

# Foundation of Cryptography, Lecture 1

## One-Way Functions

### Handout Mode

Benny Applebaum & Iftach Haitner, Tel Aviv University

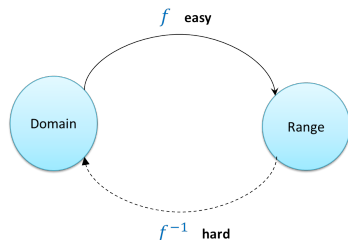
Tel Aviv University.

November 03, 2016

# Section 1

## One-Way Functions

## Informal discussion



A one-way function (OWF) is:

- Easy to compute, **everywhere**
- Hard to invert, **on the average**
  
- Why should we care about OWFs?
- Hidden in (almost) **any** cryptographic primitive: necessary for "cryptology"
- Sufficient for many cryptographic primitives

## Formal definition

### Definition 1 (one-way functions (OWFs))

A polynomial-time computable function  $f: \{0, 1\}^* \mapsto \{0, 1\}^*$  is **one-way**, if

$$\Pr_{x \leftarrow \{0,1\}^n} [A(1^n, f(x)) \in f^{-1}(f(x))] = \text{neg}(n)$$

for any PPT  $A$ .

- **polynomial-time computable**: there exists polynomial-time algorithm  $F$ , such that  $F(x) = f(x)$  for every  $x \in \{0, 1\}^*$ .
- **neg**: a function  $\mu: \mathbb{N} \mapsto [0, 1]$  is a **negligible** function of  $n$ , denoted  $\mu(n) = \text{neg}(n)$ , if for any  $p \in \text{poly}$  there exists  $n' \in \mathbb{N}$  such that  $\mu(n) < 1/p(n)$  for **all**  $n > n'$
- $x \leftarrow \{0, 1\}^n$ :  $x$  is uniformly drawn from  $\{0, 1\}^n$
- PPT: probabilistic polynomial-time algorithm.

We typically omit  $1^n$  from the input list of  $A$

## Formal definition cont.

- 1 Is this the right definition?
  - ▶ Asymptotic
  - ▶ Efficiently computable
  - ▶ On the average
  - ▶ Only against PPT's
- 2  $\text{OWF} \implies \mathcal{P} \neq \mathcal{NP}$
- 3 Does  $\mathcal{P} \neq \mathcal{NP} \implies \text{OWF}$ ?
- 4 (most) Crypto implies OWFs
- 5 Do OWFs imply Crypto?
- 6 Where do we find them?
- 7 Non uniform OWFs

### Definition 2 (Non-uniform OWF)

A polynomial-time computable function  $f : \{0, 1\}^* \mapsto \{0, 1\}^*$  is **non-uniformly one-way**, if

$$\Pr_{x \leftarrow \{0,1\}^n} [C_n(f(x)) \in f^{-1}(f(x))] = \text{neg}(n)$$

for any polynomial-size family of circuits  $\{C_n\}_{n \in \mathbb{N}}$ .

## Length-preserving functions

### Definition 3 (length preserving functions)

A function  $f: \{0, 1\}^* \mapsto \{0, 1\}^*$  is **length preserving**, if  $|f(x)| = |x|$  for every  $x \in \{0, 1\}^*$

### Theorem 4

*Assume that OWFs exist, then there exist length-preserving OWFs.*

Proof idea: use the assumed OWF to create a length preserving one.

## Partial domain functions

### Definition 5 (Partial domain functions)

For  $m, \ell: \mathbb{N} \mapsto \mathbb{N}$ , let  $f: \{0, 1\}^{\ell(n)} \mapsto \{0, 1\}^{m(n)}$  denote a function defined over input lengths in  $\{m(n)\}_{n \in \mathbb{N}}$ , and maps strings of length  $\ell(n)$  to strings of length  $m(n)$ .

Such function is efficient, if it is poly-time computable and  $f \ell$  is polynomial time computable and bounded.

The definition of one-wayness naturally extends to such (efficient) functions.

