Counter Denial of Service Client-Server Puzzles

Dr. Attila Altay Yavuz

Privacy-Preserving and Trustworthy Cyberinfrastructures (Spring 2026)

Credits: Dr. Peng Ning and Dr. Adrian Perrig

Client Puzzles

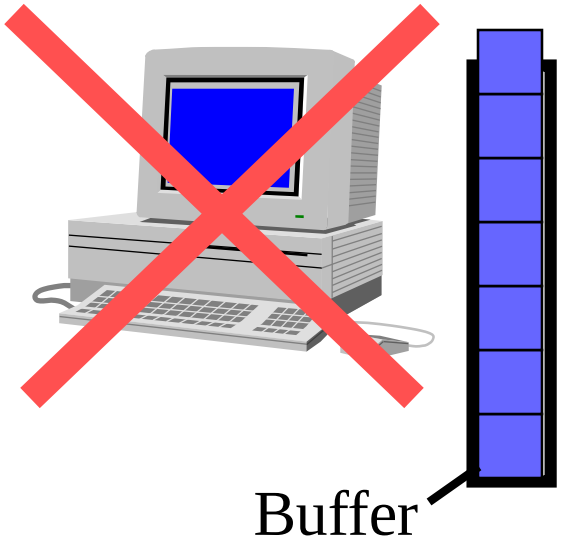
- The problem being addressed
  - **Denial of Service (DoS) attacks**
- Basic constructions to be covered
  - Use a pre-image of crypto hash functions
  - Use special image of crypto hash functions

- In general, authentication and AI tools may help to mitigate DoS attacks. However,
  - Privacy-oriented tools do not mitigate DoS, why?
  - Active research area: *Counter Denial of Service for Next-Generation Networks within the Artificial Intelligence and Post-Quantum Era*
  - https://cse.usf.edu/~attilaayavuz/article/24/Vision_TPS_CounterDoS.pdf

