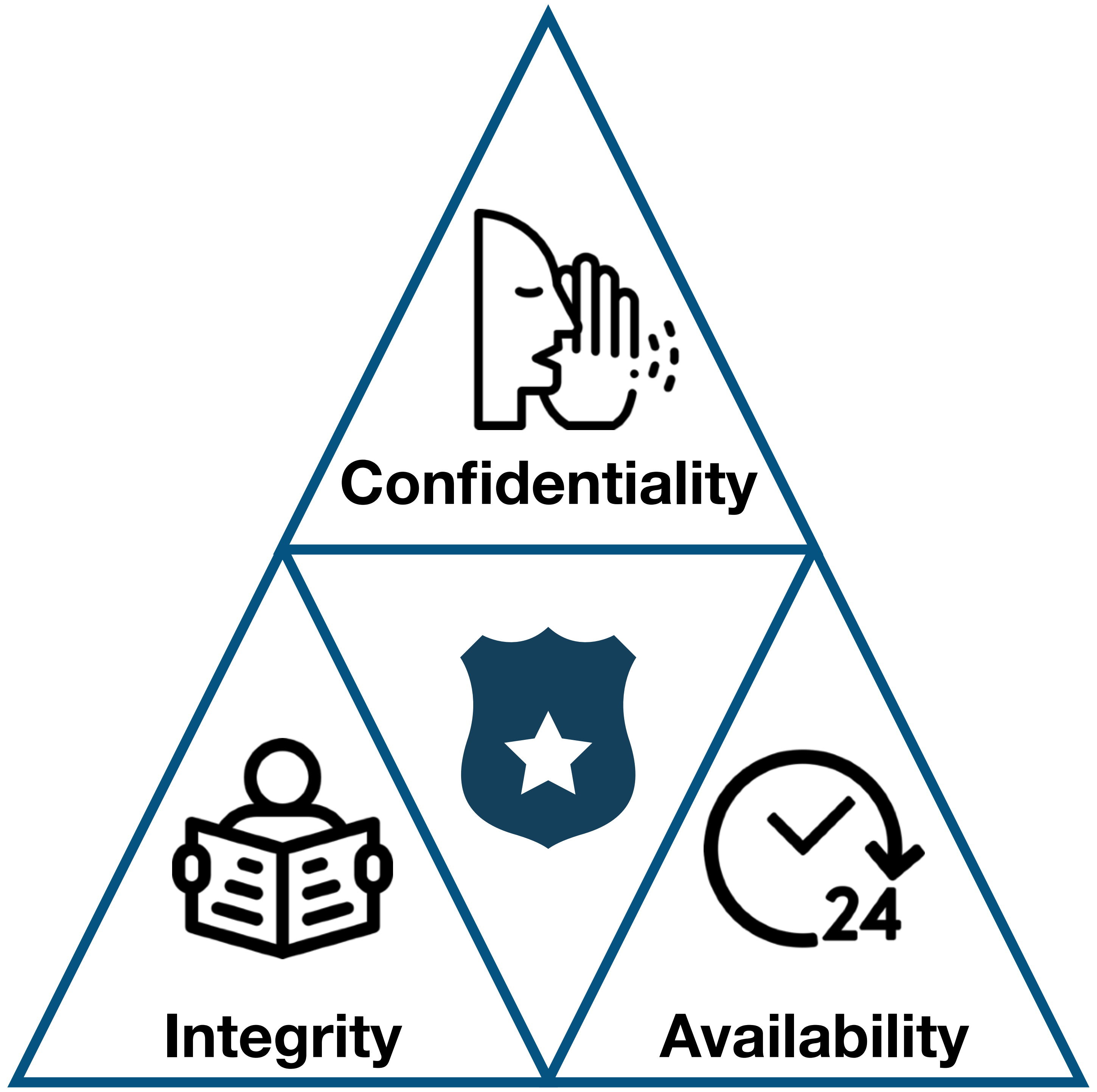
6. Availability

Kihong Heo

KAIST

Recall: The CIA Triad

- Three most important properties of computer security
- CIA: Confidentiality, Integrity, and Availability
- Example: a bank system
  - Confidentiality?
  - Integrity?
  - Availability?

