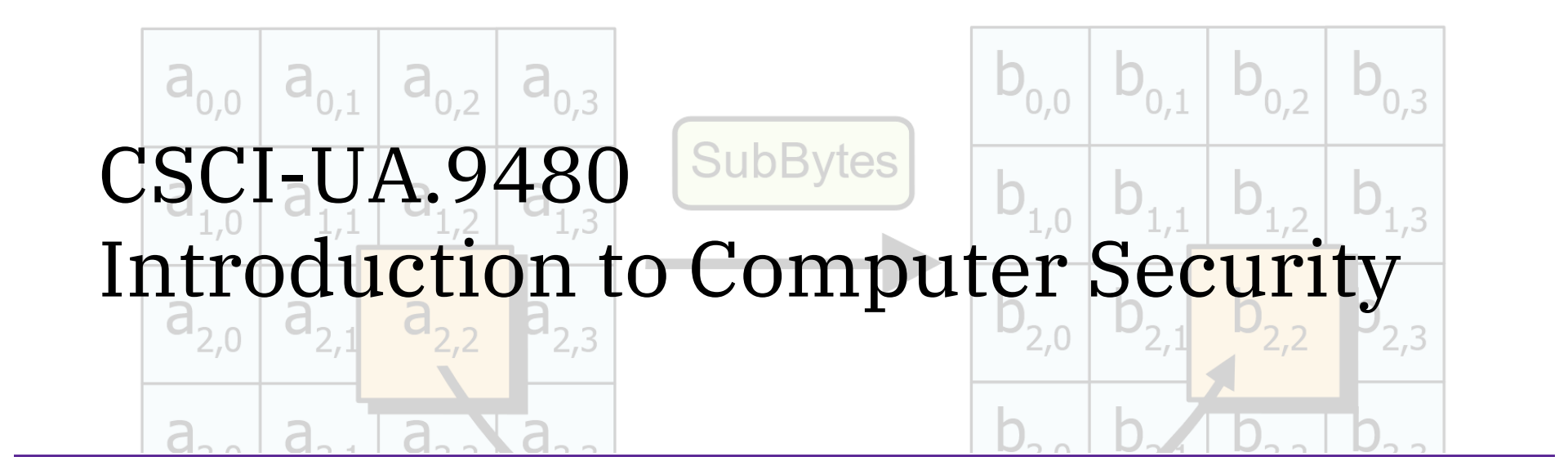


CSCI-UA.9480
Introduction to Computer Security

Session 1.2
Symmetric Key Encryption

Prof. Nadim Kobeissi

Cryptographic Security

Information Theoretical Foundation for Security.

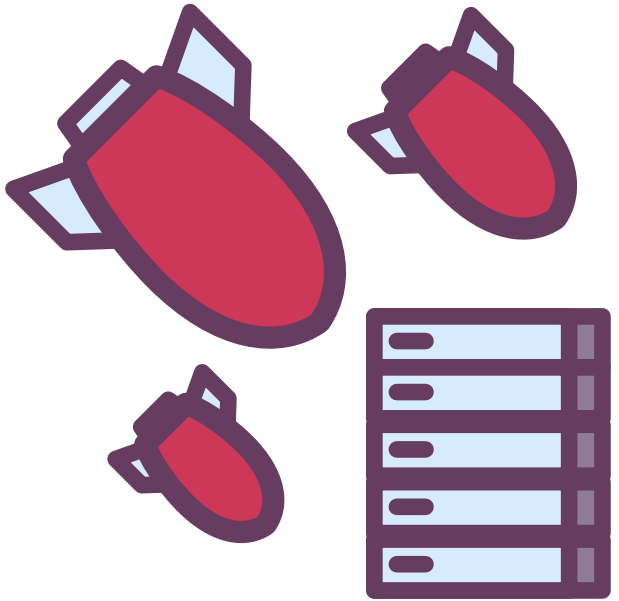
1.2a

What do we mean by “impossible?”

In hash functions, we saw:

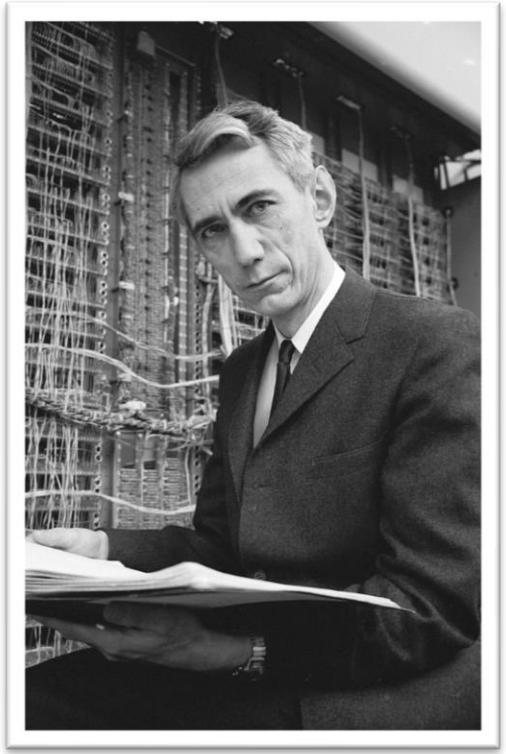
- We expect that finding a pre-image will be “extremely difficult.”
- We expect that going back from `H(x)` to x will be “impossible.”

These terms are rooted in notions of *informational and computational security*.



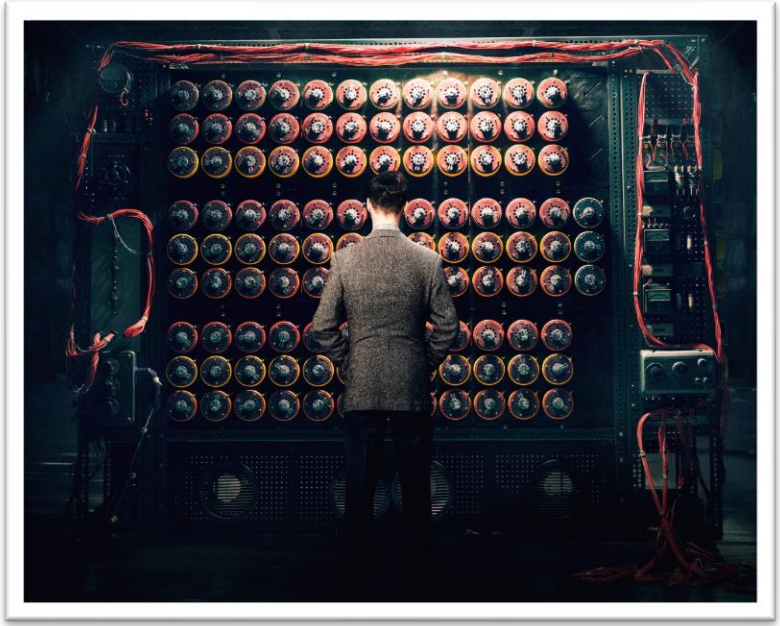
Informational security.

- Based on notions of information theory (Claude Shannon.)
- Informational security is rooted in the notion of whether something is possible *at all*.
- A “one-time pad” is informationally secure.
- We will discuss one-time pads in more detail shortly.



Computational security.

