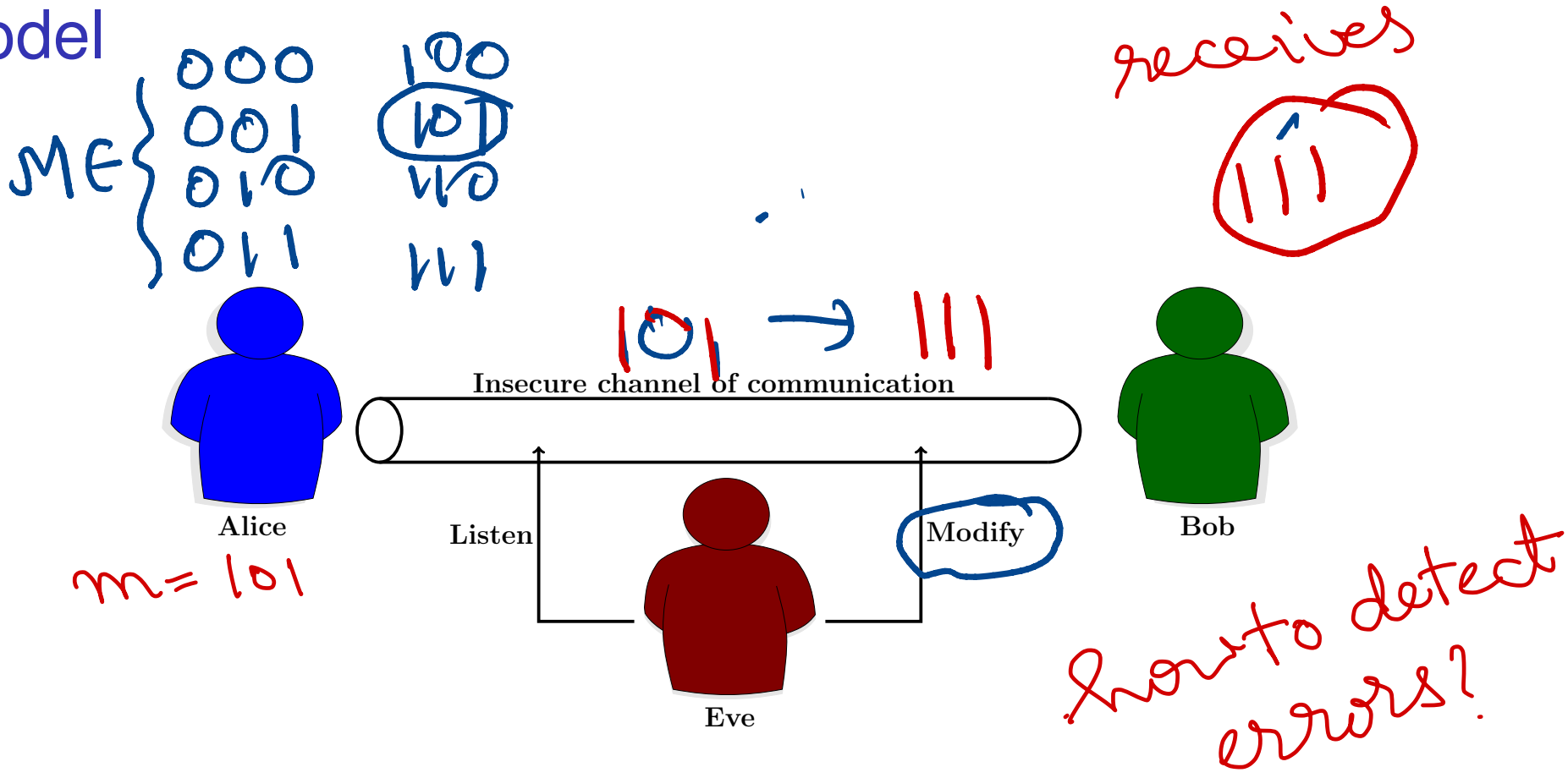
Hash Functions and Message Authentications

University of Birmingham

Outline of This Lecture

- Error Detection and non-cryptographic solutions
- Cryptographic Hash Functions
- Security of Hash Functions
- Message Authentication

Model



- Alice and Bob needs to communicate “correctly”.
- Example: Downloaded software may be corrupt.

