TECHNISCHE UNIVERSITÄT DRESDEN

PRIVACY AND SECURITY

Stefan Köpsell, Thorsten Strufe

Security and Cryptography 1

Module 4: Basic Crypto, Stream Ciphers

Disclaimer: Günter Schäfer, Mark Manulis, large parts from Dan Boneh

Dresden, WS 18

DRESDEN concept

Reprise from the last modules

You know ***threats, security goals*** (*CIA!*) and ***security services***

You know ***adversary models*** and which aspects they define

You know some historical ciphers and how they were broken (basic cryptanalysis)

You have already heard of the one-time pad and perfect secrecy

And you know some basics of discrete probability (specifically the beauty of XOR)

Module Outline

Some crypto background

Kerckhoff‘s principle

More detailed attacker models

Introduction to stream ciphers

Pseudo random generators

Semantic security (vs. Information theoretic security)

Semantic security of stream ciphers

Security Services: Achieving security goals

Recall CIA:

- ***C***onfidentiality: only authorized access to information
- ***I***ntegrity: detection of message modification
- ***A***vailability: services are live and work correctly

Where crypto can (trivially) help:

- Confidentiality: Encryption transforms plaintext to conceal it
  - — Symmetric crypto (single key)
  - — Asymmetric crypto (key-pair)
- Integrity
  - — Message authentication / signing of data with authenticated digest (cryptographic hash/signature)

Classifying Encryption Algorithms

PRIVACY AND SECURITY

Type of operation

- Substitution
- Transposition

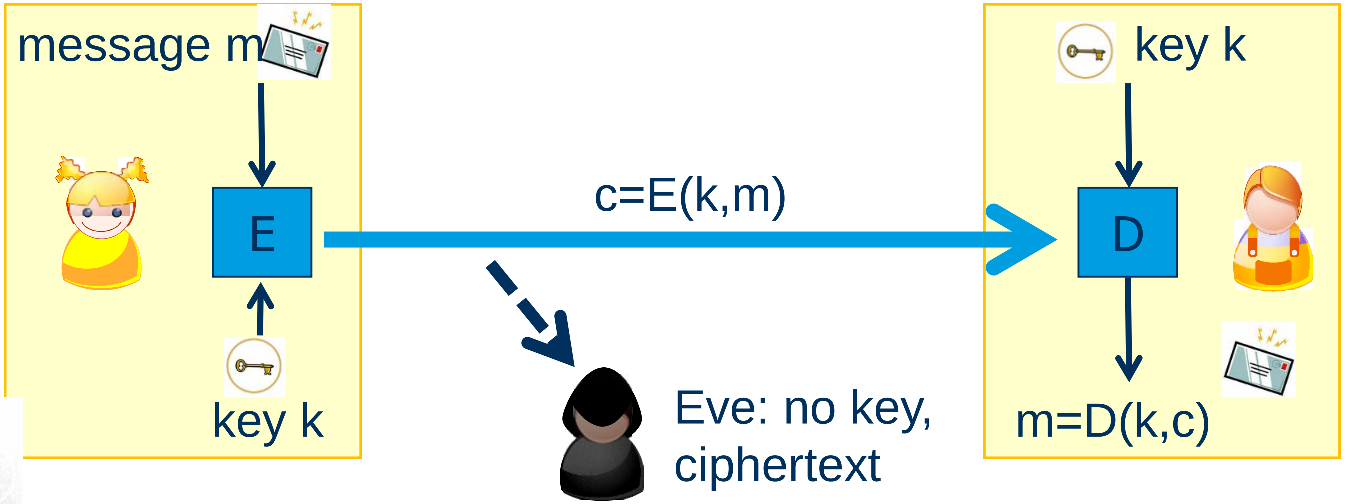
Number of keys

- Symmetric: secret key
- Asymmetric: „public key“, pair of public and private key

Processing of plaintext

- Stream ciphers: operate on streams of bits
- Block ciphers: operate on b-bit blocks

The Communication Model and Kerckhoffs



"The cipher method must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience."

i.o.w: KGen, E, and D will inevitably be discovered at some stage

→ All algorithms should be public

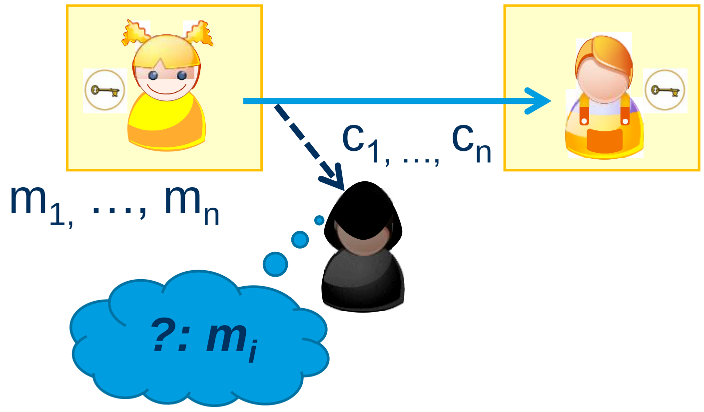
→ security must rely on secrecy ***of the key*** only

