Lecture Notes

Security on the Internet

Michael Nüsken

b-it

(Bonn-Aachen International Center
for Information Technology)

summer 2007

©2007 Michael Nüsken


Expectations
Trojans Protection
SECURE DATA COMMUNICATION (ONLINE BANKING, IDENTIFIKATION)
Secure Transmission
I hope ...
good marks!
8 Credits
SPAM
... this:
Secure communication over insecure lines
how to handle DOS/DDOS
avoiding
- SQL injection
- XSS attacks
- Trojans
- buffer overflow attacks
etc.
Authentificat
Secure message exchange over unsecure networks
a small group that is good for discussion
learn something about
formation security
at is valuable in
actice
Details of SSL, TLS
VPN, ...
Firewall
NNELING
Security on internet
Encryption
... not this:
enjoy the nice building while it's still possible ;-)

Internet
no one owns it!
Technical part
• network of networks
• client - server / peer - to - peer
• types of data - transmission:
→ duplex
→ point - to - point
→ multicast
→ broadcast
• media of transmission:
→ wireless
→ wired
• Routing (Software & Hardware)
→ static / dynamic
• Hardware involved:
→ Network card / modem (client)
→ switches / routers / card (client & provider)
• Software involved:
→ Browser (client)
→ Email-client / VPN-client ..
→
Social Part
• information (to get, fake!, and to provide)
• communication (email, chatting)
• entertainment
• services ( )
• → internet marketing
→ educational
• open to everyone.
• provide social groups (youtube, orkut)
• missuse of information, network
→ piracy
→ propaganda
→ anomimity
→ addiction
Virtual Reality
World Wide Web
protocol
standards

Security
Authorization
Accesscontrol
Illegal use
Reliabilit
Patch Day
biometry
Passwords
Trust
Authentication
Certificates
Identity theft
Security
"Data"
"Health"
"Economic"
Integrity
Hash-code
Signing
Hashing
Firewall
Privacy
„Eavesdrop"
Encryption
sym
asym.

Email

11.4.07
1

Goal

- send ~~moderate~~ message, text-only
  -size
- fast
- ~~to spe fic~~
- to specific destination
- from same source
- easy to use
- reliable (msgs should arrive, at least in most cases), available.
- cheap
+ multiple destinations
- asynchronous
- no acknowledgement

Format

- split into:

From <address> ... <date> ...

Header — <keyword> : <information>

empty line

Body — any text + terminator

- text only

Technicalities

- receive any mail (process)
- relay / forward mails via various servers

11. 4. 07 ②

- address information must be included and non-encrypted

- DNS servers necessary to provide information about the topology of the network

- SMTP (Send Mail Transfer Protocol

Reliability — format syntax — random attacks (bad weather, bad cables, ...)

DoS

Security — content semantics — intentional attacks

Security objectives

- only the intended addressee/recipient gets the mail
- make sure that the sender is who he claims to be;
- make sure nobody uses my address as sender address
- protect content from disclosure
- protect content from modification

11.4.07
(3)

Attack

Fake internal senders from hijacked mail server

Slightly faster DNS server to redirect requests to keyserver

Stop mail server by DoS

Send mails with invalid domain names to block local DNS

Flood with mails

Flood

Read incoming mail

Defend

Distributed Infrastructure

Digital signatures

Greylisting

Public key infrastructure

Key server with certificates and root certificate in locked room

Fingerprint

Distributed infrastructure.

Block affected local mail server

Can ~~easily~~ tell difference ~~between~~ ~~with~~ internal/extern

New servers

Sendback/ Re-DoS

Encrypt!

Summary: Email

16.4.07
1

Security objectives

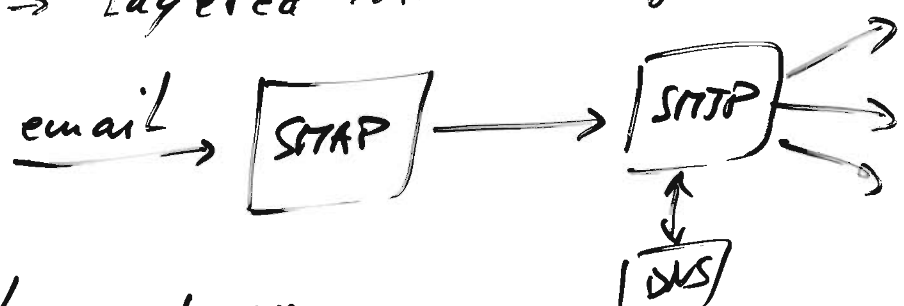
- Protect content from disclosure
  from modification
- Identify sender.
- Protect receiving host
  from attacks by incoming messages
- Mailing list handling (many recipients)

Basics

- Address (& more) in the message (headers)
- Text only
- Accept from anywhere
- No acknowledgement

Attacks

- on server — exploit vulnerabilities
  Solution: No ultimate one.
  → layered software, eg. smap



- on clients - hoaxes
  eg "Goodtimes"

16.4.07
2

Goodtimes virus ~ 1994

Here is some important information. Beware of a file called Goodtimes.

Happy Chanukah everyone, and be careful out there.There is a virus on America Online being sent by E-Mail. If you get anything called "Good Times", DON'T read it or download it. It is a virus that will erase your hard drive. Forward this to all your friends. It may help them a lot.

→ uses bandwidth
→ uses human time for explaining

- phishing
- 'evil' attachments
  → automatic ~~or~~ semi automatic execution
  (macro language, executables, ...)
- spam

Technology

1. Encryption
   → confidentiality: protect from disclosure

2. Signature
   → identification: identify sender
   → integrity: protect from modification
   → authentication, undeniability (non-repudiation): link document & sender/signer

3. Public Key Infrastructure

ENCRYPTION

Cesar

Replace every letter in the plain text with its third successor.

$$\begin{array}{ccc} YHQL & YHGL & YLFL \\ \uparrow \text{enc} & & \\ VENI & VEDI & VICI \end{array}$$

We have an alphabet

$$\Sigma = \{ \underset{\overset{\shortparallel}{0}}{A}, \underset{\overset{\shortparallel}{1}}{B}, \underset{\overset{\shortparallel}{2}}{C}, \dots, \underset{\overset{\shortparallel}{25}}{Z} \}$$

and the possible Cesar ciphers are:

$$C_i : \Sigma \longrightarrow \Sigma,$$
$$a \longmapsto (a+i) \text{ rem } 26.$$

rem ↑ remainder

To decrypt without knowing the key it suffices to try out all 26 keys. } Brute force attack.

Better attack: find most frequent character. This must ~~be the~~ correspond to the most frequent character of the plain text's language. } Frequency analysis

Even better: affine Codes:

$$A_{\alpha,i} : \Sigma \longrightarrow \Sigma,$$
$$a \longmapsto (\alpha a + i) \text{ rem } 26.$$

16.4.07
5

We have to care that decryption is possible: [CORRECTNESS]

For $\alpha = 1$ we a generalized Cesar C_i which is ~~simple~~ to decrypt (by C_{-i}).

But $\alpha = 0$ is very bad, any character be mapped to the same and no decryption is possible

With $\alpha = 2$ we always have

$$\begin{array}{ccc} A_{2,i}(\overset{\overset{A}{\shortparallel}}{0}) & = & A_{2,i}(\overset{\overset{N}{\shortparallel}}{13}) \\ \shortparallel & & \shortparallel \\ (2 \cdot 0 + i) \operatorname{rem}_{26} & & (\underbrace{2 \cdot 13}_{} + i) \operatorname{rem} 26 \\ \shortparallel & & \shortparallel \\ i & & i \end{array}$$

Thus we cannot decrypt.

The mathematical structure we need here is the ring of integers modulo 26.

This is an OO-class consisting of

class $\mathbb{Z}_{26}$
- a set of legal values: $\mathbb{Z}_{26} = \{0, 1, 2, \ldots 25\}$,
- two operations
  - $+ : \mathbb{Z}_{26} \times \mathbb{Z}_{26} \longrightarrow \mathbb{Z}_{26}$, $(a, b) \longmapsto (a+b) \operatorname{rem} 26$.
  - $\cdot : \mathbb{Z}_{26} \times \mathbb{Z}_{26} \longrightarrow \mathbb{Z}_{26}$, $(a, b) \longmapsto (a \cdot b) \operatorname{rem} 26$.

16.4.07

⑥

$\mathbb{Z}_{26}$

Properly defined: there is a set and the operations are well defined.

$P+$, $P\cdot$

Associativity: $a+(b+c) = (a+b)+c$

$A+$, $A\cdot$ $\quad a\cdot(b\cdot c) = (a\cdot b)\cdot c$

Neutral element(s): there is an element $0 \in \mathbb{Z}_{26}$:

$N+$, $N\cdot$ $\quad a+0 = a = 0+a$

there is an element $1 \in \mathbb{Z}_{26}$

$a\cdot 1 = a = 1\cdot a$

Inverse elements $\quad \forall a\ \exists b \quad a+b=0=b+a$

$I+$

Commutativity $\quad a+b = b+a$

$C+$

Distributivity: $\quad a\cdot(b+c) = a\cdot b + a\cdot c,$

$(a+b)\cdot c = a\cdot c + b\cdot c$:

$0 \neq 1$

Sometimes (for us almost always) we further

Commutativity: $\quad a\cdot b = b\cdot a$

$C\cdot$

commutative ring: $\quad PANIC+,\ PANC\cdot,\ D,\ 0\neq 1.$

If we further have $\quad \forall a \neq 0\ \exists b: \quad a\cdot b = 1 = b\cdot a$

$I\cdot$:

then we call it a field.

Examples

$\mathbb{R}$: ring, comm. ring, field.

$\mathbb{Z}$: ring, comm. ring, not a field.

$\mathbb{Q}$: – – – – – field.

$\mathbb{Z}_p$ integers modulo a prime p:

ring, comm. ring,

field? → We have to check $\hat{I}'$.

whether any non-zero element has a multiplicative inverse.

Actually, it is a field.

We see that later.

So we have the (comm.) ring $\mathbb{Z}_N$ of integers modulo N def'd similarly.

$$\begin{cases} \text{Googol} = 100^{100} \\ \text{Googolplex} = 100^{\text{Googol}} \end{cases}$$

16.4.07 (8)

back to ciphers:

we had Cesar : $\mathbb{Z}_{26} \to \mathbb{Z}_{26}$,
$a \mapsto a+3$

generalized Cesar: $C_i : \mathbb{Z}_{26} \to \mathbb{Z}_{26}$,
$a \mapsto a+i$

affine Codes: $A_{\alpha,i} : \mathbb{Z}_{26} \to \mathbb{Z}_{26}$,
$a \mapsto \alpha a + i$

for $\alpha \in \mathbb{Z}_{26}^{\times}$, $i \in \mathbb{Z}_{26}$.

we require that α has an inverse!

Try to decrypt

$b = A_{\alpha,i}(a) = \alpha \underset{\text{wanted!}}{a} + i$

then $\underbrace{\alpha^{-1}}_{?} \cdot (b - i) = a$

Problem: The inverse α^{-1} of $\alpha \in \mathbb{Z}_{26}$ does not always exist even if we require $\alpha \neq 0$.

Eg: $2 \cdot b = 1$ in $\mathbb{Z}_{26}$ has no solution.

Proof: Assume b exists.
$2 \cdot b = (\underbrace{\underbrace{\underset{\text{even}}{2 \cdot b}}_{\text{even}} \operatorname{rem} \underset{\text{even}}{26}}_{\text{even}})_{\mathbb{Z}_{26}}$

But 1 is not even! ↯ contradiction. □

Similarly, 13 has no inverse.
Actually, 21 has an inverse → 5!

still ENCRYPTION

18.4.07

(1)

had seen

Cesar : $\mathbb{Z}_{26} \to \mathbb{Z}_{26}$, $a \mapsto a+3$

gen. Cesar ciphers : C_i : $\mathbb{Z}_{26} \to \mathbb{Z}_{26}$, $a \mapsto a+i$

affine ciphers : $A_{\alpha,i}$: $\mathbb{Z}_{26} \to \mathbb{Z}_{26}$, $a \mapsto \alpha a + i$

where $\alpha \in \mathbb{Z}_{26}^{\times}$

ie. α shall be invertible wrt. multiplication in $\mathbb{Z}_{26}$.

attacks: (1) Brute force

Try all keys.

Feasible for all the above ciphers because they have only

1, 26, $12 \cdot 26$

different keys, resp., which is quite small.

There are at most 12 numbers invertible in $\mathbb{Z}_{26}$:

$\pm 1 \cdot \pm 1 = 1$
$\pm 3 \cdot \pm 9 = 1$
$\pm 5 \cdot \mp 5 = 1$
$\pm 7 \cdot \mp 11 = 1$
$\pm 9 \cdot \pm 3 = 1$
$\pm 11 \cdot \mp 7 = 1$

The others cannot be invertible because they are either even or divisible by 13.

18.4.07
(2)

permutation cipher

Fix any permutation $\pi : \mathbb{Z}_{26} \to \mathbb{Z}_{26}$
and replace letters accordingly.

How many such maps are there?

[permutation:
a map which is
- injective, ie. (into)
  $\cancel{\forall}$ $\pi(a) = \pi(b) \Rightarrow a = b$,
  or
  $a \neq b \Rightarrow \pi(a) \neq \pi(b)$,
  and
- surjective, ie. (onto)
  $\forall x \quad \exists a : \quad \pi(a) = x$.

bijective

In our case, any one of the properties implies the other because the sets $\mathbb{Z}_{26}$ and $\mathbb{Z}_{26}$ are both finite and of same size.]

There 26! such permutations of $\mathbb{Z}_{26}$.

This is very huge number.

$26! > 10^{26}$.

$\cancel{<}\ 10^{27}$

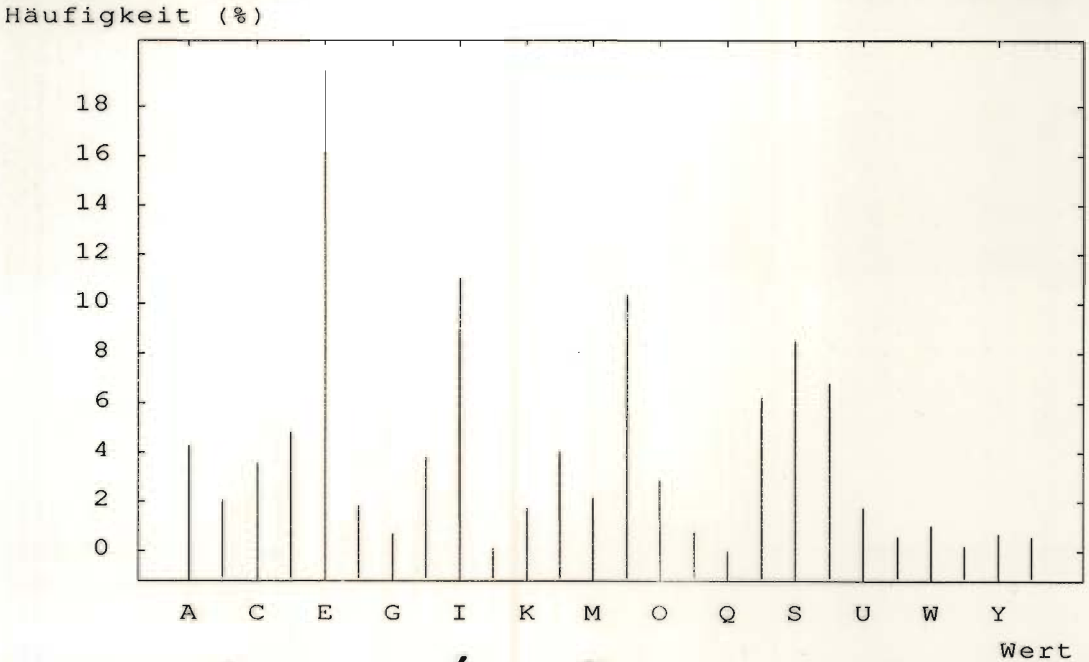
This much too far of brute force attack.

18.4.07
(3)

Every letter is still mapped to the same character.

So we can analyze the frequencies in the cipher text.

ASCII-Histogramm von <startbeispiel-de.txt> (869 Zeichen)



So most frequent letters are identifiable.

All the ciphers so far are

<u>substitution</u> ciphers

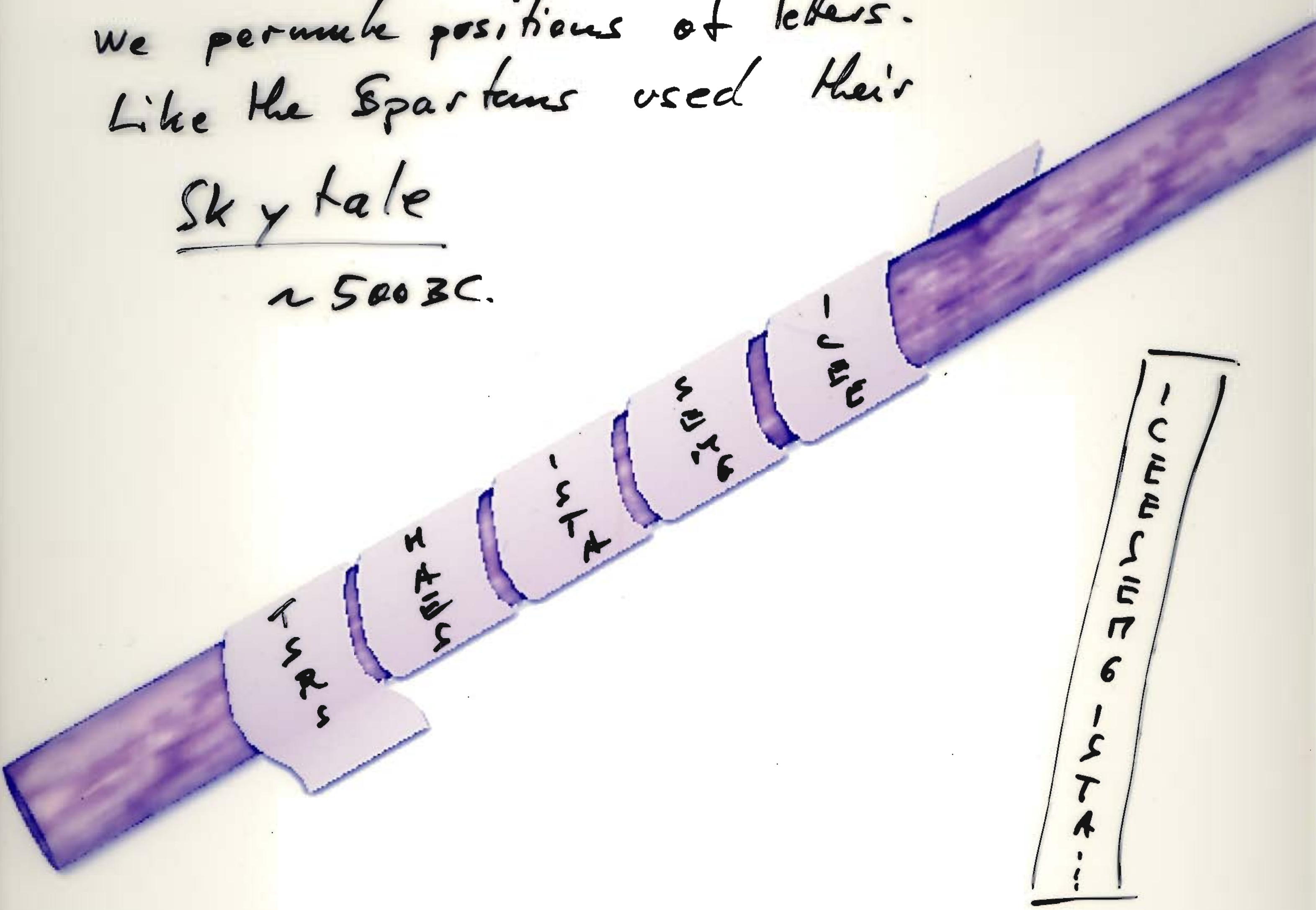
Another class are

<u>permutation</u> ciphers

we permute positions of letters.
Like the Spartans used their

<u>Skytale</u>

~ 500 BC.

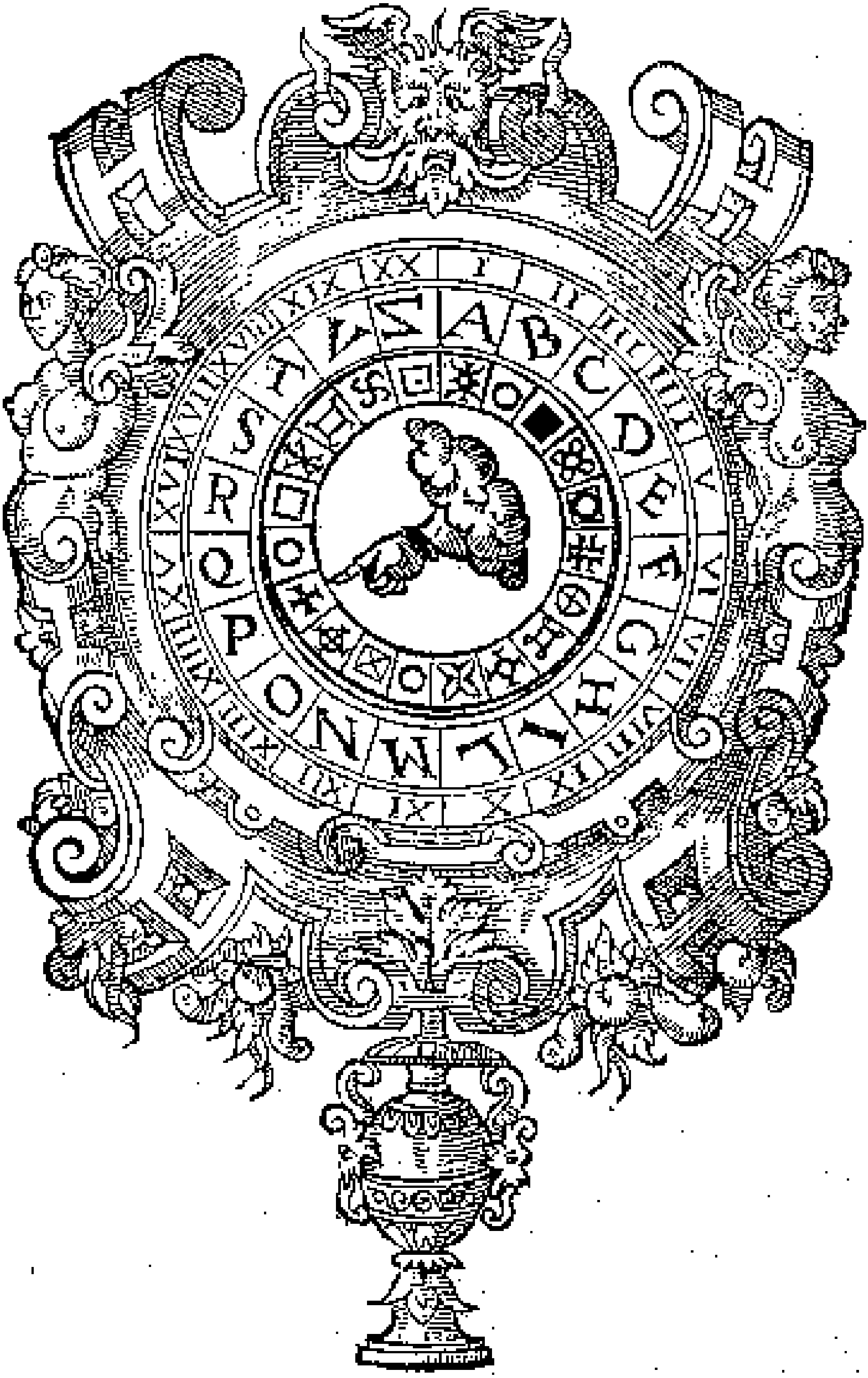


If stick is not entirely used
→ see group size.

BAD USAGE

- Brute force attack
  → try all stick sizes.

- Consider pairs of letters to find
  probable 'distances', 'group sizes', ...



One of Giovanni Battista Porta's cipher disks

18.4.07

5

Better ciphers?

Vigenère

THIS IS SECURITY

+

CARE CA RE CARECA

= =

V H ...

· Read each letter as a number in $\mathbb{Z}_{26}$ and add the key.

Brute force attack: 26^{ℓ} keys of length ℓ.

If ℓ is large enough this is not feasible.

But: there is a way to determine the key length!

Ex cryptool

After that we can do frequency analysis (or brute force on the generalized Cesar keys).

Better?

→ Use a key as long as the message.

But still: the key may have structure. → This can be used.

→ Use a random key!

18.4.07
6

One-Time-Pad

Given a plain text

$$p \in \{0,1\}^\ell$$

and a key

$$k \in \{0,1\}^\ell$$

the cipher text is

$$c \in \{0,1\}^\ell$$

given by $\quad c_i = p_i \oplus k_i$

This is completely secure!

$$\mathrm{Prob}(\text{plaintext} = p \mid \text{ciphertext} = c)$$

$$= \mathrm{prob}(P = p = c \oplus k \mid C = c \quad)$$

$$= \frac{\mathrm{prob}(K = k = p \oplus c \;, \cancel{C = c}\;)}{\cancel{\mathrm{prob}(C = c)}}$$

$$\doteq \ldots = \quad \mathrm{prob}(K = k) = 2^{-\ell}.$$

⇝ <u>Theorem</u> This is completely secure.

Problem?

18.4.07 ⑦

Bad usage: Using twice the same is bad:

$$\left.\begin{array}{l} c_1 = p_1 \oplus k \\ \\ c_2 = p_2 \oplus k \end{array}\right\} \quad c_1 \oplus c_2 = p_1 \oplus p_2$$

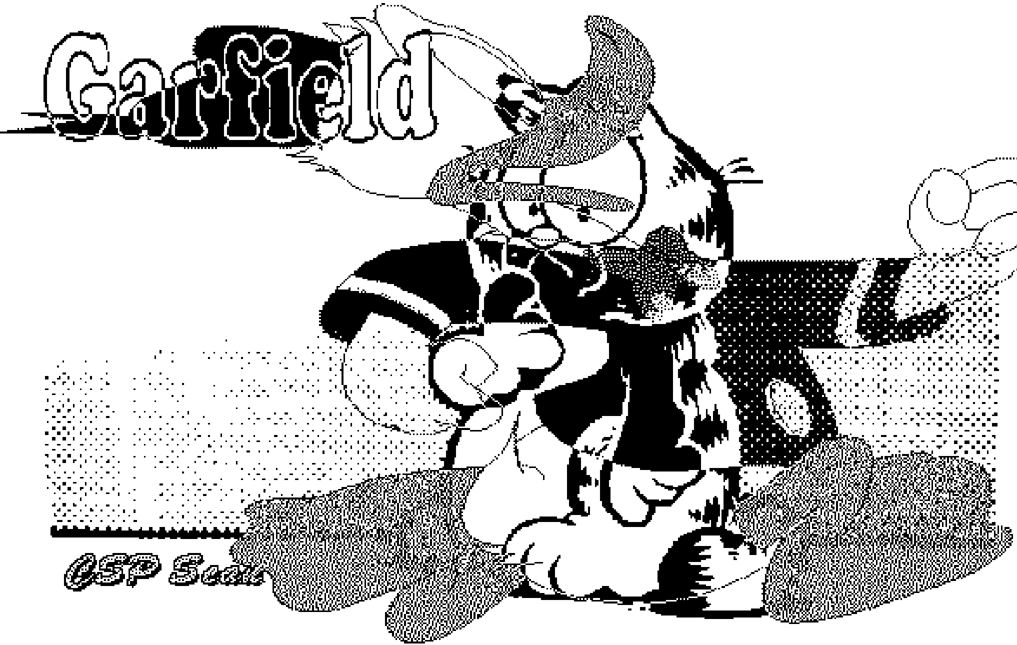
Even without redundancy in p_1, p_2 this reveals half the information about them. (ℓ out of 2ℓ bits are revealed.)