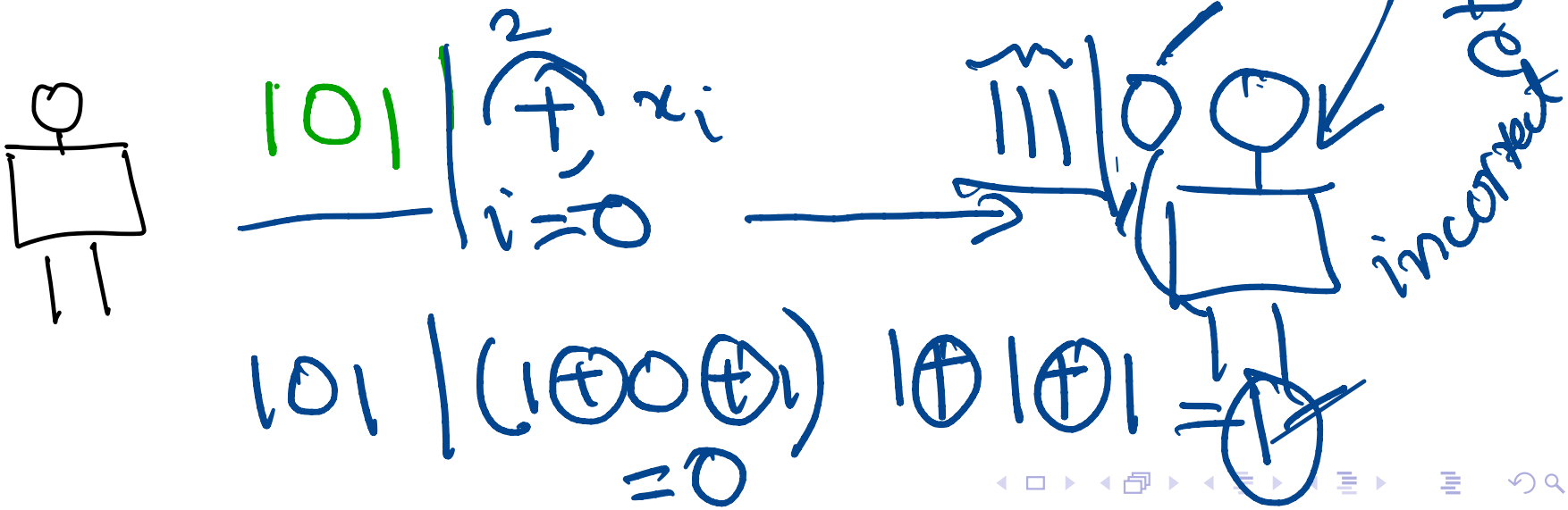
Error Detection in Communication

- Communications are prone to error as the channel is untrusted

Error Detection in Communication

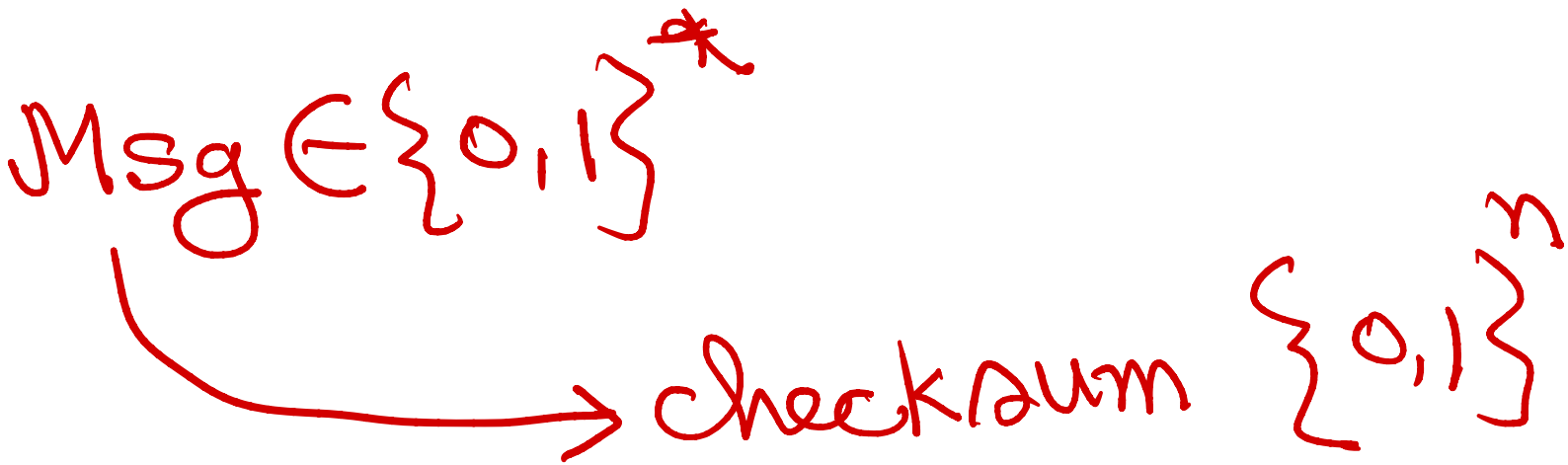
Sn! Eve could change the checksum! What happens with more than 1-bit!

