CS171: Cryptography

Lecture 11

Sanjam Garg

Cryptographic Hash Functions

Hash Functions

- Cryptographic Hash Functions: a deterministic function mapping an arbitrary long input string to a shorter output string.

- Hash functions can be keyed or unkeyed
  - In theory: Keyed
  - In practice: Unkeyed (fix a key once and for all)

Hash Function Definition

- Hash function $H: \{0,1\}^* \rightarrow \{0,1\}^\ell$
  - A collision is distinct x and x' such that $H(x) = H(x')$

- Classical use is data-structures where collisions are *undesirable*.
- However, for cryptographic hash functions, this will be a *requirement*.
- Even when an attacker is maliciously trying to find collisions.

Hash Function Definition

- Hash function $H: \{0,1\}^* \rightarrow \{0,1\}^{\ell}$
  - A collision is distinct x and x' such that $H(x) = H(x')$
- A hash function (with output length ℓ) is a pair of PPT algorithms (Gen, H) satisfying the following:
  - $Gen(1^n)$: Outputs s.
  - H: On input a key s and a string $x \in \{0,1\}^*$ output a string $H^s(x) \in \{0,1\}^{\ell(n)}$.

s is public

- If H^s is defined only for inputs $\{0,1\}^{\ell'(n)}$ where $\ell'(n) > \ell(n)$, then (Gen, H) is a fixed-length hash function for inputs of length ℓ'.

Hash Function Security

$HashColl_{A,\Pi}(n)$

1. Sample $s \leftarrow \text{Gen}(1^n)$.
2. Let x, x' be the output of $A(1^n, s)$.
3. Output 1 if $x \neq x'$ and $H^s(x) = H^s(x')$ and 0 otherwise.

$\Pi = (Gen, H)$ is collision resistant if

$\forall$ PPT A it holds that:

$\Pr\left[HashColl_{A,\Pi}(n) = 1\right] \leq$ negl(n)

No secrets!

Hash Function: In practice

- Have a fixed output length just like block ciphers
- Also, they are unkeyed.
  - Problematic in theory

Generic Attacks on Hash Functions

- Hash function $H: \{0,1\}^{\ell'} \to \{0,1\}^{\ell}$ where $\ell' > \ell$
  - A collision is distinct x and x' such that $H(x) = H(x')$

- Can we find collisions?
- Yes, let $x_1, \ldots x_{2^{\ell}+1}$ be arbitrary distinct values in $\{0,1\}^{\ell'}$
- Then we have that $\exists\ i, j$ such that $H(x_i) = H(x_j)$

Will drop the superscript s which is now implicit.

Generic Attacks on Hash Functions

- Hash function $H: \{0,1\}^{\ell'} \rightarrow \{0,1\}^{\ell}$ where $\ell' > \ell$
  - A collision is distinct x and x' such that $H(x) = H(x')$

- Can we find collisions faster?
- Let $x_1, \ldots x_q$ be distinct values in $\{0,1\}^{\ell'}$ then what is the probability that we will find a collision?
- When $q > 2^{\ell}$ then the probability is 1, what if q is smaller?
- Important: A much smaller value of q suffices, i.e. $2^{\ell/2}$

Heuristic Analysis

- View H as a random function

  For $x_1, \ldots x_q$ $\Pr\left[\exists\ i, j\ H(x_i) = H\left(x_j\right)\right] \approx \frac{q^2}{2 \cdot 2^\ell}$

  - Thus, probability is ½ for $q\ = \Theta(2^{\ell/2})$
- Birthday problem: What is the probability that q people have birthday on the same day of the year?
  - Only need $\sqrt{365} \approx 23$ people to get a collision with probability ½
- Attempt 1: The probability two hashes collide is $1/2^\ell$. Thus, probability of collision is $\binom{q}{2} \cdot\ 1/2^\ell$.
  - Error: The probabilities are not independent.
  - See Appendix A.4 (in book) for analysis.

