18733: Applied Cryptography Recitation

Number Theoretic Symmetric Key Constructions

Gihyuk Ko

February 24, 2017

Carnegie Mellon University

Number Theory

Number theory has provided very elegant solutions to many questions in cryptography

- Pseudorandom Generators
- One-way functions
- Public Key Cryptography

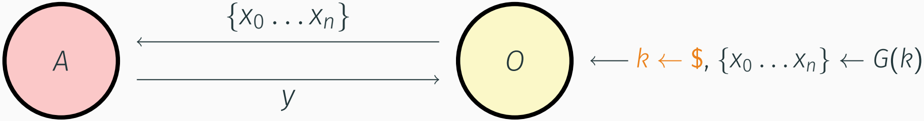
Today: a few examples of number theoretic constructions in symmetric cryptography

- Pseudorandom Generators
- Carter-Wegman MAC
- Provable Compression Functions

Pseudorandom Generators

Pseudorandom Generators: $G : K \to \{0,1\}^*$

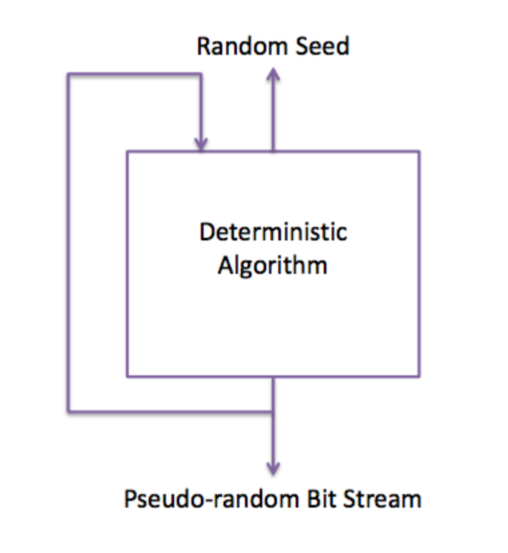
Security of Pseudorandom Generators: Unpredictability of Next Bit



Secure PRG: for all PPT algorithms A

$$Adv_{PRG}(A, G) = \Pr\left[A(\{x_0 \ldots x_n\}) = x_{n+1}\right] < \frac{1}{2} + \epsilon$$

Practical Pseudorandom Generators



- Practical pseudorandom generators are deterministic algorithms
- Seed is picked from state of the machine (e.g., temperature, time, etc) which is considered random
- Previous output(X_n) is used to generate next output(X_{n+1})

How can we make next bit unpredictable from a deterministic algorithm?

Linear Congruential Generators

Number streams calculated from linear equations of the form:

Linear Congruential Generators

$X_{n+1} = aX_n + c \pmod{m}$

- m: modulus ($m > 0$)
- a: multiplier ($0 < a < m$)
- c: the increment ($0 \leq c < m$)
- X_0: starting value, or seed ($0 \leq X_0 \leq m$)

Is this PRG always secure? No! $a = c = 1$: predictable sequence!

Selection of a, c, m is critical!

One example value: $m = 2^{31} - 1$ and $a = 1103515245, c = 12345$ (glibc)

Blum-Blum-Shub Generator

Proposed by Lenore **Blum**, Manuel **Blum**, and Michael **Shub**

Has the strongest public proof for its strength (CSPRNG)

Blum-Blum-Shub Generator

$X_{n+1} = X_n^2 \pmod{m}$

- $m = pq$: modulus where p, q are large primes ($m > 0$)
- $X_0 = s^2 \pmod{m}$: starting value, or seed ($0 \leq s \leq m$)

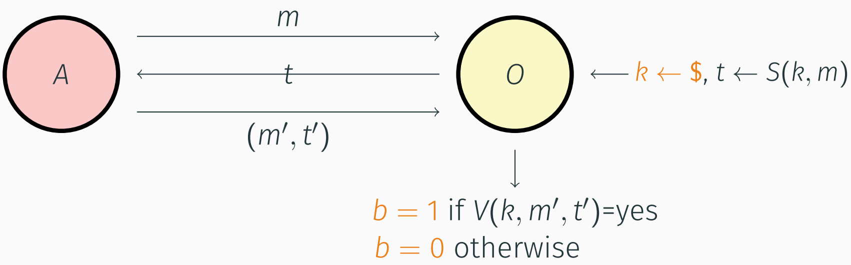
Is this PRG secure? Yes!

Security of this reduces to hardness of quadratic residue problem

In proof: Chinese Remainder Theorem, Quadratic residues, Legendre/Jacobi symbols, and more (Security proof of BBS).

MAC: $I = (S, V)$ where S: MAC signing, V: MAC verification.

Security of One-time MAC: analogous to security of one-time pad



Secure One-time MAC: for all PPT algorithms A

$$Adv_{PRG}(A, I) = \Pr[O(A, I) = 1] < \epsilon$$

Carter-Wegman MAC: Example of One-time MAC

Can be secure against *all* adversaries and faster than PRF-based MACs (HMAC, NMAC CBC-MAC, etc)

One-time MAC

$S(k, m) = P_m(a) + b \pmod{q}$

- q: a large prime (e.g., $q = 2^{128} + 51$)
- $k = (a, b)$: two random ints less or equal to q ($0 < a, b \leq q$)
- $m = (m[1] \ldots m[l])$: an l-block message with a block size 128.
- $P_m(x) = x^{l+1} + m[l]x^l + \cdots + m[1]x$: a polynomial of degree $l + 1$

It can be shown that given $S(k, m)$, adversary has no information about $S(k, m')$ (proof in Blackboard's lecture note 05-integrity)

Carter-Wegman MAC

Construction from One-time MAC

Carter-Wegman MAC

$CW((k_1, k_2), m) = (r, F(k_1, r) \oplus S(k_2, m))$

- (S, V): a secure one-time MAC scheme
- $F : K_F \times \{0,1\}^n \to \{0,1\}^n$: a secure PRF
- r: a random number in $\{0,1\}^n$

Theorem: If (S, V) is a secure one-time MAC and F a secure PRF then CW is a secure MAC outputing tags in $\{0,1\}^{2n}$

Provable Compression Functions

Compression functions: $h(H, m) : \mathcal{H} \times \mathcal{M} \rightarrow \mathcal{H}$

Used in Merkle-Damgard iterated construction

Provable Compression Function

$h(H, m) = u^H \cdot v^m \pmod p$

- p: random 2000-bit prime ($p > 0$)
- u, v: random numbers less than p ($1 \leq u, v < p$)
- m, H: inputs to the compresion function

Fact: finding collision for $h(\cdot, \cdot)$ is as hard as solving `discrete-log' modulo p

Discrete Log Problem: Given $y = g^x$ and g, output x.

Questions?