Stream Ciphers
—
An Overview

Palash Sarkar

Indian Statistical Institute, Kolkata
email: palash@isical.ac.in
Classical Encryption

Adversary’s goal: To obtain message or secret key.
One Time Pad

- Let the message be a sequence of bits: $m_0, m_1, \ldots, m_\ell$
- Let $r_0, \ldots, r_\ell$ be a sequence of random bits.
- Ciphertext is $c_0, c_1, \ldots, c_\ell$ where $c_i = m_i \oplus r_i$.
- Perfect Secrecy: Given a ciphertext, the message can be any binary string of length $\ell$.
- Impractical:
  (a) Difficult to get random bits.
  (b) Sequence must be pre-distributed to both sender and receiver.
  (c) Random sequence as long as the message.
Additive Stream Ciphers

• Use a pseudorandom generator PRG to produce the sequence $r_0, \ldots, r_\ell$.

• The PRG extends a short “seed” into a long “pseudorandom” string, i.e.,
  $\text{PRG}(K) = r_0, \ldots, r_\ell$.

• The seed is the secret key $K$.

• Security depends on the design of PRG.
Security of PRG

- PRG is a broad concept.
- For cryptographic use, a PRG must be unpredictable:
  - **Next bit test:** Given an initial segment, it should not be possible to efficiently guess the next bit.
  - **Statistical Tests:** The generated pseudorandom sequence should pass all polynomial time statistical tests.
- The above notions are equivalent.
Block Ciphers

Let
\[ E : \{0, 1\}^n \times \{0, 1\}^k \rightarrow \{0, 1\}^n. \]

- For each fixed \( K \in \{0, 1\}^k \), \( E_K() \) is a permutation of \( \{0, 1\}^n \).
- \( E(K, M) \) is written as \( E_K(M) \).
- \( E_K() \) is called an \( n \)-bit block cipher.
- \( K \) is the secret key.
- Security: Pseudorandom permutation (PRP).
Stream and Block Ciphers

- **Block Ciphers:** Applies a fixed permutation to all $n$-bit blocks.
- **Stream Ciphers:**
  - Applies a time-varying map.
  - Maintains a state vector, which allows the application of a time-varying map.
- This difference is not very important.
  - Some block cipher modes of operations are actually stream ciphers.
  - Modern day stream ciphers are becoming more and more block oriented.
- The difference between PRP and PRG is more fundamental.
Attacks

General Attack Models

- Known (or chosen) plaintext attack: The adversary learns a segment of the pseudorandom sequence.
- Chosen ciphertext attack: Adversary submits ciphertexts and gets back the corresponding plaintexts.

Attack Goals

**Distinguishing Attacks:** The adversary should be able to distinguish the output of the PRG from a random string.

**Key Recovery Attacks:** The adversary has to find the “seed” of the PRG from the output of the PRG.
Based on Hard Problems

- **Blum, Blum and Shub (BBS):**
  - Let $N = pq$, where $p, q \equiv 3 \mod 4$.
  - Seed: $x_0$. Iterates $x_{i+1} = x_i^2 \mod N$.
  - Bit extraction: $b_i = \text{parity}(x_i)$.
  - Security: If determining quadratic residuacity modulo $N$ is difficult, then guessing $b_i$ is also difficult.
  - Efficiency: Slow; one squaring per bit.
  - Recent criticism: Menezes and Koblitz have argued that security guarantee is not realistic.

- Elliptic curve based construction also known.
Classical Models

- Hardware based – well studied – based on old ideas (almost four decades).
- Uses Linear Feedback Shift Registers (LFSR) and Boolean Functions.
- LFSR: Fast generation of bit sequences (satisfying a linear recurrence) in hardware.
- Boolean function: Maps an $n$-bit string to one bit.
- S-box: Maps an $n$-bit string to an $m$-bit string.
LFSR

- Let $a = a_0, a_1, a_2, \ldots$ be a bit sequence.
- Suppose $a_{t+\ell} = c_{\ell-1}a_{t+\ell-1} \oplus \cdots \oplus c_0a_t$.
- Then $a$ is called a linear recurring sequence.
- $a$ can be produced in hardware using an LFSR.
- Let $p(x) = x^\ell \oplus c_{\ell-1}x^{\ell-1} \oplus \cdots \oplus c_1x \oplus c_0$.
- Then $p(x)$ is called the characteristic polynomial of $x$.
- If $p(x)$ is primitive over $GF(2)$, then $x$ has the maximum possible period of $2^\ell - 1$. 
Two Standard Models

