

Lecture Notes

**Security on the Internet**

Michael Nüsken

b-it

(Bonn-Aachen International Center  
for Information Technology)

Winter 2008



# eMail

28.10.08  
G17  
①

## Goal:

- communication, discussion
- speed
- send information, share it
- send text messages

(NOT entire DVDs!)

- make it easier, it's faster  
it's cheaper

- less paper

- connect geographically distributed parties

- notification

- independent of sender's & recipient's location

## Format:

- pure text, electronic

- formatted:
  - Header
  - <blank line>
  - Body

Thunderbird  
Ctrl+U

Outlook

→ ? Pull to  
desktop  
and open the  
file

→ Properties give  
headers

Special header lines:

SotI  
29.10.08  
(2)

From: <sender>

eg: From: Michael Nürken <nuerken@bit...>

To: <recipient>

Subject: <subject>

Date: <sending date>

More:

Received: ~~~~~

Return-Path:

CC:

BCC:

X-Spam...

Priority:

... encoding info ...

Message-ID: ...

... format info ...

Reply-To: ...

... confirmation ...

} inserted by mail servers

← like a CC: but must be deleted before delivery.

(is it text or HTML or multipart...)

Before all that is one line starting 'From:'

eg: From: nuerken@bit.uni-bonn.de date

Transport of email?

Alice



server



server



Bob





# Security?

SofI  
29.10.08  
(5)

Goals:

Confidentiality, Privacy

encryption

"Only Bob can read the email."

Authenticity

Bob knows that it was Alice who sent the mail.

Signature

Integrity

The text wasn't changed underway.

Message flow confidentiality

Even the existence of the message stays 'secret'.

Non-repudiation

Alice cannot deny that she sent the mail. In other words, Bob can prove to Charlie that the mail is from Alice

(Accessibility, Reliability)

Proof of submission

Proof of delivery

Anonymity

a few technicalities:

SotI  
04.11.08  
⑤

- want to receive any mail  
(process)
- relay (or forward) mail
- address info MUST be included  
and legible.

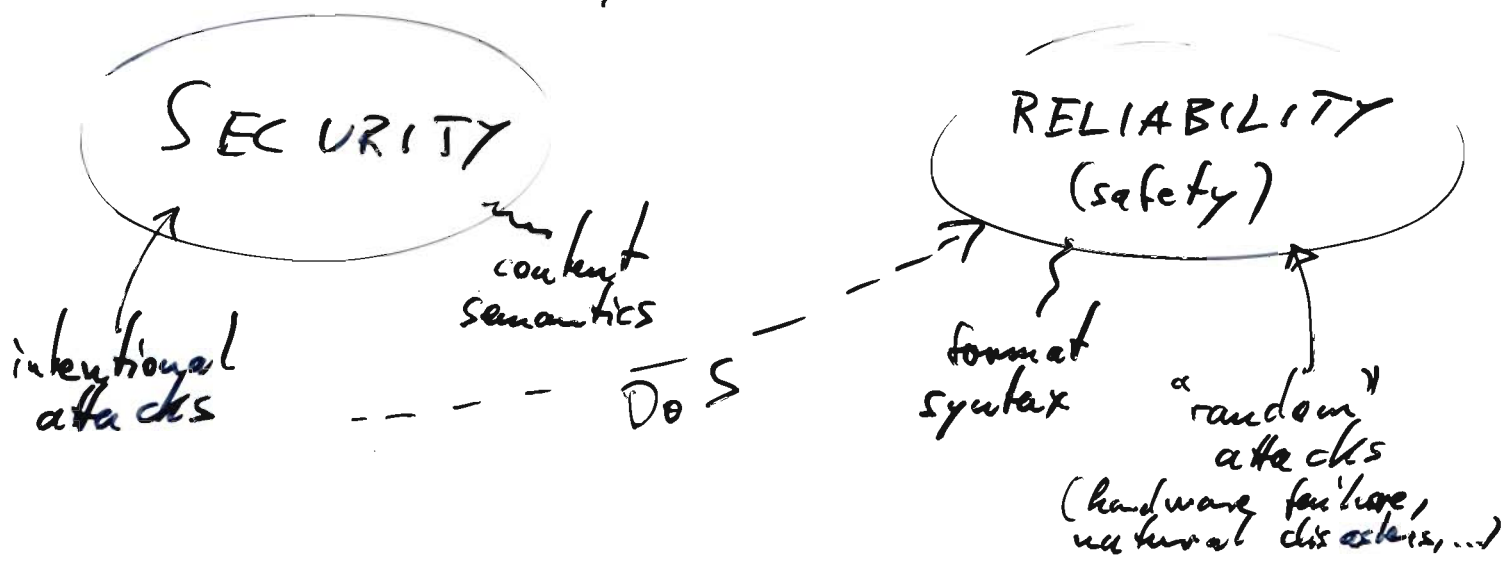
↳ DNS services supply information  
about the topology of the network.  
(Security? :))

↳ SMTP = Send Mail Transfer Protocol  
specifies details as eg.:

- each mail consists of a header and a body  
separated by a blank line.

→ headers are never encrypted

→ headers may change on the way  
and can thus not be signed  
by the sender.



```
Return-Path: <08ws-soti-admin@bit.uni-bonn.de>
X-Original-To: nuesken@math.upb.de
Delivered-To: nuesken@math.upb.de
[...]
Received: by postfix.iai.uni-bonn.de (Postfix, from userid 13020)
      id 94C365C834; Mon,  3 Nov 2008 21:10:04 +0100 (MET)
X-Sieve: cmu-sieve 2.0
X-IAI-Env-From: <08ws-soti-admin@bit.uni-bonn.de> : [131.220.8.1]
Received: from uran.iai.uni-bonn.de (uran.iai.uni-bonn.de [131.220.8.1])
      by postfix.iai.uni-bonn.de (Postfix) with ESMTP
      id 97F4F5C829; Mon,  3 Nov 2008 21:10:03 +0100 (MET)
      (envelope-from 08ws-soti-admin@bit.uni-bonn.de)
      (envelope-to VARIOUS) (2)
      (internal use: ta=0, tu=1, te=0, am=-, au=-)
Delivered-To: 08ws-soti@alias.informatik.uni-bonn.de
X-IAI-Env-From: <first.family@uni-bonn.de> : [80.136.68.129]
Received: from [192.168.178.46] (p50884481.dip.t-dialin.net [80.136.68.129])
      by postfix.iai.uni-bonn.de (Postfix) with ESMTP
      id A1CCC5C829; Mon,  3 Nov 2008 21:09:55 +0100 (MET)
      (envelope-from first.family@uni-bonn.de)
      (envelope-to VARIOUS) (2)
      (internal use: ta=1, tu=1, te=1, am=P, au=first.family)
Message-ID: <490F5A8B.6000205@informatik.uni-bonn.de>
Date: Mon, 03 Nov 2008 21:09:47 +0100
From: First Family <first.family@uni-bonn.de>
Reply-To: first.family@uni-bonn.de
User-Agent: Thunderbird 2.0.0.17 (Windows/20080914)
MIME-Version: 1.0
To: 08ws-soti@bit.uni-bonn.de
Subject: [08ws-soti] 1234567
X-Enigmail-Version: 0.95.7
Content-Type: text/plain; charset=UTF-8
Content-Transfer-Encoding: 8bit
Sender: 08ws-soti-admin@bit.uni-bonn.de
Errors-To: 08ws-soti-admin@bit.uni-bonn.de
X-BeenThere: 08ws-soti@bit.uni-bonn.de
X-Mailman-Version: 2.0.4
Precedence: bulk
[...List-Stuff...]
X-Virus-Scanned: by mailscan-system at math.uni-paderborn.de
X-Spam-Status: No, hits=0.2 tagged_above=-999.0 required=4.0 tests=AWL,
      BAYES_00, DNS_FROM_SECURITYSAGE, SPF_PASS, SUBJ_HAS_UNIQ_ID,
      UNIQUE_WORDS
X-Spam-Level:

-----BEGIN PGP MESSAGE-----
Charset: UTF-8
Version: GnuPG v1.4.9 (MingW32)
Comment: Using GnuPG with Mozilla - http://enigmail.mozdev.org

hQIOA8SRdzclIdlqEaf/VqWFWs1Y2rqD0AQgBjJAyVWshp6TnEFutXOEloM4q4z
CVtNAium3o2+6R3bToYgx7NIetmiQWsRm7o5QWmIedKu6zu2ogvn275ik7lvBAKk
0/M+IfU12WSjpmYDZm62R2iAjlwQy6BbLbPeGXJ/AICm65mqajUT/mum8PA8ako6
EezCwYpbS3A0V0xHopKWDWtc9iUBaIsGR9xLozvcVYXXWMCJSV/BAHewoTFD8U57
vnMU0oSp/j8VjI+kp6koY86MJOnPlcUUYG5j+IHnuJpfpIbxs2c5cNwYlKFuvZrV
RpnjoDg/6lATmssidZEw5mF4/utOG913ftKoCdXpGAf9Fzul4wPGUFOzcATLX4Ef
Q+I+x60keFC4K+mIwefsZHdhbT/XtilkeoFCTaHtvWwQqTuaSfxRnlaJshQzwHxL
[...]
aHvqZs9s5+264Q0yUgB8i7AVq6d64JL8lglh3vKECdDFFUbslgEYjsQ0zFI4UK0i
H+xRNHEYaC8UN1EYbul0lx1MZxz3VQ8bneX7cWmuYggkYDM0XUWfX6OP3CKoCWoU
0mZbZWGzH+Il2nzeRO9/TotHfF5enDO2yuEF3Fr6flFDjlsZIFDq4jdrZy6ucMuO
o2AR6QwuWJQ037KIiJglngcfA+SO+Mbdg803wuMH3ORVMNclejo5DYRlxw==
=suKP
-----END PGP MESSAGE-----

08ws-SotI mailing list
08ws-SotI@bit.uni-bonn.de
https://mailbox.iai.uni-bonn.de/mailman/listinfo.cgi/08ws-soti
```

# Attack

> PART

Changing content / Add extra content

Changing sender / Fake sender / reply to

Changing recipient / Redirect mail

Read content

Detect message flow

Phishing

Send trojans, worms, ... malware.

Flooding / DoS

# Defence

Filters  
Blacklisting  
(Signature)  
Grey listing

Signature (CRYPTO)

Signature (CRYPTO)  
+ ...

(Encryption)

Encrypt (CRYPTO)

(Secure channels, VPN, ...)

(Certificates)

EDUCATION

IDS, TCP-sequence # cookies,  
... signature...

Sat I  
04.11.08

(2)

# Technology

SotI  
4.11.08

(3)

(1)

Encryption

→ protect against disclosure

→ NO protection against changes of content

(2)

Signature

→ protect against fake sender (~~authentication~~ <sup>identify</sup>)

→ protect against manipulation of content  
[integrity]

→ protect against denials

(connection of content  
and signer)

[authenticate]

(3)

PKI

# Encryption

SofI  
09.11.08  
(4)

Caesar's cipher:

SECRET  $\rightarrow$  VHFUHW

Encrypt: 3<sup>rd</sup> successor of each letter

Decrypt: 3<sup>rd</sup> predecessor

Attack: Brute force, try all keys!

Key? : 3.

one out 26 possible keys.

Actually:  $\begin{matrix} 0 & 0 & 0 \\ \text{enc}_\alpha : & & \end{matrix} \begin{matrix} \mathbb{Z}_{26} \\ x \\ \end{matrix} \rightarrow \begin{matrix} \mathbb{Z}_{26} \\ (x+\alpha) \bmod 26 \end{matrix}$

What is a letter?

well:  $\{A, B, C, D, \dots, Z\}$

we code them:  $\{0, 1, 2, \dots, 25\}$

↑  
Remainder  
on division  
by 26.

Kerckhoffs principle:

05.11.08

The entire encryption scheme is known to an attacker. The only thing which the attacker does not know is the key.

Conclusion:

Since Brute force attack, i.e.  
trying all possible keys,  
is always possible, the number  
of possible should be large.

(Nowadays  $2^{80}$  or  $2^{128}$  are considered  
enough.)

A better code:

Affine codes

$$\text{affine-enc}_{\beta, \alpha} : \mathbb{Z}_{26} \rightarrow \mathbb{Z}_{26}$$
$$x \mapsto (\beta x + \alpha) \bmod 26.$$

$$\# \text{ keys} \leq 26^2$$

1. Feature : CORRECTNESS:

we can uniquely decrypt  
the ciphertext.

Here in particular we must have  $\beta \neq 0$

we actually need that  $\text{affine-enc}_{\beta, \alpha}$   
is invertible, i.e. for every  $y \in \mathbb{Z}_{26}$   
we can solve

$$y = (\beta x + \alpha) \bmod 26 \dots$$

After the following slides:

Solving means:  $(y - \alpha) \cdot \beta^{-1} = x \in \mathbb{Z}_{26}$ . #keys  
That's only possible if  $\beta$  is invertible.  $\# \mathbb{Z}_{26}^* = 12$  } 50% of 26

# Integers modulo 26 (N)

Soft  
05.11.08  
②

$$\mathbb{Z}_{26} = \{0, 1, 2, 3, \dots, 25\}$$

$$a +_{\mathbb{Z}_{26}} b = \mathbb{Z}_{26}((a +_{\mathbb{Z}} b) \text{ rem } 26)$$

$$(a +_{\mathbb{Z}} b) \text{ mod } 26.$$

Division with remainder:  
Given  $a, b \in \mathbb{Z}$ ,  $b \neq 0$ ,  
there exist  $q, r \in \mathbb{Z}$   
such that

$$a = q \cdot b + r,$$

$$\text{and } 0 \leq r < |b|$$

$$r =: a \text{ rem } b \in \mathbb{Z}$$

$$q =: a \text{ quo } b \in \mathbb{Z}$$

$$\mathbb{Z}_b(r) =: a \text{ mod } b \in \mathbb{Z}_b$$

$$a \cdot_{\mathbb{Z}_{26}} b = (a \cdot_{\mathbb{Z}} b) \text{ mod } 26.$$

Properties:

PANIC + , PANIC , DON'T  
PROOF

commutative ring

May be even

Is? Not always, if so  
we call it a field!



Def  $\mathbb{Z}_N := (\mathbb{Z}_N, +, \cdot)$  as above.

SotI  
05.11.08  
(3)

Thm 1  $\mathbb{Z}_N$  is a commutative ring.

Thm 2  $\mathbb{Z}_N$  is a field iff  $N$  is prime.

Thm 3 Given  $a \in \mathbb{Z}_N$ .

then there exists an inverse  $x$  so that  $ax = 1$  in  $\mathbb{Z}_N$ .

iff  $a, N$  are coprime

iff  $\gcd(a, N) = 1$

EEA:  
 $\gcd(a, N) = s \cdot a + t \cdot N$

Thm

Note: Thm 3  $\Rightarrow$  Thm 2.

How to compute the inverse if it exists?

Example  $3 \in \mathbb{Z}_{20}$ , inverse?

$r$	$s$	$t$	$r = s \cdot a + t \cdot b$
$a = 20$	1	0	①
$b = 3$	0	1	②
2	1	-6	← ① - 6 · ②
1	-1	7	
0	3	-20	

△ Coose check: ✓

This is an example of EXTENDED EUCLIDEAN ALGO. (EEA)

So we infer that

$$1 = -1 \cdot 20 + 7 \cdot 3 \in \mathbb{Z}!$$

Thus

$$1 = 7 \cdot 3 \in \mathbb{Z}_{20}.$$

so

$$3^{-1} = 7 \in \mathbb{Z}_{20}.$$

Further statement:

Thm 4

The EEA

- (i) always terminates.
- (ii) computes  $g = s \cdot a + t \cdot b$ ,  $g, s, t \in \mathbb{Z}$   
where  $g = \gcd(a, b)$ .
- (iii) terminates after at most  $2 \log_2 \max(a, b) + 2$  steps. In other words: the number of rows is at most twice the number of bits in  $a$  and  $b$ .  $\Rightarrow O(n^3)$

As Corollary we obtain Thm 3: where  $n = \# \text{bits in } a \text{ and } b$ .

before  $\Rightarrow 6(n^2)$

$a \in \mathbb{Z}_N$  invertible.

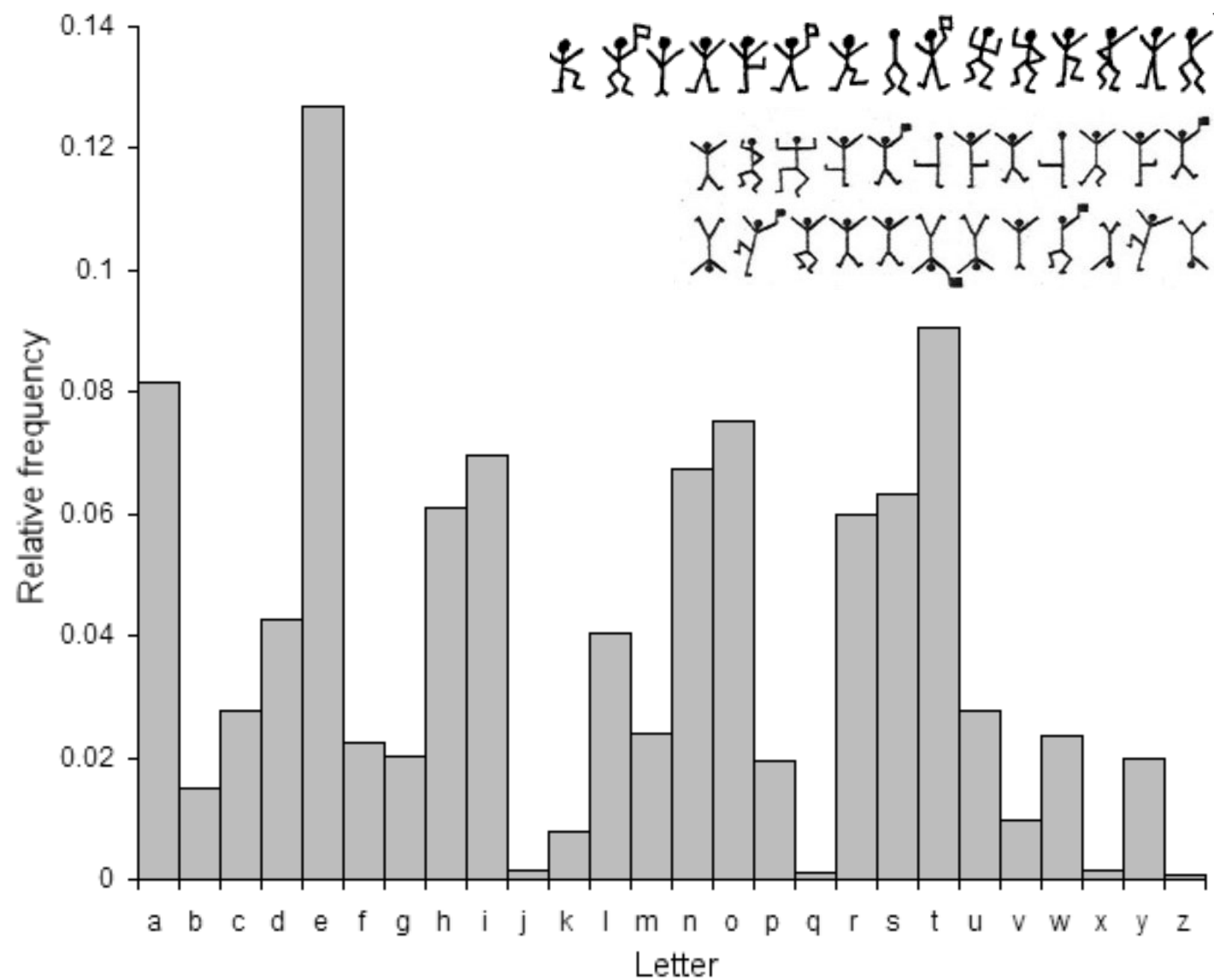
$\Rightarrow$  there is  $s \in \mathbb{Z}_N$  :  $sa = 1 \in \mathbb{Z}_N$ .

$\Rightarrow$  there is  $s \in \mathbb{Z}$ ,  $t \in \mathbb{Z}$  :  $sa + tN = 1 \in \mathbb{Z}$ .

$\Rightarrow \gcd(a, N) = 1$ .

$\Rightarrow$  there is  $s, t \in \mathbb{Z}$  :  $sa + tN = 1 \in \mathbb{Z}$ .

$\Rightarrow$  there is  $s \in \mathbb{Z}_N$  :  $sa = 1 \in \mathbb{Z}_N \Rightarrow$ .







*One of Giovanni Battista Porta's cipher disks*

Better cipher?

SotI  
11.11.08  
①

Take a permutation  $\sigma$  of  $\mathbb{Z}_{26}$ ,  
i.e. of the letters:

$$\sigma: \mathbb{Z}_{26} \rightarrow \mathbb{Z}_{26} \text{ bijjective}$$

Notions:

injective  $\equiv$  1-1 into

surjective  $\equiv$  onto

bijjective  $\equiv$  inj + surj.

Permutation cipher

Replace every letter  $x$  in the plaintext  
with  $\sigma(x)$  in the ciphertext.

Example

$x$	A	B	C	D	E	...
$\sigma(x)$	z	o	q	r	a	...

This  $\sigma$  is the key of the permutation  
cipher. There are  $26!$  such permutations,  
so  $26!$  possible keys.

$$26! \approx 2^{77}$$

Let's analyze this cipher!

1) CORRECTNESS:

Well, since  $\sigma$  is bijective,  
we can simply use  $\sigma^{-1}$  on  
each letter of the ciphertext  
and get the plaintext.

2) EFFICIENCY:

1 operation per letter:

$O(n)$ .

3) SECURITY:

Brute force attack?

Does not work any more,

$26!$  is too large.

Frequency analysis helps!

Find out the most  
frequent symbol in  
the ciphertext. Probably  
the encrypts the 'E'  
which is in many languages  
the most frequent letter.

Plug it in and continue  
with the rest...

BROKEN

SoFI  
11.11.08  
(2)

Skytale



if xy for di:

YIAS 17 TETN EIEE NACH EUNT

→



# Skytale

SoTI  
11.11.08  
(3)

CORRECTNESS?

Yes, just need another stick  
of same thickness

EFFICIENCY?

$O(n)$ , as fast as it  
can be.

SECURITY?

# keys  $\approx$  diameter of the stick  
 $\approx$  small!

→ so brute forcing has a good chance  
to succeed.

Behr analysis possible ...

→ BROKEN.

---

SUBSTITUTION  
ciphers

Caesar  
Affine  
Permutation

TRANSPOSITION  
ciphers

Skytale

# Vigenère cipher:

→ large blocks!

SotJ  
11.11.08  
④

## Encryption:

Key is a word, eg. SECRET.

Then the first letter is encrypted with the Caesar cipher corresponding to  $S = 17$ .

E	N	E	M	Y	W	H	I	T	I	N	G	A	T	B	E	R	L	I	N	.
+S	+S																			
S	E	C	R	E	T	S	E	C	R	E	T	S	E	C	R	E	T	S	E	
4	4																			
W	R																			

CORRECTNESS? ✓

EFFICIENCY?  $O(n)$ .

SECURITY?

Brute force:  $26^l$ ,  $l$  = length of the key word.

Frequency analysis: DOESN'T WORK... :)

## Possible weaknesses

- keyword may contain 'lots of Es', eg. may have structure.

- try to execute frequency analysis on every  $l$ th letter. But  $l$  is unknown. By guessing  $l$  we can read off the frequencies whether the guess was good...

This may be slow but it  
may work...

→ KASISKI attack.

Soft  
11.11.08  
(5)

BROKEN

Possible improvements

- (1) Use a key as large as the plaintext.
- (2) Choose the key at random uniformly,  
so it does not have structure.

This is the one-time-pad.

This cipher is

- correct & efficient
- absolutely secure  
in a mathematically strong sense.

# One-Time Pad

12.11.08  
Soft  
⑦

Assume the plaintext  $p$  is a string of  $n$  bits.

Now choose a key  $k$  as a random  $n$ -bit string, uniformly!

That is:  $\forall k: \text{prob}(K = k) = 2^{-n}$

$\uparrow$  random variable outputting a key

$\uparrow$  one possible choice of a key

Encrypt: ciphertext  $c = p \oplus k$ .

$\uparrow$  XOR, or addition in  $\mathbb{Z}_2$ .

Decrypt:  $p = c \oplus k$

$$\begin{aligned} \text{Proof: } c \oplus k &= (p \oplus k) \oplus k \\ &= p \oplus \underbrace{2k}_0 = p. \quad \square \end{aligned}$$

Then The one-time pad is absolutely secure.

$$\text{prob}(P=p \mid C=c) = \text{prob}(P=p)$$







Side remarks:

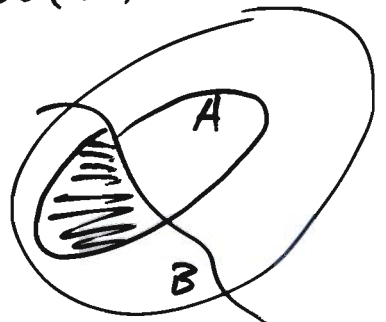
conditional probability

$$\text{prob}(A | B) =$$

probability of A  
under condition B

$$\frac{\text{prob}(A \cap B)}{\text{prob}(B)}$$

12.11.08  
SFI  
②



The theorem uses a random variable

$P$

which describes the knowledge of the attacker about the choice of the plaintext. It is independent of  $K$ .

Further:  $C := P \oplus K$ .

Proof Let's compute

$$\begin{aligned} & \text{prob}(P=p | C=c) \\ &= \frac{\text{prob}(P=p \wedge P \oplus K = c)}{\text{prob}(P \oplus K = c)} \end{aligned}$$

$$= \frac{\text{prob}(P=p \wedge K = c \oplus p)}{\text{prob}(P \oplus K = c)}$$

$$= \frac{\text{prob}(P=p) \cdot \text{prob}(K = c \oplus p)}{\text{prob}(P \oplus K = c)}$$

By definition

$$\text{prob}(K = c \oplus p) = 2^{-n}.$$

SofI  
12.11.08  
(3)

Further

$$\text{prob}(P \oplus K = c)$$

$$= \text{prob}(\exists p : \underbrace{P=p \wedge K=c \oplus p}_{\substack{\text{P} \\ \text{K}}})$$

$$= \sum_p \text{prob}(P=p) \cdot \underbrace{\text{prob}(K=c \oplus p)}_{2^{-n}}$$

$$= 2^{-n} \cdot \underbrace{\sum_p \text{prob}(P=p)}_{=1}$$

$$= 2^{-n}.$$

Thus

$$\text{prob}(P=p \mid C=c)$$

$$= \text{prob}(P=p) \cdot$$

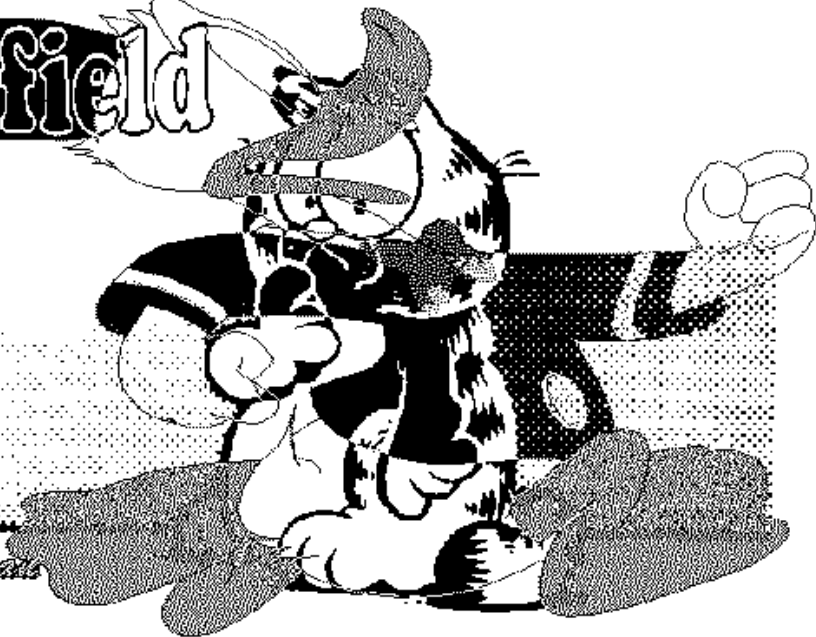
$$\frac{\text{prob}(K=c \oplus p)}{\text{prob}(P \oplus K=c)} \cdot 2^{-n}$$

$$= \text{prob}(P=p).$$

That is: the attacker does not learn anything from the ciphertext  $c$ .  $\square$

# Garfield

*CSP Scan*





Randomness is expensive!

→ what about reusing the key?

SotI  
12.11.08  
(4)

There is:

we have  $p_1, p_2$  two plain texts.

But only one key  $k$ :

$$c_1 = p_1 \oplus k,$$


$$c_2 = p_2 \oplus k$$

The attacker can add  $c_1$  and  $c_2$ :

$$c_1 \oplus c_2 = p_1 \oplus p_2.$$

This is  $n$  out  $2n$  bits!

So the attacker knows half of the plain texts. If the plain texts have structure the remaining information can be guessed.

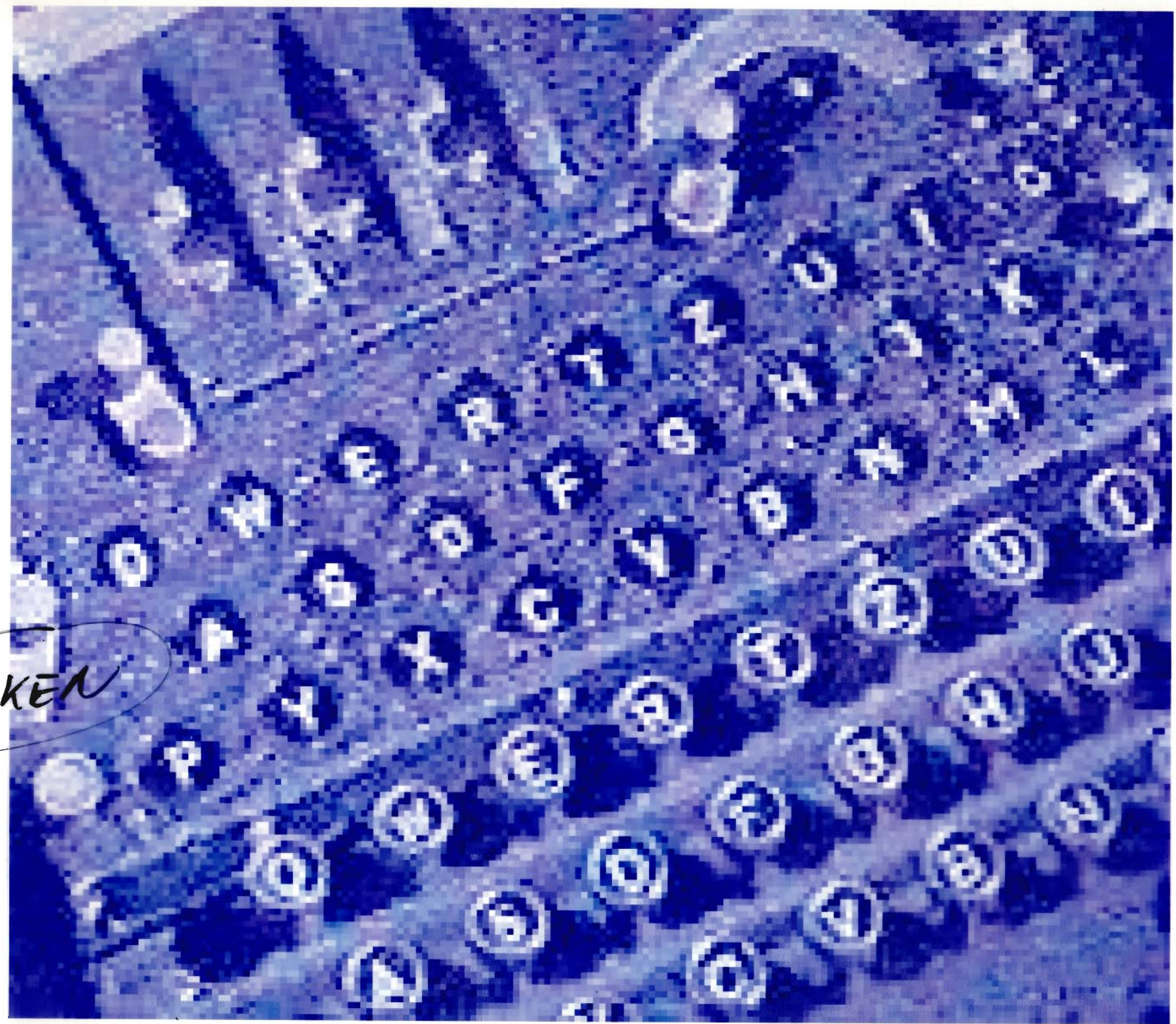
Loeuz 5242 .

Bottom:

Two Time Pad is completely insecure.

