

http://www.ecrypt.eu.org

## Cryptographic Algorithms and Protocols for Network Security

Prof. Bart Preneel  
 COSIC, K.U.Leuven, Belgium  
 Bart.Preneel(at)esat.kuleuven.be  
 http://homes.esat.kuleuven.be/~preneel  
 September 2008

## Information processing

the Internet of things, ubiquitous computing, pervasive computing, ambient intelligence ( $10^{12}$ )

Internet and mobile ( $10^9$ )

PCs and LANs ( $10^7$ )

mainframe ( $10^5$ )

mechanical processing ( $10^4$ )

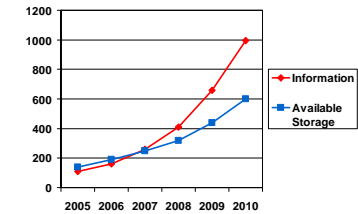
manual processing ( $10^2$ )

## Information storage and transmission

- 2010: 1 Zettabyte/year added to the digital universe; this corresponds to 400 million hard drives with a capacity of 2.5 Terabyte
- 2010: US internet traffic will grow to 1 Zettabyte/month (today: 4 Exabyte)

Megabyte	$10^6$
Gigabyte	$10^9$
Terabyte	$10^{12}$
Petabyte	$10^{15}$
Exabyte	$10^{18}$
Zettabyte	$10^{21}$


photo	1 Mbyte
song	50 Mbyte
movie	4.7 Gbyte
95 yrs life movie	$3 \cdot 10^9$ seconds
	3 Petabyte



## Exponential growth

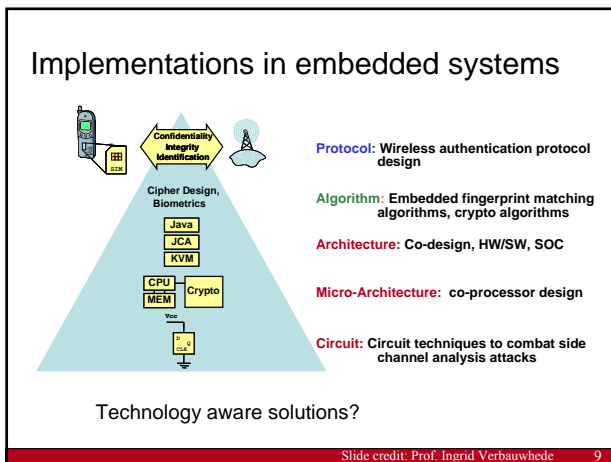
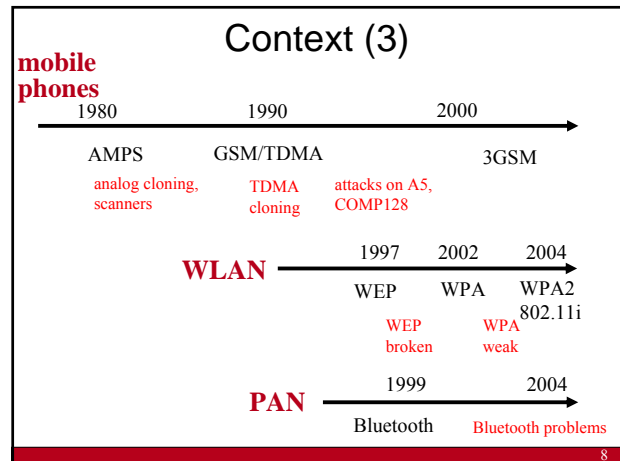
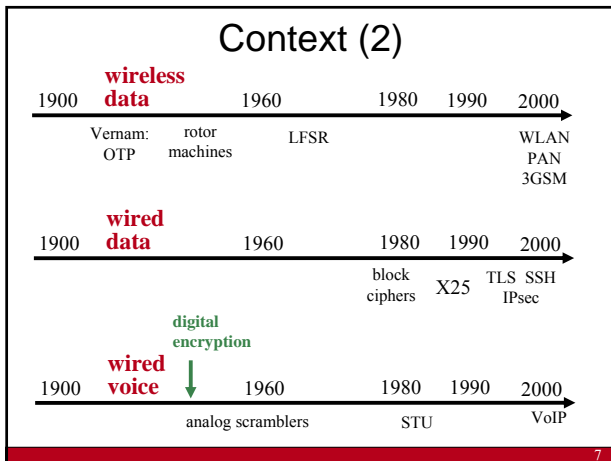
Ray Kurzweil, KurzweilAI.net

- Human brain:  $10^{14}$  ...  $10^{15}$  ops and  $10^{13}$  bits memory
- 2025: 1 computer can perform  $10^{16}$  ops ( $2^{53}$ )
- 2013:  $10^{13}$  RAM bits (1 Terabyte) cost 1000\$




## Context

DES, RSA, DH, CBC-MAC	<b>HARDWARE</b>	70
Provable security (PKC), ZK, ElGamal, ECC, stream ciphers	Limited (govt+financial sector) DES, 3DES	80
MD4, MD5	<b>SOFTWARE</b>	90
Provable security (SKC)	GSM, PGP	
Key escrow	C libraries (RSA, DH)	
How to use RSA?	SSL/TLS, IPsec, SSH, S/MIME	
Alternatives to RSA	Java crypto libraries	
PKI	WLAN	
AES	<b>EVERYWHERE</b>	
ID-Based Crypto	Trusted computing, DRM, 3GPP, RFID, sensor nodes	
	...	



### Disclaimer:

cryptography  $\neq$  security

- crypto is only a tiny piece of the security puzzle
  - but an important one
  - that often creates trouble
- most systems break elsewhere
  - incorrect requirements or specifications
  - implementation errors
  - application level
  - social engineering
- for intelligence, traffic analysis (SIGINT) is often much more important than cryptanalysis

10

[Gene Spafford] (using encryption on the Internet is like) using an armoured truck to transport rolls of pennies between someone on a park bench and someone doing business from a cardboard box

[Adi Shamir] We are winning yesterday's information security battles, but we are losing the war. Security gets worse by a factor of 2 every year.

[Andrew Odlyzko] Humans can live with insecure systems. We couldn't live with secure ones.

11

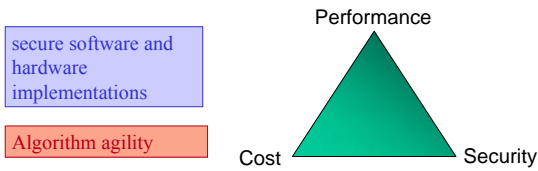
### Outline

- Cryptographic algorithms
  - Block ciphers
  - Hash functions
  - Stream ciphers
  - MAC algorithms
  - Public key encryption and digital signatures
- Cryptographic protocols
- Implementations issues
- Research challenges

12

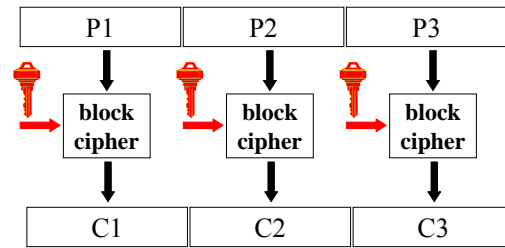
### Challenges for cryptographic algorithms