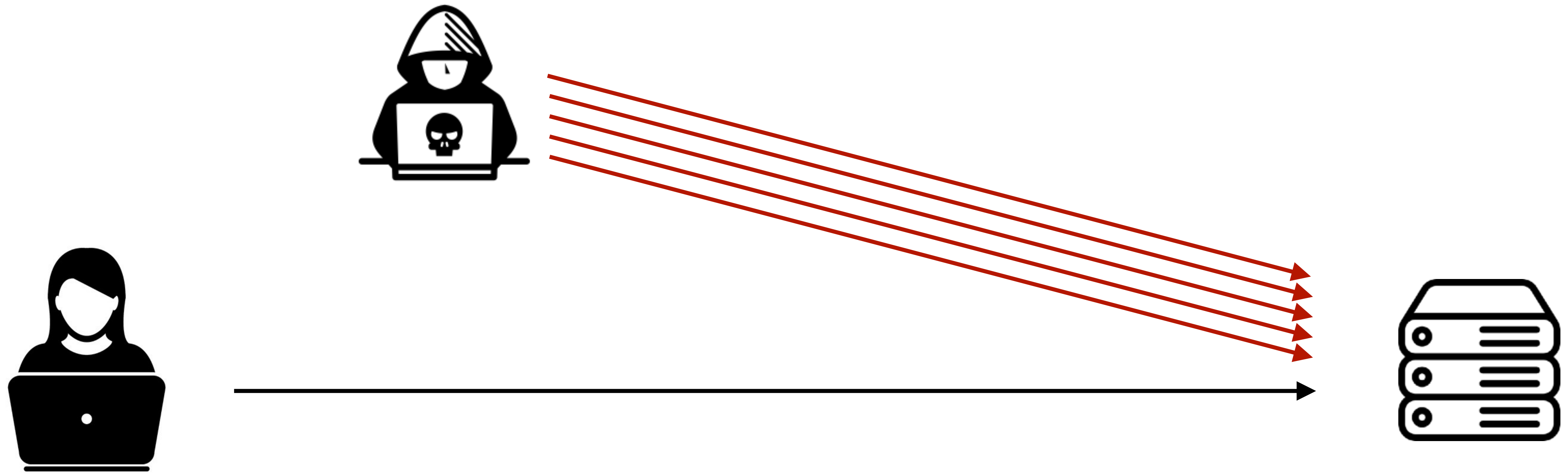


Denial-of-service (DoS) Attacks

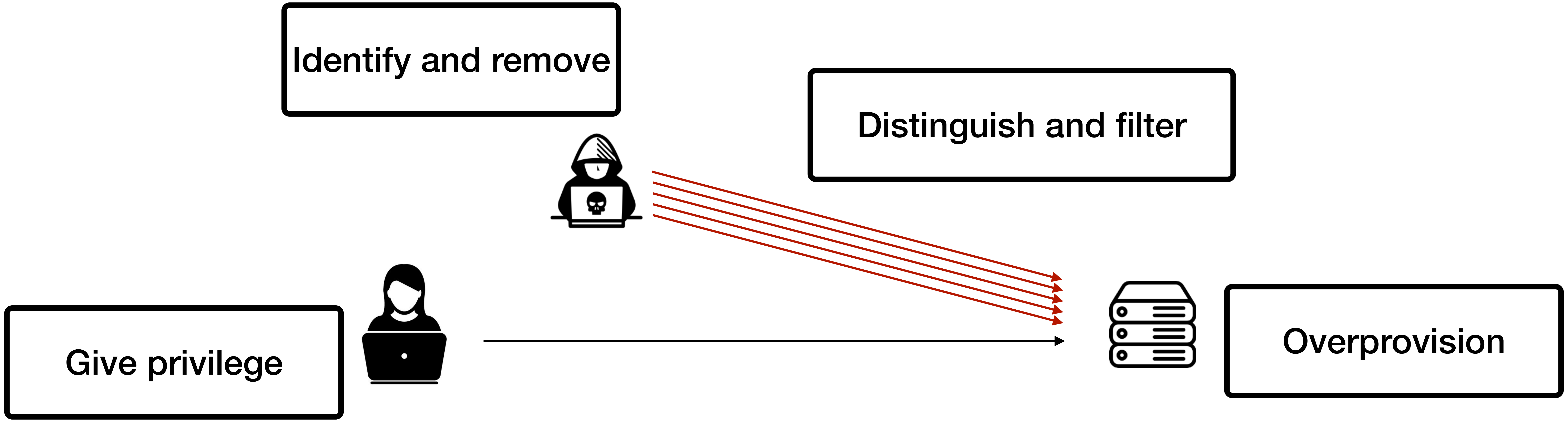
*"A group of **authorized users** of a specified service is said to **deny service to another group of authorized users** if the former group makes the **specified service unavailable** to the latter group for a period of time which exceeds the intended (and advertised) service maximum-waiting time"*