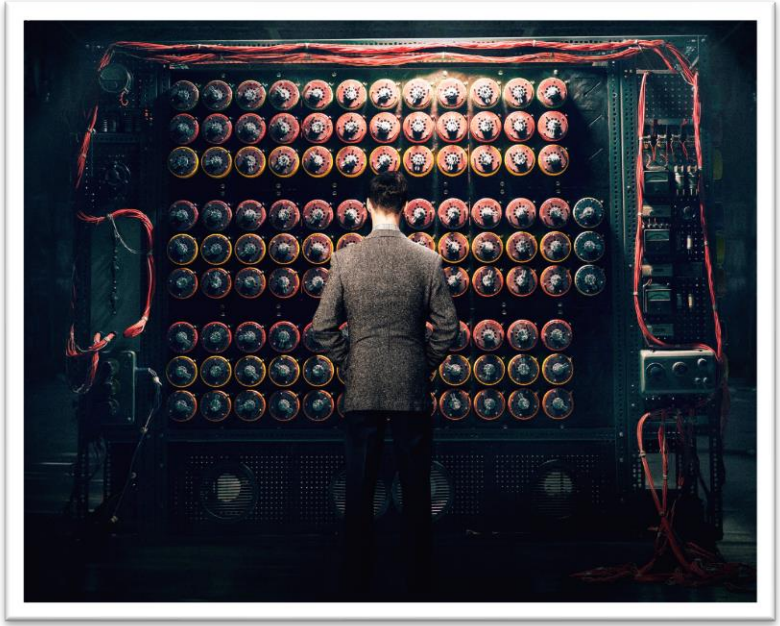
- Computational security takes somewhat relative notions into account:
- Time, memory, energy…
- Security bound is usually 2^{128} “bits of security.”
- 2^{128} = 340282366920938463463374607431768211456.



Computational security.

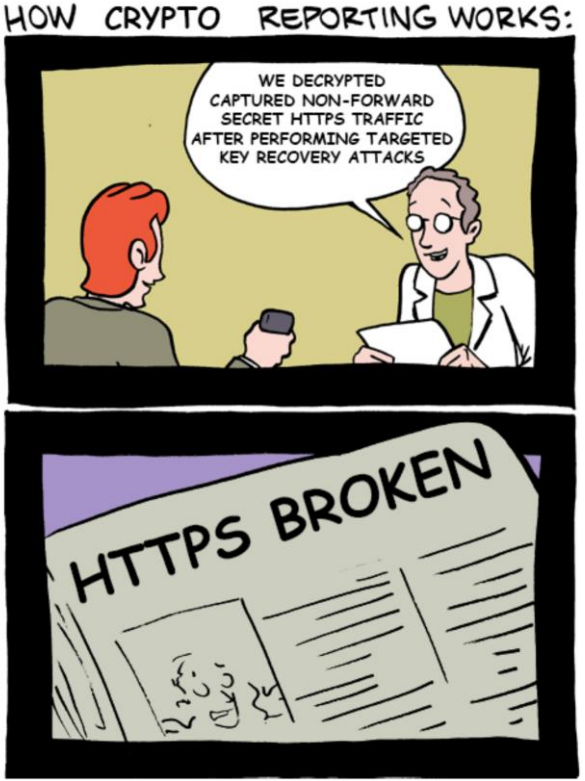
Computational attacks can be “sped up:”

- Parallelizing the computations.
- Precomputing critical steps.
- Finding breaks (or “shortcuts”) in the system:
    - Breaking a Diffie-Hellman group in half (c.f. “Socat”)
    - RC4 breaks and weaknesses.



Keep your wits about you…

- A “cryptographic break” to an academic is anything that helps them find the key *faster than exhaustive search*. By this definition, almost everything out there is broken.
- A cryptography engineer is more concerned with computational breaks, i.e. those bounded by practical notions.



Did you know?

Even at 100 billion keys per second, it would take more than 100,000,000,000,000,000,000 years to reach a key space of 2^{128}.

Test your knowledge!

What is the double of a key space of size 2^{128}?

- ☐ **A**: 2^{256}
- ☐ **B**: 2^{512}
- ☐ **C**: 2^{129}

Test your knowledge!

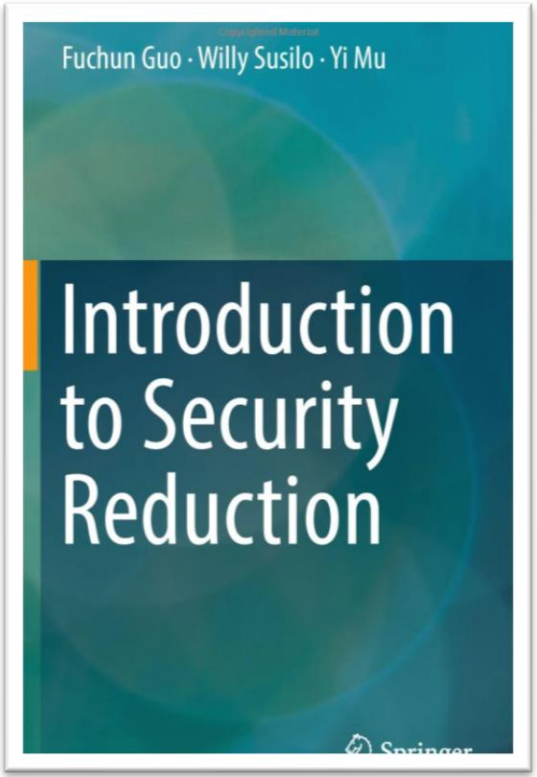
What is the double of a key space of size 2^{128}?

- ☐ **A:** 2^{256}
- ☐ **B:** 2^{512}
- ☑ **C:** 2^{129}

Ways to achieve a notion of security.

- Provable security: breaking our primitive is the same as finding an efficient solution to a mathematical problem (hopefully one that is long-thought to be difficult.)
    - Diffie-Hellman: discrete logarithm problem.
    - RSA: integer factorization problem.

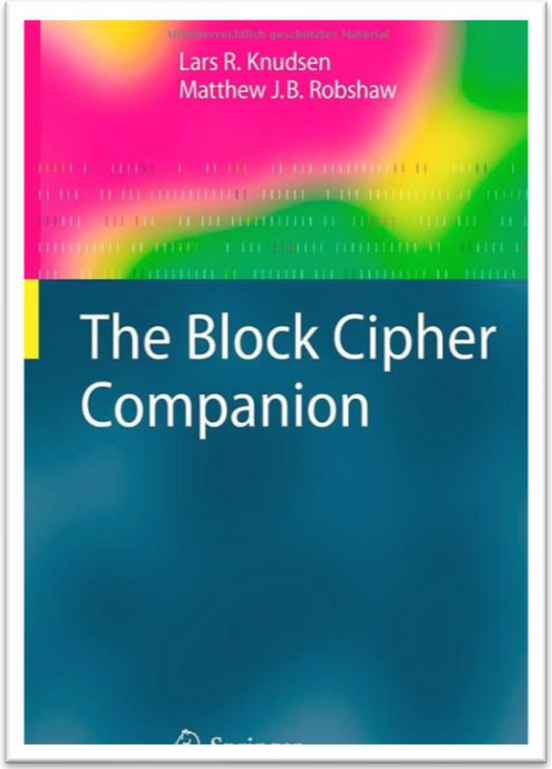
Book on the right is recommended advanced reading if you're interested in this.



Ways to achieve a notion of security.

- Basing security relative to another construction: hash-based signatures are an example.
- Heuristic security: educated attempts, wide-ranging statistical analyses, studies on simplified components of the cipher, etc. Block ciphers are an example.

