COSIC

KU LEUVEN

Cryptographic algorithms

Prof. Bart Preneel
COSIC KU Leuven and imec
Bart.Preneel(at)esatDOTkuleuven.be
http://homes.esat.kuleuven.be/~preneel

November 2018

Goal + Outline

- Understanding role of cryptography
- Understanding principles behind the most important cryptographic algorithms

- Cryptology: concepts
- Cryptology: algorithms
  – symmetric algorithms for confidentiality
  – symmetric algorithms for data authentication
  – public-key cryptology

2

Definitions

1970s

Confidentiality
Integrity
Availability

	data	entities
confidentiality	encryption	anonymity
authentication	data authentication	identification

Authorisation

Non-repudiation of origin, receipt

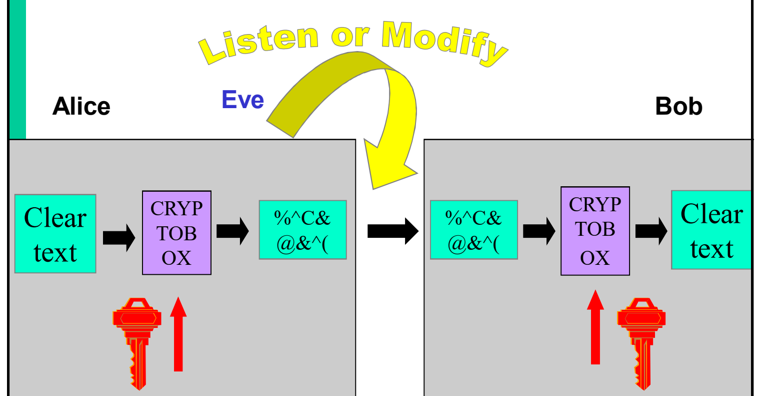
Contract signing

Notarisation and Timestamping

Don't use the word authentication without defining it

3

Cryptology: basic principles

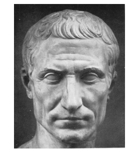


Symmetric cryptology: confidentiality

- old cipher systems:
  - transposition, substitution, rotor machines
- the opponent and her power
- the Vernam scheme
- DES and triple-DES
- AES
- RC4

Old cipher systems (pre 1900)

- Caesar cipher: shift letters over k positions in the alphabet (k is the secret key)



THIS IS THE CAESAR CIPHER

WKLV LV WKH FDHVDU FLSKHU

- Julius Caesar never changed his key (k=3).

6

Cryptanalysis example:

```
TIPGK RERCP JZJZJ WLE      GVCTX EREPC WMWMW JYR
UJQHL SFSDQ KAKAK XMF      HWDUY FSFQD XNXNX KZS
VKRIM TGTER LBLBL YNG      IXEVZ GTGRE YOYOY LAT
WLSJN UHUFS MCMCM ZOH      JYFWA HUHSF ZPZPZ MBU
XDTKO VOVGT NDNDN API      KZGXB IVITG AQAQA NCV
YNULP WKWHU OEOEO BQJ      LAHYC JWJUH BRBRB ODW
ZOVMQ XKXIV PFPFP CRK      MBIZD KXKVI CSCSC PEX
APWNR YLYJW QGQGQ DSL      NCJAE LYLWJ DTDTD QFY
BQXOS ZMXKX RHRHR ETM      ODKBF MZMXK EUEUE RGZ
CRYPT ANALY SISIS FUN      PELCG NANYL FVFVF SHA
DSZQU BOBMZ TJTJT GVO      QFMDH OBOZM GWGWG TIB
ETARV CPCNA UKUKU HWP      RGNEI PCPAN HXHXH UJC
FUBSW DQDOB VLVLV IXQ      SHOFJ QDQBO IYIYI VKD
```

Plaintext? k = 17

7

Old cipher systems (pre 1900) (2)

- Substitutions
  - ABCDEFGHIJKLMNOPQRSTUVWXYZ
  - MZNJSOAXFQGYKHLUCTDVWBIRPE

! Easy to break using statistical techniques

- Transpositions

TRANS	OIPSR
POSIT	NOTNT
IONS	OSAI

8

Security

- there are n! different substitutions on an alphabet with n letters
- there are n! different transpositions of n letters
- n=26: n!=403291461126605635584000000 = $4 . 10^{26}$ keys
- trying all possibilities at 1 nanosecond per key requires....

4.10^{26} /(10^9 . 10^5 . 4 10^2) = 10^{10} years

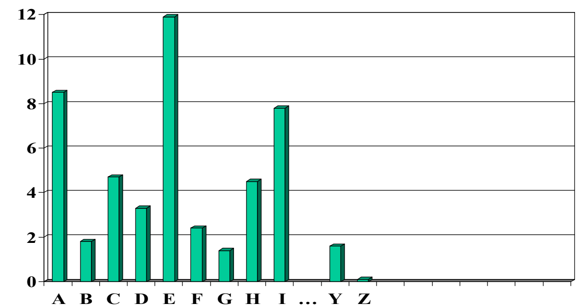
keys per second

seconds per day

days per year

9

Letter distributions



Substitutions

ABCDEFGHIJKLMNOPQRSTUVWXYZ

MZNJSOAXFQGYKHLUCTDVWBIRPE

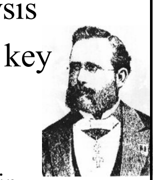
10

Assumptions on Eve (the opponent)

- A scheme is broken if Eve can deduce the key or obtain additional plaintext
- Eve can always try all keys till "meaningful" plaintext appears: a brute force attack
  - solution: large key space
- Eve will try to find shortcut attacks (faster than brute force)
  - history shows that designers are too optimistic about the security of their cryptosystems

11

Assumptions on Eve (the opponent)



- Cryptology = cryptography + cryptanalysis
- Eve knows the algorithm, except for the key (Kerckhoffs's principle)
- increasing capability of Eve:
  - knows some information about the plaintext (e.g., in English)
  - knows part of the plaintext
  - can choose (part of) the plaintext and look at the ciphertext
  - can choose (part of) the ciphertext and look at the plaintext