**Nonlinear Filter Model:** Boolean function combines outputs of different cells of a single LFSR.

**Nonlinear Combiner Model:** Boolean function combines outputs of several LFSRs.

**Filter-Combiner model** (Sarkar, Crypto 2002) combines the best features of the above two models.

- Uses linear cellular automata (CA) instead of LFSRs.
- Suitable Boolean functions are required to realize FC-model.
Filter-Combiner Model

Diagram showing the Filter-Combiner Model with CA 1, CA 2, CA k connected to a function f, and m_i, r_i, c_i as inputs and outputs.
Necessary Properties

- High linear complexity of the output sequence.
- Suitable choice of the Boolean function.
  - Balanced.
  - Correlation immune.
  - High algebraic degree.
  - High nonlinearity.
  - Resistance to algebraic attacks.
Linear Complexity

- Let $k = k_0, \ldots, k_t$ be a sequence (possibly used to encrypt a message).
- The linear complexity of $k$ is the length of a minimum length LFSR that can produce $k$.
- Berlekamp-Massey algorithm can be used to find the linear complexity and a minimal length LFSR producing $k$.
- The time complexity of the algorithm is $O(t^2)$.
- The BM algorithm was originally introduced for decoding BCH codes.
- If $k$ is a random sequence, then its expected linear complexity is $k/2$. 
Linear Complexity (contd.)

**Nonlinear-Filter Model:**
- Partial results are known – due to Rueppel.
- Depends on the choice of the Boolean function.
- For certain Boolean functions, the linear complexity can be large.

**Nonlinear-Combiner Model:**
- Let $f(x_1, \ldots, x_n)$ be the combining Boolean function.
- Let $L_1, \ldots, L_n$ be the lengths of the LFSRs which are taken to be distinct.
- Then the linear complexity is $f(L_1, \ldots, L_n)$ where the arithmetic is over the integers (Rueppel-Staffelbach).
- Upper bound on linear complexity is
Correlation Attack

Introduced by Siegenthaler.
Assume the nonlinear-combiner model.

- Let $x_1^{(i)}, \ldots, x_t^{(i)}$ be the sequence obtained from the $i$th LFSR $L_i$.
- Let $k_0, \ldots, k_t$ be the output of the Boolean function.
- Suppose $\text{Prob}[k_j = x_j^{(i)}] = \alpha_i \neq \frac{1}{2}$.
- Then the output $k_0, \ldots, k_t$ is biased with respect to the sequence $x_1^{(i)}, \ldots, x_t^{(i)}$. 
Correlation Attack (contd.)

- Assume that the sequence \( k = k_0, \ldots, k_t \) is known. (Also possible to work with the ciphertext.)
- For each of the \( 2^{L_i} - 1 \) nonzero initial states of \( L_i \), generate the sequence \( x_1^{(i)}, \ldots, x_t^{(i)} \).
- Compute \( \hat{\alpha}_i = \text{Prob}[k_j = x_j^{(i)}] \) for each such sequence.
- If the choice of initial condition is correct, then \( \hat{\alpha}_i \approx \alpha_i \) else, \( \hat{\alpha}_i \approx 1/2 \).
- If \( \alpha_i \) and \( 1/2 \) are “sufficiently” separate and \( t \) is “sufficiently” large, then we can find the correct initial state for \( L_i \).
Correlation Attack (contd.)

- If $\alpha_i = 1/2$, then this does not work for $\mathcal{L}_i$.
- If $\alpha_i = 1/2$ for all $i$, then $f()$ is said to be correlation immune (CI).
- One can then look for correlation between $k$ and XORs of several $\mathcal{L}_i$’s.
- If $f()$ is $k$th order CI, then this method will not succeed if we take the XORs of upto $k$ LFSRs.
- A balanced $k$-CI function is called $k$-resilient.
Correlation Attack (contd.)

**Fast Correlation Attack:** Avoid considering all possible initial states (Meier-Staffelbach).

- Each LFSR sequence satisfies a linear recurrence.
- Let $p(x)$ be the characteristic polynomial of the recurrence.
- Then the sequence also satisfies a recurrence given by any multiple of $p(x)$.
- Collection of all such recurrences (with “low span”) constitute a set of parity check equations.
- An iterative decoding method can be applied to obtain the original sequence from these parity check equations.
Multiples of Polynomials

- Multiples of $p(x)$ provide parity check equations.
- It is helpful for the attack if these parity check equations have few terms.
- For example, if $p(x)$ has a trinomial multiple, then the obtained parity check has three terms.
- A “good” polynomial $p(x)$ will not have low degree sparse multiples.
- Determining whether a polynomial is “good” can be difficult. Work by Wagner (Crypto 2002) based on the generalized birthday attack.
Algebraic Attack

Courtois-Meier: Consider the nonlinear-filter model, i.e., there is a single LFSR $\mathcal{L}$.