E  
N  
I  
G  
M  
A

BROKEN











**Zur Beachtung!**  
Beachte die Gebrauchsanleitung für die Chiffriermaschine (H. Dv. g. 13).

1. Zur Säuberung der Walzenkontakte alle Walzen mehrmals gegenseitig vor- und rückwärts drehen.
2. Zur Säuberung der Tastenkontakte sämtliche Tasten vor Einschaltung des Stromes mehrmals kräftig herunter drücken und hoch schnellen lassen, wobei eine Taste dauernd gedrückt bleibt.
3. Bei Einstellung der in den Fenstern sichtbaren Zeichen beachten, daß die Walzen richtig gerastet sind.
4. Die unverwechselbaren doppelpoligen Stecker sind bis zum Anschlag in ihre Buchungspare einzuführen. Die vordere Holzklappe ist danach zu schließen, da sonst 3 Lampen zugleich aufleuchten können.
5. Leuchtet bei Tastendruck keine Lampe auf, so sind die Batterie, ihre Kontaktfedern, ihre Anschlüsse am Umschalter und der Umschalter zu prüfen.
6. Leuchten bei Tastendruck eine oder mehrere Lampen nicht auf, so sind die entsprechenden Lampen, die Kontakte unter ihnen, die Kabel der doppelpoligen Stecker, die Steckerbuchsen einschließlich ihrer Kurzschlußbleche, die Walzenkontakte, die Arbeitskontakte unter den jeweils gedrückten Tasten und die Ruhekontakte unter den mit ihnen korrespondierenden Tasten zu prüfen und bei etwa vorhandenen Verschmutzungen und Oxydationen zu säubern. (Siehe auch Ziffer 2).

Von Maschine Nr. A 4388 ab dient zur Lampenprüfung die Öffnung auf der rechten Lampenseite. Von Maschine Nr. A 4388 ab dienen zur Kabelprüfung die äußerste linke und rechte Buchse der mittleren Reihe am Steckerbrett und die Kabelprüflampe auf der linken Lampenseite.

7. Walzenachse und Walzenbuchsen sind sauber zu halten und wie alle übrigen Lagerstellen hin und wieder mit harz- und säurefreiem Öl leicht einzufetten. Die festen Kontakte der Walzen sind alle 6-8 Wochen mit Polierpapier über zu schleifen und mit einem wenig getränkten Öllappen abzuwischen. Die Tastenkontakte, die Lampenkontakte und die Kurzschlußbleche sind vor Öl zu schützen.
8. Schlüsselangaben erfolgen entweder durch Zahlen oder Buchstaben.

Zum Umsetzen der Zahlen in Buchstaben oder umgekehrt dient nachstehende Tafel:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26









ENIC

$\phi(p) = p-1 = \phi(p)^2 = 0, 3 \dots$   
nos (ohj.) | Hooley's  
alms all primes





# AES

## Advanced Encryption Standard

Designed as Rijndael by JOAN DAEMEN and VINCENT RIJMEN

### The field $\mathbb{F}_{2^8}$

$\mathbb{F}_{2^8} \ni a = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5 + a_6x^6 + a_7x^7$ ,  
where  $a_i \in \mathbb{F}_2 = \{0, 1\}$ .

Representation: 8 bits for an element = 1 byte.

Addition: XOR,  $(a + b)_i = a_i + b_i$ .

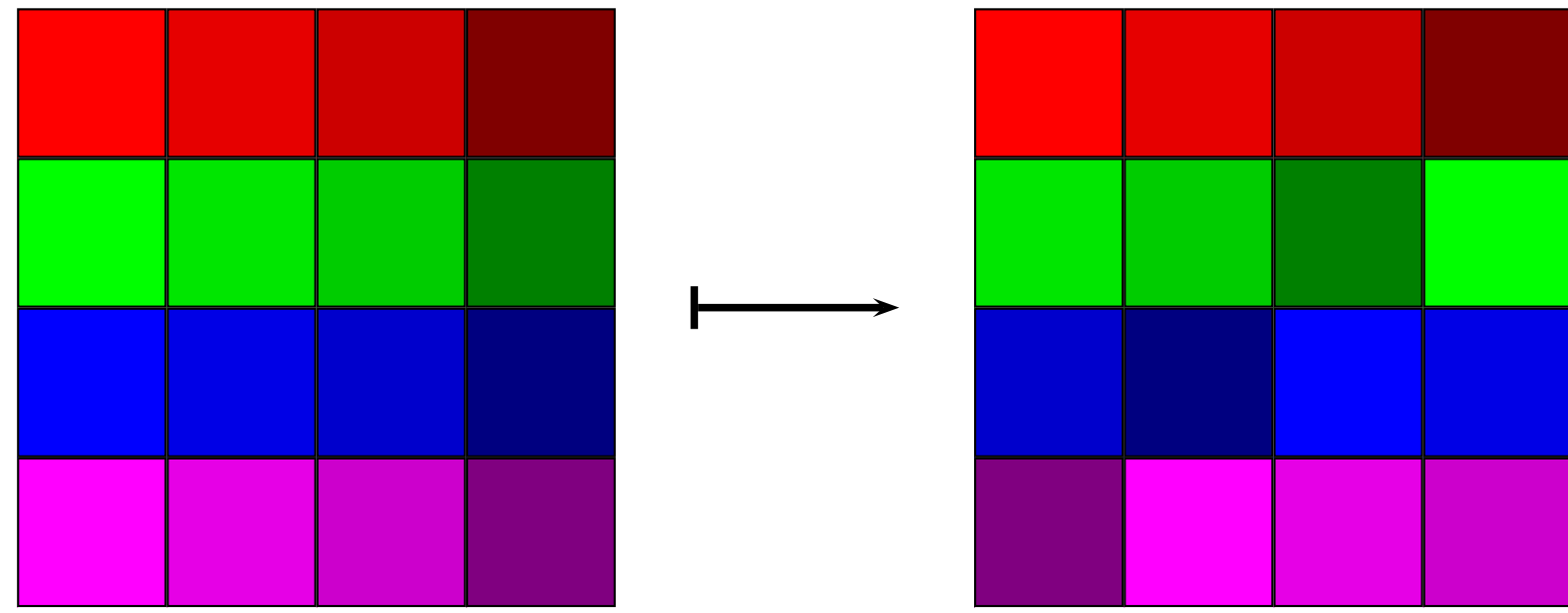
Multiplication: as for polynomials modulo  $x^8 + x^4 + x^3 + x + 1$ .

### Example $57 \cdot 83 = C1$ :

$$\begin{aligned} (x^6 + x^4 + x^2 + 1) \cdot (x^7 + x + 1) &= x^{13} + x^{11} + x^9 + x^8 + x^7 + \\ &\quad x^7 + x^5 + x^3 + x^2 + x + \\ &\quad x^6 + x^4 + x^2 + 1 \\ &= x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1 \\ &= x^7 + x^6 + 1 \pmod{x^8 + x^4 + x^3 + x + 1}. \end{aligned}$$

**Field:** You can divide by every non-zero element.

### The ShiftRows operation



The rows are shifted cyclically by zero, one, two, or three bytes.

### Polynomials over the field $\mathbb{F}_{2^8}$

$R = \mathbb{F}_{2^8}[z]/(z^4 + 1) \ni a_0 + a_1z + a_2z^2 + a_3z^3$ ,  
where  $a_i \in \mathbb{F}_{2^8}$ .

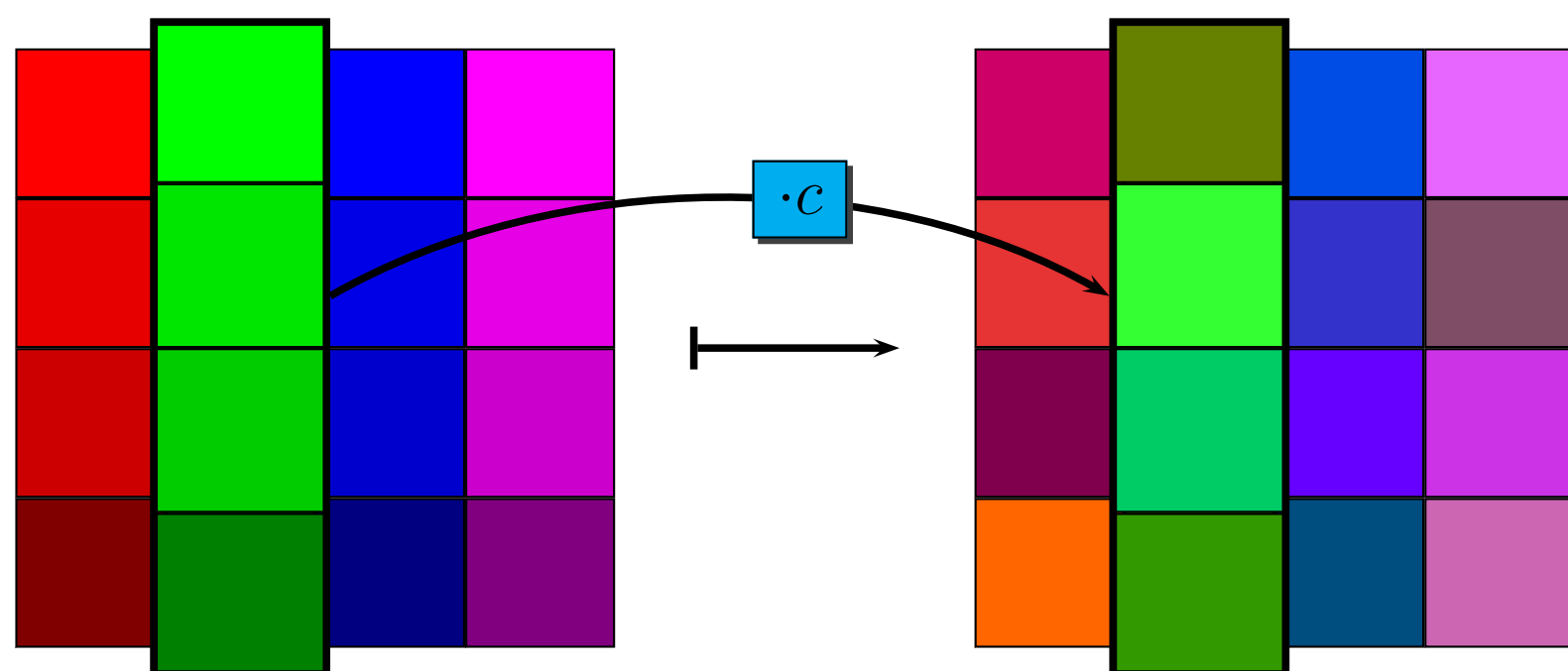
Addition: coefficient-wise  $(a + b)_i = a_i + b_i$ , XOR.

Multiplication: as for polynomials modulo  $z^4 + 1$ . Another way to express  $d = a \cdot b$  is by the following matrix equation:

$$\begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} a_0 & a_3 & a_2 & a_1 \\ a_1 & a_0 & a_3 & a_2 \\ a_2 & a_1 & a_0 & a_3 \\ a_3 & a_2 & a_1 & a_0 \end{bmatrix} \cdot \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

**Not a field:**  $(z + 1)^4 = 0$ .

### The MixColumns operation



Each column is considered as a polynomial and multiplied by  $c = 02 + 01z + 01z^2 + 03z^3$ .

Inverse: Multiply with  $d = 0E + 09z + 0Dz^2 + 0Bz^3$ .

### The S-box

$$\mathbb{F}_{2^8} \longrightarrow \mathbb{F}_{2^8} \longrightarrow \mathbb{F}_{2^8},$$

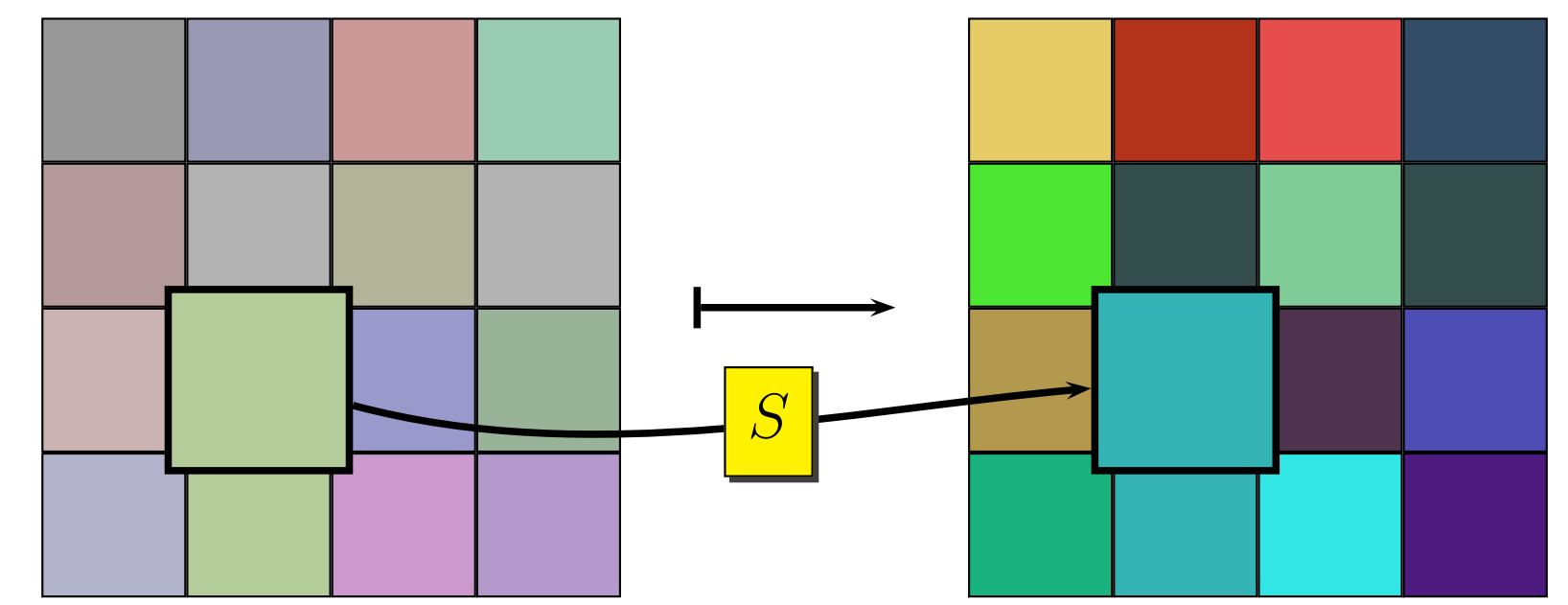
$$S: y \mapsto y^{-1} \doteq \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{bmatrix} \mapsto \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Highly nonlinear:

$$y \mapsto 05 \cdot y^{254} + 09 \cdot y^{253} + F9 \cdot y^{251} + 25 \cdot y^{247} + F4 \cdot y^{239} + 01y^{223} + B5 \cdot y^{191} + 8F \cdot y^{127} + 63.$$

Simple implementation using a 256 byte lookup table.

### The SubBytes operation



Apply the S-box to every byte.

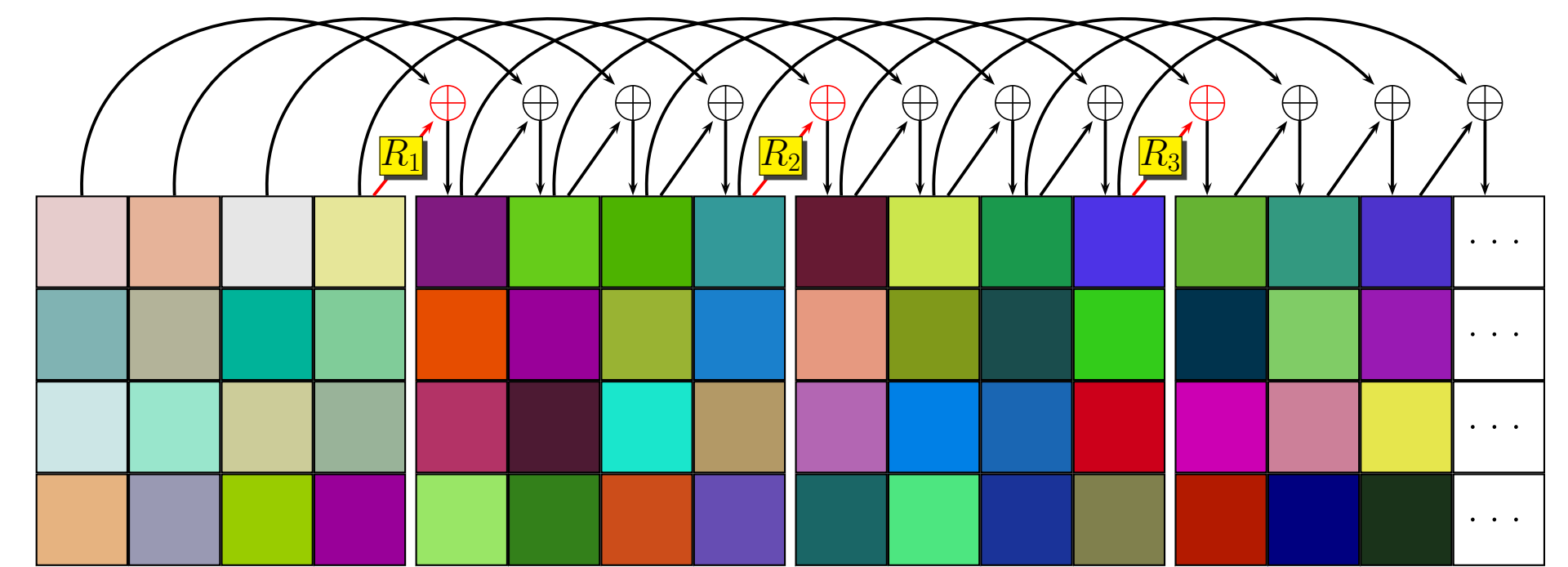
### Nonlinear part of the key schedule

$$(\mathbb{F}_{2^8})^4 \longrightarrow (\mathbb{F}_{2^8})^4,$$

$$R_i: \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \mapsto \begin{bmatrix} S(b) + x^{i-1} \\ S(c) \\ S(d) \\ S(a) \end{bmatrix}$$

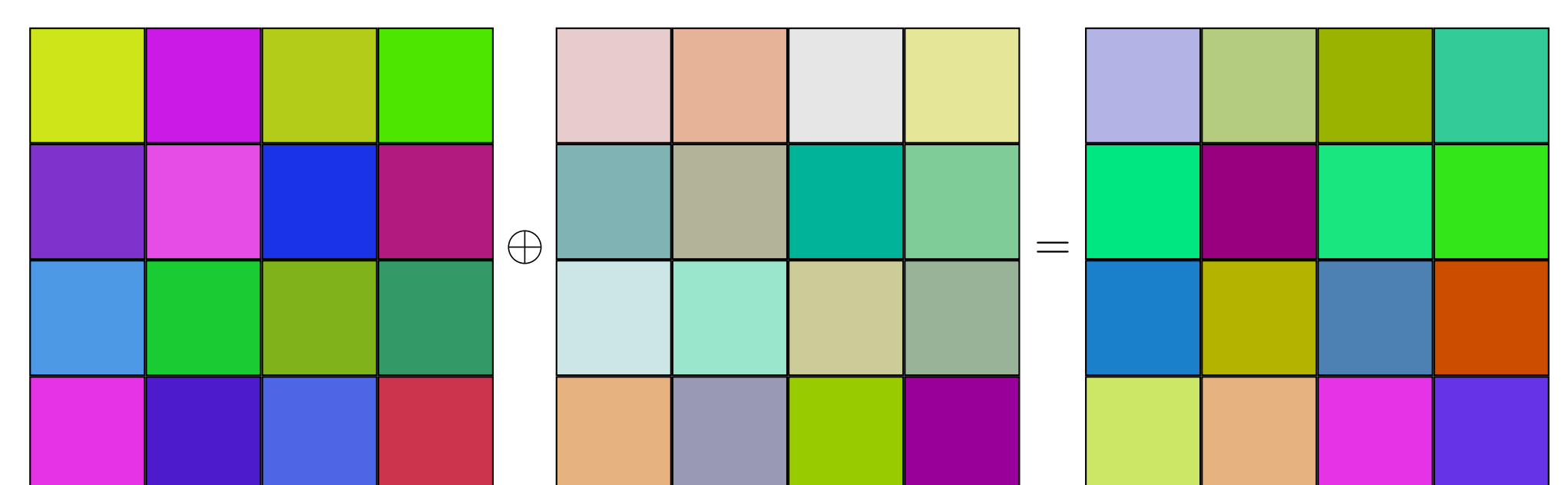
Due to the use of the S-box this map is non-linear.

### The Key Schedule



The round keys are generated from the 128 to 256 bit key.

### The AddRoundKey operation



Simple XOR with the round key.



## The field $\mathbb{F}_{2^8}$

$\mathbb{F}_{2^8} \ni a = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5 + a_6x^6 + a_7x^7$ ,  
where  $a_i \in \mathbb{F}_2 = \{0, 1\}$ .

Representation: 8 bits for an element = 1 byte.

Addition: XOR,  $(a + b)_i = a_i + b_i$ .

Multiplication: as for polynomials modulo  $x^8 + x^4 + x^3 + x + 1$ .

**Example**  $57 \cdot 83 = \text{C1}$ :

$$\begin{aligned}(x^6 + x^4 + x^2 + 1) \cdot (x^7 + x + 1) &= x^{13} + x^{11} + x^9 + x^8 + x^7 + \\ &\quad x^7 + x^5 + x^3 + x^2 + x + \\ &\quad x^6 + x^4 + x^2 + 1 \\ &= x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1 \\ &= x^7 + x^6 + 1 \pmod{x^8 + x^4 + x^3 + x + 1}.\end{aligned}$$

**Field:** You can divide by every non-zero element.

# The S-box

$$\mathbb{F}_{2^8} \longrightarrow \mathbb{F}_{2^8} \longrightarrow \mathbb{F}_{2^8},$$

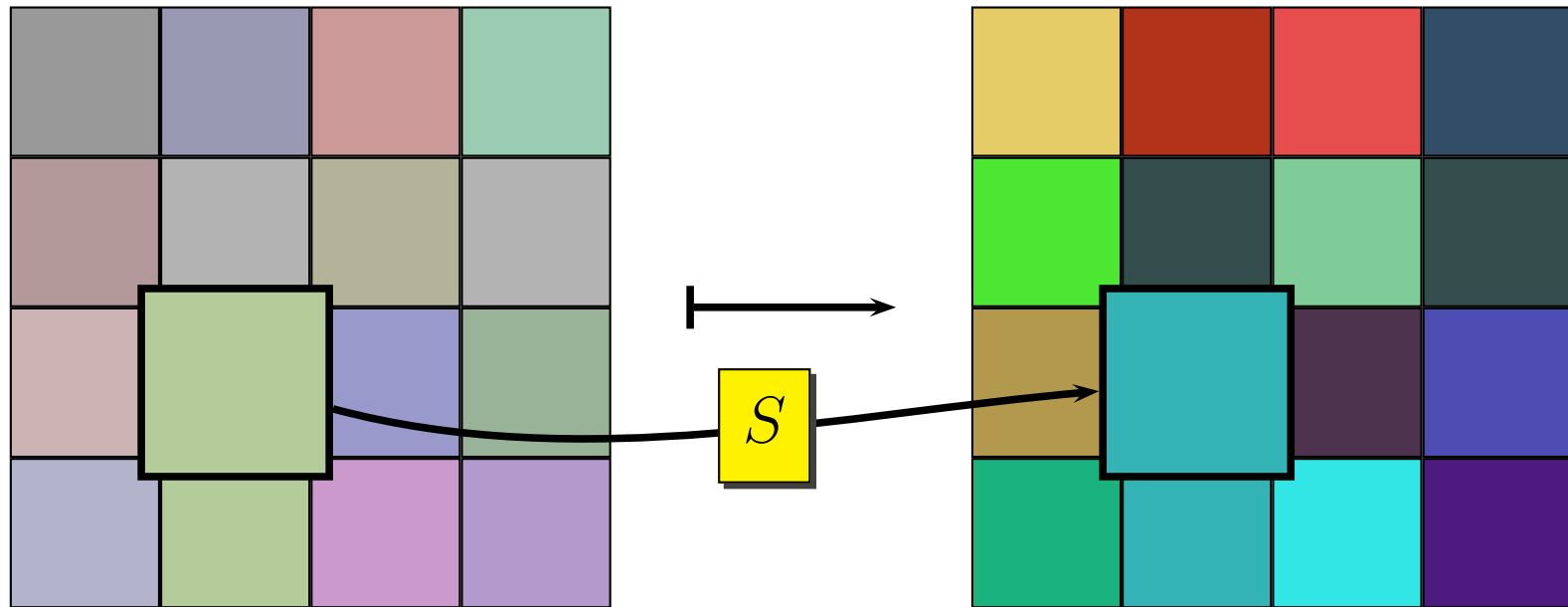
$$\boxed{S} : y \longmapsto y^{-1} \hat{=} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{bmatrix} \longmapsto \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Highly nonlinear:

$$y \mapsto 05 \cdot y^{254} + 09 \cdot y^{253} + F9 \cdot y^{251} + 25 \cdot y^{247} + F4 \cdot y^{239} + 01 y^{223} + B5 \cdot y^{191} + 8F \cdot y^{127} + 63.$$

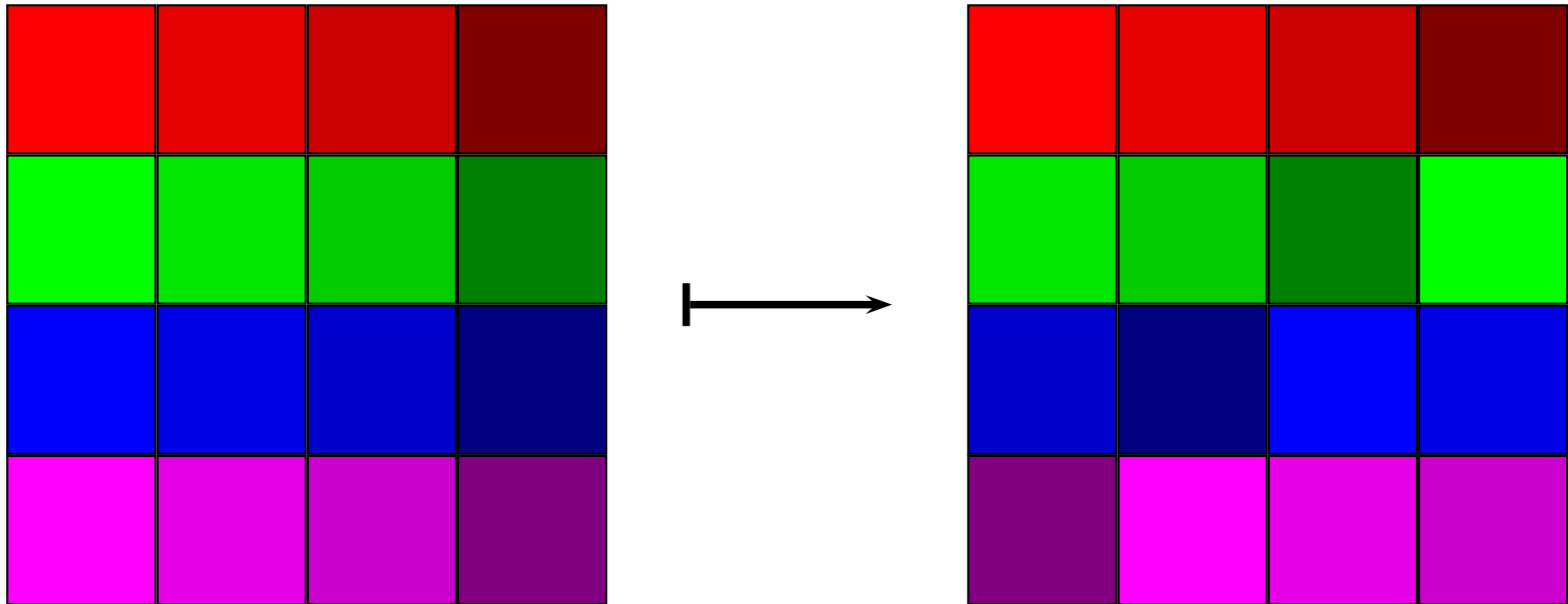
Simple implementation using a 256 byte lookup table.

# The SubBytes operation



Apply the S-box to every byte.

## The ShiftRows operation



The rows are shifted cyclically by zero, one, two, or three bytes.

## Polynomials over the field $\mathbb{F}_{2^8}$


$R = \mathbb{F}_{2^8}[z]/(z^4 + 1) \ni a_0 + a_1z + a_2z^2 + a_3z^3$ ,  
where  $a_i \in \mathbb{F}_{2^8}$ .

Addition: coefficient-wise  $(a + b)_i = a_i + b_i$ , XOR.

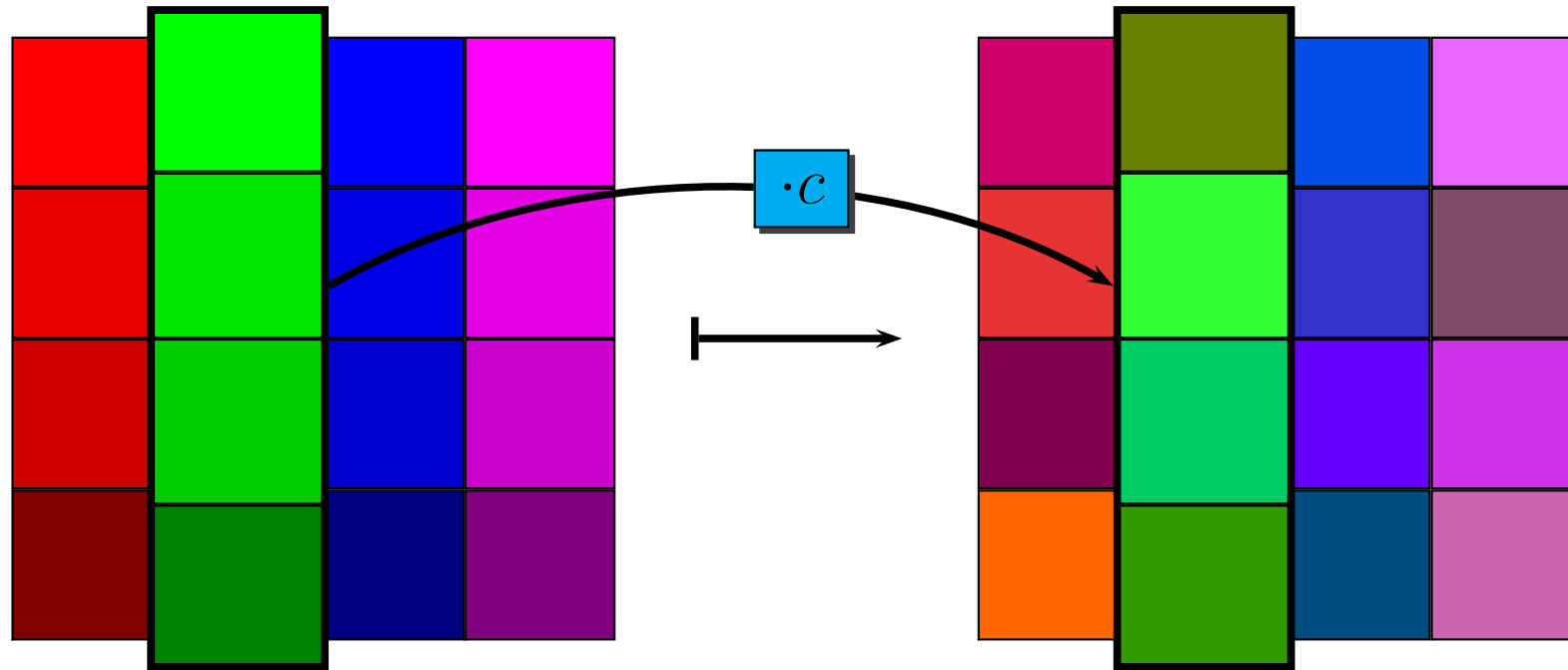
Multiplication: as for polynomials modulo  $z^4 + 1$ . Another way to express  $d = a \cdot b$  is by the following matrix equation:

$$\begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} a_0 & a_3 & a_2 & a_1 \\ a_1 & a_0 & a_3 & a_2 \\ a_2 & a_1 & a_0 & a_3 \\ a_3 & a_2 & a_1 & a_0 \end{bmatrix} \cdot \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Not a field:  $(z + 1)^4 = 0$ .