- Communications are prone to error as the channel is untrusted
- IDEA: Add a checksum after the string
- Parity Bits: 1-bit error detection
- Cyclic Redundancy Check: Algebraic Error Detection

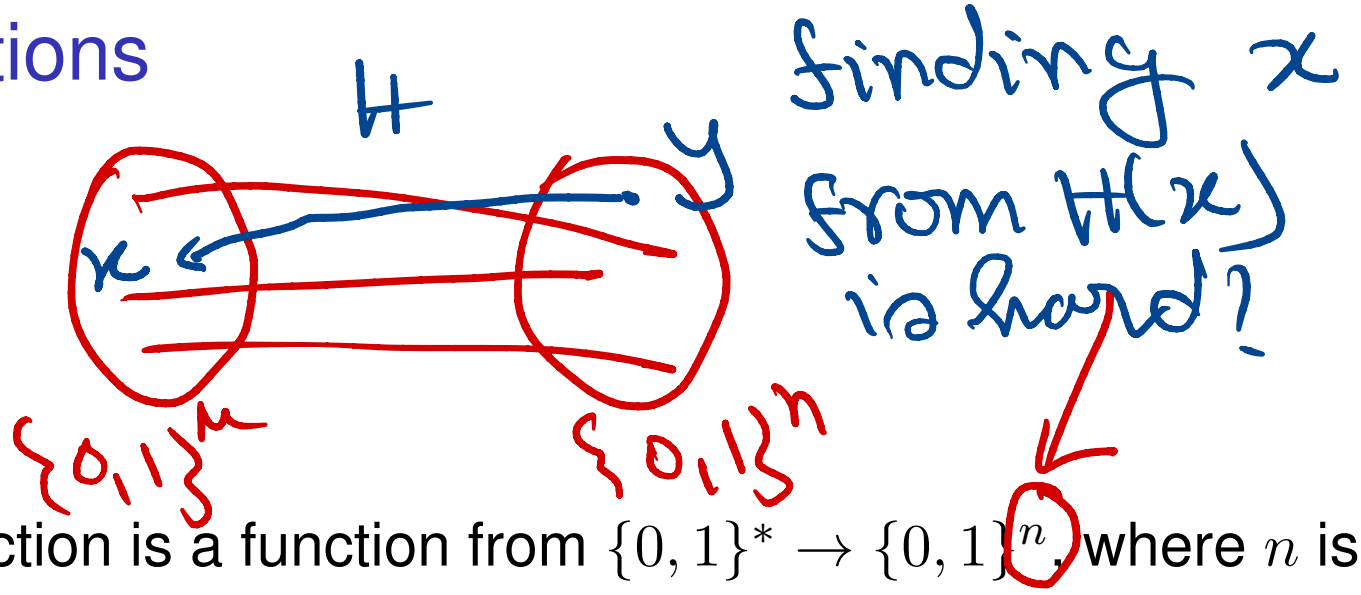


Hash Functions

A Hash function is a function from $\{0,1\}^* \rightarrow \{0,1\}^n$, where n is a fixed integer.



Hash Functions



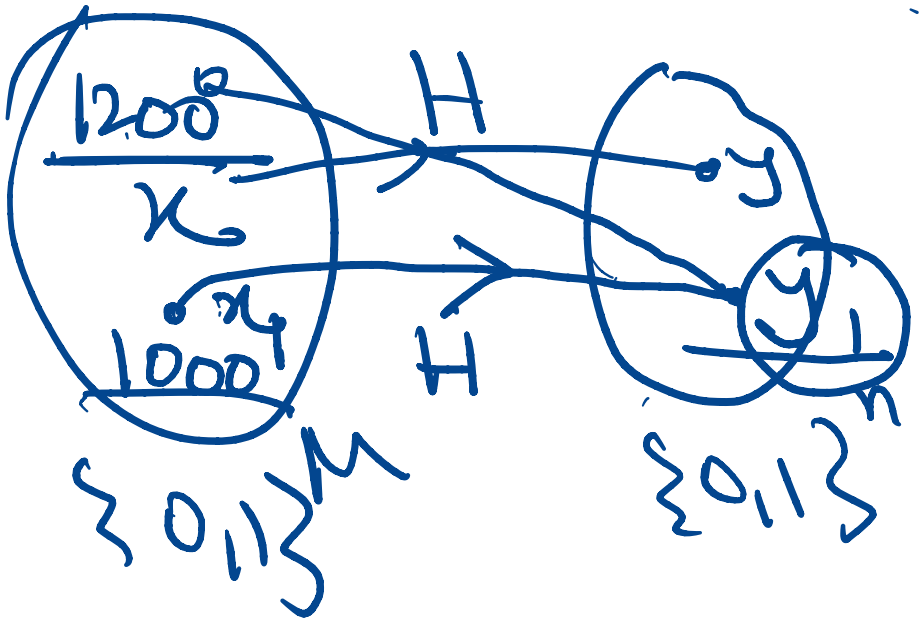
A Hash function is a function from $\{0,1\}^* \rightarrow \{0,1\}^n$, where n is a fixed integer.

- Practical hash functions have an upper bound μ on input message length with $\mu >> n$.

Hash Functions in Cryptography

Access control / login

- **Collision Attack**. If adversary could find *distinct* messages, m and m' such that $H(m) = H(m')$.
- **Preimage Attack.** Given a random $y \in \{0,1\}^n$, if the adversary could find a message m such that $H(m) = y$.