Even with good usage:

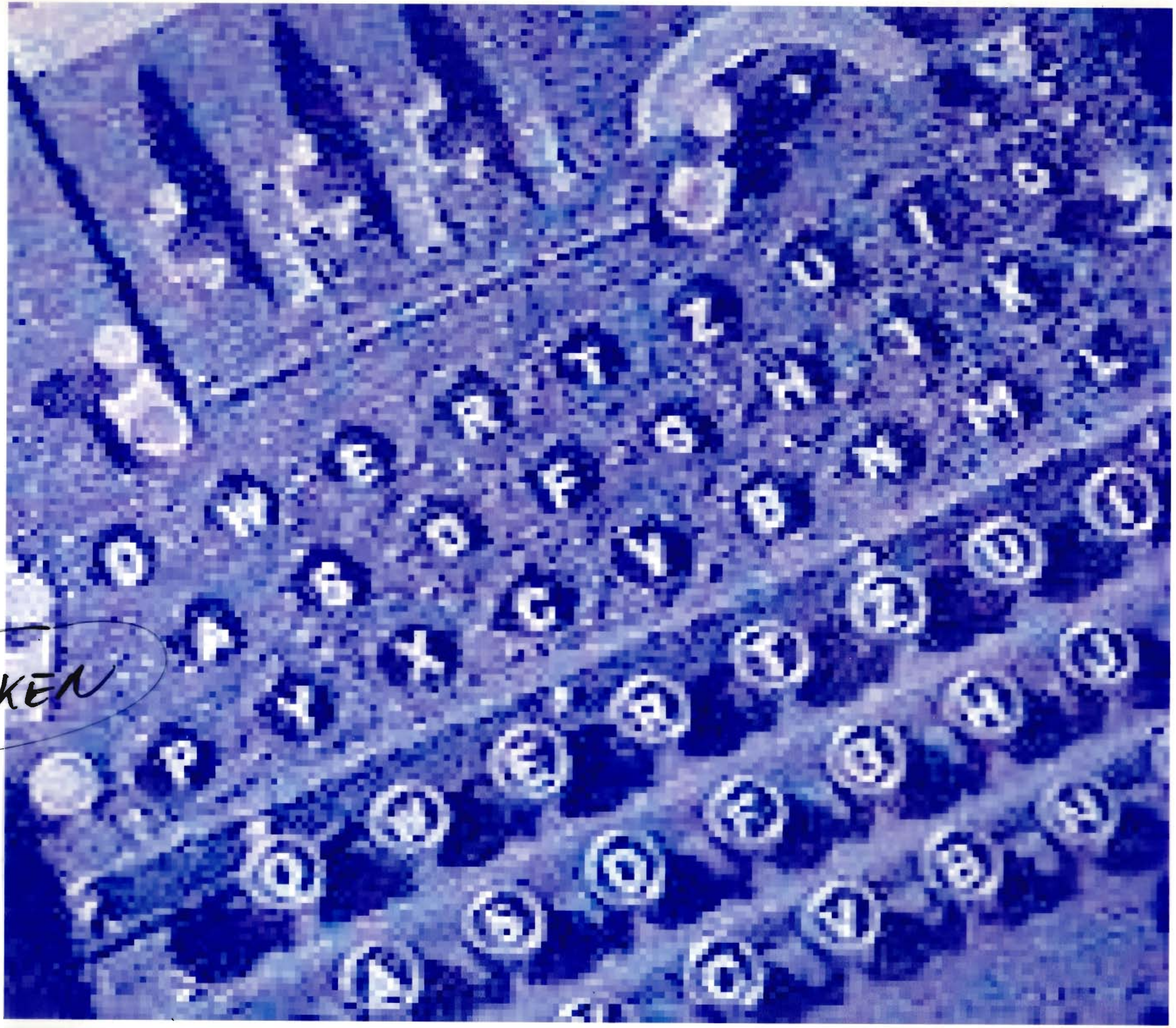
Problem?

-> hard to generate so much random data

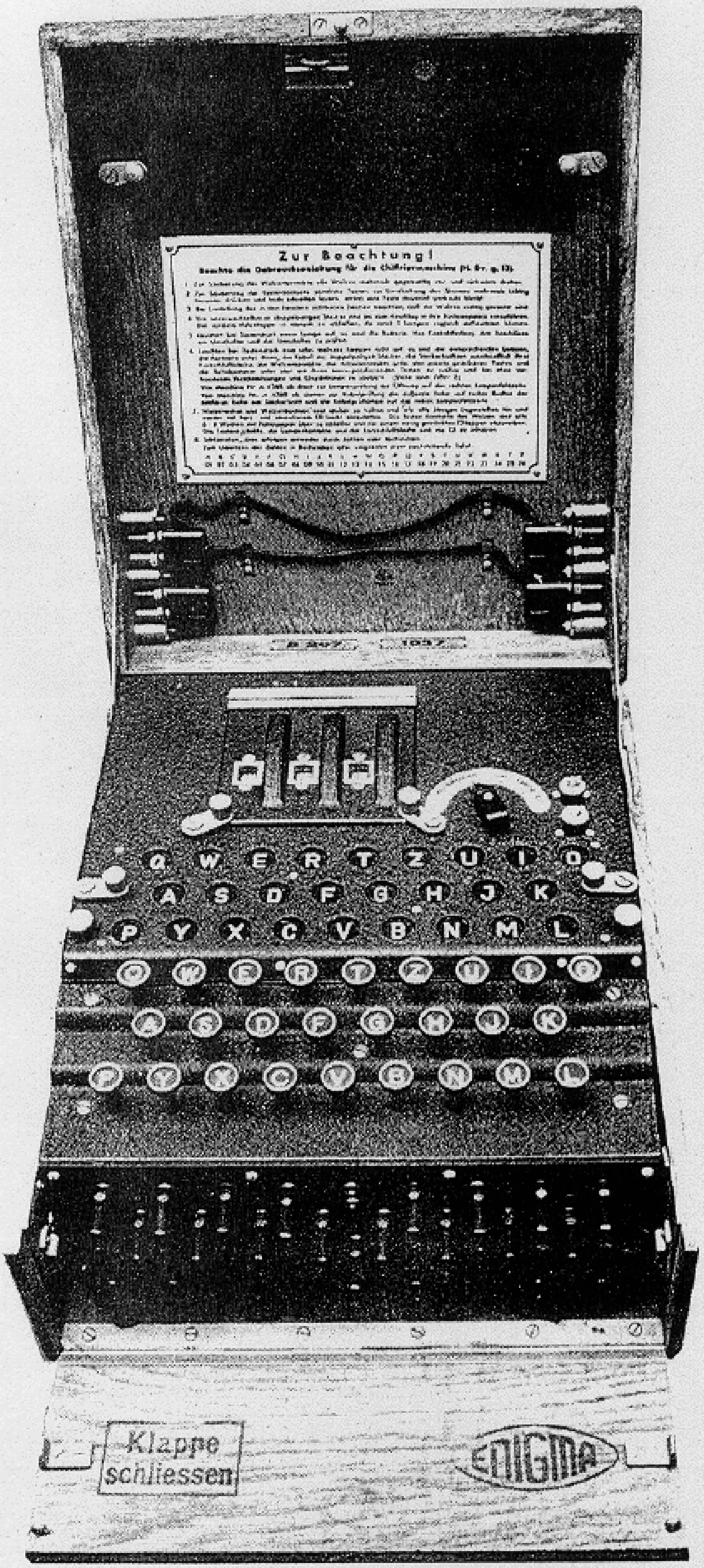
TOO LONG KEYS.


Garfield
CSP

ENIGMA



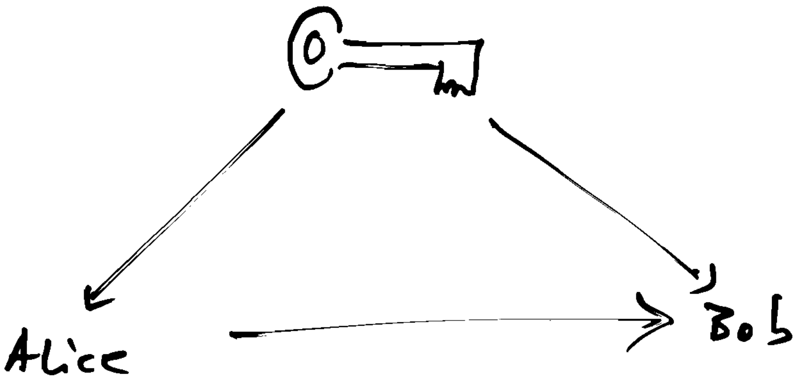
BROKEN


Zur Beachtung!
Klappe
schliessen
ENIGMA

The Advanced Encryption Standard

23.4.07 (1)

Cesar, ..., Enigma, ... One Time Pad:



AES, too. But ...

Suppose you calculate in $\mathbb{Z}_{256}$.
Is this a field?

Q: If $x \cdot y = 0$ in a field, is it possible that both $x \neq 0$ and $y \neq 0$?

Then of course $y = \underset{\text{ok, since } x \neq 0.}{x^{-1}} \cdot xy = x^{-1} \cdot 0 = 0$, but $y \neq 0$. ↯. NO!

<u>Fact</u> In any field, we have

$$xy = 0 \implies x = 0 \vee y = 0. \quad \square$$

In $\mathbb{Z}_{256}$ we have $\underset{\neq 0}{2} \cdot \underset{\neq 0}{128} = 0$

so this is <u>not</u> a field.

AES Advanced Encryption Standard

Designed as Rijndael by JOAN DAEMEN and VINCENT RIJMEN

universität bonn
Rheinische Friedrich-Wilhelms-Universität Bonn

cosec b-it computer security
Bonn-Aachen International Center for Information Technology

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.

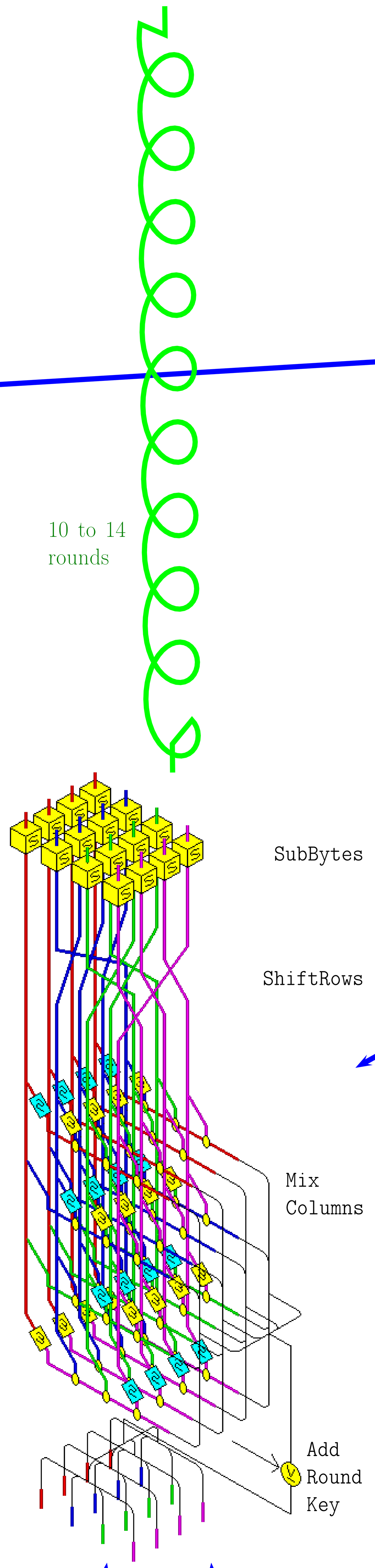
Addition: XOR, $(a + b)_i = a_i + b_i$.

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

Example $\texttt{57} \cdot \texttt{83} = \texttt{C1}$:

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

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



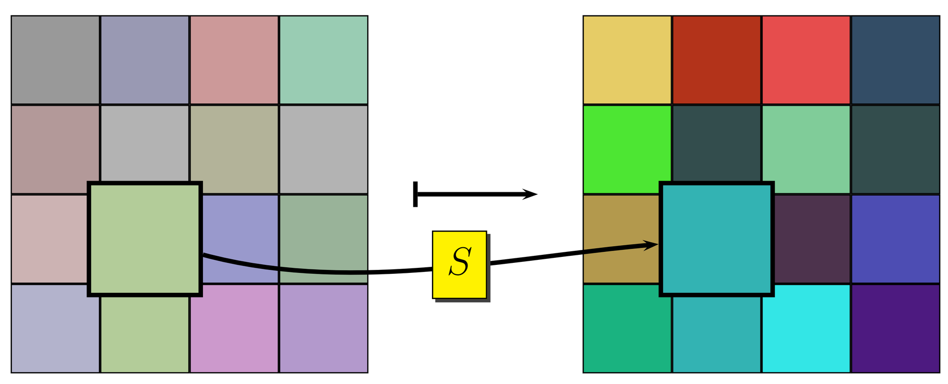
The S-box

$$\boxed{S}: \begin{array}{ccccc} \mathbb{F}_{2^8} & \longrightarrow & \mathbb{F}_{2^8} & \longrightarrow & \mathbb{F}_{2^8}, \\ y & \longmapsto & y^{-1} \hat{=} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{bmatrix} & \longmapsto & \begin{bmatrix} 1&0&0&0&1&1&1&1 \\ 1&1&0&0&0&1&1&1 \\ 1&1&1&0&0&0&1&1 \\ 1&1&1&1&0&0&0&1 \\ 1&1&1&1&1&0&0&0 \\ 0&1&1&1&1&1&0&0 \\ 0&0&1&1&1&1&1&0 \\ 0&0&0&1&1&1&1&1 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \end{array}$$

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

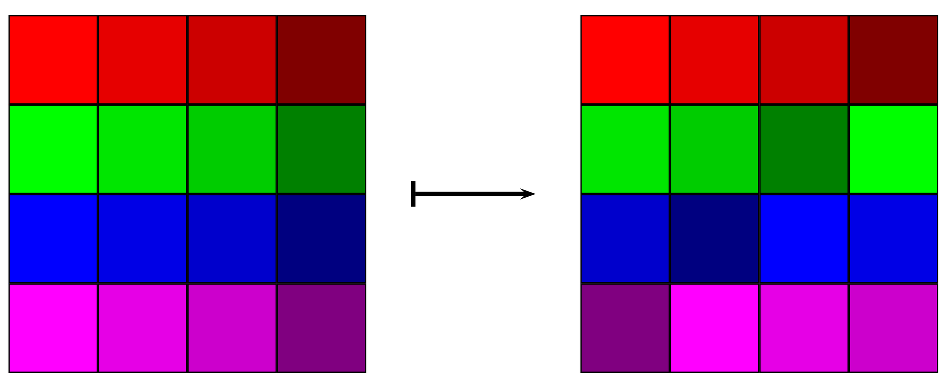
Simple implementation using a 256 byte lookup table.

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.

Nonlinear part of the key schedule

$$\boxed{R_i}: \begin{array}{ccc} (\mathbb{F}_{2^8})^4 & \longrightarrow & (\mathbb{F}_{2^8})^4, \\ \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} & \longmapsto & \begin{bmatrix} \boxed{S}(b) + x^{i-1} \\ \boxed{S}(c) \\ \boxed{S}(d) \\ \boxed{S}(a) \end{bmatrix} \end{array}$$

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

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 $a_i \in \mathbb{F}_{2^8}$.

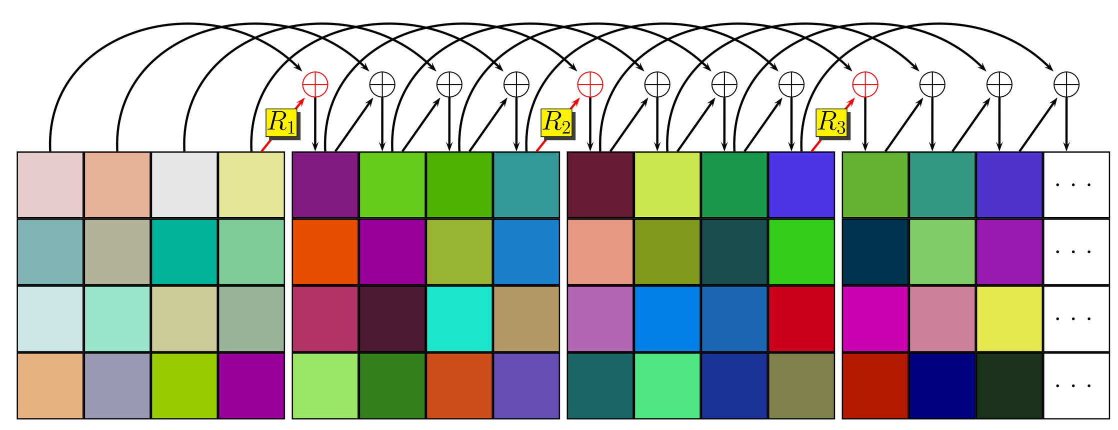
Addition: coefficient-wise $(a + b)_i = a_i + b_i$, XOR.

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

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

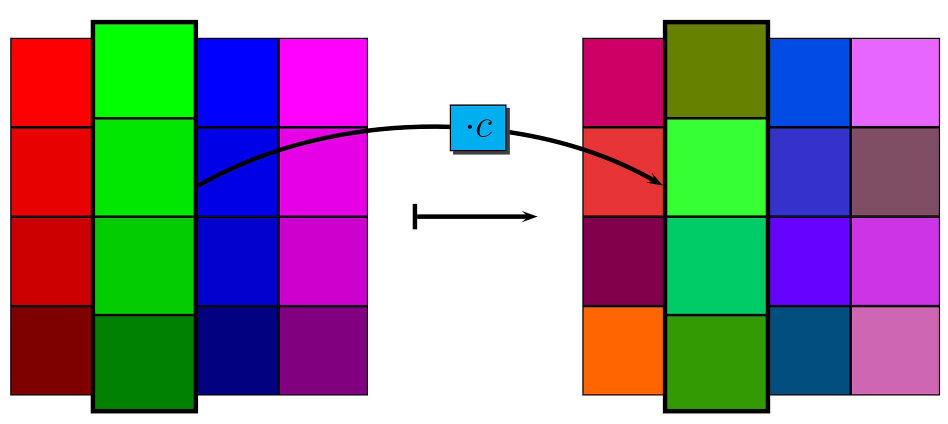
Not a field: $(z + 1)^4 = 0$.

The Key Schedule



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

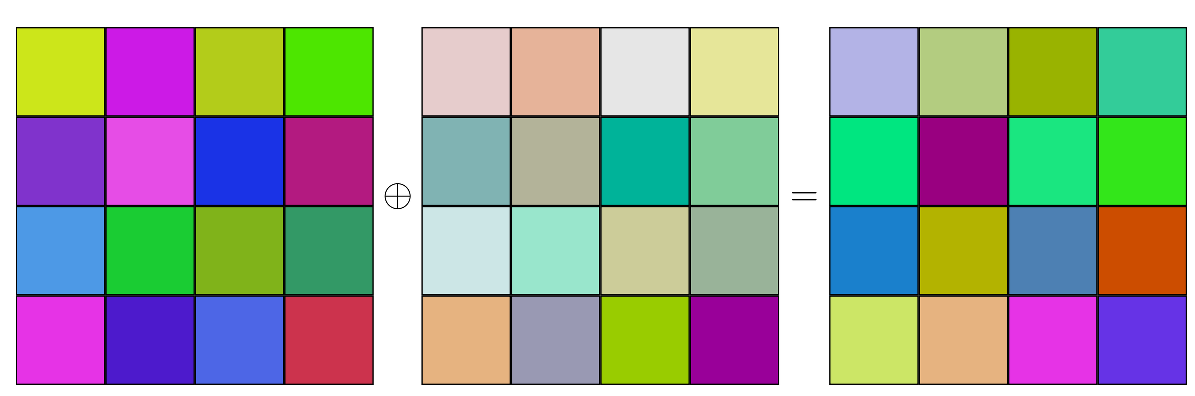
The MixColumns operation



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

Inverse: Multiply with $d = \texttt{0E} + \texttt{09}z + \texttt{0D}z^2 + \texttt{0B}z^3$.

The AddRoundKey operation



Simple XOR with the round key.

©2005 Michael Nüsken, cosec, b-it, Bonn

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

$\mathbb{F}_{2^8} \ni a = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5 + a_6 x^6 + a_7 x^7$, 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 $\mathtt{57} \cdot \mathtt{83} = \mathtt{C1}$**:**

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

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

23.4.07 ②

$a = 1 + x^2 \triangleq 1010\ 0000$

$b = x + x^2 \triangleq 0110\ 0000$

$\boxed{XOR}$

$a+b = 1 + x + \underbrace{(1+1)}_{=0} x^2 \triangleq 1100\ 0000$

$c := a \cdot b = (1+x^2)\cdot(x+x^2)$

$= 1\cdot x + 1\cdot x^2 + x^2 \cdot x + x^2 \cdot x^2$

$= x + x^2 + x^3 + x^4 \triangleq 0111\ 1000$

$c \cdot c = x^2 + x^4 + x^6 + x^8 \ ?\triangleq 00101010\,1$? ↯.

$\equiv -1 + -x + x^2 + -x^3 + 0\cdot x^4 + x^6 + 0 \cdot x^8$

$= 1 + x + x^2 + x^3 + x^6$

$\triangleq 1111\ 0010$

Point 1: If we reduce modulo $x^8+1 = (x^4+1)\cdot(x^4+1)$

then we obtain not a field.

Because the $\underset{\neq 0}{(x^4+1)} \cdot \underset{\neq 0}{(x^4+1)} = 0 \Rightarrow$ No field.

But I claim that

$p = x^8 + x^4 + x^3 + x + 1$

cannot be written as a product!

If we have $p = p' \cdot a_2$

and $a = a_1 \cdot a_2$

then $\underset{\neq 0}{p'} \cdot \underset{\neq 0}{a} = a_1 \cdot (p' a_2) = a_1 \cdot p = 0.$

The S-box

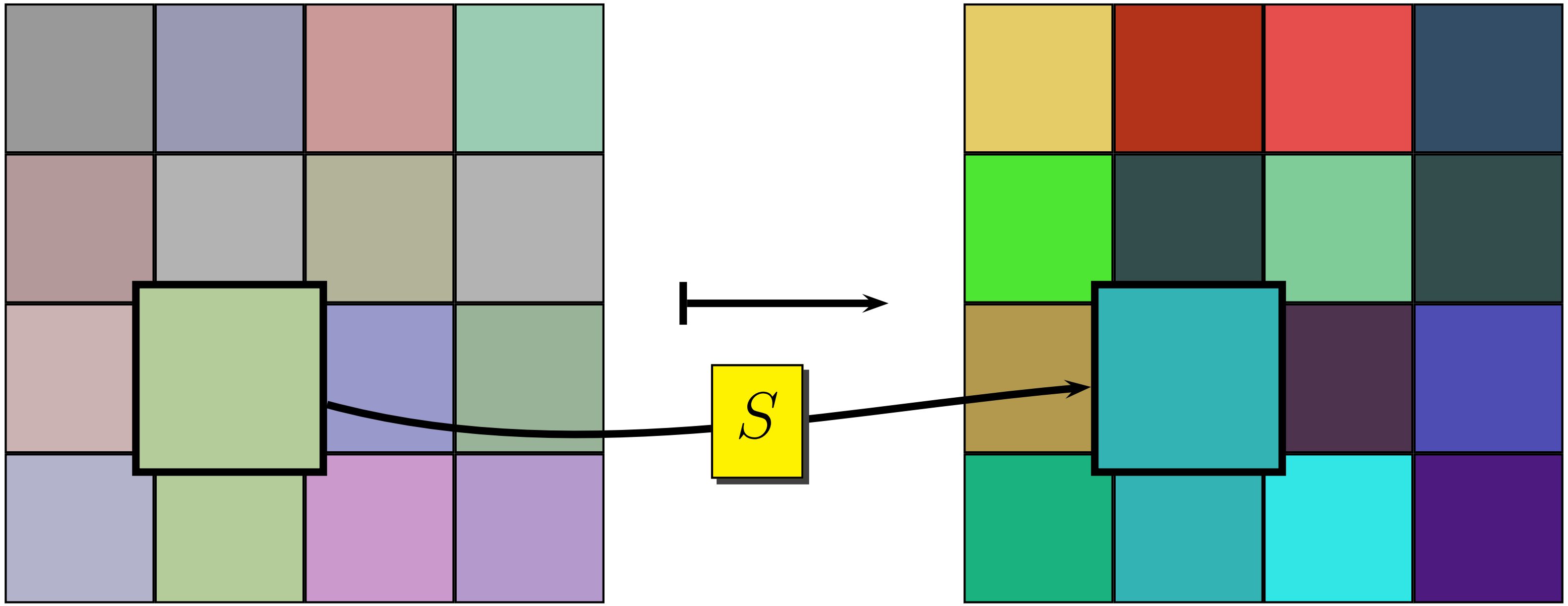
$$
\begin{array}{rcccl}
 & \mathbb{F}_{2^8} & \longrightarrow & \mathbb{F}_{2^8} & \longrightarrow \quad \mathbb{F}_{2^8}, \\
\boxed{S}: & y & \longmapsto & y^{-1} \hat{=} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{bmatrix} & \longmapsto \begin{bmatrix} 1&0&0&0&1&1&1&1 \\ 1&1&0&0&0&1&1&1 \\ 1&1&1&0&0&0&1&1 \\ 1&1&1&1&0&0&0&1 \\ 1&1&1&1&1&0&0&0 \\ 0&1&1&1&1&1&0&0 \\ 0&0&1&1&1&1&1&0 \\ 0&0&0&1&1&1&1&1 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}
\end{array}
$$

Highly nonlinear:

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

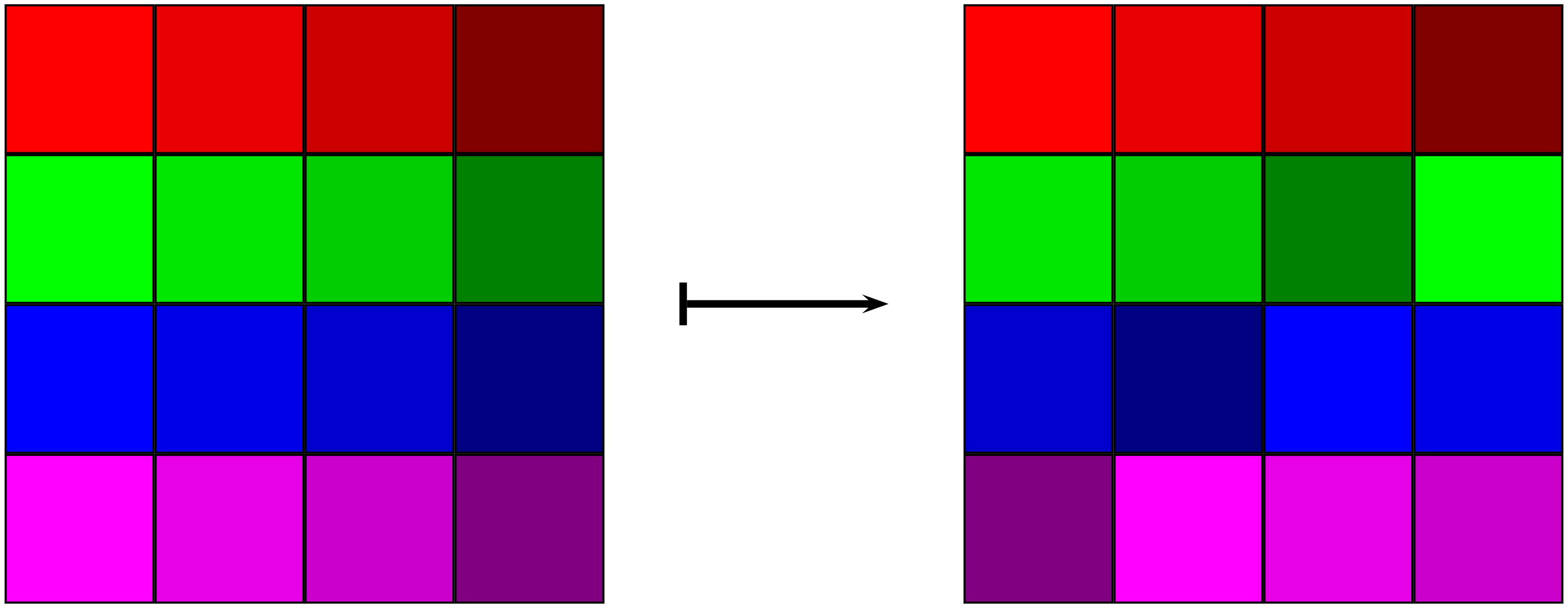
Simple implementation using a 256 byte lookup table.