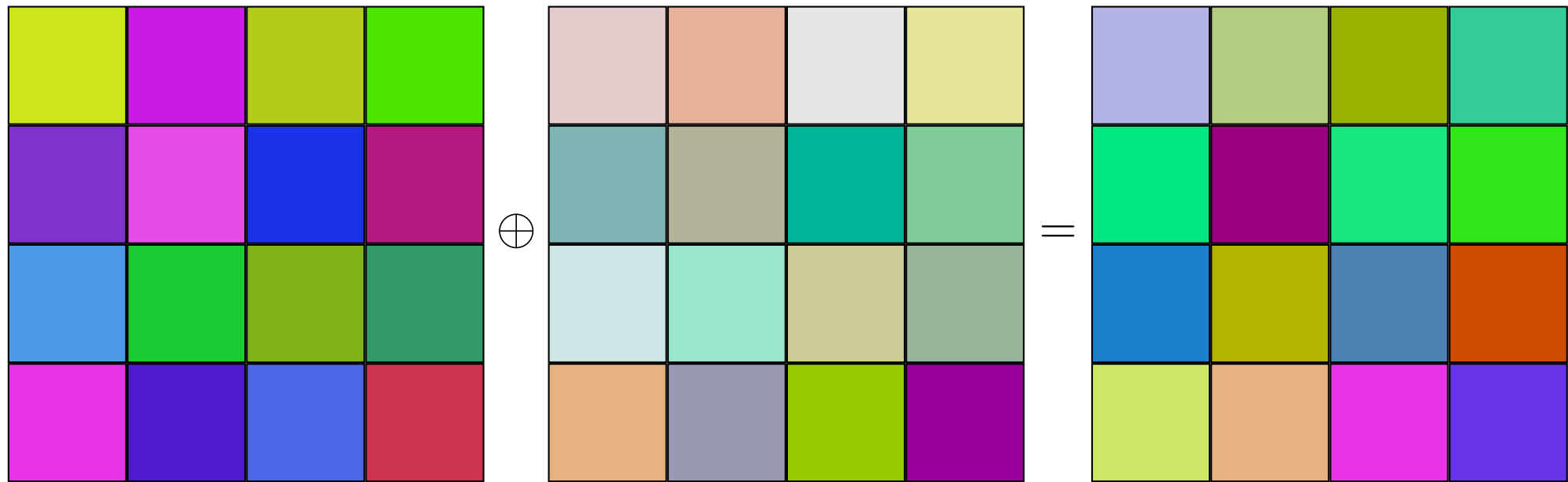
## The MixColumns operation



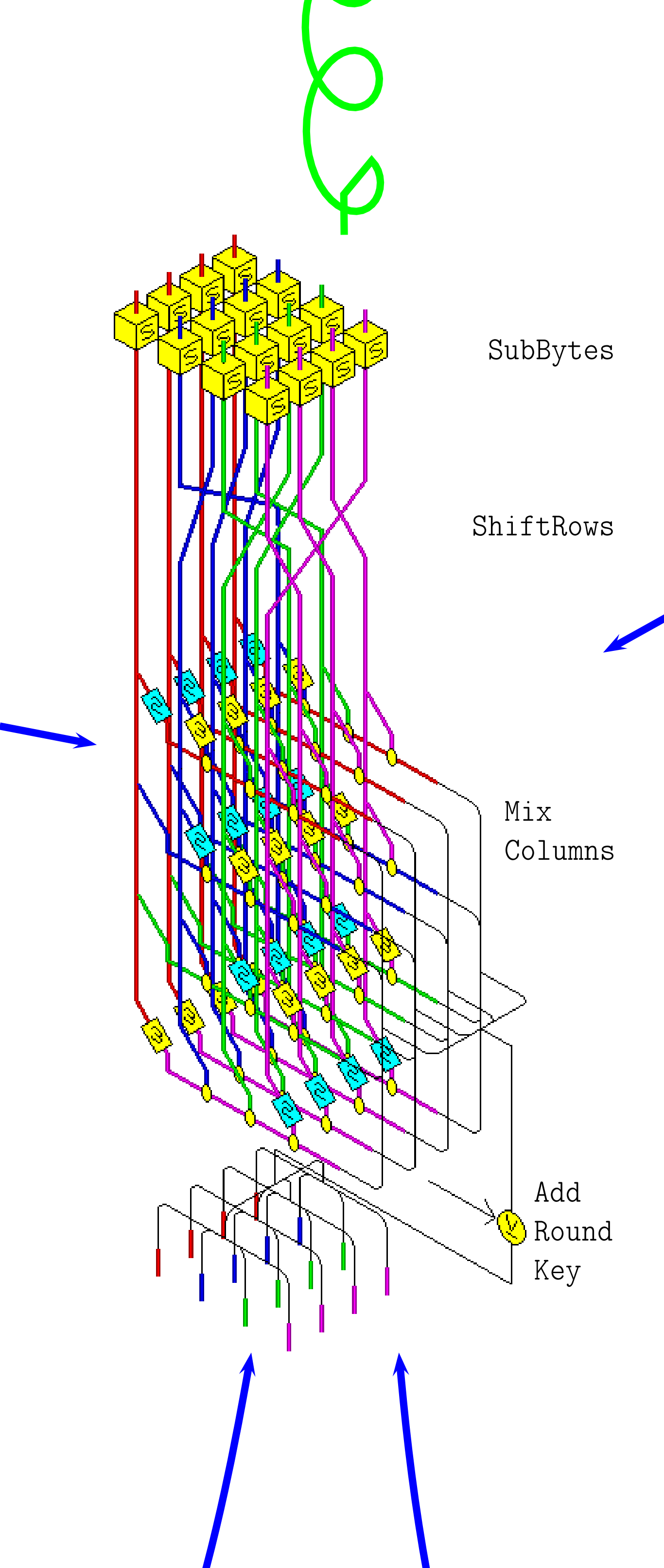
Each column is considered as a polynomial and multiplied by  $c = 02 + 01z + 01z^2 + 03z^3$ .

Inverse: Multiply with  $d = 0E + 09z + 0Dz^2 + 0Bz^3$ .

# The AddRoundKey operation



Simple XOR with the round key.

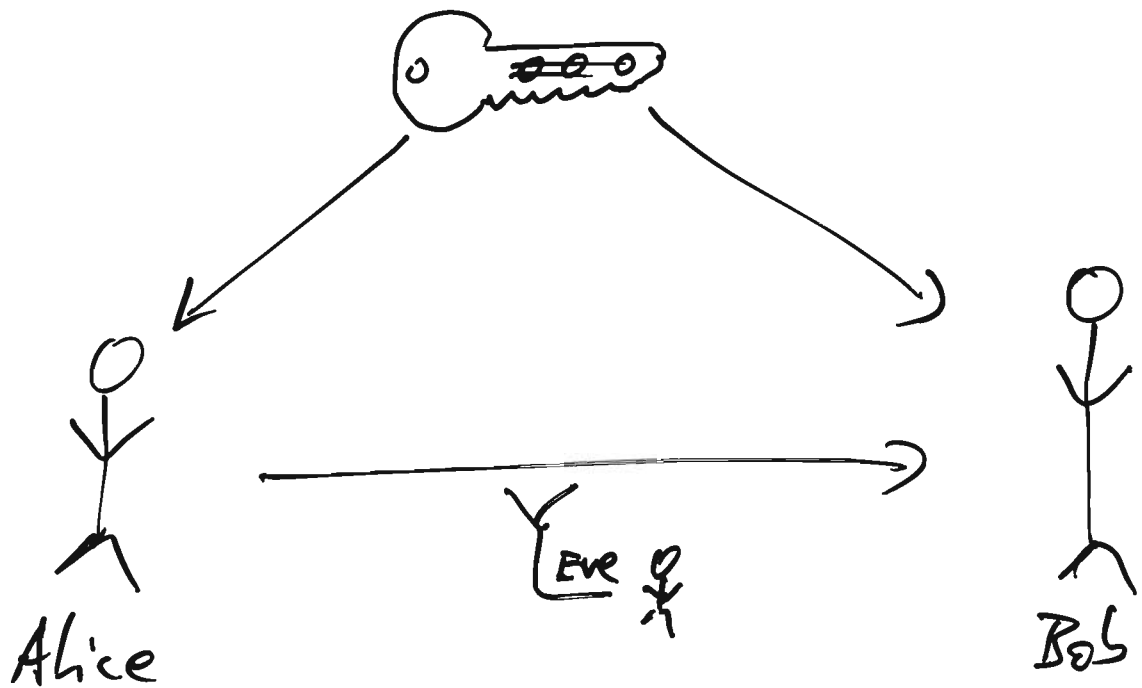




We have seen

- transposition ciphers
- substitution ciphers
  - mechanical & electrical devices for ciphers
  - computer programs, AES
  - ← various attacks.

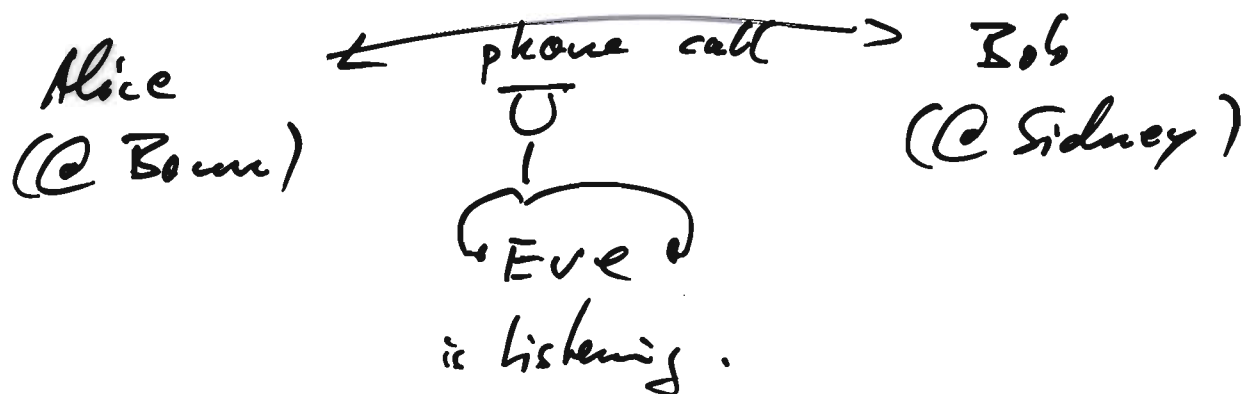
But all this just is in the situation,



symmetric  
encryption.

What if there is no safe way  
to send the secret key?

18.11.08  
SoFI  
①



Can that work?

Answers:

1971-74	British Secret Service	
	"Non-secret encryption"	
1976	Diffie & Hellman	} $\rightarrow$ public key crypto
1978	RSA	

We need a finite group, & say commutative!

For example:

$$\mathbb{Z}_p^x, \text{ or}$$

$$\mathbb{Z}_N^+, \text{ or}$$

$$\mathbb{F}_q^x, \text{ or}$$

$$GL_2(\mathbb{F}_q)^x = GL_2(\mathbb{F}_q) \text{ non-commutative...}$$

$$\mathbb{Z}_N^x, \text{ or}$$

All these are finite (comm.) groups

Remember: group = o.o. class of elements with 1 binary op. &  $\text{FAN}(C)$ .

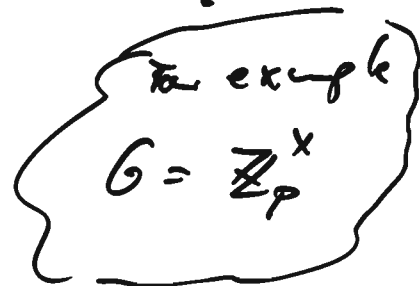
18.11.08  
Lot I  
(2)

So what?

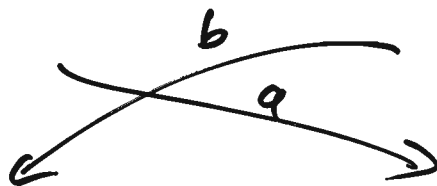
we can use multiplication,  
or repeated multiplication...

So: given  $g \in G$  where  $G$  is our group.  
and  $n \in \mathbb{N}$ .  
we can compute  $g^n$  in  $G$ .

greek letter  
nu  $\nu$



Alice  
 $\alpha \in_{\mathbb{R}} \mathbb{N}$   
 $a = g^\alpha$   
 $k_1 = b^\alpha$



Bob  
 $\beta \in_{\mathbb{R}} \mathbb{N}$   
 $b = g^\beta$   
 $k_2 = a^\beta$

Diffie  
Hellman  
key  
exchange

Actually:  $k_1 = k_2$  !

$$k_1 = b^\alpha = (g^\beta)^\alpha = g^{\beta \cdot \alpha}$$

Need the exponentiation law  $\nearrow$   $\forall$   $g^{\alpha\beta} = (g^\alpha)^\beta = a^\beta = k_2$

CORRECT?

Yes: we define correctness here to mean that  $k_1 = k_2$ .

Sot I  
18.11.08  
(3)

EFFICIENT?

Yes, by square & multiply

By example:

Give  $g$  in a group  $G$ ,

$x \in \mathbb{N}$ .

Compute  $g^x$ .

1. Write  $x$  in binary, say:

$(\alpha_{r-1}, \alpha_{r-2}, \dots, \alpha_0)$  with  $\alpha_i \in \{0, 1\}$ .

[to be precise:  $\sum_{i=0}^{r-1} \alpha_i 2^i = x$ .]

2.  $h \leftarrow g$ .

3. FOR  $i$  from  $r-2$  down to 0 DO

4.      $h \leftarrow h^2$ .

5.     if  $\alpha_i = 1$  then  $h \leftarrow h \cdot g$ .

6. End FOR

7. Return  $h$ .

Theorem     square & multiply  
(i) does compute  $g^x$ .

(ii) needs ~~at most~~  $r-1$  squarings  
and at most  $r-1$  multiplications.

Fact     It can be proved that we need  
(in a suitable sense) at least  $r-1$  squarings.

Corollary

If the group operation is efficient  
(as e.g. in  $\mathbb{Z}_p^*$  we have  $O(n^2)$   
bit operations when  $n = \#bits(p)$ )

SotI  
18.11.08  
(4)

The exponentiation using square & multiply  
is efficient.

So together: Diffie & Hellman  
is efficient.

SECURITY?

What does Eve know?

$G, g, a, b$  and the algorithm.

What does Eve want?

$$k := k_1 = k_2$$

Diffie - Hellman - problem for  $G$

Given  $g, a, b$  in ~~a fixed~~  $G$ .

Compute  $k = g^{a \cdot b}$  when  $a = g^x, b = g^y$ .

DHP

What help would be useful?

well, give  $x$  such that  $a = g^x$ .

# Discrete Logarithm problem for $G$

SofI  
18.11.08  
(5)

(DLP) Given  $g, a$  in  $G$ .  
Compute  $x$  such that  $a = g^x$   
or report that no  $x$  exists.

Assume Eve can solve the DLP.

Then Eve can solve the DHP:

Algorithm

Input:  $g, a, b$ .

Output:  $k$

1. Call the DLP oracle to find  $x$   
with  $a = g^x$ .

2. Compute  $k \leftarrow b^x$ .

3. Return  $k$ .

Then

If the Diffie Hellman key exchange  
is secure

then the DLP in the used group  $G$   
must be difficult.

Proof: we just did that!

□

Breaking the DLP ...

or at least: how far can we get?

SotI  
11.08  
①

Task: give  $g, a$  in a group  $G$ ,  
find  $x$  such that  $a = g^x$   
(or tell that there is none.)  
we assume that  $g, a$ , and  $x$  each fit into  $n$  bits.

Trivial solution:

- Try out all  $x$ .

$\tilde{O}(2^n)$

- Try a random  $x$  and  
check whether it works.  
Repeat until successful.

$\tilde{O}(2^n)$

Intuition:

Repeat  
something  
Until condition

where the condition holds with  
probability  $p$  each time

then the expected runtime is:  $\frac{1}{p}$ .

⌈ You have to sum over the probabilities to exit after  
 $k$  rounds:  $\sum_{k=0}^{\infty} (1-p)^k \cdot p \cdot k$ . The rest is analysis ... ⌋

Better?

SotI  
13.11.08  
(2)

Try to write  $x = \alpha_1 \cdot q + \alpha_0$   
with  $q = 2^{n/2}$ .

Now: there are  $\frac{n}{2}$  bits in  $\alpha_0$   
and  $\frac{n}{2}$  bits in  $\alpha_1$

Baby step - Giant step - algorithm

Input:  $g, a$ .

Output:  $x$ .

1. Determine  $q \approx \Theta(2^{n/2})$ .

2. Compute a table of  
 $g^{\alpha_0}$

for  $\alpha_0 = 0, \dots, q-1$ .  
Sort it!  
3. Compute

$$c = a (g^{-q})^{\alpha_1}$$

for  $\alpha_1 = 0, \dots, 2q$

until  $c = g^{\alpha_0}$  for some  $\alpha_0$ .  
(table lookup!)

// Now:  $a (g^{-q})^{\alpha_1} = g^{\alpha_0}$   
ie  $a = g^{\alpha_1 q + \alpha_0}$

4. Return  $\alpha_1 \cdot q + \alpha_0$ .

$\tilde{O}(2^{n/2})$

$\tilde{O}(2^{n/2})$

~~$\tilde{O}(2^{n/2})$~~

Problem:  
huge  
table,  
size  $\approx 2^{n/2}$



## Birth day paradox

SotJ  
19.11.08  
(3)

The expected number of people to invite until a birthday collision occurs is  $O(\sqrt{\# \text{birthdays}})$ .

## Pollard-ρ

"Guess" ~~pairs~~ tuple  $t = (x, y, ag^x, g^x)$

until a collision occurs of the form:

$$ag^x = g^{x'}$$

Run time: expected  $O(\sqrt{2^n}) = O(2^{n/2})$

Memory: still  $O(2^{n/2})$

To improve this we make the choice of tuples less random:

Choose the first tuple to.

Fix a deterministic function mapping tuples to tuples.  $t_i = f(t_{i-1})$ .

And consider the sequence  $(t_i)$ .

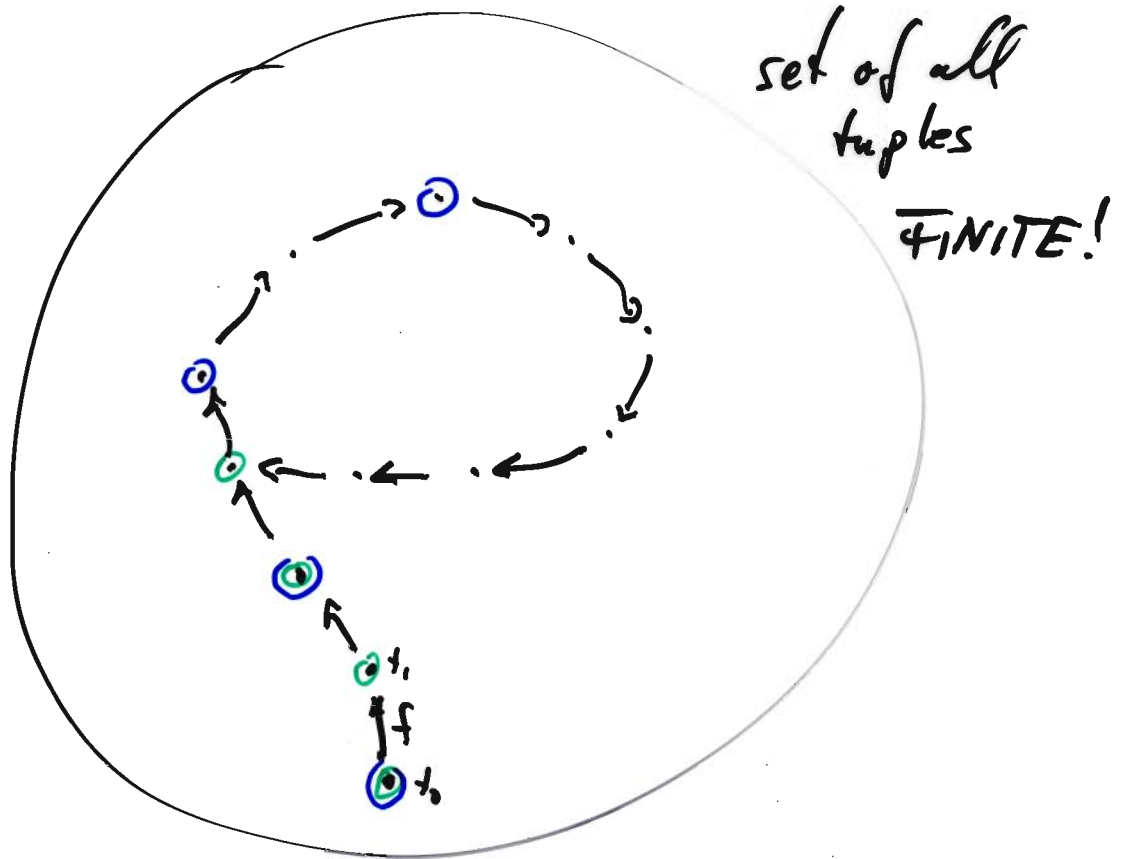
Now: implement Floyd's trick:

Consider two sequences:

$$(1) \quad t_0, t_1, t_2, \dots, \quad t_i = f(t_{i-1}).$$

$$(2) \quad s_0 = t_0, s_1, s_2, \dots, \quad s_i = f(f(s_{i-1})).$$

Idea:



Now only compare  $s_i$  to  $t_i$ .

$$\text{Note that } s_i = f^{(2i)}(s_0),$$

$$t_i = f^{(i)}(t_0)$$

$$\text{so } s_i = f^{(i)}(t_i).$$

Now if there is a collision then it must occur between  $s_i$  and  $t_i$

for same index. So we only need to store 2 elements.

Runtime? No idea.

Expected runtime? No idea.

Heuristic expected runtime:  $O(2^{n/2})$

Advantage: very small  
storage requirements.

Even better?

No: One can prove a lower bound that within a group  $G$  with  $\#G$  elements every randomized algorithm needs  $\Omega(\sqrt{\#G})$  operations in  $G$  on average...

if you do not use anything about the special structure of your group.

For example: in  $\mathbb{Z}_p^*$  there is an algorithm using  $O(\sqrt{n} \log n)$  steps. It's still not polynomial. 2

SotJ  
13.11.08  
(5)

So  $\mathbb{Z}_7^*$  is not that nice? j

SfI  
19.11.08  
(6)

There is a particularly well-suited family of groups where no attacks essentially better than Baby step - Giant step or Pollard- $\rho$  are known.

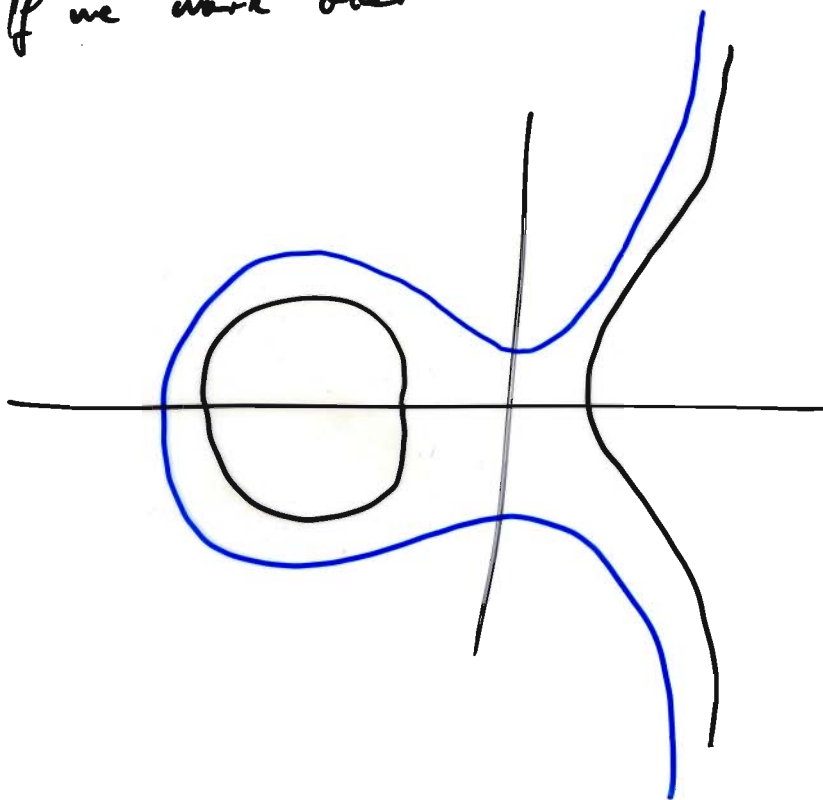
Interlude: Elliptic curves

Consider an equation of the form

$$y^2 = x^3 + ax + b$$

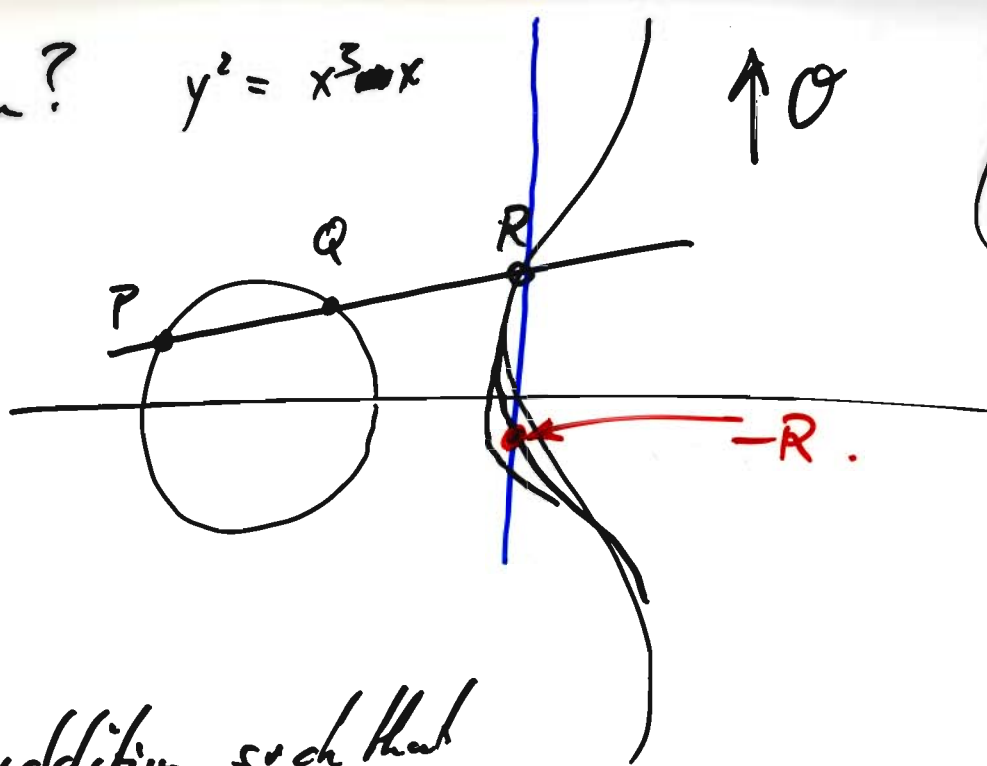
in two unknowns  $x, y$ .

If we work over  $\mathbb{R}$ :



Addition?  $y^2 = x^3 - x$

SotI  
19.11.08  
(7)



Define addition such that

$$P + Q + R = O$$

So define

$$P + Q := -R.$$

Let

$$x = x_0 + x_1 \cdot \lambda$$

$$y = y_0 + y_1 \cdot \lambda$$

$$\downarrow$$
$$y^2 = x^3 - x$$

gives a  
cubic equation  
for  $\lambda$ .

Want  $R + (-R) = O$

ie  $R + (-R) + (-O) = O$

Where is the line through  
 $R, -R,$  and  $O = -O$ ?

We expect:  $-O = O$ .

... so if  $R = (x_3, y_3)$

$$\text{then } -R = (x_3, -y_3).$$

Replace  $\mathbb{R}$  by a finite field!

Then the curve is finite and that's what to use.

Recall

SofI  
25.11.08

(4)

# Diffie - Hellman

Alice

$$\alpha \in \mathbb{Z} \neq$$

$$a = g^\alpha$$

$$k_1 = b^\alpha$$

$$(g \in G)$$

Bob

$$\beta \in \mathbb{Z} \neq$$

$$b = g^\beta$$

$$k_2 = a^\beta$$

Note:  $a, b, k_1, k_2$  are in  $G$   
and must be calculated  
using the operation(s) in  $G$ !

Example  $G = \mathbb{Z}_p^\times$ ,  $p$  prime

$$g = ?$$

Obvious:  $g=1$  is bad!

Well, what is good?

Necessary: the number of possible  
outcomes for powers of  $g$   
must be large!

(That's the number of keys!)

Define:  $\text{ord } g := \# \{ g^i \mid i \in \mathbb{Z} \}$   
Order of  $g$   $\langle g \rangle$

the subgroup of  $G$   
generated by  $g$

Then

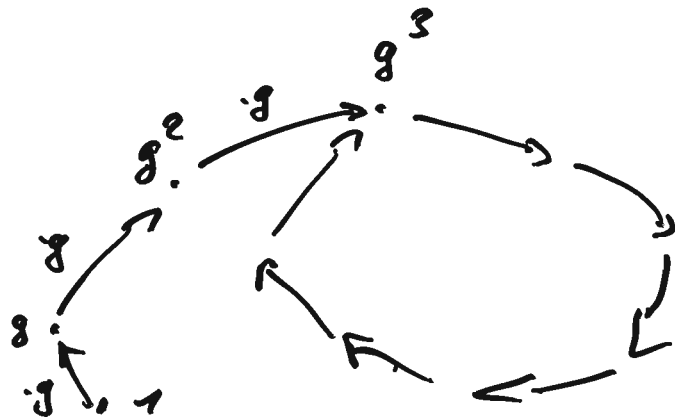
and  $g = \text{mitte } \{ i \in \mathbb{N}_{>0} \mid g^i = 1 \}$ .

(when  $\min \emptyset = \infty = \# \mathbb{Z}$ )

50+J  
25.11.08  
(2)

observation: why should  $g^i = 1$  be ever fulfilled  
for  $i > 0$ ? Sag, if the group is finite.

Consider the picture:



Since  $G$  is finite, we must have

$$g^i = g^j$$

for same  $i, j$ , say  $i > j$ .

(Pigeon hole principle)

Then:  $g^{i-j} = g^{j-j} = 1.$

because  $G$  is a group and thus we can divide by  $g^j$ .

Now with this we have, assuming  $g' = \tau, \epsilon > 0$ :

$$\langle g \rangle = \{ 1, g, g^2, \dots, g^{i-1} \}$$



Proof (Thm)

Assume

$$n = \min \{ i \mid g^i = 1 \}.$$

SotI  
25.11.08  
(2)

Then  $g^n = 1.$

And thus

$$\langle g \rangle = \{ 1, g, g^2, \dots, g^{n-1} \}.$$

Obviously:  $\geq \checkmark$

$\leq$ : Take  $g^j, j \in \mathbb{Z}.$

write  $j = q \cdot n + r$

with  $0 \leq r < n.$

then  $g^j = g^{qn+r}$   
 $= (g^n)^q g^r = g^r$   
 $\in \text{rhs.}$

Thus  $\text{ord } g \leq n = \min \{ i > 0 \mid g^i = 1 \}.$

Consider again (\*). Assume that  $\# \langle g \rangle < n.$

then for some  $0 \neq i \neq j < n$  we have

$$g^i = g^j.$$

Say  $i > j$ . then  $g^{i-j} = 1.$

But  $0 < \underline{i-j} < n$  contradicting

$$n = \min \{ i \in \mathbb{N}_{>0} \mid g^i = 1 \}.$$

So our assumption is false, and  $\# \langle g \rangle \geq n. \quad \square$

Go back to choosing  $g \in \mathbb{Z}_p^*$ ...

Aim: Make the DLP difficult!

Set I  
25.11.08  
(4)

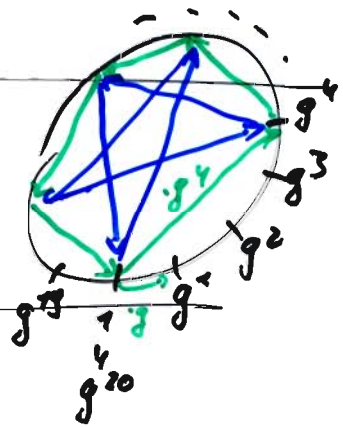
Exmp  $p = 2^3 \cdot 5 + 1 = 41$ .

as a candidate of a prime  
where  $p-1$  is a product  
of many small primes.

Take  $g = 2$ .

$$\mathbb{Z}_p = \{0, 1, 2, \dots, 20, -20, -19, \dots, -1\}.$$

i	0	1	2	3	4	5	6	7	8	9	10
$g^i$	1	2	4	8	16	-9	-18	5	10	20	-1
i	10	11	12	13	14	15	16	17	18	19	20
$g^i$	-1	-2	-4	-8	-16	9	18	-5	-10	-20	1



Thus: ord  $g = 20 = 2^2 \cdot 5$ .

Let's try to find  $\alpha$  such that  $2^\alpha = 5$ .

Brute force: ~~time~~ time  $O(\#G) = O(2^u)$

where  $u$  = size of storage  
for an element of  $G$ .

(Remember: Brute force is no solution!)

We can simplify the task:

$$2^2 = 5,$$

SotI  
25.11.08  
(5)

well, we have 20 choices for  $x$  here.

Exponentiate:

$$(2^r)^x = 5^r$$

for some nice  $r$ .

For example with  $r=10$  we get:

$$(-1)^a = -1$$

Thus:  $x$  must be odd!

$$\begin{aligned} 5 &= 2^3 \\ 5^{10} &= (2^3)^{10} \\ &= (2^3)^2 \\ &= (-1)^2 = -1 \end{aligned}$$

Driving this to the limit  $\rightarrow$  Pohlig-Hellman.

It will turn out that in order to solve the DLP for the element 2

of order  $20 = 2^2 \cdot 5^1$

it is enough to solve

one DLP for an element of order 5,

two DLPs for an element of order 2.