- security for 50-100 years
- authenticated encryption of Terabit/s networks
- ultra-low power/footprint



13

### Block cipher



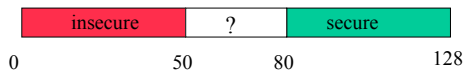
- larger data units: 64...128 bits
- memoryless
- repeat simple operation (round) many times

14

### Block ciphers

- 3-DES (112-168)
- AES (128-192-256)
- Camellia (128-192-256)
- KASUMI (128 in 3G, 64 in 2G)
- Keeloq (64)
- IDEA (128)
- DBV-CSA3

Symmetric key lengths



15

### DES (1977)

- 56-bit key length is too short
- 25/10/99: DES reaffirmed for the 4th time as FIPS 46-3
- 2008: \$1 million search machine: 15 seconds
  - cost per key: less than \$0.30
- 2008: 250 PCs at night: 1 month
  - Cost per key: essentially 0 (+ some patience)



### 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.

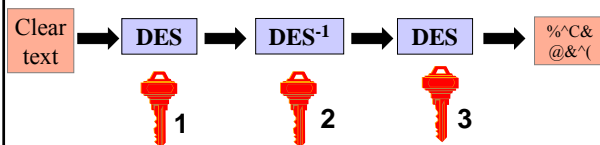
**SUMMARY:** The Data Encryption Standard (DES), currently specified in Federal Information Processing Standard (FIPS) 46–3, was evaluated pursuant to its scheduled review. At the conclusion of this review, **NIST determined that the strength of the DES algorithm is no longer sufficient to adequately 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).

17

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

(May 2004)

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



Financial sector will not be ready with upgrade to 3-key 3-DES in 2010

18

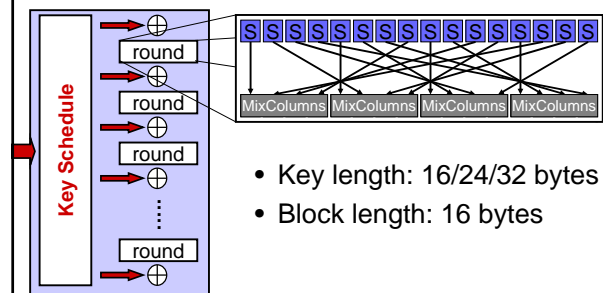
## AES (2001)

- open competition: 1997-2000
- FIPS 197 published on December 2001
- mandatory for sensitive US govt. information
- fast adoption in the market
  - > 1000 products
  - Sept. 2008: 857 AES product certifications by NIST
  - standardization: ISO, IETF, IEEE 802.11,...
- slower adoption in financial sector
- mid 2003: AES-128 also for **classified** information and AES-192/-256 for **secret** and **top secret** information!

AES may well be the last block cipher

19

## AES/Rijndael



- Key length: 16/24/32 bytes
- Block length: 16 bytes

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

20

## AES: rich mathematical structure

- very compact/efficient implementations
  - SW: 14 cycles per byte or 1-2 Gbit/s on high end PCs
  - HW: most compact: 3600 gates
  - HW: fastest up to 43 Gbit/s in 130nm CMOS
  - Intel (+AMD): new AES instruction: 0.75 cycles/byte
- security
  - is it hard to solve sets of non-linear Boolean equations?
  - no attack has been found that can exploit this structure (in spite of earlier claims)
  - main threat is implementation level attack (cache timing, fault attacks): requires special countermeasures

21

## Modes of Operation for AES

- encryption: ECB/CBC/CFB/OFB;
  - CTR mode allows for pipelining ('01)
- data authentication: CMAC ('05), EMAC

22

## Block ciphers: Keeloq

- Microchip Inc. algorithm, designed in the 1980s
- Allegedly used in 80% of the cars for car locks, car alarms
- Block cipher with 32-bit blocks, 64-bit keys and 528 simple rounds



23

## Block ciphers: Keeloq (2)

- Leaked on the internet in 2006
- [Bogdanov07] in some cases car key = Master key + Car ID
- [Bogdanov07], [Courtois-Bard-Wagner07] first cryptanalysis
- [Biham-Dunkelman-Indesteeghe-Keller-Preneel07]:
  - 1 hour access to token ( $2^{16}$  known texts)
  - 2 days of calculation on 50 PCs (10.000\$) -  $2^{44.5}$  encryptions
- [Eisenbarth-Kasper-Moradi-Paar-Salmasizadeh-Manzuri-ShalmaniPaar 08]
  - Side channel attack allows to recover master key

in 2010 cryptographers will drive expensive cars

24

### Block ciphers: conclusions

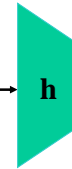
- Several mature block ciphers available
- Security well understood
  - in particular against statistical attacks (differential, linear) and structural attacks
  - algebraic attacks may be further developed
- Modes for authenticated encryption

25

### Hash functions

- MDC (manipulation detection code)
- Protect short hash value rather than long text
- collision resistance
- preimage resistance
- 2<sup>nd</sup> preimage resistance
- pseud-random properties

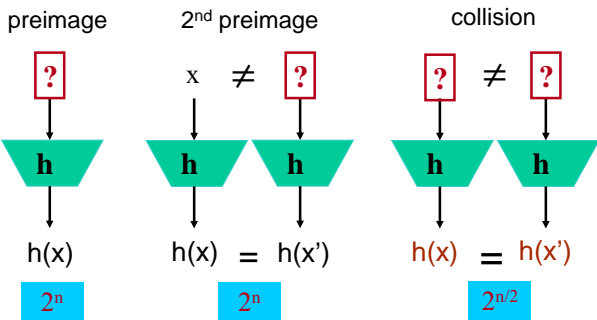
This is an input to a cryptographic hash function. The input is a very long string, that is reduced by the hash function to a string of fixed length. There are additional security conditions: it should be very hard to find an input hashing to a given value (a preimage) or to find two colliding inputs (a collision).



1A3FD4128A198FB3CA345932

26

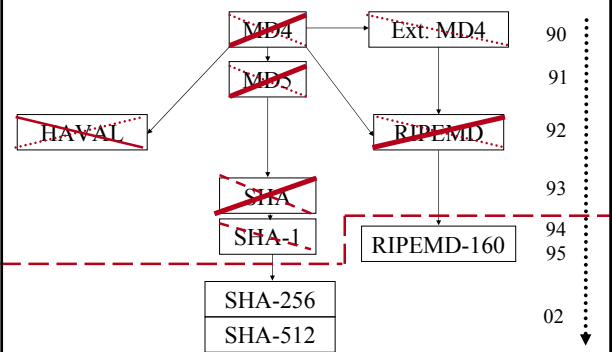
### Security requirements (n-bit result)