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

$R = \mathbb{F}_{2^8}[z]/(z^4 + 1) \ni a_0 + a_1 z + a_2 z^2 + a_3 z^3$,
where $a_i \in \mathbb{F}_{2^8}$.

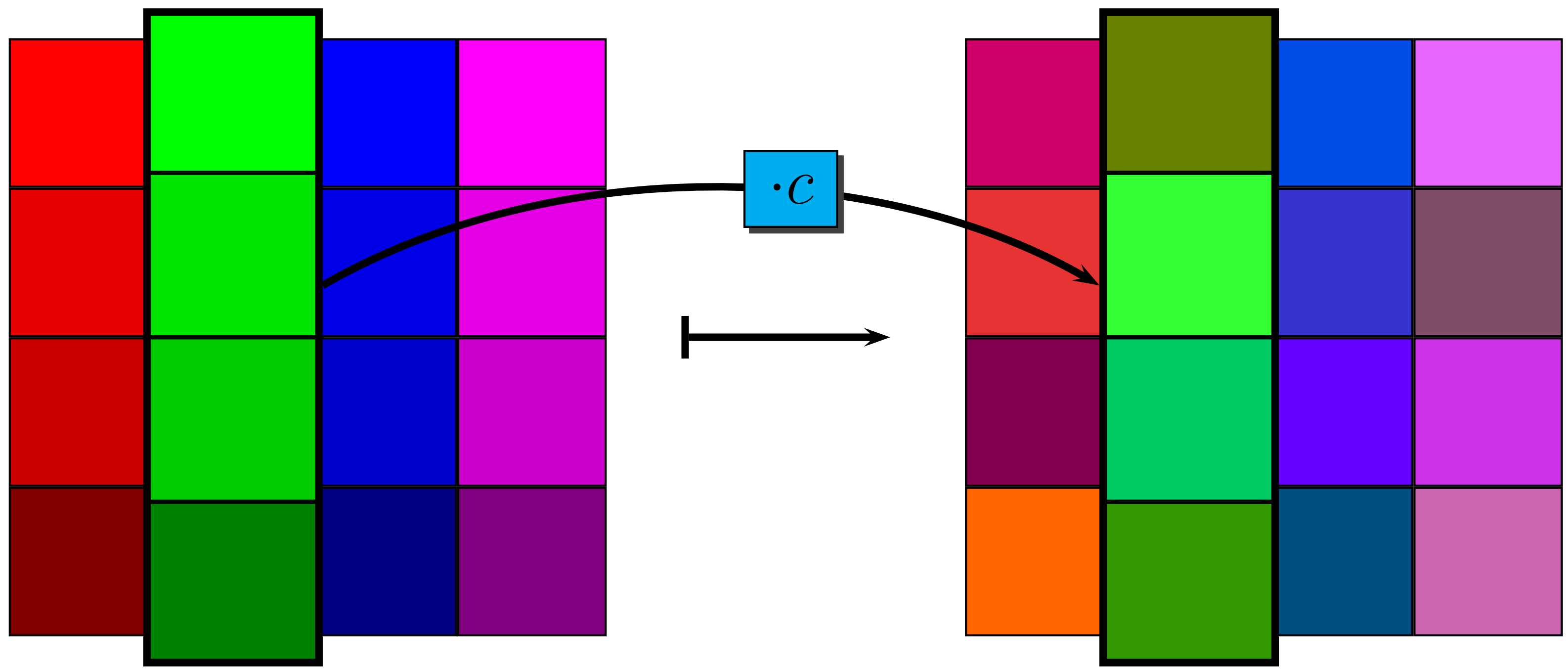
Addition: coefficient-wise $(a + b)_i = a_i + b_i$, XOR.

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: $(z + 1)^4 = 0$.

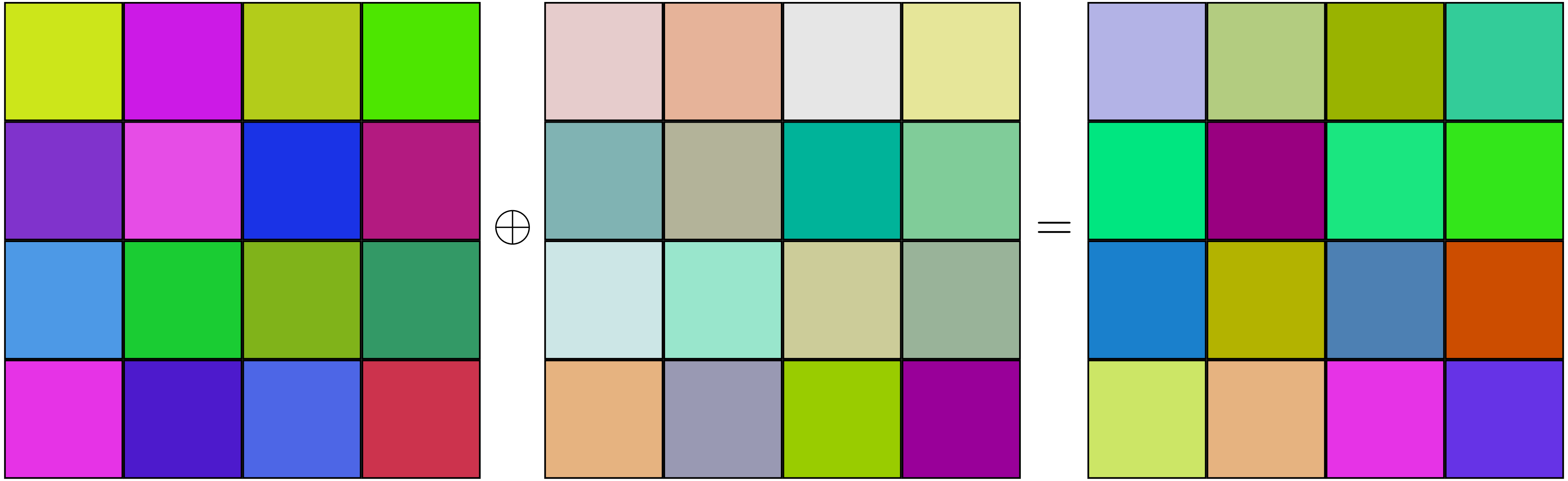
The MixColumns operation



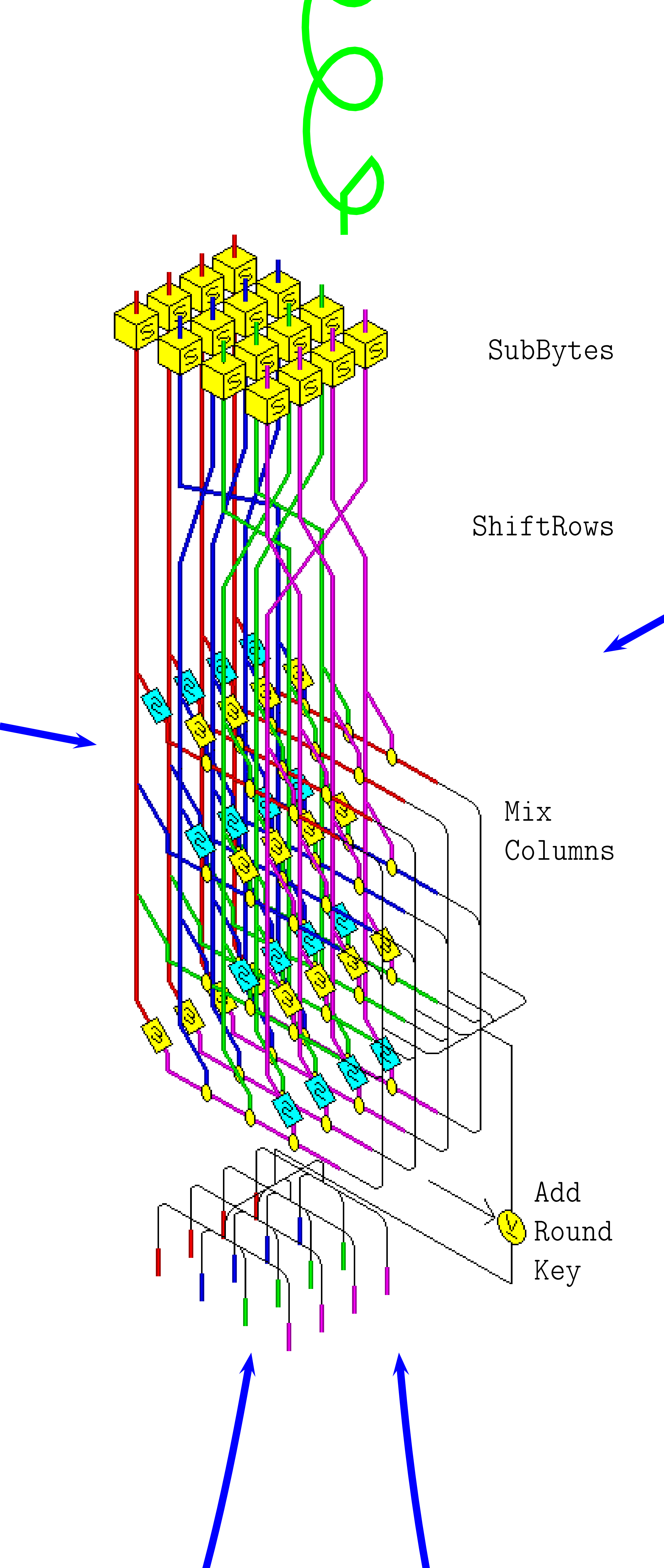
Each column is considered as a polynomial and multiplied by $c = \texttt{02} + \texttt{01}z + \texttt{01}z^2 + \texttt{03}z^3$.

Inverse: Multiply with $d = \texttt{0E} + \texttt{09}z + \texttt{0D}z^2 + \texttt{0B}z^3$.

The AddRoundKey operation



Simple XOR with the round key.


SubBytes
ShiftRows
Mix
Columns
Add
Round
Key

Security of One-Time Pad

What happens?

plaintext: $p \in \{0,1\}^n$ string of n bits

How are they distributed?

Somehow! So: for any $p \in \{0,1\}^n$

$$\text{prob}(\omega \mid \underset{\substack{\uparrow \\ \text{random variable}}}{P(\omega)} = \underset{\substack{\uparrow \\ \text{specific message}}}{p}) = \pi_p \in [0,1]$$

is given such that $\sum_p \pi_p = 1$

key: $k \in \{0,1\}^n$ string of n bits

How are they distributed?

For any $k \in \{0,1\}^n$ we have:

$$\text{prob}(\omega \mid \underset{\substack{\uparrow \\ \text{random variable}}}{K(\omega)} = k) = 2^{-n}.$$

And

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

in other words:

the r.v. P and K are independent.

Laxly spoken: we choose the key independently of the plaintext.

ciphertext: $c \in \{0,1\}^n$ bit string of length n.

Let $C(\omega) = P(\omega) \oplus K(\omega)$.

What is ~~K~~

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

Example: $\text{prob}(P = 0\ldots0) = 1$

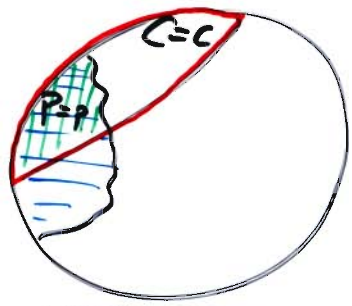
then Eve can easily guess the correct plain text.

But does the cipher text in that guess? No.

<u>Theorem</u> For any plaintext $p \in \{0,1\}^n$, and any ciphertext $c \in \{0,1\}^n$ we have

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

In other words: the cipher text does not help Eve at all.



<u>Pf</u>

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

$$P = p \wedge C = c$$
$$\Updownarrow$$
$$P = p \wedge C = c \wedge K = c \oplus p$$
$$\Updownarrow$$
$$P = p \wedge \underline{C = c} \wedge K = c \oplus p$$
$$\Updownarrow$$
$$P = p \wedge K = c \oplus p$$

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

$$= \sum_{p \in \{0,1\}^n} \text{prob}(P = p \wedge \underbrace{P}_{=p} \oplus K = c)$$

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

25.4.07
(3)

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

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

So,

$$\text{prob}(P = p \mid C = c)$$
$$= \text{prob}(P=p) \cdot \underbrace{\frac{\text{prob}(K = c \oplus p)}{\text{prob}(C = c)}}_{1}$$
$$= \text{prob}(P=p).$$

$\square$

That's best of all we can hope for:

Eve does not learn anything from the cipher text.

Calculating and deciding inverses

25.4.07
(4)

First, let's summarize where we need this:

Suppose $N \in \mathbb{N}_{>1}$.

$\mathbb{Z}_N$ ring of integers modulo N:

elements: $\{0, 1, 2, \ldots, N-1\}$

operations:

$+ : (a,b) \longmapsto (a+b) \operatorname{rem} N,$

$\cdot : (a,b) \longmapsto (a \cdot b) \operatorname{rem} N.$

$- : a \longmapsto \begin{cases} 0 & \text{if } a=0, \\ N-a & \text{if } a \neq 0 \end{cases}$

$= (-a) \operatorname{rem} N.$

TODO: $?^{-1}$: $a \longmapsto \begin{cases} a^{-1} & \text{if exist} \\ \text{FAIL} & \text{otherwise} \end{cases}$

Axioms: DON'T PANIC PAN(C).

maybe: 'I'.

Suppose $N = p$ is prime. Then (as is to be proved)

$\mathbb{Z}_p$ is a field, which we call $\mathbb{F}_p$.

Consider polynomials with coefficients in $\mathbb{F}_p$.

(Think of $p=2$.) Suppose m is a polynomial of degree $n \geq 1$.

$\mathbb{F}_p[X]/\langle m \rangle$ ring of polynomials modulo m with coefficients in $\mathbb{F}_p$

elements: $a_0 + a_1 X + a_2 X^2 + \ldots + a_{n-1} X^{n-1}$

with $a_0, a_1, \ldots, a_{n-1} \in \mathbb{F}_p$.

operations:

$+ : (a,b) \longmapsto (a+b) \operatorname{rem} m$

$= a+b = (a_0+b_0) + (a_1+b_1)X + \ldots + (a_{n-1}+b_{n-1})X^{n-1}$

$\cdot : (a,b) \longmapsto (a \cdot b) \operatorname{rem} m.$

$- : a \longmapsto -a \operatorname{rem} m = -a.$

25.4.07
5

TODO: $?^{-1}: a \longmapsto \begin{cases} a^{-1} & \text{if exist} \\ \text{FAIL} \end{cases}$

Note that $\mathbb{F}_{2^8} = \mathbb{F}_{256} \cong \mathbb{F}_2[X] / \langle X^8 + X^4 + X^3 + X + 1 \rangle$ in AES, and this has no non-trivial factors.

Let's start with better known situation: integers mod N.

We are given $a \in \mathbb{Z}_N$.

Find $b \in \mathbb{Z}_N$ such that $b \cdot_{\mathbb{Z}_N} a = 1$ in $\mathbb{Z}_N$.

ie. $b \cdot_{\mathbb{Z}} a \operatorname{rem}_{\mathbb{Z}} N = 1$ in $\mathbb{Z}$

ie. $\exists t \in \mathbb{Z}$: $b \cdot_{\mathbb{Z}} a + t \cdot_{\mathbb{Z}} N = 1$ in $\mathbb{Z}$

Find $b, t \in \mathbb{Z}$ such that

$$\boxed{b} \cdot a + \boxed{t} \cdot N = 1 \quad \text{in } \mathbb{Z}$$

Let's try some examples:

$N = 42$, $a = 5$.

Our aim is to b, t such that $ba + tN$ is as small as possible (but positive).

Trivially: $b_0 = 1, t_0 = 0$: $b_0 a + t_0 N = a = 5$.

$b_1 = 0, t_1 = 1$: $b_1 a + t_1 N = N = 42$.

Find new equation: $b_2 a + t_2 N = 42 - 5 = 37$

And again: $37 - 5$

Or all this at once: $= 42 - 8 \cdot 5 = 2$.

25.4.07 ⑥

i	r_i	q_i	b_i	t_i	comment
0	5		1	0	$1\cdot 5 + 0\cdot 42 = 5$
1	42	0	0	1	$0\cdot 5 + 1\cdot 42 = 42$
2	5	8	1	0	$1\cdot 5 + 0\cdot 42 = 5$ $\quad \vert \cdot 8 \vert -$
3	2 $= 42 - 8\cdot 5$	$\boxed{2}$	-8 $= 0 - 8\cdot 1$	1 $= 1 - 8\cdot 0$	$-8\cdot 5 + 1\cdot 42 = 2$
4	1	2	17	-2	$17\cdot 5 + (-2)\cdot 42 = 1$
5	0		-42	5	$\boxed{-42\cdot 5 + 5\cdot 42 = 0}$

How often does 5 fit into 42?

$\to 42 = 8\cdot 5 + 2$

How often does 2 fit in 5?

$5 = 2\cdot 2 + 1$

Always do this extra step as a cross check

Thus we find

$$17 \cdot_{\mathbb{Z}} 5 +_{\mathbb{Z}} (-2) \cdot_{\mathbb{Z}} 42 =_{\mathbb{Z}} 1 \quad \text{in } \mathbb{Z}.$$

Thus

$$17 \cdot_{\mathbb{Z}_{42}} 5 \; \boxed{\cancel{+_{\mathbb{Z}_{42}} (-2) \cdot_{\mathbb{Z}_{42}} 0}} =_{\mathbb{Z}_{42}} 1 \quad \text{in } \mathbb{Z}_{42}$$

so

$$17 \cdot_{\mathbb{Z}_{42}} 5 =_{\mathbb{Z}_{42}} 1 \quad \text{in } \mathbb{Z}_{42}$$

Fact: Time for multiplying two n-bit integers is $O(n^2)$ by school method, $O(n^{\log_2 3})$ by Karatsuba, $O(n \log n (\log\log n)^2)$ by Strassen-Schönhage [BN!]. Same times for division with remainder.

xxxxy · yyyyy

25.1.07
⑦

EFFICIENT

Theorem

The above Extended Euclidean Algorithm needs $\mathcal{O}(n^3)$ operations.

Even $\mathcal{O}(n^2)$ is true.

(30.4.07

Another example

i	r_i	q_i	s_i	t_i
0	$95 = a$		1	0
1	$25 = b$	3	0	1
2	20	1	1	-3
3	5	4	-1	4
4	0		5	-19
			$\frac{25}{5}$	$-\frac{95}{5}$

$(95 = 1 \cdot a + 0 \cdot b)$

$(25 = 0 \cdot a + 1 \cdot b)$

STOP indicator

$0 = \frac{25}{5} \cdot 95 - \frac{95}{5} \cdot 25$

→ Use last a cross check: $0 = 5 \cdot 95 - 19 \cdot 25$

This line is always easy to check but most easily if the last non-zero r_i equals 1.

CORRECT

Lemma The EEA computes the greatest common divisor g and s, t such that

$$g = s \cdot a + t \cdot b$$

Indeed, if ℓ is the number of the line with last non-zero r_i then $g = r_\ell$.

Actually, in the algorithm we choose a quotient q_i suitably and then

$$r_{i+1} = r_{i-1} - q_i \cdot r_i$$

We do that until $r_{\ell+1} = 0$.

Then $\gcd(r_\ell, \underbrace{r_{\ell+1}}_{=0})$

$$= \gcd(r_\ell, 0) = r_\ell$$

Remind: 'The' greatest common divisor g of two elements a, b is an element g such that

(i) $g \mid a$ and $g \mid b$

(g divides a and g divides b)

(ii) $\forall h : h \mid a \wedge h \mid b \Rightarrow h \mid g$

$(h \leq g)$

$c \mid d$ iff $\exists c' : c \cdot c' = d$

Now, we can show

$$\gcd(r_{i+1}, r_i) = \gcd(r_i, r_{i-1})$$

Say h is a common divisor of r_i and r_{i-1}.

Then $r_{i+1} = r_{i-1} - q_i r_i = r'_{i-1} h - q_i r'_i h$

$= (r'_{i-1} - q_i r'_i) \cdot h$. Thus h divides r_{i+1}.

So h divides r_i and r_{i+1}.

Now, the other way round, say k is a common divisor of r_{i+1} and r_i:

30.4.07
③

then $r_{i-1} = r_{i+1} + q_i r_i$

is a multiple of k.

Thus k is a common divisor of r_i and r_{i-1}. □

By induction we have

$$\gcd(a,b) = \gcd(r_0, r_1) = \dots = \gcd(r_e, 0) = r_e.$$

Further, for any i we have

$$r_i = s_i \cdot a + t_i \cdot b$$

This is trivially true for $i=0$ and $i=1$ and for $i>1$ we have

$$\begin{array}{r|l} & r_{i-2} = s_{i-2} a + t_{i-2} b \\ \cdot(-q_{i-1}) & r_{i-1} = s_{i-1} a + t_{i-1} b \end{array}$$

$$\underbrace{r_{i-2} - q_{i-1} r_{i-1}}_{r_i} = \underbrace{(s_{i-2} - q_{i-1} s_{i-1})}_{s_i} a + \underbrace{(t_{i-2} - q_{i-2} t_{i-1})}_{t_i} b$$

In particular,

$$g = r_e = \underline{\underline{s_e}}\, a + \underline{\underline{t_e}}\, b$$

claimed values.

□

30.4.07
(4)

Speed?

Claim: $\ell \leq 2n = 2\,\#$ bits in a and b.

We choose q_i such that

$$r_{i+1} = r_{i-1} - q_i r_i$$

$$\underbrace{r_{i-1}} = q_i \underbrace{r_i} + r_{i+1}$$

$$|r_{i+1}| \leq |r_i|. \quad \text{(for integers)}.$$

It ~~is~~ easy to see that

$$|r_{i+1}| < \frac{1}{2} |r_{i-1}| \qquad \text{Ex}$$

Thus

$$\underline{|r_\ell|} < \frac{1}{2^{\lfloor \ell/2 \rfloor}} \max(|r_1|, |r_0|)$$

that implies that

$$\ell/2 \leq \underbrace{\lceil \log_2 \max(|r_1|, |r_0|) \rceil}_{n}$$

$$\ell \leq 2n.$$

Thus the number of lines is at most twice the number of bits in a, b.

And each steps costs at most $O(n^2)$.

In total we have $O(n^3)$ bit operations at most.

Actually, the bound is bad, one can prove that we need $O(n^2)$ bit operations.

30.9.07

5

Recall that for the EEA we only need

- a ring, incl. a test for equality
- a division with remainder, i.e. for any a, b with $b \neq 0$ there exists q, r such that

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

and $\nu(r) < \nu(b)$ or $r = 0$.

for some suitable measure ν.

<u>Example</u> $\mathbb{Z}$: $\nu(a) = |a|$.

$F[X]$ with F a field :
ring of polynomials

$$\nu(a) = \deg a$$

$$[\, \nu(0) = -\infty \,]$$

Division in $F[X]$, say $F = \mathbb{F}_2$.

$a = x^7 + x^3 + x^2 + 1$

$b = x^4 + x + 1$

$$x^7 + x^3 + x^2 + 1 = (x^3 + 1) \cdot b + (x^2 + x)$$

$$\begin{array}{l} x^7 + x^3 + x^2 + 1 \\ -(x^7 + x^4 + x^3) \\ \hline x^4 + x^2 + \cancel{x} + 1 \\ -(x^4 + x + 1) \\ \hline x^2 + x \end{array}$$

$\deg(\cdot/\cdot) < \deg(b)$: DONE!

