

Encrypting Long Messages

Lecture 16: A short key, a continuing stream

Lecture overview

Alice and Bob share a short random seed. They use it to generate the same sequence of masking bits, one bit at a time.

Stateful encryption, its security proof, and why stream positions must not repeat.

Based on the instructor's Lecture 16 [Maj26].

Abstract and lecture roadmap

The one-time pad needs one secret random bit for every message bit. A pseudorandom generator lets a short seed play the role of a much longer pad, against efficient observers.

1. Recall the one-bit generator from Lecture 15.
2. Generate the pad as the message arrives.
3. Prove that Alice and Bob stay synchronized and recover the message.
4. Replace the pad by uniform bits to prove confidentiality.

Proof roadmap

One hidden bit gives a pseudorandom stream by a hybrid argument. A pseudorandom stream gives encryption by comparison with the one-time pad.

[BS23, §§3.2, 3.4.2]; background: [HPS14, §1.7.5].

Main results

Assume a one-way permutation $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$. Sample independent uniform $R, X \in \{0, 1\}^n$ and put $b_i = \langle R, f^i(X) \rangle$ modulo 2.

Pseudorandom stream

For every positive, efficiently computable polynomial length $L = L(n)$,

$$P_L(R, X) := (b_0, \dots, b_{L-1}) \approx_c U_L.$$

Here U_L is uniform; $\approx_c$ denotes computational indistinguishability.

Encryption with a short key

The $2n$ -bit key (R, X) hides an L -bit message via $c_i = m_i \oplus b_i$. Decryption is exact when the states agree; confidentiality requires that stream positions are never reused.

Hard-core input: [GL89]; sequential expansion: [BS23, Thm. 3.3].

What changes from the one-time pad?

	One-time pad	Today's construction
Random key	L uniform bits	$2n$ uniform bits
Masking string	Uniform in $\{0, 1\}^L$	Generated from a seed
Security goal	Perfect secrecy	Computational secrecy
Observers	Even unbounded	Probabilistic polynomial time

Why there is no contradiction

When $L > 2n$, at most 2^{2n} pads can occur. An unbounded observer can enumerate them. The guarantee is that efficient observers cannot exploit this small support.

[BS23, §§2.1.3, 3.1.1]; symmetric-cipher motivation: [HPS14, §1.7.5].

Recall: one extra pseudorandom bit

Let $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ be an efficiently computable permutation that is hard to invert on a uniform input. For $r, x \in \{0, 1\}^n$, define

$$h(r, x) = \langle r, x \rangle = r_1 x_1 \oplus \cdots \oplus r_n x_n.$$

The generator from Lecture 15

$$G(r, x) = (r, f(x), h(r, x)), \quad G : \{0, 1\}^{2n} \rightarrow \{0, 1\}^{2n+1}.$$

The first $2n$ bits form the next state. The last bit is emitted.

Goldreich–Levin hides $h(r, x)$ even given $(r, f(x))$. Bijectivity makes $(R, f(X))$ uniform, so $G(R, X) \approx_c U_{2n+1}$.

The final section states the imported hard-core result precisely.

[HILL99, §4.1, Prop. 4.3; §4.2, Prop. 4.4, p. 1376].

Generating one bit at a time

Write $f^0(x) = x$ and $f^{i+1}(x) = f(f^i(x))$. Starting with $s_0 = x$, repeat

$$\boxed{b_i = h(r, s_i)}, \quad \boxed{s_{i+1} = f(s_i)}.$$

Position i	Current state s_i	Emitted bit b_i	Next state
0	x	$h(r, x)$	$f(x)$
1	$f(x)$	$h(r, f(x))$	$f^2(x)$
2	$f^2(x)$	$h(r, f^2(x))$	$f^3(x)$

Small working memory

Store r and the current state. Each new bit needs one inner product and one evaluation of f .

What does a pseudorandom stream mean?

Fix a polynomial length $L = L(n)$. An efficient test D sees either

$$P_L(R, X) = (b_0, \dots, b_{L-1}) \quad \text{or} \quad U_L.$$

Its distinguishing advantage is

$$\text{Adv}_{P_L, D}(n) = |\Pr[D(P_L(R, X)) = 1] - \Pr[D(U_L) = 1]|.$$

The guarantee concerns the whole prefix

For every probabilistic polynomial-time D , this advantage is negligible: eventually smaller than n^{-a} for every constant $a > 0$.

The bits are determined by a $2n$ -bit seed, so they are *not* truly independent. The theorem concerns polynomially bounded prefixes, even though the algorithm can keep running.

[BS23, Defs. 3.1–3.2, pp. 46–48].

Why does repeating the generator work?