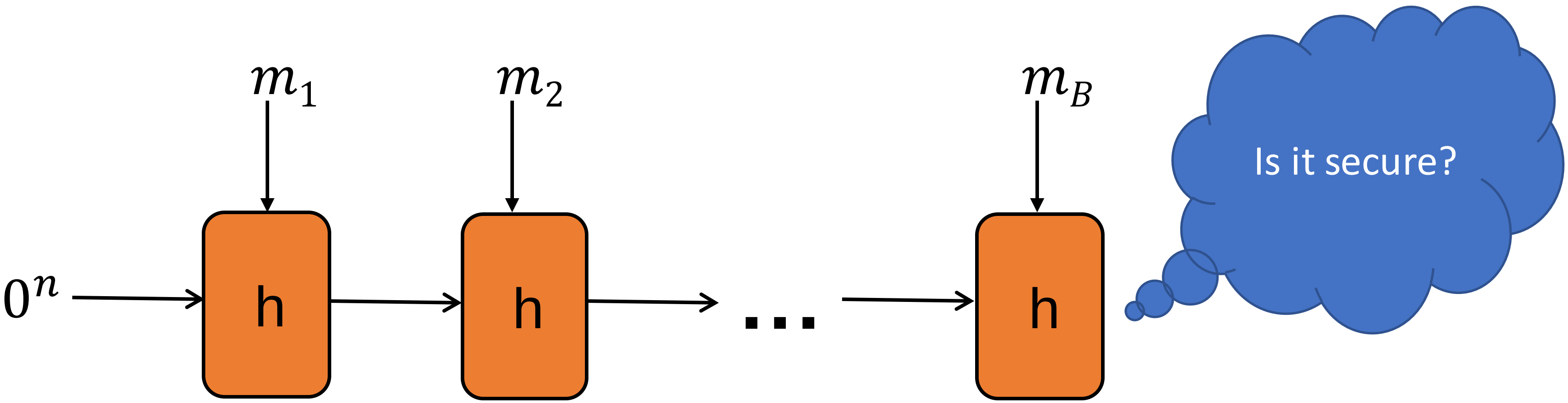
Implications of the birthday attack

- Need hash output to be $\ell = 2n$ to get security against attackers running in time 2^n.
- This is double the length of the keys needed for block ciphers.

- Thus, to get 128-bits of security we need a hash output of 256 bits.
- Necessary but not a sufficient condition
  - Birthday attack works for all hash functions, but there could be other more ``devastating'' attacks.

Domain Extension: The Merkle-Damgård Transform

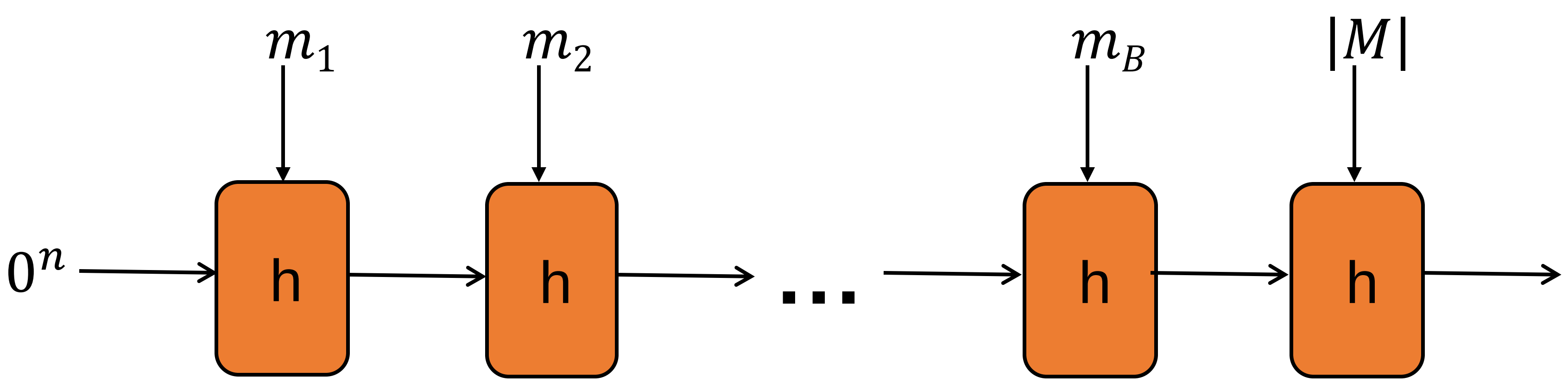
- Given (Gen, h) a fixed length hash function from $2n$ bit inputs to n bit outputs. Construct (Gen, H) as follows:
- H(M): Parse M as $m_1 \ldots m_B$, where m_B is padded with 0s to make it of appropriate length