> 90% of all designs for collision resistant hash functions are broken

27

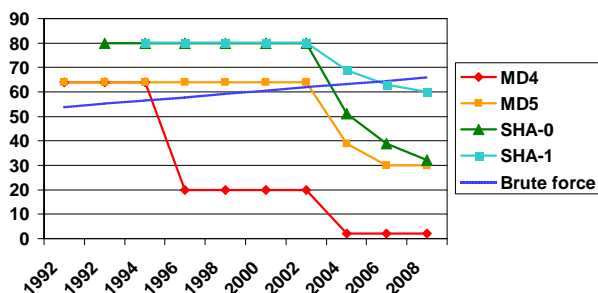
### MDx-type hash function history



28

### The complexity of collision attacks

Brute force: 1 million PCs or US\$ 100 000 hardware



29

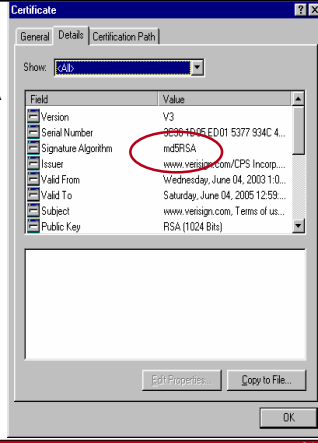
### MD4

- Design from 1990
- Problems in reduced round versions found quickly
- MD4 is still in use today
  - “encrypt” passwords in Windows NT
  - the S/KEY one-time-password system
  - integrity checks (rsync – eDonkey)
  - HMAC-MD4

30

## MD5

- Advice (RIPE since '92, RSA since '96): **stop using MD5**
- Largely ignored by industry (click on a cert...)
- Collisions for MD5 are within range of a brute force attack anyway ( $2^{64}$ ): with 100.000\$ a few days
- [Wang+'04] collision in 15 minutes on a PC
- 2007: collisions in seconds




31


## SHA-1

- SHA designed by NIST (NSA) in '93
- redesign after 2 years ('95) to SHA-1
- Collisions found for SHA-0 in  $2^{51}$  [Joux+'04]
- Reduced to  $2^{39}$  [Wang+'05] and  $2^{32}$  [Rechberger+'07]
- Collisions for SHA-1 in  $2^{63}$  [Wang+'05]
- Collisions for SHA-1 found for 70 out of 80 rounds [De Cannière-Mendel-Rechberger+'07] in  $2^{44}$

32



33

From: "Cryptography Simplified in Microsoft .NET" Paul D. Sheriff (PDSA.com) [Nov. 2003] 

### How to Choose an Algorithm

- For example, SHA1 uses a 160-bit encryption key, whereas MD5 uses a 128-bit encryption key; thus, SHA1 is more secure than MD5.
- Another point to consider about hashing algorithms is whether or not there are practical or theoretical possibilities of collisions. Collisions are bad since two different words could produce the same hash. **SHA1, for example, has no practical or theoretical possibilities of collision. MD5 has the possibility of theoretical collisions, but no practical possibilities.**

In September 2008 this information is still available on MSDN

34

## Hash function attacks:

cryptographic meltdown yet with limited impact

- **Collisions**
  - MD4, MD5 trivial
  - **SHA-1 expected in 2009** ( $2^{60}$  steps) [Rechberger+'07]
  - fake X.509 certs based on collisions
  - colliding Word/ps/TIFF documents
  - issue for digital signatures for non-repudiation (cf. traffic tickets in Australia?)
- **2<sup>nd</sup> preimages**
  - MD4:  $2^{102} < 2^{128}$
  - progress is being made: 45 out of 80 rounds of SHA-1
- **HMAC**
  - HMAC-MD4 broken
  - HMAC-MD5 questionable for the long term
- **APOP Post Office Protocol Version 3**
  - Need a few hundred identifications to reduce search for password by a factor 250,000

35

## Hash functions: what to do?

- alternatives today:
  - RIPEMD-160 seems more secure than SHA-1 ☺
  - SHA-256, SHA-512
  - Whirlpool
- upgrading MD5 and SHA-1 in Internet protocols:
  - it doesn't work: **algorithm flexibility is much harder than expected**
- randomized hashing
- NIST will run an open competition from 2008 to 2012
  - The AHS must support 224, 256, 384, and 512-bit message digests, and must support a maximum message length of at least  $2^{64}$  bits
  - 31 October 2008: submissions
  - February 2008: kickoff workshop
  - 2Q10 Announce finalists
  - 4Q11 Announce decision
  - 3Q12 Publish Advanced Hash Function Standard

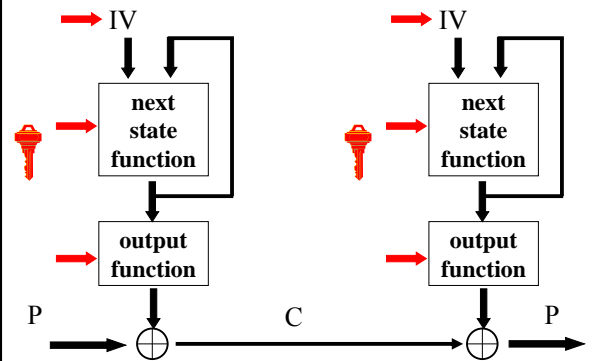
36

### Hash functions: conclusions

- Area much less mature
  - Structural attacks discovered in last 5 years
- Early popular designs not very secure
- Performance over security, even if this is not justified by most applications today
- NIST AHS will improve things by 2012

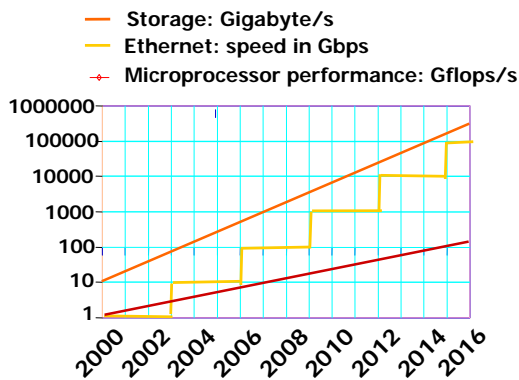
37

### Model of a practical stream cipher



38

### Moore's Law: computation/storage 2000-2020



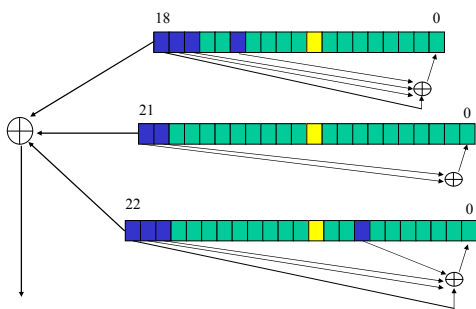
39

### Stream ciphers

- historically very important (compact)
  - LFSR-based: A5/1, E0 – practical attacks known
  - software-oriented: RC4 – serious weaknesses
  - block cipher in CTR or OFB (slower)
- today:
  - many broken schemes
  - exception: SNOW2.0, MUGI
  - lack of standards and open solutions

40

### A5/1 stream cipher (GSM)



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

41

### A5/1 stream cipher (GSM)

#### A5/1 attacks

- 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**
  - Problem with order of encryption and error correction: testing key is easy

42

### Bluetooth stream cipher

- brute force:  $2^{128}$  steps
- [Lu+05] 24 known bits of  $2^{24}$  frames,  $2^{38}$  computations,  $2^{33}$  memory

### A simple cipher: RC4 (1992)

- designed by Ron Rivest (MIT)
- $S[0..255]$ : 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
    
```

### A simple cipher: RC4 (1992)

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]
    
```

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

Diagram illustrating the state of the permutation table S. The index i points to the current element being swapped, and j points to the element it is swapped with. The output t is the element at index S[i] + S[j].