The stronger statement used in the proof

For independent uniform R, X , define

$$Q_L(R, X) = (R, f^L(X), b_0, \dots, b_{L-1}).$$

Then $Q_L(R, X) \approx_c U_{2n+L}$.

Proof roadmap. Replace the emitted bits by fresh fair bits, starting at position 0. Each replacement reduces to the one-bit generator. There are only L replacements.

After the replacements, the bits are independent of $(R, f^L(X))$, which is uniform because f^L is a permutation.

Discarding the first $2n$ coordinates proves $P_L(R, X) \approx_c U_L$. The appendix gives the reduction and its exact loss.

Sequential composition: [BS23, Thm. 3.3, pp. 59–61].

The shared key and the two states

Alice and Bob want to communicate $m_0, m_1, \dots$, one bit at a time.

Initialization

$$\text{Gen}(1^n) : \quad r, x \xleftarrow{\$} \{0, 1\}^n, \quad \text{sk} = (r, x).$$

Both parties receive the same key and initialize

$$\text{state}_A = x, \quad \text{state}_B = x.$$

- r remains fixed throughout this stream.
- Alice's state advances once per encrypted bit.
- Bob's state advances once per decrypted bit.
- A message boundary does not reinitialize either state.

Encryption and decryption

Alice encrypts the next bit

On input $m_i \in \{0, 1\}$:

$$b = h(r, \text{state}_A),$$

$$c_i = m_i \oplus b,$$

$$\text{state}_A \leftarrow f(\text{state}_A).$$

Send c_i to Bob.

Bob decrypts the next bit

On receiving $\tilde{c}_i \in \{0, 1\}$:

$$b' = h(r, \text{state}_B),$$

$$\tilde{m}_i = \tilde{c}_i \oplus b',$$

$$\text{state}_B \leftarrow f(\text{state}_B).$$

Output $\tilde{m}_i$.

Both parties compute the mask from the *current* state, then advance it.

For honest delivery in order, $\tilde{c}_i = c_i$ and the two masks agree. Neither party needs to invert f .

Instructor's slides, p. 4 [Maj26]; PRG encryption: [BS23, §3.2].

Correctness: the states stay aligned

Invariant at stream position i

After Alice has encrypted i bits and Bob has decrypted those same i bits in order,
$$\text{state}_A = \text{state}_B = f^i(x).$$

Proof. For $i = 0$, both states are x . If the invariant holds at position i , both masks equal $b_i = h(r, f^i(x))$, so

$$\tilde{m}_i = c_i \oplus b_i = (m_i \oplus b_i) \oplus b_i = m_i.$$

Both parties then move to $f(f^i(x)) = f^{i+1}(x)$, proving the next step.

Alice may send ahead of Bob. What matters is that the two operations associated with position i use the same state.

Correctness claim in the source slides, p. 4 [Maj26].

A worked state example

For the arithmetic only, take $n = 4$, $r = 1010$, $x = 1101$, and let f rotate the bits left by one place. Encrypt $m = 1001$.

i	s_i	$b_i = \langle 1010, s_i \rangle$	m_i	$c_i = m_i \oplus b_i$	$c_i \oplus b_i$
0	1101	1	1	0	1
1	1011	0	0	0	0
2	0111	1	0	1	0
3	1110	0	1	1	1

This example illustrates correctness only

The ciphertext is 0011 and Bob recovers 1001. Rotation is easy to invert and repeats after four steps, so this generator is insecure.

The confidentiality experiment

Fix a polynomial length L . An efficient observer A tries to distinguish encryptions of two messages of its choice.

1. A chooses $m^{(0)}, m^{(1)} \in \{0, 1\}^L$ and retains its working memory.
2. Sample a fresh key (R, X) and a fair hidden bit $\beta \in \{0, 1\}$.
3. Give $c = m^{(\beta)} \oplus P_L(R, X)$ to A .
4. A outputs a guess $\hat{\beta}$.

Security goal

$$\text{Adv}_{\text{enc}, A}(n) := \left| \Pr[\hat{\beta} = \beta] - \frac{1}{2} \right| \text{ is negligible.}$$

Equal lengths are required because this scheme exposes the message length.

Bit-guessing formulation: [BS23, §2.2.5]; stream encryption: [BS23, Thm. 3.1].

Security: replace the pad by uniform bits

Proof roadmap. Any successful observer of the ciphertext would give a successful test of the pad.

A PRG test D receiving a candidate pad $w \in \{0, 1\}^L$

Run A to obtain $m^{(0)}, m^{(1)}$. Choose a fair β , give $m^{(\beta)} \oplus w$ to A , and output 1 exactly when $\hat{\beta} = \beta$.

If $w = P_L(R, X)$, this is the real experiment. If $w = U_L$, then for each $c \in \{0, 1\}^L$,

$$\Pr[m^{(\beta)} \oplus U_L = c \mid m^{(0)}, m^{(1)}, \beta] = 2^{-L}.$$

The ciphertext is independent of β , and the success probability is $1/2$.