- V. Gligor, A Note on the Denial-of-Service Problem, IEEE Security & Privacy, 1983

- Not considered as a security problem until the late 80s
- Distributed denial-of-service (DDoS): DoS attacks by a large number of devices
- Solutions to DoS attacks in the Internet?



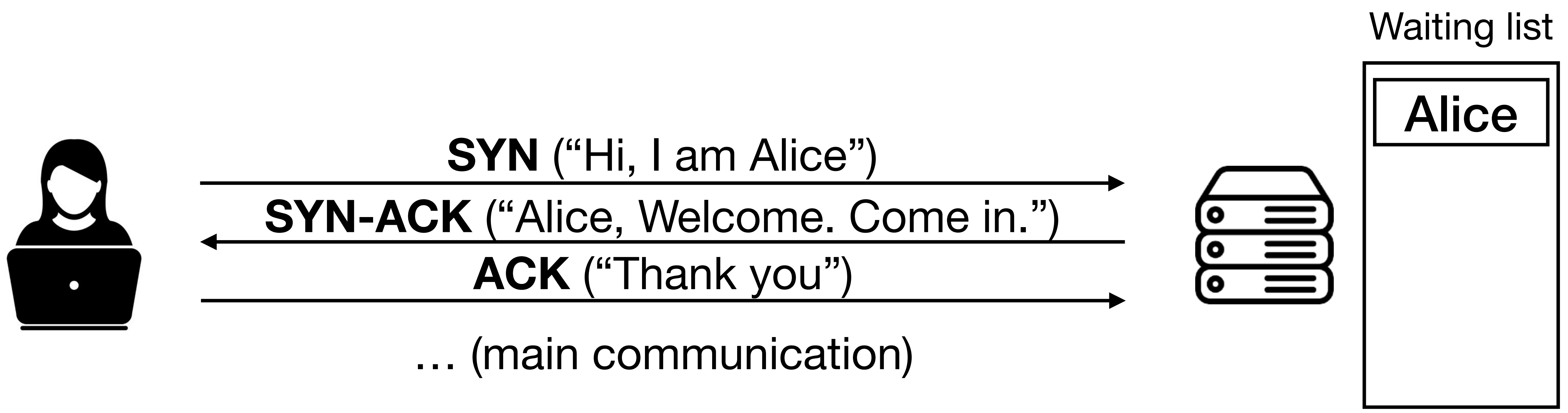
Solutions to DoS Attacks



- Unfortunately, there is no bulletproof scheme for availability!
  - Unlike other properties: encryption for confidentiality and MAC for integrity
- Theoretical solution: user agreement [Yu and Gligore 1988]
  - External constraints on service invocations that must be obeyed by all users
  - Hard to achieve in practice

