### Lecture 19: Candidate One-way Functions

Candidate OWF

▲御 ▶ ▲ 臣 ▶

### Intuition: OWF

A function  $f: \{0,1\}^n \to \{0,1\}^n$  is a one-way function if

- **()** The function f is easy to evaluate, and
- **2** The function f is difficult is hard to invert
  - We believe certain functions are one-way functions
  - If P = NP then one-way functions cannot exist (see appendix).
    So, proving that a particular function *f* is a one-way function will demonstrate that P ≠ NP, which we believe is a very difficult problem to resolve
  - So, based on our current knowledge in mathematics, we have invested faith in believing that a few specially designed functions are one-way functions

## First Candidate: Discrete Log is Hard

- Let  $(G, \times)$  be a group and g be a generator. That is,  $G = \{g^0, g^1, g^2, \dots, g^{K-1}\}$
- Let  $f \colon \{0, \dots, K-1\} \to G$  be defined as follows

$$f(x) = g^x$$

- Think: Why is this function efficient to evaluate?
- It is believed that there exists group G where f is hard to invert
- Clarification: We are not saying that f is hard to invert in any group G. There are special groups G where f is believed to be hard to invert
- Note that the inversion problem asks you to find the "logarithm," given y find x such that g<sup>x</sup> = y. This is known as the discrete logarithm problem

◆□→ ◆□→ ◆三→ ◆三→

- Let p and q be n-bit prime numbers
- Let N = pq
- Rabin's function is defined as follows

$$f(x) = x^2 \mod N$$

- Think: Why is this function efficient to evaluate?
- It is believed that finding square-roots mod N is hard when N is the product of two large primes
- Think: How can you invert Rabin's function if you know the factorization of N. That is, given p and q, how can you efficiently compute x' such that  $(x')^2 \mod N = y$ , where  $y = x^2 \mod N$

- Let P<sub>n</sub> be the set of prime numbers that require n-bit for their binary representation (i.e., the primes in the range {2<sup>n-1</sup>,...,2<sup>n</sup> − 1}). For example, P<sub>4</sub> = {11,13}
- Consider the function  $f: \mathcal{P}_n \times \mathcal{P}_n \to \mathbb{N}$

$$f(x,y)=xy$$

- Think: Why is this function efficient to compute?
- Assuming that the factorization of product of large prime numbers is difficult, this function is hard to invert

## Fourth Candidate: Elliptic Curves

- Elliptic curves are sets of pairs of elements x, y in a field that satisfy the equation  $y = x^3 + ax + b$ , for some suitably chosen values of a, b
- There is a definition of "point addition" over an elliptic curve, i.e., given two points P and Q on the curve, we can suitably define a point P + Q on the curve
- Given a point *P* on the elliptic curve, we can add x-times

 $\overrightarrow{P+P+\cdots+P}$  and represent the resulting point as xP

• Then the following function is believed to be one-way for suitable elliptic curves

$$f(x,P)=(P,xP)$$

• Think: Can you connect this assumption to the discrete log problem?

### Definition

A function  $f: \{0,1\}^n \to \{0,1\}^n$  is a one-way permutation if it is a one-way function and the function f is a bijection

We introduce this primitive because the construction of pseudorandom generators from one-way permutations is significantly more intuitive than the construction of pseudorandom generators from OWF

A (1) > A (2) > A

# Appendix: Efficient Inversion of Efficiently Computable Functions I

#### We shall show the following result

#### Theorem

Let  $f: \{0,1\}^n \to \{0,1\}^n$  be a function that can be efficiently computed. If P = NP then there exists an efficient algorithm to find an inverse x' of y, where y = f(x) for some  $x \in \{0,1\}^n$ 

Before we begin the proof of the theorem, let me emphasize that there is always an inefficient algorithm to find x', an inverse of y

Invert-Ineffcient (y):

**)** For 
$$x' \in \{0,1\}^n$$
: If  $f(x') == y$ , then return  $x'$ 

2 Return -1

This is an inefficient algorithm to compute an inverse of y = f(x)

< □ > < □ > < □ >

Let us prove the theorem now. First, let us introduce a few notations.

- Recall  $f: \{0,1\}^n \to \{0,1\}^n$  is the function
- Let φ(x) be a 3-SAT formula that tests whether f(x) = y or not. That is, φ(x) evaluates to true if and only if f(x) = y.
- If f can be evaluated in polynomial time, then the size of  $\varphi(x)$  is polynomial in n
- If P = NP then we can efficiently determine: Is φ(x) satisfiable or not

# Appendix: Efficient Inversion of Efficiently Computable Functions IV

Let us introduce the notion of a partial assignment of variables  $\{x_1, x_2, \ldots, x_n\}$ 

• Consider the following example.

$$\varphi(x) = (x_1 \lor x_2 \lor \neg x_3) \land (\neg x_1 \lor x_2 \lor x_3)$$

 The formula "φ(x) under the restriction x<sub>i</sub> → b," is obtained by substituting b as the value of x<sub>i</sub> in the formula φ(x) and simplifying. For example, "φ(x) under the restriction x<sub>1</sub> → 0" is the following formula

$$egin{aligned} arphi(x)|_{x_1\mapsto 0} &= (0 \lor x_2 \lor \lnot x_3) \land (\lnot 0 \lor x_2 \lor x_3) \ &= (0 \lor x_2 \lor \lnot x_3) \land (1 \lor x_2 \lor x_3) \ &= (x_2 \lor \lnot x_3) \end{aligned}$$

Candidate OWF

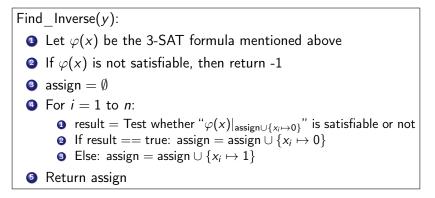
# Appendix: Efficient Inversion of Efficiently Computable Functions V

- Given a set of partial assignments assign = { $x_{i_1} \mapsto b_1, x_{i_2} \mapsto b_2, \dots, x_{i_k} \mapsto b_k$ }, we define  $\varphi(x)|_{\text{assign}}$  by setting the values of  $x_{i_1}, \dots, x_{i_k}$  as  $b_1, \dots, b_k$  in  $\varphi(x)$  and simplifying
- Again, if P = NP and f is efficiently computable, then it is efficient to find whether  $\varphi(x)|_{assign}$  is satisfiable or not

・ロト ・ 日本・ ・ 日本・

# Appendix: Efficient Inversion of Efficiently Computable Functions VI

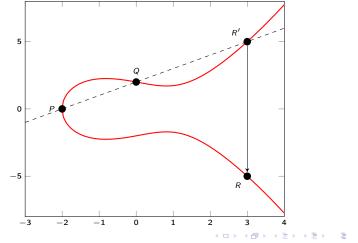
Now consider the following algorithm. We will construct a solution  $x_1x_2...x_n$  such that  $f(x_1x_2...x_n) = y$  one bit at a time.



Note that this is an efficient algorithm to compute an inverse of y if f can be computed efficiently and P = NP

### Appendix: Defining Addition on Elliptic Curves

- Onsider the field (ℝ, +, ×)
- Let us consider the plot of the curve  $y^2 = x^3 + ax + b$  (in this example, we have a = -2 and b = 4)
- Given two points P and Q on the curve, draw the line through them and find R', the third intersection point of the line with the curve
- Reflect R' on the X-axis to obtain the point R
- We define the point R as the sum P + Q



Candidate OWF