Security Notions, revisiting passive attackers

PRIVACY AND SECURITY

Ciphertext-only attack:

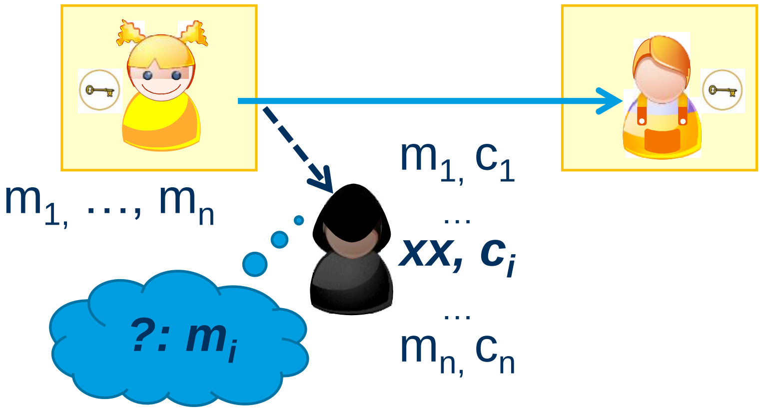
- despite concealed key
- using ciphertext only
- learn about plaintext (or key)



- *Represents weakest attacker!*

Known-plaintext attack:

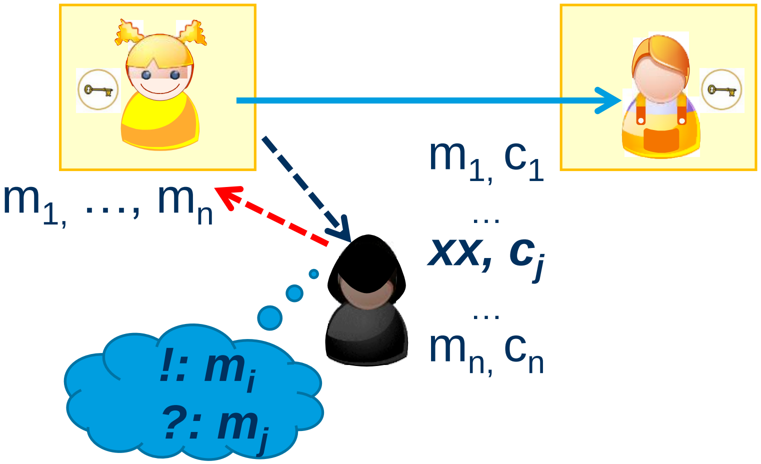
- despite concealed key
- Knowing some plaintexts
- Learn about plaintext (or key)



Security Notions, revisiting active attackers

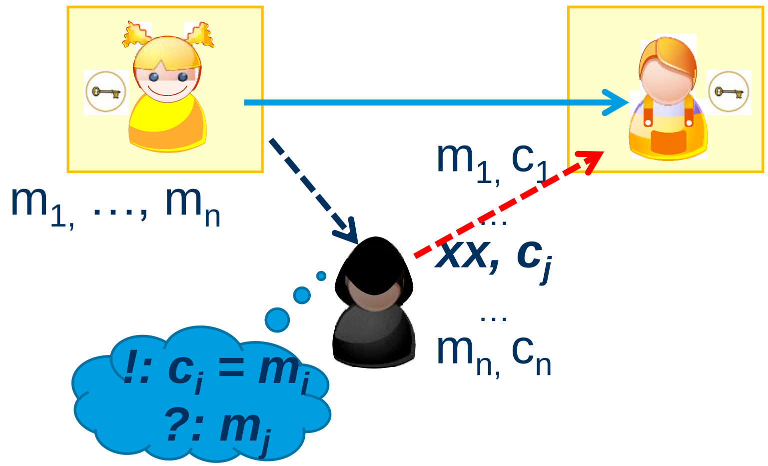
Chosen-plaintext attack:

- despite concealed key
- asking Alice to encrypt m_a
- learn about $\boldsymbol{m_j}$ (or key)



Chosen-ciphertext attack:

- despite concealed key
- asking Bob to decrypt c_a
- learn about $\boldsymbol{m_j}$ (or key)



Strongest attacker!