## OWFs imply length-preserving OWFs cont.

Let  $f: \{0, 1\}^* \mapsto \{0, 1\}^*$  be a OWF, let  $p \in \text{poly}$  be a bound on its computing-time, and assume wlg. that  $p$  is monotony increasing (can we?).

### Construction 6 (the length preserving function)

Define  $g: \{0, 1\}^{p(n)+1} \mapsto \{0, 1\}^{p(n)+1}$  as

$$g(x) = f(x_1, \dots, x_n), 1, 0^{p(n) - |f(x_1, \dots, x_n)|}$$

Note that  $g$  is well defined, length preserving and efficient.

### Claim 7

$g$  is one-way.

How can we prove that  $g$  is one-way?

Answer: using reduction.



## Proving that $g$ is one-way

Proof: Assume that  $g$  is **not** one-way. Namely, there exists PPT  $A$ ,  $q \in \text{poly}$  and **infinite** set  $\mathcal{I} \subseteq \{p(n) + 1 : n \in \mathbb{N}\}$ , with

$$\Pr_{x \leftarrow \{0,1\}^{n'}} \left[ A(1^{n'}, y) \in g^{-1}(g(x)) \right] > 1/q(n') \quad (1)$$

for every  $n' \in \mathcal{I}$ .

We show how to use  $A$  for inverting  $f$ .

### Claim 8

$$w \in g^{-1}(y, 1, 0^{\rho(n)-|y|}) \implies w_{1,\dots,n} \in f^{-1}(y)$$

Proof: Since  $g(w) = f(w_{1,\dots,n}), 1, 0^{\rho(n)-|f(w_{1,\dots,n})|} = y, 1, 0^{\rho(n)-|y|}$ ,

it follows that  $f(w_{1,\dots,n}) = y$  (?).  $\square$

### Algorithm 9 (Inverter B for $f$ )

Input:  $1^n$  and  $y \in \{0, 1\}^*$

- 1 Let  $x = A(1^{\rho(n)+1}, y, 1, 0^{\rho(n)-|y|})$
- 2 Return  $x_{1,\dots,n}$

### Claim 10

Let  $\mathcal{I}' := \{n \in \mathbb{N} : \rho(n) + 1 \in \mathcal{I}\}$ . Then

- 1  $\mathcal{I}'$  is infinite
- 2  $\Pr_{x \leftarrow \{0,1\}^n} [B(1^n, f(x)) \in f^{-1}(f(x))] > 1/q(\rho(n) + 1)$  for every  $n \in \mathcal{I}'$

This contradicts the assumed one-wayness of  $f$ .  $\square$

Proof: (1) is clear, (2)

$$\begin{aligned} & \Pr_{x \leftarrow \{0,1\}^n} [B(1^n, f(x)) \in f^{-1}(f(x))] \\ &= \Pr_{x \leftarrow \{0,1\}^n} [A(1^{\rho(n)+1}, f(x), 0^{\rho(n)-n})_{1,\dots,n} \in f^{-1}(f(x))] \\ &= \Pr_{x' \leftarrow \{0,1\}^{\rho(n)+1}} [A(1^{\rho(n)+1}, g(x'))_{1,\dots,n} \in f^{-1}(f(x'_{1,\dots,n}))] \\ &\geq \Pr_{x' \leftarrow \{0,1\}^{\rho(n)+1}} [A(1^{\rho(n)+1}, g(x')) \in g^{-1}(g(x'))] \geq 1/q(\rho(n) + 1) \end{aligned}$$

# From partial-domain length-preserving OWFs to length-preserving OWFs

## Construction 11

Given a function  $f: \{0, 1\}^{\ell(n)} \mapsto \{0, 1\}^{\ell(n)}$ , define  $f_{\text{all}}: \{0, 1\}^n \mapsto \{0, 1\}^n$  as

$$f_{\text{all}}(x) = f(x_{1,\dots,k}), 0^{n-k}$$

where  $n = |x|$  and  $k := \max\{\ell(n') \leq n: n' \in [n]\}$ .

Clearly,  $f_{\text{all}}$  is length preserving, defined for **every** input length, and efficient if  $f$  is.