So: A further condition on  $g$ , namely:

the order of  $g$   
must not be a product  
of small primes

SotI  
25.11.09  
(6)

Back to our ~~the~~ group choice for Diffie & Hellman:

Choose  $\mathbb{Z}_p^*$

and  $g \in \mathbb{Z}_p^*$

such that  $q := \text{ord } g$

is a large prime.

Idea 0: . Choose  $p$  prime, sufficiently large.  
(say 2024 bits).

. Pick  $h \in \mathbb{Z}_p^*$ .

. Determine  $\text{ord}(h)$ . ← Difficult.

... and hope that it is a large prime.

I

↗ Improbable.

Solution : . Choose  $q$  prime, sufficiently large  
(say 200 bit).

. Choose  $p$  prime, so that  $\mathbb{Z}_p^*$  has elements  
of order  $q$ .

. Pick  $h \in \mathbb{Z}_p^*$  and let  $g \leftarrow h^{\frac{p-1}{q}}$ .

We already know that  
 given  $g \in G$ ,  $G$  finite group,

for some  $i > 0$  we find  $g^i = 1$ .

Question: Which orders can occur?

Back to an example:  $\mathbb{Z}_{41}^\times$ .

$g$	1	2	3	4	5	6	7	8	9	10	
ord $g$	1	20	40	10	20			20	4	5	...

Note:  $\# \mathbb{Z}_{41}^\times = 40$ .

All orders - so far - divide 40.

Theorem (Little Fermat Theorem)

Given  $a \in \mathbb{Z}_p^\times$ ,  $p$  prime.  
 Then  $a^{p-1} = 1$ .

Theorem (Lagrange)

Given  $G$  a finite group, and  $a \in G$ .  
 Then  $a^{\#G} = 1$ .

SotI  
 25.11.08  
 (7)

Sketch:

List all group elements

$$g_0, g_1, g_2, \dots, g_{x-1}$$

with  $x = \#G$  (all distinct and the list is complete.)

Multiply with  $a$ :

$$ag_0, ag_1, ag_2, \dots, ag_{x-1}$$

Multiply each list:

$G$  commutative

$$\underbrace{g_0 \cdot g_1 \cdot \dots \cdot g_{x-1}}$$

$$=$$

$$ag_0 \cdot ag_1 \cdot \dots \cdot ag_{x-1}$$

$$a^{\#G} \cdot \underbrace{g_0 \cdot \dots \cdot g_{x-1}}$$

so

...

---

To Do

proof Lagrange ✓

derive corollaries ✓ update DH. ✓

and  $(g^k) = ?$

Pohlig-Hellman again

CRT

SoFI  
25.11.08  
②

Proof Assume additionally:  $G$  commutative.  
Consider a list\* of all group elements:

SotI  
25.11.08  
(7)

$$g_0, g_1, \dots, g_{\gamma-1}$$

where  $\gamma = \#G$ . \* = list means: no repetitions,  
no omissions.

Now, multiply by  $a$ :

$$ag_0, ag_1, \dots, ag_{\gamma-1}$$

Obvious: all these are group elements.

Routine: no repetitions,  
no omissions.

no repetitions:

Assume to the contrary that  $ag_i = ag_j$

for some  $i \neq j$ ,  $i, j \in \{0, \dots, \gamma-1\}$ .

Multiply by  $a^{-1}$  and obtain  $g_i = g_j$ .

Since the first list has no repetitions

we must have  $i = j$ .  $\gamma$ .

no omissions

Take any group element, say  $g_i$ .

Find  $j$  with  $g_i = ag_j$ .

To do so find  $j$  such that  $g_j = a^{-1}g_i$ .

Now:  $g_i = ag_j$ . ✓

Thus the two lists are equal  
up to order.



Thus

$$\prod_{i \in S} g_i = \prod_{i \in S} (a g_i)$$

Sot I  
26.11.08  
(2)

since  $G$  is commutative...  
(if)

Now, divide by the l.h.s.:

$$1 = a^r \cdot \frac{\prod_{i \in S} g_i}{\underbrace{\prod_{i \in S} g_i}_1} = a^{\#G}$$

□

Example cont.

$$G = \mathbb{Z}_{41}^{\times}, \quad g = 2. \rightarrow \text{ord } g = 20, \\ \#G = 40.$$

Lagrange tells us:

$$g^{40} = 1.$$

we checked:

$$g^{20} = 1. \Leftrightarrow (g^{20})^2 = 1$$

Corollary

Given a finite group  $G$ , and  $a \in G$ .

Then

$$\text{ord } a \mid \#G.$$

Proof

Assume it's wrong.

Write

$$\#G = g \cdot \text{ord } a + r, \quad 0 < r < \text{ord } a$$

Now:

$$1 = a^{\#G} = \underbrace{(a^{\text{ord } a})}_1 a^r = a^r \neq 1 \text{ (since } r < \text{ord } a).$$

□

What happens if we apply the Theorem (Lagrange) to the unit group  $\mathbb{Z}_N^\times$  of the ring of integers modulo  $N$ ? Sol I  
26.11.08  
(3)

### Theorem (Euler)

Given  $N \in \mathbb{N}_{\geq 2}$ ,  $a \in \mathbb{Z}_N^\times$  i.e.  $\gcd(a, N) = 1$ .

Then  $a^{\varphi(N)} = 1$

where  $\varphi(N) := \# \mathbb{Z}_N^\times$

↑ This is called the Euler totient function.  $\square$

Further we can restrict  $N$  to primes:

### Theorem (Little Fermat Theorem)

Given  $p$  prime,  $0 < a < p$ .

Then  $a^{p-1} = 1$  in  $\mathbb{Z}_p^\times$ .

Proof well, specialize Euler to  $N = p$  prime, or Lagrange to  $\mathbb{Z}_p^\times$ .  $\square$

For Diffie & Hellman the most central building block is the exponentiation: Set I  
26.11.08  
(4)

$$\begin{array}{ccc} \mathbb{Z} & \longrightarrow & G \\ k & \longmapsto & g^k \end{array}$$

We know now that  $g^{\#6} = 1$   
and  $\langle g \rangle = \{1, g, g^2, \dots, g^{\#6-1}\}$ .

So when computing

$$g^{\alpha}$$

we can first reduce  $\alpha$  modulo  $\#6$ !

write

$$\alpha = q \cdot \#6 + s, \quad 0 \leq s < \#6$$

then

$$\begin{aligned} g^{\alpha} &= \underbrace{(g^{\#6})^q}_{=1} g^s \\ &= g^s. \end{aligned}$$

Thus we get a map

$$\begin{array}{ccc} \mathbb{Z}_{\#6} & \longrightarrow & G \\ \alpha & \longmapsto & g^{\alpha} \end{array}$$

$\text{exp}_g^{\#6}:$

Note: whichever  $k \in \mathbb{Z}$  you choose with  $\alpha = k \bmod \#6$  the

the element  $g^k$  is always the same!

So 45  
28.11.08  
(5)

$$\begin{array}{ccc} \alpha & \xrightarrow{\quad} & g^\alpha \\ \beta & \xrightarrow{\quad} & g^\beta \\ \alpha + \beta & \xrightarrow{\quad} & g^{\alpha+\beta} \stackrel{!}{=} g^\alpha \cdot g^\beta \end{array}$$

Variant: consider a <sup>homomorphism</sup> group element  $g$  of order  $l$ :  $g^l = 1$   
Then we obtain similarly:

This is also a homeomorphism  
but additionally bijective!

- computing  $exp_g$  is easy  $\rightarrow$  Square & Multiply
- computing  $exp_g^{-1}$  is sometimes probably difficult!  $\rightarrow$  Discrete Logarithm Problem

So the correct Diffie & Hellman  
key exchange  
is this:

SotJ  
26.11.08  
⑥

Setup:  $G$  a finite group,  
 $g \in G$  an element  
of large prime order  $e$

Alice

$$\alpha \in_R \mathbb{Z}_e$$

$$a = g^\alpha$$

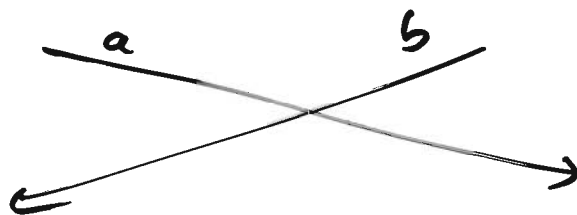
$$k_1 = b^\alpha$$

Bob

$$\beta \in_R \mathbb{Z}_e$$

$$b = g^\beta$$

$$k_2 = a^\beta$$



Eve wants to compute  $k = k_1 = k_2$  given  $g, a, b$ . } Diffie Hellman Problem

She can do so easily if she can solve  
the discrete logarithm problem.

We would like to prove however:

| if Eve can solve DHP  $\Rightarrow$  Eve can solve DLP for  $g$ .

But that's unknown (if not known to be wrong).

Next task:

Suppose we know  $\text{ord}(g)$ .

Compute  $\text{ord}(g^k)$ .

SotI  
26.11.08  
(7)

Side remark: since it is difficult to  
compute orders  
this may be important.

Example cont

In  $\mathbb{Z}_{21}^*$  we have  $\text{ord } 2 = 20$ .

And we have found that from that

we can determine  $\text{ord } 2^5 = \frac{20}{5} = 4$ .

or  $\text{ord } 2^4 = \frac{20}{4} = 5$ .

But  $\text{ord } 2^8 = 5 \neq \frac{20}{8} = 2.5$

$$\frac{20}{\gcd(8, 20)}$$

Theorem  $\text{ord}(g^k) = \frac{\text{ord } g}{\gcd(k, \text{ord } g)}$

Proof Case 1  $k \mid \text{ord } g$ , ie.  $\gcd(k, \text{ord } g) = k$

Case 2  $\gcd(k, \text{ord } g) = 1$ .

Case 3 general: put together...

Case 1  $l := \text{ord } g$

SotI  
26.11.08  
(P)

~~obvious~~  
We have

$$g^l = 1$$

$$\text{and } l = a \cdot b.$$

$$\text{Thus } (g^b)^a = 1$$

$$\text{and so } \text{ord}(g^b) \leq a.$$

Further, take  $0 < b < a$ .

then

$$(g^b)^b = g^{\underbrace{b^2}_{< l = \text{ord } g}} \neq 1. \quad \left. \begin{array}{l} \text{ord}(g^b) \\ \geq a. \end{array} \right\}$$

Case 2

We have

$$g^l = 1,$$

we have

$$1 = s \cdot b + t \cdot l$$

Bézout equation

for some  $s, t$   
using the EEA.

well... let's try:

$$g = g^1 = g^{s \cdot b} (g^l)^t = g^{s \cdot b} = (g^b)^s.$$

Assume that

$$(g^b)^a = 1$$

then

$$\underbrace{(g^{bs})^a}_{=g} = 1^s$$

so

$a$  must be at least  $\text{ord } g$ .

$$\text{And so } \text{ord } g^b = \text{ord } g.$$

Case 3 left to you...

□

# Pohlig & Hellman again

SofI  
26.11.08  
(8)

Given  $g$  in some group  
of order  $e = \prod p_i^{e_i}$

and  $a \in \langle g \rangle$ .

Find  $x \in \mathbb{Z}_e$  such that  $g^x = a$

Take  $k \mid e$  then

$$(g^k)^x = a^k$$

and this determines  $x$  modulo  $\text{ord}(g^k) = \frac{e}{k}$ .

From that we can obtain partial answers  
on  $x$ !

Start with  $k = \frac{e}{p_i}$  then

$$(g^k)^x \stackrel{?}{=} a^k$$

order  $p_i$  DLP

determines  $x$  modulo  $p_i$ .

Now, write  $x = \alpha_1 \cdot p_i + \alpha_0$ . By this

we know  $\alpha_0$ . Take  $k = e/p_i^2$  then

$$(g^{kp_i})^{\alpha_1} g^{k\alpha_0} = (g^k)^{\alpha_1 \cdot p_i + \alpha_0} = a^k$$



Solve this:

$$(g^{kp_i})^{\alpha_1} \stackrel{(1)}{=} a^k g^{-k\alpha_0}$$

SotI  
26.11.08  
(10)

and observe  $\text{ord } g^{kp_i} = \text{ord } g^{\frac{ep_i}{p_i^2}}$

$$= \text{ord } g^{e/p_i} = p_i$$

order  $p_i$  DLP

so we can obtain  $\alpha_1$  modulo  $p_i$ :

from (1)

Thus we know  $\alpha = \alpha_1 p_i + \alpha_0$  modulo  $p_i^2$ .

known  
partially  
mod  $p_i$

Next step: consider  $k = \frac{e}{p_i^3}$

and write

$$\alpha = \alpha_2 p_i^2 + \alpha_1 p_i + \alpha_0$$

... We can continue this as long as

$$\frac{e}{p_i^f} \in \mathbb{Z}, \text{ i.e. } f \leq e_i.$$

$$\uparrow$$

$$e = \prod p_i^{e_i}$$

So we obtain

Put this together to obtain  $\alpha$  modulo  $p_i^{e_i}$  for each index  $i$ , using CRT...

Repetition:  $G$  is a finite group.

SotI  
2.12.08  
④

Define

$$\text{ord } g := \# \underbrace{\{g^i \mid i \in \mathbb{Z}\}}_{\langle g \rangle}$$

(Order of  $g$ )

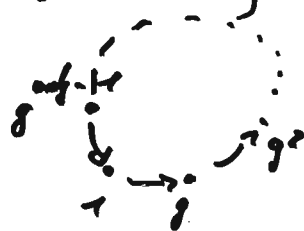
Then

$$\text{ord } g = \min \{i \in \mathbb{N}_{>0} \mid g^i = 1\}$$

subgroup  
generated  
by  $g$

Key fact:

$$\langle g \rangle = \{1, g, g^2, \dots, g^{\text{ord } g - 1}\}$$



Then

$$\text{ord } g \mid \# G$$

or  
Then (Lagrange)

$$g^{\#G} = 1$$

for any  $g \in G$ .

~~Little Fermat~~

Then (Euler)  $G = \mathbb{Z}_N^\times$  unit group of integers modulo  $N$ :

For  $0 < a < N$ ,  $\gcd(a, N)$

we have

$$a^{\varphi(N)} = 1 \quad \text{in } \mathbb{Z}_N^\times$$

where

$$\varphi(N) := \# \mathbb{Z}_N^\times.$$

Then (Little Fermat)  $G = \mathbb{Z}_p^\times$ ,  $p$  prime:



For  $p$  prime,  $0 < a < p$

we have

$$a^{p-1} = 1 \quad \text{in } \mathbb{Z}_p^\times$$

In other words:

$$p \mid a^{p-1} - 1 \quad \text{if } p \nmid a.$$

Corollary

For  $a \in \mathbb{Z}_p$  we have  $a^p = a$ . ( $p$  prime)

# Diffie-Hellman

SoFI  
2.12.08  
(2)

Setup: Fix a group  $G$  and an element  $g$  such that  $\text{ord } g$  is a large prime  $q$ . we work in  $\langle g \rangle$ .

Alice

$$\alpha \in \mathbb{Z}_q$$

$$a = g^\alpha$$

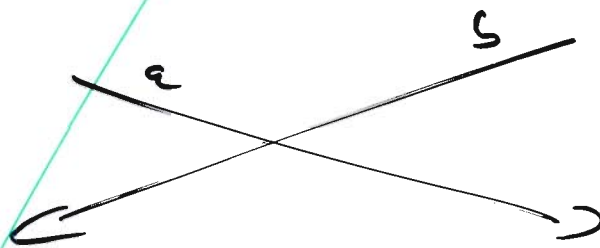
$$k_1 = a^\alpha$$

Bob

$$\beta \in \mathbb{Z}_q$$

$$b = g^\beta$$

$$k_2 = a^\beta$$



How to choose  $G$  and  $g$ ?

→ find discrete logarithm attacks:

- $\tilde{O}(\sqrt{q})$  • BabyStep-GiantStep :  $\tilde{O}(\sqrt{q})$
- $\tilde{O}(\sqrt{q})$  • Pollard- $\rho$  : heur. exp.  $\tilde{O}(\sqrt{q})$
- $\tilde{O}(\sqrt{\text{largest prime factor of } q})$  • Pollard-Hellman →  $\tilde{O}(\sqrt{\text{largest prime factor of } \text{ord } g})$
- • Index calculus for  $\mathbb{Z}_p^*$   $\tilde{O}(2^{\sqrt{\log p \cdot \log \log p}})$
- Note: nothing like index calculus known for elliptic curves.

How large should things be?

SofI  
2.12.08

(3)

(1)  $q$  should be prime  
to prevent Pohlig-Hellman  
from making discrete log easier

(2)  $q$  should be so large that

$\tilde{O}(\sqrt{q})$  is beyond scope  
of an attacker... Of course the  
constant is important then.

In practice:  $q \gtrsim 2^{160}$  ( $2^{200}$ )

(3) If  $G = \mathbb{Z}_p^x$  then  $p$  should be  
so large that index calculus  
becomes infeasible:  $\tilde{O}(2^{\sqrt{\log p \log \log p}})$

In practice:  $p \gtrsim 2^{1024}$  ( $2^{2048}$ )

Remaining task:

Example Say we are looking for  $\alpha \in \mathbb{N}$ :

such that  $\alpha = 1 \in \mathbb{Z}$ . (2)

Sot I  
2.12.08  
(4)

SoTI  
2.12.08  
(4)

$$\alpha = 1 \quad \rightarrow \mathbb{Z}_2, \quad (2)$$

$$\alpha = 2 \in \mathbb{Z}_3, \quad (3)$$

$$\alpha = 3 \quad \rightarrow \quad \cancel{Z_5} \quad (5)$$

What's  $\alpha$ ?

Here we can guess:

(5)  $\Rightarrow \alpha = 3, 8, 13, 18, 23, 28, \dots$

(3)  $\Rightarrow \alpha = \cancel{3}, 8, \cancel{13}, \cancel{18}, \cancel{23}, \cancel{28}, \dots$

$\quad \quad \quad \cancel{3}, 38, \dots$

②  $\rightarrow x = \cancel{8}, 23, \cancel{28}, \dots$

Thus  $\alpha = 23 \dots$

Brute force run time!  $O(2^n)$

where  $u = \text{bitlength}(2 \cdot 3 \cdot 5)$ .

$$O_r \quad \alpha = 53.$$

because  $30 \equiv 0 \pmod{2}$ ,

$$30 = 0 \quad \text{--- } z_3$$

$$30 = 0 \quad \Rightarrow \frac{1}{5}$$

bei  $30 = 2 \cdot 3 \cdot 5$ .

Better solution?

Let's only consider only (3), (5):

$$\alpha \equiv 2 \pmod{3},$$

$$\alpha \equiv 3 \pmod{5}.$$

Re translate these:

$$\alpha = 2 + s \cdot 3 \quad \text{for some } s \in \mathbb{Z},$$

$$\alpha = 3 + (-t) \cdot 5 \quad \text{for some } t \in \mathbb{Z}.$$

$$\text{Thus: } 2 + s \cdot 3 = 3 + (-t) \cdot 5$$

$$\text{or: } s \cdot 3 + t \cdot 5 = 3 - 2.$$

This can be done by the extended Euclidean algorithm.

## Chinese Remainder Theorem

Down To Earth variant:

Given moduli  $m_1, m_2, \dots, m_r \in \mathbb{N}_{\geq 2}$  pairwise coprime,  
and numbers  $a_1, a_2, \dots, a_r \in \mathbb{Z}$

Then there exists a number  $\alpha \in \mathbb{Z}$

such that  $\forall i: \alpha \equiv a_i \pmod{m_i}$

and  $\alpha$  is unique modulo  $m_1 \cdot m_2 \cdot \dots \cdot m_r$ .

Further, one such solution can be found in

polynomial time (wrt the bitlength  $(m_1, \dots, m_r, a_1, \dots, a_r)$ )  
with help of the EEA.

SotI  
2.12.08  
(5)



Proof CRT in case  $r=2$ .

SofIT  
2.12.08  
(2)

That's enough:

$$\mathbb{Z}_m \longrightarrow \mathbb{Z}_{m_1} \times \underbrace{\mathbb{Z}_{m_2} \dots \mathbb{Z}_{m_r}}_{\substack{\cong \mathbb{Z}_{m_2} \times \mathbb{Z}_{m_3} \dots \mathbb{Z}_{m_r} \\ \cong \dots}}$$

Now, we are looking for  $\alpha \in \mathbb{Z}_{m_1 \cdot m_2}$  such that

$$\alpha = a_1 \in \mathbb{Z}_{m_1},$$

$$\alpha = a_2 \in \mathbb{Z}_{m_2}.$$

It suffices to consider  $a_1 = 1, a_2 = 0 \rightarrow \alpha^{(10)}$   
and  $a_1 = 0, a_2 = 1 \rightarrow \alpha^{(01)}.$

Given  $\alpha^{(10)}$  with  $\alpha^{(10)} = 1 \in \mathbb{Z}_{m_1}, \alpha^{(10)} = 0 \in \mathbb{Z}_{m_2},$   
 $\alpha^{(01)} = 0 \in \mathbb{Z}_{m_1}, \alpha^{(01)} = 1 \in \mathbb{Z}_{m_2}.$

we can construct

$$\alpha = a_1 \cdot \alpha^{(10)} + a_2 \cdot \alpha^{(01)}.$$

Namely, now:

$$\alpha = a_1 \cdot 1 + a_2 \cdot 0 = a_1 \in \mathbb{Z}_{m_1},$$

$$\alpha = a_1 \cdot 0 + a_2 \cdot 1 = a_2 \in \mathbb{Z}_{m_2}.$$

So let's find  $\alpha^{(10)}.$



Task is :

$$\alpha^{(10)} = 1 + (-s) \cdot m_1 \\ = 0 + t \cdot m_2$$

SotJ  
2.12.08  
(8)

for some  $s, t \in \mathbb{Z}$ .

Or :

$$1 = s \cdot m_1 + t \cdot m_2$$

This we can solve by the EEA.

And  $\alpha^{(10)} = t \cdot m_2$

Further  $\alpha^{(01)} = s \cdot m_1$

Thus  $\alpha = a_1 \cdot t m_2 + a_2 \cdot s m_1$

□

Exmp  $m_1 = 3, m_2 = 5$   
 $a_1 = 2, a_2 = 3$

EEA(3,5)  $\rightarrow 1 = \frac{(-3) \cdot 3}{-9} + \frac{2 \cdot 5}{10}$

So decd  $\begin{array}{l|l} 10 = 1 & \in \mathbb{Z}_3 \\ 10 = 0 & \in \mathbb{Z}_5 \end{array} \quad \begin{array}{l} -9 = 0 \in \mathbb{Z}_3 \\ -9 = 1 \in \mathbb{Z}_5 \end{array}$

Thus we obtain

$$\alpha = a_1 2 \cdot 10 + 3 \cdot (-9) \\ = -7 = 8 \in \mathbb{Z}_{15}$$

Now combine this with  $\alpha = 1 \in \mathbb{Z}_2$   $\rightarrow \alpha = 23 \in \mathbb{Z}_{30}$

# Number Theory

$G$  finite group

Lagrange / Euler / Fermat

Orders  $\text{ord}(g)$

CRT  $\text{gcd}(m, n) = 1 \Rightarrow \mathbb{Z}_{m \cdot n} \cong \mathbb{Z}_m \times \mathbb{Z}_n$

## Algorithms

Square & multiply

EEA

Baby-step-giant-step, Pollard- $\rho$ , Pollard-Hellman

## Crypto

Diffie-Hellman key exchange

DHP :  $(g, g^x, g^y) \mapsto g^{xy}$

DLP :  $(g, g^x) \mapsto x$

CORRECT?

EFFICIENT?

SECURE?

Would want something like:

HOPE if DLP is difficult the DHP is difficult

Easy: if DLP is easy the DHP is easy.

Thus for security it is necessary that DLP is difficult. But it is not known to be sufficient.

Example

$$G = \langle g \rangle \subseteq \mathbb{Z}_p^*$$

or

$$G = \langle P \rangle \subseteq E(\mathbb{F}_p) \text{ - elliptic curve}$$

But:  $p \neq 6$  prime

$$a^{\#G} = 1$$

$$a^{\phi(N)} = 1 \in \mathbb{Z}_N^*$$

$$a^{p-1} = 1 \in \mathbb{Z}_p^*$$

SofI  
9.12.08  
①

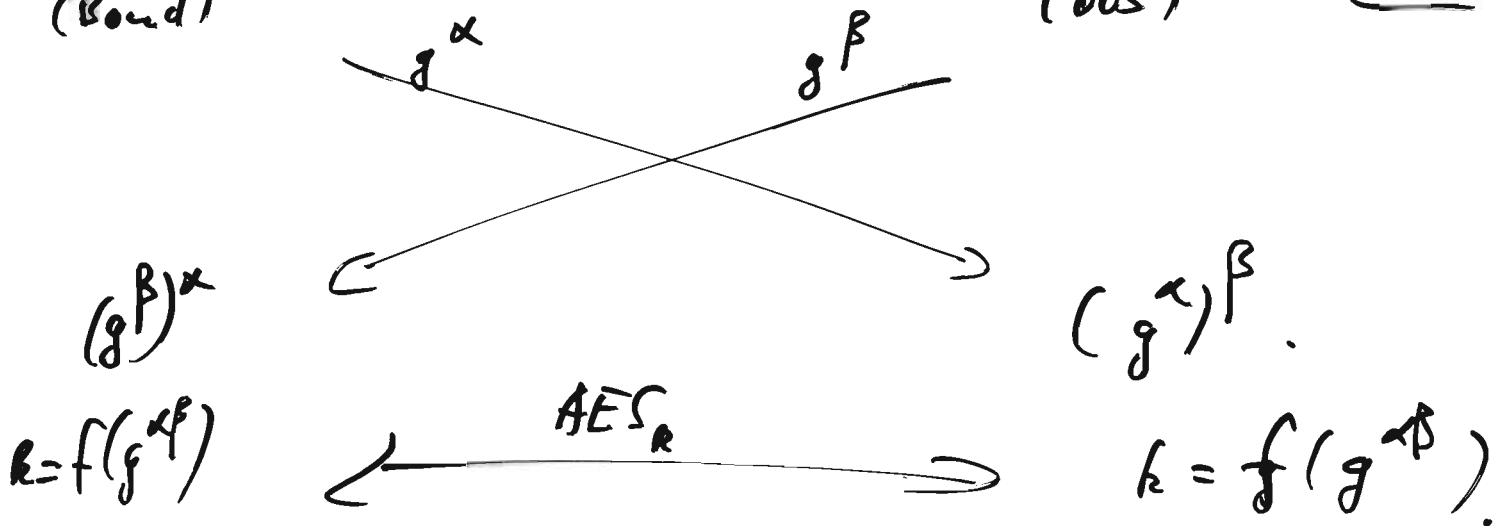
$$\begin{aligned} g^x &= a \\ g^y &= a \\ (g^x)^y &= a^y \\ (g^y)^x &= a^x \end{aligned}$$

Alice  
(Bond)

$g \in G$

Bob  
(005)

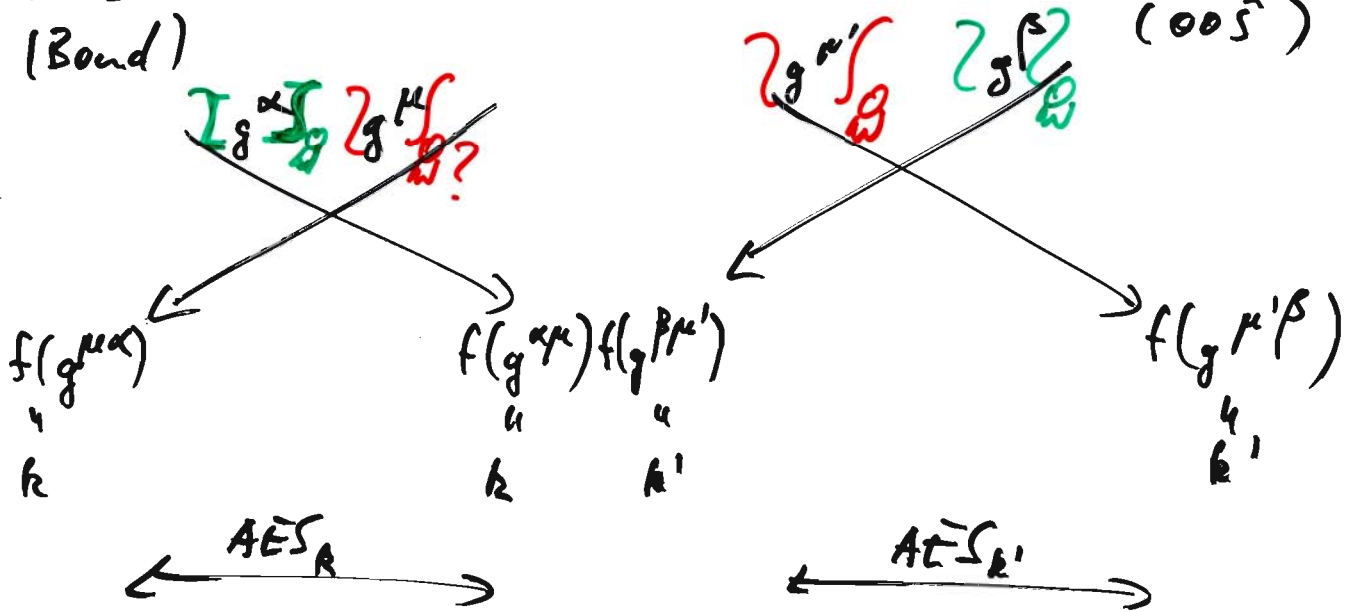
SotJ  
9.12.08  
②



Alice  
(Bond)

Mallory

Bob  
(005)



(Mallory is the unickel)

Signatures would solve that problem!

# What does a physical signature?

So+I  
9.12.08  
(3)



Document unchanged.  
(Integrity)

Person who signed is ~~Bob~~ Bob.  
(Authenticity, Identification)

Connection doc  $\leftrightarrow$  signer.  
(Non-repudiation...!?)

Easy to verify

Hard to generate for others

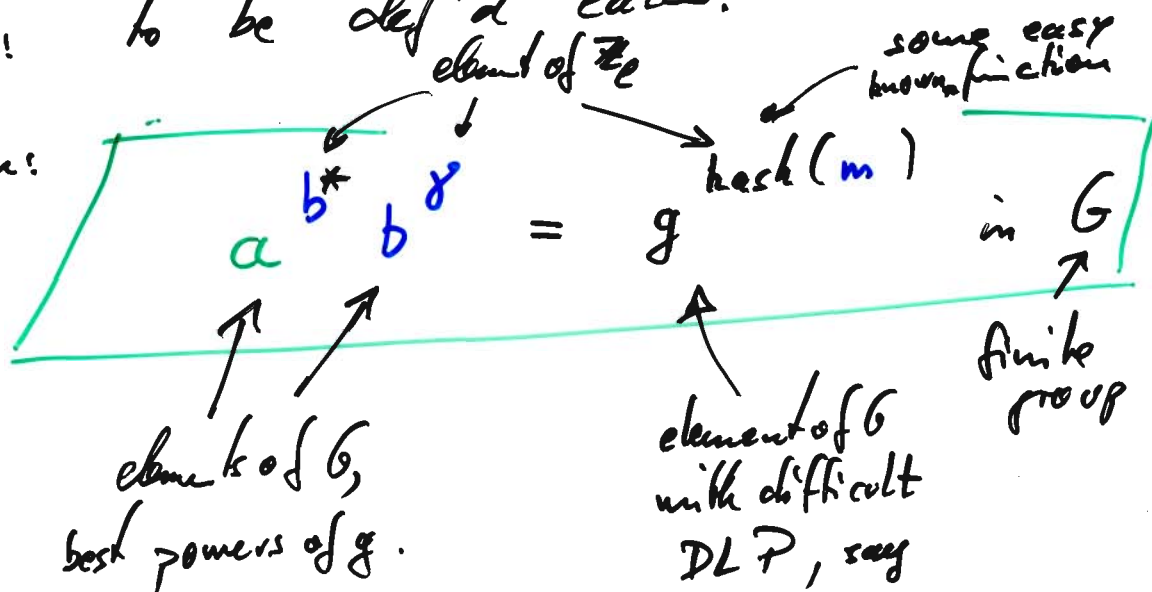
Easy to generate for the signer.

## El Gamal type signatures

Setup: to be def'd later.

Generation: to be def'd later.

Verification:



Note:  $*$ :  $G \rightarrow \mathbb{Z}_e$   
in simple-minded way.

consequently we construct:

Setup: Choose a group  $G$   
and an element  $g$   
of known (prime) order  $\ell$

Sot

9.12.0

(4)

For example:

choose  $\ell$  a 160-bit prime.

choose  $p$  a 1024-bit prime  
with  $\ell \mid p-1$ .

(choose  $a$  such that  $1+a\cdot\ell$   
has 1024 bits  
check whether it's prime,  
if not retry!)

choose  $h \in \mathbb{Z}_p^*$  at random

and set  $g := h^{\frac{p-1}{\ell}}$ .

(Then  $g^\ell = 1$ . If  $g = 1$ : retry.)

Now  $\text{ord } g = \ell$ .