$$H(M) = H(M||0)$$

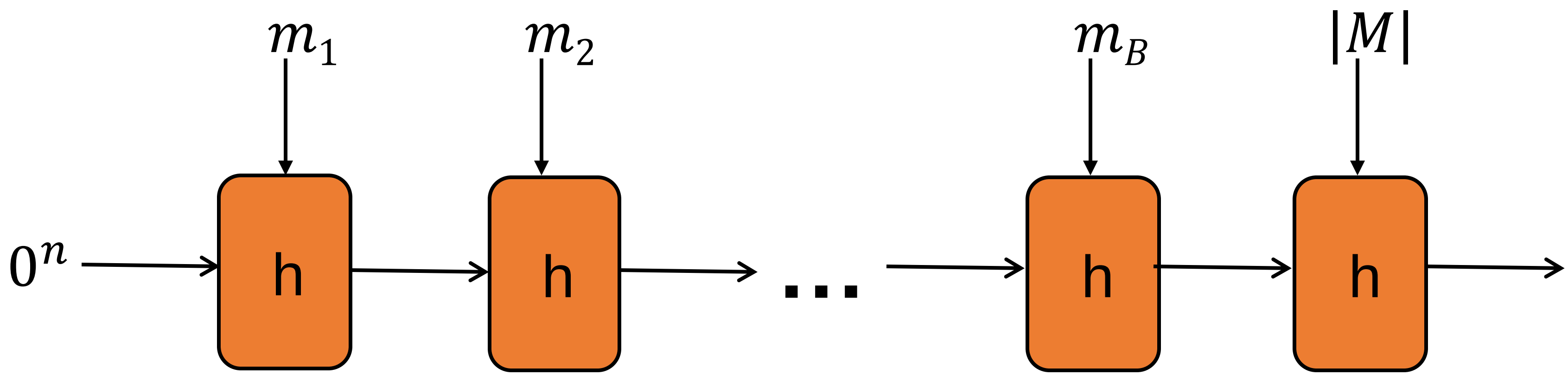
Domain Extension: The Merkle-Damgård Transform

- Given (Gen, h) a fixed length hash function from $2n$ bit inputs to n bit outputs. Construct (Gen, H) as follows:
- H(M): Parse M as $m_1 \ldots m_B$, where m_B is padded with 0s to make it of appropriate length



Domain Extension: The Merkle-Damgård Transform

- If h is collision-resistant, then so is H.



- Proof: Collision on H
  - Say $H(m_1, \ldots, m_B) = H\left(m'_1, \ldots, m'_{B'}\right)$
  - $|M| \neq |M'|$, then $h(\cdot, |M|) = h(\cdot, |M'|)$
  - $|M| = |M'|$, largest i such that $h(\cdot, m_i) = h(\cdot, m'_i)$

MACs using Hash Functions: Hash-and-MAC

- Previously, saw construction of MACs from PRF/block-cipher
- Also, CBC-MAC allowed to construct MACs with short tag lengths for arbitrary length messages
- Hash-and-MAC paradigm to do the same.

Hash-and-MAC

- Let $(Gen, Mac, Vrfy)$ be a MAC on messages of length $\ell(n)$ and (Gen_H, H) be a hash function with output length $\ell(n)$. Then MAC $(Gen', Mac', Vrfy')$ for arbitrary-length messages is:
- $Gen'(1^{\mathrm{n}})$: Output $k' = (k, s)$ where $k \leftarrow Gen(1^n)$ and $s \leftarrow Gen_H(1^n)$.
- $Mac'_{k'}(m \in \{0,1\}^*)$: Output $Mac_k(H^s(m))$
- $Vrfy'_{k'}(m, t)$: Output 1 iff $Vrfy_k(H^s(m), t) = 1$.