## Claim 12

Assume  $f$  is efficient,  $f$  is one-way, and  $\ell$  satisfies  $1 \leq \frac{\ell(n+1)}{\ell(n)} \leq p(n)$  for some  $p \in \text{poly}$ , then  $f_{\text{all}}$  is one-way function.

Proof: ?

We conclude that the existence of OWF implies the existence of length-preserving OWF that is defined over all input lengths.

## Few remarks

More “security-preserving” reductions exists.

### Convention for rest of the talk

Let  $f: \{0, 1\}^n \mapsto \{0, 1\}^n$  be a one-way function.

## Weak one-way functions

### Definition 13 (weak one-way functions)

A poly-time computable function  $f: \{0, 1\}^* \mapsto \{0, 1\}^*$  is  $\alpha$ -one-way, if

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

for any PPT  $A$  and large enough  $n \in \mathbb{N}$ .

- 1 (strong) OWF according to Definition 1, are neg-one-way according to the above definition
- 2 Can we “amplify” weak OWF to strong ones?

## Strong to weak OWFs

### Claim 14

Assume there exists OWFs, then there exist functions that are  $\frac{2}{3}$ -one-way, but **not** (strong) one-way

Proof: For a OWF  $f$ , let

$$g(x) = \begin{cases} (1, f(x)), & x_1 = 1; \\ 0, & \text{otherwise } (x_1 = 0). \end{cases}$$

## Weak to strong OWFs

### Theorem 15 (weak to strong OWFs (Yao))

Assume there exist  $(1 - \delta)$ -weak OWFs with  $\delta(n) \geq 1/q(n)$  for some  $q \in \text{poly}$ , then there exist (strong) one-way functions.

- Idea: parallel repetition (i.e., direct product): Consider  $g(x_1, \dots, x_t) = f(x_1), \dots, f(x_t)$  for large enough  $t$
- Motivation: if something is somewhat hard, than doing it many times is (very) hard
- But, is it really so?

Consider matrix multiplication: Let  $A \in \mathbb{R}^{n \times n}$  and  $x \in \mathbb{R}^n$

Computing  $Ax$  takes  $\Theta(n^2)$  times, but computing  $A(x_1, x_2, \dots, x_n)$  takes ... only  $O(n^{2.3\dots}) < \Theta(n^3)$

- Fortunately, parallel repetition **does** amplify weak OWFs :-)

## Amplification via parallel repetition

### Theorem 16

Let  $f: \{0, 1\}^n \mapsto \{0, 1\}^n$  be a  $(1 - \delta)$ -weak OWF for  $\delta(n) = 1/q(n)$  for some (positive)  $q \in \text{poly}$ , and let  $t(n) = \left\lceil \frac{\log^2 n}{\delta(n)} \right\rceil$ . Then  $g: (\{0, 1\}^n)^{t(n)} \mapsto (\{0, 1\}^n)^{t(n)}$  defined by  $g(x_1, \dots, x_{t(n)}) = f(x_1), \dots, f(x_{t(n)})$ , is a one-way function.

Clearly  $g$  is efficient. Is it one-way? Proof via **reduction**: Assume  $\exists$  PPT  $A$  violating the one-wayness of  $g$ , we show there exists a PPT  $B$  violating the weak hardness of  $f$ .

*Difficulty*: We need to use an inverter for  $g$  with **low** success probability, e.g.,  $\frac{1}{n}$ , to get an inverter for  $f$  with **high** success probability, e.g.,  $\frac{1}{2}$  or even  $1 - \frac{1}{n}$

In the following we fix (an assumed) PPT  $A$ ,  $p \in \text{poly}$  and infinite set  $\mathcal{I} \subseteq \mathbb{N}$  s.t.

$$\Pr_{w \leftarrow \{0,1\}^{t(n) \cdot n}} [A(g(w)) \in g^{-1}(g(w))] \geq 1/p(n)$$

for every  $n \in \mathcal{I}$ . We also “fix”  $n \in \mathcal{I}$  and omit it from the notation.



## Proving that $g$ is One-Way – the Naive approach

Assume  $A$  attacks each of the  $t$  outputs of  $g$  **independently**:  $\exists$  PPT  $A'$  such that  $A(z_1, \dots, z_t) = A'(z_1) \dots, A'(z_t)$

It follows that  $A'$  inverts  $f$  with probability **greater** than  $(1 - \delta)$ .