### RC4: weaknesses

- historically used with 40-bit key
  - US export restrictions until Q4/2000
- best known general shortcut attack:  $2^{241} < 2^{779}$
- weak keys and key setup (shuffle theory)
- some statistical deviations
  - e.g., 2nd output byte is biased
  - solution: drop first 256 bytes of output
- serious problem with resynchronization modes (WEP)

### Open competition for stream ciphers

<http://www.ecrypt.eu.org>

- run by ECRYPT
  - high performance in **software** (32/64-bit): 128-bit key
  - low-gate count **hardware** (< 1000 gates): 80-bit key
  - variants: authenticated encryption
- 29 April 2005: 33 submissions
- Many broken in first year
- End of competition: April 2008

### Open competition: Feb. 2007 status

SW Phase 3	HW Phase 3
CryptMT	DECIM
DRAGON	<del>Edon-80</del>
HC-128 (-256)	F-FCSR
LEX	Grain
NLS (encrypt only)	MICKEY (-128)
Rabbit	MOUSTIQUE
Salsa20	POMARANCH
SOSEMANUK	Trivium

3-10 cycles per byte      1500..3000 gates

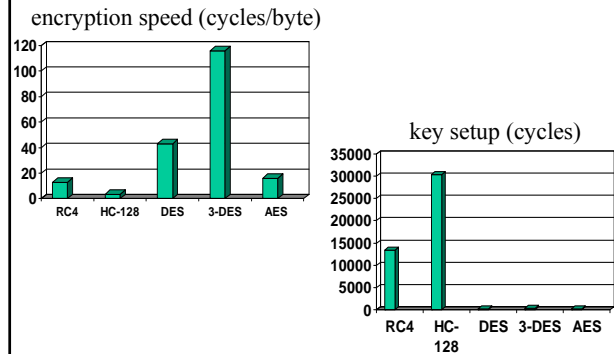
### The eSTREAM Portfolio Apr. 2008 (Rev1 Sept. 2008)

Software	Hardware
HC-128	<del>F-FCSR-H</del>
Rabbit	Grain v1
Salsa20/12	MICKEY v2
Sosemanuk	Trivium

(In alphabetical order)

49

### Performance reference data (Pentium M 1.70GHz Model 6/9/5)



50

### Stream ciphers: conclusions

- Only for niche areas
  - High speed
  - Low cost
- Substantial progress has been made in the last years, but more research is needed: design close to the edge
- Example: cube attack by Shamir

51

### Lightweight crypto

- SQUASH [Shamir07] – Crypto rump session
  - MAC algorithm for authentication in RFID chips
  - only 500 gates
  - security is related to modular squaring (Rabin cryptosystem)
- PRESENT [Bogdanov07] – CHES 2007
  - 64-bit block cipher for RFID chips
  - only 1750 gates (compare to 3600 for AES)

Stream cipher: because of time-memory trade-offs, for 80-bit security one needs 160-bit memory which costs 1000 gates

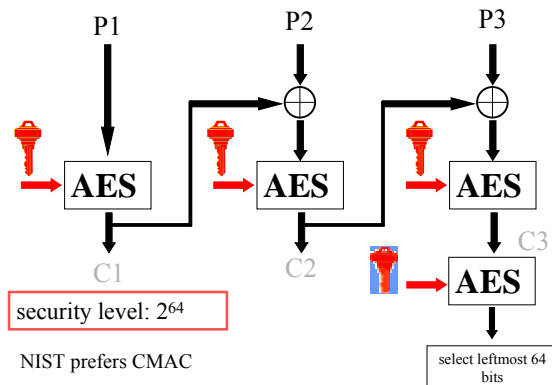
52

### MAC Algorithms

- CBC-MAC: EMAC and CMAC
- HMAC
- GCM and GMAC
- UMAC
- Authenticated encryption

53

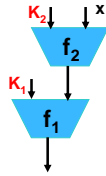
### CBC-MAC based on AES (EMAC)



54

### HMAC based on MDx, SHA

- Widely used in SSL/TLS/IPsec
- Attacks not yet dramatic
- NMAC weaker than HMAC



	Rounds in f1	Rounds in f2	Data complexity
MD4	48	48	$2^{88}$ CP & $2^{95}$ time
MD5	64	33 of 64	$2^{126}$ CP
MD5	64	64	$2^{51}$ CP & $2^{100}$ time (RK)
SHA(-0)	80	80	$2^{109}$ CP
SHA-1	80	43 of 80	$2^{154.9}$ CP

55

### GMAC: polynomial authentication code (NIST SP 800-38D 2007 + 3GSM)

- keys  $K_1, K_2 \in GF(2^{128})$
  - input  $x: x_1, x_2, \dots, x_t$ , with  $x_i \in GF(2^{128})$
- $$g(x) = K_1 + \sum_{i=1}^t x_i \cdot (K_2)^i$$
- in practice: compute  $K_1 = AES_K(n)$  (CTR mode)

- properties:
  - fast in software and hardware (support from Intel/AMD)
  - not very robust w.r.t. nonce reuse, truncation, MAC verifications, due to reuse of  $K_2$  (not in 3GSM!)
  - versions over GF(p) (e.g. Poly1305-AES) seem more robust

56

### UMAC RFC 4418 (2006)

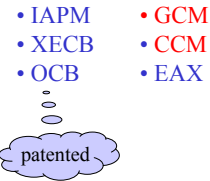
- key  $K, k_1, k_2, \dots, k_{256} \in GF(2^{32})$  (1024 bytes)
  - input  $x: x_1, x_2, \dots, x_{256}$ , with  $x_i \in GF(2^{32})$
- $$g(x) = \text{prf}_K(h(x))$$
- $$h(x) = \left( \sum_{i=1}^{512} (x_{2i-1} + k_{2i-1}) \bmod 2^{32} \cdot (x_{2i} + k_{2i}) \bmod 2^{32} \right) \bmod 2^{64}$$
- properties
    - software performance: 1-2 cycles/byte
    - forgery probability:  $1/2^{30}$  (provable lower bound)
    - [Handschuh-Preneel08] full key recovery with  $2^{40}$  verification queries (no nonce reuse needed!)

57

### Authenticated encryption

- Needed for network security, but only fully understood by crypto community around 2000 (too late)
- Standards have been selected recently:
  - CCM: CTR + CBC-MAC [NIST SP 800-38C]
  - GCM: CTR + GMAC [NIST SP 800-38D]
- Both are suboptimal