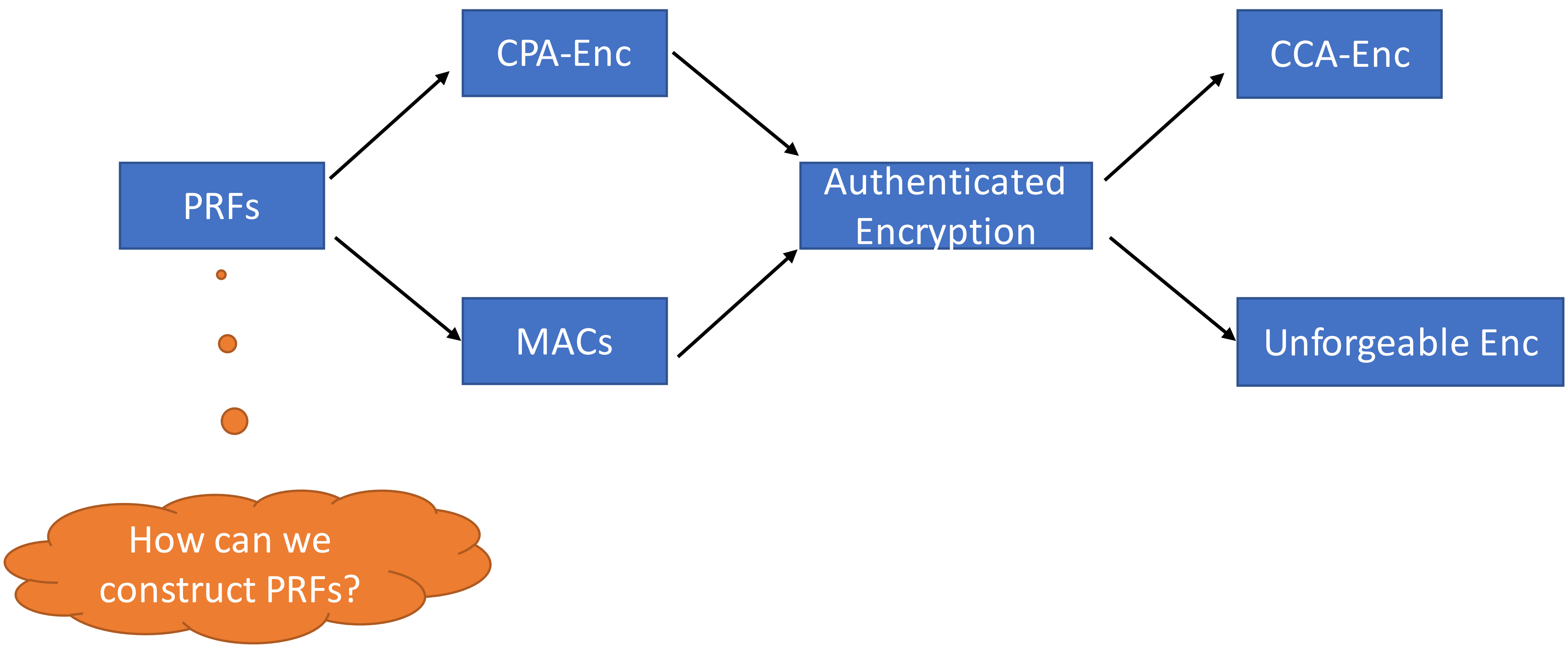
Security

- If the MAC is secure for fixed-length messages and H is collision-resistant, then the construction on previous slide is a secure MAC for arbitrary-length messages.
- Proof Sketch: Say the attacker outputs (m^*, t^*)
  - Case I: $H(m^*) = H(m_i)$ for some i, then we have a collision on H.
  - Case II: $H(m^*) \neq H(m_i)$ for all i, then we have a forgery for the underlying fixed-length MAC.

Other Applications

- Blockchains
- Virus Fingerprinting
- Deduplication
- Peer-to-peer (P2P) file sharing

See So Far…



PRFs
CPA-Enc
MACs
Authenticated Encryption
CCA-Enc
Unforgeable Enc
How can we construct PRFs?