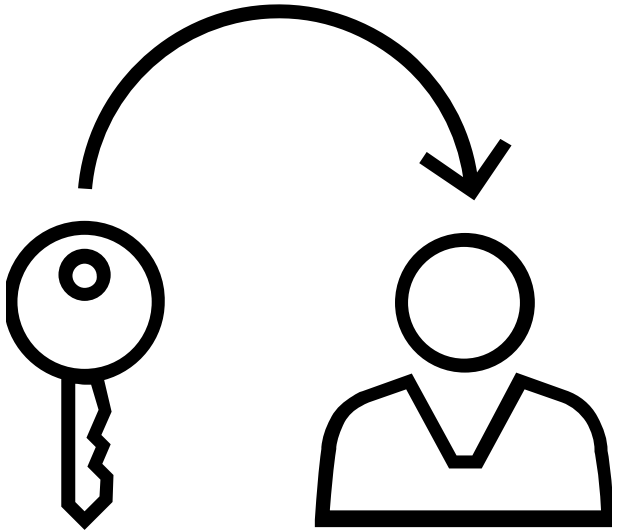
Book on the right is recommended advanced reading if you're interested in this.



“Symmetric” encryption?

It’s very simple:

- “Symmetric” means Alice and Bob have the same key.
- “Asymmetric” means public-key cryptography: each party has a different key pair.



Protocols need building blocks

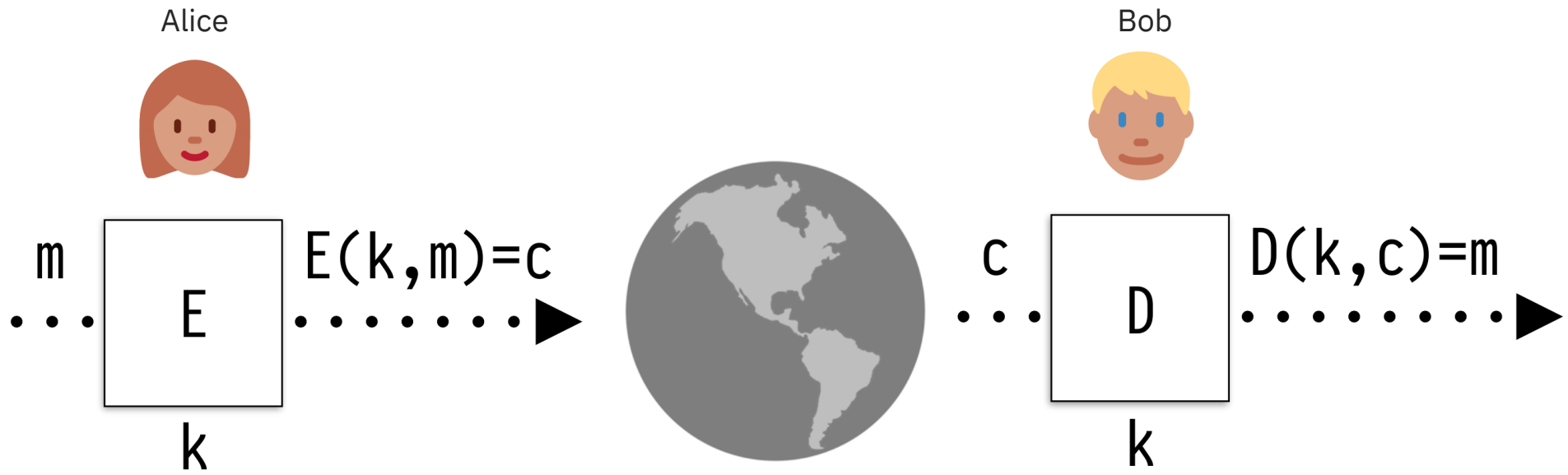
Asymmetric primitives.

- *Public key agreement algorithms*: client and server can agree on a secret encryption key over a public channel (wow!)
- *Signature algorithms*: an authority can sign a certificate proving that the server is indeed who it says it is.

Symmetric primitives.

- *Secure hash functions*: the client and the server can generate integrity-preserving codes for encrypted messages.
- *Encryption schemes*: confidential data can be encrypted and exchanged.

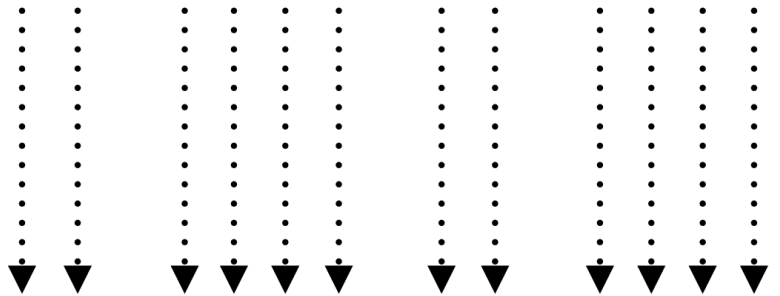
Symmetric encryption overview.



Classic example: substitution cipher.

Plaintext

JE SUIS UN CHAT



EJ TDWT DF QLMB

Ciphertext

```
{ A: 'M',   N: 'F',
  B: 'U',   O: 'N',
  C: 'Q',   P: 'P',
  D: 'C',   Q: 'O',
  E: 'J',   R: 'V',
  F: 'G',   S: 'T',
  G: 'S',   T: 'B',
  H: 'L',   U: 'D',
  I: 'W',   V: 'K',
  J: 'E',   W: 'Z',
  K: 'R',   X: 'I',
  L: 'H',   Y: 'A',
  M: 'Y',   Z: 'X' }
```

Key

Test your knowledge!

What is the key space of a substitution cipher based on an alphabet of 26 letters?

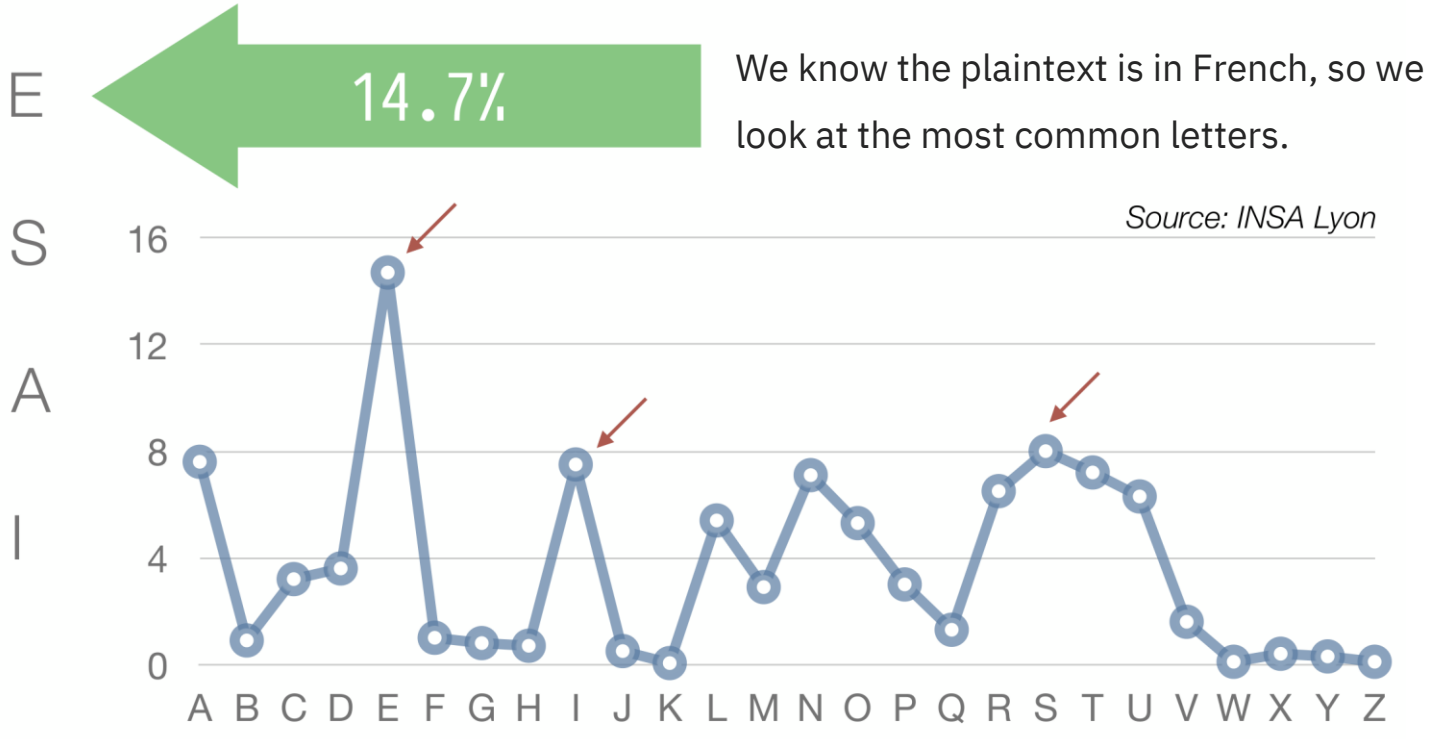
- ☐ **A**: $|K| = 26$
- ☐ **B**: $|K| = 26!$
- ☐ **C**: $|K| = 2^{26}$

