ECRYPT

http://www.ecrypt.eu.org

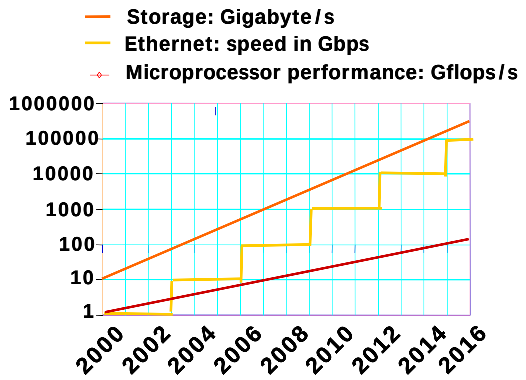
Cryptographic Algorithms for Network Security: Failures, Success and Challenges

Prof. Bart Preneel
COSIC, K.U.Leuven, Belgium
Bart.Preneel(at)esat.kuleuven.be
http://homes.esat.kuleuven.be/~preneel
September 2010

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)

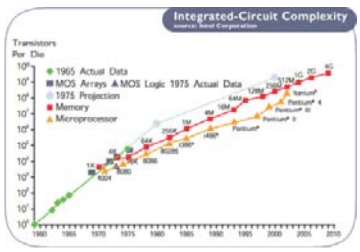
Moore's Law: computation/storage 2000-2020



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$



Information processing

Everything is always connected everywhere

Continuum between software and hardware
ASIC (microcode) – FPGA – fully programmable processor

Disclaimer: cryptography ≠ 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

[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.

7

Research ↔ Practice

DES, RSA, DH, CBC-MAC
Provable security (PKC), ZK, ElGamal, ECC, stream ciphers
Quantum crypto
MD4, MD5
Provable security (SKC)
Key escrow
Quantum cryptanalysis
How to use RSA?
Alternatives to RSA
PKI
AES
ID-Based Crypto

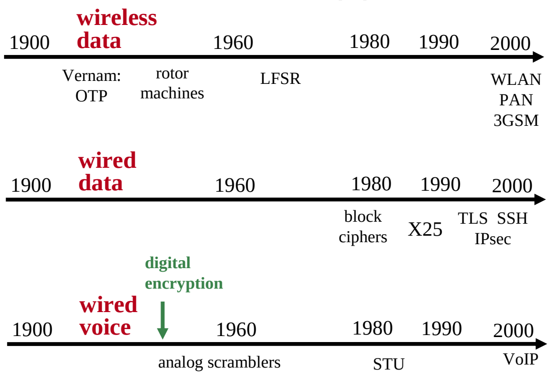
HARDWARE (70, 80)
Limited (govt+financial sector)
DES, 3DES

SOFTWARE (90)
GSM, PGP
C libraries (RSA, DH)
SSL/TLS, IPsec, SSH, S/MIME
Java crypto libraries
WLAN

EVERYWHERE
Trusted computing, DRM, 3GPP, RFID, sensor nodes
…

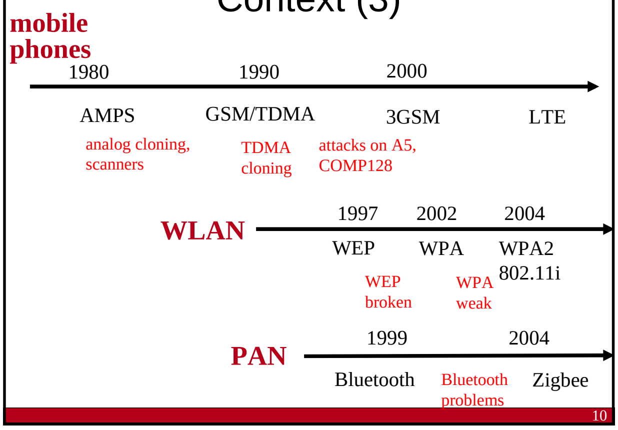
8

Context (2)