Fix a function  $*$ :  $G \rightarrow \mathbb{Z}_\ell$   
simple minded.

For example with  $G = \langle g \rangle \subseteq \mathbb{Z}_p^*$  use

$$(a \bmod p)^* = a \bmod p-1$$

for  $0 \leq a < p$ .

Fix a function  $\text{hash}: \{0,1\}^* \rightarrow \mathbb{Z}_\ell$ .

Individual setup (by Alice):

SotJ  
9.12.08  
(5)

Choose a **private key**  $\alpha \in \mathbb{Z}_e$ .  
Compute the **public key**  $a = g^\alpha$  in  $G$ .  
Now finding the private key from the public key is a DLP.

Generate a signature:

Input: message  $m$ , **public key**  $a$ , **private key**

Output: signature  $s = (b, \gamma)$  on  $m$ .

1. In order to have the entire signing equation as powers of  $g$ , choose  $b$  as one:

Choose a temporary secret  $\beta \in_R \mathbb{Z}_e$   
and compute  $b = g^\beta$  in  $G$ .

2. Determine  $\gamma \in \mathbb{Z}_e$  such that signature is valid, i.e.

$$g^{\alpha b^* + \beta \gamma} = g^{\text{hash}(m)} \text{ in } G$$

which is equivalent (by  $\exp_g: \mathbb{Z}_e \rightarrow \langle g \rangle$  being bijective)

to

$$\alpha b^* + \beta \gamma = \text{hash}(m) \text{ in } \mathbb{Z}_e$$

This is a linear equation for  $\gamma$ .

Solve it and

3. Return  $(b, \gamma)$

CORRECT?

If Alice signs  $m$  with (L.S.)  
then Bob checks this signature  
and finds it to be valid.  
This is true by construction!

SotJ  
3.12.08  
⑥

EFFICIENT?

~~Even~~ Setup :  $O(n^4)$  ~ minutes  
Individual setup }  
Generation }  $O(n^3)$  ~ seconds  
Verification }

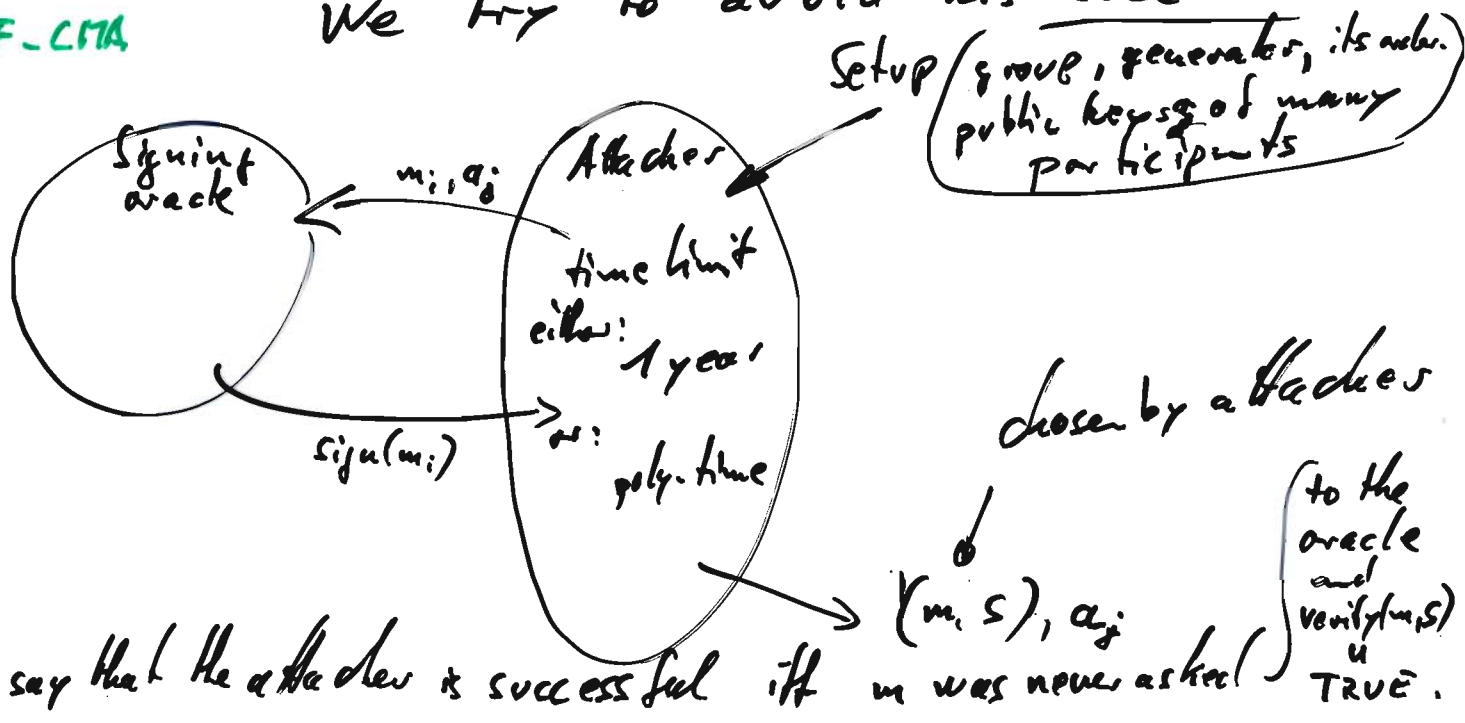
$n$  = key size.

SECURITY?

10.12.08

Security game:  
Attacker tries to attack.  
We try to avoid his success.

EF-CMA



Security goal is:

No attacker can succeed  
with high probability.

SoFI  
10.12.08  
(1)

( $2^{-n}$  is ok, but  $\frac{1}{n!}$  is too much!)

Consequences:

① No attacker can change  
a given document.

Assuming he could  
then using that subroutine changedoc  
he would win the game.

$$\text{changedoc}(m, s) = (m', s')$$

[where if  $(m, s)$  is valid then  $(m', s')$  is valid.]

② No attacker can change  
the signer.

Assuming he could, i.e. he has  
a subroutine changesigner  $(m, s, a)$

$$(m', s', a')$$

then he would win the game.



③ No one but the actual  
can have generated  
a certain signature.

SotI  
10.12.08  
②

[otherwise that would  
be a successful attacker.]

What about ElGamal type signatures?

Then The DLP must be difficult  
if the ElGamal type signature  
scheme is secure.

DLP easy  $\Rightarrow$  ElGamal type signature  
insecure

Proof Assume the attacker has a subroutine  
to solve the DLP.

Then he can use it to compute  
the secret keys corresponding to the  
public keys  $a_j$ .

Thus he wins the game easily. 13

Actually, then the attacker can sign any  
message he is given.

Another option for the attacker would be to try to find  $b$  last and choose  $m, y$  at random before. Then he to solve an equation

$$a^{b^*} b^y = h.$$

If he ~~could~~ <sup>can</sup> then he could win the game

We would like to prove:

if he could then he <sup>can solve</sup> the DLP.

But:

Then If the ElGamal type signature is secure  
| then that problem is difficult. □

Further building block: hash.

Assume the attacker has a subroutine

that can, given a message  $m$ ,

find a second message  $m' \neq m$

with  $\text{hash}(m') = \text{hash}(m)$

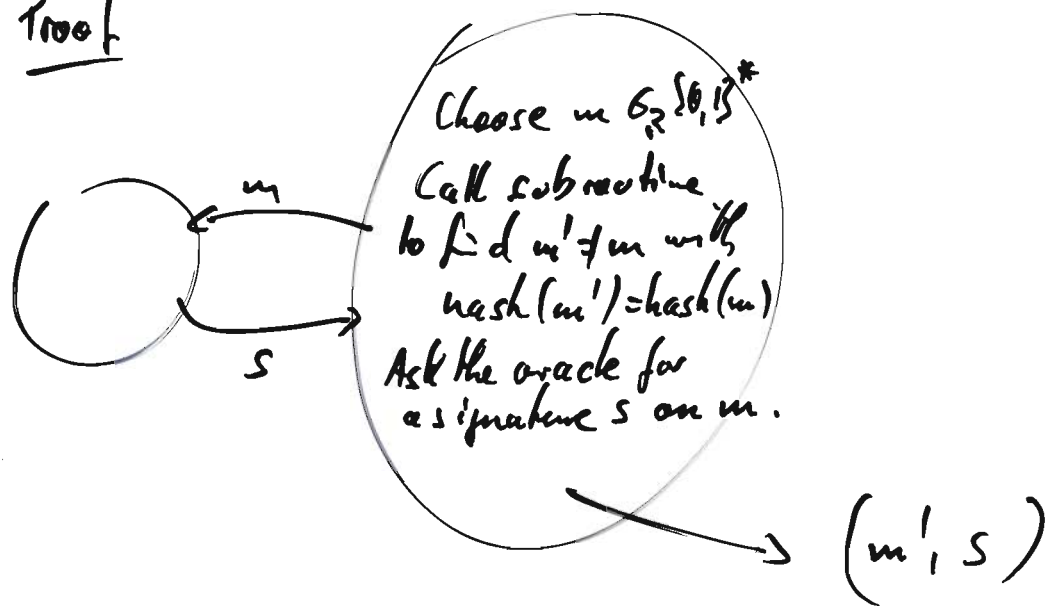
then the attacker can win the game.

SotI  
10.12.08  
(3)

Then If the hash function is not second preimage resistant, i.e. the attacker has such a subroutine, then the scheme is insecure.

SotJ  
10.12.08  
④

Proof



Then If the hash function is not collision resistant, i.e. the ~~data owner~~ attacker has no subroutine that outputs two messages  $m, m'$  such that  $m \neq m'$  and  $\text{hash}(m) = \text{hash}(m')$ , then the scheme is insecure.

Proof As above ...

Note: Finding  $m'$  given  $m$  with  $\text{hash}(m') = \text{hash}(m)$  can be done

## Example

One common hash function is

SHA1

it gets any message as a bitstring  
and outputs a 160-bit string.

$$\text{SHA1: } \{0,1\}^* \longrightarrow \{0,1\}^{160}$$

practical\*  
( $=2^{64}$ )

Possible routine to find a second preimage:

Input:  $m$   
Output:  $m'$

1. REPEAT
2.     Choose  $m' \in_R \{0,1\}^{160}$  \*
3.     UNTIL  $\text{hash}(m') = \underbrace{\text{hash}(m)}_{\text{fixed!}}$
4.     Return  $m'$

$$\text{exit-probability} = \frac{1}{2^{160}}$$

and so the expected runtime =  $2^{160}$ .

Sat  
10.12.08  
(5)

Possible machine to find a collision

Input: -

Output:  $m, m'$ .

SotJ  
10.12.08  
⑥

1.  $L \leftarrow$  empty list,  $i \leftarrow 0$ .

2. REPEAT

3. Choose  $m_i \in_R \{0, 1\}^k$ ,  $i \leftarrow i+1$ , append  $m_i$  to the list.

4. UNTIL  $\text{hash}(m_i) = \text{hash}(m_j)$  for some  $j < i$

5. Return  $(m_i, m_j)$ .

$$\text{exit-probability} = \frac{i}{2^{160}}$$

$$\Rightarrow \text{expected run time} \approx 2^{80} = \sqrt{2^{160}}.$$

That's much larger than anything  
doable now a days.

(By a factor  $\approx 1000000$ .)

Fact There exists a procedure to find  
a collision for SHA-1 which  
needs only  $2^{63}$  calls to SHA-1.

thus it's considered broken,  
but nobody did it so far. HASH  
CRISIS.

# Summary      Signatures

SotI  
16.12.08

(1)

Example scheme:

ElGamal type signatures:

$$a b^x b^y = g^{\text{Hash}(m)} \text{ in } G$$

Instances: ElGamal signatures

$$G = \mathbb{Z}_p^*, \text{ ord}(g) = p-1.$$

Schnorr signatures, DSA

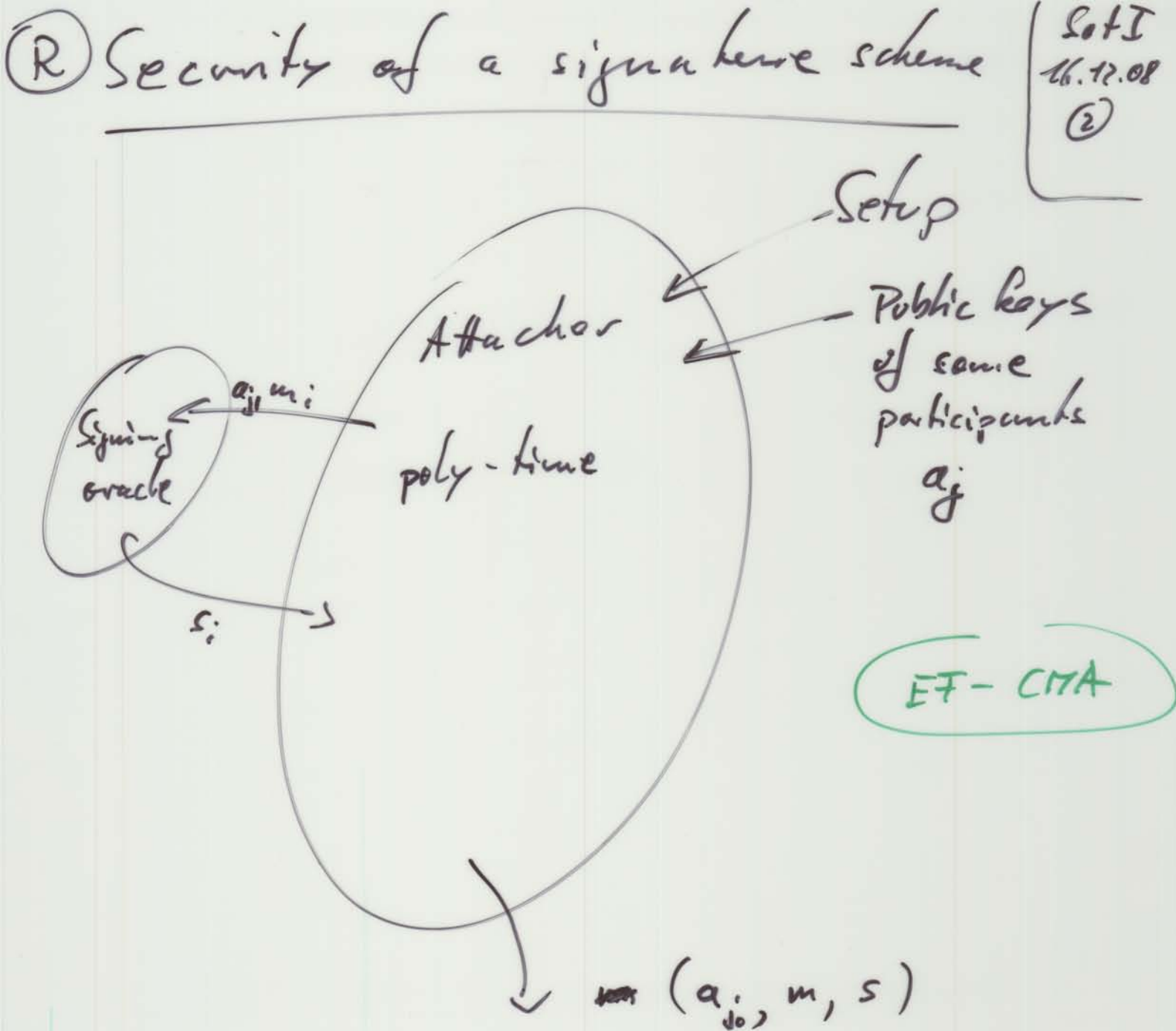
$$G = \mathbb{Z}_p^*, \text{ ord}(g) = \ell$$

+ additional trick  
to replace <sup>1024 bit</sup>  $b$  by <sup>160 bit</sup>  $b^x$   
in the signature  
(thus saving space!)

ECDSA

$G$  = an elliptic curve,

$\text{ord}(P) = \ell$  either prime  
or a small ( $\leq 256$ )  
multiple of  
a prime



Attacker's success:

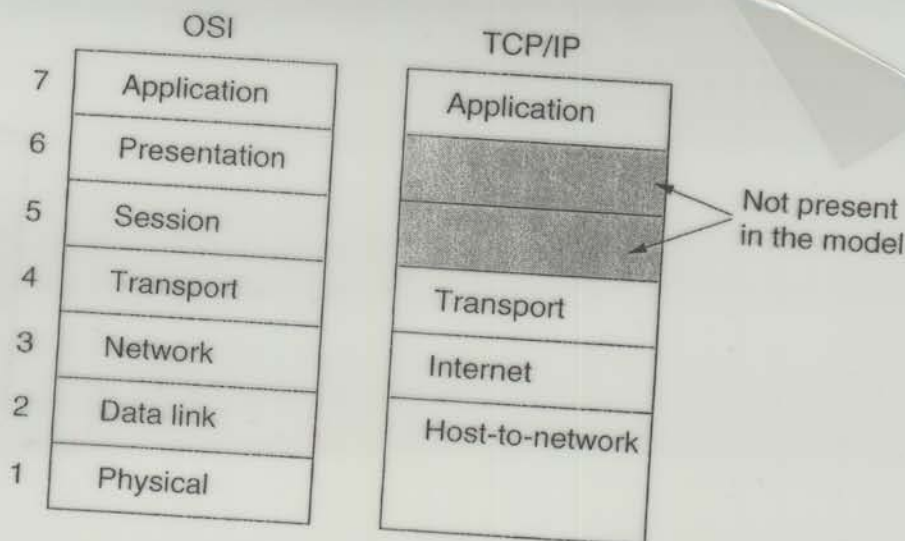
$(a_{j_0}, m)$  has never been queried to the oracle.

$(m, s)$  is a valid signature <sup>w.r.t.</sup> the public key  $a_{j_0}$ .

Our scheme is secure if no attacker can win this game.



SoTI  
17.12.08  
(7)



← SSH / SCP  
← SSL / TLS  
← IPsec  
IP

Figure 1-21. The TCP/IP reference model.

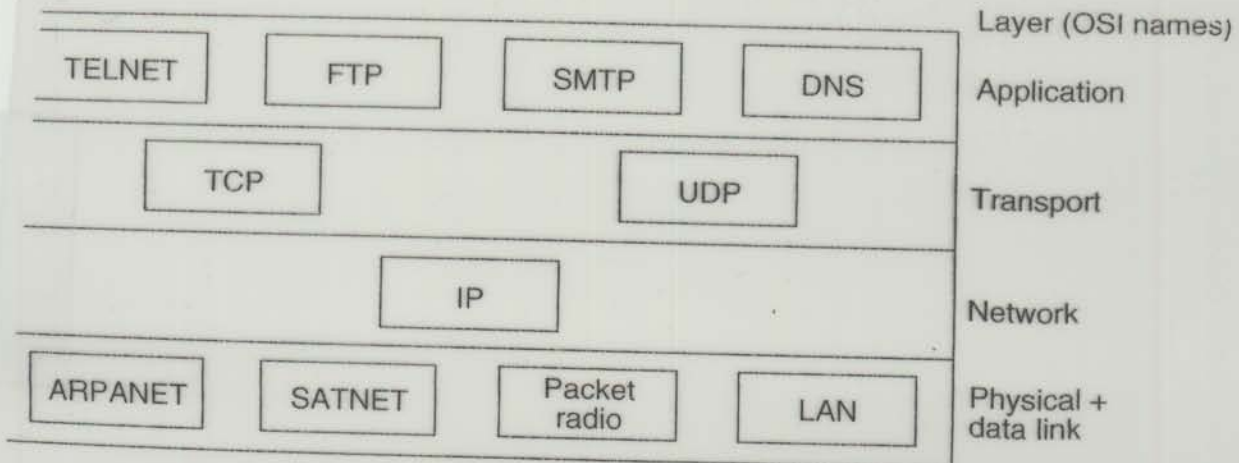


Figure 1-22. Protocols and networks in the TCP/IP model initially.



# IPsec

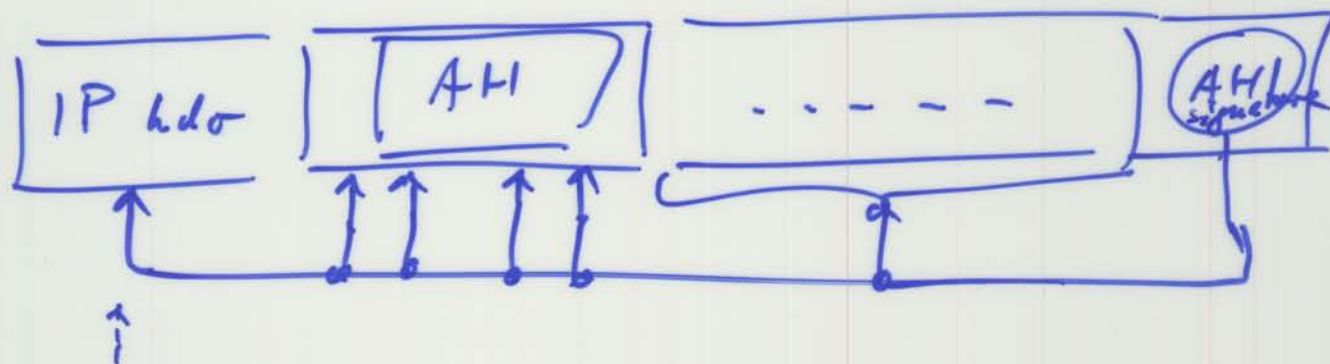
Sofia  
17.12.08  
(2)

AH - authentication header

ESP ~~ESP~~ - encapsulating security ~~payload~~ payload.

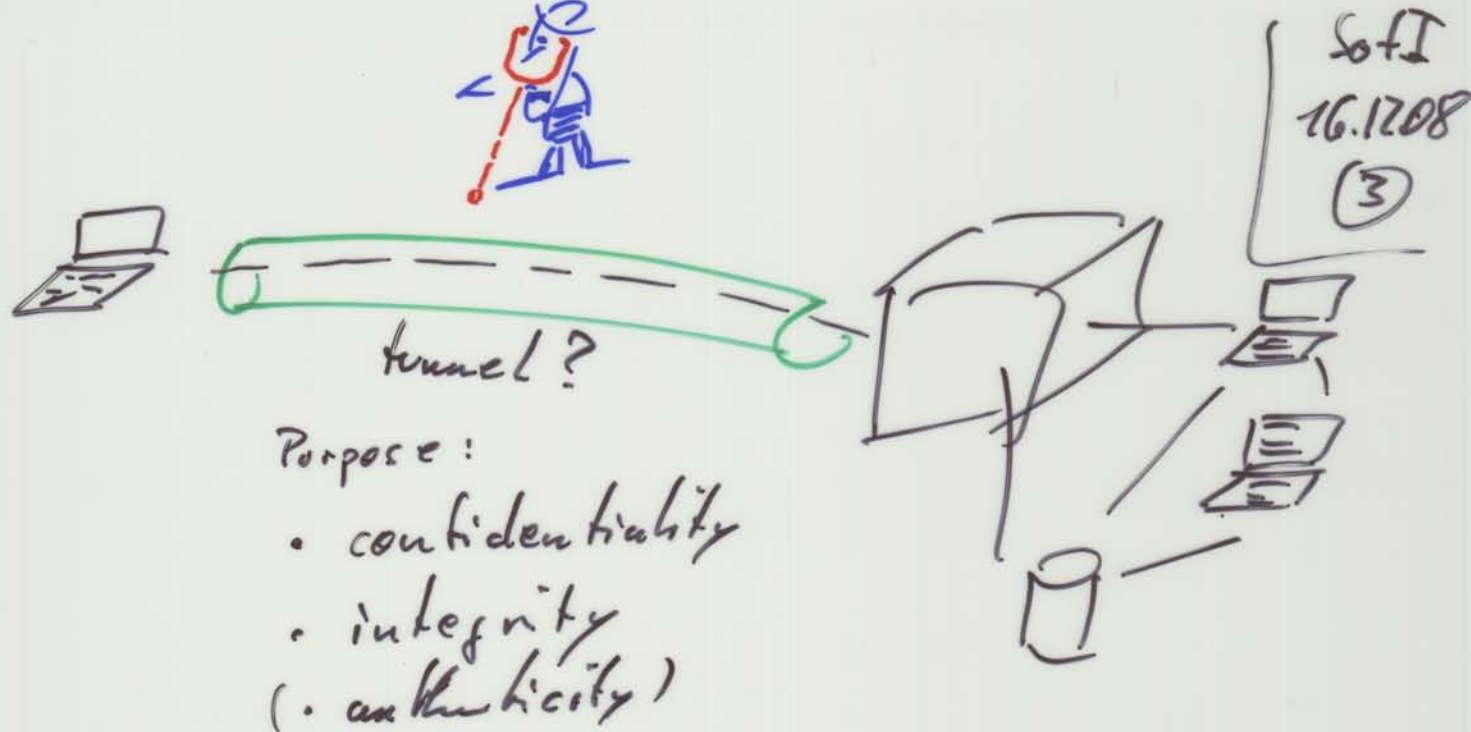
AH  
|  
signature  
↓  
integrity  
authenticity

ESP  
/      \ optional  
encryption      signature  
[may be null]  
↓  
confidentiality  
(unless you use null encryption)  
↓  
integrity  
authenticity



# remaining  
hops.

some fields change during transport  
es. hops, IP-src, IP-dest. NAT



Need a common key for all this!



Where to get the common key from?

Soft I  
17.12.08  
(3)

Task	AH	ESP enc	ESP both
Access control	+	+	+
Connection integrity	+	-	+
Data origin authentication	+	-	+
Rejection of replay attacks	+	(+)	+
Confidentiality	-	+	+
Limited Traffic Flow Confidentiality	-	+	+

## Building blocks / terminology

SA ... security association  
= all connection related data  
stored by the communication  
partners, in particular key  
material:

- IP destination address
- sequence number counter (32 bits)
- sequence counter overflow
- SPI security protocol identifier
  - AH → encryption only
  - ESP → both

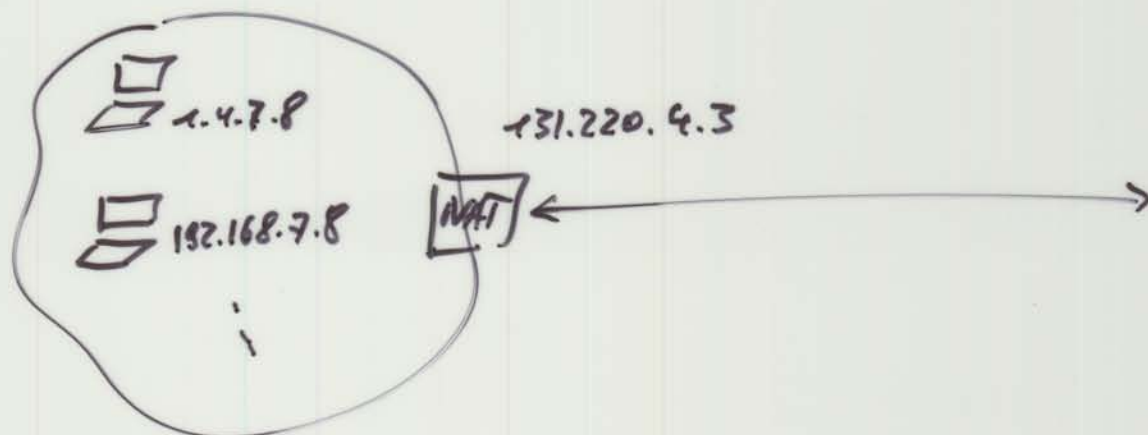


- used algorithms
- Life time of SA  
(usually 8 hours, 12 hours)

Soft  
17.12.08  
(4)

## Problematic issues

### NAT



→ causes problems when trying to authenticate source or destination IP.

(With IPv6 no NATs are necessary, thus IPv6 finds support ESP rather than AH.)

### Firewalls



Among many other things a firewall filters packets according to the TCP port number. But if that is encrypted, as with IPsec+ESP, then this cannot be used...

# IPSEC & IKE

MICHAEL NÜSKEN

25 June 2007

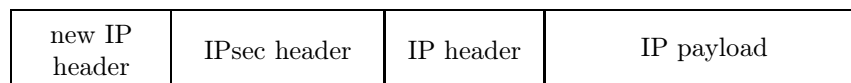
Before all: we are talking about a collection of protocols. Each partner of the exchange has to keep some information on the connection. This is in our context called the security association (SA). It contains specification about the algorithms that should be used for encryption and authentication, it contains keys for these, it may contain traffic selectors (filtering rules), and more. Each SA manages a simplex connection for one type of service. In each direction there will be an SA for the key exchange (IKE\_SA) and one for the encapsulating security payload or for the authentication header. So each partner has to maintain at least four SAs. Such an SA is selected by an identifier, the so-called security parameter index (SPI). It is chosen randomly but so that it is unique.

## 1. IPsec

The secure internet protocol modifies the internet protocol slightly. We have the choice between transport and tunnel mode. In tunnel mode, an IP packet



is wrapped in with a new IP header and an IPsec header to

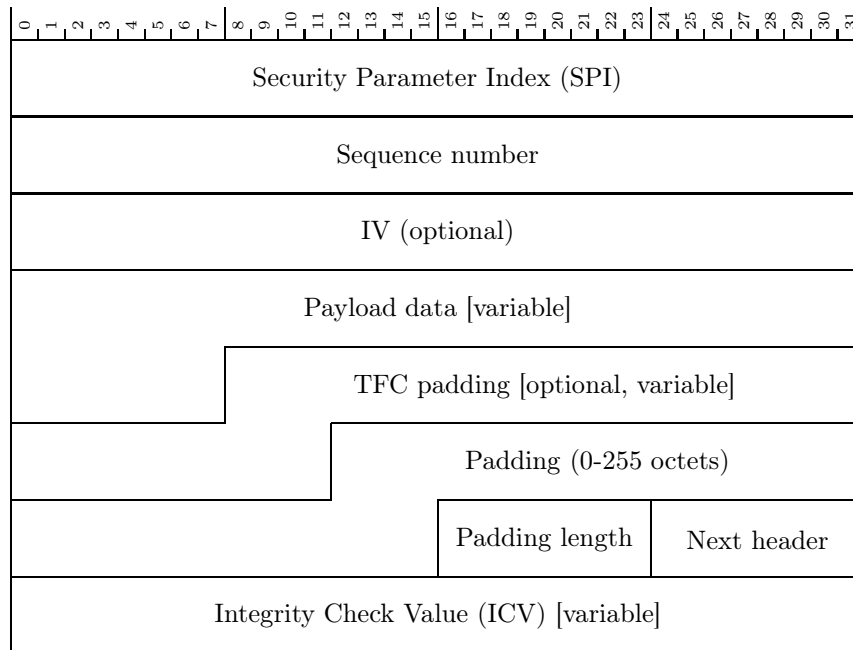


In transport mode, only the IPsec header is added:



There are two types of IPsec headers: the encapsulating security payload (ESP) and the authentication header (AH).

**1.1. IPsec encapsulating security payload.** The ESP specifies that and how its payload is encrypted and (optionally) authenticated. Actually, this ‘header’ is split into a part before and one after the data:

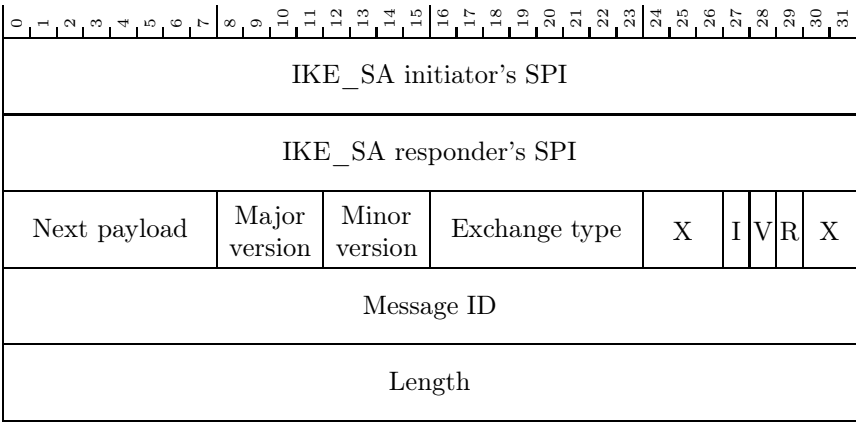


The security parameter index identifies the SA and thus all necessary algorithms and key material. To create the secured packet from the original one, it is first padded. Padding is used to enlarge the data length to a multiple of a block size that might be associated with the encryption. Traffic flow confidentiality (TFC) padding can be used to disguise the real size of the packet. Then the data is encrypted; in tunnel mode including the old IP header. To be precise, all the information from Payload data to Next header is encrypted. Next, a message authentication code is calculated for this encrypted text and security parameter index, sequence number, initialization vector (IV) and possibly further padding; actually the message authentication code covers the entire packet but the header and the integrity check value plus the extended sequence number and integrity check padding if any.

**1.2. IPsec authentication header.** The AH authenticates its payload and also parts of the IP header. (Yes, this does violate the hierarchy.)