Test your knowledge!

What is the key space of a substitution cipher based on an alphabet of 26 letters?

- ☐ **A:** $|K| = 26$
- ☑ **B:** $|K| = 26!$
- ☐ **C:** $|K| = 2^{26}$

2^{88} doesn't last long when we have differentials.



We know the plaintext is in French, so we look at the most common letters.

Source: INSA Lyon

Another example (in English.)

```
UKBYBIPOUZBCUFEEBORUKBYBHOBBRFESPVKBWFOFERVNBCVBZPRUBOFERVNBCVBPCYYFVUFO
FEIKNWFRFIKJNUPWRFIPOUNVNIPUBRNCUKBEFWWFDNCHXCYBOHOPYXPUBNCUBOYNRVNIWN
CPOJIOFHOPZRVFZIXUBORJRUBZRBCHNCBBONCHRJZSFWNVRJRUBZRPCYZPUKBZPUNVPWPCYVFZIXUP
UNFCPWRVNBCVBRPYYNUNFCPWWJUKBYBIPOUZBCUIPOUNVNIPUBRNCHOPYXPUBNCUB
OYNRVNIWNCPOJIOFHOPZRNCRVNBCUNENVVFZIXUNCHPCYVFZIXUPUNFCPWZPUKBZPUNVR
```

B	36
N	34
U	33
P	32
C	26

Letters

E

T

A

NC	11
PU	10
UB	10
UN	9

Digrams

IN

AT

UKB	36
RVN	34
FZI	33

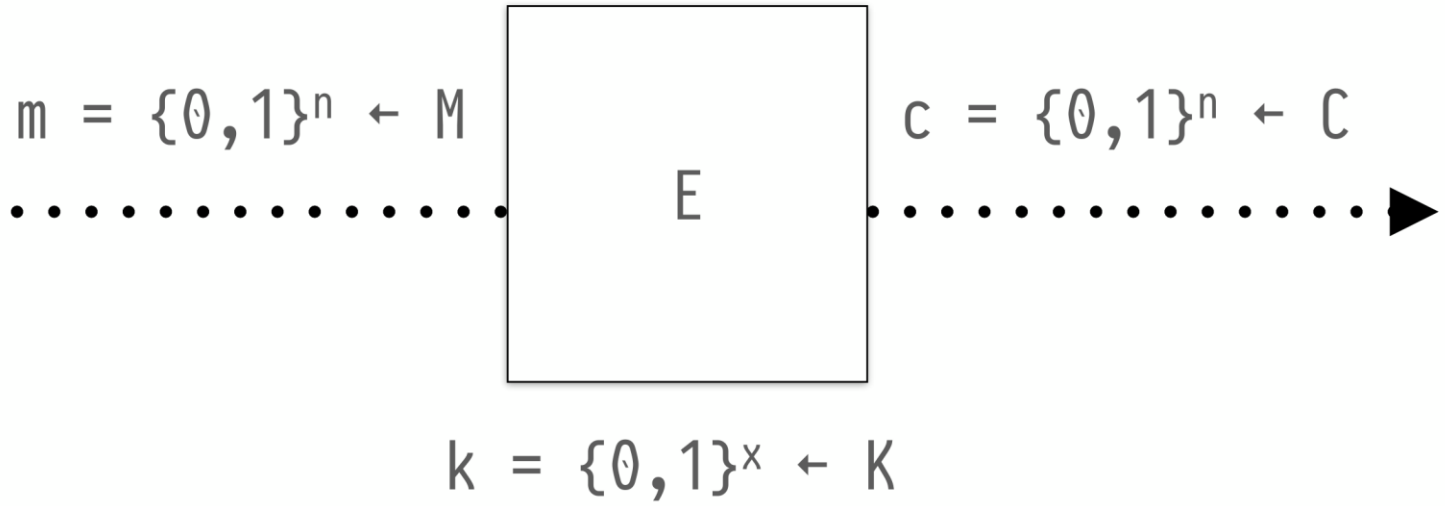
Trigrams

THE

Block Ciphers

1.2b

Block ciphers: a closer look

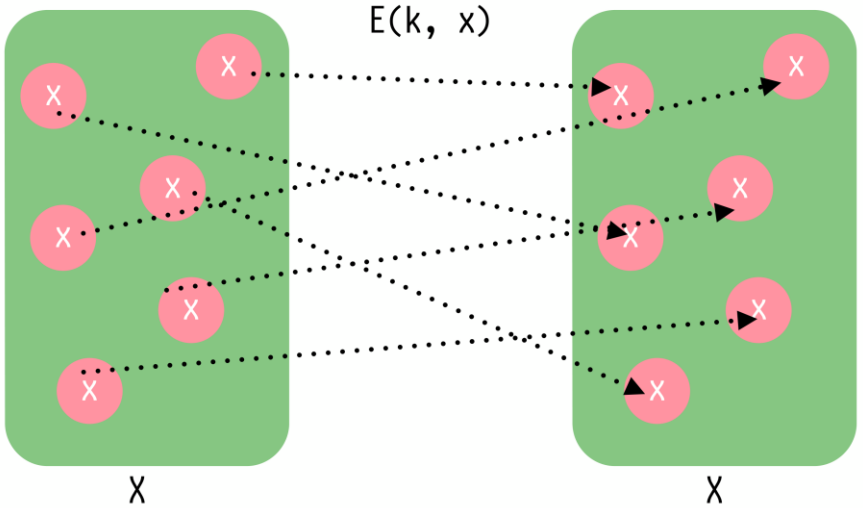


- 3DES: n = 64, x = 168
- AES: n = 128, x = 128, 192, 256

Block ciphers are “PRPs.”

“One-to-one” pseudorandom permutations.

- The space of plaintexts is the same as the space of ciphertexts.
- Only one mapping is possible from one to the other.
- Mappings are *uniform* and *pseudorandom.*



Block ciphers: a brief history.