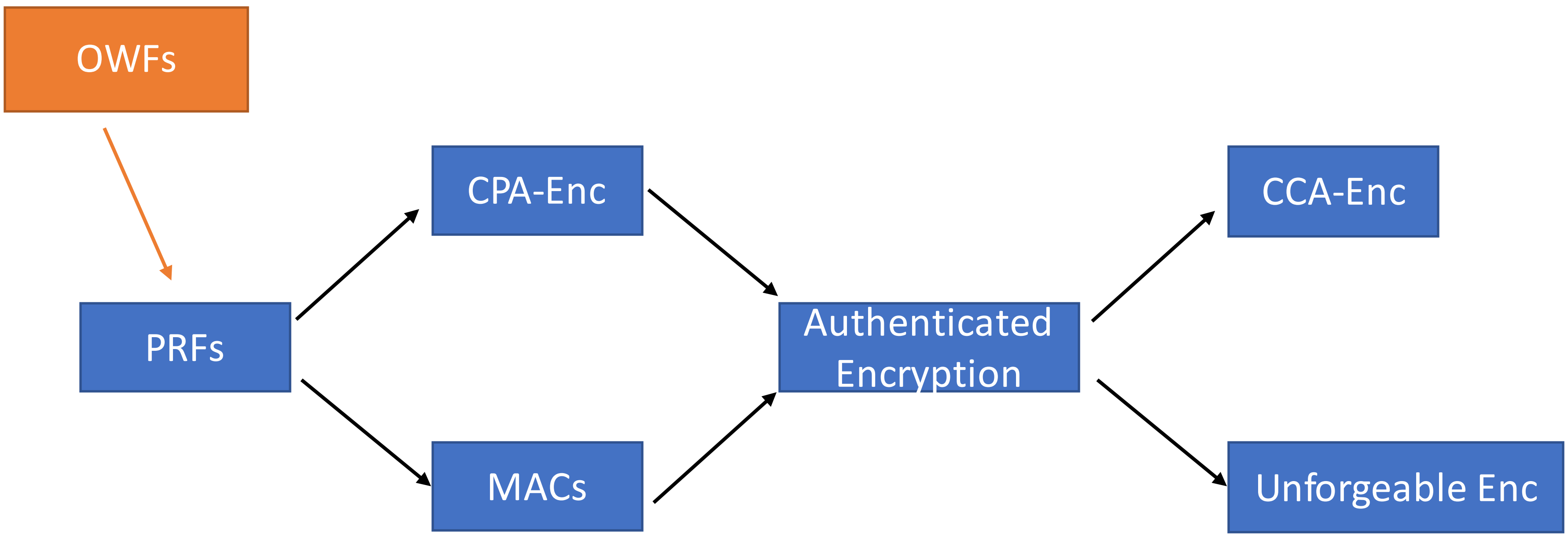
Constructions of PRFs/Block-Ciphers

- Theoretical Constructions
- Practical Constructions

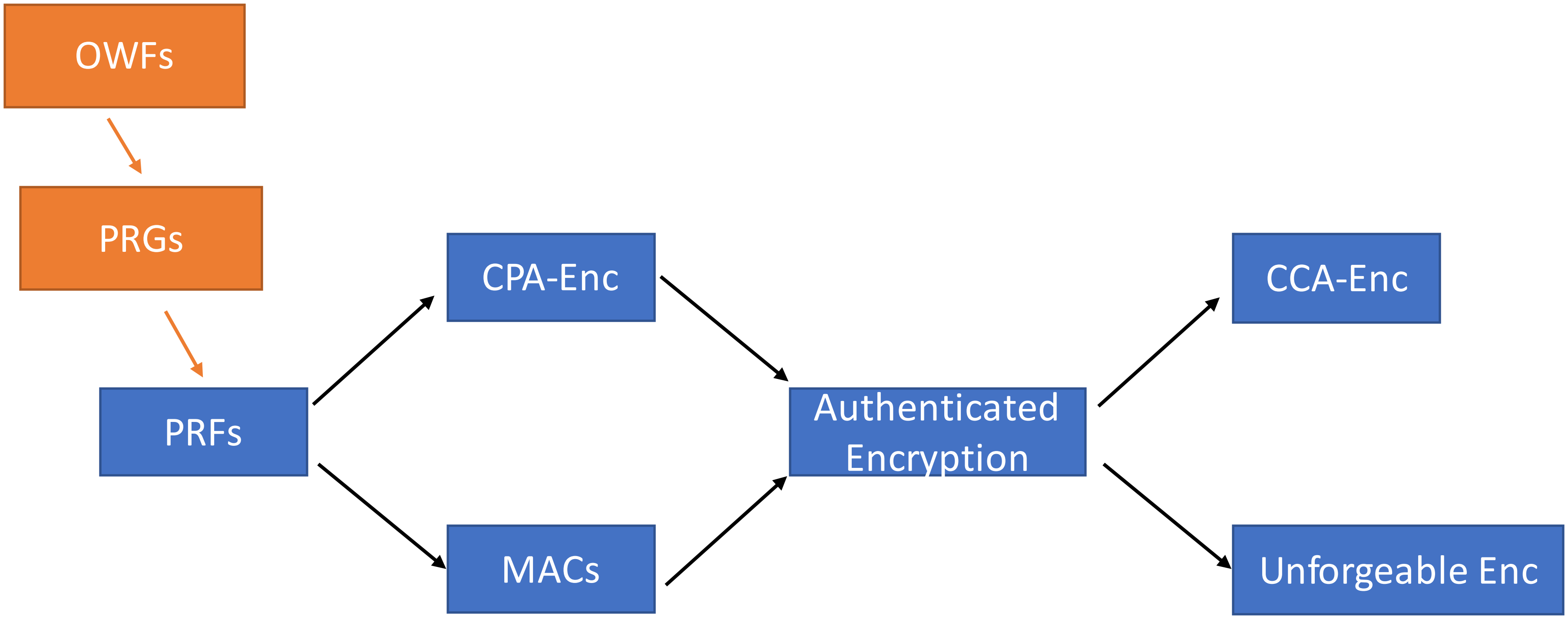
A rigrous approach!