30.4.07 ⑥

Let's do an EEA for

$a = x^8 + x^4 + x^3$

$b = x \cdot (x-1) = x^2 + x$

Predict $\gcd(a,b) = ?$

	votes
b	0
$x+1$	1
x	19
1	0

i	r_i	q_i	s_i	t_i
0	$x^8+x^4+x^3$		1	0
1	x^2+x	$x^6+x^5+x^4+x^3+x+1$	0	1
2	x	$x+1$	1	$x^6+x^5+x^4+x^3+x+1$
3	0		$x+1$	$x^7+x^3+x^2$

Check: true!

gcd: x

repr:

$$\boxed{x = 1 \cdot (x^8+x^4+x^3) + (x^6+x^5+x^4+x^3+x+1) \cdot (x^2+x)}$$

$$\begin{array}{l|l}
x^8+x^4+x^3 & x^6+x^5 \\
x^8+x^7 & +x^4+x^3 \\
\hline
x^7+x^4+x^3 & +x+1 \\
x^7+x^6 & \\
\hline
x^6+x^4+x^3 & \text{rem} \\
x^6+x^5 & x. \\
\hline
x^5+x^4+x^3 & \\
\hline
x^3 & \\
x^3+x^2 & \\
\hline
x^2 & \\
\hline
x &
\end{array}$$

EFFICIENT? Yes: each step reduces the degree by 1. Thus after degree many steps (+2) we are done.

30.4.07 (7)

We have been talking about the rings

$\mathbb{Z}_N$ integers modulo N

(operation: modulo N) $\mathbb{F}_q[X]$

$\mathbb{F}_q[X]/\langle m \rangle$ polynomials over $\mathbb{F}_q$ modulo m

(operation: like for polynomials but taking remainders modulo m)

What about in these rings?

In $\mathbb{Z}_N$ we had translated the task to find b such that $ab \equiv 1$ in $\mathbb{Z}_N$ to the task of finding b, t such that

(*) $$b \cdot a + t \cdot N = 1 \text{ in } \mathbb{Z}.$$

Such a solution can be found using the EEA, if it exists... ?

we know: if the $\gcd(a, N) = 1$ then the EEA finds b, t such that (*).

Otherwise, if $\underbrace{\gcd(a,N)}_{=: g} \neq 1$?

Then $a = a'g$, $N = N'g$

thus $ba + tN = (ba' + tN') \cdot g \overset{\text{Assuming (*)}}{=} 1$

we would have $g \mid 1$, thus g is trivial.

So $g = 1$. ↯. So (*) has no solution.

So $ab = 1$ in $\mathbb{Z}_N$ has no solution, i.e. $\nexists$ inverse.

30.4.07
(8)

<u>Thm</u> The EEA decides wether $a \in \mathbb{Z}_N$ has an inverse and in case it has, computes it.

Actually, a has an inverse ($a \in \mathbb{Z}_N^\times$) $\iff \gcd(a, N) = 1$.

Or, $\mathbb{Z}_N^\times = \{ a \mid \gcd(a,N) = 1 \}$ □

Same for polynomials!:

$$\left(\mathbb{F}_q[X]/\langle m\rangle\right)^\times = \{ a \mid \gcd(a, m) = 1 \}$$

and the EEA computes the inverse if it exists:

$$\underline{b} \cdot a + \underline{t} \cdot m = 1 \rightsquigarrow b = a^{-1} \text{ in } \mathbb{F}_q[X]/\langle m\rangle$$

2.5.07
(1)

Example $\mathbb{Z}_6[X] / \langle X^2 + X + 1 \rangle$

$b \cdot (X-1) = 1$?

i	r_i	q_i	s_i	t_i
0	X^2+X+1		1	0
1	$X-1$	$X+2$	0	1
0	3	?	1	$-X-2$

Tröööt!

What happened?

... ... ???

$$X^2+X+1 = (X-1)\cdot(X+2) + 3$$

$$\begin{array}{r} X^2+X+1 \\ X^2-X \\ \hline 2X+1 \\ 2X-2 \\ \hline 3 \end{array}$$

Answer: $\mathbb{Z}_6$ is not a field. ($2 \cdot 3 = 0$)

Thus there is no division with remainder for polynomials over $\mathbb{Z}_6$.

Thus EEA needs not work.

Check the conditions!

2.5.07
②

When is $\mathbb{Z}_N$ a field?

If N is not irreducible.

⌈ Two definitions:

p is <u>irreducible</u> iff whenever we write $p = a \cdot b$ then a or b is multiplicatively invertible, i.e. <u>trivial</u>.

in other words: if p cannot be written as a proper product.

$\forall a \cdot b: p = a \cdot b \Rightarrow a | 1 \vee b | 1.$

p is <u>prime</u> iff whenever p divides a product $a \cdot b$ then p divides one of the factors a, b.

$\forall a, b: p | a \cdot b \Rightarrow p | a \vee p | b$

Remark: p prime $\Rightarrow$ p irr.

<u>Example</u> where $\Leftarrow$ does not hold:

$$\mathbb{Z}[\sqrt{-5}] = \{ a + b\sqrt{-5} \mid a, b \in \mathbb{Z} \}.$$

$$(a + b\sqrt{-5})(a' + b'\sqrt{-5})$$
$$= aa' + (ba' + ab')\sqrt{-5} + bb' \cdot (-5)$$
$$= (aa' - 5bb') + (ba' + ab')\sqrt{-5}!$$

Now: $6 = 2 \cdot 3$
$= (1 + \sqrt{-5}) \cdot (1 - \sqrt{-5})$

Actually, $2, 3, 1 \pm \sqrt{-5}$ are all irreducible.

So: $2 | (1+\sqrt{-5})(1-\sqrt{-5})$ but $2 \nmid (1 \pm \sqrt{-5})$ ⌟

Back to our interests:

When is $\mathbb{Z}_N$ a field?

If N is not irreducible
then it is not a field.

Pf Write $N = N_1 \cdot N_2$ with N_1, N_2 both non-trivial
ie. $N_1 \neq \pm 1$, $N_2 \neq \pm 1$.

Then $(N_1 \bmod N) \cdot (N_2 \bmod N) = 0$ in $\mathbb{Z}_N$

and $N_1 \bmod N \neq 0$, (otherwise $N_1 = c \cdot N$, ie.
$N_2 \bmod N \neq 0$. $N_2 = \pm 1$ 8.)

□

Example: $N = 4$: $2 \cdot 2 = 0$ in $\mathbb{Z}_4$.
$N = 6$: $2 \cdot 3 = 0$ in $\mathbb{Z}_6$.
So these are no fields!

If $N = p$ is ~~pr~~ irreducible
then $\mathbb{Z}_N$ is a field.

Pf we need to show that all elements
but 0 have a multiplicative inverse.

We know $\mathbb{Z}_p^\times = \{ a \in \mathbb{Z}_p \mid \gcd(a, p) = 1 \}$

Now the divisors of p are $\pm 1, \pm p$
Consider $a \in \mathbb{Z}_p$, ie. $a \in \{0, 1, 2, \ldots, p-1\}$.
Then $\gcd(a, p) \neq \pm 1$, iff $a = 0$
(iff a is a multiple of p).

Thus $\mathbb{Z}_p^\times = \{1, 2, \ldots, p-1\} = \mathbb{Z}_p \setminus \{0\}$.

So we are done. $\square$

2.5.07

Concluding:

Theorem $\mathbb{Z}_N$ is a field iff N is irreducible (prime). $\square$

$\mathbb{F}_q[X]/\langle m \rangle$ polynomials modulo same polynomial m.

When is this a field?

Theorem $\mathbb{F}_q[X]/\langle m \rangle$ is a field iff m is irreducible.

Pf If m is reducible then write $m = m_1 \cdot m_2$ as a proper product.

$$m_1 \not\equiv 0, \quad m_2 \not\equiv 0$$

then $(m_1 \bmod m) \cdot (m_2 \bmod m) = 0$ in $\mathbb{F}_q[X]/\langle m \rangle$.

If m is irreducible then

$$\left(\mathbb{F}_q[X]/\langle m \rangle \right)^\times = \{ a \in \mathbb{F}_q[X]/\langle m \rangle \mid \gcd(a,m) = 1 \}$$

Now, if m has no proper factors then $a \neq 0$ is enough to ensure that $\gcd = 1$, so any element but 0 has an inverse. $\Rightarrow$ it's a field $\square$

Ex: $\mathbb{Z}$: 2 is irr. 12.5.07 (5)

$\rightsquigarrow$

$\mathbb{Z}_2$ a field $=: \mathbb{F}_2$

$\mathbb{F}_2[X]$: X^2+X+1 is irr.

$\rightsquigarrow$

$\mathbb{F}_2[X]/\langle X^2+X+1\rangle$ a field $=: \mathbb{F}_4$.

$= \{a_0 + a_1 X \mid a_0, a_1 \in \mathbb{F}_2\} = \{0, 1, X, X+1\}$

$\mathbb{F}_4[Y]$: Y^3+Y+1 is irr.

$\rightsquigarrow$

$\mathbb{F}_4[Y]/\langle Y^3+Y+1\rangle$ a field $=: \mathbb{F}_{4^3} = \mathbb{F}_{64}$.

$= \{b_0 + b_1 Y + b_2 Y^2 \mid b_0, b_1, b_2 \in \mathbb{F}_4\}$

AES: $\mathbb{F}_2[X]$: $X^8+X^4+X^3+X+1$ is irr.

$\rightsquigarrow$

$\mathbb{F}_{256} = \mathbb{F}_{2^8}$.

So: wonderful tool.

⌈ $\mathbb{F}_2[Z]$, $Z^6 + \ldots$ irr.

$\rightsquigarrow$

$\mathbb{F}_{64}'$ It turns out that this $\mathbb{F}_{64}' \cong \mathbb{F}_{64}$ from above. ⌋

8.5.07 ⑥

Additional information:

there exists a field with q elements iff q is a prime power and essentially one such.

a power of a prime e.g. 7^4.

Set of invertible numbers:

$$\mathbb{Z}_N^\times$$

$$\left(\mathbb{F}_q[X] / \langle m \rangle\right)^\times$$

EX

$$\mathbb{Z}_6^\times = \{1, 5\}$$

$$\underbrace{\left(\mathbb{F}_2[X] / \langle \underset{\underset{(X+1)^2}{\shortparallel}}{X^2+1} \rangle\right)}_{\underset{\{0, 1, X, X+1\}}{\shortparallel}}^\times = \{1, X\}.$$

Addition stay <u>not</u> inside:

in $\mathbb{Z}_6$: $\underset{\mathbb{Z}_6^\times}{\underset{\cap}{1}} + \underset{\mathbb{Z}_6^\times}{\underset{\cap}{1}} = \underset{\mathbb{Z}_6^\times}{\underset{\not\cap}{2}}$.

Multiplication? Works! PANIC!

2.5.07 ⑦

Whenever R is a ring, commutative, then the set ~~of~~ $R^\times$ of invertible elements is a

commutative group !
wrt. to multiplication !

Ex $\quad \mathbb{Z}_{15}$: $\quad \mathbb{Z}_{15}^\times = (\{\pm 1, \pm 2, \pm 4, \pm 7\}, \cdot)$
~~is~~ a comm. group!

$\mathbb{F}_{256}[Z]/(Z^4+1)$: $\left(\mathbb{F}_{256}[Z]/(Z^4+1) \right)^\times$
is a comm. group.

Def A comm. group is a set with one operation such that the axioms PANIC hold.

7.5.07

①

How to exchange a key without pre shared secret?

How to talk secretly even if Eve listens to everything <u>including</u> the description of the scheme?

Diffie & Hellman (1976) Key exchange:

<u>Setup</u>: a group: $\mathbb{Z}_p^\times$ p prime

$q \mid (p-1)$ prime

$g \in \mathbb{Z}_p^\times$ with good properties (related to q!)

Example: $\underbrace{\mathbb{Z}_{47}^\times}$ $q = 23$, $p = 47$.

it is a group of 46 elements.

$g = 2$: $G = \langle g \rangle$ group generated by g

$:= \{1, g, g^2, \dots, g^{22}, \dots g^{45}, \dots\}$

Alice (Cesar)	G, g	Bob (Cleopatra)
$x \in_R \mathbb{Z}$		$y \in_R \mathbb{Z}$
$h_A = g^x$	$h_A = g^x \rightarrow$ $\leftarrow h_B = g^y$	$h_B = g^y$
$k_A = h_B^x$		$k_B = h_A^y$

$= g^{xy} =$

17.5.07
(2)

Now

$$k_A = (g^y)^x = g^{y\cdot x} = g^{x\cdot y} = (g^x)^y = k_B ;$$

so Alice and Bob have a shared secret now. They can use it to encipher further messages.

Correctness? This is $k_A = k_B$. ✓

Efficiency? $O(n^2)$ (bit) operations per multiplication.

Auction: We sell 2^{28}.

Who does it cheapest?

First bid: 27 mult.

Dennis 15 mult: calc 2^{14}, square.

Tillmann 10 mult: calc 2^7, square, square.

Til 8 mult: — " —

$$(2),\ 2^2,\ 2^3,\ 2^6,\ 2^7,\ 2^{14},\ 2^{28}$$

1 2 3 4 5 6

6 multiplications!

7.5.07
3

We sell 2^{35}!

Sumit: 12 op's: calc 2^7, then raises this to the fifth power.

Tilman 7 op's:

$$2^2, 2^4, 2^5, 2^{10}, 2^{20}, 2^{30}, 2^{35}$$

Tillmann 6 op's!

$$2^2, 2^4, 2^8, 2^{16}, 2^{17}, 2^{34}, 2^{35}$$ 7!

Sumit 6 op's

$$2^2, 2^4, 2^8, 2^{16}, 2^{17}, 2^{33}, 2^{35}$$ 7!

Square & multiply (Repeated squaring)

Note: $35 = 100011_2$

Now compute:

2^{10_2}

$\downarrow$ square

2^{100_2}

$\downarrow$ square

2^{1000_2}

$\downarrow$ square & mult with 2

2^{10001_2}

$\downarrow$ square & mult with 2

2^{100011_2}

Try: g^{382}: Square & mult → 14 mult.
Same thinking → 12 mult.
Optimum: 11 mult.

7.5.07 (4)

Theorem Given a group G and an element $g \in G$ we can compute the map

$$\mathbb{Z} \longrightarrow G$$
$$e \longmapsto g^e$$

Efficiency

with $2\left(\underbrace{(\lfloor \log_2 e \rfloor + 1)}_{\#\text{bits for } e} - 1\right)$ group operations.

$2^{s-1} \leq e < 2^s$

$s-1 \leq \log_2 e < s$

□

~~to~~ calcul-

~~Proo~~ Proof? Implementation? $\longrightarrow$ Ex

SECURITY?

What does EVE see?

Setup: group G, generator g

Communication: $h_A = g^x$, $h_B = g^y$

Wants: common key: g^{xy}

DHP (Diffie-Hellman-Problem)

$$(g, g^x, g^y) \longmapsto g^{xy}.$$

for example with $G = \mathbb{Z}_{17}^\times$, $g = 2$ we might ask:

$g = 2$, $g^x = 3$, $g^y = 5$. What is g^{xy}?

It is enough to find x or y!

Because then, say we found x, we can compute $g^{xy} = (g^y)^x = 5^x$.

Consider the

DLP (Discrete Logarithm Problem)

$(g, g^k) \longmapsto k$.

What we have seen is:

If we can solve the DLP then solve the DHP.

So we must choose the setup, group G and the generator g, such that at least the DLP is difficult.

Necessary for the security:

DLP is difficult (in $G = \langle g \rangle$).

Good examples:

Use $g \in \mathbb{Z}_p^\times$ such that $g^q = 1$, $g \neq 1$ where q is a large prime (& $q \mid p-1$).

Interludium

Other groups, with particularly difficult DLP:

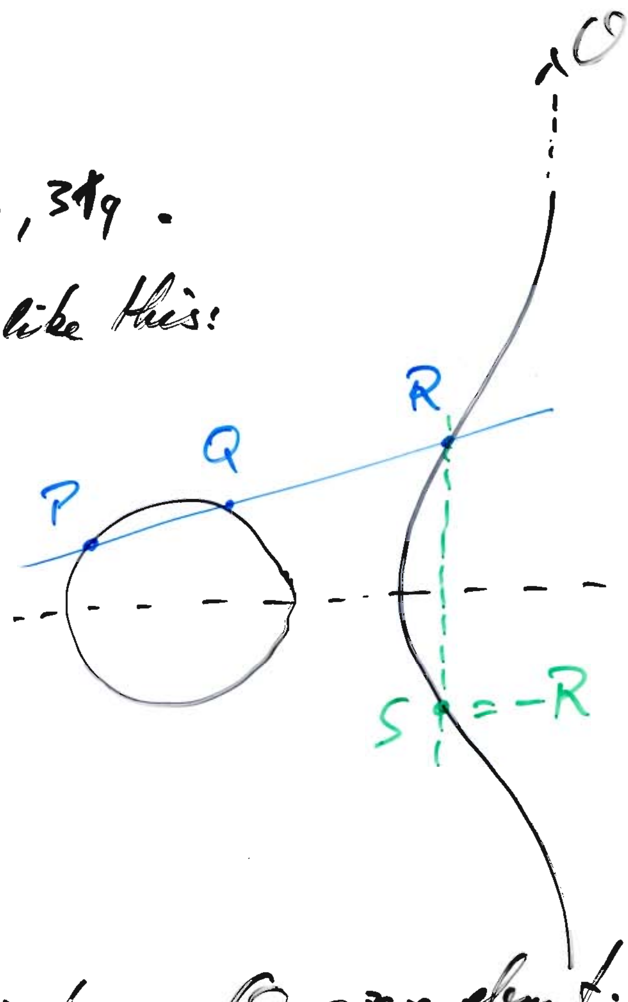
Elliptic curves

Given an equation

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

with $a, b \in \mathbb{F}_q$, $2 \nmid q, 3 \nmid q$.

Over $\mathbb{R}$ the picture is like this:



We require

$$P + Q + R = \mathcal{O}$$

so we should define

$$P + Q = -R = S$$

This defines a group & if we add one point: $\mathcal{O}$ = zero element.

Define:

$$P+Q = \begin{cases} S \text{ as above} & \text{if } P \neq Q,\ P \neq -Q,\ P \neq \mathcal{O},\ Q \neq \mathcal{O}, \\ Q & \text{if } P = \mathcal{O} \\ P & \text{if } Q = \mathcal{O} \\ \mathcal{O} & \text{if } Q = -P \end{cases}$$

Group?

P ✓
N by construction of $\mathcal{O}$. ✓
I mirror at x-axis ✓
C obvious ✓

A ? Difficult to see. But true!!

17.5.07

For these groups the

DLP

is supposedly 'more' difficult.

Thus we can use smaller versions (measured by q, say) to get same security.

Eg. using $\mathbb{Z}_p^\times$ with 1024-bit p

corresponds to E ~~over~~ $\mathbb{F}_p$ with 160-bit p.

In total: E might be cheaper at same security.

end interludium

7.5.07

8

How to find, say p, q, and g

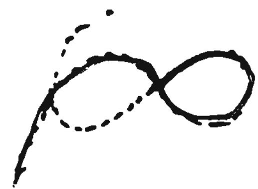
such that $g \in \mathbb{Z}_p^\times$, $\underline{\underline{g^q = 1}}$, $g \neq 1$.

Need to know more about exponentiation, powering:

Say we are given $g \in G$, G some group.

(Think $G = \mathbb{Z}_p^\times$, for example.)

Consider $g, g^2, g^3, g^4, g^5, g^6, \ldots$



<u>Ex</u> $p = 11$; $g = 2$.

$e \in \mathbb{Z}$	0	1	2	3	4	5	6	7	8	9	10	11
$2^e \in \mathbb{Z}_{11}^\times$	1	2	4	-3	5	-1	-2	-4	3	-5	1	2 ...

repeat this part!

DH Protocol

$G \ni g$

Alice		Bob
$x \in_R \mathbb{Z}$		$y \in_R \mathbb{Z}$
	$\xrightarrow{\;g^x\;}$	
	$\xleftarrow{\;g^y\;}$	
$(g^y)^x$	$= g^{xy} =$	$(g^x)^y$

Correctness ✓

Efficiency ✓ (Square & multiple)

Security

Eve has to solve the DHP:

$$(g, g^x, g^y) \longmapsto g^{xy}$$

at least with some probability.

[Here, "amplification" is possible! (Ex)
From a solution for $(g, g^{x+\delta}, g^{y+\varepsilon})$
we can derive g^{xy}, so try various δ, ε.]

If Eve can solve the DLP:

$$(g, g^x) \longmapsto x$$

with some probability

then she can solve the DHP.

["Amplification" possible.]

Beware of Eve becoming active: Mallory.

(Wo)Men in the middle attack:

Alice	Mallory (Wilma)		Bob
x	y'	x'	y
$\xrightarrow{g^x}$ $\xleftarrow{g^{y'}}$		$\xrightarrow{g^{x'}}$ $\xleftarrow{g^{y}}$	
$(g^{y'})^x$	$(g^x)^{y'}$	$(g^y)^{x'}$	$(g^{x'})^y$

$$(g^{y'})^x = g^{xy'} = (g^x)^{y'} \qquad (g^y)^{x'} = g^{x'y} = (g^{x'})^y$$

$\xrightarrow{E_{g^{xy'}}(m)}$ decrypt & reencrypt $\xrightarrow{E_{g^{x'y}}(m)}$

Mallory can read everything

Somehow Alice should whom she is talking to!

ALWAYS: be aware of your model of security.
Which type of attacks do you consider?

9.3.07 (3)

Turning in circles

We work in some group G and there is an element $g \in G$.

Question: when does $1, g, g^2, g^3, \ldots$ start repeating?

Thm (Lagrange)

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

In other words, the picture of $1, x, x^2, x^3, \ldots$ looks not only like

but like this:

and the length of the circle divides $\#G$ for any x.

Pf for G commutative.

Take a list of all group elements:

$$g_1, g_2, g_3, \ldots, g_{\#G}$$

and multiply each element with x:

$$xg_1, xg_2, xg_3, \ldots, xg_{\#G}.$$

Up to order, this also a list of all group elements!

(a) if $\underline{xg_i = xg_j}$ then $g_i = g_j$ or $i = j$. [simply multiply $xg_i = xg_j$ with x^{-1} so $g_i = x^{-1}xg_i = x^{-1}xg_j = g_j$ so then $i = j$.]

9.5.07 (4)

(b) Take an arbitrary element of G, say g_i. Find it on the new list!
we look j with $x g_j = g_i$,
so take j s.t. $g_j = \underbrace{x^{-1} g_i}_{\in G}$

then $x g_j = x x^{-1} g_i = g_i$. ⌟

Thus up to order both lists are equal.
Multiply all elements on each list:

$$g_1 \cdot g_2 \cdot \ldots \cdot g_{\#G} \underset{\uparrow}{=} x g_1 \cdot x g_2 \cdot \ldots \cdot x g_{\#G}$$

G commutative
& lists are equal
up to order.

$$\underbrace{\qquad\qquad\qquad}_{} \qquad = x^{\#G} \cdot \underbrace{g_1 \cdot g_2 \cdot \ldots \cdot g_{\#G}}_{}.$$

Divide and obtain:

$$1 = x^{\#G}.$$ □

Another example:

$p = 23$, $x = 5$: $G = \mathbb{Z}_{23}^{\times}$, $\#G = 22$.

i	0	1	2	3	4	5	6	7	8	9	10
x^i	1	5	2	10	4	-3	8	-6	-7	-12	9

11	12	13	14	15	16	17	18	19	20	21	$22 = \#G$
-1	-5	-2	-10	-4	3	-8	6	7	12	-9	1

$p = 23$, $x = 2$: $G = \mathbb{Z}_{23}^{\times}$, $\# G = 22$. 3.5.07 5

i	0	1	2	3	4	5	6	7	8	9	10	11
x^i	1	2	4	8	-7	9	-5	-10	3	6	12	1

$\rightsquigarrow x^{22} = 1$

$p = 23$, $x = -1$: $G = \mathbb{Z}_{23}^{\times}$, $\# G = 22$

i	0	1	2
x^i	1	-1	1

$\rightsquigarrow \quad x^{22} = 1$

Corollary (Euler)

Suppose $a \in \mathbb{Z}$, a coprime to N ie. $\gcd(a, N) = 1$.

Then
$$a^{\varphi(N)} = 1 \quad \text{in } \mathbb{Z}_N^{\times}$$
where
$$\varphi(N) := \# \mathbb{Z}_N^{\times}.$$

Pf This is simply the Thm of Lagrange applied to $G = \mathbb{Z}_N^{\times}$. □

Corollary (Little Fermat Theorem)

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

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

Pf Apply the previous to $N = p$ prime and compute $\varphi(p) = p - 1$.

$\mathbb{Z}_p$ is a field.

EEA

□

First consequence:

Square & multiply or exponentiating may first reduce the exponent by the group size (if known!).

Example

$$2^{5\,324\,427} = 2^7 (= -\cancel{7}). \text{ in } \underbrace{\mathbb{Z}_{11}^{\times}}_{10 \text{ elements!}}$$

$\ulcorner$
$$\begin{aligned} 2^{5\,324\,427} &= 2^{10 \cdot 532\,442 + 7} \\ &= \underbrace{(2^{10})}_{\substack{=1 \\ \uparrow \\ \text{by Fermat}}}{}^{532\,442} \cdot 2^7 = 1^{532442} \cdot 2^7 \\ &= 1 \cdot 2^7 = 2^7 \quad \text{in } \mathbb{Z}_{11}^{\times}. \end{aligned}$$
$\lrcorner$

Consequence for Diffie-Hellman key exchange:

technical: choose $x \in_R \mathbb{Z}$ in the interval $0 \leq x < \# G$.

security: $g^k = 1$ should not happen for a small k.

<u>Def</u> $\operatorname{ord}_G g = \min \{ k \in \mathbb{N}_{>0} \mid g^k = 1 \}$ is called the <u>order</u> of g in G.

Using this notion we can reformulate the theorem of Lagrange:

Corollary. Given a group G and $x \in G$.

Then $\operatorname{ord}_G x$ divides $\# G$.

Pf Suppose $x^k = 1$ and k is minimal.

By Lagrange we have $x^{\# G} = 1$.

Say, $\# G = 10$
and $x^3 = 1$.
$3 \nmid 10$ so we have to show that
3 is not minimal!

By EEA we obtain

g, s, t such that $s \cdot k + t \cdot \# G = g$

and $g = \gcd(k, \# G)$.

Then $x^g = (\underbrace{x^k}_{=1})^s (\underbrace{x^{\# G}}_{=1})^t = 1^s 1^t = 1$.

Since k is minimal we have $g \geq k$.

But $g \mid k$, thus $g = k$.

And of course $g \mid \# G$, thus $k \mid \# G$ $\square$

9.5.07

(8)

RSA (1978)

Rivest
Shamir
Adleman

Purpose: setup parameters and then send encrypted messages.