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



58

### MAC algorithms: conclusions

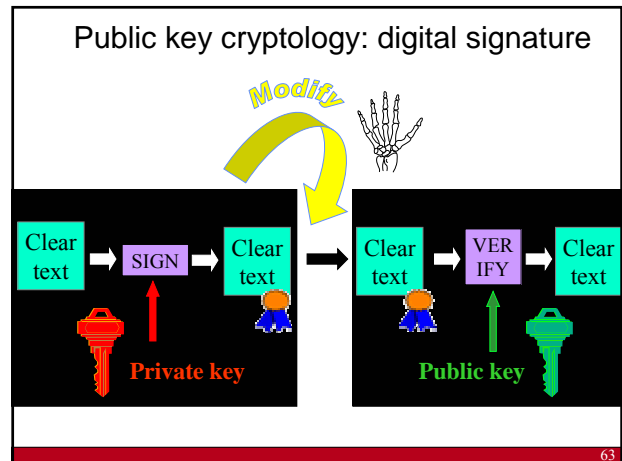
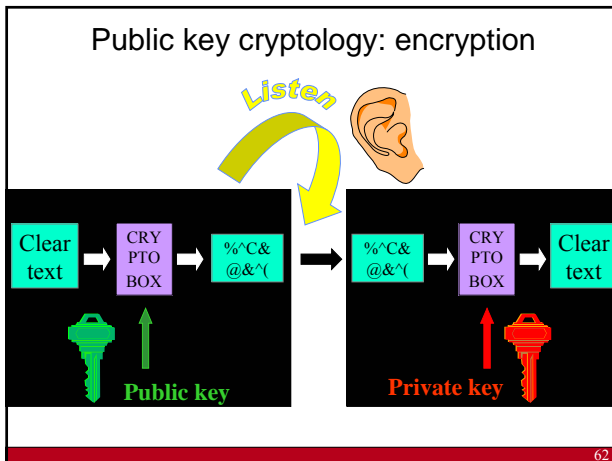
- can get better performance than encryption
- EMAC (CBC-MAC) seems fine
- widely used choices lack robustness
- Modes for authenticated encryption better understood but not widely deployed

59

### Outline

- Cryptographic algorithms
  - Block ciphers
  - Hash functions
  - Stream ciphers
  - MAC algorithms
  - Public key encryption and digital signatures
- Cryptographic protocols
- Implementations issues
- Research challenges

61

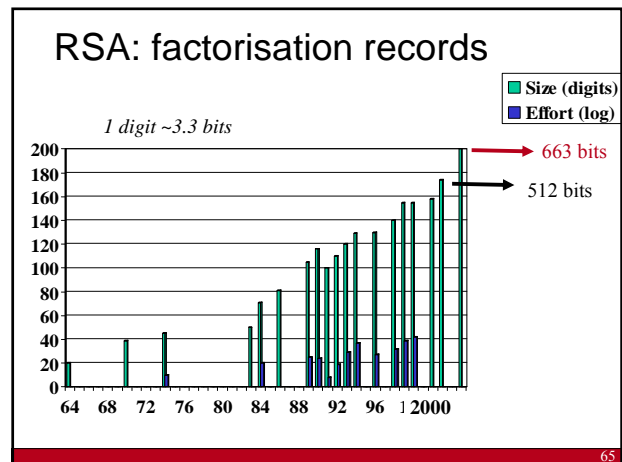


### RSA problems

- 2 large primes  $p$  and  $q$
- modulus  $n = p \cdot q$
- compute  $\lambda(n) = \text{lcm}(p-1, q-1)$
- choose  $e$  relatively prime w.r.t.  $\lambda(n)$
- compute  $d = e^{-1} \pmod{\lambda(n)}$
- public key =  $(e, n)$
- private key =  $d$  of  $(p, q)$
- encryption:  $c = x^e \pmod n$
- decryption:  $x = c^d \pmod n$

- Is factoring hard?
- Is the RSA problem, i.e. inverting  $f(x) = x^e \pmod n$  as hard as factoring?
- How to use RSA efficiently, that is, how to prove the forging a signature or learning any additional information on the plaintext from the ciphertext results in an *efficient* algorithm to solve the RSA problem
  - RSA KEM-DEM for encryption
  - RSA PSS for signature
- How to get rid of Random Oracle Model?

64



### Factorisation

- New record in May 2005: 663 bits (or 200 digits) using NFS
- New record in May 2007:  $2^{1039}-1$  (313 digits) using SNFS
- hardware factoring machine: **TWIRL** [TS'03] (The Weizmann Institute Relation Locator)
  - initial R&D cost of ~\$20M
  - 512-bit RSA keys can be factored with a device costing \$5K in about 10 minutes
  - 1024-bit RSA keys can be factored with a device costing \$10M in about 6 weeks
- ECRYPT statement on key lengths and parameters <http://www.ecrypt.eu.org>
  - 768-bit factorization in 2008 and 896-bit factorization in 2010

66

### Elliptic and Hyperelliptic Curve Cryptology

Example:  
the elliptic curve  $y^2 = x^3 - 7x + 6$   
over the field of real numbers  $R$

Advantage: shorter key lengths

67

### Key lengths for confidentiality

<http://www.ecrypt.eu.org>

duration	symmetric	RSA	ECC
days/hours	50	512	100
5 years	73	1024	146
10-20 years	103	2048	206
30-50 years	141	4096	282

Assumptions: no quantum computers; no breakthroughs; limited budget

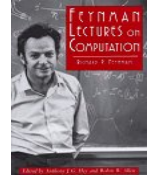
68

### New computational models: quantum computers?

- exponential parallelism  $n$  coupled quantum bits  
 $2^n$  degrees of freedom !



- Shor 1994: perfect for factoring
- But: can a quantum computer be built?



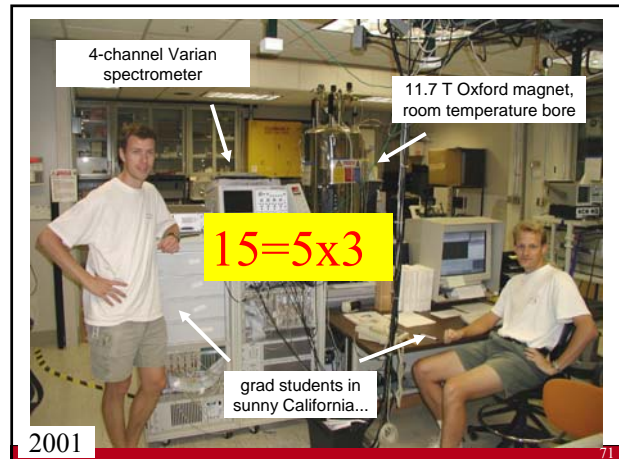
69

### If a large quantum computer can be built...

- All schemes based on factoring (such as RSA) will be insecure
- Same for discrete log (ECC)
- Symmetric key sizes: x2
- Hash sizes: x1.5
- Alternatives: McEliece, HFE, NTRU, ...
- So far it seems very hard to match performance of current systems while keeping the security level against conventional attacks



70



71

### News on 13 Sept. 2007

- "Two independent teams (led by Andrew White at the University of Queensland in Brisbane, Australia, and the other by Chao-Yang Lu of the University of Science and Technology of China, in Hefei) have implemented Shor's algorithm using rudimentary laser-based quantum computers"
- Both teams have managed to factor 15, again using special properties of the number

### News on 19 Dec. 2007

- optical quantum computer (team led by Daniel James, University of Toronto)
- factored 15

72

### Public key: conclusions

- essential for large open networks
- not suitable for bulk data
- widely deployed systems depend on a small set of mathematical problems
- long term security is an issue

73

### Key establishment protocols

- The problem
- How to establish secret keys using secret keys?
- How to establish secret keys using public keys?
  - Diffie-Hellman and STS
- How to distribute public keys? (PKI)