Hash Functions in Cryptography

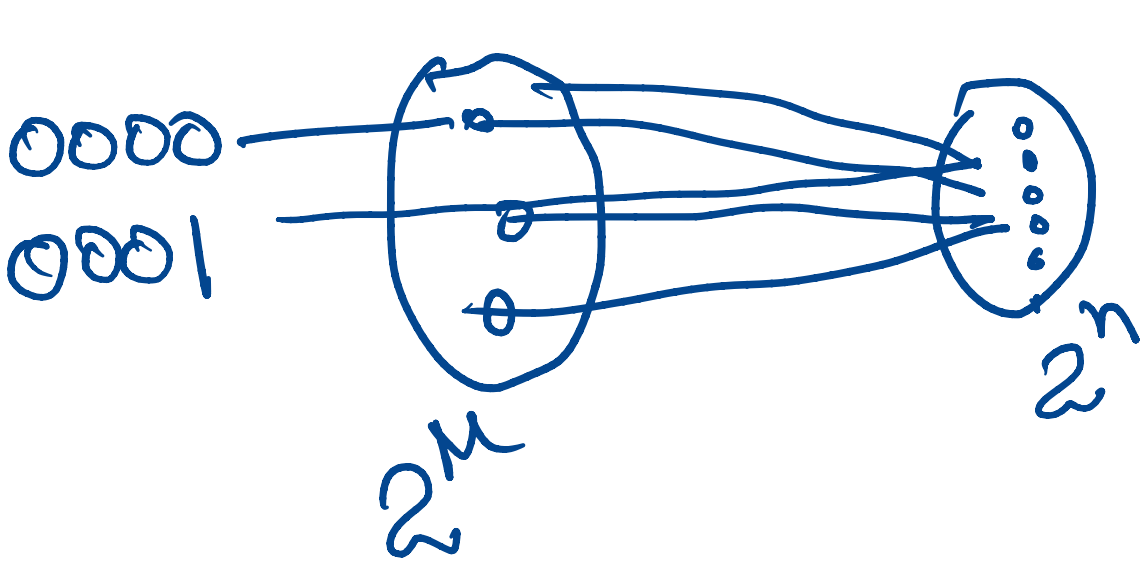
- **Collision Attack**. If adversary could find *distinct* messages, m and m' such that $H(m) = H(m')$.
- **Preimage Attack.** Given a random $y \in \{0,1\}^n$, if the adversary could find a message m such that $H(m) = y$.

The idea of *Length Extension Attack* is also attributed to hash function.

Hardness of Collision Attack

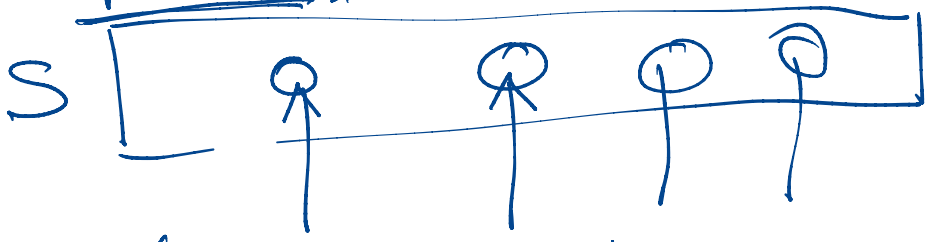
Hash Functions have collisions

- For a secure hash function $H : \{0,1\}^* \to \{0,1\}^n$, finding collision should take approximately $2^{n/2}$ computations.



$m >> n$
collisions in
$2^{n/2}$
many
computations

$N = 2^n$



for (i = 1 to q)

compute H(i)

For what q, there will be a repeatation

$i = 2$ — $\frac{1}{N}$ — No collision $\left(1 - \frac{1}{N}\right)$

$i = 3$ — $\frac{2}{N}$ — $\left(1 - \frac{2}{N}\right)$

$\vdots$

$i = q$ — $\frac{q-1}{N}$ — $\left(1 - \frac{q}{N}\right)$

No collision in any of the q samples.

$$= \Pr[\overline{coll}_2] \times \Pr[\overline{coll}_3] \times \cdots \Pr[\overline{coll}_q]$$

$$= \left(1 - \frac{1}{N}\right) \times \left(1 - \frac{2}{N}\right) \times \cdots \times \left(1 - \frac{q-1}{N}\right)$$

$$\approx \prod_{i=1}^{q-1} \left(1 - \frac{i}{N}\right) \approx \left(1 - \frac{\sum_{i=1}^{q-1} i}{N}\right) = \left(1 - \frac{q^2}{2^{n+1}}\right)$$

$1+2+\cdots+N = \frac{N(N+1)}{2}$

$1+2+\cdots+(q-1) = \frac{(q-1)\cdot q}{2}$

$$\Pr[\text{collision}]$$

$$= 1 - \Pr[\text{no collision after } q \text{ samples}]$$

$$= 1 - \left(1 - \frac{q^2}{2^{n+1}}\right) \quad \left| \frac{q^2}{2^{n+1}} = 1 \text{ when } q \approx 2^{n/2} \right.$$

$$= \frac{q^2}{2^{n+1}} \Rightarrow E[\text{evaluation before collision}] = 2^{n/2}$$