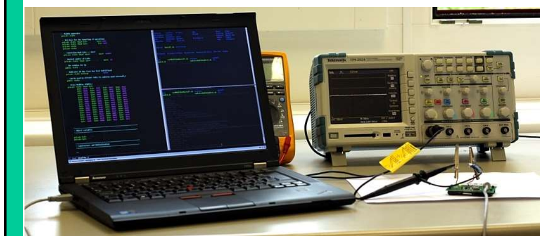
12

New assumptions on Eve

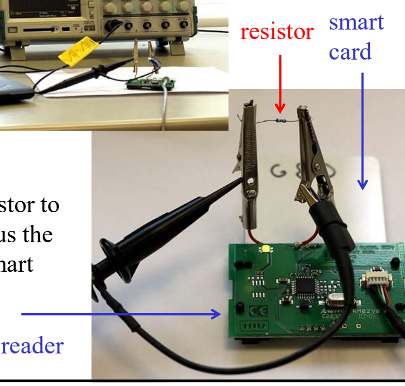
- Eve may have access to side channels
  - timing attacks
  - simple power analysis
  - differential power analysis
  - acoustic attacks
  - electromagnetic interference
- Eve may launch (semi-)invasive attacks
  - differential fault analysis
  - probing of memory or bus

13

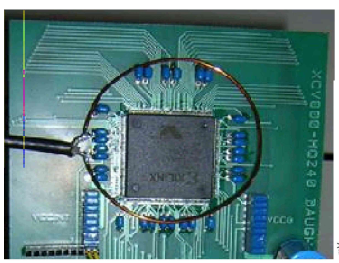
Side channel analysis: power setup



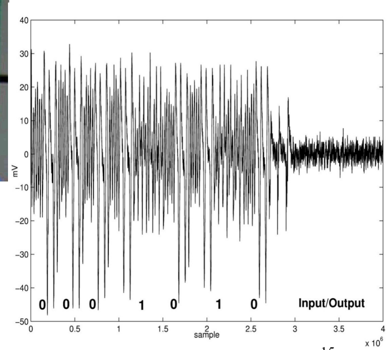
Measure voltage over a resistor to measure the current (and thus the power consumption) of a smart card



Side channel analysis: electromagnetic setup

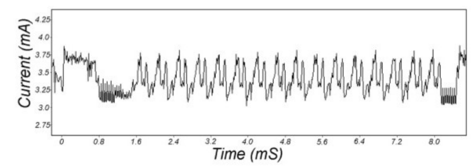


Use simple antenna to measure radiation of an FPGA computing a public key operation

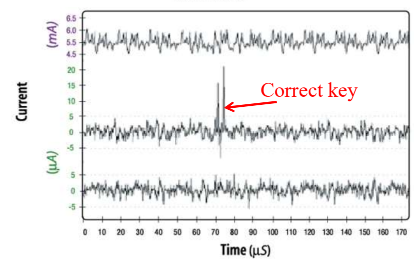


15

Simple and differential power analysis: DES block cipher



DES on a smart card: power consumption



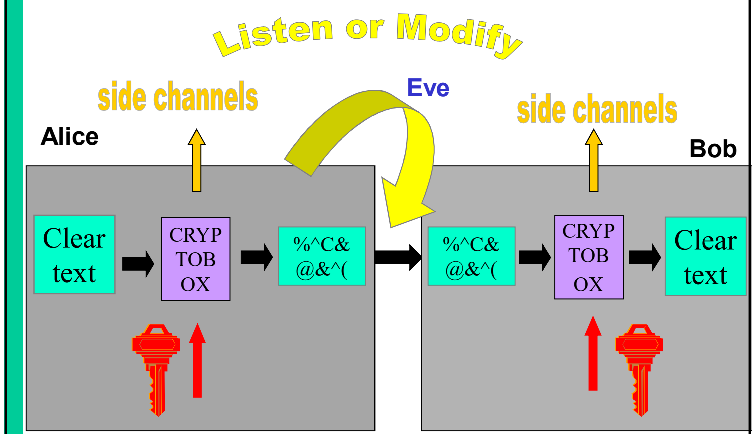
average power

2 correlation methods (example of success and failure)

16

Measurement data from 2 setups hence not comparable

Cryptology + side channels

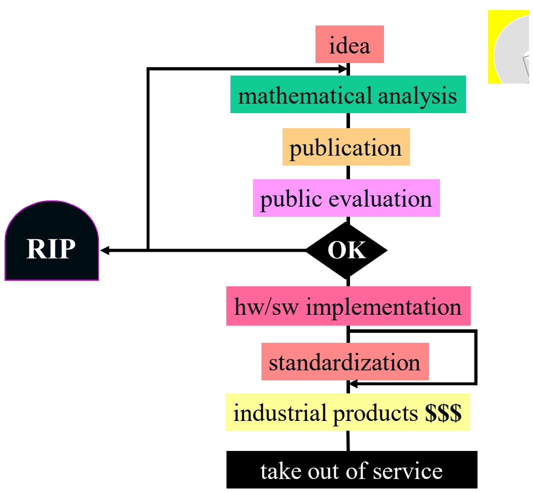


17

The Rotor machines (WW II)

18

Life cycle of a cryptographic algorithm

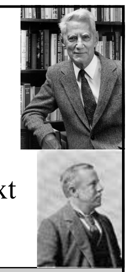


19

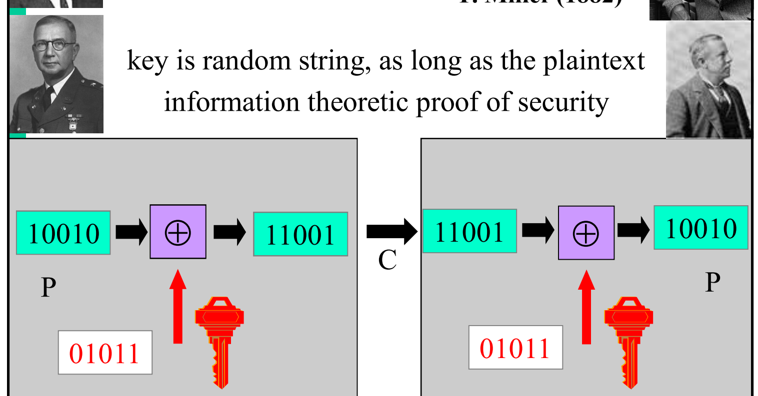
Vernam scheme (1917)
Mauborgne: one time pad (1917+x)