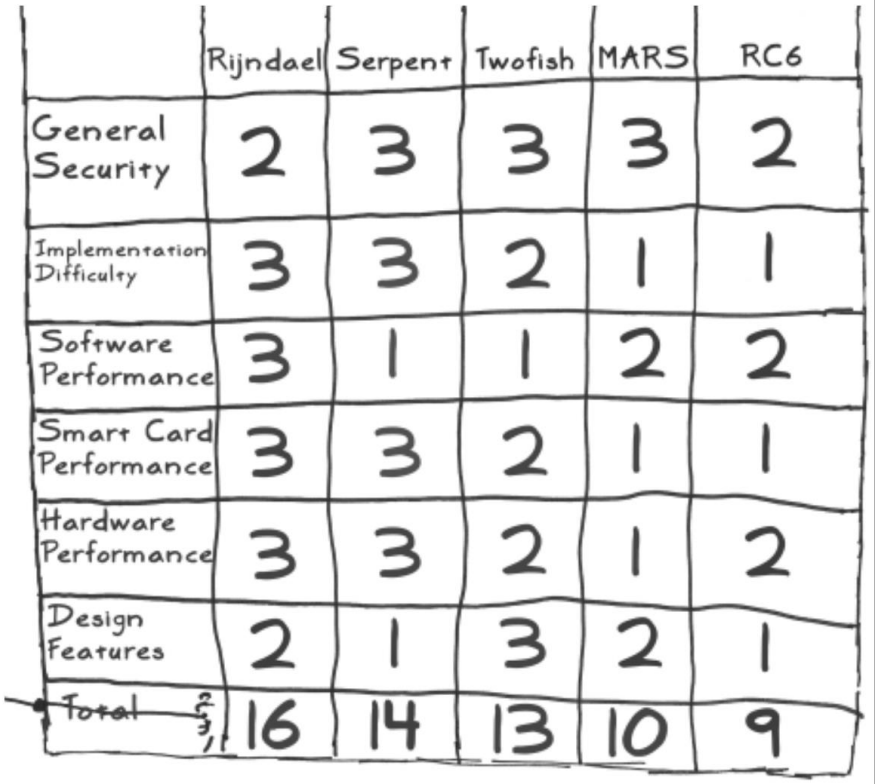
Data Encryption Standard (DES.)

- Invented in 1970 by Horst Feistel at IBM with a key size of 128 bits and a block size of 128 bits (codename: Lucifer.)
- Standardized in 1976 by the U.S. Government with a key size of 56 bits and a message size of 64 bits (hmm.)
- Broken in 1997 with practical exhaustive search

Advanced Encryption Standard (AES.)

- NIST submits RFP in 1997 and receives 15 contesting proposals.
- NIST chooses five finalists in 1995, of which AES was the winner in 2000 (codename: Rijndael.)

Block ciphers: a brief history.



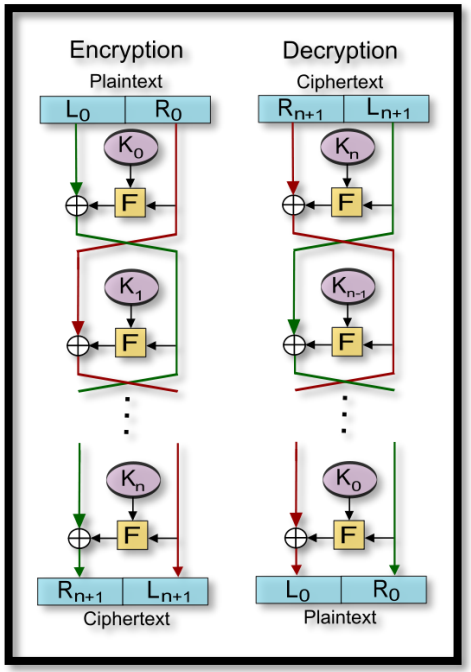
	Rijndael	Serpent	Twofish	MARS	RC6
General Security	2	3	3	3	2
Implementation Difficulty	3	3	2	1	1
Software Performance	3	1	1	2	2
Smart Card Performance	3	3	2	1	1
Hardware Performance	3	3	2	1	2
Design Features	2	1	3	2	1
~~Total~~	16	14	13	10	9

Advanced Encryption Standard (AES.)

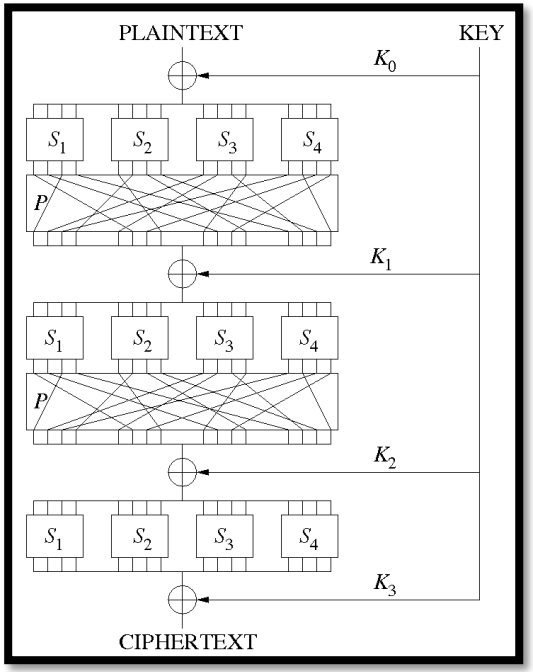
- NIST submits RFP in 1997 and receives 15 contesting proposals.
- NIST chooses five finalists in 1995, of which AES was the winner in 2000 (codename: Rijndael.)

Block ciphers: inner workings.

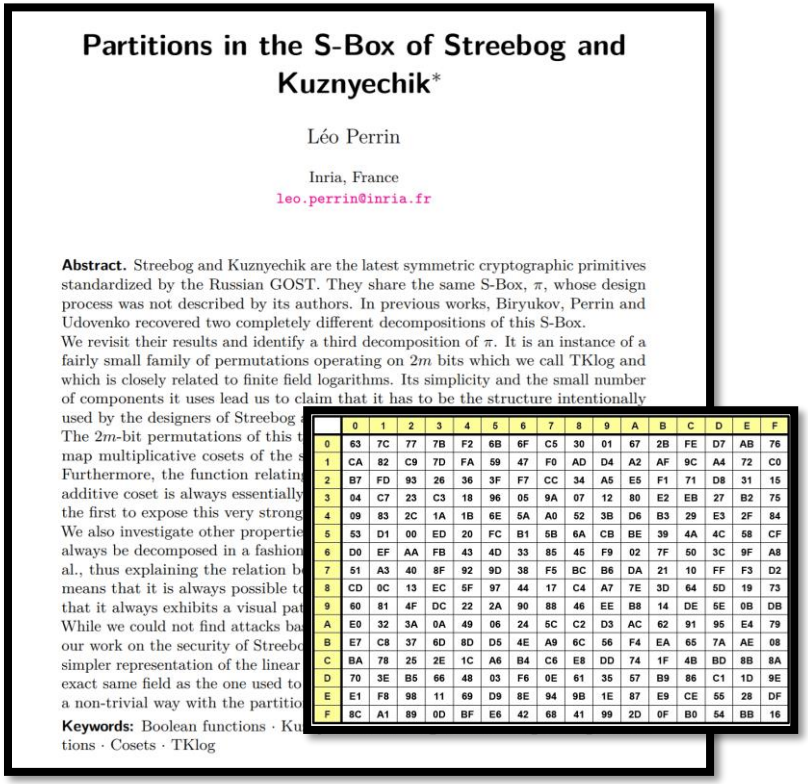
Feistel network (DES)



Substitution-permutation network (AES)



Block ciphers: hidden weaknesses.