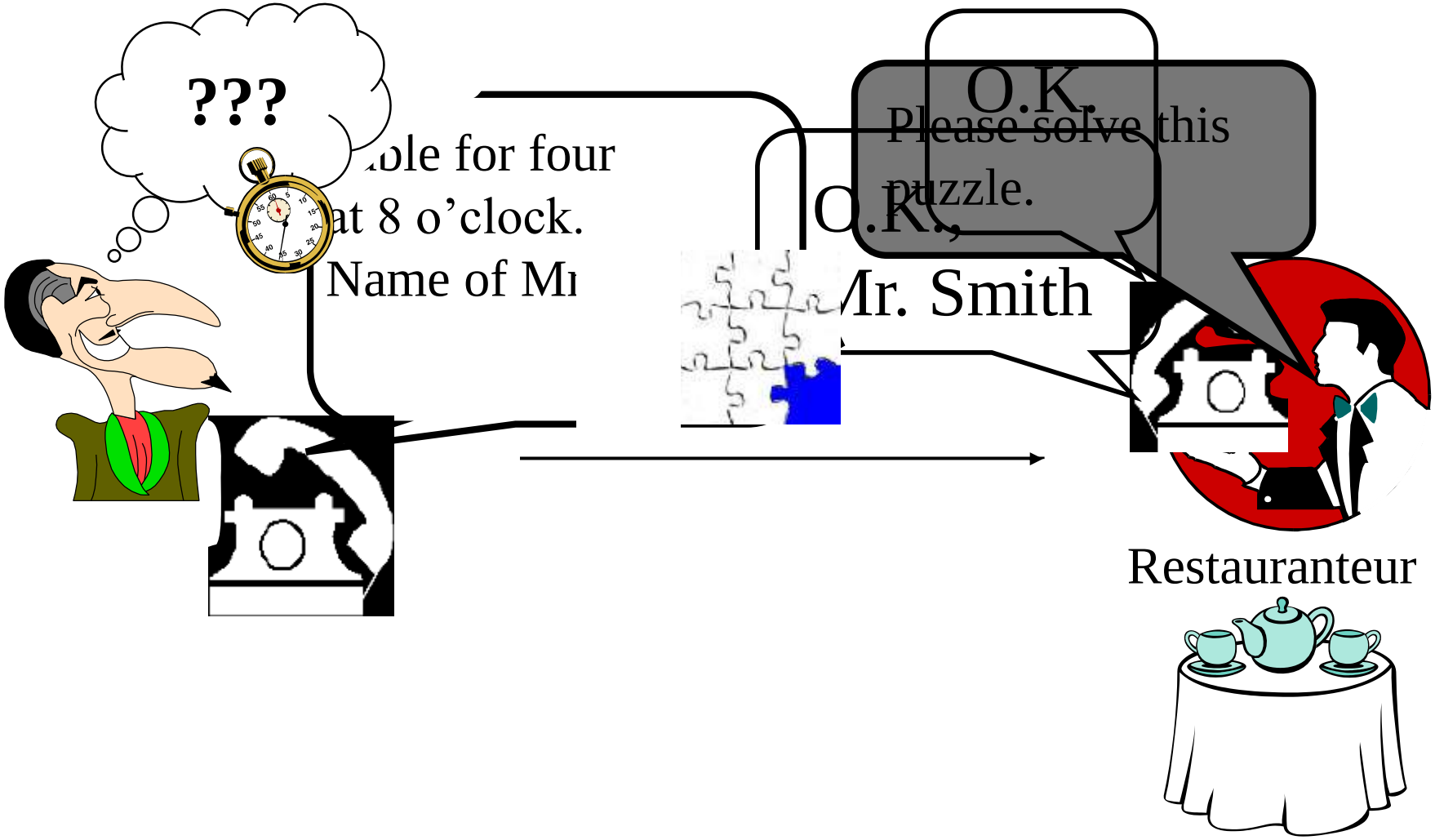
An Example Scenario: TCP SYN Flooding

"TCP connection, please."

"O.K. Please send ack."