Setup Choose two primes $p \neq q$. (large! say 512 bit each.)

Let $N = p \cdot q$.
(we will work in $\mathbb{Z}_N^\times$.) group!

Let $L = (p-1)(q-1)$
(Actually, $\# \mathbb{Z}_N^\times = L$!)

Throw away p, q.

Choose $e, d \in_R \mathbb{Z}_L^\times$ such that

$$e \cdot d = 1 \text{ in } \mathbb{Z}_L^\times.$$

Throw away p, q, L.

Store: Private key (N, d).

Publish: Public key (N, e).

Bob → Alice, Encrypt a message $x \in \mathbb{Z}_N^{(\times)}$.

$y \leftarrow x^e$ in $\mathbb{Z}_N$.

Send y

Alice, Decrypt this:

$z \leftarrow y^d$ in $\mathbb{Z}_N$.

CORRECTNESS

Claim Always: $z = x$.

CORRECTNESS

14.05.07 (1)

Easy: If $x \in \mathbb{Z}_N^\times$ the

$$z = y^d = (x^e)^d$$
$$= x^{ed} = x^{1 + t \cdot L}$$

for some t

because $ed = 1$ in $\mathbb{Z}_L$,
i.e. $ed = 1 + tL$ in $\mathbb{Z}$
for some t.

$$= x \cdot \underbrace{(x^L)^t}_{=1}$$

$= 1$ by Thm (Euler)
using that $L = \# \mathbb{Z}_N^\times$.
(Exercise 3.27:
$\# \mathbb{Z}_N^\times = \# \mathbb{Z}_{pq}^\times = (p-1)(q-1) =: L$;))

$$= x \cdot 1 = x.$$

Wow!

What if $x \notin \mathbb{Z}_N^\times$?

First: The probability is very small:

$$\text{prob}(x \notin \mathbb{Z}_N^\times \mid x \in_R \mathbb{Z}_N)$$
$$= \frac{p+q-1}{pq} \approx 2^{-511} \approx 10^{-153} \ldots$$

$p \cdot q \approx 2^{512}$

very small ≈ 0.

Second: Any such 'bad' message x reveals p and q by computing $\gcd(x, N)$.

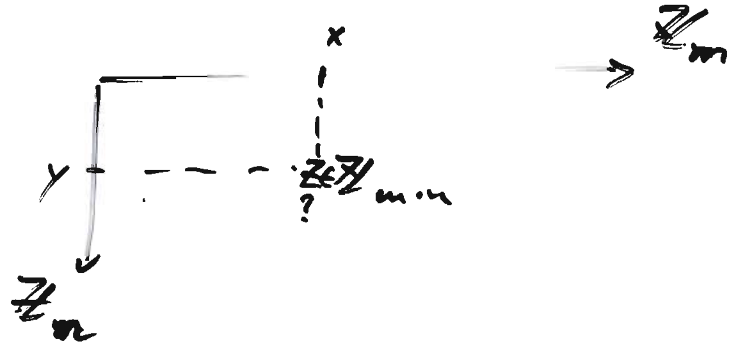
Third: $x^{ed} = x$

even in the 'bad' cases.

First proof for this: use ad hoc $x = \tilde{x}p$ or $x = \tilde{x}q$ and see what happens. (Ex)

Second proof: new tool:

Chinese Remainder Theorem



(as in Ex 4.2)

If $\gcd(m,n) =: g$ is not one

then $$\frac{m\cdot n}{g} \begin{cases} = \frac{m}{g}\cdot n = 0 \text{ in } \mathbb{Z}_n \\ = \frac{n}{g}\cdot m = 0 \text{ in } \mathbb{Z}_m \end{cases}$$

so the cell $(0,0)$ gets 0 and $\frac{mn}{g}$, and thus $\mathbb{Z}_{mn}$ cannot fill the table.

But if $\gcd(m,n) = 1$ then the table gets filled and any cell gets exactly one element.

CRT 'naive' formulation.

Suppose m, n are coprime integers. $\gcd(m,n)=1$

Given $x \in \mathbb{Z}_m$, $y \in \mathbb{Z}_n$ ....

find a number $z \in \mathbb{Z}_{m \cdot n}$

such that
$$z = x \text{ in } \mathbb{Z}_m,$$
$$z = y \text{ in } \mathbb{Z}_n.$$

Given $x \in \mathbb{Z}$, $0 \leq x < m$, $y \in \mathbb{Z}$, $0 \leq y < n$

find a number $z \in \mathbb{Z}$

such that
$$z \equiv_m x,$$
$$z \equiv_n y$$

Actually we have a map:

$$\pi: \begin{array}{ccc} \mathbb{Z}_{m \cdot n} & \longrightarrow & \mathbb{Z}_m \\ \hat{x} \bmod mn & \longmapsto & \hat{x} \bmod m \end{array}$$

This is a nice map: for x, y we have

$$\pi(x+y) = \pi(x) + \pi(y)$$

and
$$\pi(x \cdot y) = \pi(x) \cdot \pi(y)$$

[This is somehow obvious if π is defined by choosing $\hat{x} \in \mathbb{Z}$ such $x = \hat{x} \bmod mn$

Then
$$\begin{aligned} \pi(x+y) &= \widehat{x+y} \bmod m \\ &\overset{!}{=} (\hat{x} + \hat{y}) \bmod m \\ &= \hat{x} \bmod m + \hat{y} \bmod m \\ &= \pi(x) + \pi(y). \quad \ldots] \end{aligned}$$

Consider this:

<u>CRT'</u> Suppose m, n are coprime.
Then the map

$$\mathbb{Z}_{mn} \longrightarrow \mathbb{Z}_m \times \mathbb{Z}_n$$

is a bijective ring morphism.
ie. a ring isomorphism.
In particular, we obtain a group isomorphism

$$\mathbb{Z}_{mn}^{\times} \longrightarrow \mathbb{Z}_m^{\times} \times \mathbb{Z}_n^{\times}.$$

So we obtain the <u>corollary</u>:

$$\begin{array}{ccccc} \# \mathbb{Z}_{mn}^{\times} & = & \# \mathbb{Z}_m^{\times} & \cdot & \# \mathbb{Z}_n^{\times} \\ \shortparallel & & \shortparallel & & \shortparallel \\ \varphi(m \cdot n) & = & \varphi(m) & \cdot & \varphi(n) \end{array}$$

provided m, n are coprime.
[Note that $\varphi(4) = 2 \neq \varphi(2) \cdot \varphi(2)$!]

<u>Proof</u> (CRT')
Assume the naive version.
It says that the map $\mathbb{Z} \to \mathbb{Z}_m \times \mathbb{Z}_n$
is surjective and thus $\mathbb{Z}_{mn} \to \mathbb{Z}_m \times \mathbb{Z}_n$
is surjective.
Now, since $\# \mathbb{Z}_{mn} = m \cdot n = \# \mathbb{Z}_m \cdot \# \mathbb{Z}_n = \#(\mathbb{Z}_m \times \mathbb{Z}_n)$
the map must also be injective. □

Proof (CRT)

So given $x \in \mathbb{Z}_m$, $y \in \mathbb{Z}_n$

find $z \in \mathbb{Z}$ such

$$z = x \text{ in } \mathbb{Z}_m,$$
$$z = y \text{ in } \mathbb{Z}_n.$$

Consider $(x,y) = (1,0) \rightsquigarrow z_1$

and $(x,y) = (0,1) \rightsquigarrow z_2$

so $z_1 = 1$ in $\mathbb{Z}_m$, $z_1 = 0$ in $\mathbb{Z}_n$; $z_2 = 0$ in $\mathbb{Z}_m$, $z_2 = 1$ in $\mathbb{Z}_n$.

Claim If we can find z_1 and z_2 then

$$z = x z_1 + y z_2$$ solves the

original problem.

$$z \equiv_m x \underbrace{z_1}_{1} + y \underbrace{z_2}_{0} = x \cdot 1 + y \cdot 0 = x \text{ in } \mathbb{Z}_m,$$

$$z =_n x \underbrace{z_1}_{0} + y \underbrace{z_2}_{1} = x \cdot 0 + y \cdot 1 = y \text{ in } \mathbb{Z}_n.$$

By symmetry it suffices to find z_1:

So we look for

$z_1 = 1 - a \cdot m$ for some a

and $z_1 = 0 + b \cdot n$ for some b.

That is: $1 = \underbrace{a \cdot m}_{z_2} + \underbrace{b \cdot n}_{z_1}$ for some a, b.

and also .

14.5.07 (6)

We find a, b by EEA since m, n are coprime.

Then $z_1 = \underbrace{b \cdot n}_{= 1-am}$ gives $z_1 = 1 - am = 1$ in $\mathbb{Z}_m$
$z_1 = b \cdot n = 0$ in $\mathbb{Z}_n$

and $z_2 = \underbrace{a \cdot m}_{= 1-bn}$ gives $z_2 = a \cdot m = 0$ in $\mathbb{Z}_m$
$z_2 = 1 - bn = 1$ in $\mathbb{Z}_n$.

So we are done $\square$

CRA In: $x \in \mathbb{Z}_m$, $y \in \mathbb{Z}_n$
Out: $z \in \mathbb{Z}_{mn}$.

Compute $1 = am + bn$,
then $z = (x \cdot bn + y \cdot a \cdot m) \bmod m \cdot n$.

RSA Example:

$p = 5$, $q = 7$,
$N = 35$
$L = 24$
Guess $e = 17$. (uniform random choice! in $\mathbb{Z}_{24}^\times$)

Then $d_0 = 17$.

By coincidence 24 is very special any number in $\mathbb{Z}_{24}^\times$ has square 1! ← CRT

24		1	0
17	1	0	1
7	2	1	-1
3	2	-2	3
1	3	5	-7
0		-17	24

14.5.07 (7)

Wow, CRT is fun!

Alice has to calculate

$$y^d \text{ in } \mathbb{Z}_{pq}.$$

Why not do this in $\mathbb{Z}_p \times \mathbb{Z}_q$?

Compute $z_p = (y \bmod p)^d$

and $z_q = (y \bmod q)^d$

and then use CRT to find

$$\left.\begin{array}{ll} z = z_p & \text{in } \mathbb{Z}_p, \\ z = z_q & \text{in } \mathbb{Z}_q. \end{array}\right\} \Rightarrow z = y^d \text{ in } \mathbb{Z}_N.$$

Say Alice' Job is to return this value z.

And further say Alice is a smartcard

and we can disturb Alice so she

makes an error in exactly one place.

So we get

$$z' = z_p \text{ in } \mathbb{Z}_p$$

$$z' \neq z_q \text{ in } \mathbb{Z}_q.$$

But we may have prepared $y = x^e$ then $x = y^d$ in $\mathbb{Z}_N$

and

$$z' - x = 0 \text{ in } \mathbb{Z}_p$$

$$z' - x \neq 0 \text{ in } \mathbb{Z}_q$$

Thus $\gcd(z' - x, N) = p$.

16.5.07 1

Remind:

CRT Given m, n coprime then

$$\mathbb{Z}_{m \cdot n} \xrightarrow{\cong} \mathbb{Z}_m \times \mathbb{Z}_n$$

Example of its use:

Say m, n are both prime.

How many $x \in \mathbb{Z}_{mn}$ are there with $x^2 = 1$ in $\mathbb{Z}_{mn}$?

Answer: look for solution in $\mathbb{Z}_m$ and find $\pm 1 \in \mathbb{Z}_m$ there.

Since m is prime, $\mathbb{Z}_m$ is a field and there can be no other solutions.

{ If p is a polynomial over some field and $p(x) = 0$ then

$$p(T) = q(T) \cdot (T - x) + r$$

field! $\Rightarrow$ with $\underset{0 \text{ or } -\infty}{\deg(r)} < \deg(T - x) = 1$

Now, $\underbrace{p(x)}_{=0} = q(x) \cdot \underbrace{(x - x)}_{=0} + r$

so $r = 0$. Thus $p(T) = q(T) \cdot (T - x)$.

Thus there can be at most $\deg(p)$ zeros. }

Also in $\mathbb{Z}_n$ we find ± 1 as only solutions.

Thus in $\mathbb{Z}_m \times \mathbb{Z}_n$ we have 4 solutions:

$\underset{+1}{(+1, +1)}$, $\underset{-1}{(-1, -1)}$, $(+1, -1)$, $(-1, +1)$.

EX?

RSA is correct! $\quad x^{ed} = x$

in all cases!

CORRECTNESS

Pf we want this equation in $\mathbb{Z}_{pq}$.

By CRT
$$\mathbb{Z}_{pq} \cong \mathbb{Z}_p \times \mathbb{Z}_q .$$

So is $x^{ed} = x$ in $\mathbb{Z}_p$?

Now, we know that
$$x^{p-1} = 1 \quad \text{in } \mathbb{Z}_p^\times$$
by Fermat, provided $x \neq 0$.

Thus
$$x^p = x \quad \text{in } \mathbb{Z}_p$$
for $x \neq 0$. But this is true for $x = 0$ as well! Inductively, this gives

$x^1 = x^p$

$$x = \overset{1+1(p-1)}{x^{p}} = x^{1+2(p-1)} = x^{1+3(p-1)}$$
$$= \dots = x^{1+t\cdot(p-1)} \quad \text{in } \mathbb{Z}_p \text{ for any } t \geq 0.$$

Now, $ed = 1 + s\,(q-1)\,(p-1)$;

so
$$x^{ed} = x^{1+\underbrace{s\,(q-1)}_{t}\cdot(p-1)} = x \quad \text{in } \mathbb{Z}_p .$$

~~Similarly~~ Similarly, $x^{ed} = x$ in $\mathbb{Z}_q$.

So $x^{ed} = x$ in $\mathbb{Z}_p \times \mathbb{Z}_q \cong \mathbb{Z}_{pq}$

so $x^{ed} = x$ in $\mathbb{Z}_{pq}$. $\square$

RSA is efficient

16.3.17

3

Tasks:

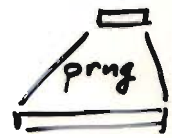
Setup: generate primes

= · generate a random number — true random bits / pseudo random number generator

practical: $O(n^3)$

· test whether it is prime

→ good probabilistic tests available $O(n^3)$



multiply $O(n^2)$

finding e, d:

· generate a random number $O(n^3)$

· EEA $O(n^2)$

$O(n^3)$

EFFICIENCY

Encryption: one exponentiation $O(n^3)$

Decryption: same.

So everything is polynomial time.

In practise: Setup for 1024 or 2048 bits takes, say a minute.

Encr/Dec takes a few milliseconds.

Is RSA secure?

16.5.07

4

What would be a <u>total</u> break?

Eve knows (N, e) and some y and lots of pairs (x, x^e) [and maybe she can get some pairs $(\tilde{y}, \tilde{y}^d)$ ...].

(i) Eve can find the primes p, q such that $N = pq$.

(ii) Eve can find the rep. length L.

(iii) Eve can derive d.

(iv) Eve finds x ~~...~~ with $x^e = y$.

SECURITY

Obvious: (i) $\Leftrightarrow$ (ii)

"$\Leftarrow$": Consider $(T-p)(T-q)$

$$= T^2 - \underbrace{(p+q)}\,T + \underbrace{pq}_{=N}$$

Eve knows $L = (p-1)(q-1)$

$$= pq - p - q + 1$$

$$= N+1 - (p+q),$$

so $\quad p+q = N+1-L.$

So Eve knows this polynomial. And thus can compute its zeroes. (midnight formula!)

$\lrcorner$

(ii) $\Rightarrow$ (iii) ✓

2·(iii) $\Rightarrow$ (ii): (iii) gives d with $ed-1 = t\cdot L$.

Second (iii) gives d' with $e'd'-1 = t'L$.

Here $\tilde{t}$ is small! So $\gcd(ed-1, e'd'-1) = \tilde{t}\cdot L$.

16.5.07
5

So compute $\hat{t} = \left\lceil \frac{\gcd(ed-1,\, e'd'-1)}{\tilde{N}} \right\rceil$.

and try $\hat{t}, \hat{t}+1, \ldots$.

This gives L then ...

iii => iv ✓

OPEN PROBLEM: Does iv imply iii?

"The security of RSA is based on the difficulty of factoring."

SECURITY

Would i <=> iv be enough?
(assuming that factoring IS difficult)

Suppose an attacker can given y compute $\mathrm{Bit}_0(x)$.

Is that a problem? YES!

say $\mathrm{Bit}_0(x)=0$ then $x = 2x'$ "hard core bit"

thus $y = 2^e x'^e$

so $y' = y/2^e = x'^e$

Now $\mathrm{Bit}_0(x') = \mathrm{Bit}_1(x)$!

--- This gives x!

16.5.07

6

Definition of Security is a very intricate problem!

Best to date:

There should not be a probabilistic polynomial time Turing machine that can decide a 0/1-question on x with non-negligible advantage.

Signatures

21.5.07

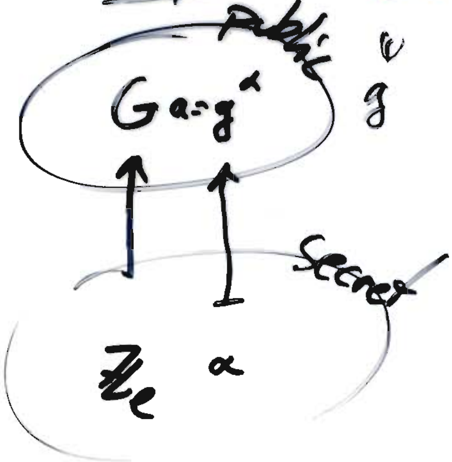
e

- identifies the signer
- makes sure the document is not modified
- binds Joachim to his yesterday's statement

21.5.07

(2)

El Gamal signatures (1978?)



Setup: G group, strictly: $G = \mathbb{Z}_p^\times$.

$g \in G$

p a large prime so that the discrete log problem is difficult.

(In particular: $p-1 = \# G$ should not be a product of small primes.)

$\ell :=$ order(g) is large, actually it should contain a large prime factor.

public!

Further, let $^*: G \to \mathbb{Z}_\ell$ be some very very simple map. (Like reinterpreting integers mod p as integers mod ℓ.)

Personal setup: Alice chooses

$\alpha \in \mathbb{Z}_\ell$ as a private key

and computes

$a = g^\alpha \in G$ as a public key.

~~Signature~~

Verify: If (*) $\boxed{a^{b^*} b^{\beta} = g^{msg}, \quad \beta \in \mathbb{Z}_\ell, b \in G}$

then (b, β) is a valid signature for the message msg.

Verifier knows $g \in G$ from setup, a from the public key, and msg, b, β from signed document.

Necessarily, $\beta \in \mathbb{Z}_\ell$, $b \in G$.

27.5.07

3

Total break

(i) Find (b, γ) so that it is a signature to msg.

i.e. find a solution of (*) .

(ii) Split the problem and solve the two equations

$$d = a^{b^*}$$

$$d \cdot b^{\gamma} = g^{msg}$$

and then brute force ...

Plan: Choose d, then find b^* by taking a dlog.

Choose b such this gives the found b^*.

Find another dlog to get γ:

$$b^{\gamma} = g^{msg} / d.$$

Need 2 DLs to solve the 'ElGamal Problem'.

(iii) Choose b then compute a dlog:

$$b^{\gamma} = g^{msg} a^{-b^*}$$

Need 1 DL to solve the EP.

(iv) Other plan: choose γ and try to find b ...

$$a^{b^*} b^{\gamma} = \text{sth.}$$

Seems to be even more difficult ...

21.5.07

(4)

Signature

The signer can use the secret key α so she has to solve

$$g^{\alpha b^*} b^{\gamma} = g^{msg}$$

So Alice chooses b as $g^{\beta} =: b$, she chooses $\beta \in \mathbb{Z}_e^{\times}$ and computes $b := g^{\beta}$.
Now she has to solve

$$g^{\alpha b^*} g^{\beta\gamma} = g^{msg}$$

$$\parallel$$

$$g^{\alpha b^* + \beta\gamma}$$

So solving

$$\boxed{\alpha b^* + \beta\gamma = msg \text{ in } \mathbb{Z}_e}$$

gives a solution.
(So if we make sure that β is invertible: $\beta \in \mathbb{Z}_e^{\times}$

then

$$\gamma = \beta^{-1}(msg - \alpha b^*) \text{ in } \mathbb{Z}_e. \quad)$$

Sign(msg)
- $\beta \in \mathbb{Z}_e^{\times}$
- $b := g^{\beta}$
- $\gamma \in \mathbb{Z}_e$ solves $\alpha b^* + \beta\gamma = msg$.
- return (b, γ)

Technical problem: the signed document is $3\times$ as large as the unsigned one.

27.5.07

(5)

We use a hash function value instead of the message itself.

Let $h: \{0,1\}^* \longrightarrow \mathbb{Z}_e$.

be a hash(?) function, easily computable.

h should be <u>one-way</u>.

It should be difficult given $\eta \in \mathbb{Z}_e$ to find a message $msg \in \{0,1\}^*$ with hashvalue $h(msg) = \eta$.

we cannot have impossible here!

h should be <u>collision-resistent</u>

h should be <u>second</u> preimage resistent.

It should be difficult given a message $msg_1 \in \{0,1\}^*$ to find another message $msg_2 \in \{0,1\}^*$ such that $h(msg_1) = h(msg_2)$.

h should be collision resistant:
It should be difficult to find two messages msg_1, $msg_2 \in \{0,1\}^*$ that are different $msg_1 \neq msg_2$ with same hash value $h(msg_1) = h(msg_2)$.

24.5.07 (6)

4.6.07 (1)

Definition Security of a signature

An attacker that can
given signatures for any number of chosen documents (which may depend on each other) — CHOSEN MSG ATTACK
can forge a new document with a valid signature — EXISTENTIAL FORGERY
with a non-negligeble propability
in polynomial time
breaks the scheme.

A signature scheme is considered secure if there is no such attacker.

Details → 'Provable security' or Reductionist's security

14.6.07 (2)

Let's apply this to the ElGamal scheme:

<u>Setup</u> Choose a group G, say $G = \mathbb{Z}_p^\times$,
choose an element $g \in G$
of large prime order $e = \mathrm{ord}_G(g)$.

(Say $e \sim 160$ bit, and $p \sim 1024$ bit.)

↳ same security wrt the known attacks on DL
in generic groups, in groups $\mathbb{Z}_p^\times$.

$\alpha \in_R \mathbb{Z}_e$, ← secret signing key

$a := g^\alpha \in G$ ← public signing key

<u>Signature generation</u>

Given a message m.

Choose $\beta \in_R \mathbb{Z}_e$,
compute $b = g^\beta \in G$, and
solve $\boxed{\alpha b^* + \beta\gamma = h(m)}$

Ex: twice same β

where $*: G \to \mathbb{Z}_e$ is some simple, (almost invertible) function

and $h: \{0,1\}^* \longrightarrow \mathbb{Z}_e$ is a hash function.

Output: (b, γ) as a signature.

4.6.07
③

Verification

Check $b \in G$, $\gamma \in \mathbb{Z}_\ell$, and

$$\boxed{a^{b^*} b^{\gamma} = g^{h(m)}} \text{ in } G.$$

Suppose h is not 2^{nd} préimage resistant,
ie. there is an algorithm TWO which
computes given msg_1 another $msg_2 \neq msg_1$
with $h(msg_1) = h(msg_2)$
in poly-time with non-negligible probability.

Then $\mathcal{A}$:
Choose msg_1 arbitrarily.
Ask the signer for a signature on $msg_1 \to (b, \gamma)$ with $a^{b^*} b^{\gamma} = g^{h(msg_1)}$.

Ask TWO for $msg_2 \neq msg_1$
with $h(msg_2) = h(msg_1)$

Output $(msg_2, (b, \gamma))$.

Clearly, $\mathcal{A}$ runs in 'same' time as TWO;
and it succeeds if TWO succeeds.
So if TWO is too good then $\mathcal{A}$ is too good
and thus the scheme is insecure.

Together:

if the signature scheme is secure then the hash must be 2^{nd} preimage resistent.

Suppose h is not collision-resistent, ie. there is an algorithm COLLISION which outputs $msg_1 \neq msg_2$ with $h(msg_1) = h(msg_2)$

in poly-time with non-negligible probability.

Thus

$\mathcal{A}'$: Call COLLISION to get $msg_1 \neq msg_2$ with $h(msg_1) = h(msg_2)$.

Ask the signer for a signature (b, γ) on msg_1.

Output: $(msg_2, (b, \gamma))$.

Thus we get $a^{b^*} b^{\gamma} = g^{h(msg_1)} = g^{h(msg_2)}$

Again: if COLLISION is too good then $\mathcal{A}'$ is too good.

Q:

Theorem. If the scheme is secure then the hash function must be collision-resistent.

Similarly

If the scheme is secure then the DL with basis $g \in G$ must be difficult.

Three properties for hash functions: 4.6.07 5

h is one-way

$\Uparrow \quad \Downarrow$

h is 2^{nd} preimage resistant

$\Uparrow \quad \not\Downarrow$

h is collision resistant.

<u>Pf</u> If OW attacks one-wayness

then TWO' : Accept msg_1

Ask OW for a preimage msg_2 of $h(msg_1)$.

Output msg_2.

is a slightly worse attacker on 2^{nd} preimage-resistance. <u>Small gap!</u>

If TWO attacks 2^{nd} preimage resistance

then COLLISION' : Choose msg_1 randomly.

Call TWO with msg_1 to obtain $msg_2 \neq msg_1$ with $h(msg_2) = h(msg_1)$.

Output: (msg_1, msg_2)

is a poly-time attacker with same success prob. as TWO.

$\square$

Brute force on these three properties: (4.6.07 / 6)

Say $h: \{0,1\}^* \to \mathbb{Z}_\ell$,

with ℓ an n-bit number.

oneway : $\boxed{\ell \approx 2^n}$

~~given~~ k what is

$$E[\text{prob}(\ h(msg) = k \mid msg \text{ random})]$$

$$= \frac{1}{\# \mathbb{Z}_\ell}$$

so we expect $\# \mathbb{Z}_\ell = \ell$

trials until we find msg with

$h(msg) = k$.

2^{nd} preimage : We expect ℓ trials.

$\boxed{\ell \approx 2^n}$

collision : $\boxed{\sqrt{\ell} \approx 2^{n/2}}$

Attacker: Repeat

choose a new msg_i.

until $h(msg_i)$

$\in \{h(msg_1), \ldots, h(msg_{i-1})\}$

Output the two colliding msgs

msg_i, msg_j.

Runtime: $\mathcal{O}(\sqrt{\ell}) = \mathcal{O}(2^{n/2})$

Trailer on Signatures

6.6.07
(1)

we have seen El Gamal signatures

$$a^{b^*} b^{\delta} = g^{h(m)}$$

- need to work in a group with difficult DLP
- need a collision-resistent hash function.

For variants of this scheme reductions to these two necessary conditions are available.

