



http://www.ecrypt.eu.org

Research Challenges in Lightweight Cryptography

Bart Preneel
 COSIC, K.U.Leuven, Belgium
 Bart.Preneel(at)esat.kuleuven.be
 http://homes.esat.kuleuven.be/~preneel
 March 2009

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)

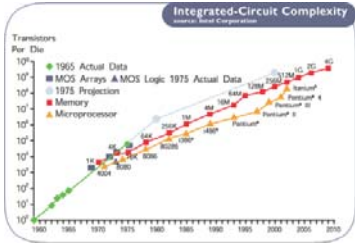
mechanical processing (10^4)

manual processing (10^2)

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	HARDWARE	70
Provable security (PKC), ZK, ElGamal, ECC, stream ciphers	Limited (govt+financial sector) DES, 3DES	80
MD4, MD5	SOFTWARE	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	EVERYWHERE	
ID-Based Crypto	Trusted computing, DRM, 3GPP, RFID, sensor nodes	

A historical perspective

mobile phones	1980	1990	2000
	AMPS	GSM/TDMA	3GSM
	analog cloning, scanners	TDMA cloning	attacks on A5, COMP128
WLAN		1997	2002
		WEP	WPA
		WEP broken	WPA weak
PAN		1999	2004
		Bluetooth	Zigbee
			Bluetooth problems

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?

Slide credit: Prof. Ingrid Verbauwhede 7

Lightweight crypto design

- Overall protocol crucial
- Security architecture: SK-PK, central-distributed
- Relative cost of computation/communication/storage
- Architectural decisions
 - area
 - clock frequency
 - power consumption and energy
- Flexibility can be sacrificed
- Side channel attacks

8

Lightweight crypto design (2)

- Typical settings
 - RFID: constraint on area and power
 - Sensor nodes: constraint on energy
 - Mobile device with large battery: constraint on power
- Research question: “optimal” algorithm/protocol
 - but note many tradeoffs in hardware design

9

Challenges for crypto

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

Performance

Cost Security

secure software and hardware implementations

Algorithm agility

10

Outline

- Context
- Block ciphers
- Stream ciphers
- Hash functions
- MAC algorithms
- Public-key cryptology
- Implementations issues
- Research challenges

11

Block cipher

P1 P2 P3

block cipher block cipher block cipher

C1 C2 C3

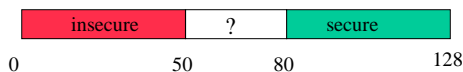
- larger data units: 64...128 bits
- memoryless
- repeat simple operation (round) many times

12

Block ciphers

- 3-DES (112-168)
- IDEA (128)
- KASUMI (128 in 3G, 64 in 2G)
- MISTY1 (128)
- HIGHT (128)
- PRESENT (80-128)
- TEA (128)
- mCRYPTON (128)
- AES (128-192-256)
- RC6
- CAMELLIA
- ...

Symmetric key lengths



13

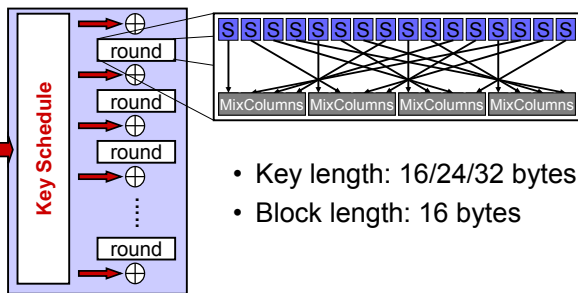
AES (2001)

- open competition: 1997-2000
- FIPS 197 published on December 2001
- mandatory for sensitive US govt. information
- fast adoption in the market
 - ten thousands of products
 - NIST validation list: 1013 implementations
 - <http://csrc.nist.gov/groups/STM/cavp/documents/aes/aesval.html>
 - 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!