9

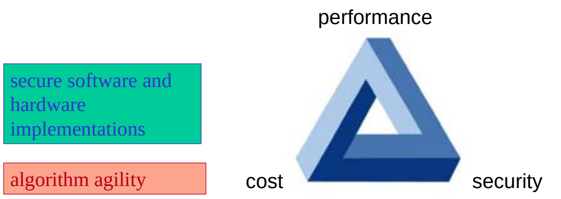
Context (3)



10

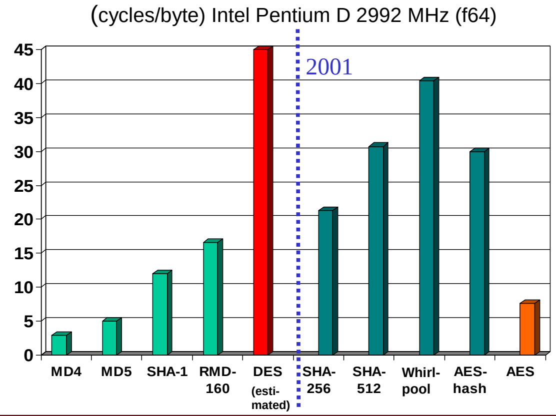
Challenges for crypto

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



11

Performance of hash functions - Bernstein
(cycles/byte) Intel Pentium D 2992 MHz (f64)



12

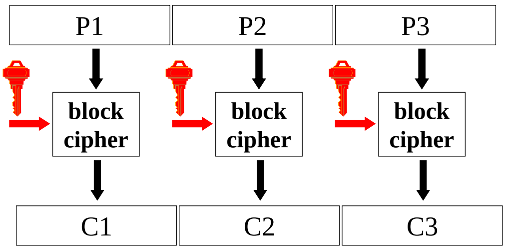
What to remember from the algorithms and protocols

- Always authenticated encryption (and not GCM)
- Dump hash functions except for applications where you really need them (digital signatures)
- Public key algorithms and protocols still a bottleneck for performance and security

Outline

- Cryptographic algorithms
  - Block ciphers
  - Hash functions
  - Stream ciphers
  - MAC algorithms
  - Public key algorithms and protocols
- Research challenges

Block cipher



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

Block ciphers

64-bit block

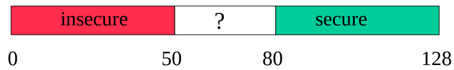
DES (56)
3-DES (112-168)
IDEA (128)
GOST (128)
MISTY1 (128)
KASUMI (128 in 3G, 64 in 2G)
HIGHT (128)
PRESENT (80-128)
TEA (128)
mCRYPTON (128)
KATAN (80)

128-bit block

AES (128-192-256)
CAMELLIA
RC6
CLEFIA

56 bits: 4 seconds with $5M
80 bits: 2 year with $5M
128 bits: 256 billion years with $5B

Symmetric key lengths

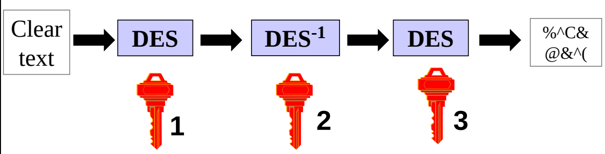


3-DES: NIST Spec. Pub. 800-67

(May 2004)

extremely vulnerable to a related key attack

- single DES abandoned (56 bit)
- double DES not good enough (72 bit)
- 2-key triple DES: until 2009 (80 bit)
- 3-key triple DES: until 2030 (100 bit)



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: 1457 implementations
    - http://csrc.nist.gov/groups/STM/cavp/documents/aes/aesval.html
- 2003: AES-128 also for classified information and AES-192/-256 for secret and top secret information!
- security:
  - algebraic attacks of [Courtois+02] not effective
  - side channel attacks: cache attacks on unprotected implementations

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

AES implementations: efficient/compact

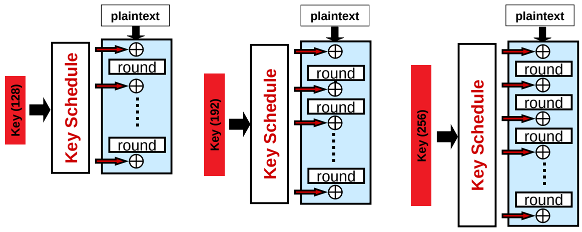
- software
  - 7.6 cycles/byte on Core 2 or 110 Mbyte/s bitsliced [Käsper-Schwabe'09]
- co-processor in Intel Westmere
  - new AES instruction: 0.75 cycles/byte ['09-'10]
- hardware
  - fast 43 Gbit/s in 130 nm CMOS ['05]
  - most compact: 3600 gates
    - PRESENT: 1029, KATAN: 1054; GOST: 650; CLEFIA: 4950