Problem : Hash crisis!

MD4 128-bit BROKEN,
(64 rounds) need only 2 or 3 hash computations
→ seconds for a new collision

MD5 → 128 bit BROKEN
(80 rounds) ~~(160?)~~ → about 15 minutes for a new collision
(instead of, say, a year for 2^{64} hash computations)

SHA1 → 160 bit BROKEN
(80 rounds) attack needs 'only' 2^{63} hash computations
NO collision published yet (afaik)

RIPEMD → 160 bit similar design!
(80 rounds)

One further family: SHA-2

SHA-256 → 256 bits (> 80 rounds?) : similar design ~~pro~~ probably secure in practice because of its dimensions.

No tested replacement, yet.

Practical security

1 No better attack than "generic" ones.

Lecture Noteselectronic

For your information the following slides show the definition of MD4, MD5 and SHA1.

WARNING!

This hash function is completely broken!

Collisions can be found within seconds.

Do NOT use it any more.

Algorithm. MD4.
Input: A message $x \in \{0,1\}^*$.
Output: A hash value $H \in \{0,1\}^{128}$.

Constants and round functions:

1. $h \leftarrow (\texttt{67452301}, \texttt{EFCDAB89}, \texttt{98BADCFE}, \texttt{10325476})$.

$$K_j \leftarrow \begin{cases} \texttt{00000000}, & 0 \le j < 16, \\ \texttt{5A827999}, & 16 \le j < 32, \quad (\text{32 bits of } \sqrt{2}) \\ \texttt{6ED9EBA1}, & 32 \le j < 47, \quad (\text{32 bits of } \sqrt{3}) \end{cases}$$

$$z[j] = \begin{cases} j, & 0 \le j < 16, \\ j_1 j_0 j_3 j_2, & 16 \le j < 32, \\ j_0 j_1 j_2 j_3, & 32 \le j < 48, \end{cases}$$

where j_i denotes bit i of the binary representation of j.

$s[\ 0..15] = [3, 7, 11, 19, 3, 7, 11, 19, 3, 7, 11, 19, 3, 7, 11, 19]$,

$s[16..31] = [3, 5, 9, 13, 3, 5, 9, 13, 3, 5, 9, 13, 3, 5, 9, 13]$,

$s[32..47] = [3, 9, 11, 15, 3, 9, 11, 15, 3, 9, 11, 15, 3, 9, 11, 15]$.

$$f_j(B, C, D) = \begin{cases} (B \wedge C) \vee (\overline{B} \wedge D), & 0 \le j < 16, \\ (B \wedge C) \vee (C \wedge D) \vee (D \wedge B), & \\ B \oplus C \oplus D, & 32 \le j < 48. \end{cases}$$

Precalculations:

2. Padding: $\tilde{x} \leftarrow x|1|0^d|\langle|x|\rangle_{64}$ with $0 \le d < 512$ such that $|\tilde{x}|$ is a multiple of $512 = 16 \cdot 32$.
3. Cut $\tilde{x}$ into 32-bit words: $\tilde{x} = x_0 x_1 x_2 \ldots x_{16m-1}$.
4. Initialize: $(H_1, H_2, H_3, H_4) \leftarrow h$.

Main calculation:

5. For $i = 0..m-1$ do 6–10
6. $\quad (A, B, C, D) \leftarrow (H_1, H_2, H_3, H_4)$.
7. $\quad$ For $j = 0..47$ do 8–9
8. $\qquad t \leftarrow (A + f_j(B, C, D) + x_{z[j]} + K_j) \circlearrowleft s[j]$.
9. $\qquad (A, B, C, D) \leftarrow (D, t, B, C)$.
10. $\quad (H_1, H_2, H_3, H_4) \leftarrow (H_1 + A, H_2 + B, H_3 + C, H_4 + D)$.
11. Return $H_1|H_2|H_3|H_4$.

WARNING!

This hash function is completely broken!

Collisions can be found within 15 minutes.

Do NOT use it for signing any more.

ALGORITHM. MD5.
Input: A message $x \in \{0,1\}^*$.
Output: A hash value $H \in \{0,1\}^{128}$.

COnstants and round functions:

1. $h \leftarrow (\texttt{67452301}, \texttt{EFCDAB89}, \texttt{98BADCFE}, \texttt{10325476})$.
   $K_j \leftarrow 32$ Bits von $|\sin(j+1)|$.

$$z[j] = \begin{cases} j, & 0 \le j < 16, \\ j_1 j_0 j_3 j_2, & 16 \le j < 32, \\ j_0 j_1 j_2 j_3, & 32 \le j < 48, \end{cases}$$

where j_i denotes bit i of the binary representation of j.

$s[\ 0..15] = [7, 12, 17, 22, 7, 12, 17, 22, 7, 12, 17, 22, 7, 12, 17, 22]$,
$s[16..31] = [5, 9, 14, 20, 5, 9, 14, 20, 5, 9, 14, 20, 5, 9, 14, 20]$,
$s[32..47] = [4, 11, 16, 23, 4, 11, 16, 23, 4, 11, 16, 23, 4, 11, 16, 23]$,
$s[48..63] = [6, 10, 15, 21, 6, 10, 15, 21, 6, 10, 15, 21, 6, 10, 15, 21]$.

$$f_j(B,C,D) = \begin{cases} (B \wedge C) \vee (\overline{B} \wedge D), & 0 \le j < 16, \\ (B \wedge D) \vee (C \wedge \overline{D}), & 16 \le j < 32, \\ B \oplus C \oplus D, & 32 \le j < 48, \\ C \oplus (B \vee \overline{D}), & 48 \le j < 64. \end{cases}$$

Precalculation:

2. Padding: $\tilde{x} \leftarrow x|1|0^d|\langle |x| \rangle_{64}$ with $0 \le d < 512$ such that $|\tilde{x}|$ is a multiple of $512 = 16 \cdot 32$.
3. Cut $\tilde{x}$ into 32-bit words: $\tilde{x} = x_0 x_1 x_2 \ldots x_{16m-1}$.
4. Initialize: $(H_1, H_2, H_3, H_4) \leftarrow h$.

Main calculation:

5. For $i = 0..m-1$ do 6–10
6. $\quad (A, B, C, D) \leftarrow (H_1, H_2, H_3, H_4)$.
7. $\quad$ For $j = 0..63$ do 8–9
8. $\quad\quad t \leftarrow (A + f_j(B,C,D) + x_{z[j]} + K_j) \lll s[j]$.
9. $\quad\quad (A, B, C, D) \leftarrow (D, B + t, B, C)$.
10. $\quad (H_1, H_2, H_3, H_4) \leftarrow (H_1 + A, H_2 + B, H_3 + C, H_4 + D)$.
11. Return $H_1|H_2|H_3|H_4$.

Algorithm. SHA-1.
Input: A message $x \in \{0,1\}^*$.
Output: A hash value $H \in \{0,1\}^{160}$.

Constants and round functions:

1. $h \leftarrow (\texttt{67452301}, \texttt{EFCDAB89}, \texttt{98BADCFE}, \texttt{10325476}, \texttt{C3D2E1F0})$.

$$K_j \leftarrow \begin{cases} \texttt{5A827999}, & 0 \le j < 20, \quad \text{(32 bits of } \sqrt{2}) \\ \texttt{6ED9EBA1}, & 20 \le j < 40, \quad \text{(32 bits of } \sqrt{3}) \\ \texttt{8F1BBCDC}, & 40 \le j < 60, \quad \text{(32 bits of } \sqrt{5}) \\ \texttt{CA62C1D6}, & 60 \le j < 80. \quad \text{(32 bits of } \sqrt{7}) \end{cases}$$

$$f_j(B,C,D) = \begin{cases} (B \wedge C) \vee (\overline{B} \wedge D), & 0 \le j < 20, \\ B \oplus C \oplus D, & 20 \le j < 40, \\ (B \wedge C) \vee (C \wedge D) \vee (D \wedge B), & \\ B \oplus C \oplus D, & 60 \le j < 80. \end{cases}$$

Precalculations:

2. Padding: $\tilde{x} \leftarrow x|1|0^d|\langle |x| \rangle_{64}$ mit $0 \le d < 512$ so, that $|\tilde{x}|$ is a multiple of $512 = 16 \cdot 32$.
3. Cut $\tilde{x}$ in 32-bit words: $\tilde{x} = x_0 x_1 x_2 \ldots x_{16m-1}$.
4. Initialize: $(H_1, H_2, H_3, H_4, H_5) \leftarrow h$.

Main calculation:

5. For $i = 0..m-1$ do 6–13
6. For $j = 0..15$ do $W_j \leftarrow x_{16i+j}$.
7. For $j = 16..79$ do
8. $W_j \leftarrow (W_{j-3} \oplus W_{j-8} \oplus W_{j-14} \oplus W_{j-16}) \lll 1$.
9. $(A, B, C, D, E) \leftarrow (H_1, H_2, H_3, H_4, H_5)$.
10. For $j = 0..79$ do 11–12
11. $t \leftarrow A \lll 5 + f_j(B,C,D) + E + W_j + K_j$.
12. $(A, B, C, D, E) \leftarrow (t, A, B \lll 30, C, D)$.
13. $(H_1, H_2, H_3, H_4, H_5) \leftarrow$
$(H_1 + A, H_2 + B, H_3 + C, H_4 + D, H_5 + E)$.
14. Return $H_1|H_2|H_3|H_4|H_5$.

IPsec

6.6.07

3

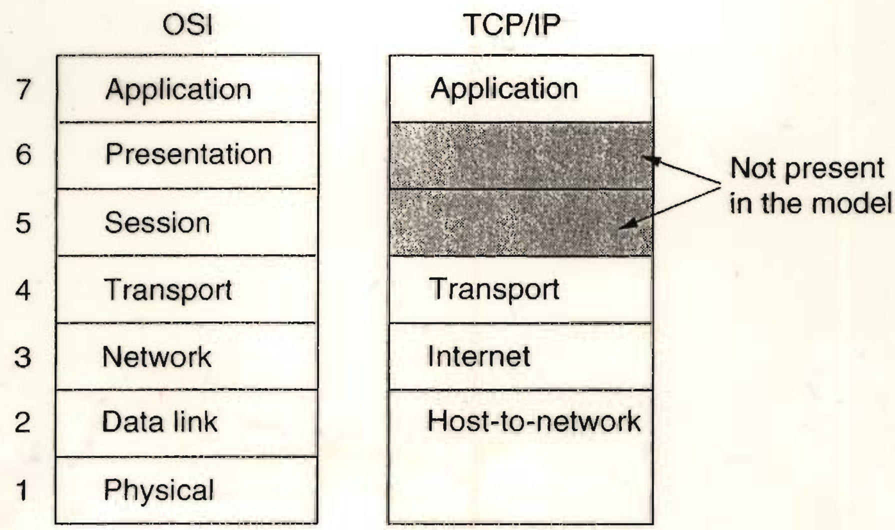


Figure 1-21. The TCP/IP reference model.

application

header | data

next lower layer

header

Physical Info | IP | IPsec | TCP | application header | data

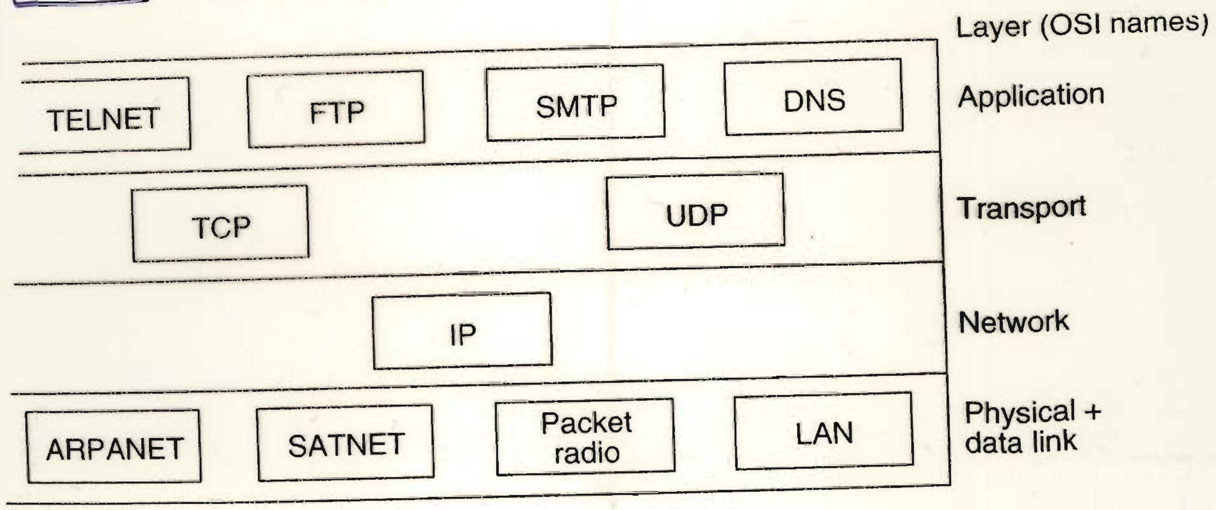


Figure 1-22. Protocols and networks in the TCP/IP model initially.

AH - authentication header
ESP - encapsulating security protocol

Task	AH	ESP encrypt	ESP both
Access control	+	+	+
Connectionless integrity	+	—	+
Data origin authentication	+	—	+
Rejection of replayed packets	+	(+)	+
Confidentiality	—	+	+
Limited traffic flow confidentiality	—	+	+

SA - security association
→ SPI (security parameter index) (32 bit #)
contains:
- IP destination address
- security protocol identifier → AH
  → ESP → encr. only
  → encr & auth.
- sequence number counter (32 bit)
- sequence counter overflow
- anti-replay
- AH - info : Authentication algorithm,
  keys
  key life time ...

- ESP-info: Encryption algorithm, (& authentication algorithm)
  keys,
  key life times,
  initial values ...
- life time of SA (usually 8 hours)
- IPsec protocol mode:
  tunnel, transport, wildcard
- path MTU: max. packet size & aging variables

SPD - security policy database

SAP → entries for each SA

SPD → allowed IPs

authentication header

AH

# octets	
1	next header
1	payload length
2	unused
4	SPI (Security Parameter Index)
4	sequence number
variable	authentication data
	data

authentication data
≙ signature
= essentially this is a keyed secure hash value

6.6.7
6

Encapsulating Security Protocol header

ESP

# octets		
4	SPI (Security Parameters Index)	header
4	sequence number	
variable	IV (initialization vector)	
variable	data	
variable	padding	trailer
1	padding length (in units of octets)	
1	next header/protocol type	
variable	authentication data	

11.6.07
1

SA , Security Association
  $\hat{=}$ simplex protected connection

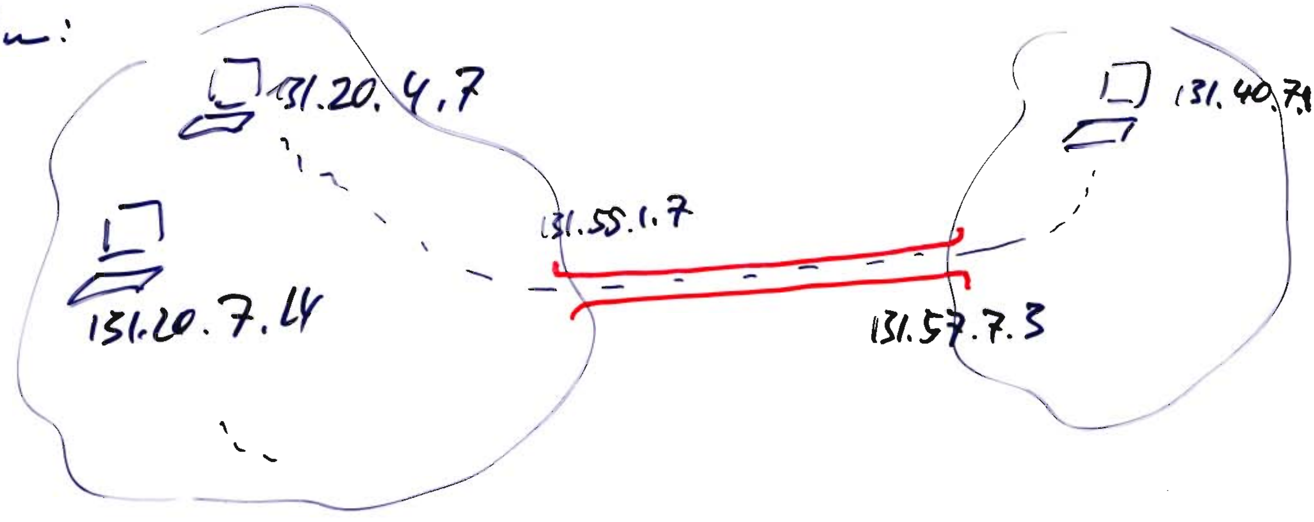
SAD – database of all inbound & outbound SAs

SPD – database of rules: which packets to
- DISCARD
- BYPASS
- PROTECT

AH/ESP – security envelopes

Tunnel and transport mode

Situation:



11.6.07

2

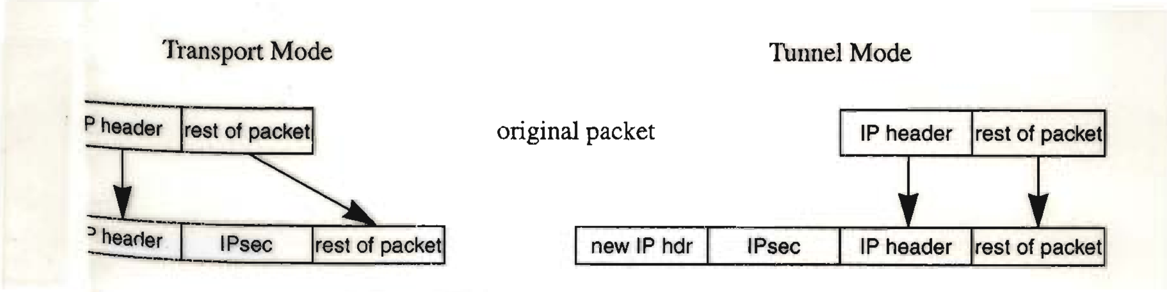


Figure 17-1. Transport Mode and Tunnel Mode

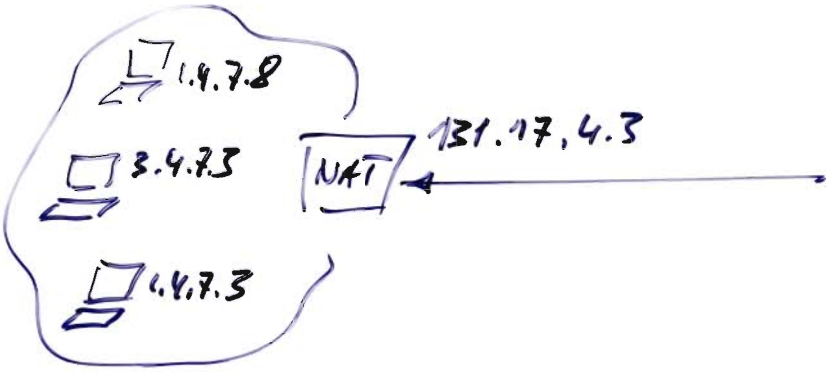
Transport mode is best suited for station to station connections.

Tunnel mode also allows to connect two subnets.

Side remark:

AH protects IP header. It is unclear why this is necessary; but even if so ESP in tunnel mode would provide that!

NAT – network address translation



AH protects destination IP address, so exchanging it destroys the signature.

11.6.07
3

NAT must be able to read information in the data part of the IP packet. But that might be encrypted.

Firewalls

→ filter packets according to
source IP
used protocol
maybe port# ← OOPS! Invisible in encrypted IPsec.
...

IPv4 / IPv6

IPv4

4=IP
6=TCP
17=UDP

size	
4 bits	version
4 bits	header length (in 4-octet units)
1 octet	type of service
2 octets	length of header plus data in this fragment
2 octets	packet identification
3 bits	flags (don't fragment, and last fragment)
13 bits	fragment offset
1 octet	hops remaining, known as TTL (time to live)
1 octet	protocol (next header)
2 octets	header checksum
4 octets	source address
4 octets	destination address
variable	options

50=ESP, 51=AH

IPv6

# octets	
4	version (4 bits) Itype of servicelflow label
2	payload length
1	next header
1	hops remaining
16	source address
16	destination address

IPv6

# octets	
1	next header
1	length of this header
variable	data for this header

← IPv4's protocol

AH

[IP] [AH] [data]

authenticates all immutable fields in IP
and the data.

IPv4 immutable: type of service
payload length

mutable: fragment offset

fragmen bits?
ASCII IPs
to be replaced?

always 0,
but similar to
payload length.

IPv6 TYPE of each option
indicates whether it's
mutable or not, eg.

type of service: mutable

Other things: destination address
mutable but predictable
→ use predicted value
for signature

ESP

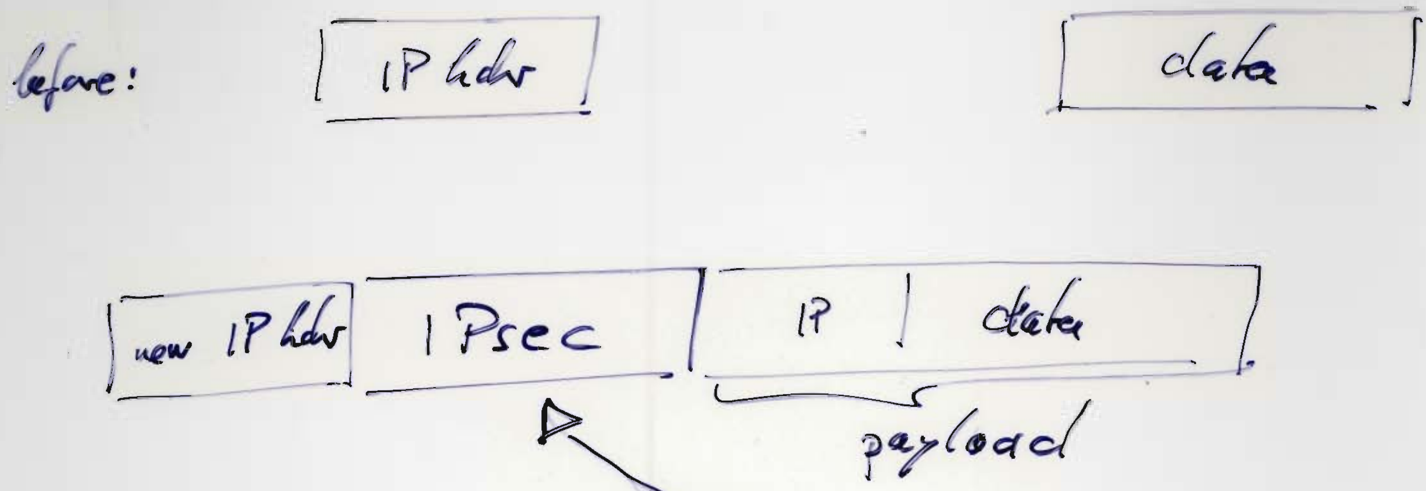
can do encryption and optionally authenticate
It does not include any IP header info in the
signature!

⇝ You can use 'null-encryption' if you
don't want to encrypt.

IPsec : more details

18.06.07
①

Say we look again at tunnel mode:



Specifically, ESP : ESP hdr.

Allows : Encryption or auth. or both.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ----
|               Security Parameters Index (SPI)                 | ^Int.
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |Cov-
|                      Sequence Number                          | |ered
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ | ----
|                    Payload Data* (variable)                   | |   ^
~                                                               ~ |   |
|                                                               | |Conf.
+               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |Cov-
|               |     Padding (0-255 bytes)                     | |ered*
+-+-+-+-+-+-+-+-+               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ |   |
|                               |  Pad Length   | Next Header   | v   v
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ ------
|         Integrity Check Value-ICV   (variable)                |
~                                                               ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

← new IP hdr

ESP hdr: Security Parameters Index (SPI) — identifies SA; Sequence Number — 32bit counter

Payload Data — the data

Padding — use this to make length a multiple of, say, 16

ESP trailer

Integrity Check Value — signature

Figure 1. Top-Level Format of an ESP Packet

* If included in the Payload field, cryptographic synchronization data, e.g., an Initialization Vector (IV, see Section 2.3), usually is not encrypted per se, although it often is referred to as being part of the ciphertext.

18.06.07

(2)

What is done?

Table 1. Separate Encryption and Integrity Algorithms

	# of bytes	Requ'd [1]	What Encrypt Covers	What Integ Covers	What is Xmtd	
SPI	4	M	–	Y +	plain	
Seq# (low-order bits)	4	M	–	Y +	plain	
IV	variable	O	–	Y +	plain	payload
IP datagram [2]	variable	M or D	Y +	Y +	cipher[3]	payload
TFC padding [4]	variable	O	Y +	Y +	cipher[3]	payload
Padding	0-255	M	Y +	Y +	cipher[3]	
Pad Length	1	M	Y +	Y +	cipher[3]	
Next Header	1	M	Y +	Y +	cipher[3]	
Seq# (high-order bits)	4	if ESN [5]	–	Y +	not xmtd	
ICV Padding	variable	if need	–	Y +	not xmtd	
ICV	variable	M [6]	–	—	plain	

integrity check value

[1] M = mandatory; O = optional; D = dummy
[2] If tunnel mode -> IP datagram
    If transport mode -> next header and data
[3] ciphertext if encryption has been selected
[4] Can be used only if payload specifies its "real" length
[5] See section 2.2.1
[6] mandatory if a separate integrity algorithm is used

How to apply this encr. & auth?

(1) encapsulate for transport/tunnel mode
    no payload

(2) add padding (TFC and encr. padding)
    as needed/wanted.

(3) encrypt as specified by SA and IV

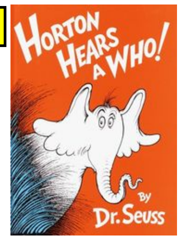
(4) sign (authenticate) the (encrypted)
    packet including ICV padding, ESN (extended seq#),
    but excluding ICV

So here: first encrypt
then authenticate/sign

18.6.07

(3)

Aka. as Horton's principle:



PARADIGMA

A signature must always protect the plaintext.

One solution: first authenticate

then encrypt.

But do not forget the secrecy: See Exercise 11.3.

Advantage of the other order: we can check integrity first and save ~~de~~cryption if it fails.

Note: encrypted text + encryption key also fixes / identifies the plain text uniquely.

Second solution: first encrypt

then authenticate this + the keys.

ESP does that in a weak sense: the authenticated part includes the SPI, yet not the keys itself.