Partitions in the S-Box of Streebog and Kuznyechik*

Léo Perrin

Inria, France
leo.perrin@inria.fr

Abstract. Streebog and Kuznyechik are the latest symmetric cryptographic primitives standardized by the Russian GOST. They share the same S-Box, π, whose design process was not described by its authors. In previous works, Biryukov, Perrin and Udovenko recovered two completely different decompositions of this S-Box.
We revisit their results and identify a third decomposition of π. It is an instance of a fairly small family of permutations operating on $2m$ bits which we call TKlog and which is closely related to finite field logarithms. Its simplicity and the small number of components it uses lead us to claim that it has to be the structure intentionally used by the designers of Streebog
The $2m$-bit permutations of this t
map multiplicative cosets of the s
Furthermore, the function relatin
additive coset is always essentially
the first to expose this very strong
We also investigate other propertie
always be decomposed in a fashion
al., thus explaining the relation b
means that it is always possible to
that it always exhibits a visual pat
While we could not find attacks ba
our work on the security of Streebo
simpler representation of the linear
exact same field as the one used to
a non-trivial way with the partitio

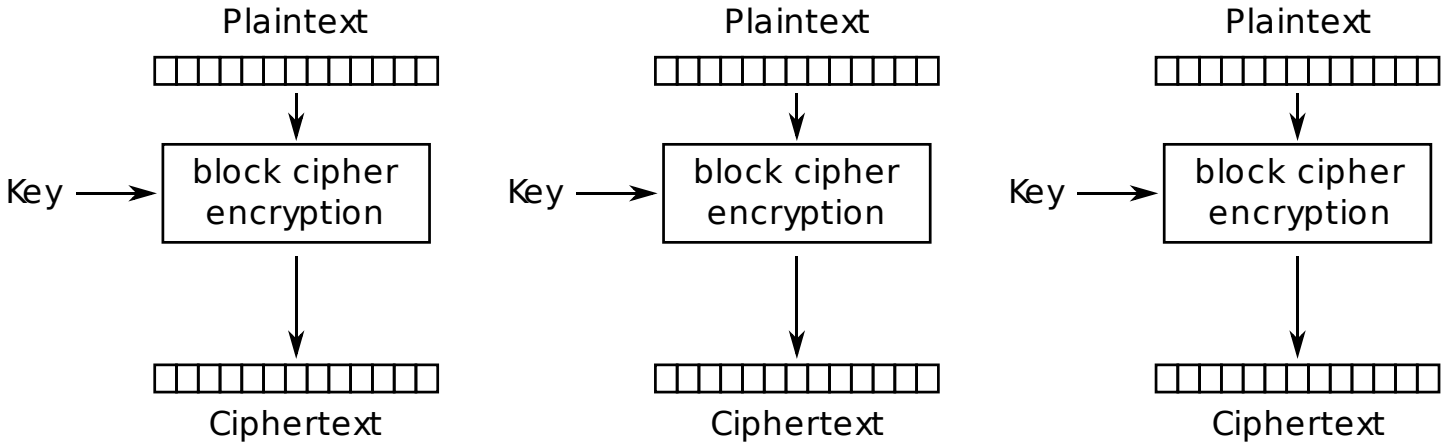
Keywords: Boolean functions · Ku
tions · Cosets · TKlog

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Substitution boxes (s-boxes) are supposed to further confuse (and render non-linear) the relationship between key and ciphertext.

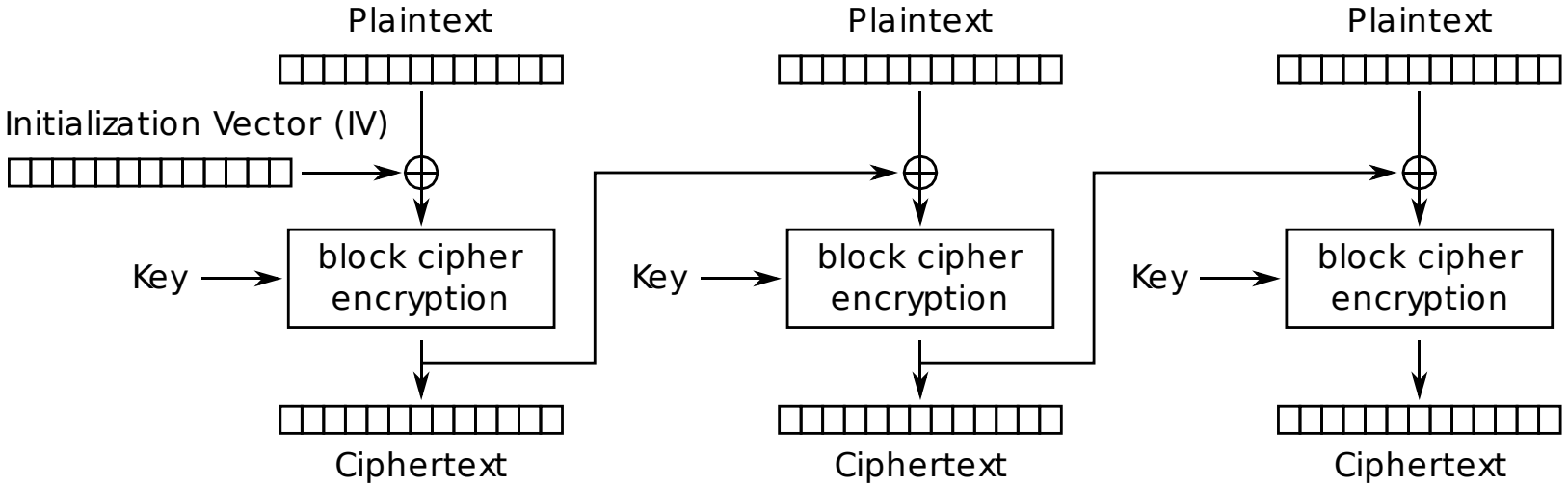
- However, they can introduce different types of attack vectors…
- *Timing side-channel*: S-box lookups can be implemented to operate in non-constant time.
- *Backdoors:* weaknesses in S-boxes can be difficult to detect by non-designers.

Electronic Codebook (ECB) mode.



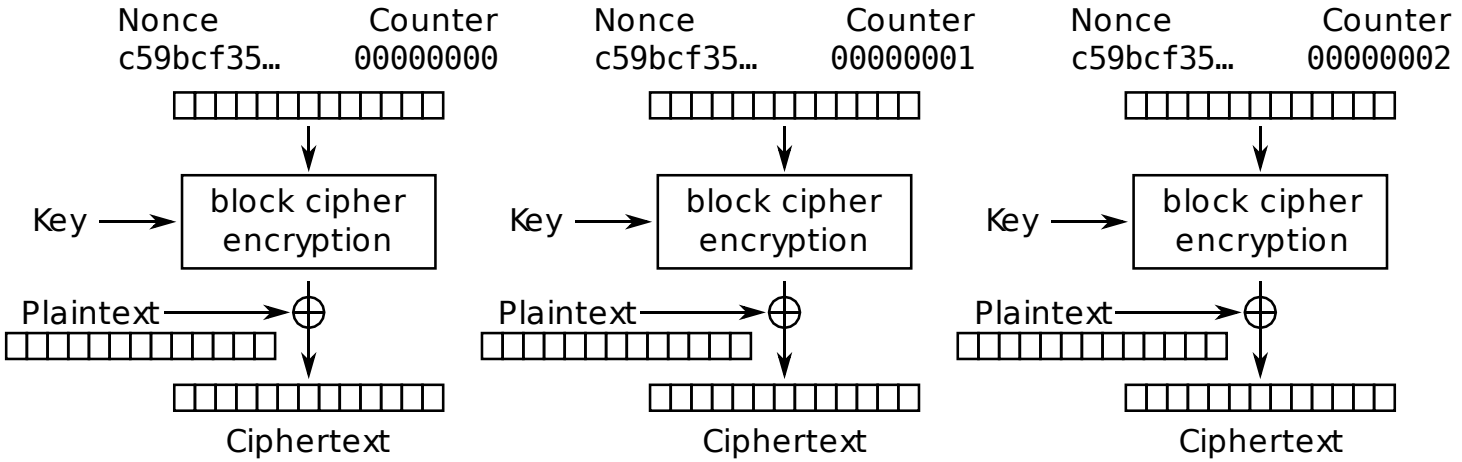
Electronic Codebook (ECB) mode encryption

Cipher Block Chaining (CBC) mode.