AES variants (2001)

- AES-128
- 10 rounds
- sensitive

- AES-192
- 12 rounds
- classified

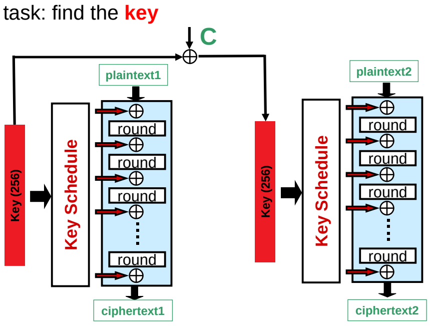
- AES-256
- 14 rounds
- secret/top secret



lightweight key schedule, in particular for the 256-bit version

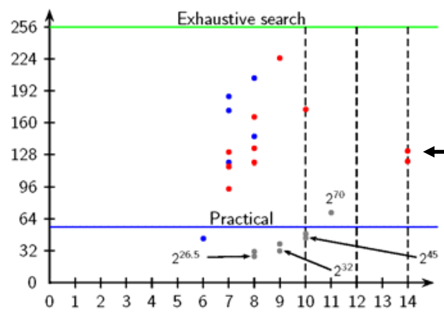
What is a related key attack?

- attacker chooses **plaintexts** and **key difference** C
- attacker gets **ciphertexts**
- task: find the **key**



AES-256

[Biryukov-Khovratovich'09]
[Biryukov-Dunkelman-Keller-Khovratovich-Shamir'09]



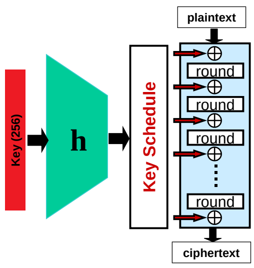
Related key attack: 4 keys, data & time complexity $2^{119} << 2^{256}$

KASUMI A5/3
4 related keys,
2^{26} plaintexts,
2^{30} bytes mem.
2^{32} time

Slide credit: Orr Dunkelman

Should I worry about a related key attack?

- very hard in practice (except for control vector and some old US banking schemes)
- if you are vulnerable to a related key attack, you are making very bad implementation mistakes
- this is a very powerful attack model: if an opponent can zeroize (= AND 0) 224 key bits of his choice (rather than ⊕ C) he can find the key in a few seconds for **any** cipher with a 256-bit key
- if you are worried, hashing the key is an easy fix



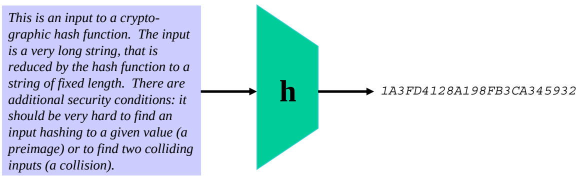
Block ciphers: conclusions

- 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
  - no justification for encryption without authentication – should be abandoned
  - efficient modes for authenticated encryption

Hash functions

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

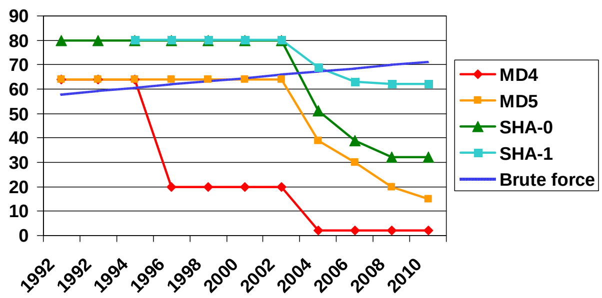
- collision resistance
- preimage resistance
- 2nd preimage resistance