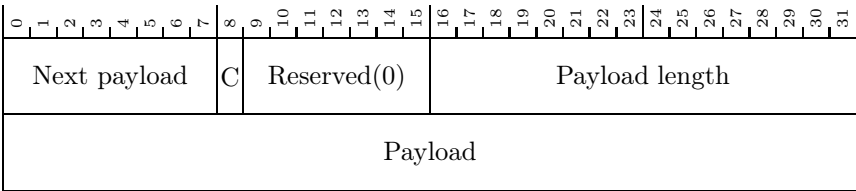
## 2. Internet key exchange (version 2)

Any message in the internet key exchange starts with a header of the form



Clearly, the version is 2.0 with the present drafts (major version: 2, minor version: 0). The flags X are reserved, the I(nitiator) bit is set whenever the message comes from the initiator of the SA, the V(ersion) bit is set if the transmitter can support a higher major version, the R(esponse) bit is set if this message is a response to a message with this Message ID. The header is usually followed by some payloads like

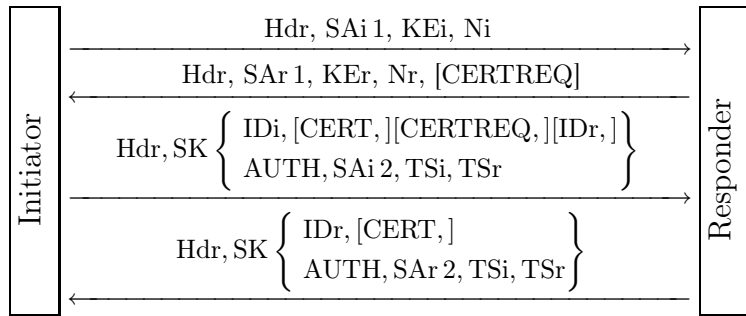
Exchange type	Value
Reserved	0-33
IKE_SA_INIT	34
IKE_AUTH	35
CREATE_CHILD_SA	36
INFORMATIONAL	37
Reserved to IANA	38-239
Reserved for private use	240-255



The C(ritical) bit indicates that the payload is critical. In case the recipient does not support a critical payload it must reject the entire message. A non-critical payload can be simply skipped. All the payloads defined in RFC4306 are to be handled as critical ones whatever the C bit says.

Next payload	Notation	Value
None		0
RESERVED		1-32
Security Association	SA	33
Key Exchange	KE	34
Identification - Initiator	IDi	35
Identification - Responder	IDr	36
Certificate	CERT	37
Certificate Request	CERTREQ	38
Authentication	AUTH	39
Nonce	Ni, Nr	40
Notify	N	41
Delete	D	42
Vendor ID	V	43
Traffic Selector - Initiator	TSi	44
Traffic Selector - Responder	TSr	45
Encrypted	E	46
Configuration	CP	47
Extensible Authentication	EAP	48
Reserved to IANA		49-127
Private use		128-255

## 2.1. Initial exchange.



### PROTOCOL 2.1. IKE\_SA\_INIT.

1. Prepare SAi1, the four lists of supported cryptographic algorithms for Diffie-Hellman key exchange (groups), for the pseudo random function used to derive keys, for encryption, and for authentication. Guess the group for Diffie-Hellman and compute  $KEi = g^a$ .

Choose a nonce Ni.

2. Choose SAR1 from SAi1 unless no variant is supported.

Hdr, SAi 1, KEi, Ni →



Compute  $K_{Er} = g^b$  if the group was guessed correctly. (Otherwise send:

Hdr, N(INVALID\_KE\_PAYLOAD, group)

.)

Choose a nonce Nr.

Hdr, SAr 1, KEr, Nr,

[CERTREQ]

3. Both parties now derive the session keys. We assume that  $prf$  is the selected pseudo random function which gets a key and a bit string as input.

$SKEYSEED = prf(N_i | N_r, g^{ab}),$

$SK\_d | SK\_ai | SK\_ar | SK\_ei | SK\_er | SK\_pi | SK\_pr$   
 $= prf+(SKEYSEED, N_i | N_r | SPI_i | SPI_r)$

where  $prf+(K, S) = T_1 | T_2 | T_3 | \dots$ , and  $T_1 = prf(K, S | 0x01)$ ,  $T_i = prf(K, T_{i-1} | S | i)$  for  $i > 1$ .  $SK\_d$  is used for the derivation of keys in a child SA.  $SK\_ai$  and  $SK\_ei$  are used for authenticating and encrypting messages sent by the initiator,  $SK\_ar$  and  $SK\_er$  for messages sent by the responder.

4. The initiator send its identity IDi, optionally one or more certificates CERT, a certificate request CERTREQ (possibly including a list of trusted CAs), and optionally the responders identity IDr (it may be that the responder serves multiple identities 'behind' it).

Further she computes an authentication AUTH (using the key from the first CERT payload) for the entire first message concatenated with the responder's nonce Nr and the value  $prf(SK\_pi, IDi)$ . The authentication method can be RSA digital signature (1), shared key message integrity code (2), or DSS digital signature (3).

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Next payload									C	Reserved(0)									Payload length												
Auth method									Reserved																						
Authentication data																															

The initiator starts to negotiate a child SA in SAi2 with proposed traffic selectors TSi, TSr.

Hdr, SK  $\left\{ \begin{array}{l} IDi, [CERT,] \\ [CERTREQ,] \\ [IDr,] \\ AUTH, SAi2, \\ TSi, TSr \end{array} \right\}$

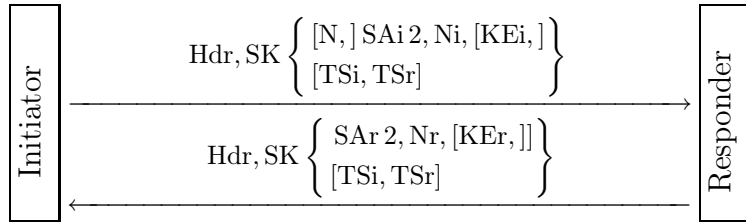
5. The responder sends its identity  $ID_r$ , certificate(s).  
 He computes an authentication  $AUTH$  for the entire second message concatenated with the initiator's nonce  $N_i$  and the value  $\text{prf}(SK_{pr}, ID_r)$ .  
 Further he supplies the answer  $SA_r 2$  to the child SA creation and sends the accepted traffic selectors  $TS_i, TS_r$ .

$$\xleftarrow{\text{Hdr, SK} \left\{ \begin{array}{l} ID_r, [CERT, ] \\ AUTH, SA_r 2, \\ TS_i, TS_r \end{array} \right\}}$$

If this initial exchange is completed successfully the  $IKE\_SA$  and a  $CHILD\_SA$  are ready for use. Keying material for the childs is generated similar to the  $IKE\_SA$  keys:

$$KEYMAT = \text{prf}+(SK\_d, N_i | N_r)$$

**2.2. Creating additional child SAs.** Further childs can be created under this  $IKE\_SA$  using a  $CREATE\_CHILD\_SA$  exchange:



In case a  $CHILD\_SA$  shall be rekeyed the notification payload  $N$  of type  $REKEY\_SA$  specifies which SA is rekeyed. This can be used to established additional SAs as well as to rekey ages ones. Create new ones and afterwards delete the old ones. Also the  $IKE\_SA$  can be rekeyed similarly.

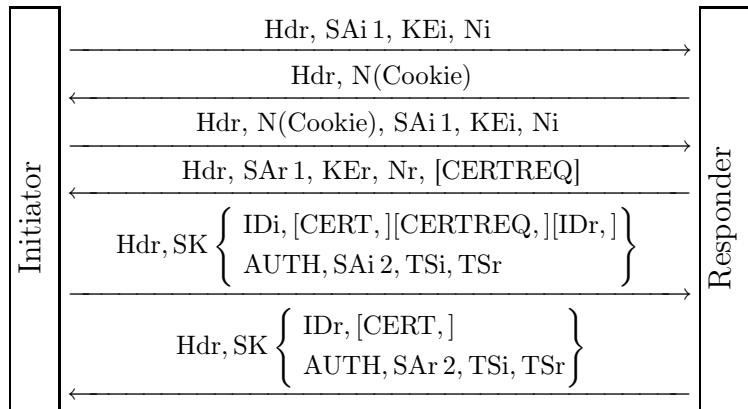
In a  $CREATE\_CHILD\_SA$  exchange including an optional Diffie-Hellman exchange new keying material uses also the new Diffie-Hellman key  $g^{ir}$ , it is concatenated left to the nonces. (Though the Diffie-Hellman key exchange is optional, it is recommended to either used it or at least to limit the number of uses of the original key.)

**2.3. Denial of Service.** If the server has a lot of half open connections (ie. the first message arrived, the second was sent but the third message is pending) it may choose to send a cookie first. (In order to defeat a denial of service attack.) It is suggested to use a stateless cookie consisting of a version identifier and a hash value of the initiator's nonce  $N_i$ , her IP  $IP_i$ , her security parameter index  $SPI_i$  and some secret:

$$\text{Cookie} = \text{verID} | \text{hash}(N_i, IP_i, SPI_i, \text{secret}_{\text{verID}})$$

This way the secret can be exchanged periodically, say every second, and the server only needs to store the last few (randomly) generated secrets.

The authentication AUTH then refers to the second version of the corresponding message, so the one including the cookie or responding to that, respectively. So the protocol becomes:



**2.4. Extended authentication protocols.** The initiator may leave out AUTH and thereby tell the responder that she wants to perform an extensible authentication which is then carried out immediately.

**2.5. IP compression.** The parties can negotiate IP compression.

## 2.6. ID payload.

The ID payload

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Next payload								C	Reserved(0)								Payload length														
ID type								Reserved																							
Identification data																															

can be an IP address (ID type 1), a fully-qualified domain name string (2), a fully-qualified RFC822 email address string (3), an IPv6 address (5), an ASN.1 X.500 Distinguished Name [X.501] (9), an ASN.1 X.500 general name [X.509] (10), a vendor specific information (11).

### 2.7. CERT payload.

The CERT payload

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
Next payload								C	Reserved(0)								Payload length															
Cert encoding								Certificate data																								
Certificate data																																

can be encoded in various widely used formats. Note that it can also carry revocation lists.

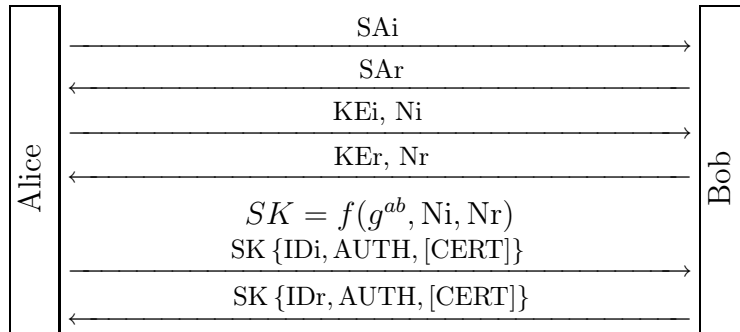
### 3. IKE version 1

The version 1 of the internet key exchange distinguishes between a main mode and an aggressive mode. Further it allows four variants in each mode depending on the desired type of authentication. Authentication can be based on

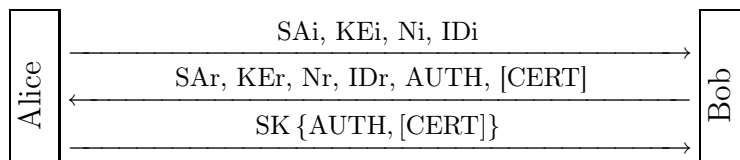
- public signature keys,
- public encryption keys, original protocol,
- public encryption keys, revised protocol, or
- a pre-shared secret.

We only give the bare protocol summaries here, using notation similar to the one used for version 1. (They are not based on RFC240x but on the book ?.)

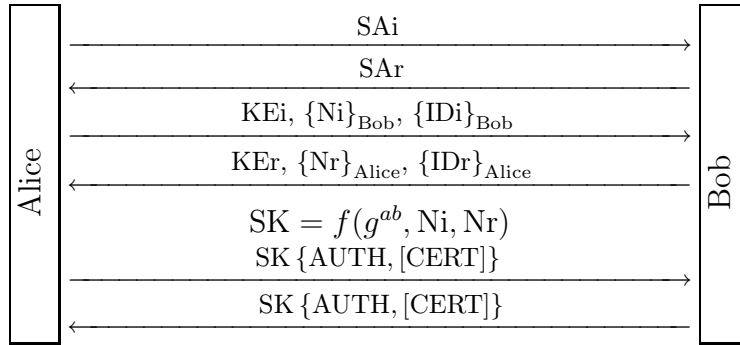
#### 3.1. Main mode, public signature keys.



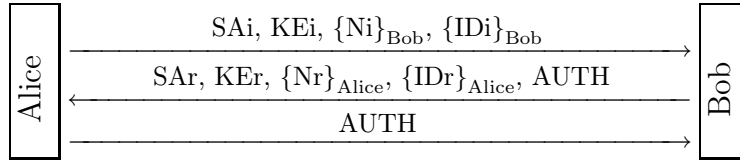
#### 3.2. Aggressive mode, public signature keys.



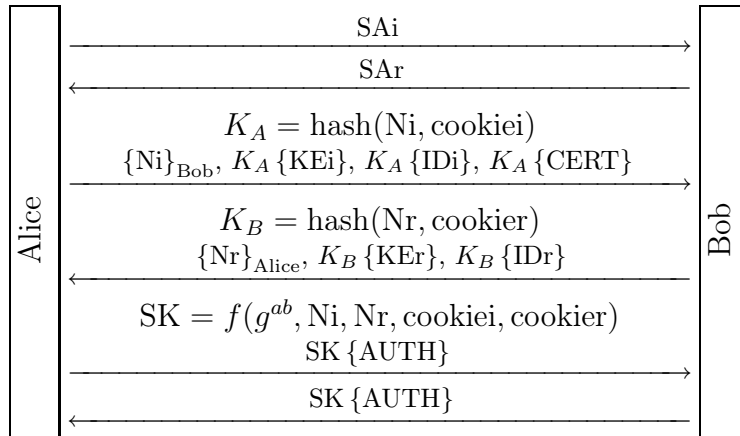
### 3.3. Main mode, public encryption keys, original protocol.

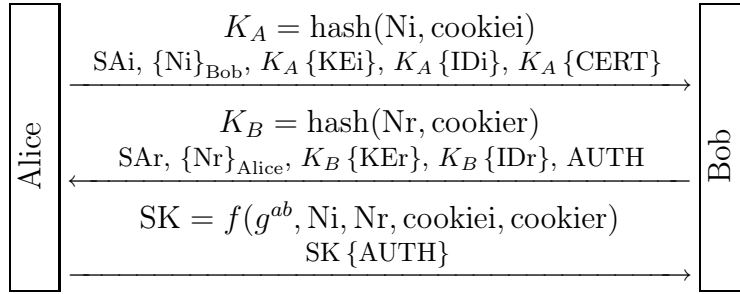
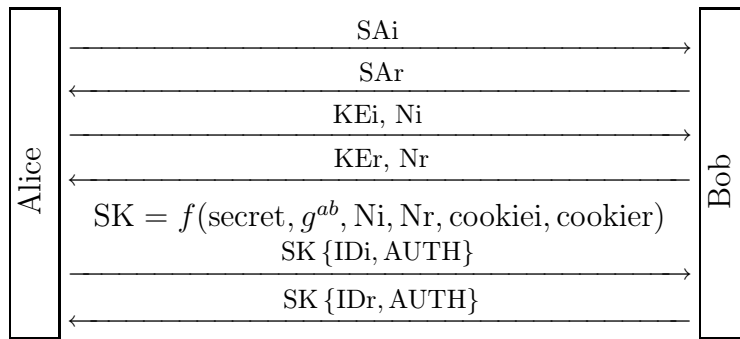
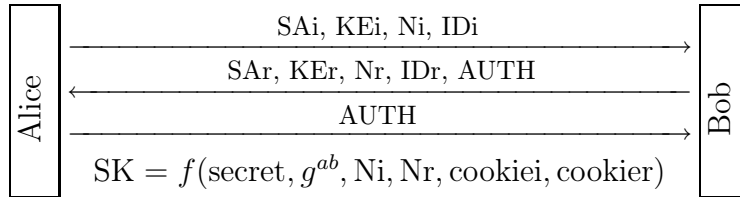


### 3.4. Aggressive mode, public encryption keys, original protocol.



### 3.5. Main mode, public encryption keys, revised protocol.



**3.6. Aggressive mode, public encryption keys, original protocol.****3.7. Main mode, pre-shared secret.****3.8. Aggressive mode, pre-shared secret.**

# History of IKE

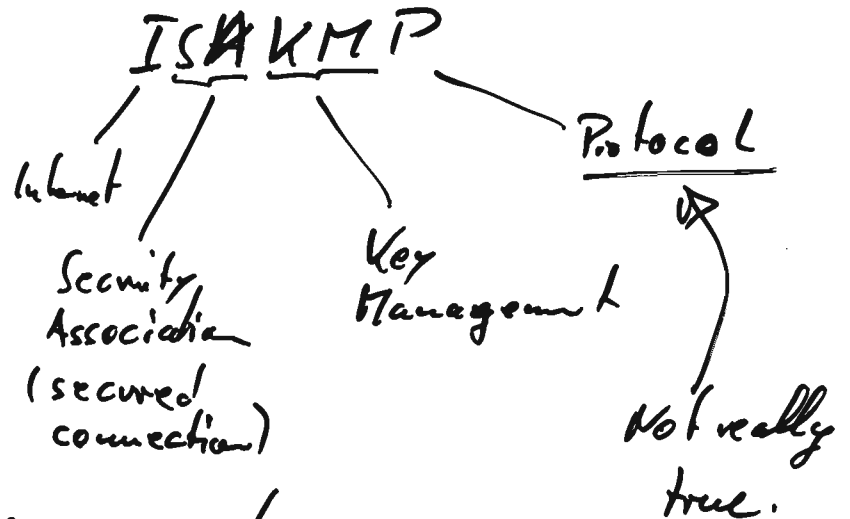
STI  
6.1.09

PHOTURIS

SKIP



NSA proposed:



- only framework
- ruled out both candidates

→ IETF could take up the development

OAKLEY, SKEME ... (new drafts)

IKEv1 puts ... into ISAKMP

- Problem:
- no clear design
  - too many variants
  - documentation:  $\geq 150$  pages  
 $\geq 3$  RFC
- & difficult to read.

IKEv2

SotI  
6.1.09  
②

- clear, simple rules
- any request gets a response
- initial exchange: 1 option, (rather than 8)  
4 msgs.
- + create child SA, 2 msgs.

security  
association

- all functionality of IKEv1 is still there!

→ easier analysis

## Security questions

① Secure?

① Session key agreement.

- How long? Random?

- Do both parties contribute to it?

- Man in the middle

② Perfect forward security

- Can an attacker decrypt  
given the long-term secrets  
after termination of the connection?

Escrow failure

- ... during the connection.



(3)

## Denial of service

- How expensive is a half-open connection (half-open = anything until authentication)?

time, space, communication!

(4)

## Endpoint identities, hiding

- Does an eavesdropper get information about the identities?
- Can an active attacker get identity information of initiator or responder?

[Cannot have both 'no'!  
→ so choose to decide what, if ever,  
is wanted. Design decision.]

(5)

## Live partner reassurance

- Replay?

(6)

## Plausible deniability

- Does the protocol log prove that
  - Alice talked?
  - Bob talked?
  - Alice talked to Bob?
  - Bob talked to Alice?

Sot I  
6.1009  
(2)

⑦

## Stream protection

• How is a logical data stream protected?

- confidentiality?
- authenticity?
- integrity → unchanged  
→ complete, correct order,  
not too much

⑧

## Negotiating crypto parameters

→ Pros ...

→ Cons ...

## Task (we'll do that tomorrow)

Answer these questions and  
classify pros and cons  
for

- (a) IKEv1 aggressive mode
- (b) IKEv1 main mode
- (c) IKEv2.

SotI  
6.1.08  
③

# IKEv1 - Aggressive Mode

- Man in the middle has no chance with pre-shared key and public key encryption
- Perfect Forward Security is assured through "short time" secrets
- Escrow Follage is prevented by the discrete log property of the "short time" secrets
- An eavesdropper get no information about the identities with public key encryption
- Nonces prevent replay attacks
- Assuming the key exchange created a secure session key, this ~~implies~~<sup>implies</sup> the fulfillment of the three notions of stream protection
- less communication effort and faster key exchange ~~than~~ compared to the Main Mode

# IKE v. 1 main mode

- ①
  - $\text{Key} = \text{hash}(g^{dR}, N_i, N_r) \leftarrow \text{fixed size}$
  - both parties contribute
  - Man-in-the-middle - attack prevented by authentication
- ② PFS: ✓ SK based on NONCES
  - Escrow failure: possible if nonce in cleartext  
↳ only in main mode, public signature key
- ③ DOS possible
- ④
  - only if ID is not encrypted  $\Rightarrow$  not the case!
  - Attacker can pretend to be the server  $\rightarrow$  find out Alice ID in main mode, public sig key
- ⑤ Replay not possible due to the nonces
- ⑥ Each party can prove the ID of the other party
- ⑦
  - Confidentiality: by ~~code~~ encryption ✓
  - authenticity: by signature ✓
  - integrity
- ⑧

# IKEv2

1. 4 messages for Key exchange

+ Random?  $\rightarrow$  Yes: use nonces + KM

+ MH?  $\rightarrow$  No: authentication

2. Perfect Forward Security

$\rightarrow$  Yes, because session keys are not based on secret keys (random  $x, p$ )

3. DoS  $\rightarrow$  possible

half conn are expensive: KM generation + storage, nonce

4. Endpoints are hidden (authentication material is encrypted)  $\rightarrow$  if the attacker is non active

5. Yes, because of the nonces (live partner reassurance)

6. Single messages are deniable, but both participants can prove that they talk to each other

+ Session keys are strong

+ resistant to MH

+ replay protection

+ extensible authentication support

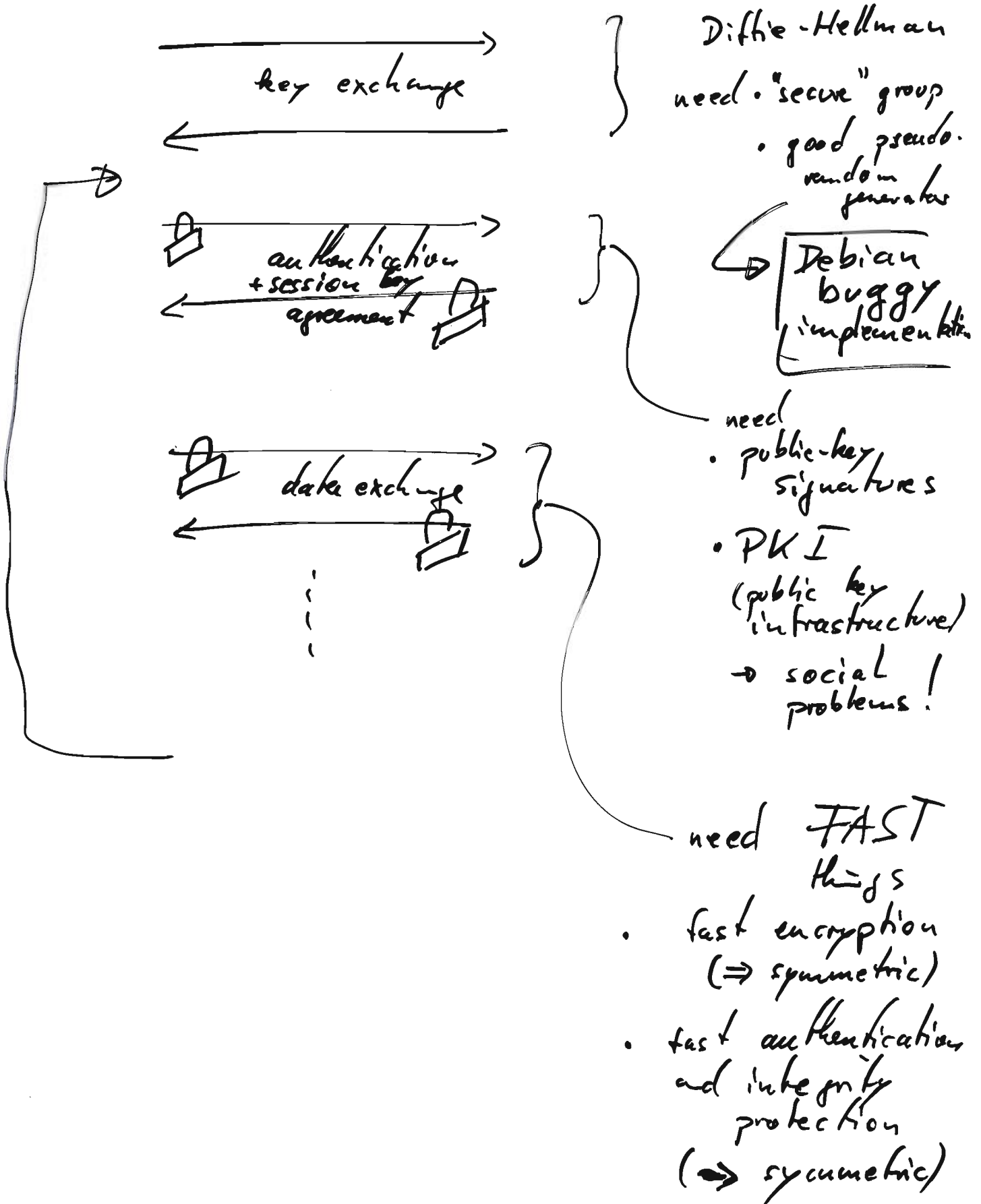
+ compression

+ NAT

+ congestion notification

# A secure connection?

SotI  
13.1.09  
①



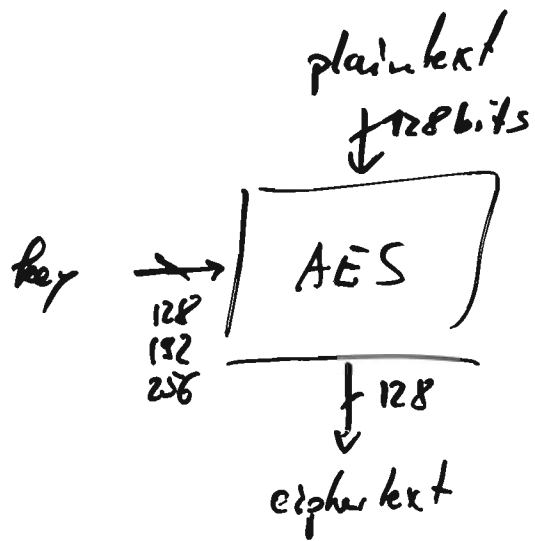


# Fast encryption and fast authentication

SotI  
14.1.09

(1)

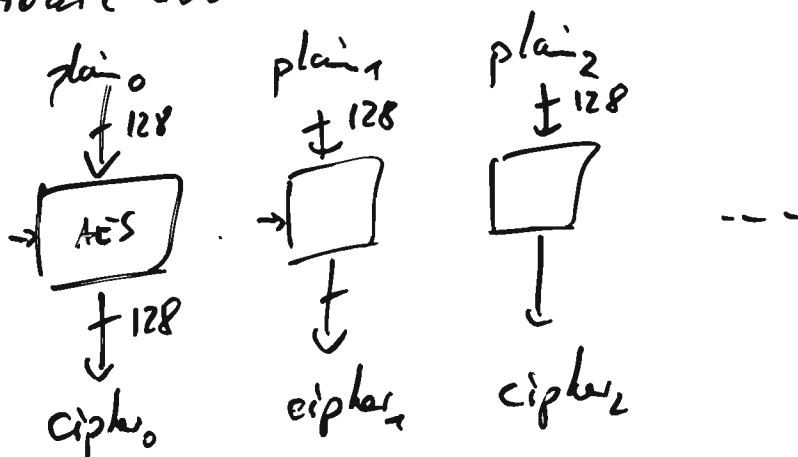
we have fast block ciphers:



How to use that kind of primitive  
to encrypt long texts?

→ Modes of Operation

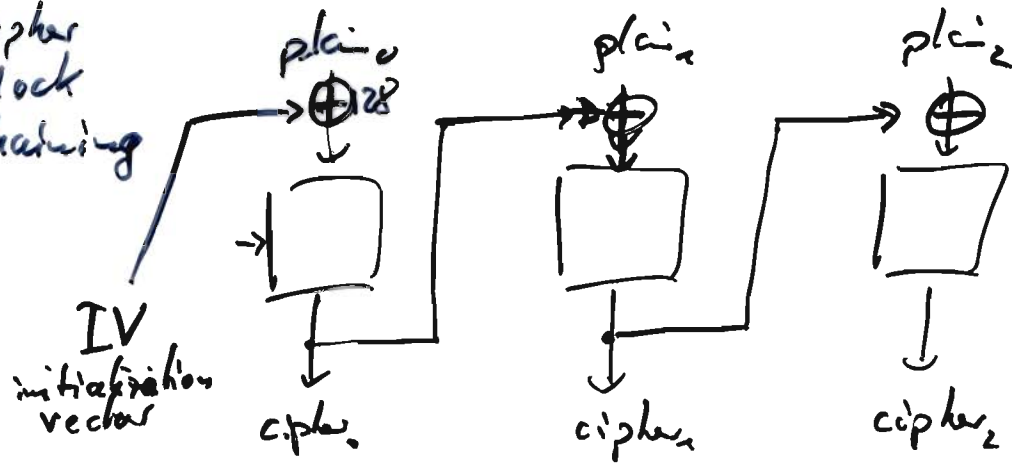
Electronic Codebook Mode



Vernam's principle  
...

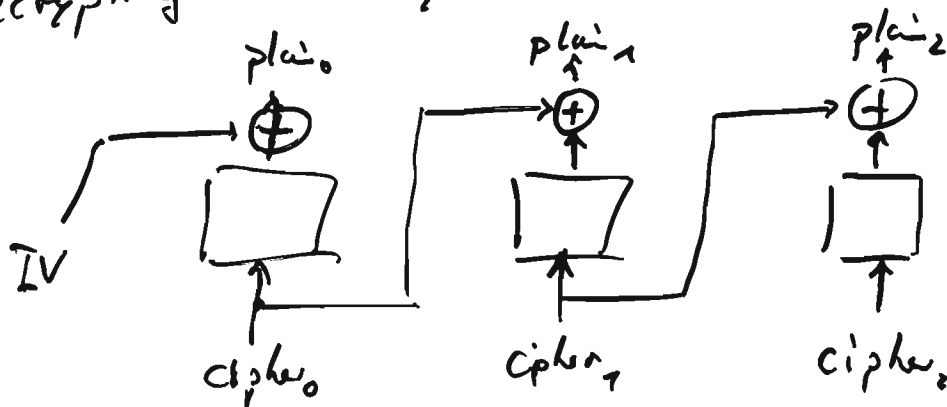
- Problems:
- same blocks are encrypted in the same way  
→ 'large' structures remain visible
  - danger of replacing or exchanging blocks underway

Cipher  
Block  
Chaining



So+I  
14.1.09  
(2)

Decryption is easy:



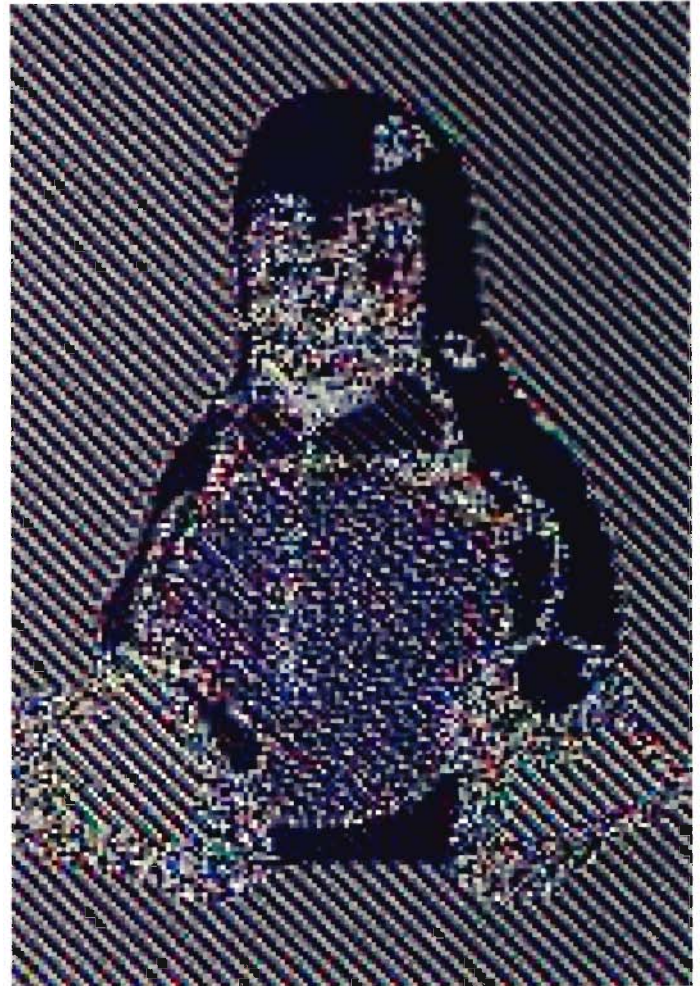
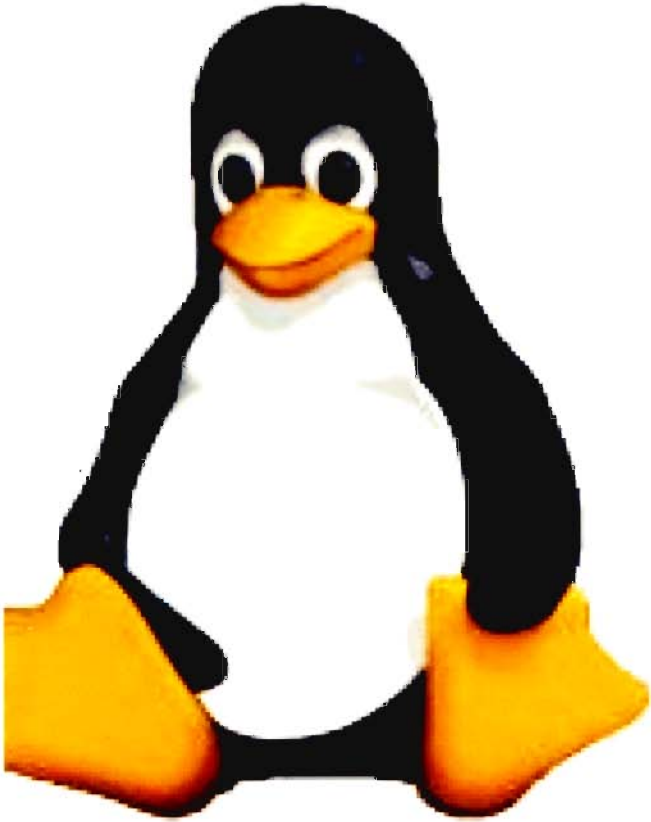
Note: if one cipherblock is corrupted or missing then at most two plaintext blocks are corrupted.