Asymmetric crypto: non-modifying („passive") attack…

Revisiting the one time pad

Recall a symmetric cipher over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$

is a pair of efficient algorithms (E, D) where

E: $\mathcal{K} \times \mathcal{M} \rightarrow \mathcal{C}$ and D: $\mathcal{K} \times \mathcal{C} \rightarrow \mathcal{M}$

such that $\forall m \in \mathcal{M}, k \in \mathcal{K}$: D(k, E(k,m)) = m

Where for the OTP:

$\mathcal{M} = \mathcal{C} = \mathcal{K} = \{0,1\}^n$

c:= E(k,m) = m $\oplus$ k and D(k,m) = c $\oplus$ k

msg:	0	1	1	1	0	0	1	1
key:	1	1	0	0	1	0	0	1

$\oplus$

CT:								

D(k, E(k,m)) =

Can you compute the key given m *and* c*? How?*

Revisiting the one time pad

Recall a symmetric cipher over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$

is a pair of efficient algorithms (E, D) where

E: $\mathcal{K} \times \mathcal{M} \rightarrow \mathcal{C}$ and D: $\mathcal{K} \times \mathcal{C} \rightarrow \mathcal{M}$

such that $\forall m \in \mathcal{M}, k \in \mathcal{K}$: $D(k, E(k,m)) = m$

Where for the OTP:

$\mathcal{M} = \mathcal{C} = \mathcal{K} = \{0,1\}^n$

$c := E(k,m) = m \oplus k$ and $D(k,m) = c \oplus k$

msg:	0	1	1	1	0	0	1	1
key:	1	1	0	0	1	0	0	1

$\oplus$

CT:								

$D(k, E(k,m)) = D(k, (m \oplus k)) = k \oplus (m \oplus k) = (k \oplus k) \oplus m = 0 \oplus m = m$

Can you compute the key given m *and* c*? How?*

Security of ciphers

So is the one time pad secure?

What does „secure“ mean, in the first place?

Let‘s assume a CT-only attacker, what is a good requirement?

- *Attacker cannot recover the secret key*
  - — *„E(k,m) = m“ !*
- *Attacker cannot recover the complete plaintext*
  - — *„E(k,* $m_0 \mid m_1$*)* = $m_0 \mid k \oplus m_1$“!

Information Theoretic Security

Shannon (1949):
„CT should not reveal ***any*** information about PT“

Def: A cipher (E,D) over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ has ***perfect secrecy*** if

$\forall\, m_0, m_1 \in \mathcal{M}$ (with $len(m_0) = len(m_1)$)
$\forall\, c \in \mathcal{C}$ and $k \xleftarrow{R} \mathcal{K}$:

$$Pr[E(k,m_0) = c] \;=\; Pr[E(k,m_1) = c]$$

So being an attacker, what do I learn?
No CT attack can tell if msg is m_0, m_1 (or any other message)
→ No CT only attacks

Recall: The One-Time-Pad (Vernam cipher)

Truly random key, as long as the message:

m =	A	T	T	A	C	K	T	H	E	C	I	T	Y	A	T	T	W	E	L	V	E
k =	P	S	P	I	U	H	G	D	S	P	H	G	D	S	P	I	W	E	E	W	O

(+ mod 26)

c =	P	L	I	I	W	R	Z	K	W	R	P	Z	B	S	I	B	S	I	P	R	S

k =	Y	H	P	R	S	R	G	F	F	D	D	X	N	S	Q	I	S	P	W	N	F
m =	R	E	T	R	E	A	T	F	R	O	M	C	O	A	S	T	A	T	T	E	N

(+ mod 26)

… or any other message of the same length, for that matter

Now, what are the two problems with this method?

Proving perfect secrecy of the OTP

PRIVACY AND SECURITY

Suppose: ∀ m, c: $\Pr_k[E(k,m) = c] = \frac{\#keys\ k \in \mathcal{K}\ s.t.\ E(k,m)=c}{|\mathcal{K}|}$

Now, if: ∀ m, c $\#\{\, k \in \mathcal{K} :\ E(k,m) = c\}$ is constant

Then: cipher has perfect secrecy

(no matter which m, c: probability is always the same!)

For OTP: c = m ⊕ k so k = m ⊕ c

$\#\{\, k \in \mathcal{K} :\ E(k,m) = c\} = 1$

Len(m) ≤ Len(k)…
So how do we make it practical?

➔ One Time Pad has perfect secrecy

Stream ciphers in general

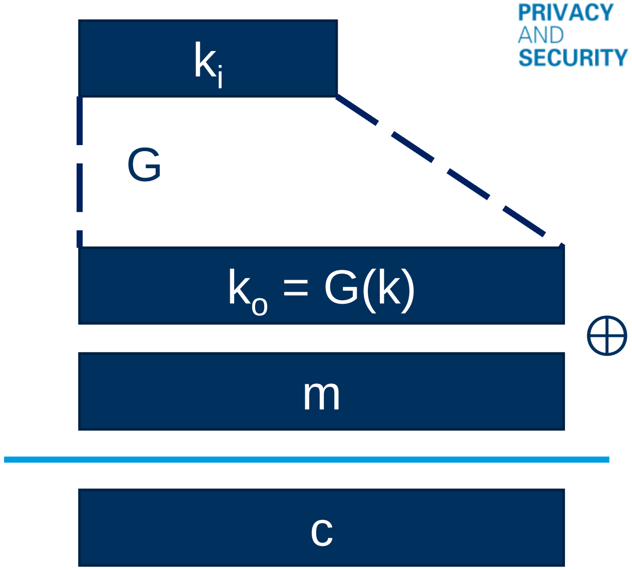
OTP:

Idea: replace random by „pseudorandom“ sequence of key bits

PRNG is a function G: $\{0,1\}^s \rightarrow \{0,1\}^n$ $n >> s$

Det. algorithm from seed space to key space (looking random)

Making the OTP practical



C:= E(k,m) = m ⊕ G(k)

D(k,c) = c ⊕ G(k)

Can this achieve perfect secrecy?