Shannon (1948)

F. Miller (1882)



key is random string, as long as the plaintext
information theoretic proof of security



Vernam scheme: Venona

$c_1 = p_1 + k$

$c_2 = p_2 + k$

then $c_1 - c_2 = p_1 - p_2$

a skilled cryptanalyst can recover p_1 and p_2 from $p_1 - p_2$ using the redundancy in the language

Example: c1 V c2 (not +)

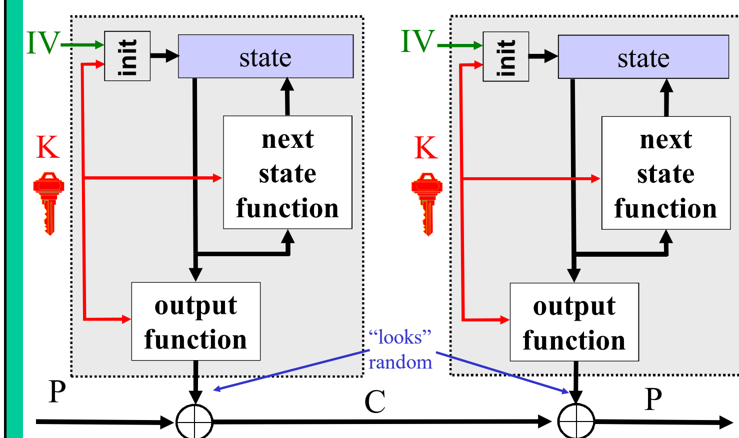
22

Three approaches in cryptography

- **information theoretic** security
  - ciphertext only
  - part of ciphertext only
  - noisy version of ciphertext
- **system-based** or practical security
  - also known as “prayer theoretic” security
- **complexity theoretic** security:
  model of computation, definition, proof
  - variant: quantum cryptography

23

Synchronous Stream Cipher (SSC)



Exhaustive key search

- 2018: 2^{40} instructions is easy, 2^{60} is somewhat hard, 2^{80} is hard, 2^{128} is completely infeasible
  - 1 million machines with 16 cores and a clock speed of 4 GHz can do 2^{56} instructions per second or 2^{80} per year
  - trying 1 key requires typically a few 100 instructions
- Moore's "law": speed of computers doubles every 18 months: key lengths need to grow in time
  - but adding 1 key bit doubles the work for the attacker
- Key length recommendations in 2018
  - < 70 bits: insecure
  - 80 bits: one year (but not for NSA)
  - 100 bits: 15-20 years
  - 256 bits: to resist quantum computers

25

SSC: Specific properties

- Recipient needs to be synchronized with sender
- No error-propagation
  - excellent for wireless communications
- Key stream independent of data
  - key stream can be precomputed
  - particular model for cryptanalysis: attacker is not able to influence the state

26

SSC: Avoid repeating key stream

- For a fixed key K and initial value IV, the stream cipher output is a deterministic function of the state.
- A repetition of the state (for a given K, IV) leads to a repetition of the key stream and plaintext recovery (think of the problem of Vernam encryption with reused key)
  - hence state needs to be large and next state function needs to guarantee a long period
  - IV can be used to generate a different key stream for every packet in a packet-oriented communication setting
  - old stream ciphers defined without IV are problematic in such a setting

27

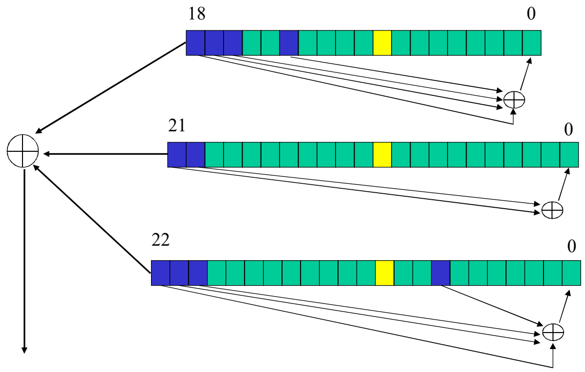
Practical stream ciphers

- A5/1 (GSM) (64 or 54)
- E0 (Bluetooth) (128)
- RC4 (browser) (40-128)

  } insecure!
- SNOW-3G (3GSM) (128)
- HC-128 (128)
- Trivium (80)
- ChaCha20 (128)

28

A5/1 stream cipher (GSM)



Clock control: registers agreeing with majority are clocked (2 or 3)

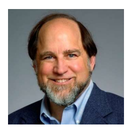
29

A5/1 stream cipher (GSM)

- exhaustive key search: 2^{64} (or rather 2^{54})
  - hardware 10K$ < 1 minute ciphertext only
- search 2 smallest registers: 2^{45} steps
- [BWS00] 1 minute on a PC
  - 2 seconds of known plaintext
  - 2^{48} precomputation, 146 GB storage
- [BB05]: 10 minutes on a PC,
  - 3-4 minutes of ciphertext only
- [Nohl-Paget'09]: rainbow tables
  - seconds with a few frames of ciphertext only

30

A simple cipher: RC4 (1987)



- designed by Ron Rivest (MIT)
- leaked in 1994
- S[0..255]: secret table derived from user key K

```
for i=0 to 255 S[i]:=i
j:=0
for i=0 to 255
    j:=(j + S[i] + K[i]) mod 256
    swap S[i] and S[j]
i:=0,  j:=0
```

31

A simple cipher: RC4 (1987)

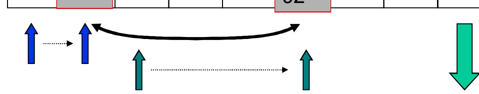
Generate key stream which is added to plaintext

```
i:=i+1
j:=(j + S[i]) mod 256
swap S[i] and S[j]
t:=(S[i] + S[j]) mod 256
output S[t]
```

t

000	001	002		093	094	095		254	255
205	162	013	...	033	92	079	...	099	143

i
j



32

RC4: weaknesses

- was often used with 40-bit key
  - US export restrictions until Q4/2000
- best known general shortcut attack: 2^{241} [Maximov-Khovratovich'09]
- weak keys and key setup (shuffle theory)
- large statistical deviations
  - bias of output bytes (sometimes very large)
  - can recover 220 out of 256 bytes of plaintexts after sending the same message 1 billion times (WPA/TLS)
- problem with resynchronization modes (WEP)
- problem with use in TLS