Encryption and authentication algorithms

18.06.07

4

RFC 4305

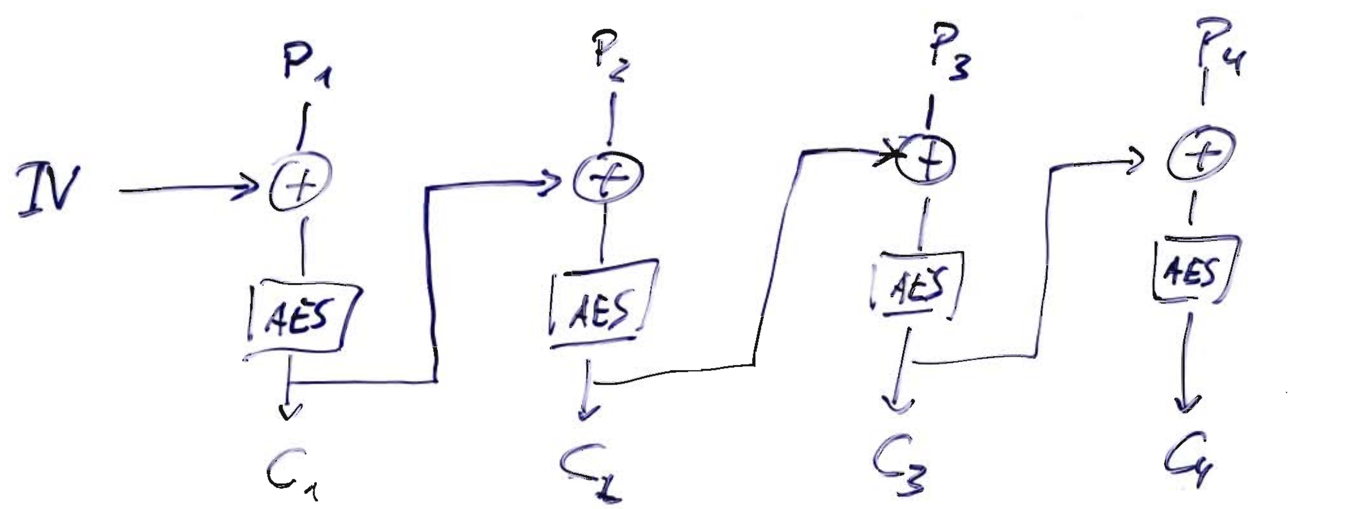
Encryption algorithms:

MUST	NULL	
MUST-	Triple DES-CBC	(RFC 2451)
SHOULD+	AES-CBC with 128 bit key	(RFC 3602)
SHOULD	AES-CTR	(RFC 3686)
SHOULD NOT	DES-CBC	(RFC 2405)

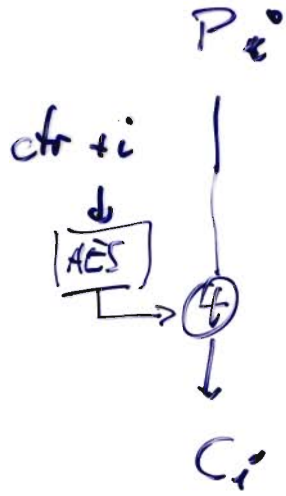
authentication algorithms

MUST	HMAC-SHA1-96	(RFC 2404)
MUST	NULL	
SHOULD+	AES-XCBC-MAC-96	
MAY	HMAC-MD5-96	

RFC 3602 AES-CBC encryption



RFC 3686 AES-CTR ~~mode~~ encryption



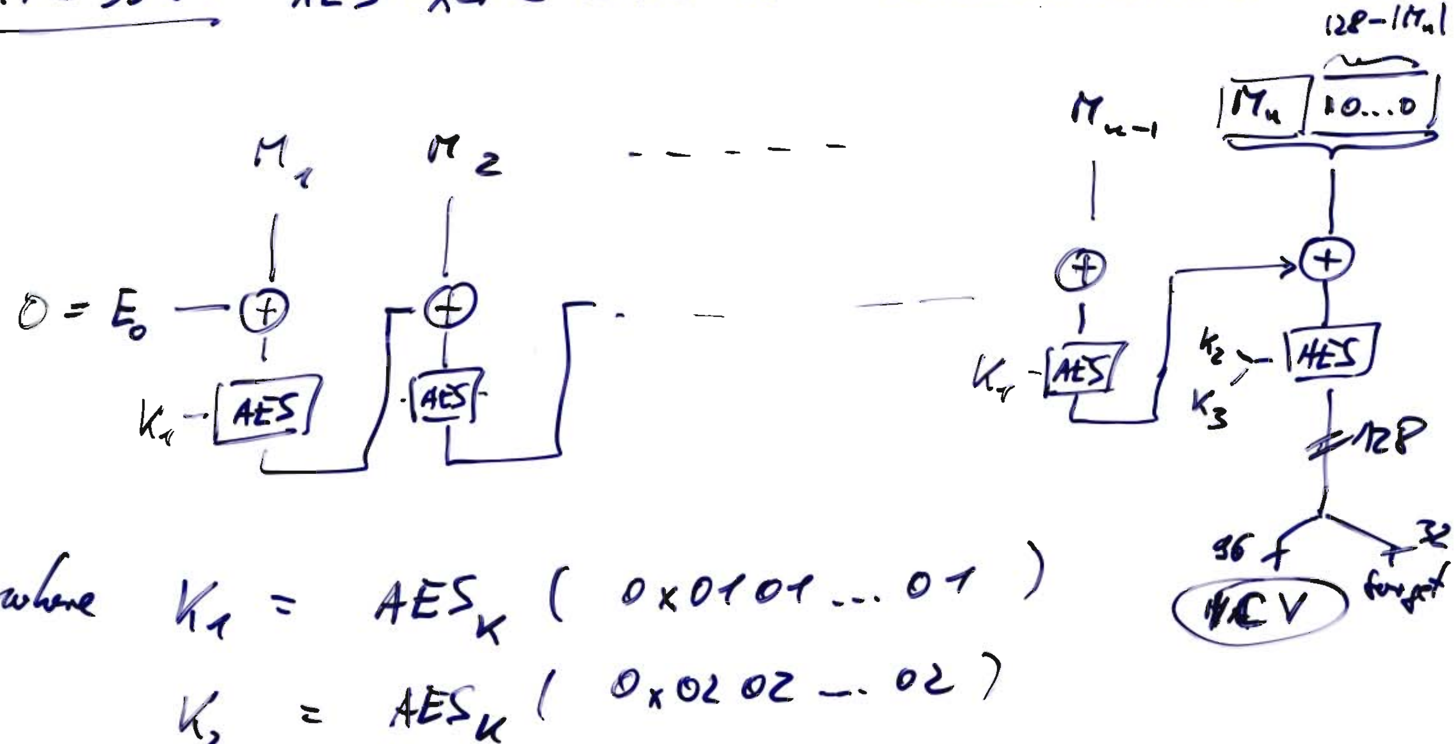
where

$$ctr = \overbrace{NONCE}^{32} \,\|\, \overbrace{IV}^{64} \,\|\, \overbrace{0}^{32}$$

NONCE: ass. with SA
IV: ass. with a packet

Advantage:
- much easier to resync
- don't need to decrypt in order (sufficient to know the position i)

RFC 3566 AES-XCBC-MAC-96 authentication



where $K_1 = AES_K(0x0101\ldots01)$

$K_2 = AES_K(0x0202\ldots02)$

$K_3 = AES_K(0x0303\ldots03)$

where K_2 is used when there is no padding and K_3 otherwise.

18.6.07

6

Ask yourself: what would happen if somebody tries to change the message? Can the attacker guess the same signature (ICV)?

⇝ Last key specifically vulnerable.

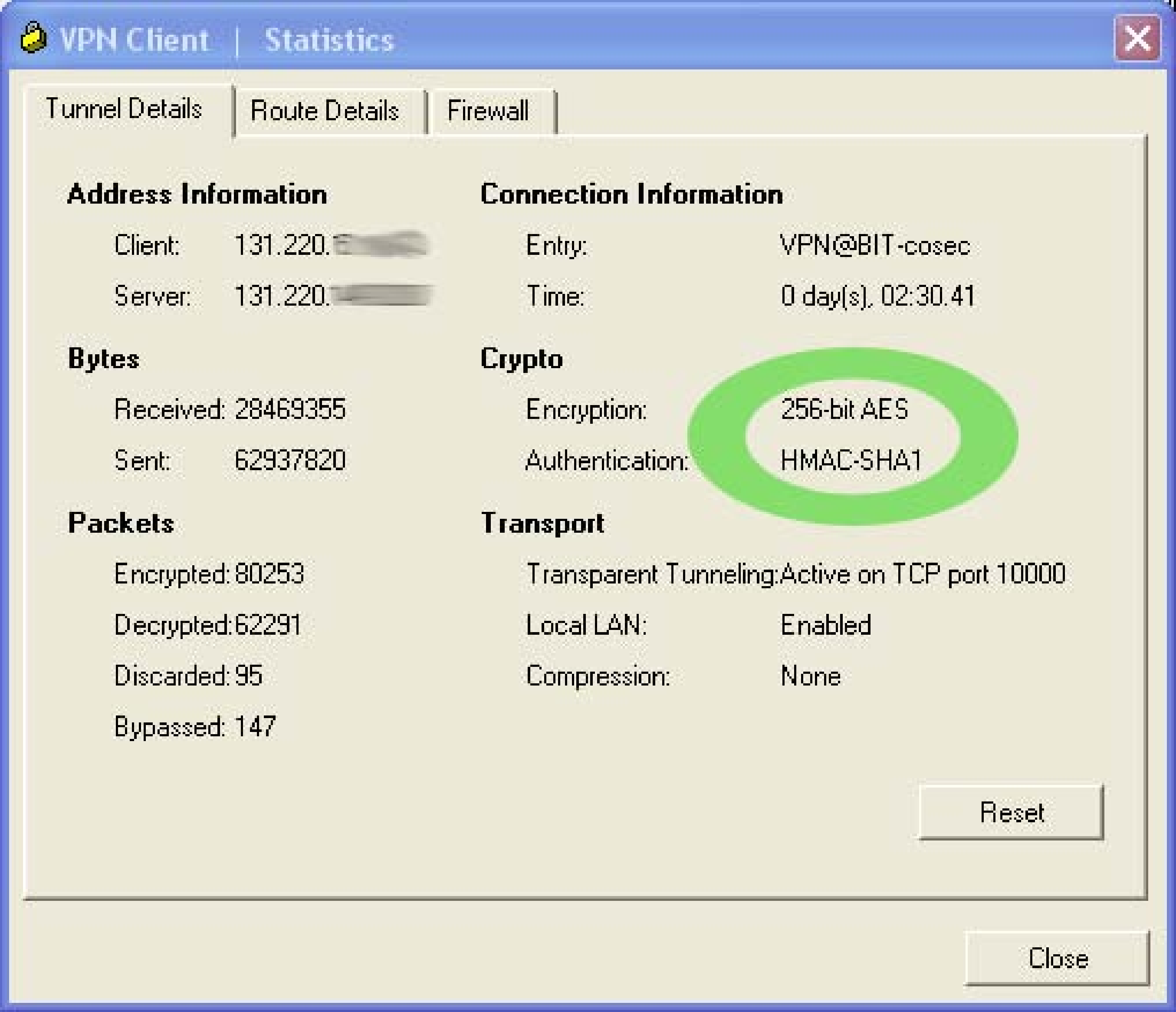
RFC 2404 / 2104   HMAC-SHA1-96

SHA1(K ⊕ opad, SHA1(K ⊕ ipad, msg ~~|| padding~~))

0x36 repeated   0x5C repeated

take the first 96 bits of this.

These lecture notes contain a description of SHA1 on an earlier page → see there.


VPN Client | Statistics
Tunnel Details
Route Details
Firewall
Address Information
Client: 131.220.
Server: 131.220.
Connection Information
Entry: VPN@BIT-cosec
Time: 0 day(s), 02:30.41
Bytes
Received: 28469355
Sent: 62937820
Crypto
Encryption: 256-bit AES
Authentication: HMAC-SHA1
Packets
Encrypted: 80253
Decrypted: 62291
Discarded: 95
Bypassed: 147
Transport
Transparent Tunneling: Active on TCP port 10000
Local LAN: Enabled
Compression: None
Reset
Close

20.6.07

(1)

Internet Key Exchange version 2

Initial contact comprises 4 messages, only the first two are not encrypted.

Initiator — **Responder**

$$\text{Hdr, SAi1, KEi, Ni} \xrightarrow{\ldots g^a \ldots}$$

- Hdr: Header (type of protocol, ..., SPI)
- SAi1: initiator's SA proposals
  - DH group
  - encr.
  - auth
- KEi: initiator's DH value in predicted group (g^a)
- Ni: initiator's Nonce

We saw: $\mathbb{Z}_p^\times \ni g$

Group 1: 768-Bit MODP

$$p = 2^{768} - 2^{704} - 1 + 2^{64}\left(\lfloor 2^{638}\pi \rfloor + 149686\right)$$

$$g = 2$$

(Too small in practice only for DES-CBC)

Group 2: 1024-bit MODP

$$p = 2^{1024} - 2^{960} - 1 + 2^{64}\left(\lfloor 2^{894}\pi \rfloor + 129093\right)$$

$$g = 2$$

$$\xleftarrow{\ldots g^b \ldots} \text{Hdr, SAr1, KEr, Nr [, CERTREQ]}$$

- SAr1: responder's choice for SA (DH group, encr, auth)
- KEr: responder's DH value (g^b)
- Nr: responder's Nonce

20.6.07
(2)

Now both parties have g^{ab}.
and derive keys from it:

$$SK = (SK\_e, SK\_a)$$

for each direction.

Hdr, SK{ ID_i, [$CERT_i$,] [$CERTREQ$,] [ID_r,] AUTH, SAi2, TSi, TSr }

SPIi,
SPIr
...

initiator's identification (e.g. IP #)

initiator's certificate (X.509) + proof that initiator knows the private key to it...

asks responder to authenticate

selects one of possibly many responders managed by the contacted server.

this authenticates the first message

may start CREATE_CHILD_SA

proposed traffic selectors define firewall like rules: which IPs and ports are allowed through this SA.

Hdr, SK{ ID_r, [$CERT$,] AUTH, SAr2, TSi, TSr }

Summary

20.6.07
(3)

$\ldots g^a \ldots$

$\longrightarrow$

$\ldots g^b \ldots$

$\longleftarrow$

may contain CERT

$SK\{ID_i \ldots\}$

$\longrightarrow$

$SK\{ID_r \ldots\}$

$\longleftarrow$

This establishes IKE_SA.

All further messages are protected by keys derived from this (and possible further DH key exchanges; REKEY-ings)

Rekeying possible at any time by any partner.

Any further exchange consists of a request and a response.

CREATE_CHILD_SA

$Hdr, SK\{[N,]\ SA, N_i, [KE_i,]\ [TS_i, TS_r]\}$

opt. REKEY_SA identifying the SA being rekeyed (maybe a CHILD_SA or IKE_SA itself)

proposed algo.

Nonce

DH value in predicted group

opt. proposed traffic selectors

$\longrightarrow$

$Hdr, SK\{SA, N_r, [KE_r,][TS_i, TS_r]\}$

$\longleftarrow$

20.6.07
4

If the predicted group is not the chosen one an informational msg with the chose group is send back and the initiator has to retry – with same proposals!.

INFORMATIONAL exchanges

... for notifications (error msg),
delete,
configuration.

→ always msg & response.
so empty msg is interpreted as "Are you still there?".

→ always protected under IKE_SA

Eg. if a connection should be closed:
ESP SA, AH SA exist in pairs
→ both have to be closed

close incoming (ESP) SA $\xrightarrow{\text{DELETE}}$ close outgoing SA

close outgoing SA $\xleftarrow{\text{DELETE}}$ close incoming SA

20.6.07

(5)

Node crash or similar

→ incoming SPI unknown

→ if another IKE_SA exists with that sender

→ may send informational msg using that.

else → may send unprotected notification.

The other node MUST NOT trust this kind of answers. Instead such half closed SAs are considerd anomalous, and the other node should retry some times

The other node sends empty info msg

If the node responds SA STILL ALIVE

If the node does not respond in a dozen (or so) attempts: then only assume SA is dead and close it.

Never delete an SA because of unprotected information.

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

IP header	IP payload

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

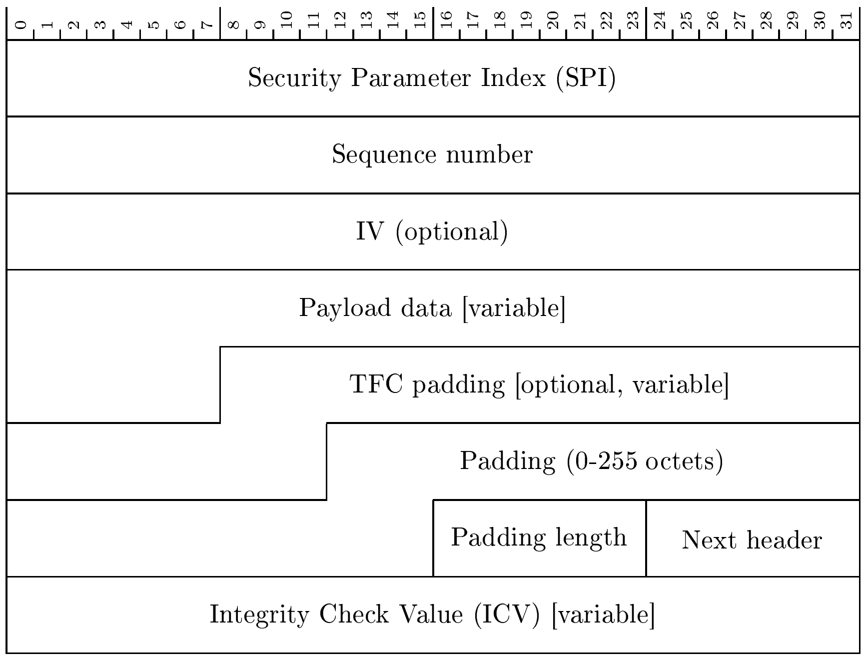
new IP header	IPsec header	IP header	IP payload

In transport mode, only the IPsec header is added:

IP header	IPsec header	IP payload

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:

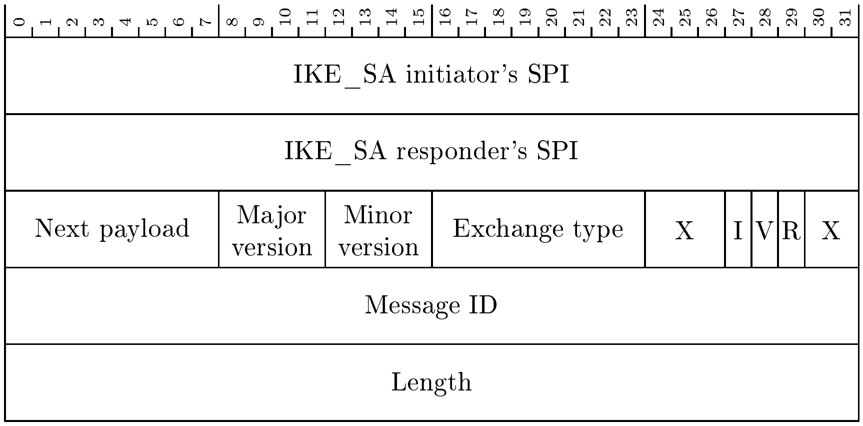


The security parameter index identifies the SA and thus all necessary algorithms and key material. To create the secured packet from the original one, it is first padded. Padding is used to enlarge the data length to a multiple of a block size that might be associated with the encryption. Traffic flow 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 authenticion 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.)

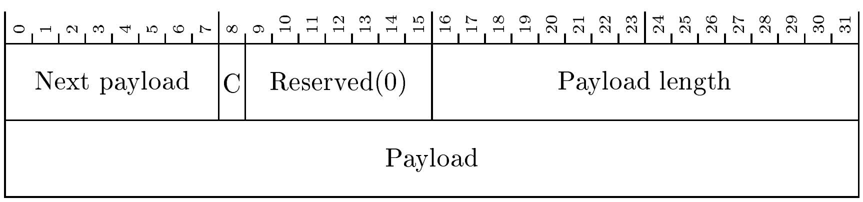
2. Internet key exchange (version 2)

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

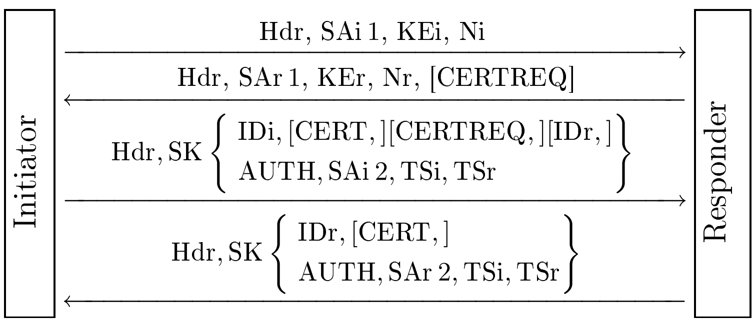
Exchange type	Value
Reserved	0-33
IKE_SA_INIT	34
IKE_AUTH	35
CREATE_CHILD_SA	36
INFORMATIONAL	37
Reserved to IANA	38-239
Reserved for private use	240-255



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 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 $\mathrm{KEi} = g^a$.
   Choose a nonce Ni. $\xrightarrow{\text{Hdr, SAi 1, KEi, Ni}}$
2. Choose SAr1 from SAi1 unless no variant is supported.

Compute KEr $= g^b$ if the group was guessed correctly. (Otherwise send:

$$\mathrm{Hdr}, \mathrm{N}(\texttt{INVALID\_KE\_PAYLOAD}, \mathrm{group})$$

.)
Choose a nonce Nr.

$$\xleftarrow{\mathrm{Hdr}, \mathrm{SAr}\,1, \mathrm{KEr}, \mathrm{Nr}, [\mathrm{CERTREQ}]}$$

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.

$$\begin{aligned} &\mathrm{SKEYSEED} = \mathrm{prf}(Ni|Nr, g^{ab}),\\ &\mathrm{SK\_d}|\mathrm{SK\_ai}|\mathrm{SK\_ar}|\mathrm{SK\_ei}|\mathrm{SK\_er}|\mathrm{SK\_pi}|\mathrm{SK\_pr}\\ &= \mathrm{prf+}(\mathrm{SKEYSEED}, \mathrm{Ni} \,|\, \mathrm{Nr} \,|\, \mathrm{SPIi} \,|\, \mathrm{SPIr}) \end{aligned}$$

where $\mathrm{prf+}(K, S) = T_1|T_2|T_3|\ldots$, and $T_1 = \mathrm{prf}(K, S|0x01)$, $T_i = \mathrm{prf}(K, T_{i-1}|S|i)$ 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), shard key message integrity code (2), or DSS digital signature (3).

0–7	8	9–15	16–31
Next payload	C	Reserved(0)	Payload length
Auth method	Reserved		
Authentication data			

The initiator starts to negotiate a child SA in SAi 2 with proposed traffic selectors TSi, TSr.

$$\xrightarrow{\mathrm{Hdr}, \mathrm{SK}\left\{\begin{matrix}\mathrm{IDi}, [\mathrm{CERT},]\\ [\mathrm{CERTREQ},]\\ [\mathrm{IDr},]\\ \mathrm{AUTH}, \mathrm{SAi}\,2,\\ \mathrm{TSi}, \mathrm{TSr}\end{matrix}\right\}}$$

5. The responder sends its identity IDr, certificate(s). He computes an authentication AUTH for the entire second message concatenated with the initiator's nonce Ni and the value prf(SK_pr, IDr). Further he supplies the answer SAr 2 to the child SA creation and sends the accepted traffic selectors TSi, TSr.

$$\xleftarrow{\mathrm{Hdr,SK}\left\{\begin{array}{l}\mathrm{IDr},[\mathrm{CERT},]\\ \mathrm{AUTH},\mathrm{SAr}\,2,\\ \mathrm{TSi},\mathrm{TSr}\end{array}\right\}}$$

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:

$$\mathrm{KEYMAT} = \mathrm{prf+}(\mathrm{SK\_d}, \mathrm{Ni} \mid \mathrm{Nr})$$

2.2. Creating additional child SAs. Further childs can be created under this IKE_SA using a CREATE_CHILD_SA exhange:

$$\boxed{\text{Initiator}}\quad\begin{array}{c}\xrightarrow{\mathrm{Hdr,SK}\left\{\begin{array}{l}[\mathrm{N},]\,\mathrm{SAi}\,2,\mathrm{Ni},[\mathrm{KEi},]\\ [\mathrm{TSi},\mathrm{TSr}]\end{array}\right\}}\\ \xleftarrow{\mathrm{Hdr,SK}\left\{\begin{array}{l}\mathrm{SAr}\,2,\mathrm{Nr},[\mathrm{KEr},]]\\ [\mathrm{TSi},\mathrm{TSr}]\end{array}\right\}}\end{array}\quad\boxed{\text{Responder}}$$

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.

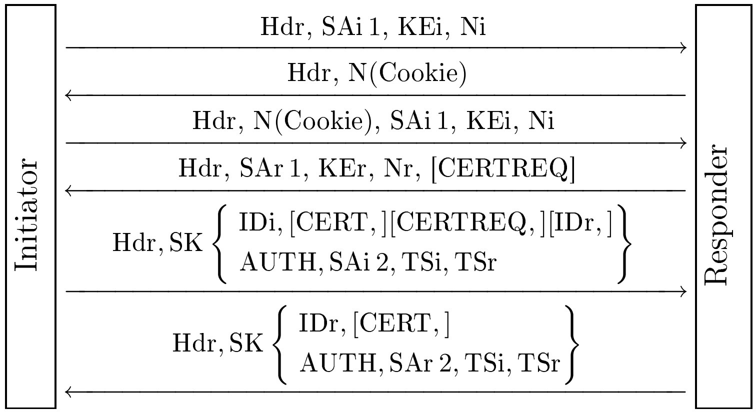
In a CREATE_CHILD_SA exchange including an optional Diffie-Hellman exchange new keying material uses also the new Diffie-Hellman key g^{ir}, it is concatenated left to the nonces. (Though the Diffie-Hellman key exchange is optional, it is recommended to either used it or at least to limit the number 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 Ni, her IP IPi, her security parameter index SPIi and some secret:

$$\mathrm{Cookie} = \mathrm{verID} \mid \mathrm{hash}(\mathrm{Ni}, \mathrm{IPi}, \mathrm{SPIi}, \mathrm{secret}_{\mathrm{verID}})$$

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.