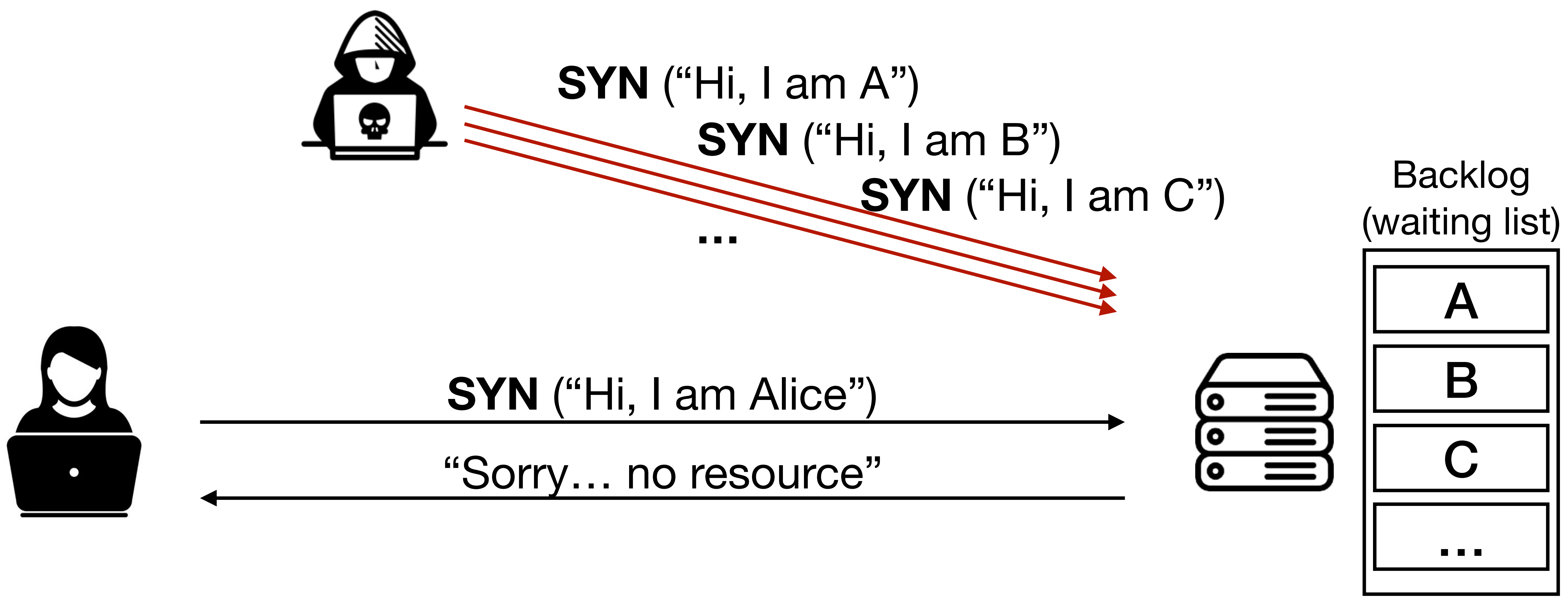
Example: SYN Flooding (1)

- Three-way handshake: establish a connection between the server and the client
  - Used in TCP/IP networks (e.g., the Internet)
- Servers store “half-open” connections while awaiting the third handshake message
  - To ensure the identity of the client



Example: SYN Flooding (2)

- Attacker floods the server with SYN packets but no follow-up ACK
- Server exhausts resources → DoS