A bit of an art!

One-Way Functions

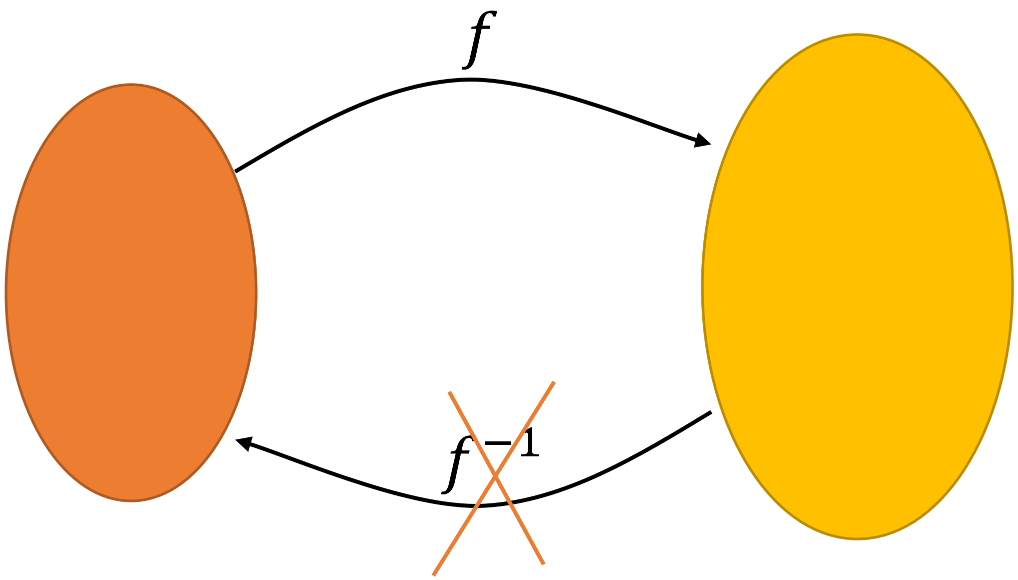


More accurately…



Define: One-Way Functions

- A function $f\colon \{0,1\}^* \to \{0,1\}^*$ that is easy to compute but hard to invert