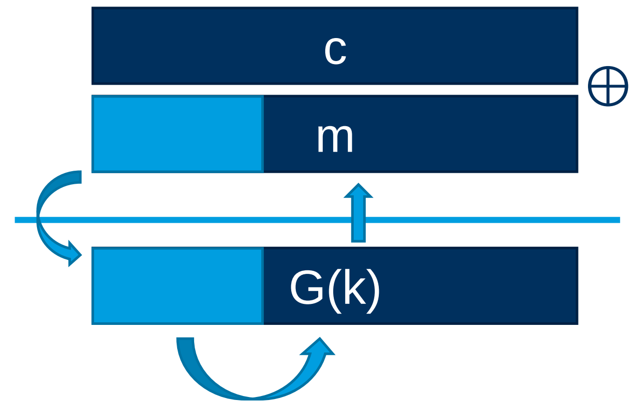
So, how „secure“ is it then?

The Problem of Predictable PRNG

What is „an efficient algorithm“?

What does „PRNG is predictable “ mean?

Suppose ∃i: $G(k)|_{1,\ldots,i} \overset{\text{alg.}}{\longrightarrow} G(k)\,|_{i+1,\ldots,n}$



Then:

We call a PRNG ($G: K \longrightarrow \{0,1\}^n$) ***predictable***, if:

∃ efficient algorithm A and ∃ $0 \leq i \leq n-1$ such that

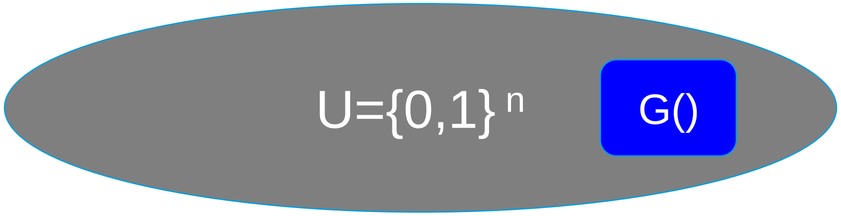
$$\Pr_{k \leftarrow \mathcal{K}}[A(G(k)|_{1,\ldots,i}) = G(k)|_{i+1}] \quad > \quad ½ + ε \quad \text{(for non-negl. ε)}$$

PRNG is ***unpredictable***: ∀i: no adv. can predict bit (i+1) for n.n. ε

On the Security of PRNG

Terminal goal: make PRNG indistinguishable from RNG (OTP)

Let $G: K \rightarrow \{0,1\}^n$ be a PRNG



So what does it mean that:

$$[k \underset{R}{\leftarrow} \mathcal{K}, output\ G(k)]$$

is „indistinguishable“ from:

$$[r \underset{R}{\leftarrow} \{0,1\}^n, output\ r]$$

Testing „Randomness“

Let‘s define statistical tests:

Let algorithm A : $\{0,1\}^n \rightarrow \{0,1\}$

denote: „0“: A(x) not random, and „1“: A(x) may be random

Numerous evidence possible:

- No. of occurrences of „0“ vs. „1“
- No. of occurrences of „00“
- Length of longest sequence of „0“
- …

Advantage of a Test

Let $G: K \rightarrow \{0,1\}^n$ be a PRNG and

A a statistical test on $\{0,1\}^n$

Then:

$$Adv_{PRG}[A,G] = \left| \Pr_{k \leftarrow \mathcal{K}}\left[A\big(G(k)\big) = 1\right] - \Pr_{r \leftarrow \{0,1\}^n}[A(r) = 1] \right| \in [0,1]$$

A cannot distinguish → Adv close to

A can distinguish → Adv close to 1

A silly example: A(x) = 0 ⇒ Adv_{PRG} [A,G] =

Advantage of a Test

Let $G: K \rightarrow \{0,1\}^n$ be a PRNG and

A a statistical test on $\{0,1\}^n$

Then:

$$Adv_{PRG}[A,G] = \left| \Pr_{k \leftarrow \mathcal{K}}\left[A\big(G(k)\big) = 1\right] - \Pr_{r \leftarrow \{0,1\}^n}[A(r) = 1] \right| \in [0,1]$$