- Let $S_0$ be the initial state of the LFSR.
- The $i$th state $S_i$ is obtained as $S_i = \mathcal{L}^i(S_0)$.
- Let $f()$ be the combining Boolean function of degree $d$.
- The key bits of $k$ can be written as

$$f(\mathcal{L}^i(S_0)) = k_i.$$

- For any $i$, the map $\mathcal{L}^i()$ is a linear map.
- Thus, for any $i$, we obtain a nonlinear equation in the bits of $S_0$ of maximum degree $d$. 
Algebraic Attack (contd.)

- Linearization: Replace each monomial of degree greater than one by a new variable.
- Let \( D = \sum_{i=0}^{d} \binom{n}{i} \).
- We obtain a system of linear equations in at most \( D \) variables.
- If there are \( D \) “independent” equations, then we can solve for these \( D \) variables and use back substitution to obtain the bits of \( S_0 \).
- Time required is \( O(D^3) \).
- Becomes infeasible if \( D \) is large, which can be ensured by having \( d \) to be large.
Algebraic Attack (contd.)

- Suppose $f$ has large degree $d$, but there are low degree polynomials $g(x)$ and $h(x)$ such that $h(x) = g(x)f(x)$.

- Then a modification of the above idea can be applied to $h()$ instead of $f()$.

- If $h(x)$ is zero, then $g(x)$ is called an annihilator.

- A low degree annihilator can be used for the attack.

- Necessary condition: Neither $f$ nor $(1 \oplus f)$ should have a low degree annihilator.
Boolean Function Properties

Let $f(x_1,\ldots,x_n)$ be an $n$-variable Boolean function.

**Balancedness:** $wt(f) = 2^{n-1}$.

**k-CI:** $\text{Prob}[f(x_1,\ldots,x_n) = x_{i_1} \oplus \cdots \oplus x_{i_k}] = 1/2$
for any choice of $i_1,\ldots,i_k$ from $\{1,\ldots,n\}$.

**Nonlinearity:** A function $l$ is affine if

$$l(x_1,\ldots,x_n) = a_0 \oplus a_1 x_1 \oplus \cdots \oplus a_n x_n$$

for some bit constants $a_0, a_1,\ldots,a_n$.

$$\text{nl}(f) = \min_{\text{affine } l} d(f,l).$$

**Algebraic Degree:** Largest degree monomial in the algebraic normal form of $f$. 

stream cipher overview, Palash Sarkar – p.25/51
Let $n =$ the number of input bits; $d =$ degree; $t =$ the order of resiliency; $\mathcal{N} =$ nonlinearity.

**Known Trade-Offs:**

- $d \leq n - t - 1$.
- $\mathcal{N} \leq 2^{n-1} - 2^{t+1+\lceil(n-t-2)/d\rceil}$.

- Recent research has shown that it is possible to construct Boolean functions with optimal trade-offs.

- The question of algebraic resistance offered by such Boolean functions is still under investigation.
Exchange-Shuffle Paradigm

- Let $A$ be an array of length $N$ containing some permutation of 0 to $N - 1$.
- Repeat the following steps $N$ times.
  - Choose two random locations of $A$.
  - Exchange the values in these locations.
- At the end of $N$ steps, $A$ is expected to have a random permutation of 0 to $N - 1$.
- **RC4:** Uses the above principle.
  - Key set-up phase: Starting from a secret key, ensures a “uniform” distribution.
  - Byte generation phase: generates bytes at each step.
RC4 – Key Setup

for i from 0 to 255
S[i] := i
j := 0
for i from 0 to 255
j := (j + S[i] + key[i mod keylength]) mod 256
swap(S[i],S[j])
RC4 – Byte Generation

Initialization:
i = 0;
j = 0;

Generation loop:
i = i + 1;
j = j + A[i];
Swap(A[i], A[j]);


Note: \( A[] \) is a byte-array of length 256.
RC4: Byte Generation
RC4: Weaknesses and Extensions

- Very efficient in software. But cannot take advantage of 32-bit architecture.
- Weaknesses pointed out by Shamir, Mantin, Golić, Paul-Preneel and others.
- Attempts to extend the design to 32-bit architecture. Recent study of weaknesses by Paul-Preneel (Asiacrypt 2006).
Initialization Vector

- In a stream cipher, the same secret key cannot be used more than once.
- Regular key change is a serious problem.
- Modern day stream ciphers use an initialization vector (IV).
- IV can be known to the adversary. Strength of stream cipher does not depend on IV.
- IV need not be random or unpredictable.
- Constraint: The same (key, IV) pair cannot be used twice.
- With this constraint, for the same key, several IVs can be used. Eases the problem of supplying fresh keys.
Security Goals

Confidentiality: This is the basic goal.