Adi Shamir: AES may well be the last block cipher

14

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

15

AES: rich mathematical structure

- very compact/efficient implementations
 - SW: 14 cycles per byte or 2 Gbit/s on high end PCs (8 cycles/byte in bit-slicing)
 - HW: most compact: 3600 gates (PRESENT: 1750)
 - HW: fastest up to 43 Gbit/s in 130 nm 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

16

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



17

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)

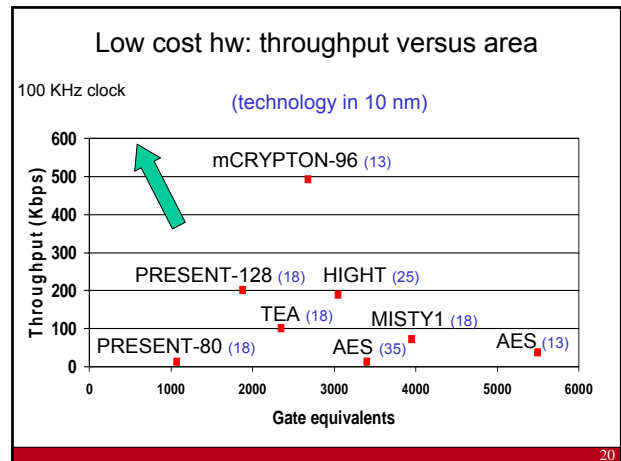
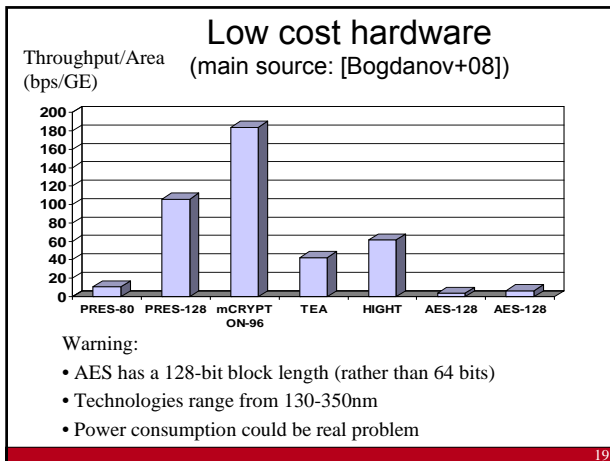
2 days of calculation on 50 PCs (10.000\$) - $2^{44.5}$ encryptions

[Eisenbarth-Kasper-Moradi-Paar-Salmasizadeh-Manzuri Shalmani-Paar 08]

Side channel attack allows to recover master key

in 2010 cryptographers will drive expensive cars

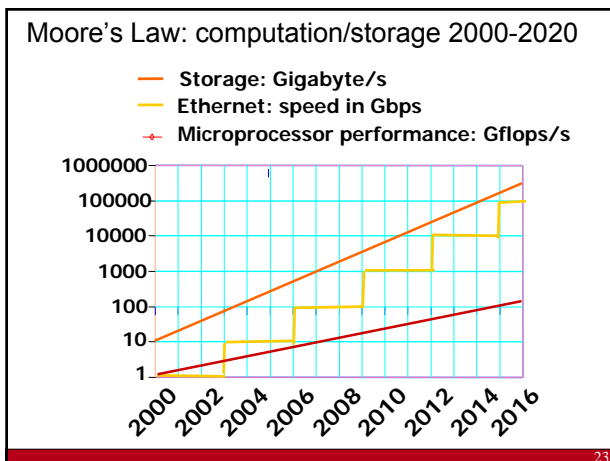
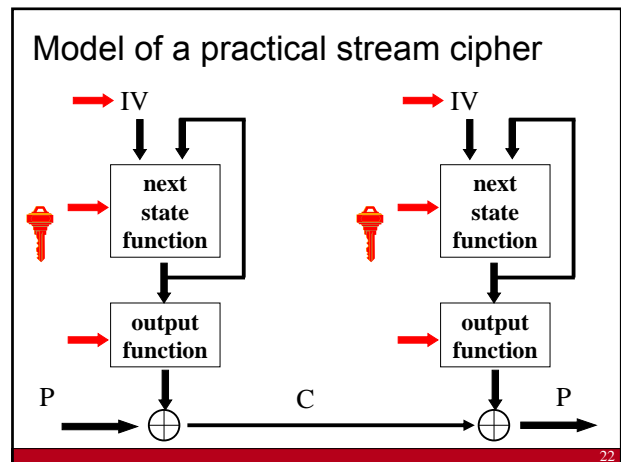
18