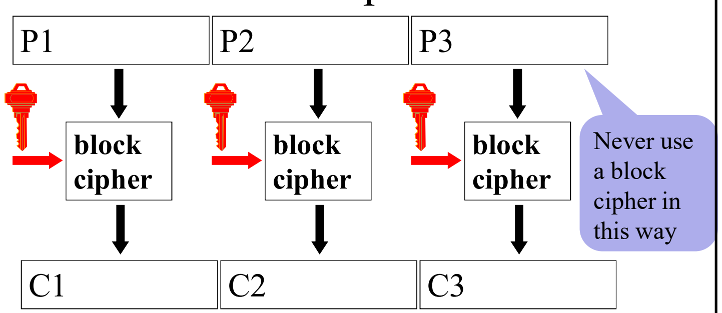
33

Block cipher

- large table: list n-bit ciphertext for each n-bit plaintext
  - if n is large: very secure (codebook)
  - but for an n-bit block: 2^n values
  - impractical if $n \geq 32$
- alternative n = 64 or 128
  - simplify the implementation
  - repeat many simple operations

34

Block cipher



- larger data units (blocks): 64…128 bits
- memoryless
- repeat simple operation (round) many times

35

Practical block ciphers

- DES: outdated
- 3-DES: financial sector
- AES
- KASUMI (3GSM)
- Keeloq (remote control for cars, garage doors)

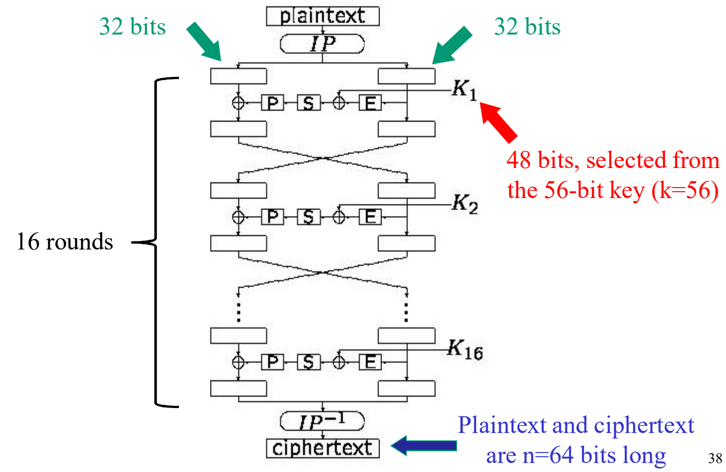
36

Data Encryption Standard (1977)

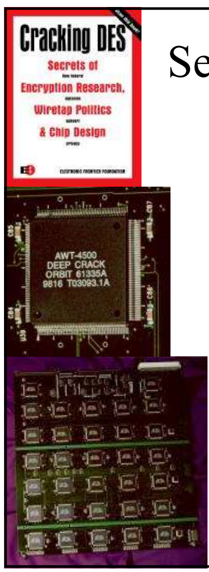
- encrypts 64 plaintext bits under control of a 56-bit key
- 16 iterations of a relatively simple mapping
- FIPS: US government standard for sensitive but unclassified data
- worldwide de facto standard since early 80ies
- surrounded by controversy

37

Data Encryption Standard (DES)



38



Security of DES (56-bit key)

- PC: trying 1 DES key: 7.5 ns
- Trying all keys on 128 PCs:
  1 month: $2^{27} \times 2^{16} \times 2^5 \times 2^7 = 2^{55}$
- M. Wiener's design (1993):
  1,000,000 \$ machine: 3 hours
  (in 2017: 0.3 seconds)

EFF Deep Crack (July 1998)
250,000 \$ machine: 50 hours…

39

Federal Register, July 24, 2004

DEPARTMENT OF COMMERCE

National Institute of Standards and Technology
[Docket No. 040602169– 4169– 01]

Announcing Proposed Withdrawal of Federal Information Processing Standard (FIPS) for the Data Encryption Standard (DES) and Request for Comments

AGENCY: National Institute of Standards and Technology (NIST), Commerce.

ACTION: Notice; request for comments.

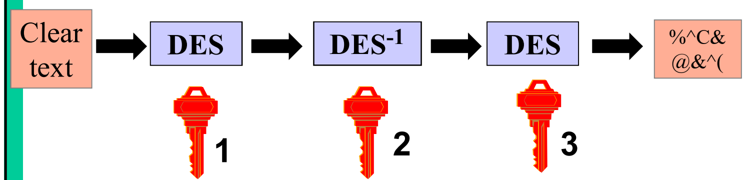
- **SUMMARY:** The Data Encryption Standard (DES), currently specified in Federal Information Processing Standard (FIPS) 46–3, was evaluated pursuant to its scheduled review. At the conclusion of this review, **NIST determined that the strength of the DES algorithm is no longer sufficient to adequately protect Federal government information**. As a result, NIST proposes to withdraw FIPS 46–3, and the associated FIPS 74 and FIPS 81. Future use of DES by Federal agencies is to be permitted only as a component function of the Triple Data Encryption Algorithm (TDEA).

40

3-DES: NIST Spec. Pub. 800-67

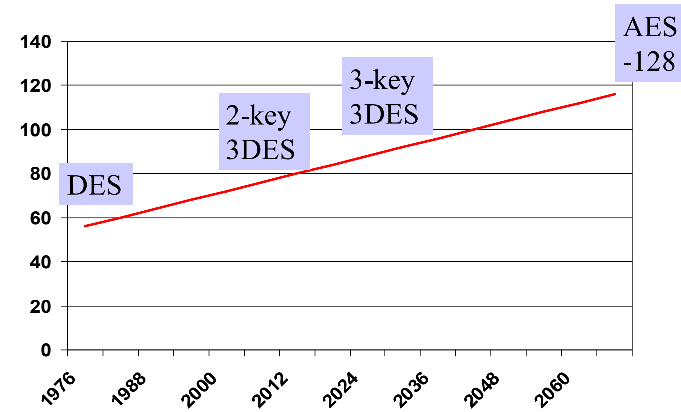
(May 2004)