Authentication: If the message has been tampered with, then it should be possible to detect it. Usually achieved by computing a tag of the message.

Authenticated Encryption: The combined goal of confidentiality plus authentication.

Self-Synchronization:
- In a noisy channel, there may be bit flips, bit slips or bit inserts.
- Suppose a contiguous sequence of bits are received correctly.
- From this point, the decryption algorithm can correctly decrypt.
BC Modes of Operations

message: $M_1, M_2, M_3, \ldots$ (n-bit blocks);
initialization vector: $n$-bit IV (used as nonce).

Electronic codebook (ECB) mode:
$C_i = E_K(M_i), \ i \geq 1.$
ECB Mode

Electronic Codebook (ECB) mode encryption
Insecurity of ECB Mode

CBC Mode

Cipher Block Chaining (CBC) Mode:

\[ C_1 = E_K(M_1 \oplus IV); \]
\[ C_i = E_K(M_i \oplus C_{i-1}), \ i \geq 2. \]
CBC Mode

\[ E_K \]

\[ P_1 \rightarrow C_1 \]

\[ P_2 \rightarrow C_2 \]

\[ \text{IV} \]

\[ E_K \]

\[ P_{m-1} \rightarrow C_{m-1} \]

\[ E_K \]

\[ P_m \rightarrow C_m \]
Modes of Operations (contd.)

Output feedback (OFB) mode:
\[ Z_1 = E_K(\text{IV}); \quad Z_i = E_K(Z_{i-1}), \quad i \geq 2; \]
\[ C_i = M_i \oplus Z_i, \quad i \geq 1. \]
- This is essentially an additive stream cipher.

Cipher feedback (CFB) mode:
\[ C_1 = M_1 \oplus E_K(\text{IV}); \]
\[ C_i = M_i \oplus E_K(C_{i-1}), \quad i \geq 2. \]
- Can be used as a self-synchronizing stream cipher in a 1-bit feedback mode.
OFB Mode

Output Feedback (OFB) mode encryption
CFB Mode

stream cipher overview, Palash Sarkar – p.41/51
Modes of Operations (contd.)

Counter (CTR) mode:

\[ C_i = M_i \oplus E_K(\text{nonce}||\text{bin}(i)), \]

\( i \geq 1. \)

- Recent proposal – Salsa20 – due to Bernstein is based on this idea.
- nonce is an \( n/2 \)-bit string.
- \( \text{bin}(i) \) is an \( n/2 \)-bit binary representation of \( i \).
- Requires an adder.
- Use of parallel generation of LFSR sequences in hardware (Mukhopadhyay-Sarkar, 2006).
Counter Mode

Counter (CTR) mode encryption
Bluetooth: E0

LFSR1

LFSR2

LFSR3

LFSR4

XOR

Keystream $Z_t$

$CT_{t+1}$

$ST_{t+1}$

$Z^{-1}$

$T_1$

$Z^{-1}$

$T_2$

$2$

$/2$

$XOR$

$+/2$

$3$

$2$

$3$

$+/2$

$+/3$

stream cipher overview, Palash Sarkar – p.44/51
Bluetooth: E0

- The key length may vary; usually 128 bits.
- At each step, E0 generates a bit using four shift registers (lengths 25, 31, 33, 39 bits) and two 2-bit internal states.
- At each step, the registers are shifted and the two states are updated with the current state, the previous state and the values in the shift registers.
- Four bits are then extracted from the shift registers and added together.
- The algorithm XORs that sum with the value in the 2-bit register.
- The first bit of the result is output for the encoding.
GSM: A5/1
GSM: A5/1

- The registers are clocked in a stop/go fashion using a majority rule.
- Each register has an associated clocking bit.
- At each cycle, the clocking bit of all three registers is examined and the majority bit is determined.
- A register is clocked if the clocking bit agrees with the majority bit.
- Hence at each step two or three registers are clocked, and each register steps with probability 3/4.
SNOW 2.0 Stream Cipher

Fig. 2. A schematic picture of SNOW 2.0
**Phelix**

![Diagram](image-url)  

**Fig. 2.** One block of Phelix encryption

*stream cipher overview, Palash Sarkar – p.49/51*
Ecrypt Stream Cipher Proposals

- Phelix.
- HC-256.
- Salsa20.
- Rabbit.
- Many others – generating a lot of interest and discussion on various aspects of stream cipher design and analysis.

- http://www.ecrypt.eu.org/stream/
Thank you for your attention!