Usage of Cryptographic Hash Functions

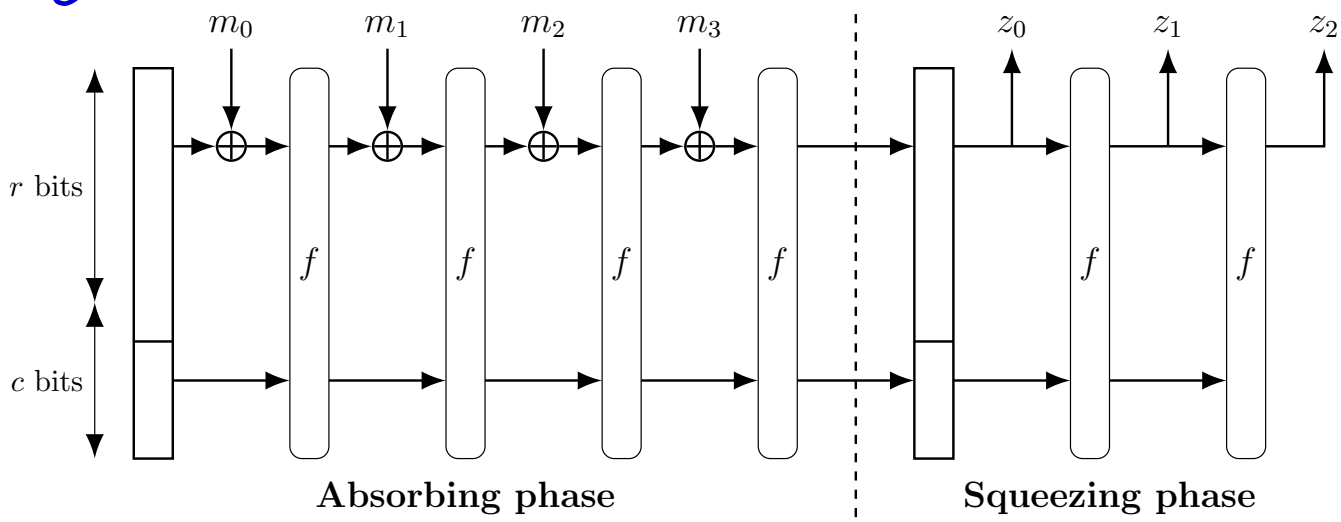
- Hash functions are useful where only the message (without the checksum) is transferred via the channel, and the receiver can compute and compare the checksum

example: file download checksums.