Client Puzzle: Intuition

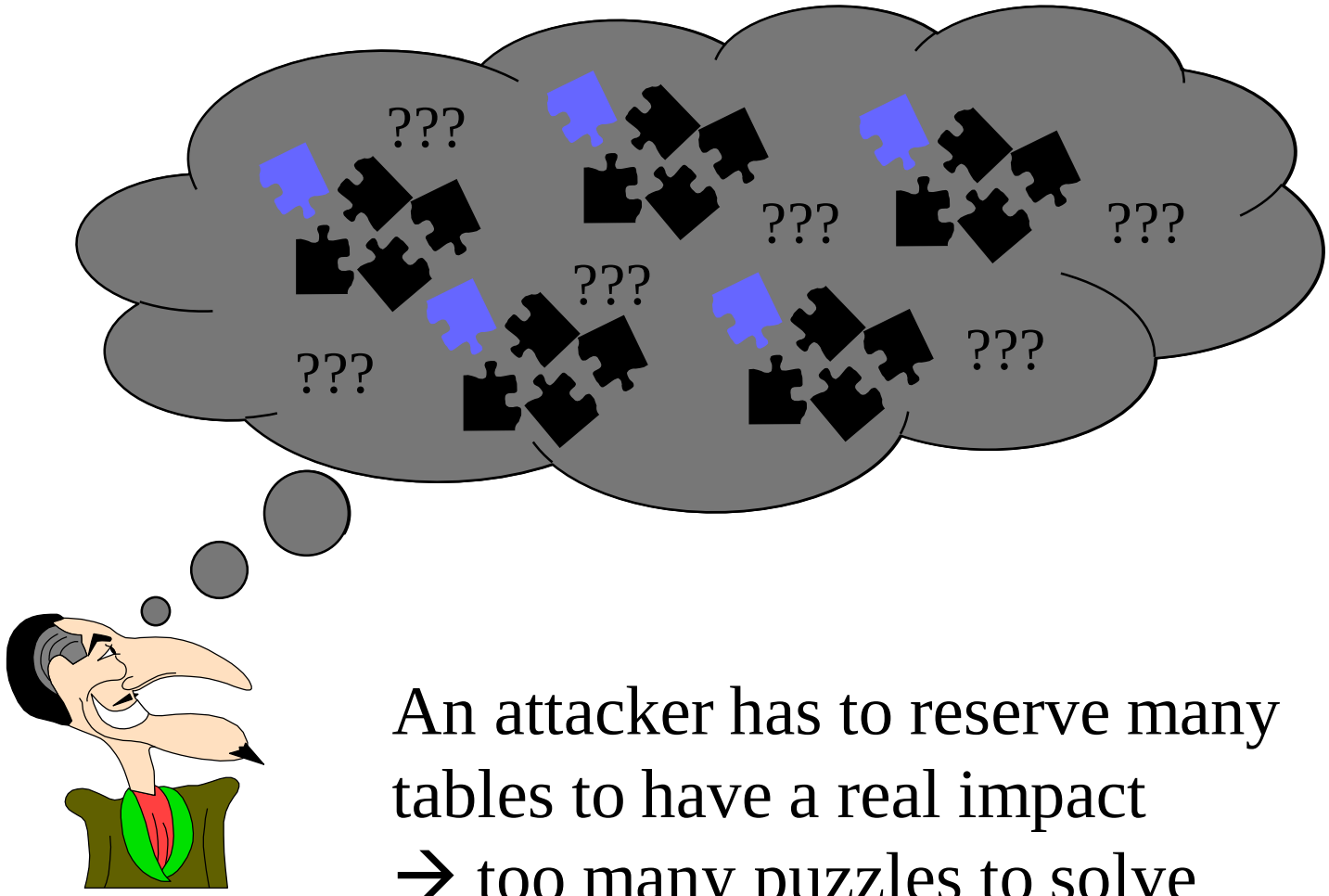


Client Puzzle: Intuition

- A puzzle takes an hour to solve
- There are 40 tables in restaurant
- Reserve at most one day in advance

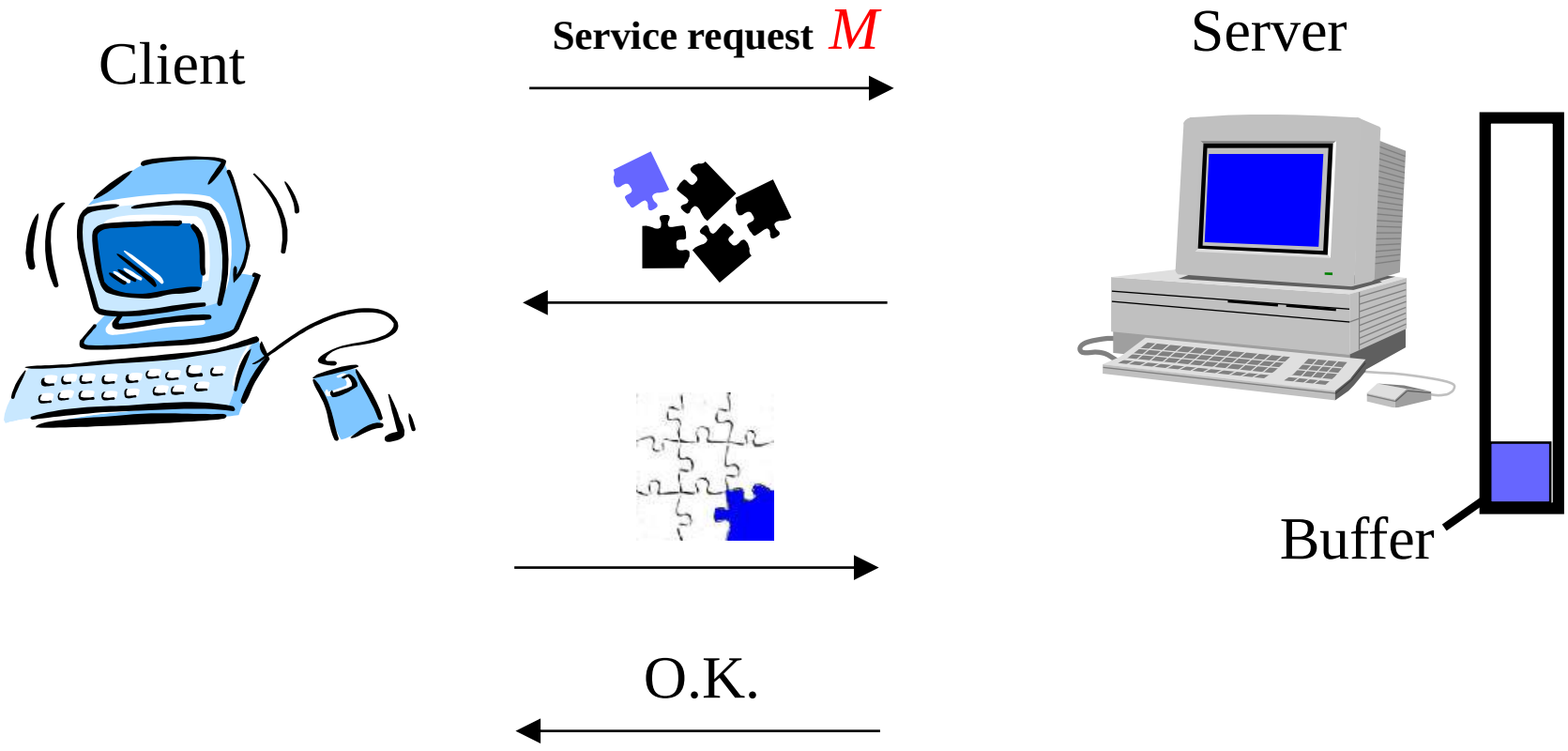
A legitimate patron can easily reserve a table