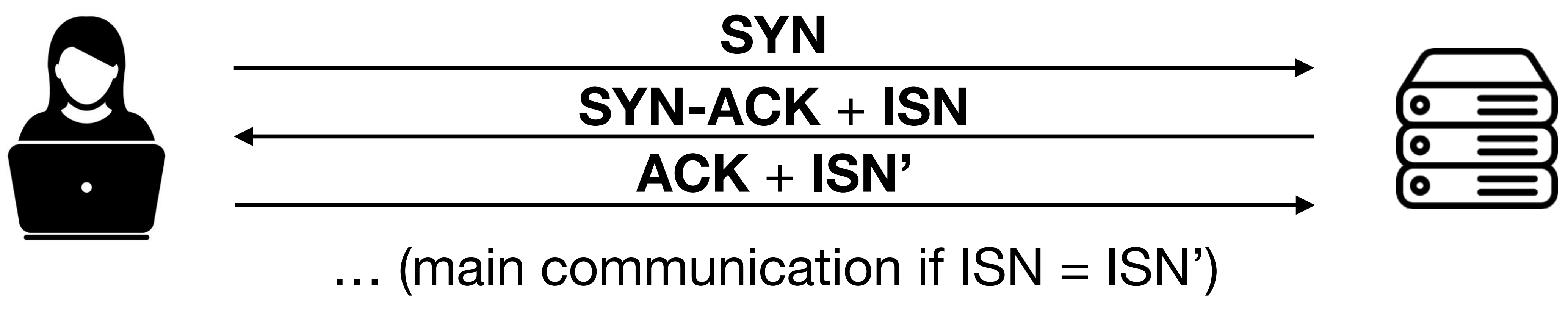


Analysis

- Why does the server exhaust resources?
    - Need to store requests for 511 seconds
    - A finite-size queue for incomplete connections (usually 1024 entries)
- Why not store all requests (736 bytes/entry)?
    - Arms race! Attackers can easily win

SYN Cookie (1)

- Idea: DO NOT store! DO recompute!
- Server sends SYN-ACK with ISN (initial sequence number) based on the connection state
- Client sends ACK with the ISN
- Server verifies ISN and allocates the connection state if correct
- ISN = $H(S_{ip}, C_{ip}, S_{port}, C_{port})$

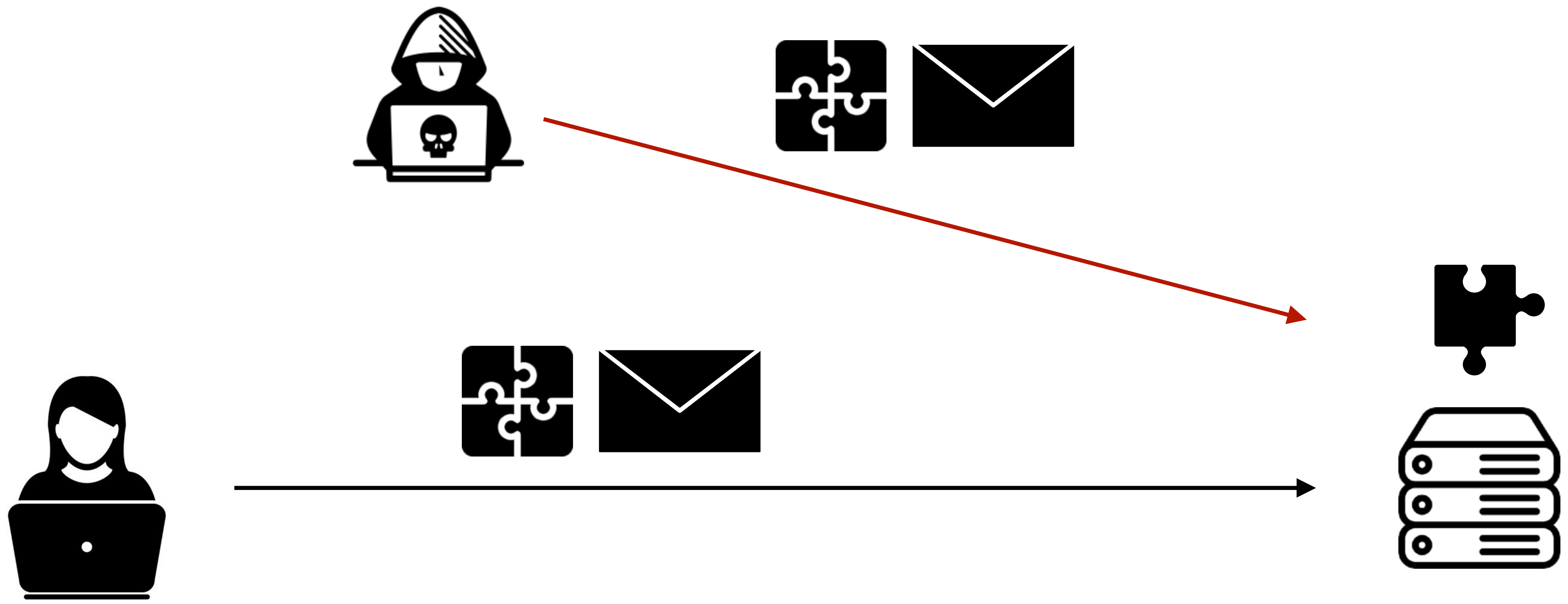


SYN Cookie (2)