- two-key triple DES: until 2009
- three-key triple DES: until 2030



41

Symmetric Key Lengths and Moore's "law"



Moore's "law": speed of computers doubles every 18 months

42

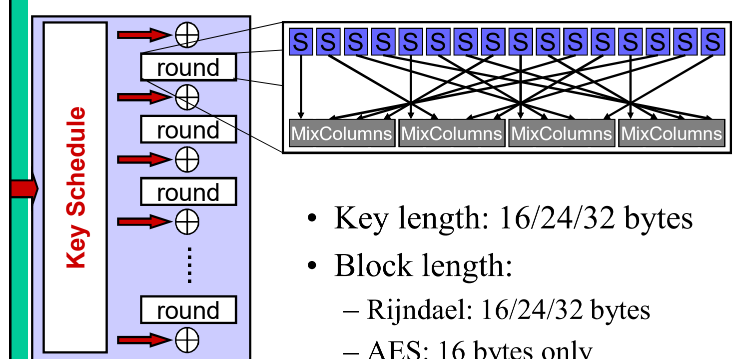
AES (Advanced Encryption Standard)

- open competition launched by US government (Sept. '97) to replace DES
- 22 contenders including IBM, RSA, Deutsche Telekom
- 128-bit block cipher with key of 128/192/256 bits
- as strong as triple-DES, but more efficient
- royalty-free

A machine that cracks a DES key in 1 second would take 149 trillion years to crack a 128-bit key

43

AES: Rijndael



- Key length: 16/24/32 bytes
- Block length:
  - Rijndael: 16/24/32 bytes
  - AES: 16 bytes only

44

AES (2001)

- FIPS 197 published on December 2001after 4-year open competition
  - other standards: ISO, IETF, IEEE 802.11,…
- fast adoption in the market
  - except for financial sector
  - NIST validation list: ≥ 5900 implementations
    - http://csrc.nist.gov/groups/STM/cavp/documents/aes/aesval.html
- 2003: AES-128 also for secret information and AES-192/-256 for top secret information!
- 2015: NSA recommends to switch to AES-256 for the long term (reason: quantum computers)

45

AES (2001)

- **security:**
  - algebraic attacks of [Courtois+02] not effective
  - side channel attacks: cache attacks on **unprotected** implementations
- **speed:**
  - software: 7.6 cycles/byte [Käsper-Schwabe'09]
  - hardware: Intel provides AES instruction (since 2010) at 0.63..1.5 cycles/byte for decryption – AMD one year behind; ARM a bit more

[Shamir '07] AES may well be the last block cipher

46

Encryption limitations

- Ciphertext becomes random string: "normal" crypto does not encrypt a credit card number into a (valid) credit card number
- Typically does not hide the length of the plaintext (unless randomized padding)
- Does **not** hide existence of plaintext (requires steganography)
- Does **not** hide that Alice is talking to Bob (e.g. Tor)
- Does **not** hide traffic volume (requires dummy traffic)

Symmetric cryptology: data authentication

- the problem
- hash functions without a key
  - MDC: Manipulation Detection Codes
- hash functions with a secret key
  - MAC: Message Authentication Codes

48

Data authentication: the problem

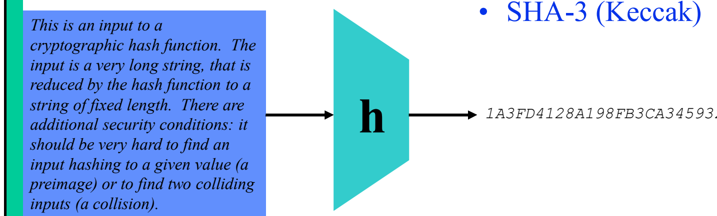
- encryption provides confidentiality:
  - prevents Eve from learning information on the cleartext/plaintext
  - but does not protect against modifications (active eavesdropping)
- Bob wants to know:
  - the **source** of the information (data origin)
  - that the information has not been **modified**
  - (optionally) the **destination** of the information
  - (optionally) **timeliness** and **sequence**
- data authentication is typically more complex than data confidentiality

There are no applications that require encryption without data authentication (but this can still be found in legacy applications with as excuse performance)

Data authentication: MDC

- MDC (manipulation detection code)
- Protect short hash value rather than long text

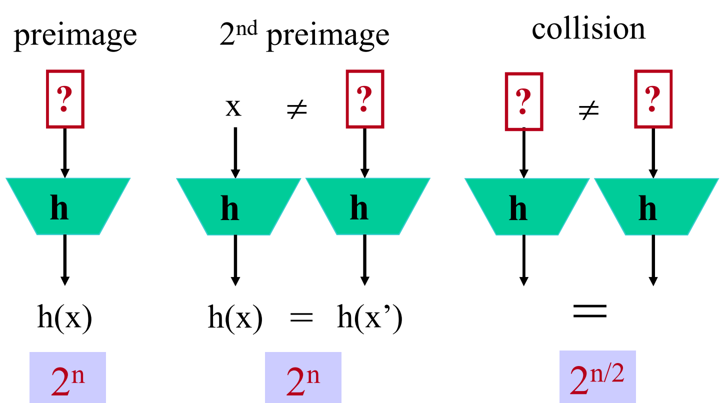
- (MD5)
- (SHA-1), SHA-256, SHA-512
- RIPEMD-160
- SHA-3 (Keccak)



Shift authenticity of file to authenticity of short hash value

50

MDC Security requirements (n-bit result)



Data authentication: MDC

- n-bit result
- preimage resistance: for given y, hard to find input x such that h(x) = y (2^n operations)
- 2^{nd} preimage resistance: hard to find x'≠ x such that h(x') = h(x) (2^n operations)
- Collision resistance: hard to find (x,x') with x' ≠ x such that h(x') = h(x) ($2^{n/2}$ operations)