Therefore $\text{Adv}_{\text{enc}, A}(n) = \text{Adv}_{P_L, D}(n)$, which is negligible. □

Compare [BS23, Thm. 3.1, pp. 49–51].

Continuing across message boundaries

Suppose the first message uses positions $0, \dots, a - 1$. The next message must begin at position a , with state $f^a(x)$.

One stream can carry several messages

Partitioning a stream into messages does not change the construction. The total number of encrypted bits must remain polynomially bounded, and every position is consumed once.

The same proof permits chosen plaintexts. Replace the entire pad by U_T , where T bounds the total number of bits. Even if later messages depend on earlier ciphertexts, every unused pad bit is fresh and uniform.

An observer may submit equal-length message pairs adaptively, with one fixed hidden choice bit. Using fresh uniform pad segments makes its view independent of that bit.

Extension of the preceding reduction; chosen-plaintext framework: [BS23, §5.2].

Synchronization is part of correctness

Delivery event	Effect in the basic bit-stream construction
Delay, order kept	Bob decrypts later using the correct sequence of states.
Dropped bit	Without position information, subsequent masks are misaligned.
Extra bit	Advancing on an inserted or repeated bit shifts Bob's state.
Reordered bits	Bob applies masks to the wrong ciphertext positions.

The model used in the correctness proof

Bob processes each ciphertext bit once, in its original order. Other delivery patterns require additional protocol structure.

Synchronization condition in the instructor's slides, p. 4 [Maj26].

Reusing a stream position exposes a relation

Suppose the same pad segment z encrypts two message segments:

$$c = m \oplus z, \quad c' = m' \oplus z.$$

The pad cancels

$$c \oplus c' = m \oplus m'.$$

If m is known, then $m' = m \oplus c \oplus c'$ is fully recovered.

- Resetting to the same initial key restarts the same pad.
- Copying a running state creates two copies of the same future stream.
- For communication in both directions, use independent streams.

Reusing the *generator algorithm* is fine. Reusing its pad positions is the problem.

[BS23, §3.3.1, p. 53].

Confidentiality does not detect modifications

An observer can modify the ciphertext without learning the pad. For any chosen bit string δ , let

$$c' = c \oplus \delta.$$

Bob then obtains

$$c' \oplus z = (m \oplus z \oplus \delta) \oplus z = m \oplus \delta.$$

A chosen bit flip survives decryption

Changing ciphertext bit i changes plaintext bit i . This construction provides no test that the received ciphertext is authentic.

The theorem proves confidentiality under the stated experiment. Detecting modification requires authentication as an additional ingredient.

[BS23, §3.3.2, pp. 53–54].

The construction and its guarantees

The complete mechanism

$$\text{state} = x, \quad c_i = m_i \oplus \langle r, \text{state} \rangle, \quad \text{state} \leftarrow f(\text{state}).$$

Bob runs the same state evolution and XORs the same bit to recover m_i .

- **Key length:** $2n$ bits for a polynomially long message stream.
- **Correctness:** the same state at the same stream position.
- **Confidentiality:** one-wayness $\Rightarrow$ hidden bit $\Rightarrow$ pseudorandom pad $\Rightarrow$ encryption security.
- **Use condition:** a fresh initial seed, with no repeated positions.
- **Scope:** efficient observers, visible lengths, no authentication.

The security proof is a comparison with the ordinary one-time pad.

Questions to check your understanding

1. Why do both encryption and decryption use f , rather than one of them using f^{-1} ?
2. Alice has encrypted 100 bits, but Bob has decrypted only 80. Has correctness already failed? Which states should they hold?
3. If a new message begins, should either party reset its state?
4. Suppose an observer learns the plaintext of the first message. Why does this reveal its pad segment but not efficiently reveal fresh segments?
5. What breaks if the same initial key is used for two independent streams?
6. Where does the polynomial bound on the number of encrypted bits enter the proof?

Appendix: replacing the bits one at a time

Let $R, X \xleftarrow{\$} \{0,1\}^n$, and let $C_0, \dots, C_{L-1}$ be independent fair bits, independent of (R, X) . Put $b_i = h(R, f^i(X))$.

The hybrids

For $0 \leq j \leq L$, define

$$H_j = (R, f^L(X), C_0, \dots, C_{j-1}, b_j, \dots, b_{L-1}).$$

The random prefix is empty at $j = 0$; the generated suffix is empty at $j = L$.

Thus $H_0 = Q_L(R, X)$, whereas H_L is exactly uniform in $\{0,1\}^{2n+L}$.

Roadmap. An adjacent pair differs at only one bit. A reduction can sample the earlier fair bits and compute the later bits from the new state. It never needs to invert f .