Client Puzzle: Intuition



An attacker has to reserve many tables to have a real impact
→ too many puzzles to solve

The Client Puzzle Protocol [1]



Puzzle Properties

- Puzzles are stateless
  - Or becomes storage DoS attack
- Puzzles are easy to verify
  - Or becomes comp. DoS attack
- Hardness of puzzles can be carefully controlled
  - Adaptivity
- Puzzles use standard cryptographic primitives

Puzzle Basis (Cont'd)

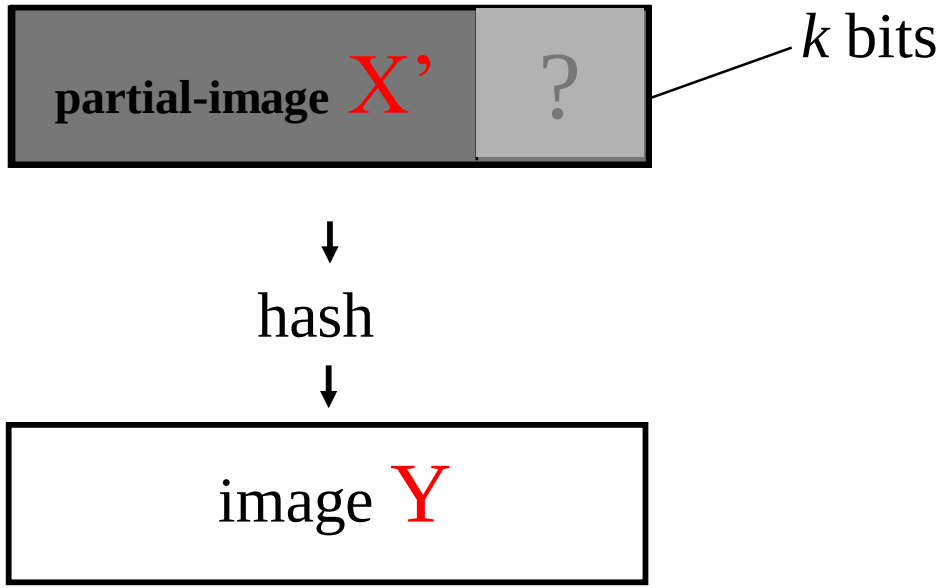
- Cryptographic hash functions (e.g., HMAC)
  - Keyed hash (weakest assumption, stronger version)

- Only way to solve puzzle (X', Y) is brute force method. (hash function is not invertible)

- Expected number of steps (hash) to solve puzzle: $2^k / 2 = 2^{k-1}$

- Do not forget 2^{k-1} if the expected steps is asked (hence avg.)
- One-time random guessing effort is 2^k