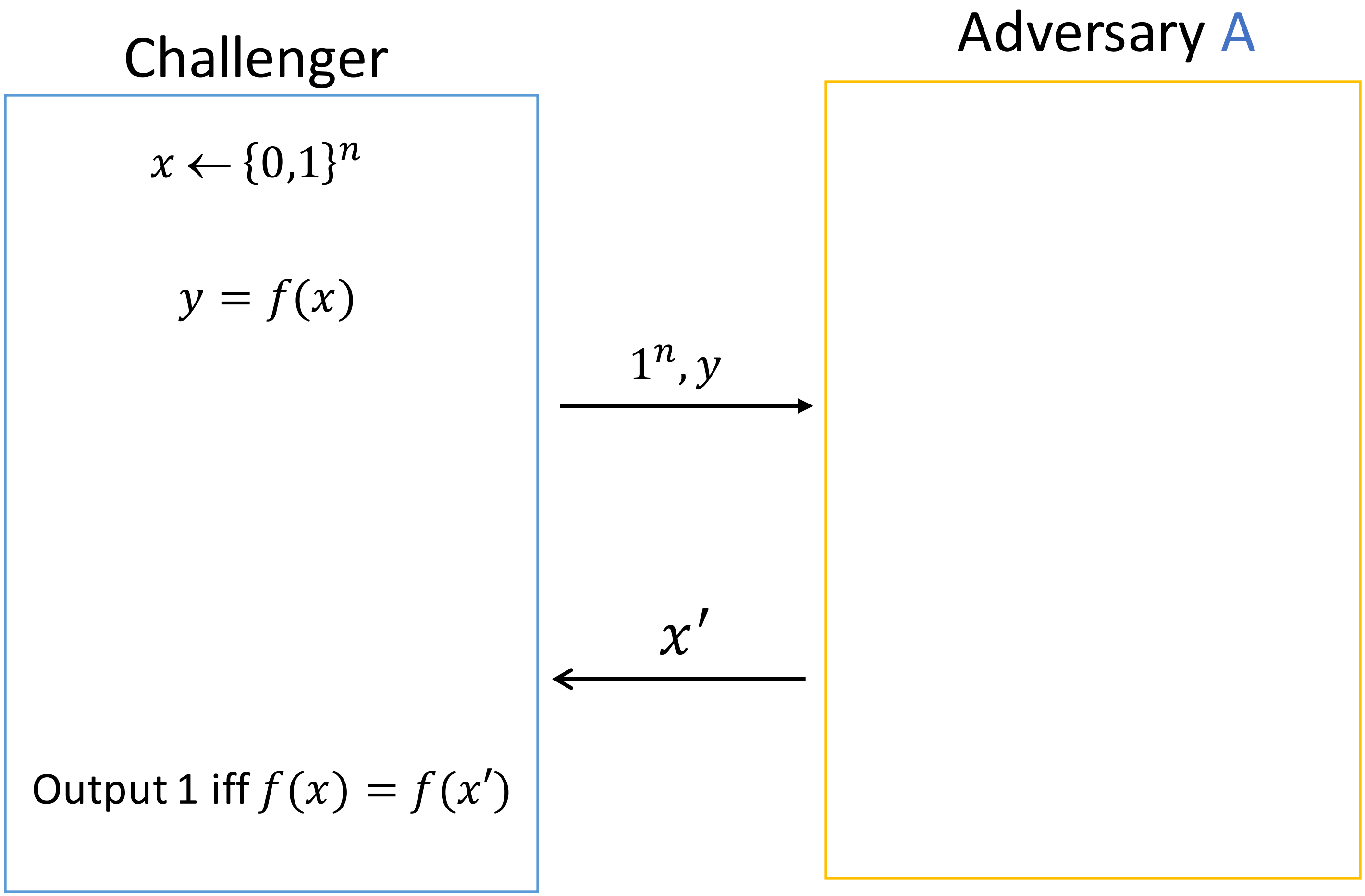


One-Way Functions: Formally

- A function $f\colon \{0,1\}^* \to \{0,1\}^*$ is a one-way function if:
- (easy to compute) There exists a polynomial-time algorithm M_f computing f; i.e., for all x, $M_f(\mathrm{x}) = f(x)$.
- (hard to invert) For all PPT A, there is a negligible function $negl$ such that

$$\Pr_{x \leftarrow \{0,1\}^n}\left[A\left(1^n, f(x)\right) \in f^{-1}\left(f(x)\right)\right] \leq negl(n)$$

One-Way Functions (Pictorially)



Is g a OFW?

- Given: f is a OWF
- $g(x) = \begin{cases} f(x) & \text{if } x \neq 0^n \\ x & \text{otherwise} \end{cases}$

- Yes, because $x = 0^n$ with negligible probability

Candidate One-Way Functions

- Factoring Based

$$f_{mult}(x, y) = x \cdot y$$

where x and y are two equal length primes.

- Subset-sum Based

$$f_{SS}(x_1, \dots, x_n, J) = (x_1, \dots x_n, [\sum_{j \in J} x_j \bmod 2^n])$$

- Discrete-Log Based:

$$f_{p,g}(x) = [g^x \, mod \; p]$$

where p is large prime and (a special value) $g \in \{2, \dots p - 1\}$

Thank You!