Fix an efficient test D , and write $p_j = \Pr[D(H_j) = 1]$.

Specialization of [BS23, Thm. 3.3, pp. 59–61].

Appendix: the adjacent-hybrid reduction

Fix $j \in \{0, \dots, L-1\}$. The challenge (r, y, b) is either $G(R, Z)$, for independent uniform $R, Z \in \{0, 1\}^n$, or uniform in $\{0, 1\}^{2n+1}$.

Building the input to D

Sample $C_0, \dots, C_{j-1}$ independently. Output

$$(r, f^{L-j-1}(y), C_0, \dots, C_{j-1}, b, h(r, y), \dots, h(r, f^{L-j-2}(y))).$$

For $j = L-1$, the suffix after b is empty and the terminal state is y .

- **Real challenge:** $y = f(Z)$ and $b = h(r, Z)$. Since Z has the same uniform law as $f^j(X)$, the result has distribution H_j .
- **Uniform challenge:** r, y, b are independent and uniform. Since y has the law of $f^{j+1}(X)$, the result has distribution H_{j+1} .

Only forward evaluations of f are needed; the simulation takes polynomial time.

Appendix: the exact security loss

Choose j uniformly from $\{0, \dots, L-1\}$ and use the preceding reduction. Let B output whatever D outputs.

On real and uniform inputs, its acceptance probabilities are respectively $\frac{1}{L} \sum_{j=0}^{L-1} p_j$ and $\frac{1}{L} \sum_{j=0}^{L-1} p_{j+1}$. Their difference telescopes to $(p_0 - p_L)/L$.

A single reduction handles all hybrids

$$\text{Adv}_{Q_L, D}(n) = |p_0 - p_L| = L \text{Adv}_{G, B}(n).$$

The one-step advantage is negligible. Multiplying it by a polynomial L preserves negligibility, proving the theorem.

For a test of P_L , let D discard the first $2n$ coordinates. □

Compare [BS23, Thm. 3.3 and Eq. (3.10), pp. 59–61].

Boneh–Shoup, version 0.6 (January 2023)

- §3.2, Theorem 3.1, pp. 48–51: encryption using a PRG.
- §3.3.1, p. 53: why a pad cannot be used twice.
- §3.3.2, pp. 53–54: controlled ciphertext modifications.
- §3.4.2, Theorem 3.3, pp. 59–61: sequential expansion.

Hoffstein–Pipher–Silverman, second edition (2014)

§1.7.5, pp. 44–45: random bit sequences and symmetric ciphers.

§8.2, pp. 474–476: pseudorandom number generation.

[BS23, HPS14]. Page numbers refer to printed pages in these editions.

References I

- [BS23] Dan Boneh and Victor Shoup.
A Graduate Course in Applied Cryptography.
Author-published online textbook, 2023.
Version 0.6, January 2023.
URL: https://crypto.stanford.edu/~dabo/cryptobook/BonehShoup_0_6.pdf.
- [GL89] Oded Goldreich and Leonid A. Levin.
A hard-core predicate for all one-way functions.
In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*, pages 25–32.
ACM, 1989.
URL: <https://www.cs.bu.edu/fac/lnd/pdf/hard.pdf>, doi:10.1145/73007.73010.
- [HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby.
A pseudorandom generator from any one-way function.
SIAM Journal on Computing, 28(4):1364–1396, 1999.
URL: <https://www.cs.bu.edu/fac/lnd/pdf/hill.pdf>, doi:10.1137/S0097539793244708.

References II

- [HPS14] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman.
An Introduction to Mathematical Cryptography.
Undergraduate Texts in Mathematics. Springer, New York, second edition, 2014.
doi:10.1007/978-1-4939-1711-2.
- [Maj26] Hemanta K. Maji.
Lecture 16: Encrypting long messages.
CS355, Purdue University, Fall semester, 2026.
Instructor's supplied lecture slides, 4 pages.

Technical input: the Goldreich–Levin theorem

Imported hard-core theorem, in distribution form

Let $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ be a one-way function. For independent uniform $R, X \in \{0, 1\}^n$ and an independent fair bit C ,

$$(R, f(X), \langle R, X \rangle) \approx_c (R, f(X), C).$$

Precise citation: Håstad–Impagliazzo–Levin–Luby, §4.1, Proposition 4.3, p. 1376 [HILL99].
Original: Goldreich–Levin, §3, Theorem 1, author version p. 3 [GL89].

Assumption and use in this lecture

Assume also that f is a permutation. Then $(R, f(X), C)$ is uniform, giving the one-bit PRG [HILL99, Prop. 4.4, p. 1376]. The slides prove iteration and encryption security. The model throughout is classical probabilistic polynomial time.