A cannot distinguish → Adv close to 0

A can distinguish → Adv close to 1

A silly example: A(x) = 0 ⇒ Adv_{PRG} [A,G] = 0

Defining „Secure“ PRNG

Def: We say that $G:K \longrightarrow \{0,1\}^n$ is a ***secure PRNG*** if

∀ "efficient" statistical tests A:

$Adv_{PRG}[A,G]$ is "negligible"

In other words: A secure PRNG is ***unpredictable***

More generally:

Let P_1 and P_2 be two distributions over $\{0,1\}^n$

Def: We say that P_1 and P_2 are ***computationally indistinguishable*** (denoted $P_1 \approx_p P_2$)

iff ∀ "efficient" statistical tests A:

$$\left|\Pr_{x \leftarrow P_1}[A(x)=1] \; - \; \Pr_{x \leftarrow P_2}[A(x)=1]\right| < negligible$$

Secure Ciphers: another Attempt

Recall Shannons perfect secrecy:

(E,D) has perfect secrecy, if $\forall m_0, m_1 \in \mathcal{M} \; (|m_0|=|m_1|)$

$$\{E(k, m_0)\} = \{E(k, m_1)\} \text{ where } k \underset{R}{\leftarrow} \mathcal{K}$$

Let‘s say instead:

(E,D) has „some“ secrecy, if $\forall m_0, m_1 \in \mathcal{M} \; (|m_0|=|m_1|)$

$$\{E(k, m_0)\} \approx_p \{E(k, m_1)\} \text{ where } k \underset{R}{\leftarrow} \mathcal{K}$$

under given m_0 and m_1

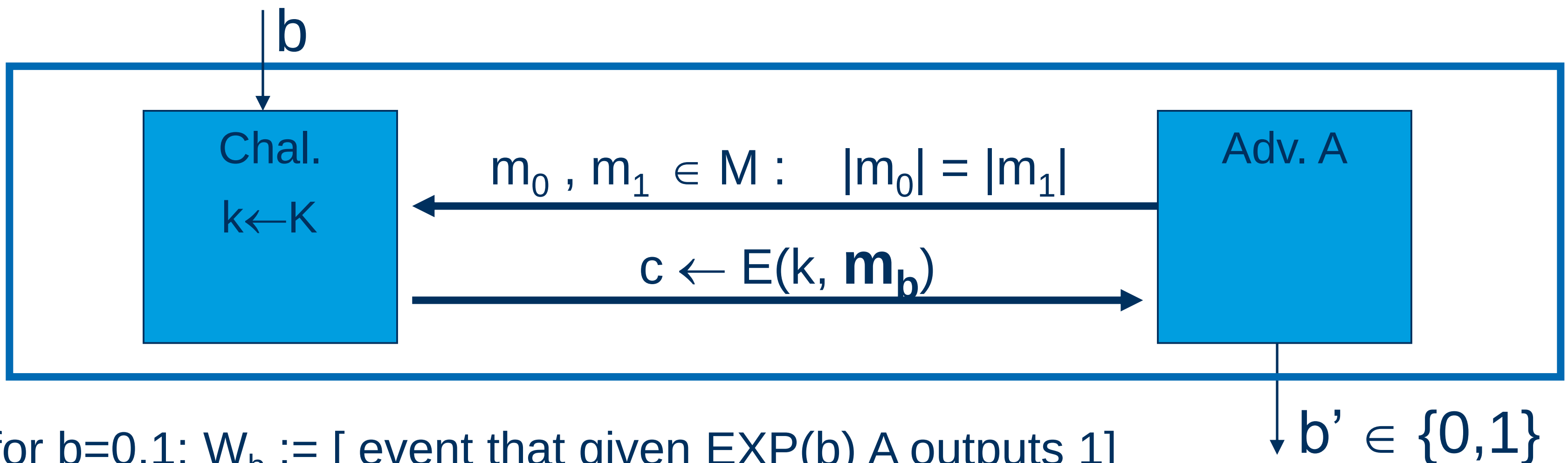
Semantic Security (for one time key)

PRIVACY AND SECURITY

Let's play our game:

A challenger flips a coin, and the adversary guesses the outcome

For b=0,1 define experiments EXP(0) and EXP(1) as:



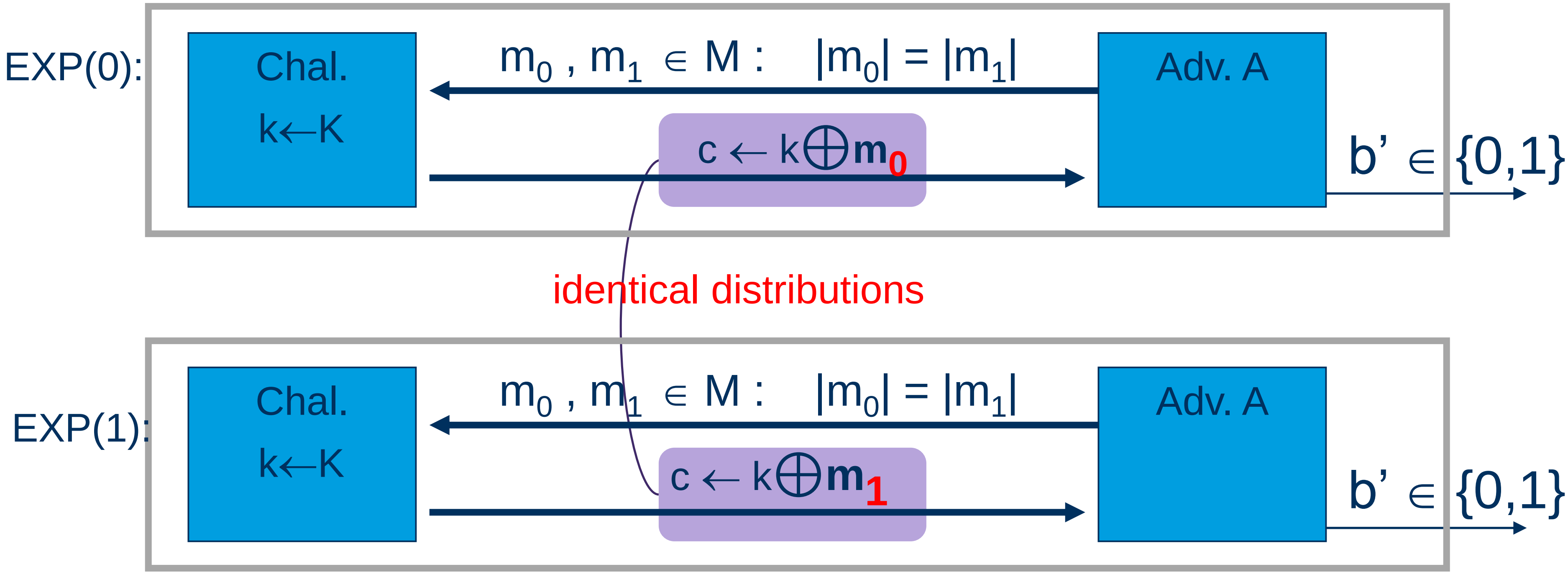
for b=0,1: W_b := [event that given EXP(b) A outputs 1]

$$Adv_{SS}[A,E] := |\Pr[W_0] - \Pr[W_1]| \quad \in [0,1]$$

Again:

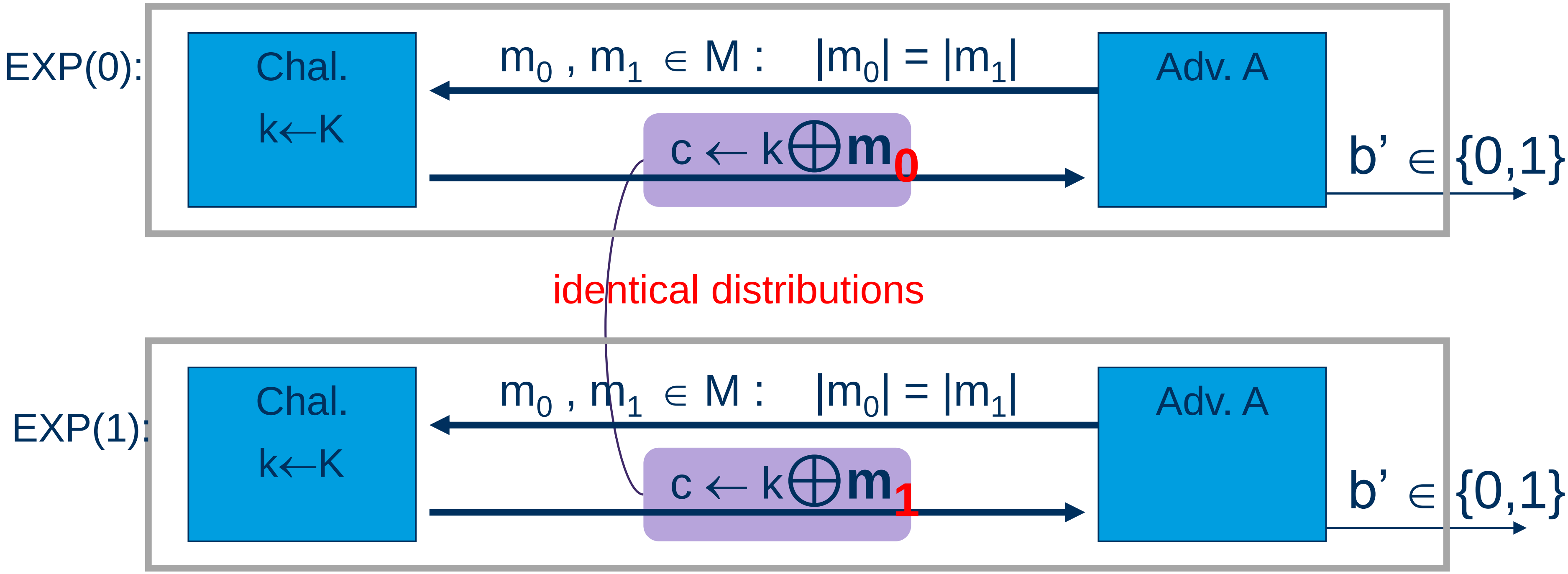
E is called ***semantically secure*** if for all eff. A, $Adv_{SS}[A,E]$ is negligible