74

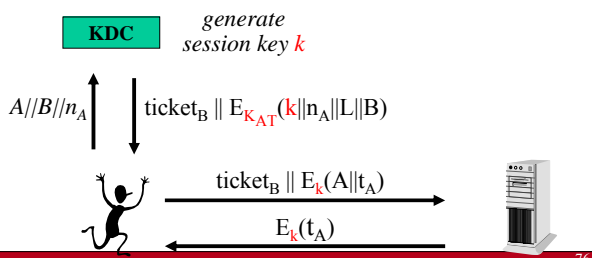
### Key establishment: the problem

- Cryptology makes it easier to secure information, by replacing the security of information by the security of **keys**
- The main problem is how to establish these **keys**
  - 95% of the difficulty
  - integrate with application
  - if possible transparent to end users
  - but need entity authentication

75

### Symmetric key distribution with 3rd party (KDC Key Distribution Center) - Kerberos

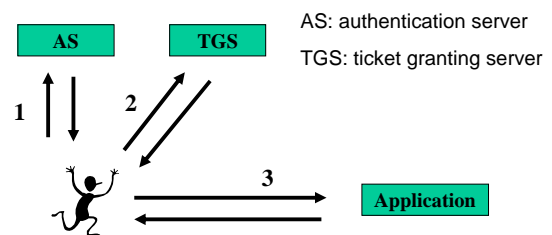
- Alice/Bob shares a long term secret with KDC:  $K_{AT}/K_{BT}$
- Alice/Bob/KDC have synchronized clocks
- $ticket_B = E_{K_{AT}}(k || n_A || L || B)$
- L life time of a ticket – limits validity of a key



76

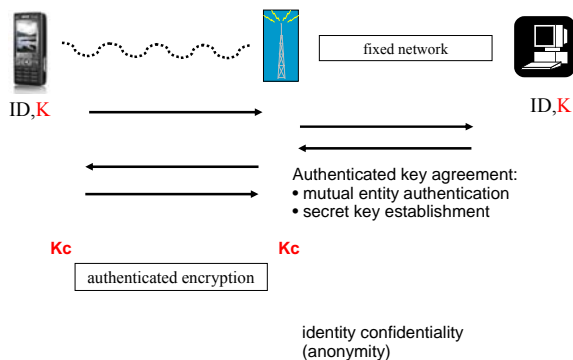
### Kerberos/Single Sign On (SSO)

- Alice's long term key  $K_{AT}$  is derived from a password  $P$
- Alice stores  $E_{K_{AT}}(k||n_A||L||B)$  on her disk for the period L
- To avoid one password entry per application: use intermediate server (ticket granting server)



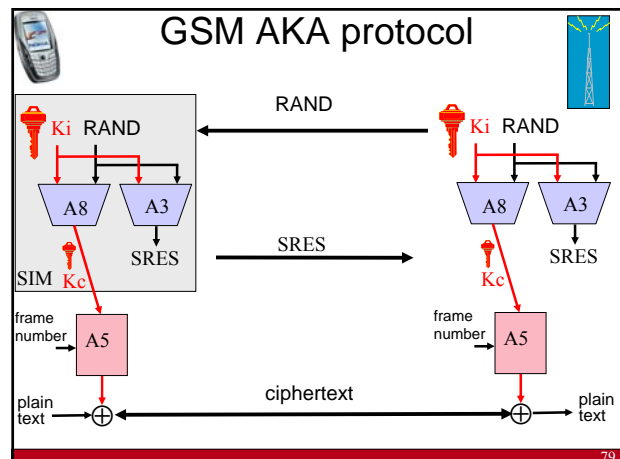
77

### Wireless security: principles

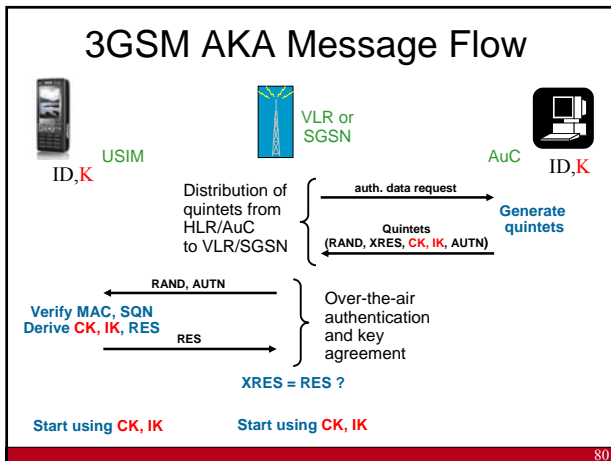


78

### GSM AKA protocol



79



### AKA Variables and Functions

RAND = random challenge generated by AuC  
 XRES =  $f_{2_K}(\text{RAND})$  = expected user response computed by AuC  
 RES =  $f_{2_K}(\text{RAND})$  = actual user response computed by USIM  
 CK =  $f_{3_K}(\text{RAND})$  = cipher key  
 IK =  $f_{4_K}(\text{RAND})$  = integrity key  
 AK =  $f_{5_K}(\text{RAND})$  = anonymity key  
 SQN = sequence number  
 AMF = authentication management field  
 MAC =  $f_{1_K}(\text{SQN} \parallel \text{RAND} \parallel \text{AMF})$  = message authentication code computed over SQN, RAND and AMF  
 AUTN =  $\text{SQN} \oplus \text{AK} \parallel \text{AMF} \parallel \text{MAC}$  = network authentication token, concealment of SQN with AK is optional  
 Quintet = (RAND, XRES, CK, IK, AUTN)

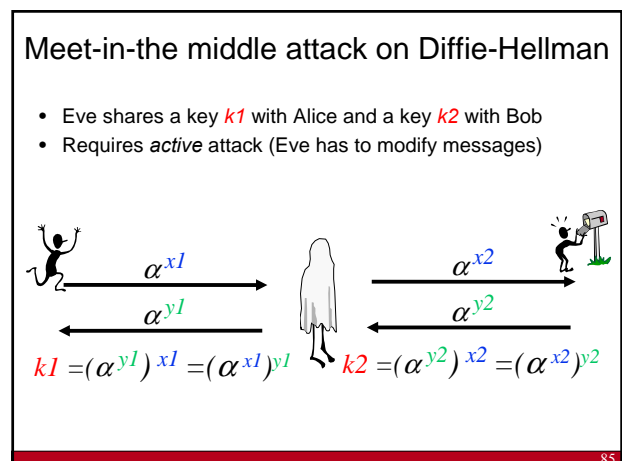
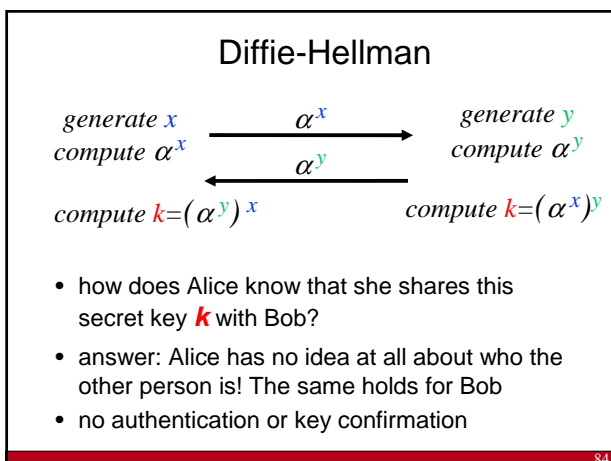
Anonymity key (=optional) hides SQN, which could be used for tracking

