Message Integrity

Rohit Musti

CUNY - Hunter College

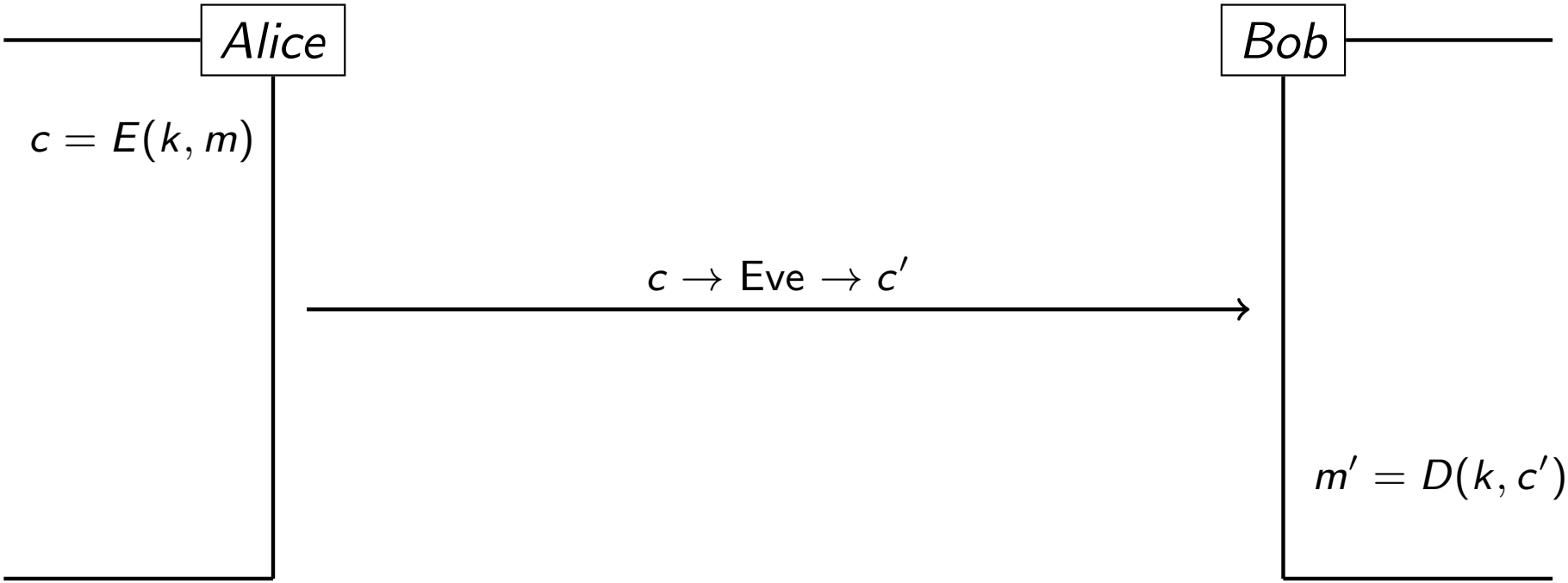
February 23, 2022

Message Integrity Motivation

- Up until now we have considered *passive* actors, who observe but do not attempt to modify messages
- What about *active* actors who are trying to modify the messages before they're recieved? Do any examples come to mind?
- Imagine you are delivering stock information to the public information. The message secrecy is not important (everyone is allowed to know the contents of the message), but message integrity is very important!
- Imagine you are delivering the results of medical tests to a patient, the message secrecy is important and the message integrity is (a mis-diagnosis can have massive ramifications)!
- Imagine you are downloading open source software online, it is extremely important to be able to verify that the software you downloaded is the correct source

Man in the Middle Attack

Let $\mathcal{E} = (E, D)$ be a cipher secure against chosen plaintext attacks



Message Integrity Requirement

- without a secret key, message integrity isn't possible
- an adversary can simply compute arbitrary tags on the message
- you may have learned that ethernet uses keyless message integrity system, this is not designed to be secure against attackers, rather check for random bit flips that occur during transmission

Message Authentication Codes (MAC)

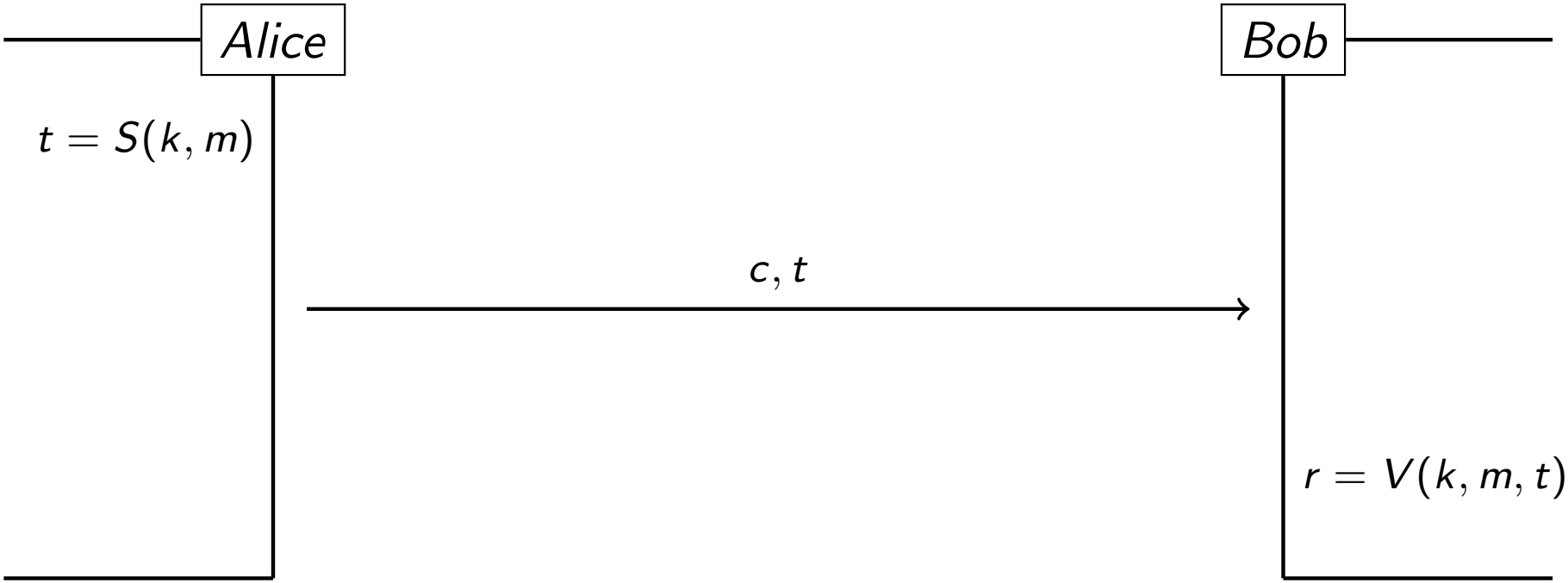
- a MAC system $\mathcal{I} = (S, V)$ is a pair of efficient algorithms s.t.
- S is the signing algorithm used to generate tags
- V is the verification algorithm used to verify tags