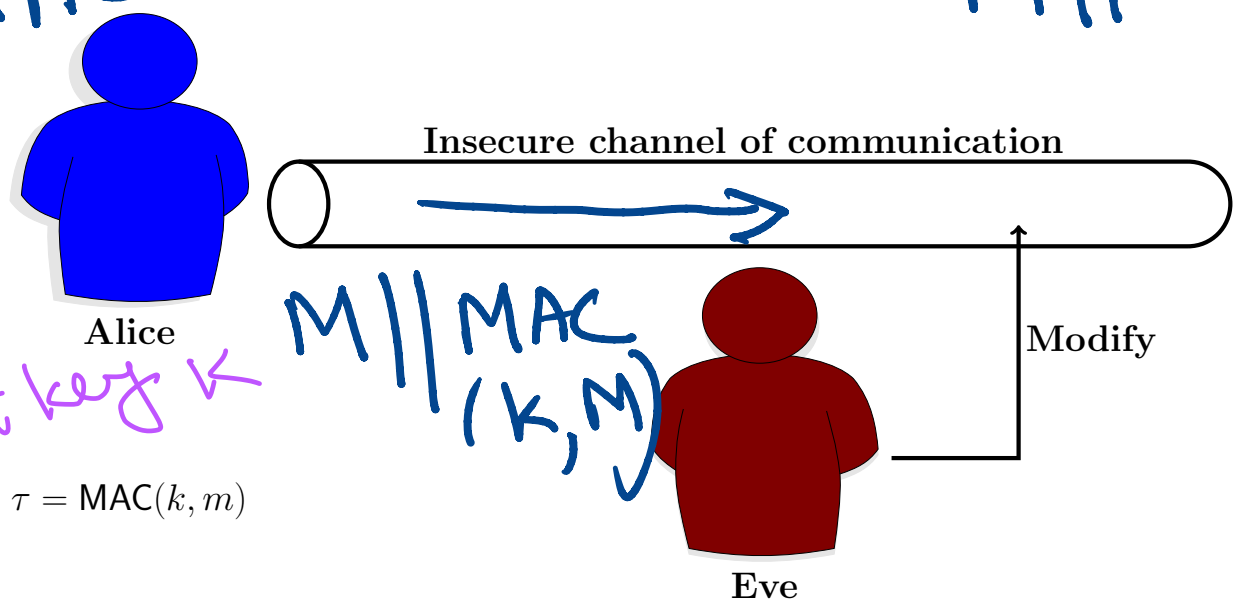
Design of Cryptographic Hash: SHA3

$in = m_0 || m_1 || m_2 || m_3 \cdots || m_t$

Here $t = 3$



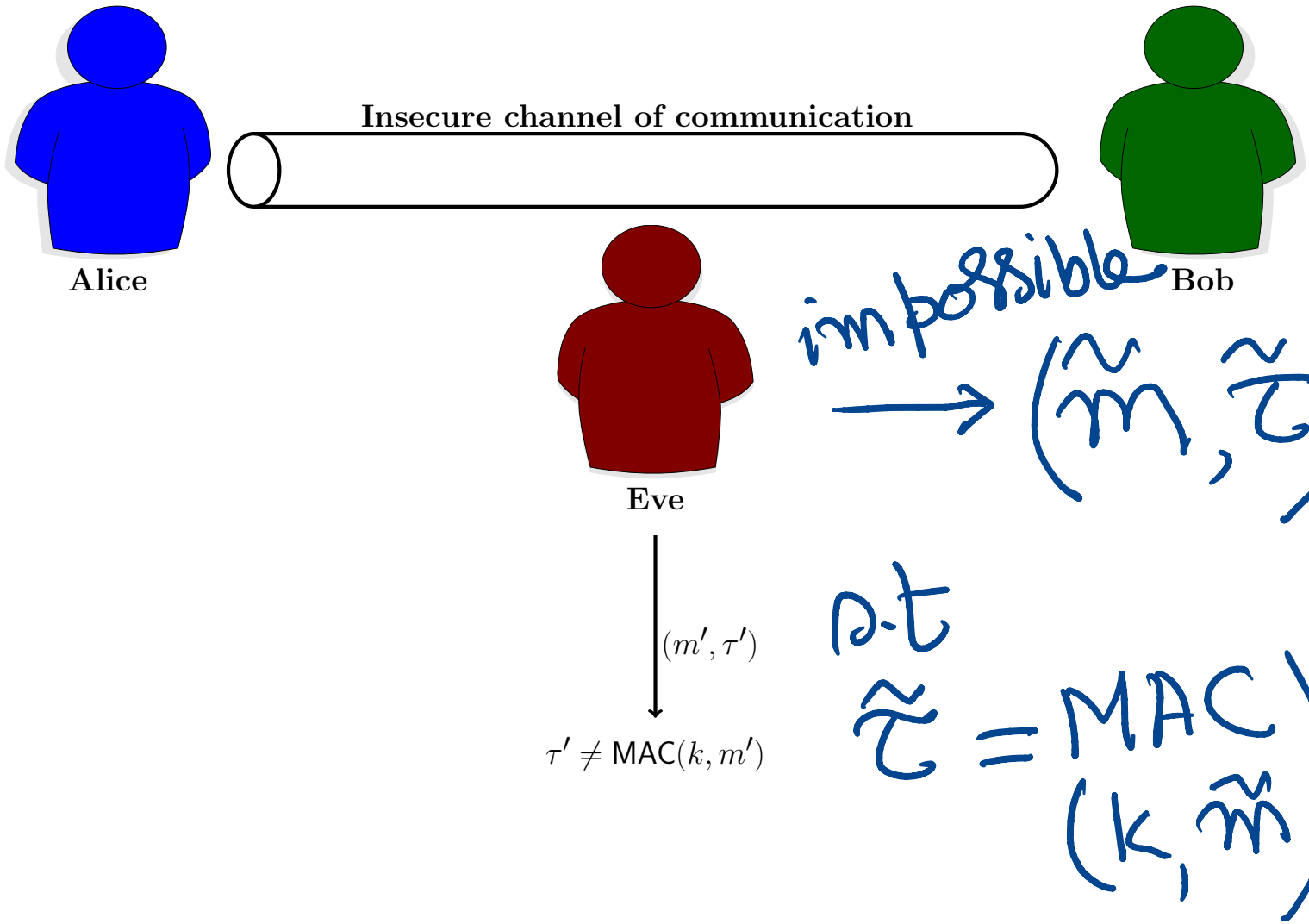
Message Authentication Codes