OTP is semantically secure



For **all** A: $Adv_{SS}[A,OTP] = \left| \Pr[\, \mathbf{A(k \oplus m_0)}=1 \,] - \Pr[\, \mathbf{A(k \oplus m_1)}=1 \,] \right| =$

OTP is semantically secure



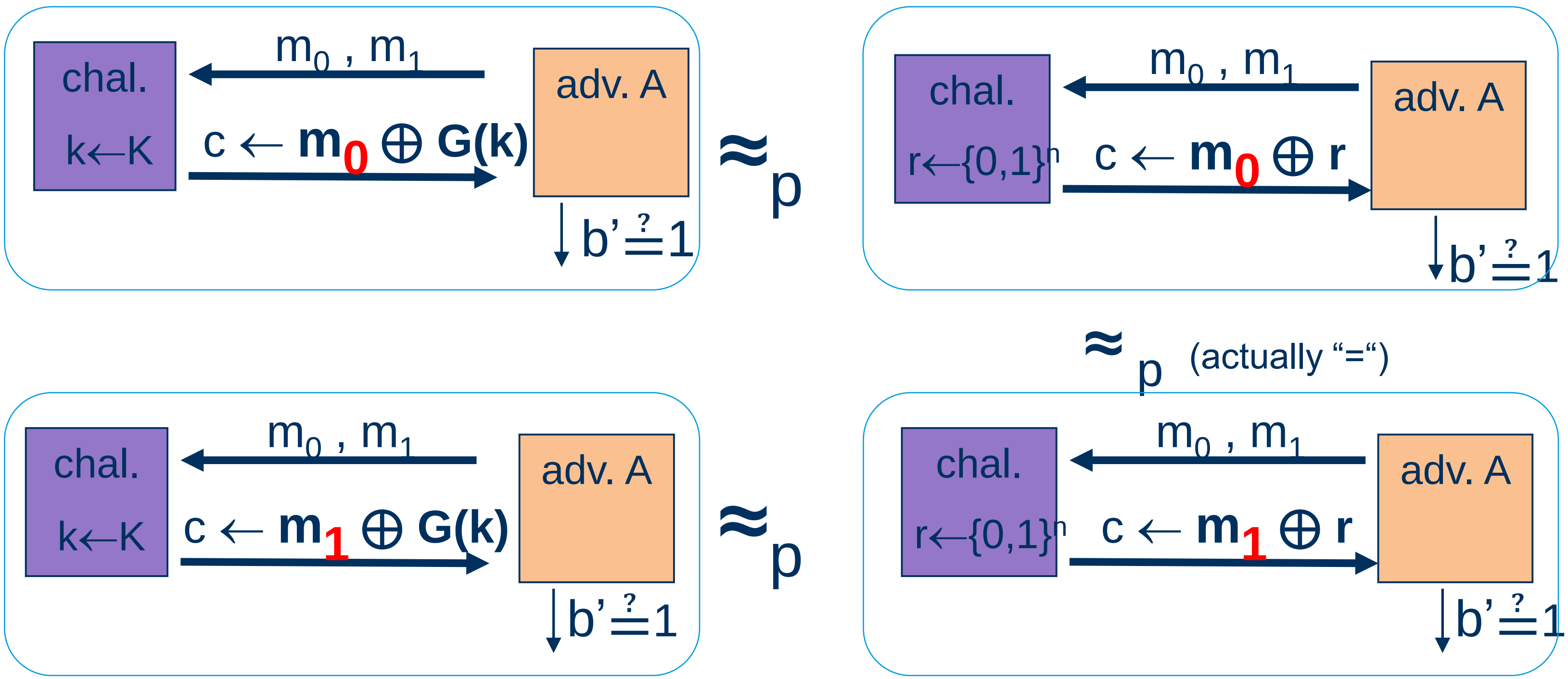
For **all** A: $Adv_{SS}[A,OTP] = \left| Pr[\,A(k\oplus m_0)=1\,] - Pr[\,A(k\oplus m_1)=1\,] \right| = 0$

So, are stream ciphers semantically secure?

We have shown:

- Unpredictable PRNG are indistinguishable from real randomness
- OTP (XOR with truly random bitstring) is secure

Proof intuition:



Two known problems of OTP constructions

The ***two*** time pad:

Assume you use a stream cipher ***key*** more than once:

$c_1 = m_1 \oplus PRNG(k)$

$c_2 = m_2 \oplus PRNG(k)$

How can this be attacked?