Puzzle Basis: Partial Hash Image



Pair (X', Y) is *k*-bit-hard puzzle

X' is disclosed so only Y-X' bits must be checked

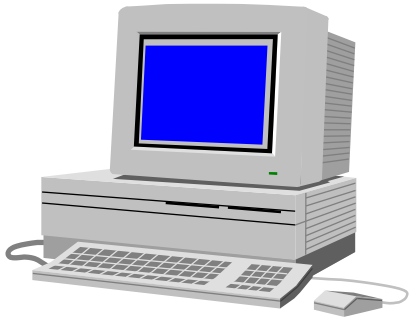
Pair (234,256) is k-22=bit hard puzzle

Puzzle Construction

Client

Service request M

Server



Secret S

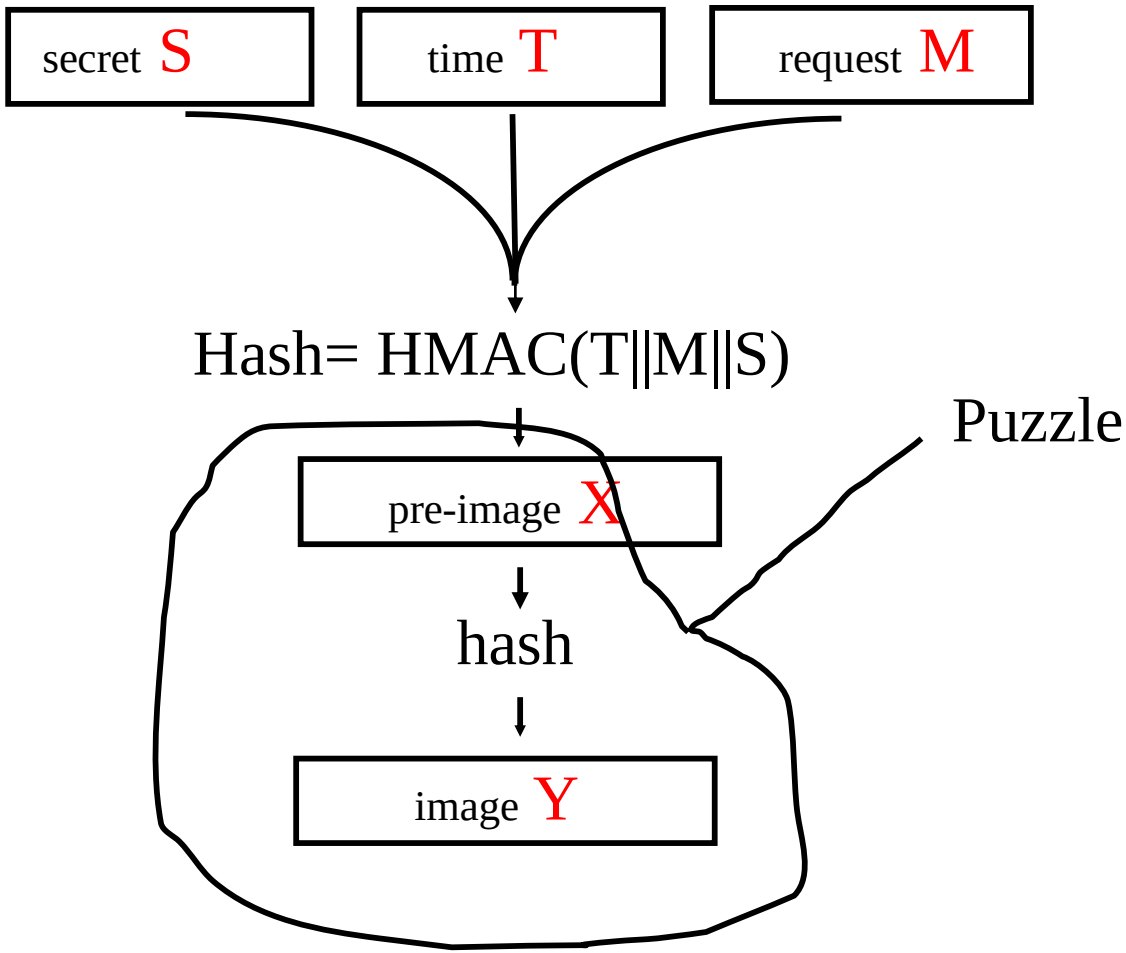
Example

- *You are given a (240,256) puzzle. Later 8-bits of security is added to strengthen the puzzle. If one hash computation takes 2 msec, how many msec would it take for a client (in average based on expected number steps) to solve the puzzle after the adjustment?*