25

The complexity of collision attacks

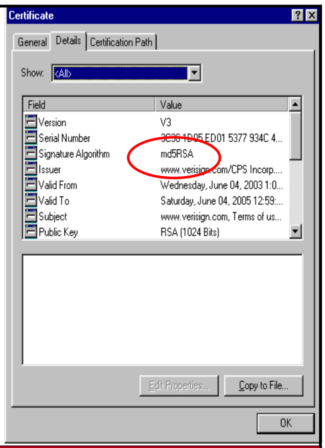
Brute force: 4 million PCs or US$ 100K hardware (1 year)



26

MD5

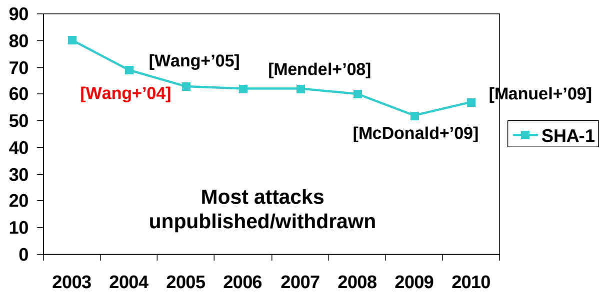
- Advice (RIPE since '92, RSA since '96): stop using MD5
- Largely ignored by industry (click on a cert...)
- Collisions for MD5
  - brute force (2^{64}): 1M$ 6 hours in 2010
  - [Wang+'04] collision in 15 minutes on a PC
  - [Stevens+'09] collisions in **milliseconds**
- 2nd preimage:
  - [Sasaki-Aoki'09] 2^{123}



27

SHA-1

- SHA designed by NIST (NSA) in '93
- redesign after 2 years ('95) to SHA-1



Prediction: collision for SHA-1 in the next 12-18 months

28

Hash function attacks:

cryptographic ***meltdown*** yet with limited impact

- collisions problematic for future
  - digital signatures for non-repudiation (cf. traffic tickets in Australia?)
- 2nd preimage:
  - MD2: 2^{73} [Knudsen+09]
  - MD4: $2^{97}/2^{70}$ with precomputation [Rechberger+10]
  - MD5: 2^{123} [Sasaki-Aoki'09]
  - SHA-1: 48/80 steps in $2^{159.3}$ [Aoki-Sasaki'09]

- RIPEMD-160 seems more secure than SHA-1 ☺
- use more recent standards (slower and larger)
  - SHA-2 (SHA-256, SHA-224,…SHA-512)
  - SHA-3?

29

Hash function attacks: impact

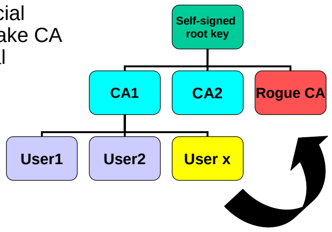
- High profile attack on CAs in December 2008
- TLS/SSL has been designed for algorithm negotiation and flexible upgrades
  - …but the negotiation algorithm uses MD5 || SHA-1
  - negotiation cannot be upgraded without changing the standard: TLS 1.1 -> 1.2
  - brings serious cost: no upgrade until there is an economic attack
- HMAC:
  - HMAC-MD4: replace it
  - HMAC-MD5 not recommended
  - HMAC-SHA-1 ok

30

Rogue CA attack

[Sotirov-Stevens-Appelbaum-Lenstra-Molnar-Osvik-de Weger '08]

- request user cert; by special collision this results in a fake CA cert (need to predict serial number + validity period)
- impact: **rogue CA** that can issue certs that are trusted by all browsers

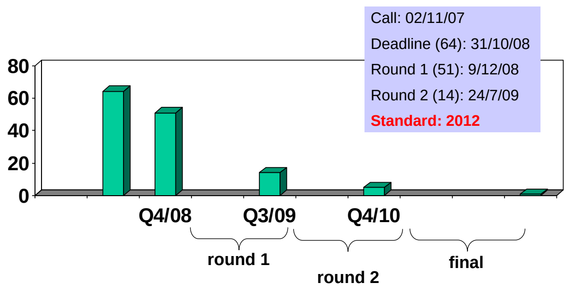


- 6 CAs have issued certificates signed with MD5 in 2008:
  - Rapid SSL, Free SSL (free trial certificates offered by RapidSSL), TC TrustCenter AG, RSA Data Security, Verisign.co.jp