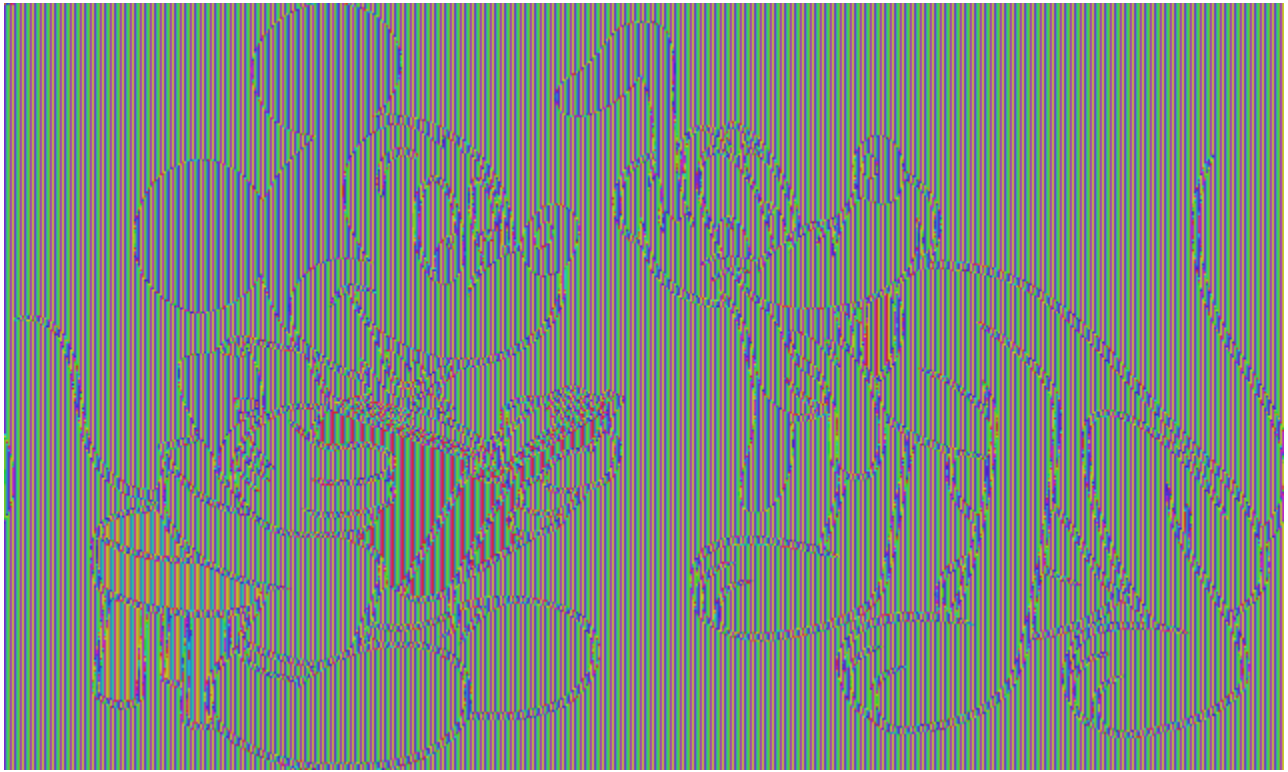
Cipher Block Chaining (CBC) mode encryption

Counter (CTR) mode.

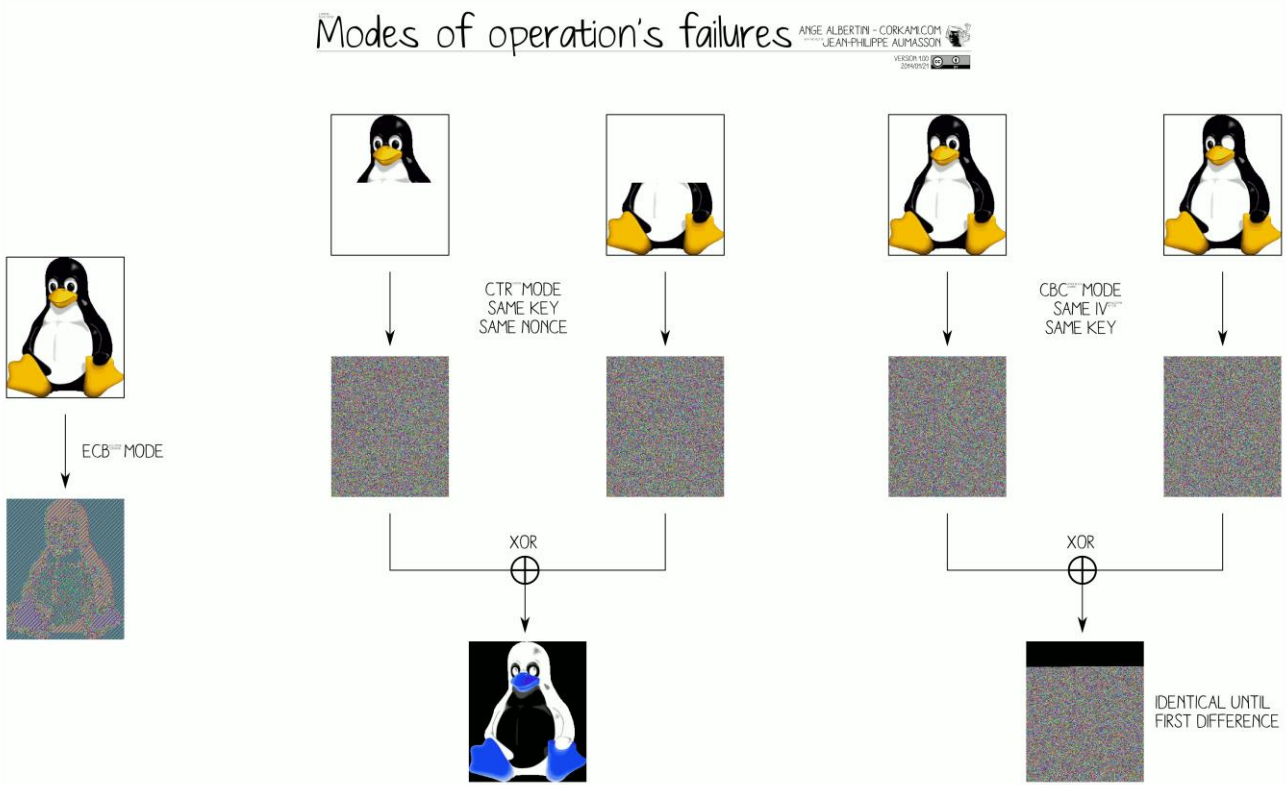


Counter (CTR) mode encryption

A not-so-great ciphertext.