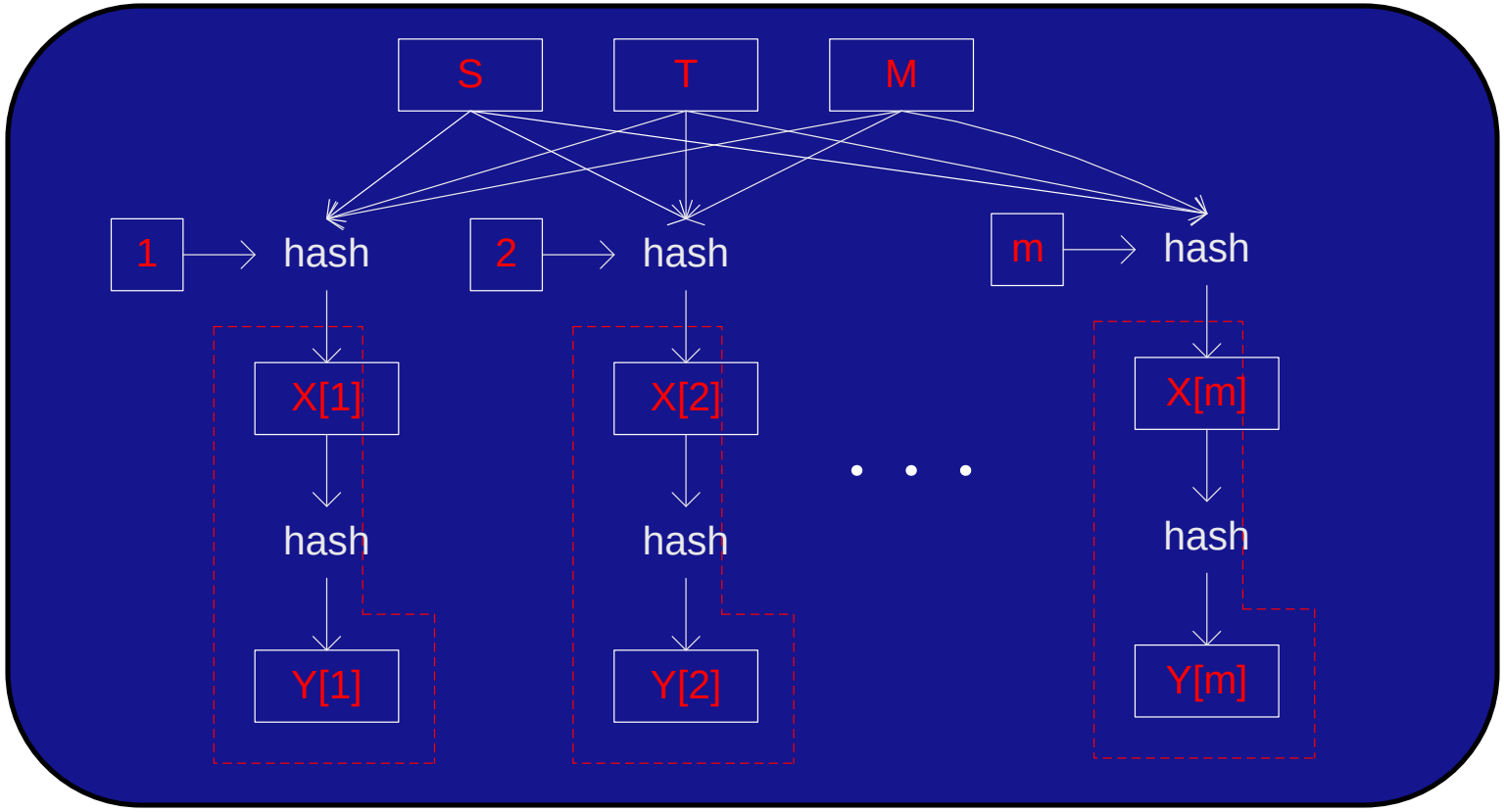
- Expected steps to solve (240,256) puzzle is 2^{k-1}, where k=256-40. So, 2^{15}
- 8-bit is added on top, 2^{23}
- Total time 2^{23}*2=2^24 msec

Puzzle Construction

Server computes:



Sub-puzzle



- Construct a puzzle consisting of *m* k-bit-hard sub-puzzles.
- Increase the difficulty of guessing attacks.
- Expected number of steps to solve (invert): $m \times 2^{k-1}$.

Why not use k+logm bit puzzles?

- (k+logm)-bit puzzle
  - Randomly guess it $m \times 2^k$

- One bigger puzzle versus multiple smaller (not solving it, but guessing at once), the successful probability, m=8, k=10
  - One (k+logm)-bit puzzle
    - $2^{-(k+\log m)}$ (e.g., $2^{-(13)}$)
  - M k-bit subpuzzles
    - $(2^{-k})^m = 2^{-km}$ (e.g., 2^{-80})

Example

Consider k-(226,256) bit puzzle and m=8. How many single (k+log_2(m)) bit puzzle is needed to achieve the same level of guessing attack resiliency (not solving the puzzle) that of m k-bit sub-puzzles?