52

Important hash algorithms

- MD5
  - (2^{nd}) preimage 2^{128} steps (improved to 2^{123} steps)
  - collisions 2^{64} steps

shortcut: Aug. '04: 2^{39} steps; '09: 2^{20} steps

- SHA-1:
  - (2^{nd}) preimage 2^{160} steps
  - collisions 2^{80} steps

0.3 M$ for 1 year in 2018

shortcut: Aug. '05: 2^{69} steps

collisions 23/02/2017: 2^{61} steps

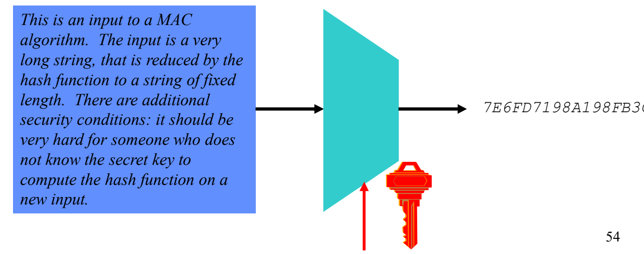
- SHA-2 family (2002)
- SHA-3 family (2013) – Keccak (Belgian design)
  - (2^{nd}) preimage 2^{256} .. 2^{512} steps
  - collisions 2^{128} .. 2^{256} steps

53

Data authentication: MAC algorithms

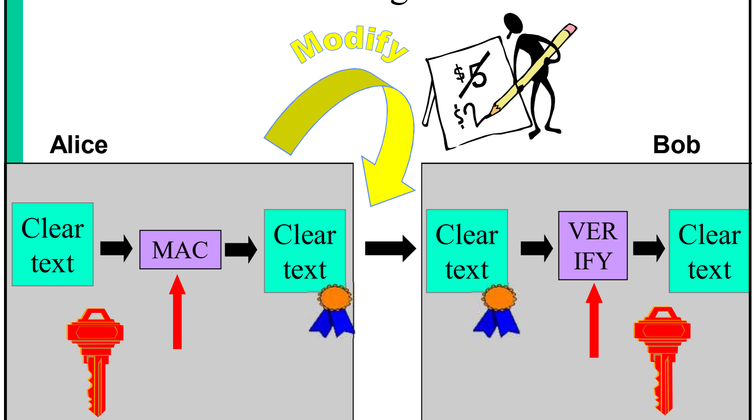
- Replace protection of authenticity of (long) message by protection of secrecy of (short) key
- Add MAC to the plaintext

- CBC-MAC (CMAC)
- HMAC
- GMAC



54

MAC algorithms



Data authentication: MAC algorithms

- typical MAC lengths: 32..96 bits
  - Forgery attacks: 2^m steps with m the MAC length in bits
- typical key lengths: (56)..112..160 bits
  - Exhaustive key search: 2^k steps with k the key length in bits
- birthday attacks: security level smaller than expected

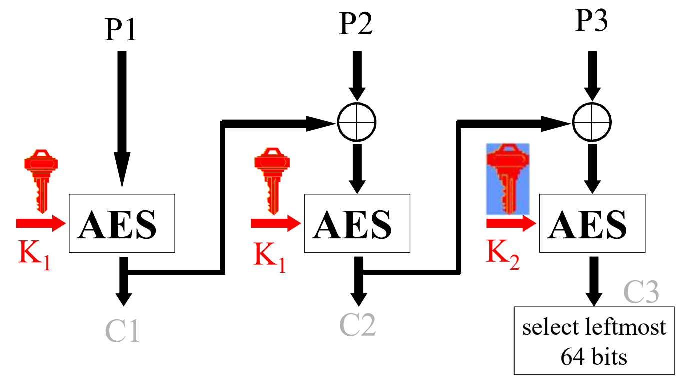
56

MAC algorithms

- Banking: CBC-MAC based on triple-DES
- Internet: HMAC and CBC-MAC based on AES
- information theoretic secure MAC algorithms (authentication codes): GMAC/UMAC/Po1y1305
  - highly efficient
  - rather long keys (some)
  - part of the key refreshed per message

57

CBC-MAC based on AES (LMAC)

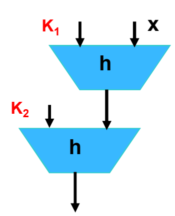


Secure up to 2^{50} message blocks

58

MAC based on a hash function

- **HMAC**
  - $h_K(X) = h(K_2 \| h(K_1 \| x))$
  - not secure with MD4/MD5
  - ok with SHA-1 and SHA-2
  - there is a an alternative (simple) construction with SHA-3



NIST's Modes of Operation for AES

- ECB/CBC/CFB/OFB + CTR (Dec 01) Use only with MAC
- MAC algorithm: CMAC (May 05)
- Authenticated encryption:
  - CCM: CTR + CBC-MAC
  - GCM: Galois Counter Mode (robustness issues)

Issues:
- associated data
- parallelizable
- on-line
- provable security

- IAPM
- XECB
- OCB

patented

- CCM
- GCM
- (EAX)
- (CWC)

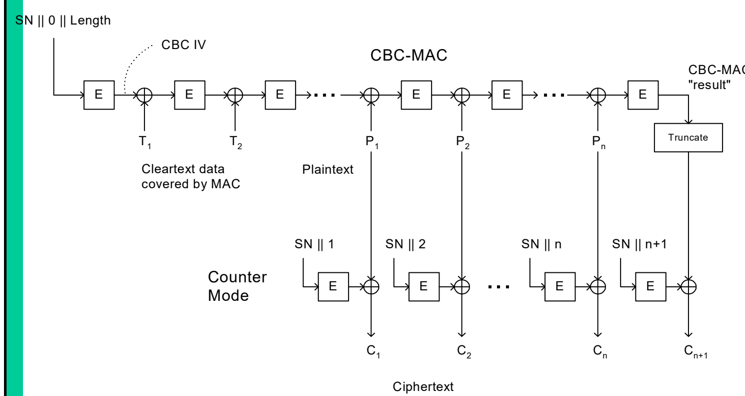
60

Concrete recommendations

- AES-128 in CCM mode
  - CCM = CTR mode + CBC-MAC
  - change key after 2^{40} blocks
- Stream ciphers (better performance)
  - hardware: SNOW-3G or Trivium
  - software: HC-128 or ChaCha20
  - combine with Poly1305 or robust version of GMAC
- CAESAR: open competition from 2013-2018
  - http://competitions.cr.yp.to/caesar.html

61

Example: CCM: CTR + CBC-MAC



SN = packet sequence number (WEP "IV")