- What if attackers send massive valid ISNs?
  - $H(S_{ip}, C_{ip}, S_{port}, C_{port})$: public because H, S_{ip}, C_{ip}, S_{port} and C_{port} are all public
- Idea
  - $H(S_{ip}, C_{ip}, S_{port}, C_{port}, skey)$ where $skey$ is a secret key of server

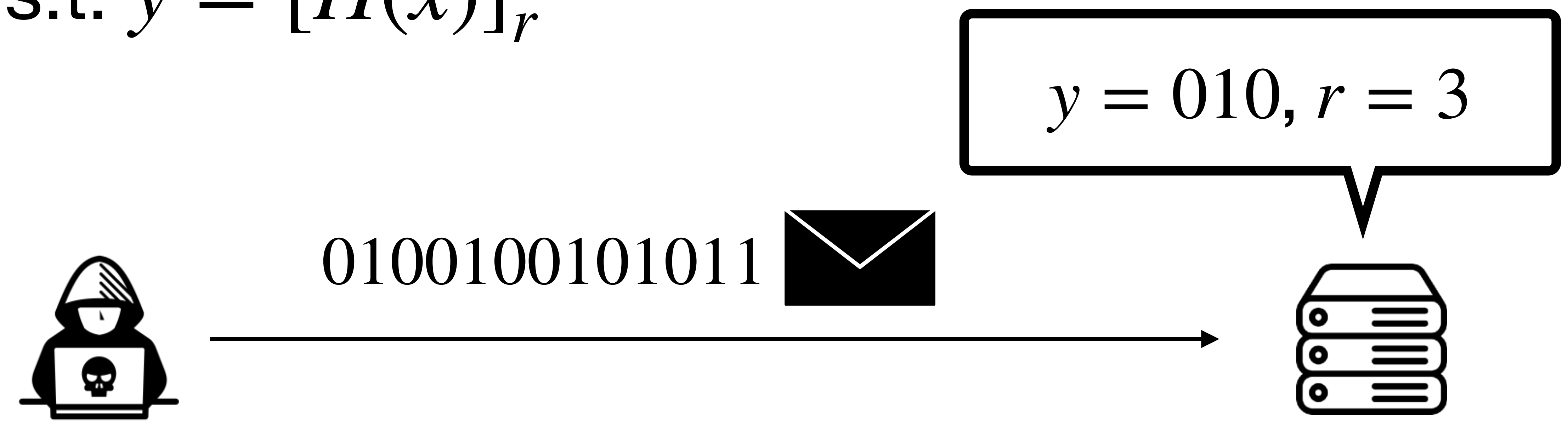
Puzzle-based DDoS Defense

- Idea: slow down attackers with puzzles
- Requirements
  - Making puzzle-solution pairs: easy
  - Sending puzzles to clients: easy
  - Solving puzzle: hard
  - Verifying solutions: easy



Example (1)

- Secure-hash-based proof of work: given y, find x s.t. $y = [H(x)]_r$
  - $[A]_b$: b LSB bits of A
- Effort: ~2^r
- Issues
  - Stateful operation: server has to remember all previous pairs of (user, puzzle)
  - Rainbow table: precomputation



Example (2)

- Idea:
  - Generate puzzles with a secret key k
  - Verify puzzle solutions without keeping any state (stateless)
- Server sends $T = MAC_k(C_{ip}, C_{port})$ and r

Make server stateless

- Client searches for x s.t. $[H(x \,||\, T)]_r = [T]_r$
  - ~2^r operations of H

Make attackers hard to precompute

Summary

- Denial-of-service: a common attack on the availability of systems
- No bulletproof scheme
- Lesson: stateless system design
- Example: SYN cookie, puzzle-based defense