$c_1 \oplus c_2 = (m_1 \oplus PRNG(k)) \oplus (m_2 \oplus PRNG(k)) = 0 \oplus m_1 \oplus m_2$

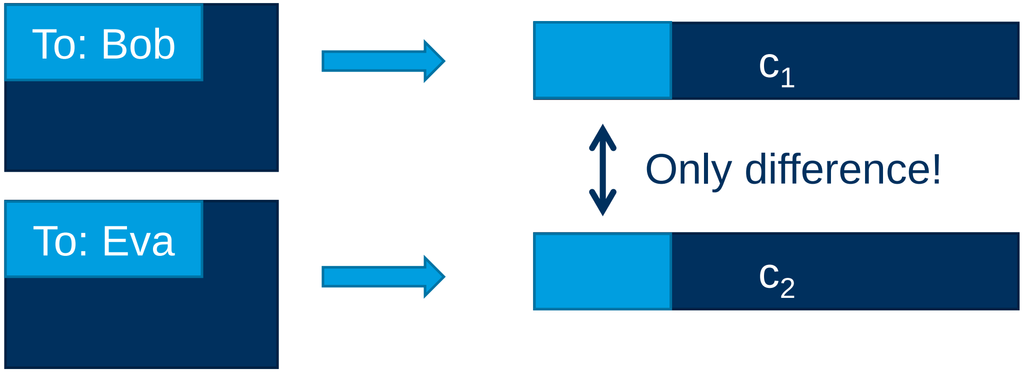
ASCII and natural language highly redundant and regular,

deriving m_1, m_2 from $m_1 \oplus m_2$ is easy…

Two Time Pads in the real world

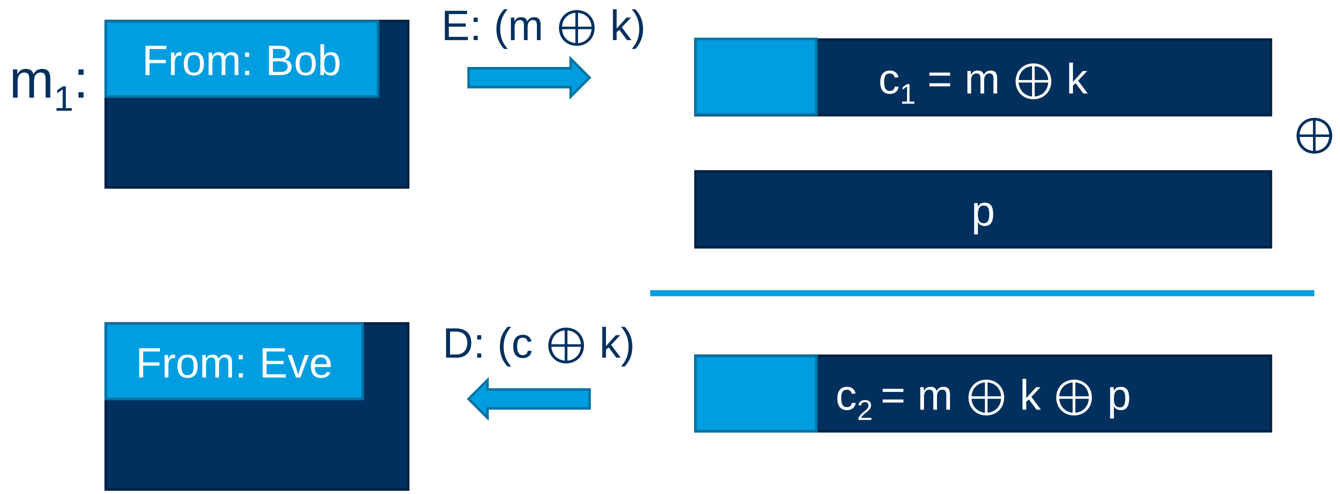
Examples are plentiful:

- Project Venona
- MS-PPTP (Win NT: two streams of messages enc with same key)
  - — Lesson learned: use separate key per direction!
- WEP (802.11b)
  - — $PRG(IV||k)$ to avoid identical keys (k=long time key)
  - — IV 24 bits and commonly reset to 0 after power cycle
  - — After each cycle ($2^{24} \approx$ 16M) again a two time pad
- Disk encryption:



Lack of integrity (Malleability)

Assuming a similar example as before:



In this simple example:

Bob = 42 6F 62; Eve = 45 76 65; Bob ⊕Eve = 07 19 07
(p:= 0 0 0 0 0 0 07 19 07)

Lesson: Modification is undetected and has predictable impact!

Summary

You know the different classes of encryption algorithms

You remember Kerckhoff‘s principle

You‘ve seen four different adversary models

You can prove that the One Time Pad has perfect secrecy

You‘ve been introduced to the idea of stream ciphers

You understand why PRNG have to be unpredictable

You know why we play games in crypto

You will avoid Two Time Pads and recall Malleability

You know semantic security and that stream ciphers are semantically secure