For IPsec we don't care too much because a clever administrator will always authentication to check for integrity.

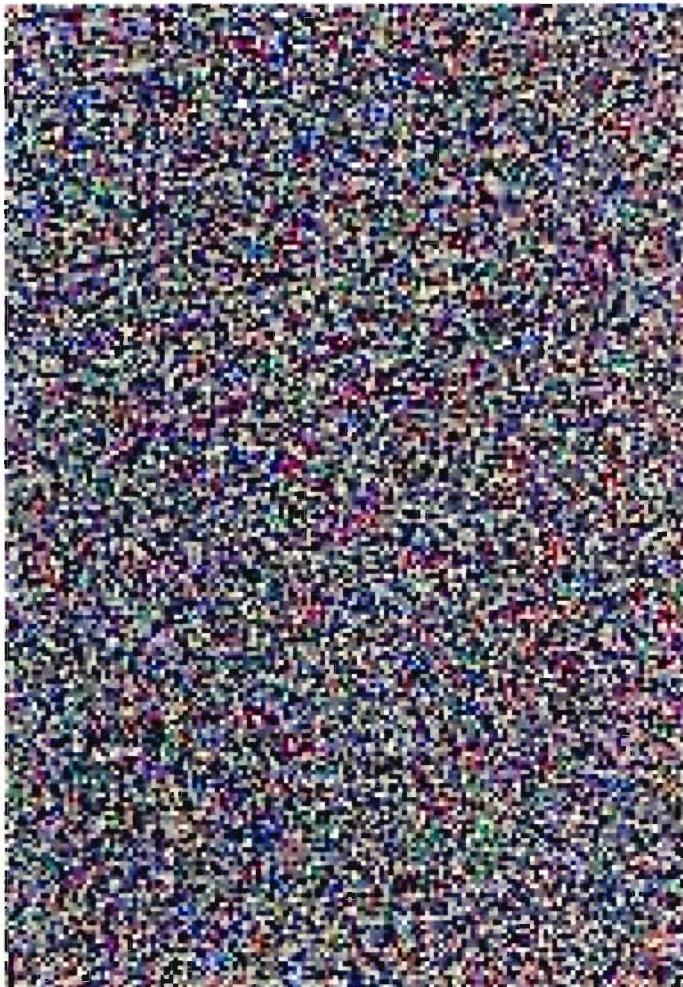
For IPsec AES-CBC is one of the allowed encryption algorithms.

Known: For fixed sized messages security of CBC can be reduced to security of the used block cipher.

Problem: Our messages are not fixed size.



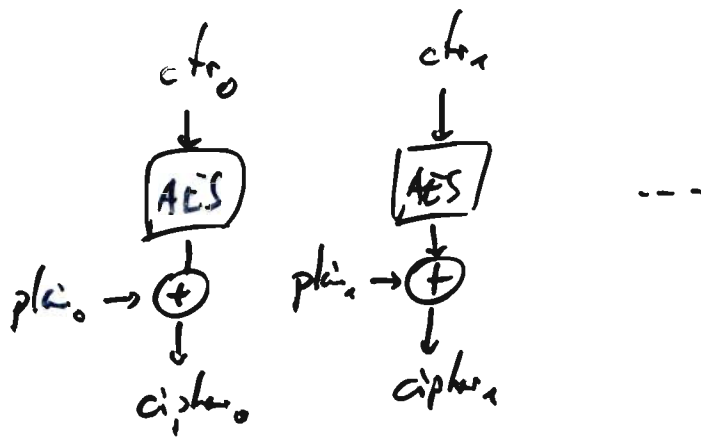
ECB mode



CBC or other  
mode

# CTR counter mode

SofI  
14.1.09  
(3)



Use IV to define where to start the counter  
(and maybe how to increment it).  
Easy to decrypt and even in case the order is garbled...

Known? security of CTR  
can be reduced to security of the  
used block cipher.

Of course now it is easy to manipulate  
the plain text at will unless there is  
an integrity check.



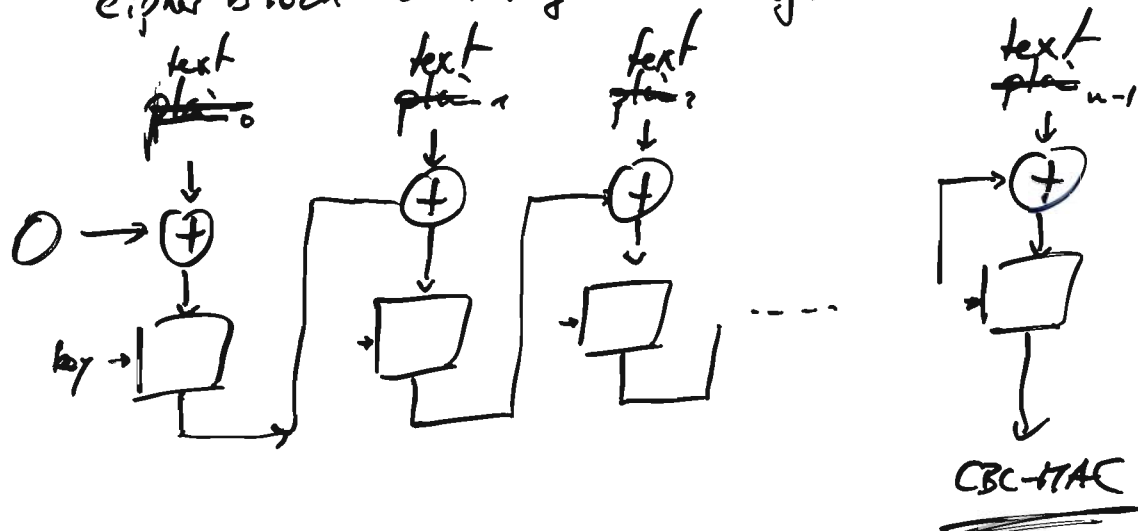
authentication?  $\rightarrow$  integrity check?

SotI  
14.1.09  
(4)

Solution 1

CBC-MAC

cipher block chaining - message authentication code

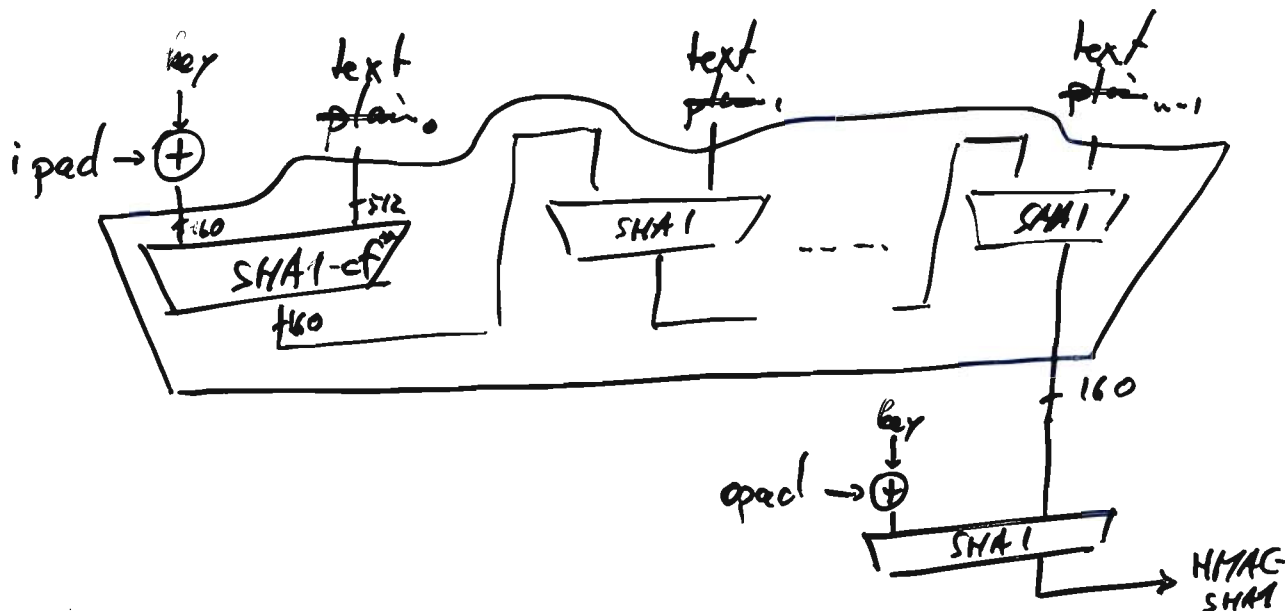


- Note:
- an attacker, actually third party, can neither generate nor check the CBC-MAC value, because it depends on a key.
  - if any plain text block is changed that (usually) affects the CBC-MAC value.

Need collision-resistance, kind of, ...

Solution 2

HMAC - SHA1



\* compression function

Fact: One can (almost) prove that  
this construction is secure  
if the used hash function  
is good.

So I  
14.1.03  
(5)

Horton hears a who?

Horton's principle

A signature (or authentication value)  
must depend on the meaning  
of the message (ie. plaintext).

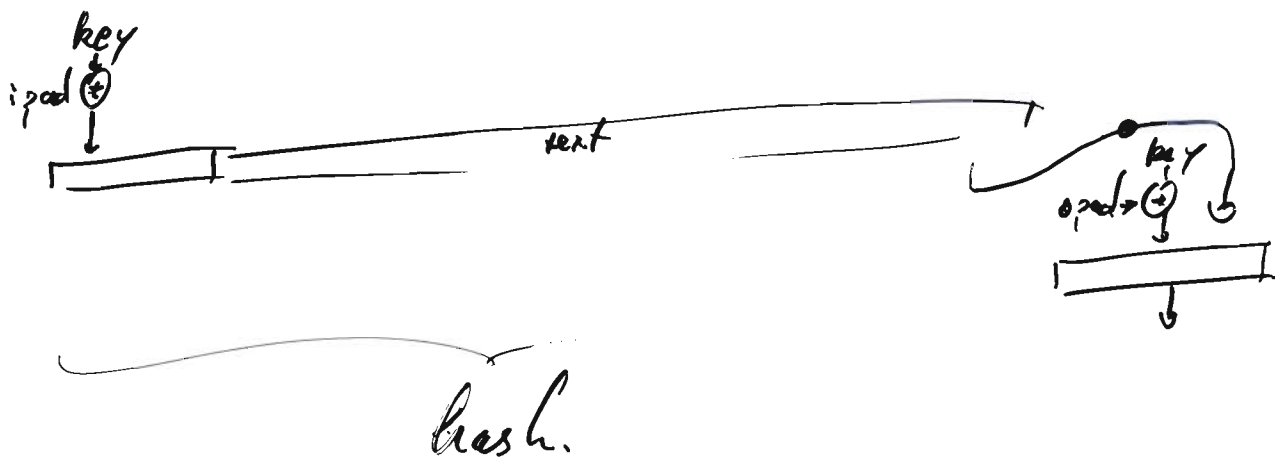
→ order of encryption  
and authentication



# More on attacking MACs

SotI  
20.1.09  
(1)

We have seen that the HMAC essentially pre- and postpends a single key to the message and hashes that.



- Q: Can't we do that simpler or more secure?
- ① why not use different keys at the beginning and the end?
  - ② Why not leave out the initial key?
  - ③ why not leave out the final key?

ad ③ EXTENSION ATTACK.  
Neglecting the hash function's padding we can extend the text and adapt the MAC.

ad

ad(2) COLLISION ATTACKS on the  
keyless hash function

SotI  
20.1.03  
(2)

If leave out the initial key  
then we can use a collision  
of the keyless function to  
obtain a collision of the keyed  
version. So it is not — as  
it should be — much more  
difficult to attack the keyed version.

Security for (keyed) MAC

~~For~~ the attacker is successful if  
he can find a collision for  
the MAC without knowing the key.  
That — due to the ignorance of the key —  
is much more difficult than finding  
a collision for a hash function.

ad(1) DIVIDE & CONQUER ATTACK

Instead of having to try all pairs of  
(key<sub>1</sub>, key<sub>2</sub>) it is enough to try  
all keys key<sub>2</sub> and then all key key<sub>1</sub>.

Eg. with length of key: being 80 bit  
needing  $2^{160}$  operations, only  $2 \cdot 2^{80}$  are enough

this is of course impractical, but it shows that it doesn't help...

Bottom line

SfS  
20.1.09  
(3)

One can prove that

- (i) if someone breaks CBC-MAC then he can also break the underlying block cipher.
- (ii) if someone breaks HMAC-hash then he can also break the underlying hash function.

Now combine authentication and encryption

How to do this?

Well, recalling Horst's principle we should

At E authenticate and then encrypt.  
and not  
E & A encrypt and then authenticate.

Because with ETA we only authenticate the cipher text. If an attacker is able to exchange one of the encryption keys the recipient would not detect a problem (unless there is structure in the plaintext).

However, in ex. 10.2 we see that  
AtE may be bad as well.

SotI  
20.109  
(\*)

Lesson: Whenever we combine  
primitives, we not ~~only~~  
on their individual security  
but we have to reduce  
the security of the new  
construction to something  
difficult, eg. the security  
of the old construction.

$P_{sec}/IKE$  actually uses EtA.

That seems to violate Horstmann's principle.

A way out of the EtA-problem would be  
to authenticate encryption key + ciphertext.  
Because this determines the plaintext, the authentication  
now authenticates the meaning of the plaintext.  
And so this variation does not violate Horstmann's  
principle.

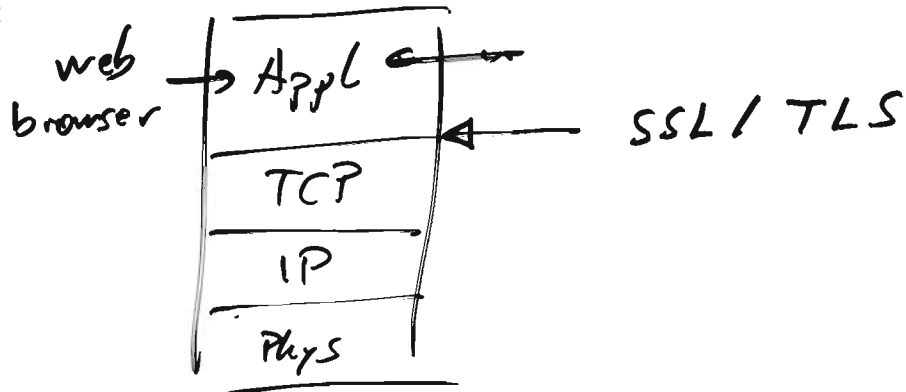
After IKE there is a (strong) connection  
between encryption key and authentication key.  
Actually, both are different sections of a certain  
pseudorandomly generated bit string. So you  
would have to break the pseudorandom generator.

People started to think about securing in particular www connections.

SotI  
21.1.09  
②

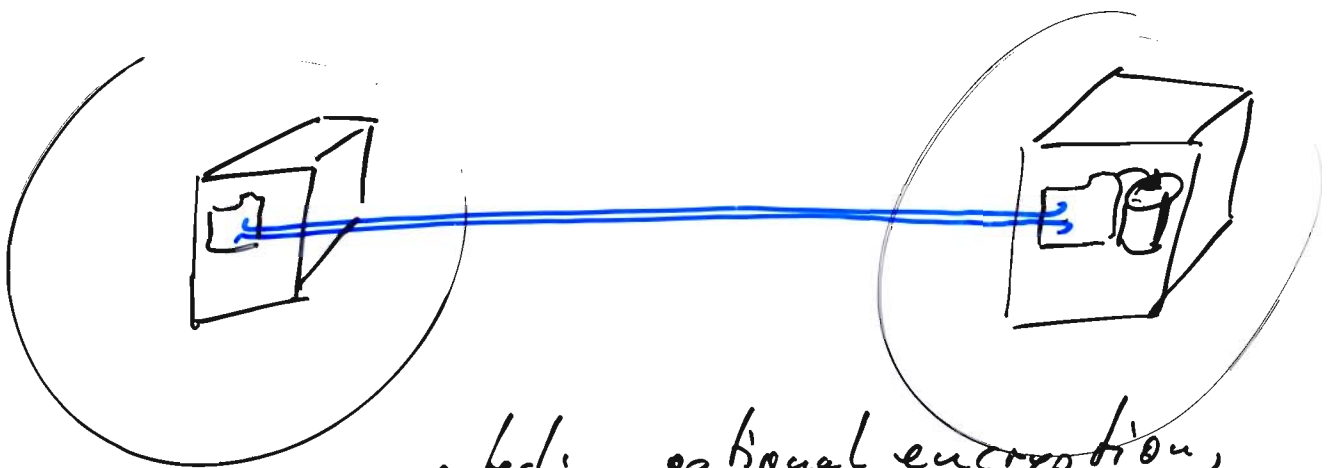
First steps : 1994 (?) Netscape  
→ SSL.

Decision :



Reasons :

- wanted fast, easily embeddable solution.
- should link application (web browser) to application (web server) rather than station to station

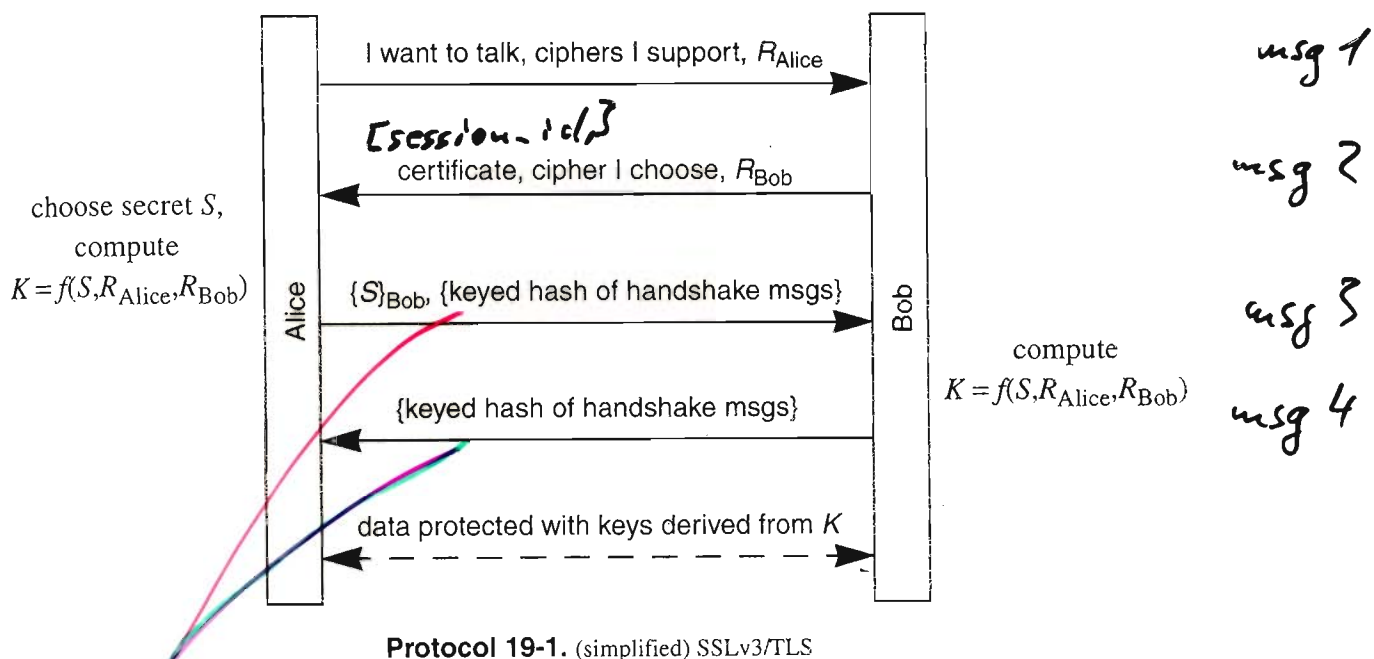


• wanted: optional encryption,  
definitely authentication

and: IPsec was not yet there.

# Shape of SSL/TLS: Initial handshake

SofI  
27.1.09  
(3)



$S$  = premaster key ( bit )

$R_{Alice} / R_{Bob}$  = nonces ( bit )

$K$  = master key ( 384 bit )  
" 3.128

one part for encryption  
one for authentication  
one for ?

hash ( 'client finished' ||  $K$  || msg 1 & 2 )  
↑ depends on version  
Re one chosen by Bob ( SSLv2, SSLv3, TLS 1.0, TLS 1.1, TLS 1.2 )  
hash ( 'server finished' ||  $K$  || msg 1 & 2 ( & 3 ? ) )



Now from  $K$  we derive

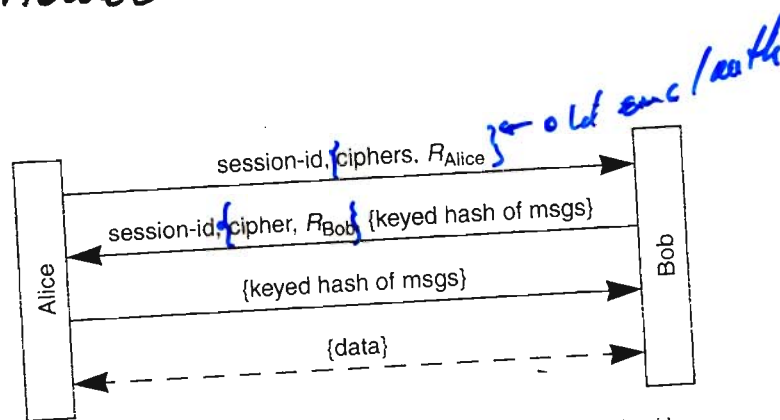
2 encryption keys,

2 authentication/integrity keys,

2 IV (for CBC or similar)

one triple for each direction.

Next: optional 'session resumption'



Protocol 19-3. Session resumption if both sides remember session-id

Further purpose: this allows to upgrade to higher security primitives better

[Background: US export restriction]  
(max. allowed: 40-bit symmetric, 512 bit RSA)  
dropped meanwhile!

SSL fulfilled this restriction by offering modes that publish 88 of 128 bits secret key.

# possible cipher suites

soft  
27.1.09  
(5)

CipherSuite	Key Exchange	Cipher	Hash
TLS_NULL_WITH_NULL_NULL	NULL	NULL	NULL
TLS_RSA_WITH_NULL_MD5	RSA	NULL	MD5
TLS_RSA_WITH_NULL_SHA	RSA	NULL	SHA
TLS_RSA_WITH_RC4_128_MD5	RSA	RC4_128	MD5
TLS_RSA_WITH_RC4_128_SHA	RSA	RC4_128	SHA
TLS_RSA_WITH_IDEA_CBC_SHA	RSA	IDEA_CBC	SHA
TLS_RSA_WITH_DES_CBC_SHA	RSA	DES_CBC	SHA
<u>TLS_RSA_WITH_3DES_EDE_CBC_SHA</u>	RSA	3DES_EDE_CBC	SHA
TLS_DH_DSS_WITH_DES_CBC_SHA	DH_DSS	DES_CBC	SHA
TLS_DH_DSS_WITH_3DES_EDE_CBC_SHA	DH_DSS	3DES_EDE_CBC	SHA
TLS_DH_RSA_WITH_DES_CBC_SHA	DH_RSA	DES_CBC	SHA
TLS_DH_RSA_WITH_3DES_EDE_CBC_SHA	DH_RSA	3DES_EDE_CBC	SHA
TLS_DHE_DSS_WITH_DES_CBC_SHA	DHE_DSS	DES_CBC	SHA
TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA	DHE_DSS	3DES_EDE_CBC	SHA
TLS_DHE_RSA_WITH_DES_CBC_SHA	DHE_RSA	DES_CBC	SHA
TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA	DHE_RSA	3DES_EDE_CBC	SHA
TLS_DH_anon_WITH_RC4_128_MD5	DH_anon	RC4_128	MD5
TLS_DH_anon_WITH_DES_CBC_SHA	DH_anon	DES_CBC	SHA
TLS_DH_anon_WITH_3DES_EDE_CBC_SHA	DH_anon	3DES_EDE_CBC	SHA

Key Exchange Algorithm	Description	Key size limit
DHE_DSS	Ephemeral DH with DSS signatures	None
DHE_RSA	Ephemeral DH with RSA signatures	None
DH_anon	Anonymous DH, no signatures	None
DH_DSS	DH with DSS-based certificates	None
DH_RSA	DH with RSA-based certificates	None
		RSA = none
NULL	No key exchange	N/A
RSA	RSA key exchange	None

Cipher	Type	Key Material	Expanded Key Material	IV Size	Block Size
NULL	Stream	0	0	0	N/A
IDEA_CBC	Block	16	16	8	8
RC2_CBC_40	Block	5	16	8	8
RC4_40	Stream	5	16	0	N/A
RC4_128	Stream	16	16	0	N/A
<del>DES40</del> _CBC	Block	5	8	8	8
DES_CBC	Block	8	8	8	8
3DES_EDE_CBC	Block	24	24	8	8

Note: each suite is a combination.  
So one can rule out bad combination. (Other than the IKE procedure.)

2: • session key agreement  
→ need PKI, application must have root certificates!

- Perfect forward security?
- Escrow attack?

Soft  
27.1.09  
⑥

We do not have with RSA-encrypted  
S sent to Bob!

Having Bob's secret encryption key (long term)  
the attacker simply decrypts  $\{S\}_{Bob}$ .

- Do S?

- No extra protection (still?)
- not that important because the connection relies on lower protocols that have DoS protection.

- Endpoint id hiding

- Server is always known publicly
- Client is always protected.

- Give partner reassurance?

Nonces  $R_{Alice}$ ,  $R_{Bob}$  prevent replays...  
Additionally: different keys for different directions  
• message numbers...

- Deniability

• No!

- Stream protection ✓

- Negotiate crypto params ✓

# Encryption & authentication in TLS

SotJ  
21.109

(7)

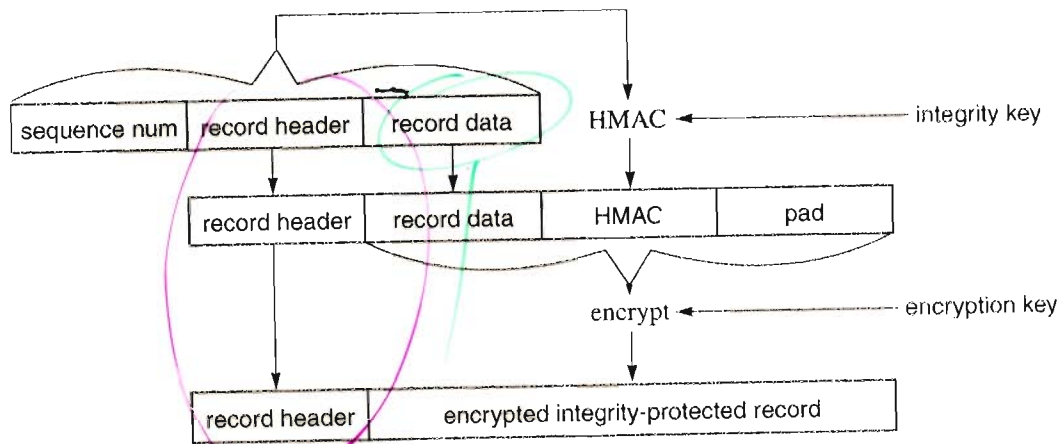


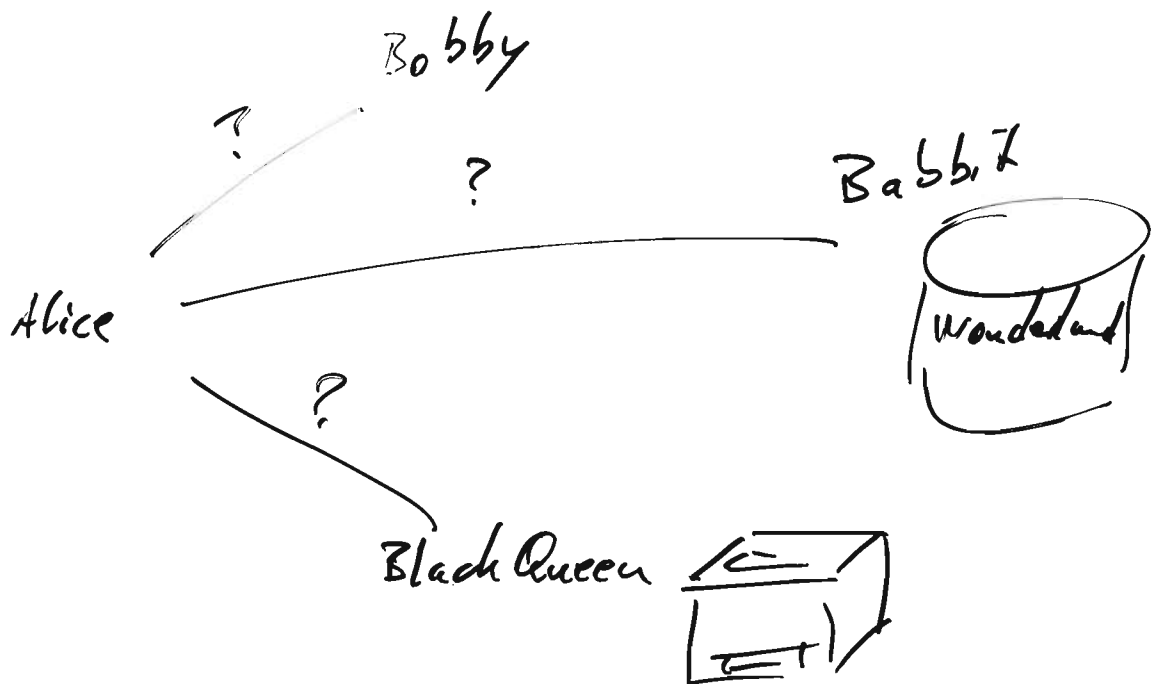
Figure 19-4. Cryptographically protected record format

- Note:
- sequence num is authenticated but never sent!
  - all header data is authenticated.
  - AE

# Authentication in a <sup>small</sup> network

SotI  
27.1.09  
①

- Is Alice allowed to access the file Wonderland on the harddrive Babbit?
- May Alice use the printer Black Queen?
- May Alice send mail to Bobby?



How do all the B's know that/whether a specific A (eg Alice) has the associated privilege/right?

## Problems

SotI  
27.1.05  
(2)

(a) authenticate Alice

→ all the B's need  
all the A keys.

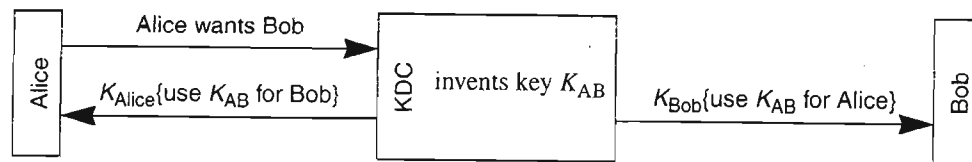
(b) privilege management

→ administration needs  
to (re) inform / configure  
each B separately.

(c) each pair A-B needs a shared key  
for fast secured data exchange.

Possible solution: a trusted third party,  
key distribution center.