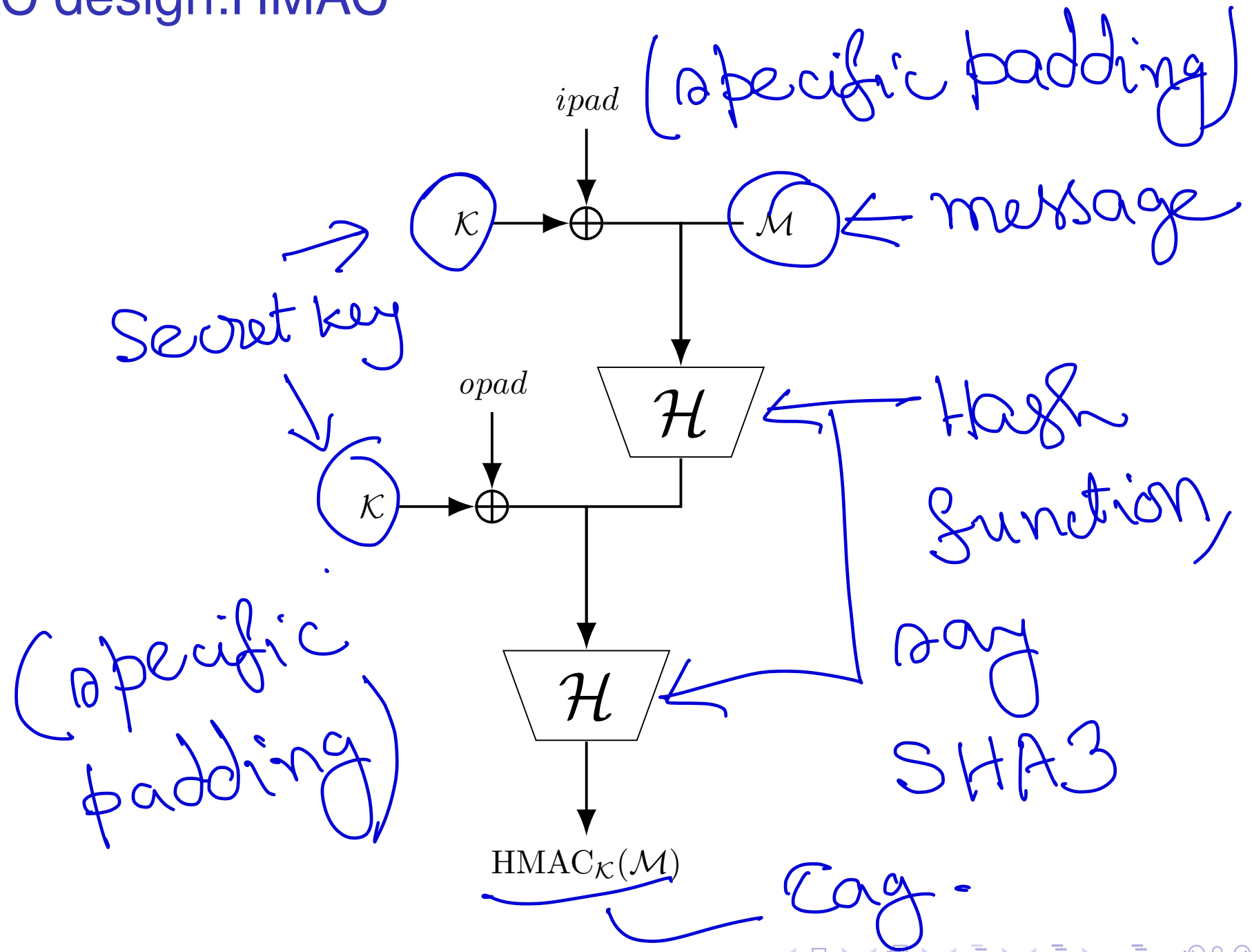
$\tau = \mathsf{MAC}(k, m)$

$\tau = \mathsf{MAC}(k, m)$

Message Authentication Codes: Security

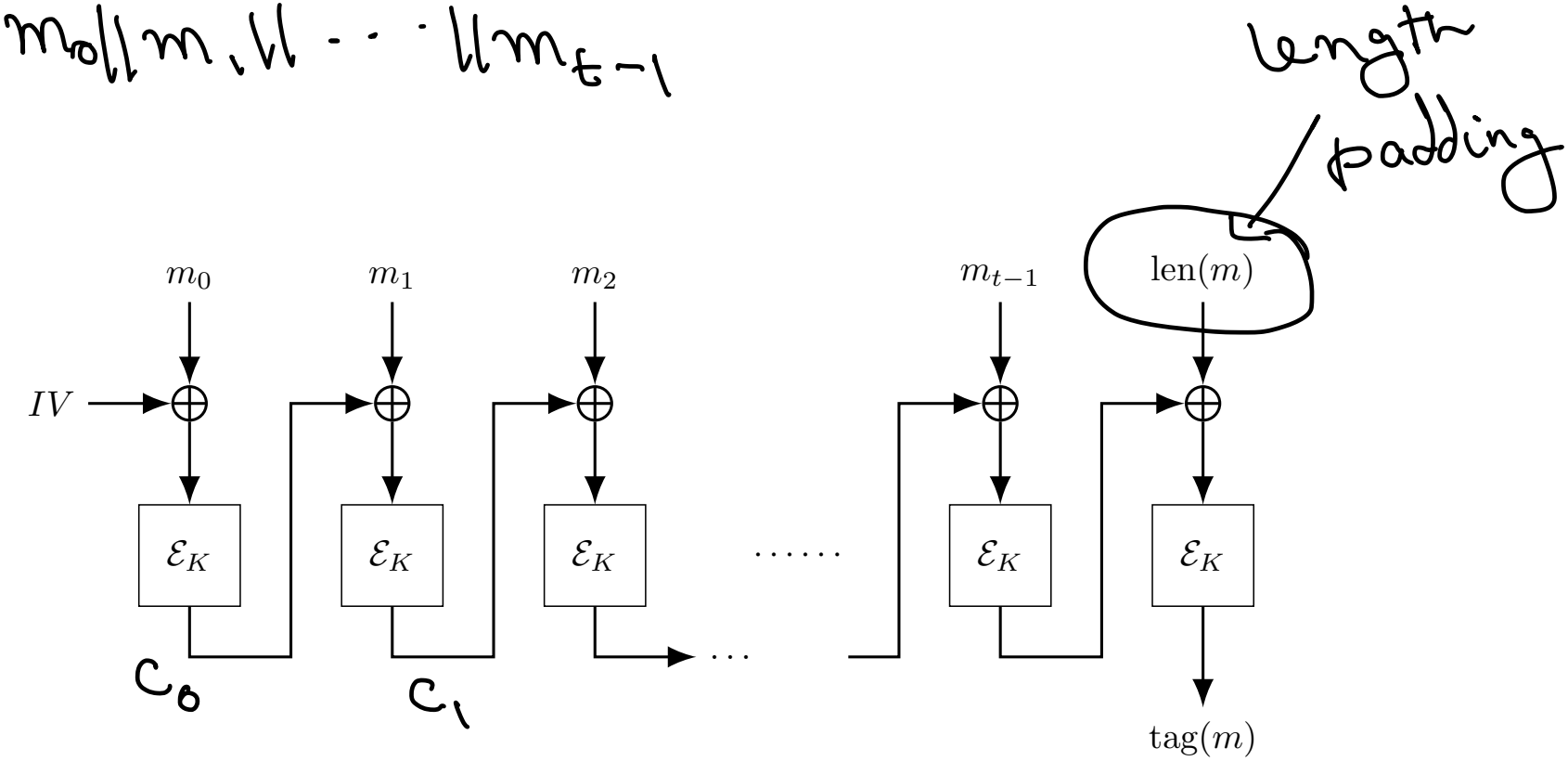


MAC design:HMAC



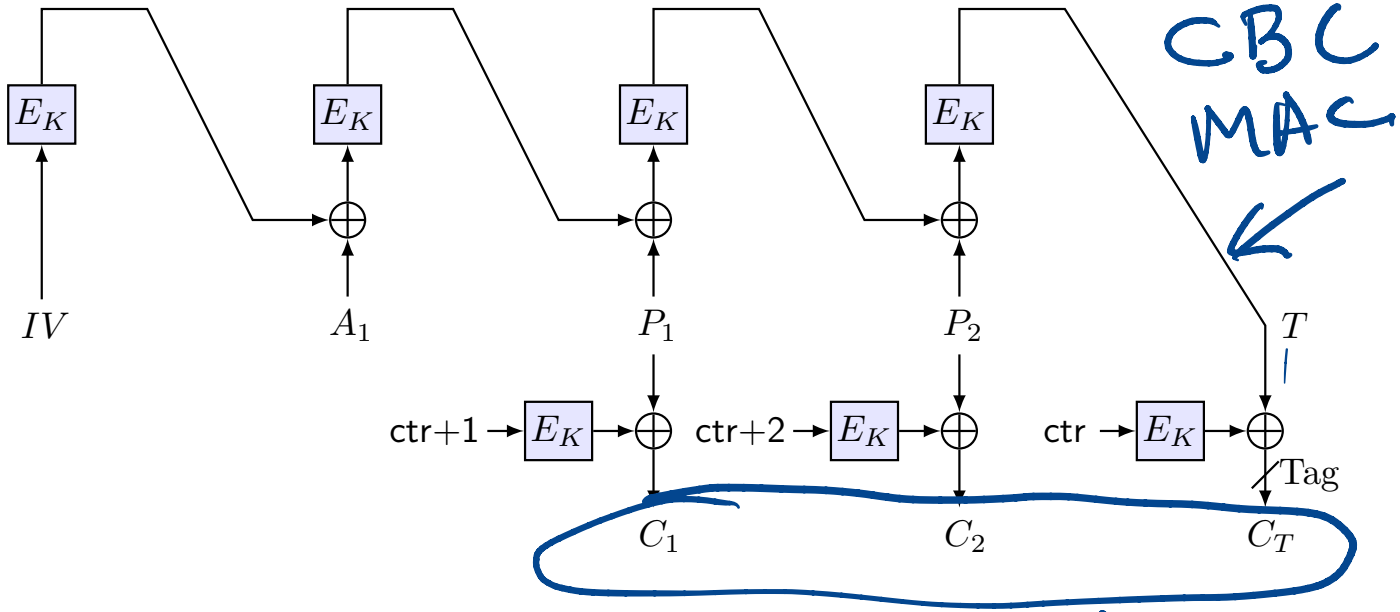
MAC design: CBC-MAC

$m = m_0 \| m_1 \| \cdots \| m_{t-1}$



$T(k,m) = \underbrace{E(k, IV \oplus m_0)}_{c_0} \| E_k(m_1 \oplus c_0) \|$

Authenticated Encryption:CCM



A_1 is auxiliary data, initialized to 0^n.