62

Public-key cryptology

- the problem
- public-key encryption
- digital signatures
- an example: RSA
- advantages of public-key cryptology

63

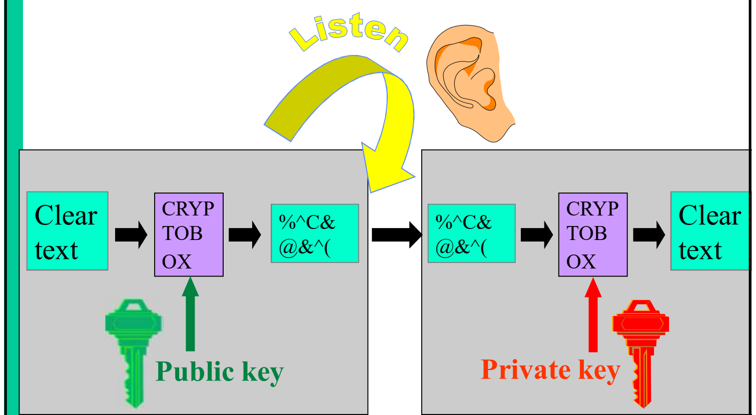
Limitation of symmetric cryptology

- Reduce security of information to security of keys

- But: how to establish these secret keys?
  - cumbersome and expensive
  - or risky: all keys in 1 place
- Do we really need to establish secret keys?

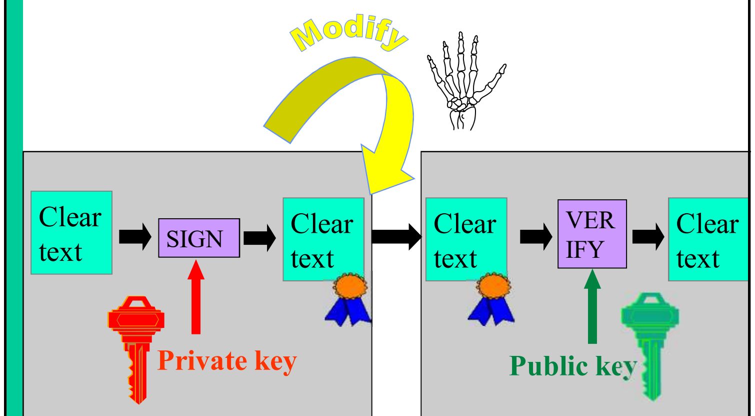
64

Public key cryptology: encryption



65

Public key cryptology: digital signature



66

A public-key distribution protocol: Diffie-Hellman

- Before: Alice and Bob have never met and share no secrets; they know a public system parameter α

Alice		Bob
generate x	α^x →	*generate* y
compute α^x	← α^y	*compute* α^y
compute $k=(\alpha^y)^x$		*compute* $k=(\alpha^x)^y$

- After: Alice and Bob share a short term key k
  - Eve cannot compute k : in several mathematical structures it is hard to derive x from α^x (this is known as the discrete logarithm problem)

67

RSA ('78)

- choose 2 "large" prime numbers p and q
- modulus n = p.q
- compute $\lambda(n) = \mathrm{lcm}(p-1,q-1)$
- choose e relatively prime w.r.t. $\lambda(n)$
- compute $d = e^{-1} \bmod \lambda(n)$

The security of RSA is based on the "fact" that it is easy to generate two large primes, but that it is hard to factor their product

- public key = (e,n)
- private key = d of (p,q)

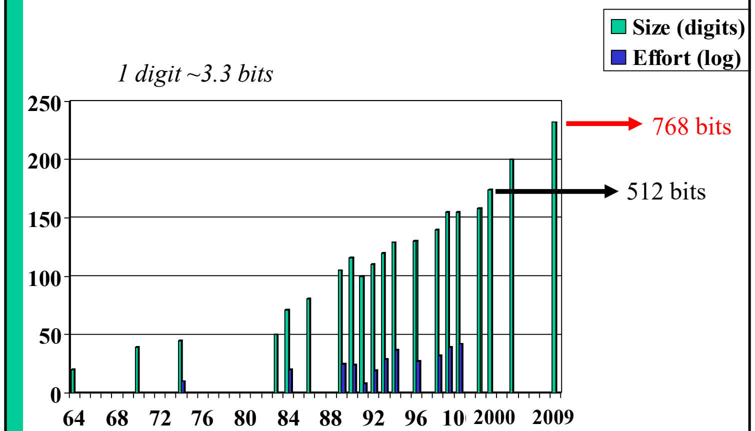
- encryption: $c = m^e \bmod n$
- decryption: $m = c^d \bmod n$

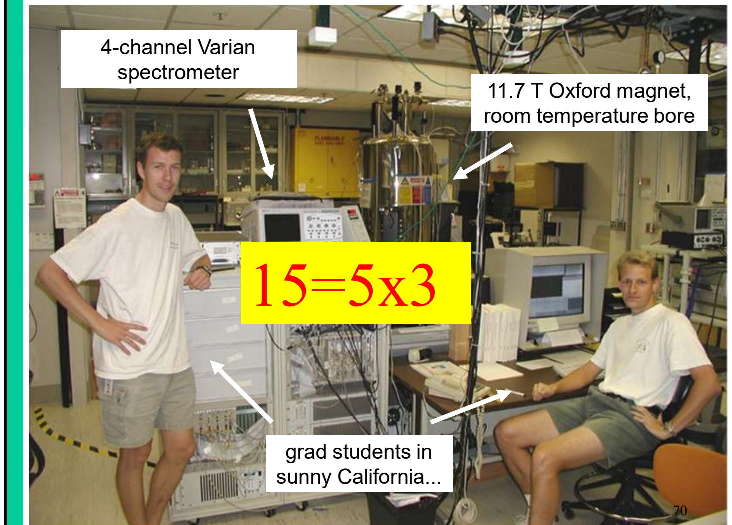
try to factor 2419

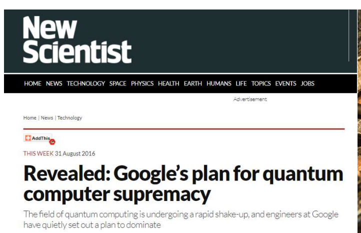
68

Factorisation records

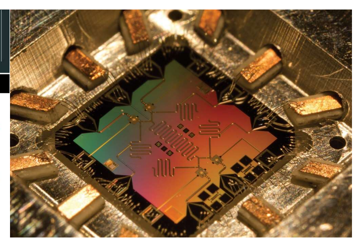
2009: 768 bits or 232 digits

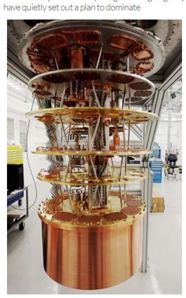





New Scientist

Revealed: Google's plan for quantum computer supremacy





IBM 2017: 50 qubits

Google 2018: 72 qubits
RSA-1024 would require 2048 ideal qubits or 1.5 million real qbits