Otherwise

$$\begin{aligned} \Pr_{w \leftarrow \{0,1\}^{t \cdot n}} [A(g(w)) \in g^{-1}(g(w))] &= \prod_{i=1}^t \Pr_{x \leftarrow \{0,1\}^n} [A'(f(x)) \in f^{-1}(f(x))] \\ &\leq (1 - \delta)^t \leq e^{-\log^2 n} \leq n^{-\log n} \end{aligned}$$

Hence  $A'$  violates the weak hardness of  $f$

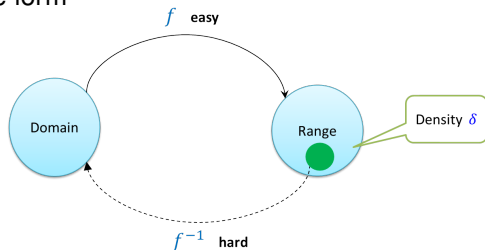
A less naive approach would be to assume that  $A$  goes over the inputs **sequentially**.

Unfortunately, we can assume **none** of the above.

Any idea?

## Hardcore sets

Assume  $f$  is of the form



### Definition 17 (hardcore sets)

$\mathcal{S} = \{\mathcal{S}_n \subseteq \{0, 1\}^n\}$  is a  $\delta$ -hardcore set for  $f: \{0, 1\}^n \mapsto \{0, 1\}^n$ , if:

- 1  $\Pr_{x \leftarrow \{0,1\}^n} [f(x) \in \mathcal{S}] \geq \delta(n)$  for large enough  $n$ , and
- 2 For any PPT  $A$  and  $q \in \text{poly}$ : for large enough  $n$ , it holds that  $\Pr [A(y) \in f^{-1}(y)] \leq \frac{1}{q(n)}$  for every  $y \in \mathcal{S}_n$ .

Assuming  $f$  has such a  $\delta$ -HC set seems like a good starting point :-)

Unfortunately, we do not know how to prove that  $f$  has hardcore set :-<

## Failing sets

### Definition 18 (failing sets)

$f: \{0, 1\}^n \mapsto \{0, 1\}^n$  has a  $\delta$ -failing set for a pair  $(A, q)$  of algorithm and polynomial, if **exists**  $\mathcal{S} = \{\mathcal{S}_n \subseteq \{0, 1\}^n\}$ , such that the following holds for large enough  $n$ :

- 1  $\Pr_{x \leftarrow \{0,1\}^n} [f(x) \in \mathcal{S}_n] \geq \delta(n)$ , and
- 2  $\Pr [A(y) \in f^{-1}(y)] \leq 1/q(n)$ , for **every**  $y \in \mathcal{S}_n$

### Claim 19

Let  $f$  be a  $(1 - \delta)$ -OWF, then  $f$  has a  $\delta/2$ -failing set, for **any** pair of PPT  $A$  and  $q \in \text{poly}$ .

Proof: Assume  $\exists$  PPT  $A$  and  $q \in \text{poly}$ , such that for any  $\mathcal{S} = \{\mathcal{S}_n \subseteq \{0, 1\}^n\}$  at **least** one of the following holds:

- 1  $\Pr_{x \leftarrow \{0,1\}^n} [f(x) \in \mathcal{S}_n] < \delta(n)/2$  for infinitely many  $n$ 's, or
- 2 For infinitely many  $n$ 's:  $\exists y \in \mathcal{S}_n$  with  $\Pr [A(y) \in f^{-1}(y)] \geq 1/q(n)$ .

We'll use  $A$  to contradict the hardness of  $f$ .

## Using $A$ to invert $f$

For  $n \in \mathbb{N}$ , let  $\mathcal{S}_n := \{y \in \{0, 1\}^n : \Pr[A(y) \in f^{-1}(y)] < 1/q(n)\}$ .