Hash function status today

NIST AHS competition (SHA-3)

- SHA-3 must support 224, 256, 384, and 512-bit message digests, and must support a maximum message length of at least **2^64** bits



The Candidates

Preliminary Cryptanalysis

Round 2 Candidates

Hash functions: conclusions

- cryptographic meltdown but fortunately implications so far limited
- designers often too optimistic (usually need 2x more rounds)
- other weaknesses have been identified in general approach to construction hash functions
- SHA-2 and SHA-3 will co-exist
- SHA-4: probably not before 2030

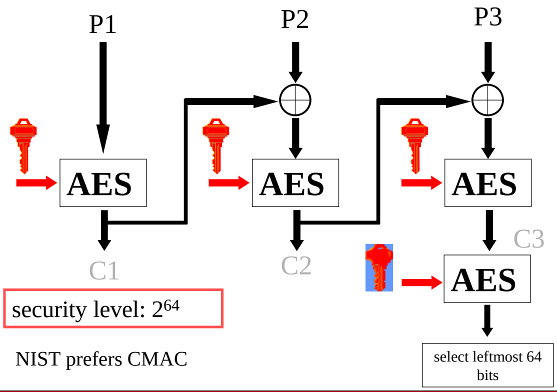
37

MAC Algorithms

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

38

CBC-MAC based on AES (EMAC)

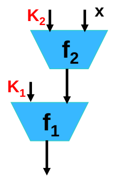


NIST prefers CMAC

39

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

40

GMAC: polynomial authentication code

(NIST SP 800-38D 2007 + 3GSM)

- keys $K_1, K_2 \in GF(2^{128})$
- input x: $x_1, x_2, \ldots, x_t$, with $x_i \in GF(2^{128})$

$$g(x) = K_1 + \Sigma_{i=1}^{t}\ x_i \bullet (K_2)^i$$

- in practice: compute $K_1 = \text{AES}_K(n)$ (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

41

UMAC RFC 4418 (2006)

- key $K, k_1, k_2 \ldots, k_{256} \in GF(2^{32})$ *(1024 bytes)*
- input x: $x_1, x_2, \ldots, x_{256}$, with $x_i \in GF(2^{32})$

$g(x) = prf_K(h(x))$

$h(x) = \left(\Sigma_{i=1}^{512}\ (x_{2i-1} + k_{2i-1}) \bmod 2^{32} \cdot (x_{2i} + k_{2i}) \bmod 2^{32} \right) \bmod 2^{64}$

- 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!)

42

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

- IAPM
- XECB
- OCB

patented

- GCM
- CCM
- EAX

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
  - only 5-30% slower than encryption only
  - GCM should be fixed

Outline

- Cryptographic algorithms
  - Block ciphers
  - Hash functions
  - Stream ciphers
  - MAC algorithms
  - Public key algorithms and protocols
- Research challenges

RSA problems

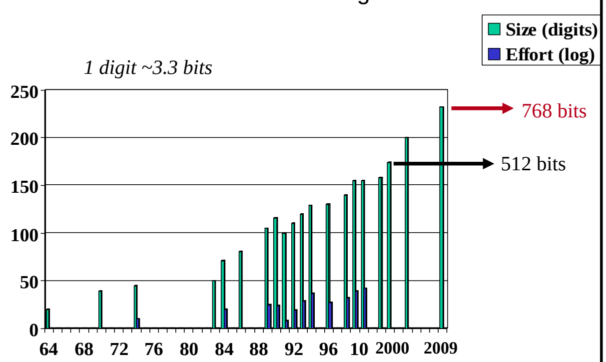
- 2 large primes p and q
- modulus n = p.q
- compute $\lambda(n) = \mathrm{lcm}(p-1,q-1)$
- choose e relatively prime w.r.t. $\lambda(n)$
- compute $d = e^{-1} \bmod \lambda(n)$

- public key = (e,n)
- private key = d of (p,q)
- encryption: $c = x^e \bmod n$
- decryption: $x = c^d \bmod n$

- Is factoring hard?
- Is the RSA problem, i.e, inverting $f(x) = x^e \bmod n$ as hard as factoring?
- Can we show that forging a signature implies factoring (and this without the Random Oracle assumption)?