### Length of AKA Cryptographic Parameters

• K	128 bits	
• RAND	128 bits	
• RES	32-128 bits	
• CK	128 bits	
• IK	128 bits	
• AUTN	128 bits	
- SQN	Sequence number	48 bits
- AMF	Authentication management field	16 bits
- MAC	Message authentication code	64 bits

### Key agreement with asymmetric cryptography

- Diffie–Hellman & variants
- Station to Station
- all calculations are done modulo a large (safe) prime  $p$  with generator  $\alpha$



### Station to Station protocol (STS)

- The entity authentication problem can be fixed by adding digital signatures
- This protocol plays a very important role on the Internet (under different names)

86

### IKE - Main Mode with Digital Signatures

K derived from master = prf(N\_i || N\_r, g^s)

SIG\_i = Signature on H(master, g^s || g^r || ... || ID\_i)

SIG\_r = Signature on H(master, g^r || g^s || ... || ID\_r)

H is equal to prf or the hash function tied to the signature algorithm (all inputs are concatenated)

87

### STS properties

- mutual explicit key authentication
- mutual entity authentication
- mutual key confirmation
- anonymity (unless certificates are exchanged in the beginning)
- (perfect) **forward secrecy**
- no problem if k leaks (known key attack)

88

### Protocols (1)

- key transport (email)
- authenticated key agreement (TLS, SSH, GSM, UMTS)
- time-stamping
- notarisation
- credentials (TPM)
- anonymous communication
- e-cash
- voting
- auctions
- threshold cryptography
- robust networking

89

### Protocols (2)

- multi-party computation
- threshold crypto
- privacy protecting data mining
- social and group crypto

decryption based on location and context

distance bounding

“you can trust it because you don’t have to”

90

### TLS 1.2 Data Encapsulation Options

(1.0: Jan '99 – 1.1: Apr'06 - 1.2: Aug'08)

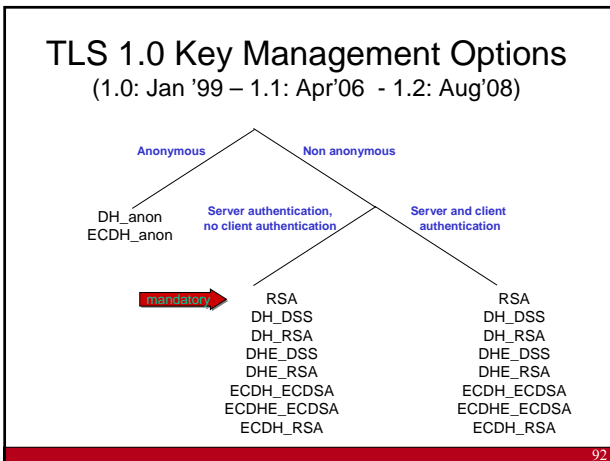
Integrity			
key size	128	160	160
algorithm options	AES-XCBC-MAC HMAC-MD5	HMAC-SHA	HMAC-SHA256

mandatory

Confidentiality			
key size	128	168	256
algorithm options	RC4 AES_CBC (AES_CTR) CAMELLIA_CBC SEED_CBC	3DES_EDE_CBC	AES_CBC CAMELLIA_CBC

mandatory

91



- ### TLS evolution
- 2002: AES
  - 2005: Camellia + PSK (pre-shared key variants)
  - 2006: ECC
  - 2008: PRF based on SHA-256

### IKE Algorithm Selection

#### Mandatory Algorithms

Algorithm Type	IKE v1	IKE v2
Payload Encryption	DES-CBC	3DES_CBC (AES_128_CBC)
Payload Integrity	HMAC-MD5 HMAC-SHA1	HMAC-SHA1
DH Group	768 Bit	1024 (2048) Bit
Transfer Type 1 (Encryption)	ENCR_DES_CBC	ENCR_3DES (ENCR_AES_128_CBC)
Transfer Type 2 (PRF)	PRF_HMAC_SHA1 [RFC2104]	PRF_HMAC_SHA1 [RFC2104]
Transfer Type 3 (Integrity)	AUTH_HMAC_SHA1_96 [RFC2404]	AUTH_HMAC_SHA1_96 [RFC2404]

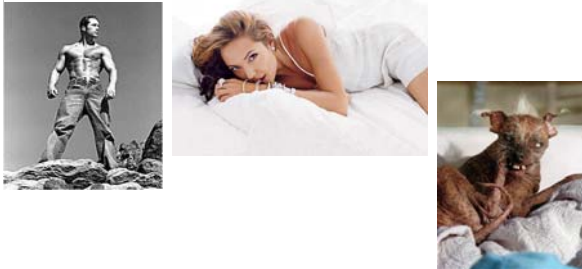
Source: RFC 4384, April 07

- ### NSA Suite B (2005)
- (see also RFC 4869 – April 07)
- Block cipher: AES-CBC (128 + 256-bit keys)
  - MAC algorithms: GMAC (128 + 256-bit keys)
  - Authenticated encryption: AES-GCM (128 + 256-bit keys)
  - Digital signatures: ECDSA (256 + 384-bit prime moduli)
  - Key agreement: ECDH (256 + 384-bit prime moduli)
  - Hashing: SHA-256 and SHA-384
- Note: NSA has licensed the rights to 26 patents held by Certicom Inc. covering a variety of elliptic curve technology.

- ### Protocols: conclusions
- more modularity and less complexity would be desirable, but large body of legacy standards and code
  - advanced protocols can bring added value from the simple (password-based AKE) to more complex multi-party interactions

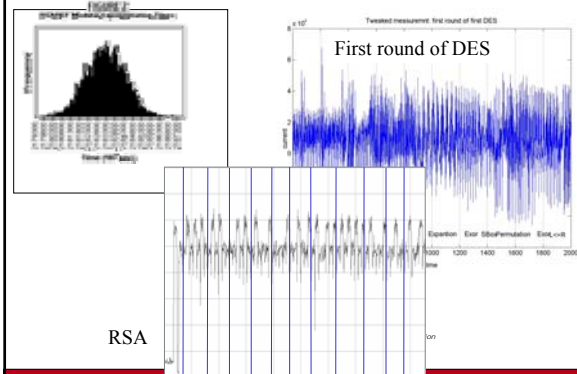
- ### Outline
- Cryptographic algorithms
    - Block ciphers
    - Hash functions
    - Stream ciphers
    - MAC algorithms
    - Public key encryption and digital signatures
  - Cryptographic protocols
  - Implementations issues
  - Research challenges

## Models and reality



98

## Implementations: side channel attacks



99

## Implementation attacks

- measure: time, power, electromagnetic radiation, sound
- introduce faults
- bug attacks in hardware
- combine with statistical analysis and cryptanalysis
- software: reaction attacks and API attacks
- **major impact on implementation cost**

Sun Tzu, The Art of War:  
In war, avoid what is strong and attack what is weak

100

## Some comments [Shamir, Kocher]

- defending a smart card is like defending an embassy
- protection against implementation attacks is like airport security
  - each attack utilized a completely different approach
  - each countermeasure works only against a specific attack
  - we have no way to predict the next attack and protecting against all conceivable attacks is impossible
- unlike conventional cryptanalysis, implementation attacks are very efficient
- new attacks will emerge

need for better modeling and theory  
we may have to return to security through obscurity