Rigetti 2018: 128 qubits

http://www.qubitcounter.com/

When to switch to quantum resistant cryptography? [Mosca]

Q = #years until first large quantum computer
x = #years it takes to switch (3-10 years)
y = #years data needs to be confidential (10 years)

Need to start switching in the year
2018 + Q – x – y
e.g. Q = 15, x=5, y=10: today!

For data and entity authentication: y = small
(and defense-in-depth)

Advantages of public key cryptology

- Reduce protection of information to protection of authenticity of public keys
- Confidentiality without establishing secret keys
  - extremely useful in an open environment
- Data authentication without shared secret keys: digital signature
  - sender and receiver have different capability
  - third party can resolve dispute between sender and receiver

73

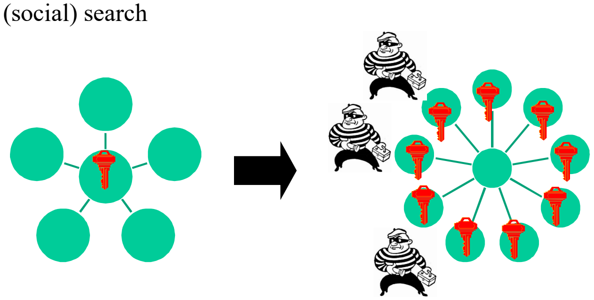
Disadvantages of public key cryptology

- Calculations in software or hardware two to three orders of magnitude slower than symmetric algorithms
- Longer keys: 512..4096 bits rather than 80…256 bits
- What if factoring is easy? (e.g. if large quantum computers can be built)

74

Secure multi-party computation

- auctions
- medical statistics and advice
- e-voting
- road pricing
- (social) search



Crypto software libraries

Wikipedia

Javascript: https://gist.github.com/jo/8619441

http://ece.gmu.edu/crypto_resources/web_resources/libraries.htm

C/C++/C#

- Botan (C++)
- BoringSSL
- cryptlib (C)
- Crypto++ (C++)
- GnuTLS (C)
- libgcrypt (C++)
- libtomcrypt (C)
- libsodium (C)
- Miracl (binaries)
- NaCl (C/Assembly)
- Nettle (C)
- OpenSSL (C)
- WolfCrypt (C)

C/C++ (embedded)

- CyaSSL (C)
- MatrixSSL (C++)

Java

- SunJCA/JCE
- BouncyCastle (BC, C#)
- EspreSSL
- FlexiProvider
- GNU Crypto
- IAIK
- Java SSL
- RSA JSafe

Crypto recommendations

http://www.enisa.europa.eu/activities/identity-and-trust/library/deliverables/algorithms-key-sizes-and-parameters-report
http://www.ecrypt.eu.org/

Good

Authenticated encryption
- AES-CCM
- HC-128 + Poly1305
- Chacha20 + Poly1305

HMAC-SHA-2

SHA-3

Diffie-Hellman
- $Z_p \geq 2048$
- ECC ≥ 256 and up

ECIES ≥ 256 and up

RSA KEM-DEM ≥ 2048

RSA-PSS

Bad

Encryption only, e.g. AES-CBC

RC4, A5/1, A5/2, E0, DST, Keeloq, Crypto-1, Hitag-2, DSAA, DSC, GMR-1, GMR-2, CSS

MD2, MD4, MD5, SHA-1

RSA PKCS#1v.5

DSA

Dual_EC_DRBG

ECC curves from NIST?

SSL 3.0/TLS 1.0/TLS 1.1

TLS with RSA key exchange

Skype

Implementations that do not run in constant time

Reading material

- B. Preneel, Modern cryptology: an introduction.
  - This text corresponds more or less to the second half of these slides
  - It covers in more detail how block ciphers are used in practice, and explains how DES works.
  - It does not cover identification, key management and application to network security.

78

Selected books on cryptology

- D. Stinson, *Cryptography: Theory and Practice*, CRC Press, 3rd Ed., 2005. Solid introduction, but only for the mathematically inclined.
- A.J. Menezes, P.C. van Oorschot, S.A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1997. The bible of modern cryptography. Thorough and complete reference work – not suited as a first text book. Freely available at http://www.cacr.math.uwaterloo.ca/hac
- N. Smart, *Cryptography, An Introduction*: 3rd Ed., 2008. Solid and up to date but on the mathematical side. Freely available at http://www.cs.bris.ac.uk/~nigel/Crypto_Book/
- B. Schneier, *Applied Cryptography*, Wiley, 1996. Widely popular and very accessible – make sure you get the errata, online
- Other authors: Johannes Buchmann, Serge Vaudenay

79

Books on network security and more

- W. Stallings, *Network and Internetwork Security: Principles and Practice*, Pearson, 7th Ed., 2016. Solid background on network security. Explains basic concepts of cryptography.
- W. Stallings, *Network Security Essentials: Applications and Standards*, 6th Ed., 2016, Pearson. Short version of the previous book.
- W. Diffie, S. Landau, *Privacy on the line. The politics of wiretapping and encryption*, MIT Press, 2nd Ed., 2007. The best book so far on the intricate politics of the field.
- Ross Anderson, *Security Engineering*, Wiley, 2nd Ed., 2008. Insightful. A must read for every information security practitioner. First and second editions are available for free at http://www.cl.cam.ac.uk/~rja14/book.html
- Jay Ramachandran, *Designing Security Architecture Solutions,* Wiley 2002.
- Gary McGraw, *Software Security: Building Security In*, Addison Wesley, 2006.

80