Factorisation records

2009: 768 bits or 232 digits



Factorisation

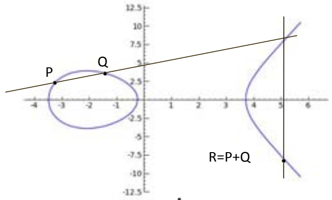
- New record in 2009: 768 bits (or 231 digits) using NFS
- New record in May 2007: $2^{1039}-1$ (313 digits) using SNFS
- hardware factoring machine: TWIRL [TS'03] (The Weizmann Institute Relation Locator)
  - initial R&D cost of ~$20M
  - 512-bit RSA keys can be factored with a device costing $5K in 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

896-bit factorization in 2012, 1024-bit factorization in 2020?

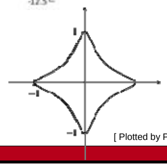
Elliptic curve cryptography

Elliptic curve : E: $y^2=x^3-13x-3$

Point multiplication:
$rP = \underbrace{P + P + \ldots + P}_{r}$



Edwards curve : E: $x^2 + y^2 = 1 - 30x^2y^2$



[Plotted by P. Schwabe]

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

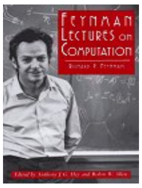
Assumptions: no quantum computers; no breakthroughs; limited budget

New computational models: quantum computers?

- exponential parallelism

n coupled quantum bits
⇩
2^n degrees of freedom !

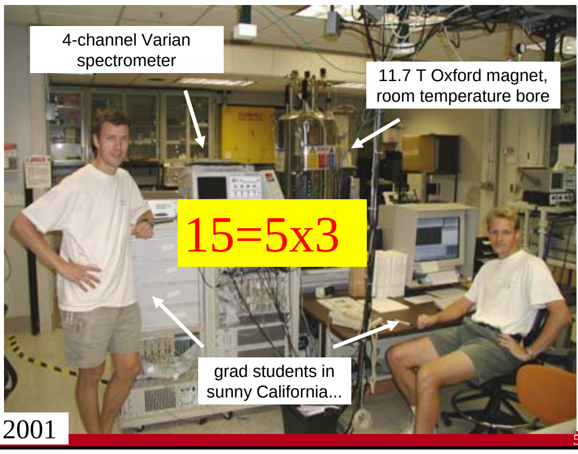
- Shor 1994: perfect for factoring
- But: can a quantum computer be built?



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: unchanged for collisions, x2 for preimages

- alternatives: Post Quantum Crypto: McEliece, HFE, NTRU,…
- So far it seems very hard to match performance of current systems while keeping the security level against conventional attacks



2 approaches to key establishment

RSA with long term keys

choose k —— $RSAPK_B(k \| tA)$ ——→ decrypt with SK_B to get k

Signed Diffie-Hellman (STS)

choose x	α^x →	choose y
$k=(\alpha^y)^x$	← α^y	$k=(\alpha^x)^y$
	$SigA(\alpha^x,\alpha^y)$ →	
√ $SigB$	← $SigB(\alpha^y,\alpha^x)$	√ $SigA$

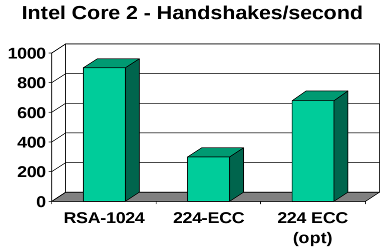
Diffie-Hellman/STS offers one major advantage

- **forward secrecy**: compromise of long term private keys does not expose past session keys
- but more expensive
  - 3 moves rather than 1
  - more public operations
  - incompatible with optimizations such as session caching, session tickets, false start