More not-so-great ciphertexts.



Test your knowledge!

Which block cipher mode was used to encrypt the previous ciphertext?

- ☐ **A**: ECB mode.
- ☐ **B**: CBC mode.
- ☐ **C**: CTR mode.

Test your knowledge!

Which block cipher mode was used to encrypt the previous ciphertext?

☑ **A:** ECB mode.

☐ **B:** CBC mode.

☐ **C:** CTR mode.

Stream Ciphers

1.2c

Why stream ciphers?

- No set plaintext size.
- Can encrypt as plaintext is being produced (phone conversations, etc.)
- Let's look at one-time pads:
  - $c \leftarrow E(k, m) = k \oplus m$
  - $m = D(k, c) = k \oplus c$

Ultimately founded on a simple property: XORing a non-random element with a pseudorandom, uniform element produces a pseudorandom and uniform output.

Test your knowledge!

You are given a one time pad-encrypted message `c` and its plaintext `m`. Can you obtain the key?

- ☐ **A**: No.
- ☐ **B**: $k = m \oplus c$
- ☐ **C**: $k = m \oplus m$

Test your knowledge!

You are given a one time pad-encrypted message c and its plaintext m. Can you obtain the key?

- ☐ **A:** No.
- ☑ **B:** $k = m \oplus c$
- ☐ **C:** $k = m \oplus m$

One-time pads, a good idea?

- Excellent security.
- High performance.

But...

- Key as long as the message.

We need PRFs to create keystreams.

- Pseudorandom Functions (PRFs) can take an arbitrarily small input and create an arbitrarily large, uniform, pseudorandom output.

$$G: \{0,1\}^s \longrightarrow \{0,1\}^n$$

$$c \leftarrow E(k, m) = m \oplus G(k)$$

$$m = D(k, c) = c \oplus G(k)$$

Test your knowledge!

Can a PRF-based stream cipher achieve information-theoretic security?

Test your knowledge!

Can a PRF-based stream cipher achieve information-theoretic security?

No: the key is smaller than the message.

Test your knowledge!

$$c_1 \leftarrow E(k, m_1) = G(k) \oplus m_1$$
$$c_2 \leftarrow E(k, m_2) = G(k) \oplus m_2$$

Test your knowledge!

$$c_1 \leftarrow E(k, m_1) = G(k) \oplus m_1$$
$$c_2 \leftarrow E(k, m_2) = G(k) \oplus m_2$$

$$m_1 \oplus m_2 = c_1 \oplus c_2$$

Test your knowledge!

$$c_1 \leftarrow E(k, m_1) = G(k) \oplus m_1$$
$$c_2 \leftarrow E(k, m_2) = G(k) \oplus m_2$$

$$m_1 \oplus m_2 \quad = \quad c_1 \oplus c_2$$

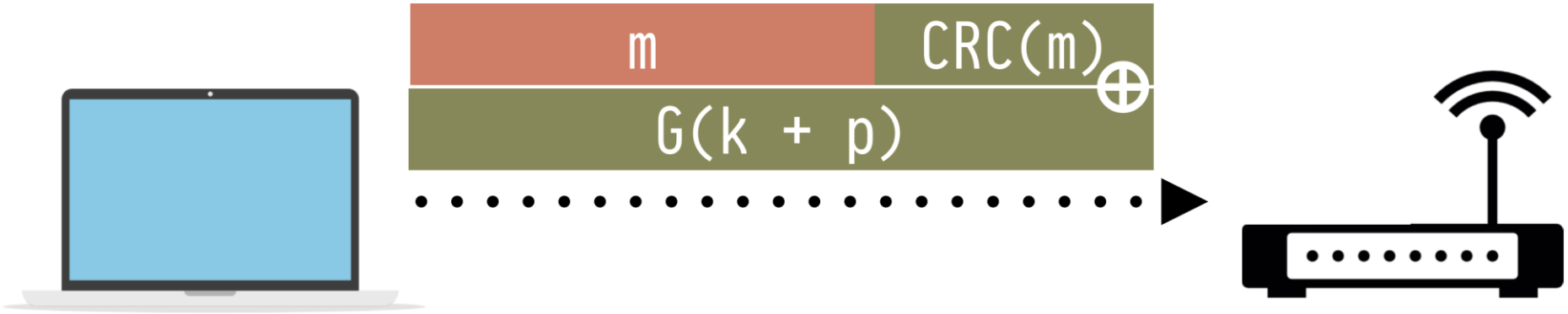
$$m_1 \oplus m_2 + \text{linguistic analysis} = m_1 \ , \ m_2$$

When dealing with stream ciphers, we base ourselves on a new security definition: the unpredictability of G's output.

$G: \{0,1\}^s \longrightarrow$

Knowing part of the output does not allow an attacker to predict the rest.

WEP: Case study of a broken stream cipher.

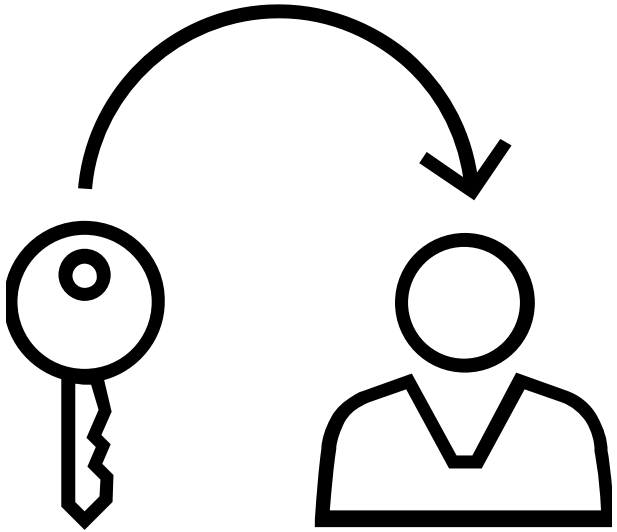


Because of weaknesses in the underlying stream cipher generator (here, RC4), WEP was broken.

A note on authenticity and integrity.

Block ciphers and stream ciphers are both unauthenticated.

- In block ciphers, corruption often “cascades”…
- …but in stream ciphers, even individual bits can be flipped!



A note on authenticity and integrity.

In stream ciphers, even individual bits can be flipped!

```
{user: “alice”, recipient: “bob”, amount: 100}
```

```
1d ec e2 85 3e 35 c4 51 5c 68 92 7c 65 fa d6 6b
59 c7 c3 7a a4 8f 3b 38 85 f4 37 0c ca 22 52 56
  37 7e dc 33 0a 82 c6 81 94 31 bb 80 99 9c 3a
```

Modify here for catastrophic consequences

Test your knowledge!

Which previously discussed primitive could help us achieve integrity for symmetric encryption?

- ☐ **A**: Public-key cryptography.
- ☐ **B**: HMACs.
- ☐ **C**: Proper threat modeling.

Test your knowledge!

Which previously discussed primitive could help us achieve integrity for symmetric encryption?

- ☐ **A:** Public-key cryptography.
- ☑ **B:** HMACs.
- ☐ **C:** Proper threat modeling.

Test your knowledge!

Which previously discussed primitive could help us achieve integrity for symmetric encryption?

- ☐ **A:** Public-key cryptography.
- ☑ **B:** HMACs. Send c || HMAC(k_{mac}, c) over the network.
- ☐ **C:** Proper threat modeling.

Next time: Public Key Cryptography

Diffie-Hellman, signature schemes and more.

1.3