101

## Latest implementation attacks

- Power attack on PC through USB port
  - even if USB port is disabled by software
  - **even if fuse is blown!**
- Power attacks on passive RFID tags (monitoring reflected field)
- Cache attacks on AES
- Branch prediction attacks on RSA
- Acoustic attacks on PCs

Lars Knudsen: “It’s not cryptanalysis, it’s vandalism”

102

## Maybe rethink how we design algorithms

- **BAD**
  - iterate a simple round
  - large S-boxes
  - complex operations
- **GOOD**
  - more parallelism
  - use large chunks of key and data in every step
  - avoid cache memories
  - special instructions in microprocessors

103

### Reaction attacks: chosen ciphertext security

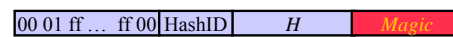
- [Bleichenbacher98] PKCS #1v1.5 – 1 million chosen ciphertexts
- [Klima-Pokorny-Rosa03] improved previous attack
- [Manger01] improved attack on OAEP PKCS #1v2 – a few thousand chosen ciphertexts
- [Bellare-Kohno-Namprempre 02]: SSH
- [Canvel-Hiltgen-Vaudenay-Vuagnoux03]: SSL/TLS

104

### RSA Signatures: PKCS #1 v1.5 [source: RSA Labs]

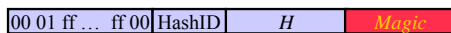


Most signature verification software would accept a signature on M of the following form:



105

### Attack on PKCS #1 v1.5 implementations [Bleichenbacher06]



- Consider RSA with public exponent 3
- For any hash value H, it is easy to compute a string "Magic" such that the above string is a perfect cube of 3072 bits
- Consequence:
  - One can sign any message (H) **without knowing the private key**
  - This signature works **for any public key** that is longer than 3072 bits
- Vulnerable: OpenSSL, Mozilla NSS, GnuTLS
- Fix
  - Write proper verification code (but the signer cannot know which code the verifier will use)
  - Use a public exponent that is at least 32 bits long
  - Upgrade – finally – to RSA-PSS

106

### Outline

- Cryptographic algorithms
  - Block ciphers
  - Hash functions
  - Stream ciphers
  - MAC algorithms
  - Public key encryption and digital signatures
- Cryptographic protocols
- Implementations issues
- Research challenges

107

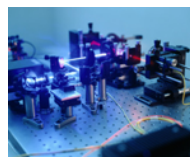
### Challenges for long term security

- cryptanalysis improves:
  - mathematical attacks A5/1, E0, MD5, SHA-1
  - implementation attacks
- computational power increases:
  - Moore's law
  - exponential progress with quantum computers?
- environment changes – new assumptions
  - packet switched networking
  - open networks
  - dynamic networks
  - untrusted nodes
  - ratio power CPU/memory size
  - outsourcing of data processing

108

### Quantum cryptography

- <http://www.secoqc.net/>
- Security based
  - **on the assumption that the laws of quantum physics are correct**
  - **rather than on the assumption that certain mathematical problems are hard**



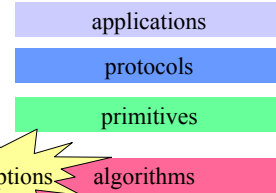
109

## Quantum cryptography

- no solution for entity authentication problem (bootstrapping needed with secret keys)
- no solution (yet) for multicast
- dependent on physical properties of communication channel
- cost
- implementation weaknesses (side channels)

110

## Layers



Proofs: link security at different levels in a quantitative way

L.R. Knudsen:  
"If it is provably secure, it is probably not"

111

## Assumptions

research on **hard problems?**

James L. Massey:  
A hard problem is one that nobody works on

good lower bounds  
average versus worst case  
find new hard problems

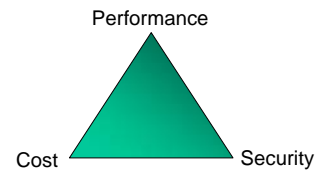
112

## Challenges for cryptographic algorithms

- security for 50-100 years
- authenticated encryption of Terabit/s networks
- ultra-low power/footprint

secure software and hardware implementations

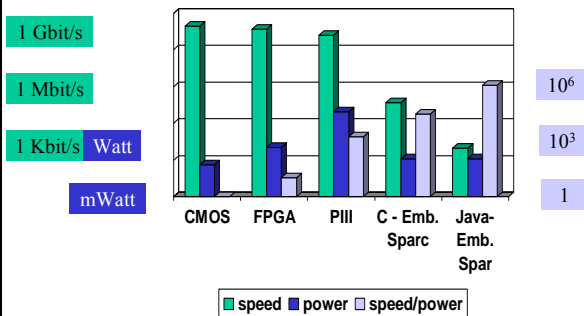
Algorithm agility



113

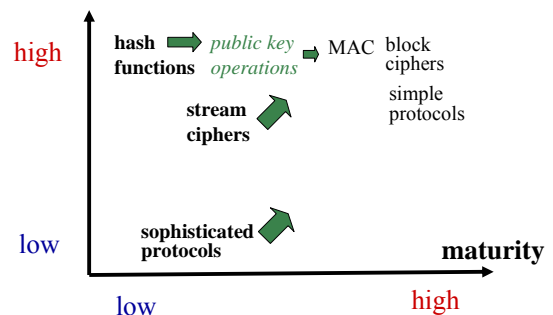
## The power challenge:

AES-128 speed/power for various platforms (Gb/Joule)

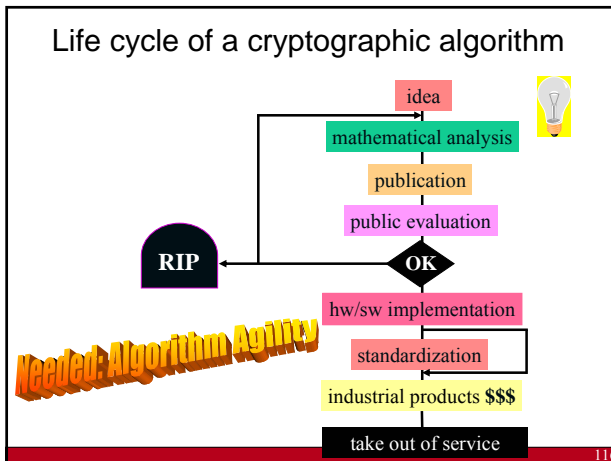


114

## demand in applications



115



- ### Challenges for advanced crypto
- privacy enhancing technologies
  - linking crypto with physical world
    - biometrics, physically unclonable functions
  - distributed secure execution
  - whitebox cryptography
  - cryptography in the encrypted domain
    - searching in encrypted databases – data mining on health care data
    - zero knowledge watermarking – intelligent media sharing
  - perceptual hashing
  - crypto for nanotechnology
- 115

- ### Conclusions
- The “security problem” is not solved
    - Many challenging problems ahead...
    - Make sure that you can upgrade your crypto algorithm and protocol
    - Bring advanced cryptographic protocols to implementations
- When will the IACR hold its elections on-line?  
When will everyone pay with e-cash?  
Can we reconcile privacy, DRM and data mining?
- 118