56

How to solve this

- [Käsper10] optimize OpenSSL
- ECC (NIST P-224 curve) + RSA-1024



57

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

58

Public key protocols: conclusions

- hard to figure out what is recommended in IETF
- more modularity and less complexity would be desirable, but large body of legacy standards and code
- public key operations are still a bottleneck at the server side
- advanced protocols can bring added value from the simple (password-based AKE) to more complex multi-party interactions

59

Outline

- Cryptographic algorithms
  - Block ciphers
  - Hash functions
  - Stream ciphers
  - MAC algorithms
  - Public key algorithms and protocols
- Research challenges

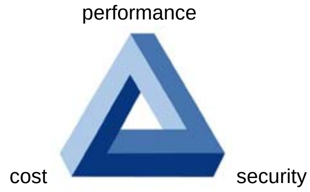
60

Challenges for crypto

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

secure software and hardware implementations

algorithm agility



61

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

62

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

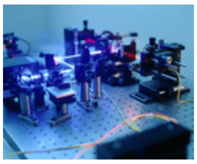
- major impact on implementation cost

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

63

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



64

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)

65

Layers

applications
protocols
primitives
assumptions algorithms

Proofs: link security at different levels in a quantitative way

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

66

Assumptions

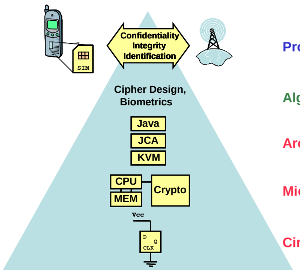
research on hard problems?

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

good lower bounds
average versus worst case
find new hard problems

67

Implementations in embedded systems



Protocol: Wireless authentication protocol design

Algorithm: Embedded fingerprint matching algorithms, crypto algorithms

Architecture: Co-design, HW/SW, SOC

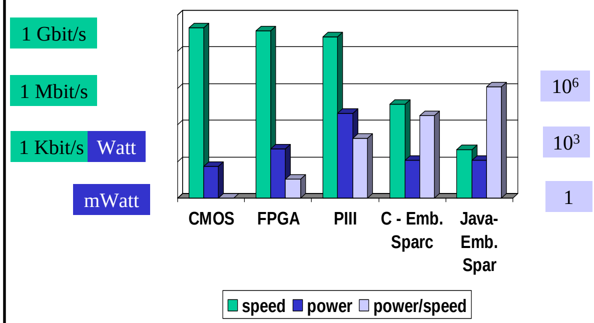
Micro-Architecture: co-processor design

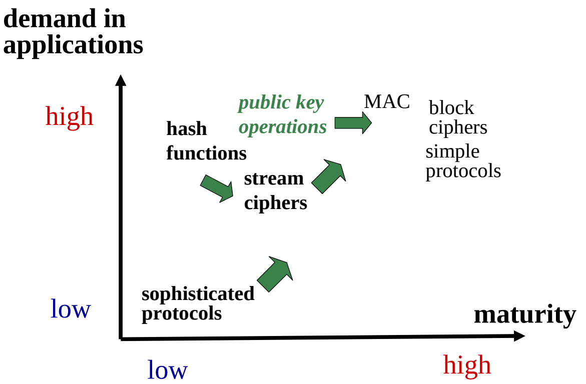
Circuit: Circuit techniques to combat side channel analysis attacks

Technology aware solutions?

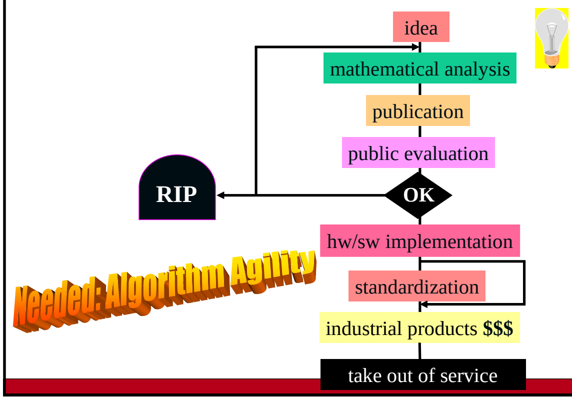
The power challenge:

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





Life cycle of a cryptographic algorithm



Conclusions

- the “crypto 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 everyone pay with e-cash?

can we reconcile privacy, cloud computing, DRM and data mining?