- Puzzle A = Guess security of m=8, k-bit sub-puzzle is: 1/(2^{k*m) = 1/ 2^{30*8} = ½^{240}, where k=256-226=30 is the security of guess for one k-(226,256) bit puzzle

- Puzzle B = Guess security one (k+log_2(m)) puzzle is ½^{30+3} = ½^{33}

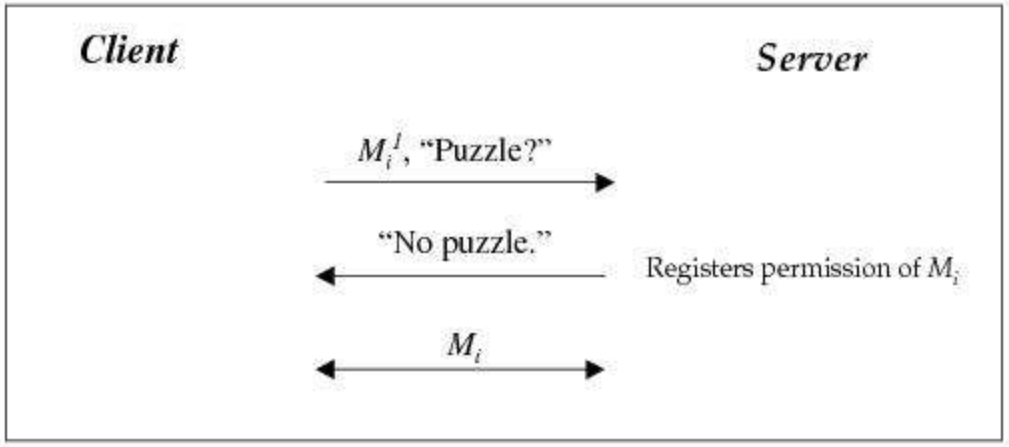
- To achieve the same level of security A, you need r number of puzzle Bs. R=207, so you need 2^{207} of puzzle B so that you end up 2^240 bit security.

Puzzle Properties

- Puzzles are stateless
- Puzzles are easy to verify
- Hardness of puzzles can be carefully controlled
- Puzzles use standard cryptographic primitives

A Possible Way to use Client Puzzle

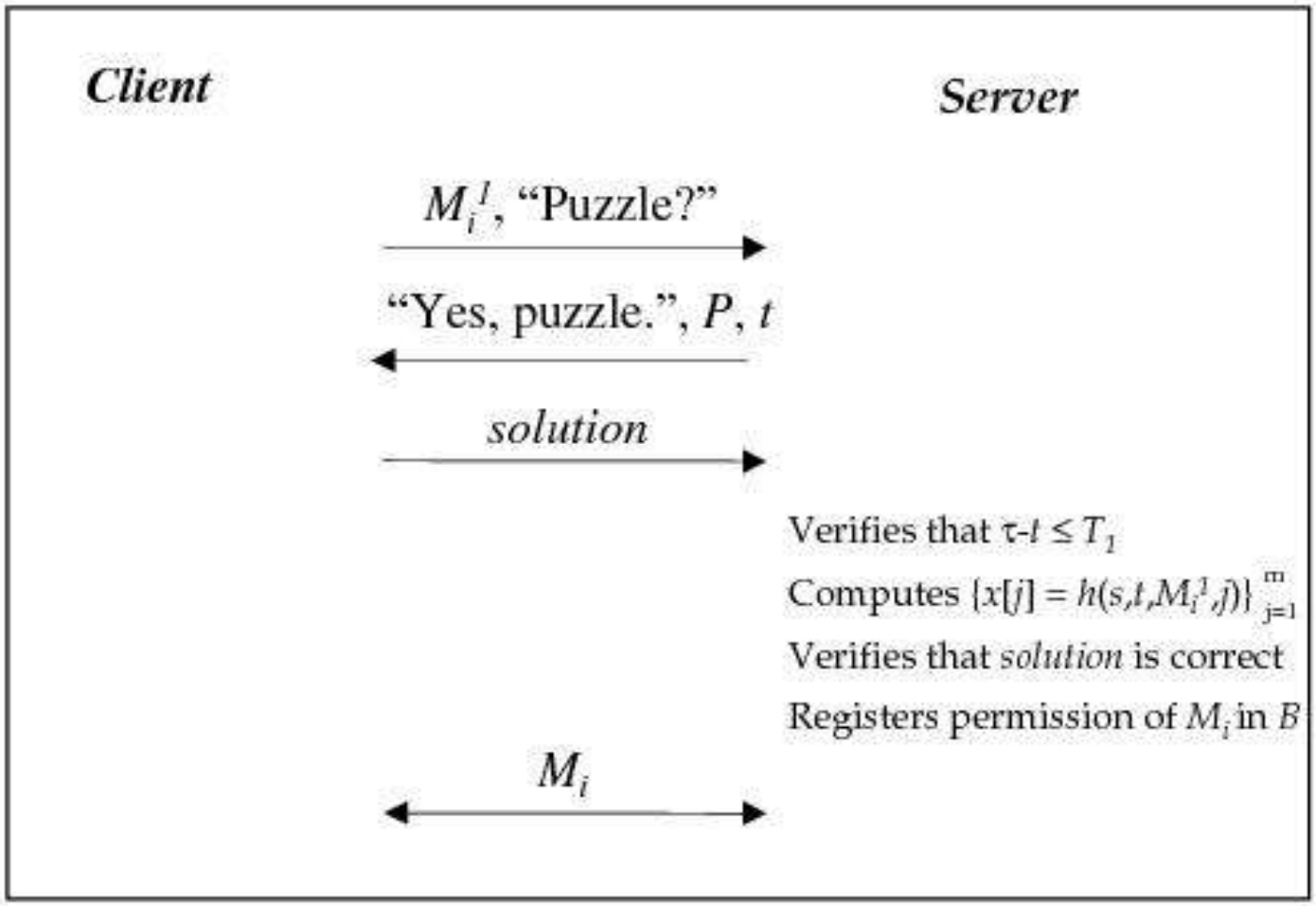
Client puzzle protocol (normal situation)