### Claim 20

$\exists$  infinite  $\mathcal{I} \subseteq \mathbb{N}$  with  $\Pr_{x \leftarrow \{0,1\}^n} [f(x) \in \mathcal{S}_n] < \delta(n)/2$  for every  $n \in \mathcal{I}$ .

### Algorithm 21 (The inverter $B$ on input $y \in \{0, 1\}^n$ )

Do (with fresh randomness) for  $n \cdot q(n)$  times:

If  $x = A(y) \in f^{-1}(y)$ , return  $x$

Clearly,  $B$  is a PPT

### Claim 22

For  $n \in \mathcal{I}$ , it holds that  $\Pr_{x \leftarrow \{0,1\}^n} [B(f(x)) \in f^{-1}(f(x))] > 1 - \frac{\delta(n)}{2} - 2^{-n}$

Proof: ?

Hence, for large enough  $n \in \mathcal{I}$ :  $\Pr_{x \leftarrow \{0,1\}^n} [B(f(x)) \in f^{-1}(f(x))] > 1 - \delta(n)$ .

Namely,  $f$  is **not**  $(1 - \delta)$ -one-way  $\square$

$g$  is **not** one-way  $\implies f$  has **no**  $\delta/2$  failing set

We show:  $g$  is **not** one way  $\implies f$  has **no**  $\delta/2$  failing-set for some PPT  $B$  and  $q \in \text{poly}$ .

### Claim 23

Assume  $\exists$  PPT  $A$ ,  $p \in \text{poly}$  and an infinite set  $\mathcal{I} \subseteq \mathbb{N}$  such that

$$\Pr_{w \leftarrow \{0,1\}^{t(n) \cdot n}} [A(g(x)) \in g^{-1}(g(w))] \geq \frac{1}{p(n)}$$

for every  $n \in \mathcal{I}$ . Then  $\exists$  PPT  $B$  such that

$$\Pr_{x \leftarrow \{0,1\}^n | y=f(x) \in \mathcal{S}_n} [B(y) \in f^{-1}(y)] \geq \frac{1}{t(n)p(n)} - n^{-\log n}$$

for every  $n \in \mathcal{I}$  and **every**  $\mathcal{S}_n \subseteq \{0,1\}^n$  with  $\Pr_{x \leftarrow \{0,1\}^n} [f(x) \in \mathcal{S}_n] \geq \delta(n)/2$ .

Fix  $\mathcal{S} = \{\mathcal{S}_n \subseteq \{0,1\}^n\}$ . By **Claim 23**, for every  $n \in \mathcal{I}$ , either

- $\Pr_{x \leftarrow \{0,1\}^n} [f(x) \in \mathcal{S}_n] < \delta(n)/2$ , or
- $\Pr_{x \leftarrow \{0,1\}^n | y=f(x) \in \mathcal{S}_n} [B(y) \in f^{-1}(y)] \geq \frac{1}{t(n)p(n)} - n^{-\log n}$   
(for large enough  $n$ )  
 $\geq \frac{1}{2t(n)p(n)}$   
(for large enough  $n$ )  
 $\implies \exists y \in \mathcal{S}_n: \Pr [B(y) \in f^{-1}(y)] \geq \frac{1}{2t(n)p(n)}$ .

## The no failing-set algorithm

### Algorithm 24 (Inverter B on input $y \in \{0, 1\}^n$ )

- 1 Choose  $w \leftarrow (\{0, 1\}^n)^{t(n)}$ ,  $z = (z_1, \dots, z_t) = g(w)$  and  $i \leftarrow [t]$
- 2 Set  $z' = (z_1, \dots, z_{i-1}, y, z_{i+1}, \dots, z_t)$
- 3 Return  $A(z')_i$

Fix  $n \in \mathcal{I}$  and a set  $\mathcal{S}_n \subseteq \{0, 1\}^n$  with  $\Pr_{x \leftarrow \{0, 1\}^n} [f(x) \in \mathcal{S}] \geq \delta(n)/2$ .