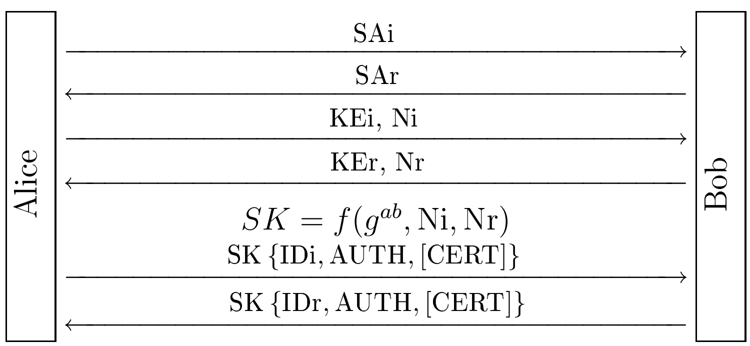
3. IKE version 1

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

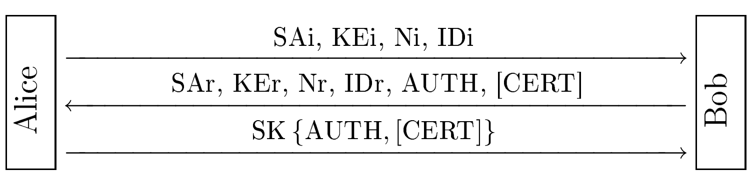
- public signature keys,
- public encryption keys, originial protocol,
- public encryption keys, revised protocol, or
- a pre-shared secret.

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 Kaufmann *et al.* 2002.)

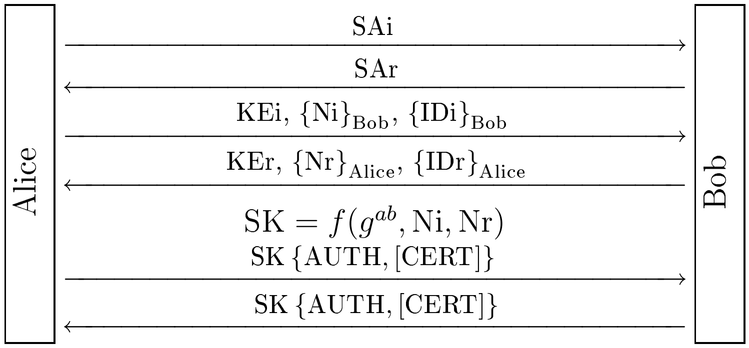
3.1. Main mode, public signature keys.



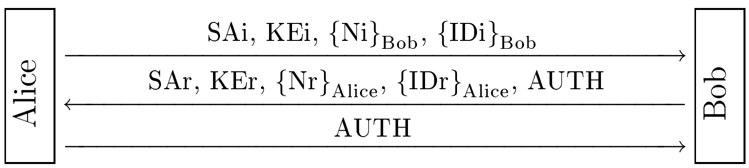
3.2. Aggressive mode, public signature keys.



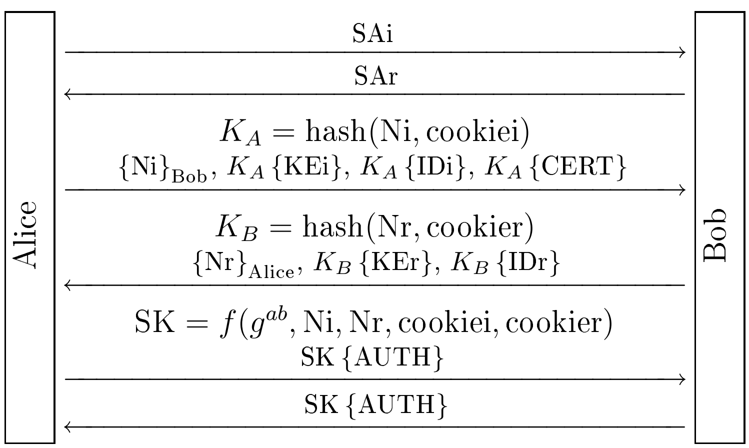
3.3. Main mode, public encryption keys, original protocol.



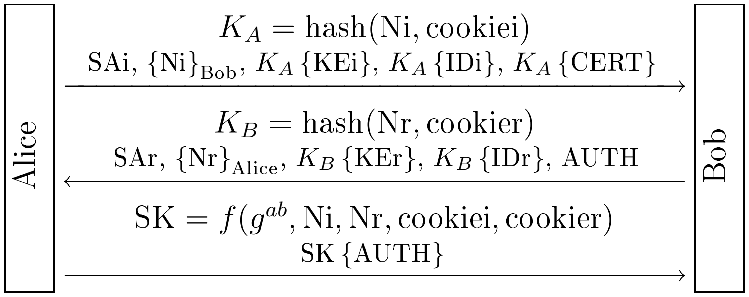
3.4. Aggressive mode, public encryption keys, original protocol.



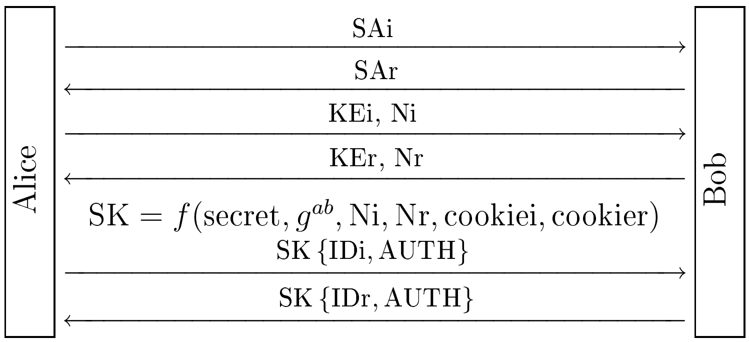
3.5. Main mode, public encryption keys, revised protocol.



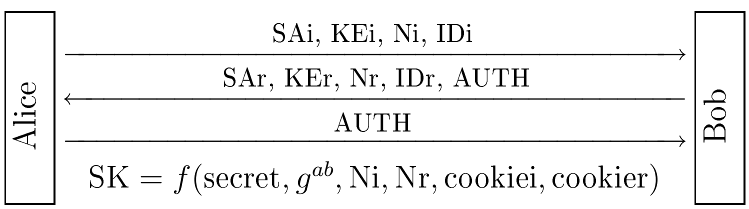
3.6. Aggressive mode, public encryption keys, original protocol.



3.7. Main mode, pre-shared secret.



3.8. Aggressive mode, pre-shared secret.



References

Charlie Kaufmann, Radia Perlman & Mike Speciner (2002). *Network Security*. Prentice-Hall, Inc., New Jersey. ISBN 0-13-046019-2.

Michael Nüsken
b-it, Bonn, Germany

Sof I
25.6.07
7

Have seen: IPsec — ESP / AH

tunnel / transport

IKEv2
- IKE_SA_INIT (4 msgs)
- CREATE_CHILD_SA (2 msgs)
- INFORMATIONAL
  - DELETE SA
  - ⋮

History of IKE:

PHOTURIS   SKIP

NSA-proposal: ISAKMP
- only framework
- ruled out both candidates

→ IETF could take up the development.

OAKLEY, SKEME ... (new drafts)

IKE puts → into ISAKMP.

Problem:
- no clear design
- too many variants
- ≥ 150 pages, ≥ 3 RFC, partially very unclear & difficult to read.

IKEv2 : clear, simple rules

[SofI 25.6.07 ②]

- any request gets a response

- initial exchange: 1 way, 4 msgs. („IKEv1 „phase 1")

- create child SA = 2 msgs. („phase 2")

- ~~prior variants~~
- functionality of all the IKEv1 variants is still there but now as options or additional request.

e.g. $\xrightarrow{\text{Hdr, SK\{... CERTREQ ...\}}}$

$\xleftarrow{\text{Hdr, SK\{ ... CERT ...\}}}$

Fact-finding committees

SofI
25.6.07
(3)

(1) IKEv1 aggressive mode

(2) IKEv1 main mode

(3) IKEv2

Look at: PROs & CONs!?

Specific questions:

(0) SECURITY, SECURITY, SECURITY.

(1) Session key agreement
- How long? Random?
- Do both parties contribute to it?
- Man in the middle

(2) Perfect forward security
- Can an attacker given the long-term secrets afterwards and all messages decrypt?

Escrow foilage
- Is the conversation secret even if the long-term secrets are known to the attacker in advance?

SofI
25.6.07
(4)

(3) Denial of Service

(4) Endpoint identifier hiding

- Does an eavesdropper get info about identities?
- ~~Does~~ Can an active attacker get identification information from initiator (client) or the responder (server).

(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 ?

(7) Stream protection
How is a logical data stream protected? → confidential
→ authentic in its entirety

(8) Negotiating crypto parameters
→ Pros
→ Cons

main mode

pro

public signature keys

- nonces: prevent replay attacks / life partner reass.
- active attacks (i.e., man-in-the-middle) would modify message 5 or 6 → detectable
- certificates → plausible deniability

pre-shared secret

- man-in-the-middle / active attacks do not work anymore: key depends on preshared secret
- certificates are no longer necessary → simpler infrastructure

con

- DoS

public signature keys

- no stream protection (no sequence numbers)
- active attacks: could reveal certificate content → confidentiality partially compromized (no endpoint identifier hiding, esp. for client)
- crypto parameter exchange: CP could be modified to enforce usage of a weak algorithm
- if attacker knows a and b: no perfect forward security, all messages can be decrypted (same for escrow foilage)

pre-shared secret

- shared secret needs to be exchanged in a secure way (i.e., different channel)
- DoS still possible

IKEv1 agressive mode

PROs

- less #messages => faster
- authentication with a pre-shared key allows for a wide range of identifiers (not only IP addresses)
- replay not possible, because we use nonces
- revised protocol (public key encryption): cookies for counter measuring DOS

CONs

- authentication without using a session key
- not all modes hide the identities (4)
- original protocol (pke): does not use cookies

IKEv2

0) see below ↓

1) key exchange: two Diffie-Hellman groups / Size of group Min 768 bits.

Randomness. : Pseudo-Random function

* IKEv2 has one Single four-message exchange

* no entity-in-the-middle with Certificate.

3) * it does, but better than Version(1) in terms of DoS attack.

*InSecure Because

4) *An outside Attacker can't get any info when Listening to a conversation

An active attacker can REQ first & get the certificate

5) Due to randomness of b the chance to be able to reuse an old conversation is minute

2) Since a & b are short-term-secrets: No

2.7.07
(1)

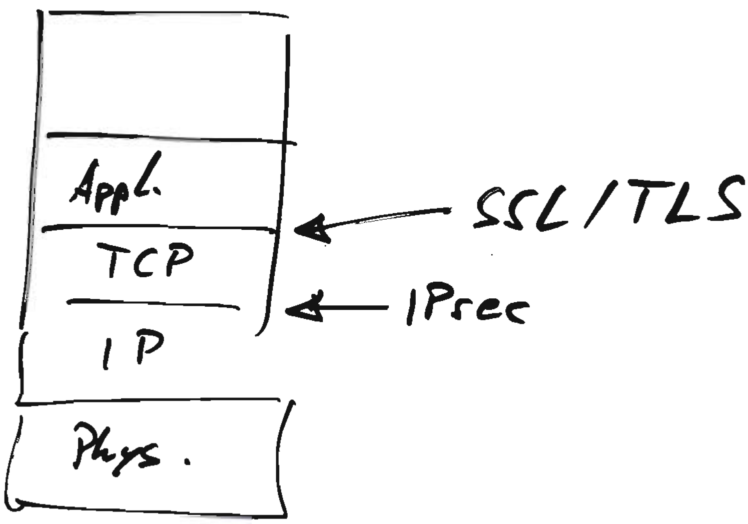
SSL Secure Socket Layer

TLS Transport Layer Security.

First steps: 1994 (?) Netscape

...

Decision:



Reasons:

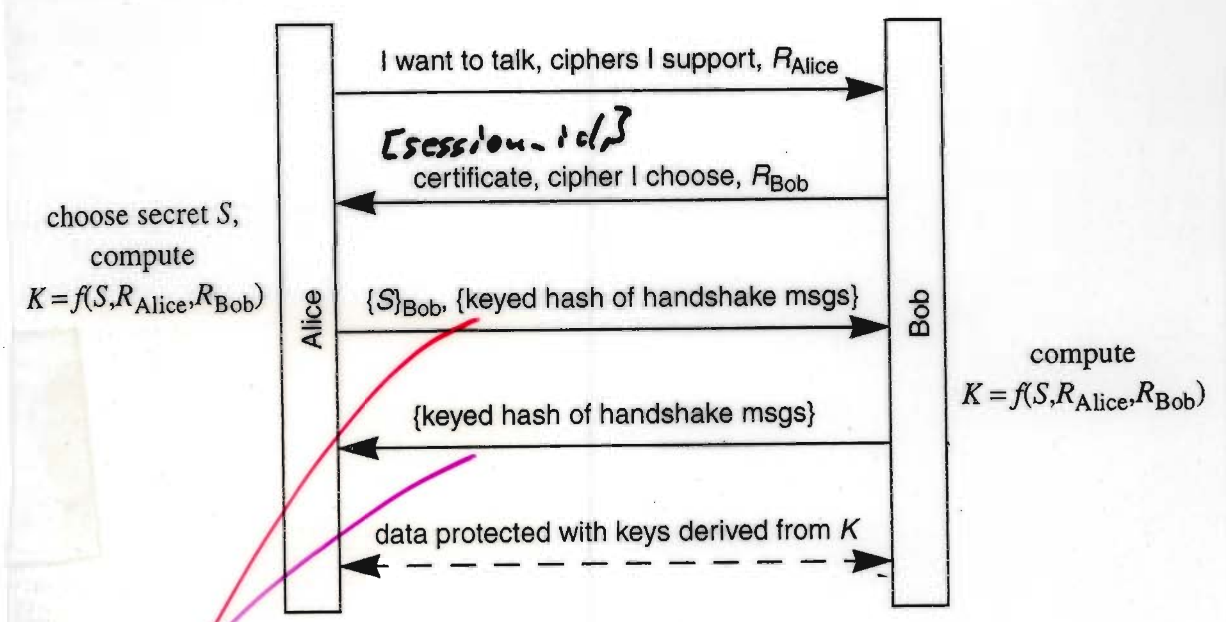
- Wanted fast, easily embeddable solution.
- Should link application (Browser) to application (webserver) rather than station to station
- Encryption maybe, but definitely authentication – of server and – optionally of client needed.

IPsec was not there yet.

3.7.07
(2)

'Same' shape

Initial handshake ($\hat{=}$ IKE_SA_INIT)



msg 1

msg 2

msg 3

msg 4

Protocol 19-1. (simplified) SSLv3/TLS

S = pre master key (bit)

R_{Alice}/R_{Bob} = random numbers(bit)

K = master key (384 bits)

hash({ 'CLNT' / 'client finished' , K, msg 1&2)

(as selected by Bob in msg 2

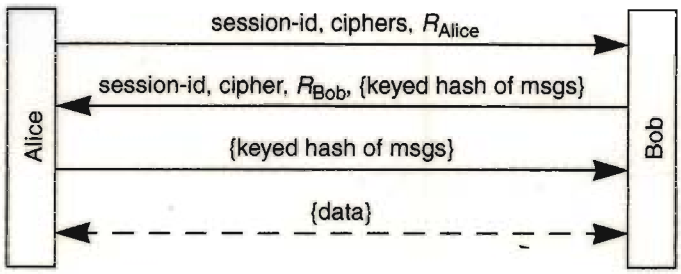
depends on version (SSL v2, SSL v3, TLS 1.0, TLS 1.1)

hash ({ 'SRVR' / 'server finished' , K, msg 1&2 (&3?))

From K we derive: 2 encryption keys
2 authentication/integrity keys
2 IV (for CBC mode...)

2.7.07
3

If a session-id was fixed, another TCP session may use the same keys using the 'Session resumption'



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

Further purpose: this allows to upgrade to higher security ciphers.

[Background: US export restriction on any cryptography using more than 40-bit keys in the symmetric scenario or more than 512 bit RSA... }

That restriction has been dropped in the meantime...]

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

Another reasons may be that Bobs policies have changed...

Why?

How does Alice know?

Encryption & authentication in SSL

2.7.07
4

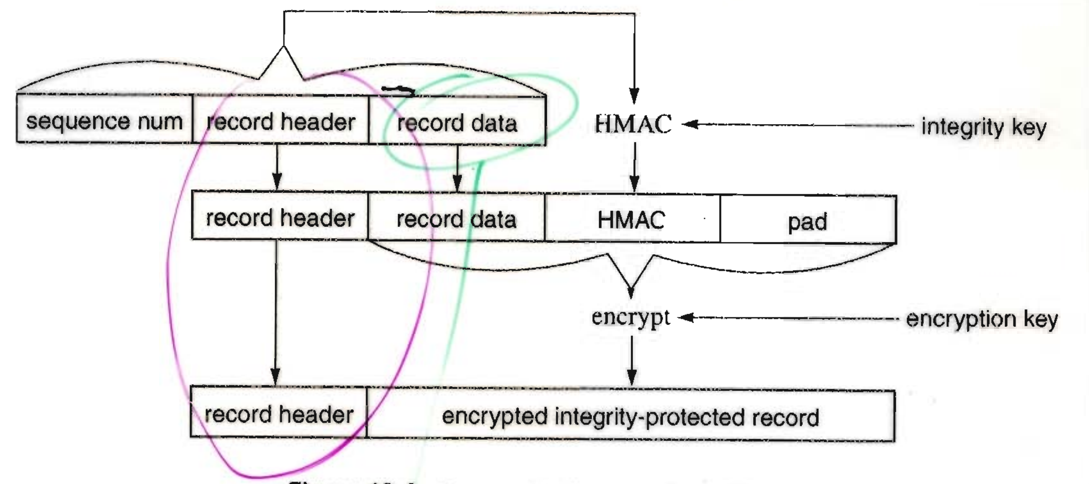


Figure 19-4. Cryptographically protected record format

payload

from / to / ... ... length

$\leq 2^{14} + \ldots$

$\approx 16kB.$

much longer than in IPsec

SSL/TLS does not have to care about fragmentation, resending, ...

Note: shape of the protected record is:

$$\text{Hdr}, \quad ENC_{K_e}(\, m \mid MAC_{K_a}(m) \mid \text{pad} \,)$$

possible ciphers

2.7.07
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
TLS_RSA_WITH_3DES_EDE_CBC_SHA	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
DES40_CBC	Block	5	8	8	8
DES_CBC	Block	8	8	8	8
3DES_EDE_CBC	Block	24	24	8	8

Our questions?

- Session key agreement:
  → need PKI to verify server identity
  With https or email over SSL/TLS browsers and email client are usually delivered with built-in root certificates, so that we can easily verify certificates going to one of those.

And it's there.

- Perfect forward security / Escrow attack.

  SSL seems to be vulnerable to this attack

  If S is used as in this top-level view, we simply decrypt $\{S\}_{Bob}$ and derive all further keys as necessary ...

- Denial of Service
  - No extra protection
  - and this is not necessary because lower layers will care for this.

- Endpoint identifier hiding
  - Server id is not hidden.
  - Client id is hidden as long as it closely inspects and verifies the server's certificate.

- Live partner reassurance
  → Message id's and random numbers (used as nonces) protect from that

- Deniability?
  - Alice cannot prove that Bob talked to her.
  - The other way round: with login/password : NO!

- Stream protection

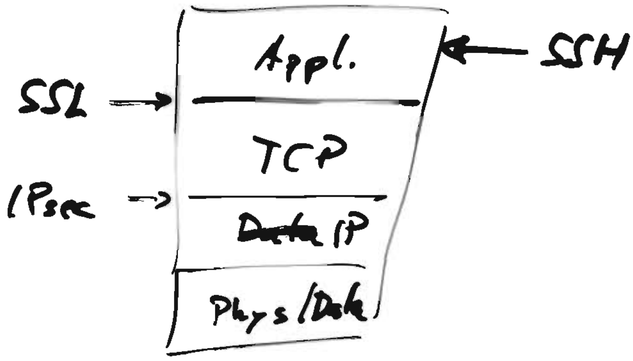
  The keys and their use guarantee that all records belong to the same session

- Negotiate crypto parameters

  - Yes, they are.
  - Downgrade? · In the first place: yes, but have to forge msg 4.
    - Certificates may contain upgrade information allowing the client to resume the session with better ciphers.
  - Use of version #s.

    Apart from SSLv2 we have control here.

SSH

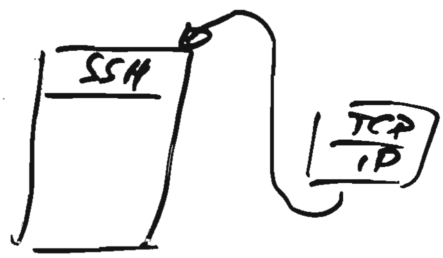


1995 Tatu Ylönen
started as a secure replacement of
remote terminals (telnet, rsh, ...)

1996 ssh 2

1999 open SSH ↔ ssh techa

Now:
- sftp, scp : files transfer
- forward x11
- tunnel TCP/IP
- ⋮



Identification: · RSA certificate
(rather than X.509 certificate
or similar)

Key exchange: · DH

Encryption: · AES128 but many others
possible

Authentication: · HMAC SHA1 but other possible

4.7.07

2

aes128-ctr	RECOMMENDED	AES (Rijndael) in SDCTR mode, with 128-bit key
aes192-ctr	RECOMMENDED	AES with 192-bit key
aes256-ctr	RECOMMENDED	AES with 256-bit key
3des-ctr	RECOMMENDED	Three-key 3DES in SDCTR mode
blowfish-ctr	OPTIONAL	Blowfish in SDCTR mode
twofish128-ctr	OPTIONAL	Twofish in SDCTR mode, with 128-bit key
twofish192-ctr	OPTIONAL	Twofish with 192-bit key
twofish256-ctr	OPTIONAL	Twofish with 256-bit key
serpent128-ctr	OPTIONAL	Serpent in SDCTR mode, with 128-bit key
serpent192-ctr	OPTIONAL	Serpent with 192-bit key
serpent256-ctr	OPTIONAL	Serpent with 256-bit key
idea-ctr	OPTIONAL	IDEA in SDCTR mode
cast128-ctr	OPTIONAL	CAST-128 in SDCTR mode, with 128-bit key
3des-cbc	REQUIRED	three-key 3DES in CBC mode
blowfish-cbc	OPTIONAL	Blowfish in CBC mode
twofish256-cbc	OPTIONAL	Twofish in CBC mode, with a 256-bit key
twofish-cbc	OPTIONAL	alias for "twofish256-cbc" (this is being retained for historical reasons)
twofish192-cbc	OPTIONAL	Twofish with a 192-bit key
twofish128-cbc	OPTIONAL	Twofish with a 128-bit key
aes256-cbc	OPTIONAL	AES in CBC mode, with a 256-bit key
aes192-cbc	OPTIONAL	AES with a 192-bit key
aes128-cbc	RECOMMENDED	AES with a 128-bit key
serpent256-cbc	OPTIONAL	Serpent in CBC mode, with a 256-bit key
serpent192-cbc	OPTIONAL	Serpent with a 192-bit key
serpent128-cbc	OPTIONAL	Serpent with a 128-bit key
arcfour	OPTIONAL	the ARCFOUR stream cipher with a 128-bit key
idea-cbc	OPTIONAL	IDEA in CBC mode
cast128-cbc	OPTIONAL	CAST-128 in CBC mode
none	OPTIONAL	no encryption; NOT RECOMMENDED

hmac-sha1	REQUIRED	HMAC-SHA1 (digest length = key length = 20)
hmac-sha1-96	RECOMMENDED	first 96 bits of HMAC-SHA1 (digest length = 12, key length = 20)
hmac-md5	OPTIONAL	HMAC-MD5 (digest length = key length = 16)
hmac-md5-96	OPTIONAL	first 96 bits of HMAC-MD5 (digest length = 12, key length = 16)
none	OPTIONAL	no MAC; NOT RECOMMENDED

diffie-hellman-group1-sha1 MUST
Oakley Group 2 [RFC2409] (1024-bit MODP Group)

diffie-hellman-group14-sha1 MUST
Oakley Group 14 [RFC3526] (2048-bit MODP Group)

Public Key Infrastructure

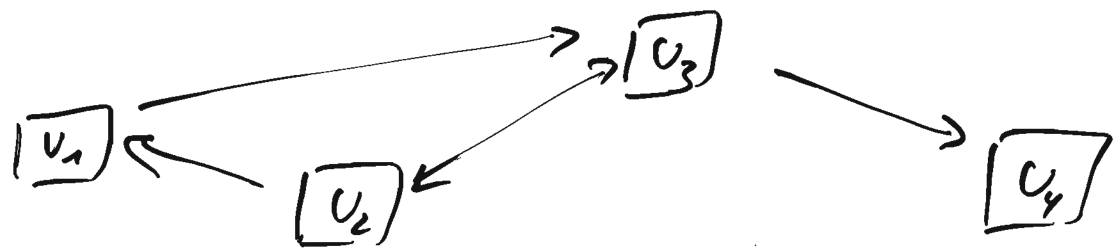
4.7.07 ③

- manage certificates → TRUST
- Distribute certificates → AVAILABILITY.

Trust models

- **Anarchy model, web of trust**

  Users signe others key and manage 'key rings'



  (as in PGP)

- **Monopoly model**

  One world CA

  Pros: • Mathematically appealing
  • Simple
  • CA's ~~k~~ public key certificate → ease of use

  Cons: • High load on CA: Identify users?
  • Very critical, only one point to break
  • High cost
  • High concentration of power (selecting users to certify)

4.7.07
5

· High danger of treachers, sabotage...

Monopoly model + registration authorities (RA)

→ solves bottleneck
but still high risk

Oligarchy

CA_1 CA_2 CA_3 ....

→ even less secure because even one compromised CA is a problem (need several certificates to resolve it)

→ CAs trusted by vendor of your software

→ might be easy to introduce a bogus CA in such a list

→ In practice checking all these root CAs is difficult to impossible.

→ **User's do not understand**

~ PSYCHOLOGY

Psychology

```
Warning. This was signed by an unknown CA.
Would you like to accept the certificate anyway?

[OK]


Would you like to accept this certificate without
being asked in the future?

[OK]


Would you like to always accept certificates from
the CA that issued that certificate?

[OK]


Would you like to always accept certificates from
any CA?

[OK]

(User thinks: Grrrr.... isn't it enough by now?)


Since you're willing to trust anyone for anything,
would you like me to make random edits to the files
on your hard drive without bothering you
with a pop-up box?

[OK]

(User thinks: Gosh, another box.... No more pop-ups? YES!)
```

Notes added in proof

- PGP web of trust reveals social network:
  Who knows me? Who do I know?
  → Ask the key server.
- Ask "Who generates the private key?"
  PGP: Your own computer.
  Thawte (a CA run by AOL): The CA does!

4.7.07

(6)

Organization

(1) Go to the public course this afternoon on

intrusion detection

limits of

Today 16^{00}
Römerstr. 164
Hörsaal C

or alternatively inform yourself on the topic

(2) Next monday we discuss that.

(3) No course next wednesday