Idea:



Protocol 11-16. KDC operation (in principle)



## Pros & cons:

SotI  
27.1.09  
(3)

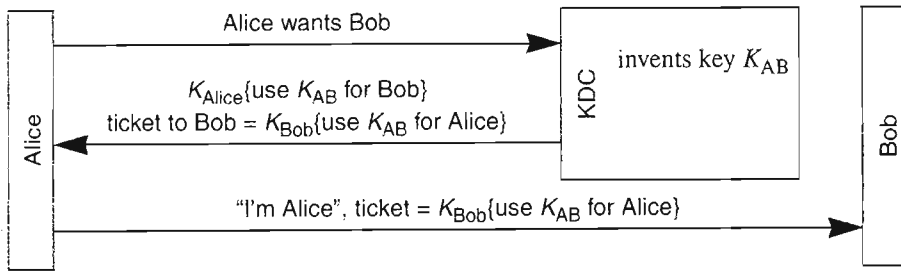
- + few keys stored at one place only  
(Alice needs  $K_{\text{Alice}}$  and the key center needs all keys)
- + (re)configuration at one place
- + each pair Alice-Bob gets a short-lived session key whenever needed (and allowed)
- single point of failure (if key center is down, nothing works)
- performance bottleneck
- initial effort for setup is large
- if the key<sup>center</sup> is corrupted all authentication is lost. Actually, the key center can impersonate every one.

The cons may be weakened by duplicating the key center....

- + each user only needs to authenticate once to the key center.

Actually, we do it slightly different.

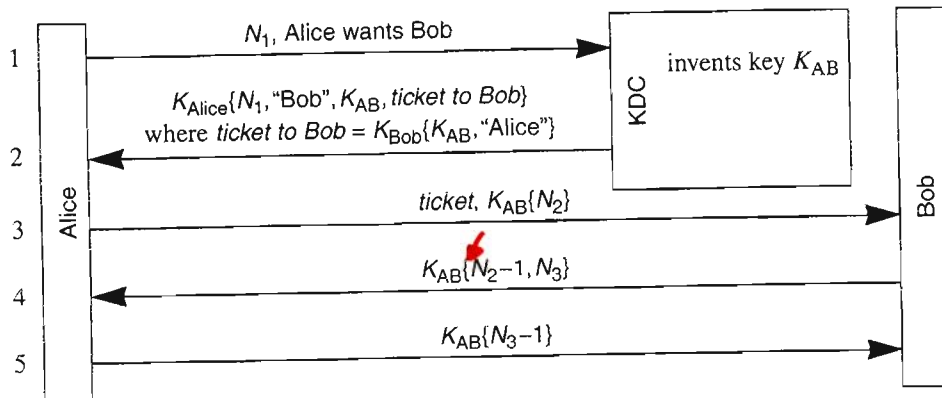
SotI  
27.1.09  
(4)



Protocol 11-17. KDC operation (in practice)

Here, a traitor can just replay the third message

Needham-Schroeder protocol



Protocol 11-18. Needham-Schroeder

$N_1$  nonce that grants that the supposed key center has the shared key  $K_{\text{Alice}}$ .

$N_2$  nonce that grants to Alice that she is talking to Bob.

$N_3$  nonce that prevents replays. That grants to Bob that he is talking to Alice.

Wandering:

SotI  
27.1.09

(5)

one could think of a replay attacker  
that uses parts of message 4  
to obtain a valid answer as message 5.

(Idea: start a second connection  
with ticket,  $K_{AB}$  &  $N_3$  . )

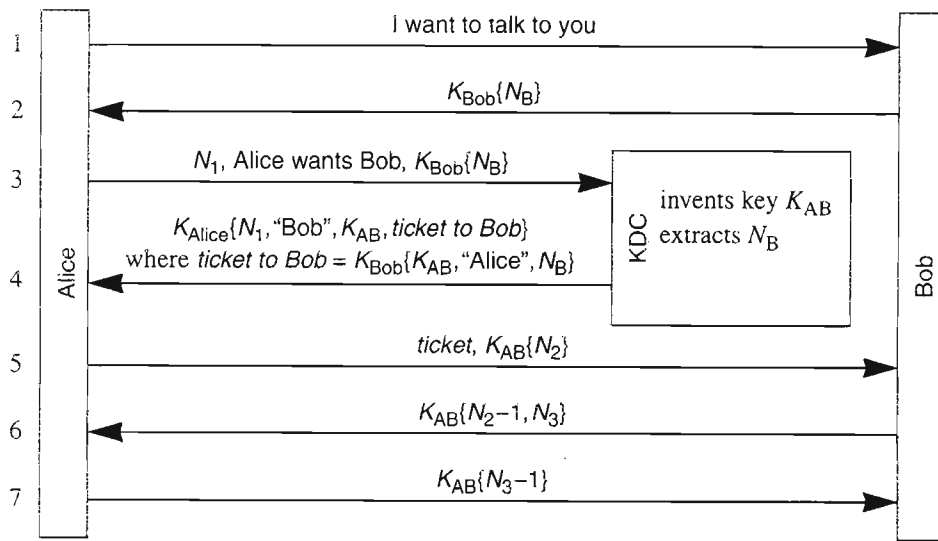
That is prevented if each message has  
an integrity check!

A further attack comes when Trudy, the tracker,  
is able to steal Alice's key and  
get a bunch of tickets before Alice  
notice and revokes her key (and logs in  
again to the key anchor).

Problem: the tickets stay valid!

A possible solution to this problem!

SotJ  
27.1.08  
(6)



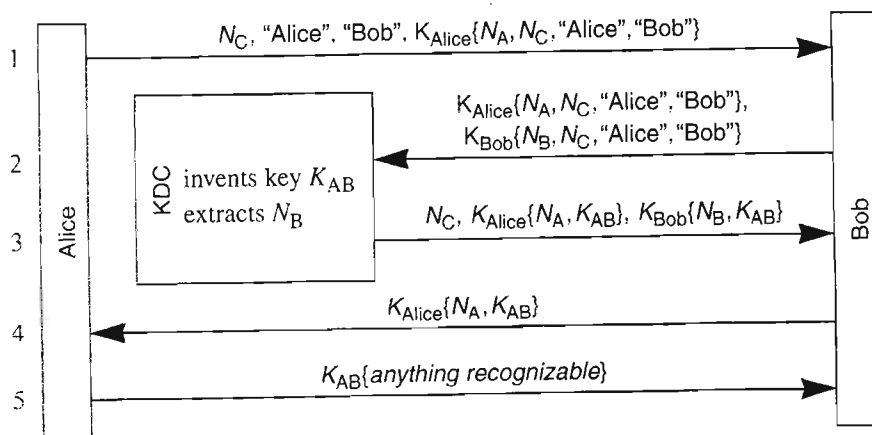
Protocol 11-19. Expanded Needham-Schroeder

} start  
timeout!

This now grants Live Partner Reassurance.

(EX) Reduce this to six messages.

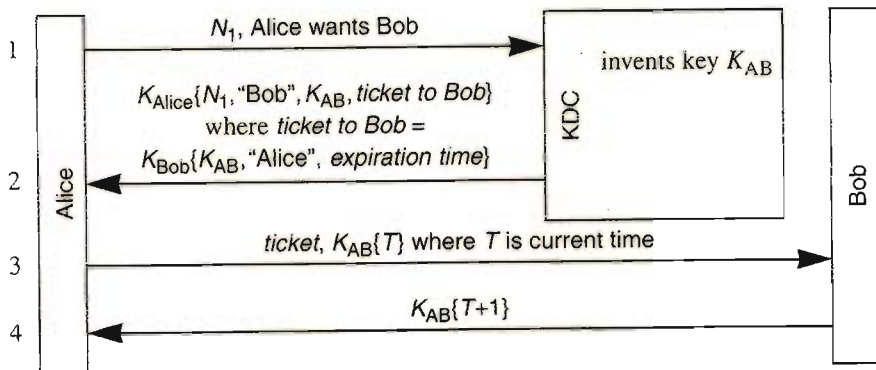
variant Otway - Rees protocol



Protocol 11-20. Otway-Rees

} timeout

The Kerberos authentication service (see Chapter 13 *Kerberos V4*) is roughly based on the Needham-Schroeder protocol. It looks a lot simpler than these protocols because it assumes a universal idea of time, and includes expiration dates in messages. The basic Kerberos protocol is:



**Protocol 11-21.** Kerberos

# Kerberos Authentication

## Server

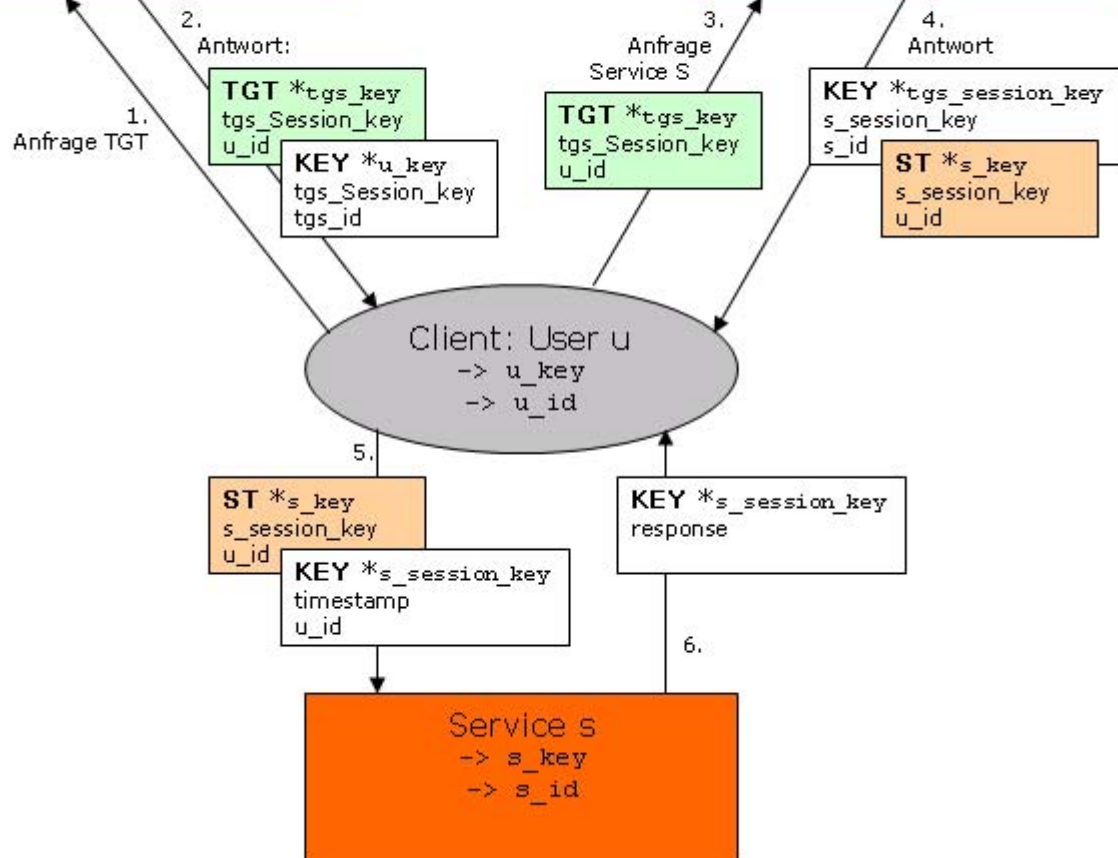
-> u\_key  
-> s\_key

## Realm DB

u\_id->u\_key  
s\_id->s\_key

## Ticket Granting Service

-> u\_key  
-> s\_key  
-> tgs\_key  
-> tgs\_id





# Kerberos V5

Soft  
28.1.09

①

History : • Needham-Schroeder : 1979(?)  
• Kerberos was developed at MIT  
in the early 1980s,  
V5 in the early 1990s.

RFC 4120 : July 2005.

even here : only 4 out of 139 pages  
deal with security  
considerations.

'Own' security remarks:

- "By itself Kerberos does not provide authentication."
- Denial of Service attacks are not solved.
- Interoperability conflicts
- Password-guessing  
The tickets are derived from passwords in a way that allows offline dictionary attacks. (Passive!)
- V4 only used DES-CBC-MD5.
- ~~even~~ V5 still has DES-CBC-MD5 as a SHOULD.

~~Everything~~ Everything relies on synchronized clocks!

- No identifier hiding.
- No perfect forward security.

SotI  
28.10.9  
(2)

## Schneier on Kerberos security

- some replay seem possible,  
in particular if a ticket lifetime  
has not expired  
or one manipulate  
clocks
- Kerberos (or any such system) is  
vulnerable to malware.

## Security model?

- Lowe :
- attacks to Needham-Schroeder  
(as seen yesterday)
  - formulated a change that he  
proved secure in a certain  
model.

- Wieruschi :
- Needham-Schroeder-Lowe publicly  
is still insecure in practice <sup>version</sup>

key center

SotI  
28.1.09

(3)



The attacker transports all messages.

He is further <sup>allowed</sup> to corrupt some (but not all) parties.

Aim: Mutual authentication of <sup>uncorrupted</sup> Alice and Bob,

Attacker tries to 'perform' an authentication to Alice or to Bob without corrupting any of the two.

Then For any asymmetric encryption scheme  
Needham-Schroeder based on it  
is insecure.

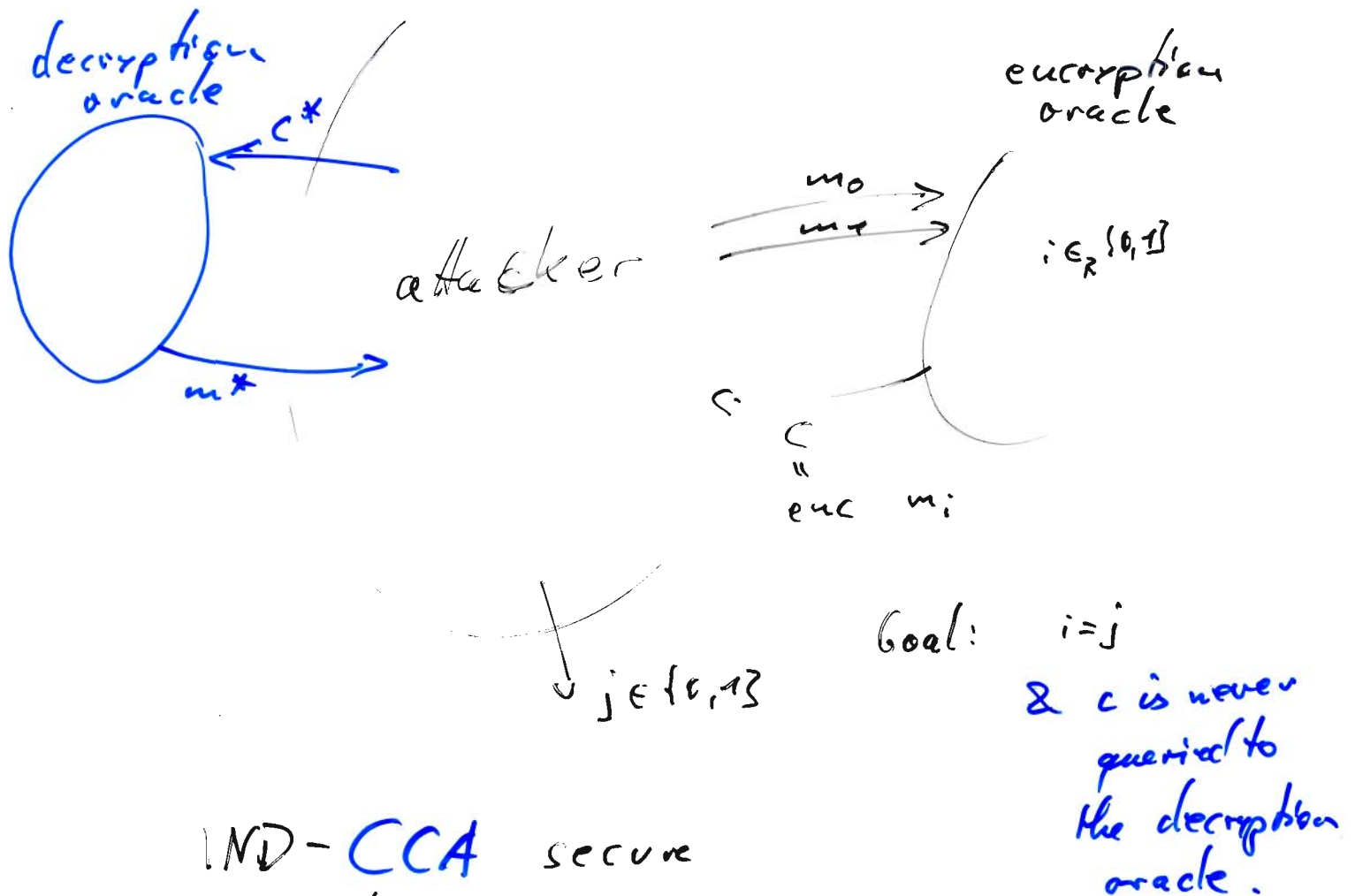
Then There exists an <sup>IND-CPA secure</sup> asymmetric encryption scheme  
such that Needham-Schroeder-Lowe  
is still insecure.

ElGamal encryption scheme

Then If we base Needham-Schroeder-Lowe) SoTJ  
 on a **IND-CCA** secure asymmetric encryption scheme then  
 it is secure in the above sense

28.1.09  
 (4)

For an encryption scheme IND-CCA secure means that the attacker cannot distinguish two self chosen messages even if ~~it~~ is allowed to get ~~messages~~ ciphertexts decrypted!



IND-CCA secure means that there is no successful poly time attacker.

# ElGamal encryption

SotI  
28.1.09  
(5)

Global setup: a group  $G$ , an element  $g$   
of known order  $q$   
(such that  $DL$  with basis  $g$   
is difficult)

Personal setup: secret key  $\alpha \in \mathbb{Z}_q$ ,  
public key  $a = g^\alpha \in G$

Bob

> Alice

$\tau \in_R \mathbb{Z}_q$   
(temporary secret)

$$t = g^\tau \in G$$

$$y = a^\tau \cdot x$$

$(t, y)$

$$t^{-\alpha} \cdot y$$

"   
 x

This scheme obviously (?) is not IND-CCA secure.

Proof The attacker chooses  $m_0, m_1$  different,  
and asks for an encryption  $c = (t, y)$   
of one of them. Then he computes

$$c^* = (g^\alpha t, a^\alpha y)$$

and asks the decryption oracle for its  
decryption  $m^*$ . So he can see whether it was  
 $m_0$  or  $m_1$ . □

However, one can prove that  
it is IND-CPA secure.

---

SotI  
28.7.09

(6)



# Overview

SotI  
3.2.09

1

## Primitives

	Symmetric	Public-key
Confidentiality	Encryption (AES, ...)	Encryption (RSA), ElGamal encryption, ...)
Integrity	Message Authentication Codes e.g. HMAC-SHA1 AES-CBC-MAC	Signature (ElGamal type signa e.g. ElGamal, DSA, ECDSA, (Schmorr) ...)
Authenticity (UnDeniability)	—	—
Key agreement		Diffie-Hellman
Fingerprint	Hash functions: <del>SHA1</del> , <del>ECDS</del> , ... SHA2	



# Protocols

SoFI  
3.2.09

(2)

IPsec/IKE  $\approx$  DH + Authentication + Symm. Enc.  
+ MAC  
+ Certificates/PKI

TLS/SSL  $\approx$  — " —

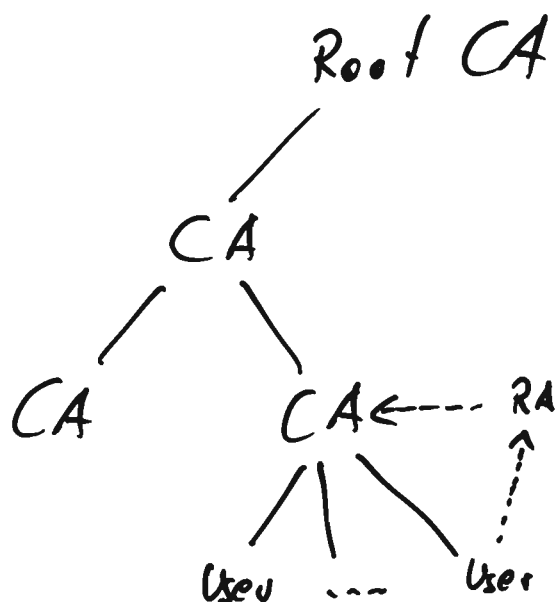
SSH  $\approx$  — " —

Kerberos  $\approx$  Public encryption + Symm. Enc. + MAC.

PGP  $\approx$  Public<sup>key</sup> encryption, public key signatures,  
Symm. encryption.

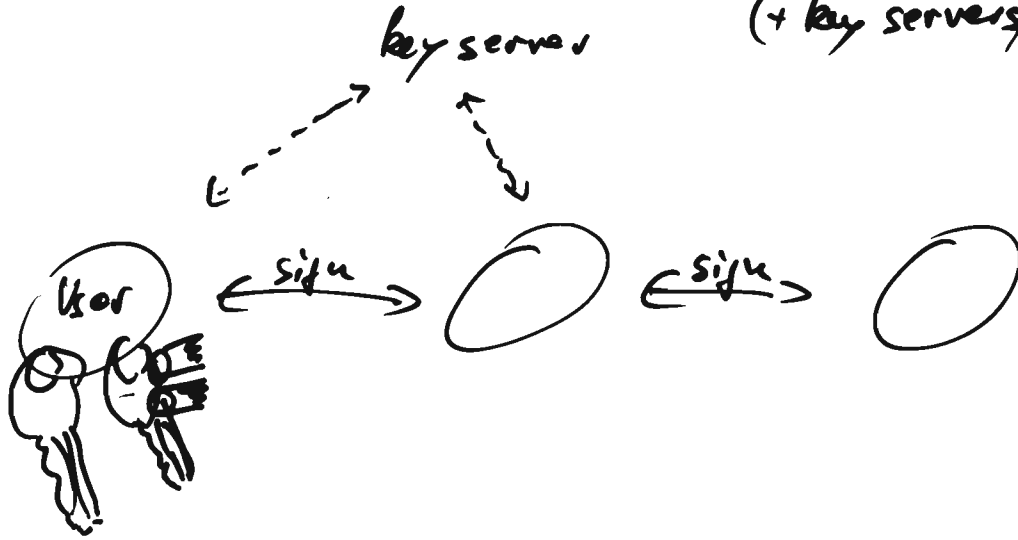
# Structures

PKI  $\approx$  Signatures + Certificates +  
Trusted Third Parties



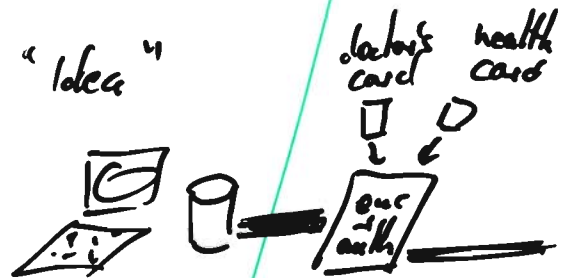
Web of trust  $\approx$  Signatures + Certificates + Trust Rules  
 "Anarchy model" (+ key servers)  
 (PGP)

Sot2  
 3.2.09  
 (3)



## Solutions

- electronic banking
  - dedicated solutions ("old")
  - TLS/SSL over http.
- (secure email)
- virtual private networks
- electronic health card + infrastructure
- citizen portals ("Bürgerportal")



- (electronic elections)
- (banking card)
- electronic passports and id-cards.
- e commerce  
 (usually just TLS/SSL over http)

Need anonymity  
 and verifiability.

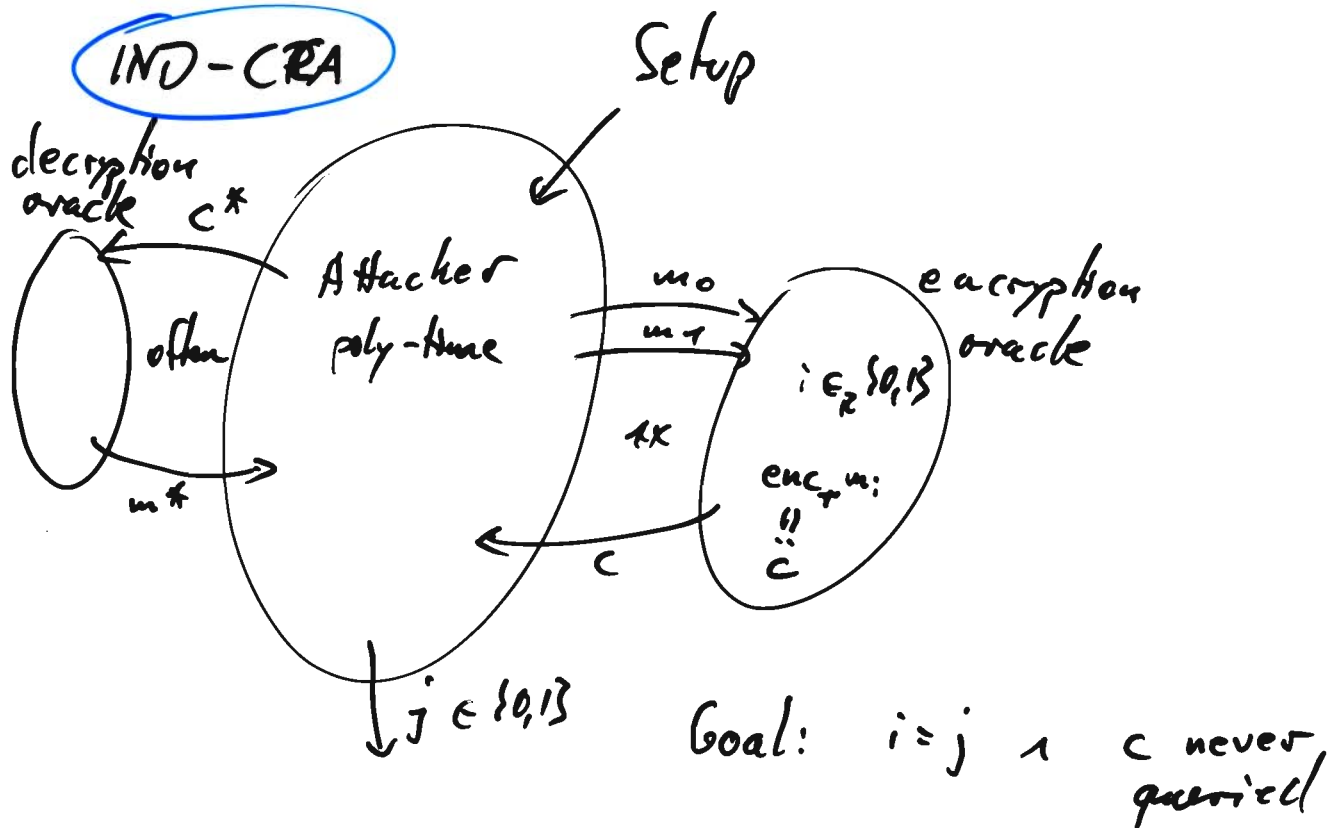
# The final cut: Security?

SoFI  
3.2.09  
(4)

Security model:

For encryption:

IND-CRA



Security notion: There is no such attacker.

Helpful for finding

- necessary conditions
- sufficient conditions (security reductions)

For signatures:

EF-CMA

Setup

SotI  
3.2.05  
(5)

Signing  
oracle

$m^*$

often

$s^*$

Attacker  
(poly-time)

$(m, s)$

Goal:  $(m, s)$  is a valid signature  
&  $m$  never queried

Security notion: There is no such attacker.



# Your questions:

- (1) Repeat major issues
  - Pollia-Hellman
  - Pollard's, Baby-Step-Giant-Step
  - Chinese Remainder Algorithm
- (2) ...
  - IPsec
  - Modes of operation
  - Security notions
- (3) ...
  - Key Exchange Threats

CRT If  $m, n$  are coprime then

$\mathbb{Z}_{mn} \longrightarrow \mathbb{Z}_m \times \mathbb{Z}_n$  is an isomorphism.  
 $(a \bmod mn) \mapsto (a \bmod m, a \bmod n)$

$\mathbb{Z}_{12}$	0	1	2	3
0	0	9	6	3
1	4	1	10	7
2	8	5	2	11
$\mathbb{Z}_3$				

$\mathbb{Z}_4$

compatible  
with  $+$ ,  $\cdot$

DLP: Give  $g, a \in G$   
 Find  $x \in \mathbb{Z}_{\text{ord}(g)}$  such that  
 $g^x = a$

SotI  
 4.7.09  
 (2)

Pohlig-Hellman

... uses the factorization of  $\text{ord}(g)$   
 to break down the DLP to  
 easier instances.

If  $g^x = a$  then  $(g^k)^x = a^k$ .

Choose nice  $k$ 's!

Best:  $\frac{\text{ord}(g)}{k}$  is prime  
 in case  $k \mid \text{ord}(g)$ .  
 $\text{ord}(g^k) = \frac{\text{ord}(g)}{k}$

The new problem determines  $x$  modulo  $\text{ord}(g^k)$ .

Altogether we can obtain

$x \bmod p$  for any  $p \mid \text{ord}(g)$ .

Actually, we can extend that to

$x \bmod p^e$  for any  $p^e \mid \text{ord}(g)$

by writing  $x$  in base  $p$  (Details ...)

and choosing  $k = \frac{\text{ord}(g)}{p^f}$  for  $f = 1, 2, 3, \dots, e$ .

Overall runtime:  $\sum e_i \cdot \text{DLP}(p_i)$  where  $\text{ord}(g) = \prod p_i^{e_i}$   
 $\in O(\text{DLP}(\text{largest prime}) \cdot \prod p_i^{e_i})$

To solve DLPs in general and  
in particular if  $\text{ord}(g)$  is prime

SofD  
4.2.09  
(2)

use Pollard- $g$

or Baby-Step-Giant-Step.

In baby-giant we write  $q = \text{ord}(g)$ .

$$\alpha = \alpha_1 \cdot \sqrt[q]{q} + \alpha_0$$

and consider

$$\left(g^{\sqrt[q]{q}}\right)^{\alpha_1} = g^{-\alpha_0} a$$

So we only need runtime  $O(\sqrt{q})$ .

Improvement: Pollard- $g$  including Floyd's trick.

→ runtime: expected  $O(\sqrt{q})$ .

memory:  $O(1)$

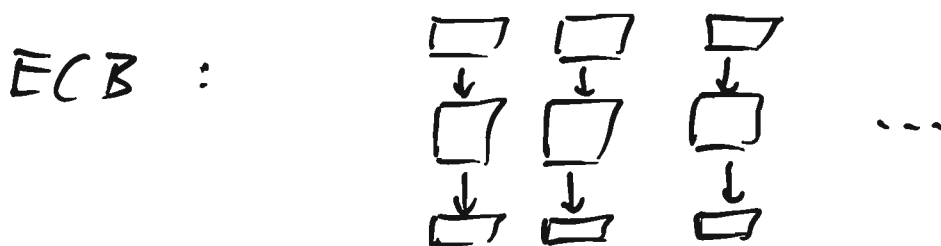
# Modes of operation

SotI  
4.2.09  
(4)

We have block ciphers, many good ones.  
They can encrypt a fixed sized block  
using a (fixed sized) key.

We need to construct ciphers that can  
encrypt arbitrarily long messages.  
"from short to long"

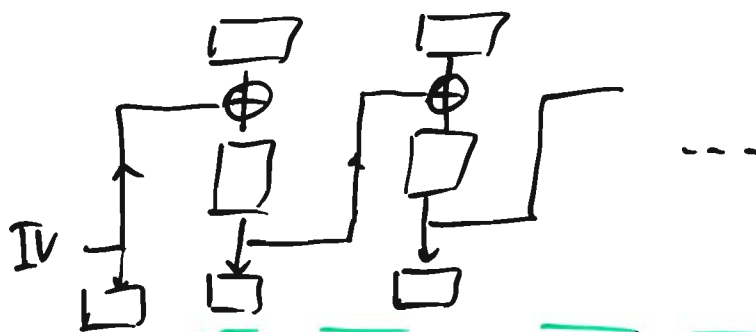
Various solutions:



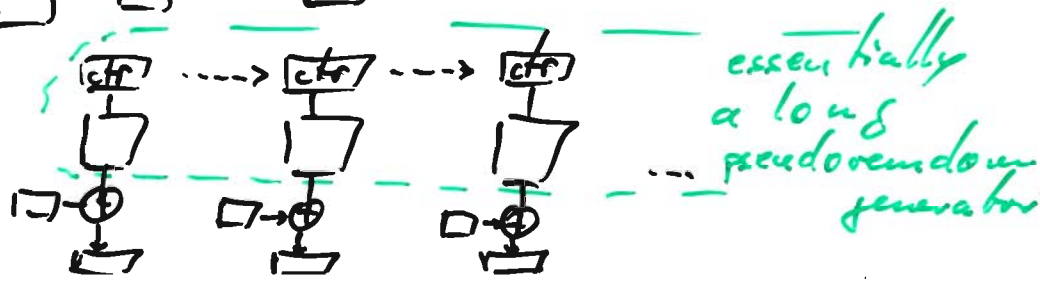
We should prove: if this can be attacked  
then a single block cipher entity  
a little can be attacked.

ECB is <sup>a little</sup> weak because it will reveal  
some patterns in the message...

CBC



CTR



# Threats & Security

SotJ  
4.2.09  
(5)

Various attack aims:

- Perfect Forward Security, Escrow foilage
- Identifier Hiding
- Denial of Service

...

Security Notions

... see yesterday.

IPsec

... see notes of 13 January...