### Claim 25

$$\Pr_{x \leftarrow \{0, 1\}^n | y = f(x) \in \mathcal{S}_n} [\mathbf{B}(y) \in f^{-1}(y)] \geq \frac{1}{t(n) \cdot p(n)} - n^{-\log n}$$

**Proving**  $\Pr_{x \leftarrow \{0,1\}^n | y=f(x) \in \mathcal{S}_n} [\mathbf{B}(y) \in f^{-1}(y)] \geq \frac{1}{t(n) \cdot p(n)} - n^{-\log n}$

**Algorithm 26 (Inverter B on input  $y \in \{0, 1\}^n$ )**

- 1 Choose  $w \leftarrow (\{0, 1\}^n)^{t(n)}$ ,  $z = (z_1, \dots, z_t) = g(w)$  and  $i \leftarrow [t]$
- 2 Set  $z' = (z_1, \dots, z_{i-1}, y, z_{i+1}, \dots, z_t)$
- 3 Return  $\mathbf{A}(z')$

- For  $\mathit{Typ} = \{v \in \{0, 1\}^{t \cdot n} : \exists i \in [t] : v_i \in \mathcal{S}_n\}$ , it holds  $\Pr_z [\mathit{Typ}] \geq 1 - n^{-\log n}$
- $\forall \mathcal{L} \subseteq \{0, 1\}^{t(n) \cdot n}$  :  

$$\Pr_z [\mathcal{L}' = \mathcal{L} \cap \mathit{Typ}] = \sum_{\ell \in \mathcal{L}'} \Pr[z = \ell] \leq \sum_{\ell \in \mathcal{L}'} \frac{\Pr[z' = \ell]}{t} = \frac{\Pr_{z'} [\mathcal{L}']}{t}.$$
- Hence  $\forall \mathcal{L} \subseteq \{0, 1\}^{t(n) \cdot n} : \Pr_{z'} [\mathcal{L}] \geq \frac{\Pr_z [\mathcal{L} \cap \mathit{Typ}]}{t(n)} \geq \frac{\Pr_z [\mathcal{L}] - n^{-\log n}}{t(n)}$ .
- Assume  $\mathbf{A}$  is *deterministic* and let  $\mathcal{L}_{\mathbf{A}} = \{v \in \{0, 1\}^{t \cdot n} : \mathbf{A}(v) \in g^{-1}(y)\}$ .

$$\begin{aligned} \Pr_{x \leftarrow \{0,1\}^n | y=f(x) \in \mathcal{S}_n} [\mathbf{B}(y) \in f^{-1}(y)] &\geq \Pr[z' \in \mathcal{L}_{\mathbf{A}}] \geq \frac{\Pr[z \in \mathcal{L}_{\mathbf{A}}] - n^{-\log n}}{t(n)} \\ &\geq \frac{1}{t(n) \cdot p(n)} - n^{-\log n} \end{aligned}$$

**Proving**  $\Pr_{x \leftarrow \{0,1\}^n | y=f(x) \in \mathcal{S}_n} [\mathbf{B}(y) \in f^{-1}(y)] \geq \frac{1}{t(n) \cdot p(n)} - n^{-\log n}$ , **cont.**

In the case that  $\mathbf{A}$  is randomized, let

- $\mathbf{A}_r$  —  $\mathbf{A}$  whose coins fixed to  $r$
- $\alpha_r(n)$  — the inversion probability of  $\mathbf{A}_r$

Note that  $\mathbb{E}_r [\alpha_r(n)] \geq 1/p(n)$ .

It follows that

$$\begin{aligned} \Pr_{x \leftarrow \{0,1\}^n | y=f(x) \in \mathcal{S}_n} [\mathbf{B}(y) \in f^{-1}(y)] &\geq \mathbb{E}_r \left[ \frac{\alpha_r(n)}{t(n)} - n^{-\log n} \right] \\ &= \mathbb{E}_r [\alpha_r(n)] / t(n) - n^{-\log n} \\ &\geq \frac{1}{t(n) \cdot p(n)} - n^{-\log n}. \end{aligned}$$



## Closing remarks

- Weak OWFs can be **amplified** into strong one
- Can we give a more security preserving amplification?
- Similar hardness amplification theorems for other cryptographic primitives (e.g., Captchas, general protocols)?
- What properties of the weak OWFs have we used in the proof?