MAC Algorithms

- S is a probabilistic algorithm ($t \xleftarrow{R} S(k, m)$
- V is a deterministic algorithm ($r \xleftarrow{R} S(k, m, t)$
- Correctness requirement:

$$Pr[V(k, m, S(k, m))] = 1$$

MAC Visualized



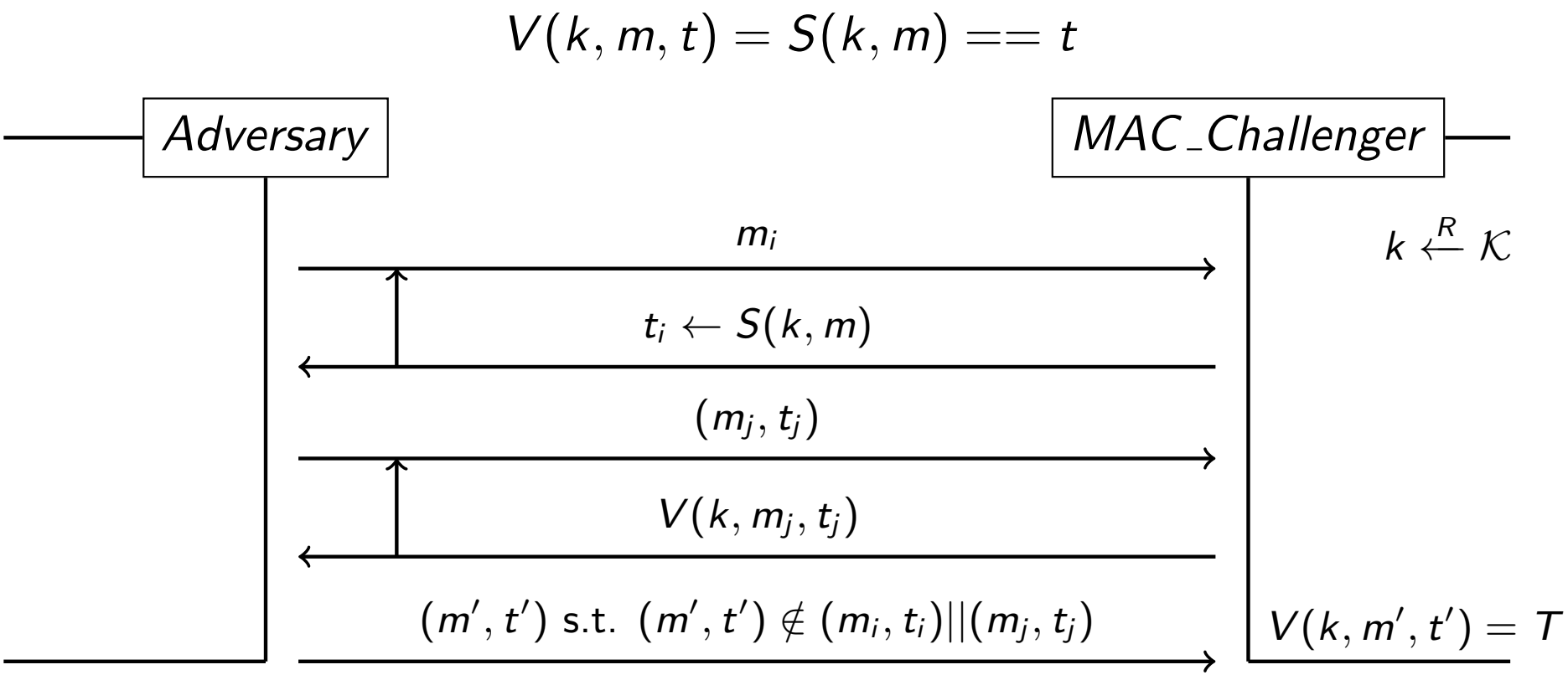
Determinstc MAC

- Deterministic MAC system example

$$V(k, m, t) = S(k, m) == t$$

- these systems have unique tags
- randomized algorithms tend to achieve better security and efficiency trade offs

MAC Attack Game



In Class Activity (20 minutes)

Design a MAC based on a PRF