M_i^1 : first message of i-*th* execution of protocol M

A Possible Way to use Client Puzzle

Client puzzle protocol (under attack)



New Requirements from the Puzzle

- Preserve the previous properties PLUS
- The same puzzle can be given to several clients
  - Knowing solution for a client should not help the other (e.g., the adversary) to find another solution
  - Broadcast puzzles!
  - Not one-to-one connection required to initiate.
- The server should be able to pre-compute the broadcast puzzles. Even faster at online stage
  - Previous: *M* hash operations per-client (1-1),
- A client can re-use the same broadcast puzzle to create multiple solutions, multiple access tickets

Puzzle Construction

- S → All clients (broadcast): Digitally sign: k, Ts, N_S
- Client C → S: C, N_S, N_C, X
- S: verify $h(C, N_S, N_C, X)$ has k leading zero's

$$h(C, N_S, N_C, X) = \overbrace{000\ldots000}^{\text{the } k \text{ first bits of the hash}} \underbrace{Y}_{\text{the rest of the hash bits}}$$

h	=	a cryptographic hash function (e.g. MD5 or SHA)
C	=	the client indentity
N_S	=	the server's nonce
N_C	=	the client's nonce
X	=	the solution of the puzzle
k	=	the puzzle difficulty level
$000\ldots000$	=	the k first bits of the hash value; must be zero
Y	=	the rest of the hash value; may be anything

References

- [1] Juels, A. & Brainard, J. G. (1999). Client Puzzles: A Cryptographic Countermeasure Against Connection Depletion Attacks.. NDSS, : The Internet Society. ISBN: 1-891562-05-3
  - https://www.ndss-symposium.org/wp-content/uploads/2017/09/A-Cryptographic-Defense-Against-Connection-Depletion-Attacks-Ari-Juels.pdf

- [2] Tuomas Aura, Pekka Nikander, and Jussipekka Leiwo. 2000. DOS-Resistant Authentication with Client Puzzles. In Revised Papers from the 8th International Workshop on Security Protocols. Springer-Verlag, Berlin, Heidelberg, 170–177.
  - https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.106.9259&rep=rep1&type=pdf

- [3] Brent Waters, Ari Juels, J. Alex Halderman, and Edward W. Felten. 2004. New client puzzle outsourcing techniques for DoS resistance. In Proceedings of the 11th ACM conference on Computer and communications security (CCS '04). Association for Computing Machinery, New York, NY, USA, 246–256.
  - https://dl.acm.org/doi/10.1145/1030083.1030117