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

21



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, SPEED
 - lack of standards and open solutions

24

A5/1 stream cipher (GSM)

- exhaustive key search: 2^{64} (or rather 2^{54})
 - Hardware 10K\$ < 1 minute ciphertext only
- [BB05]: 10 minutes on a PC,
 - 3-4 minutes of **ciphertext only**

25

Bluetooth stream cipher

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

26

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
- April 2005: 34 submissions
- Many broken in first year
- End of competition: April 2008

27

Open competition: Feb. 2007 status

SW Phase 3	HW Phase 3
CryptMT	DECIM
DRAGON	Edon-80
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

28

The eSTREAM Portfolio

Apr. 2008 (Rev1 Sept. 2008)

in alphabetical order

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

3-10 cycles per byte 1500..3000 gates

29

Performance reference data

(Pentium M 1.70GHz Model 6/9/5)

encryption speed (cycles/byte)

key setup (cycles)

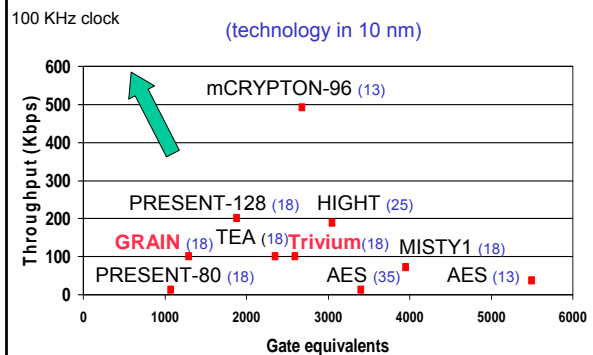
30

Cube attack [Dinur-Shamir'08]

- Exploits low degree equations in stream cipher
- Can break certain ciphers which could not be broken before
- ...Media hype
- Trivium:
 - key setup can be broken if number of rounds is reduced from 1024 to 735
 - attack can probably be further improved
 - solution: increase number of rounds to 2048

31

Low cost hw: throughput versus area



32

Stream ciphers: conclusions

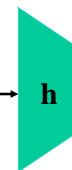
- Substantial progress made in last 5 years
 - Concrete designs
 - Data-time-memory tradeoffs
- 80-bit security implies 160-bit memory or 1000 gates (seems to be a lower bound)
- Many designs still “at the edge” (quite risky)
- For low cost: use a block cipher
 - But 64-bit block cipher has 2^{32} distinguishing attack while for eSTREAM the requirement was no 2^{80} or 2^{128} distinguishing attack

33

Hash functions

- MDC (manipulation detection code)
- Protect short hash value rather than long text
- collision resistance
- preimage resistance
- 2nd preimage resistance

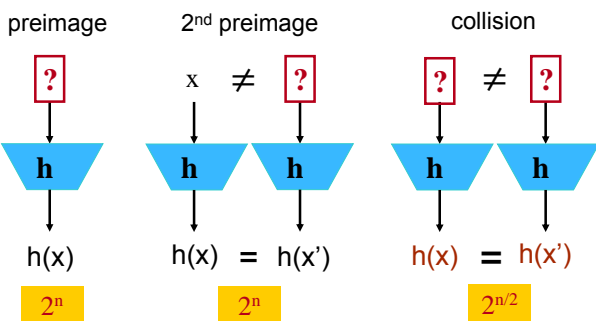
This is an input to a cryptographic hash function. The input is a very long string, that is reduced by the hash function to a string of fixed length. There are additional security conditions: it should be very hard to find an input hashing to a given value (a preimage) or to find two colliding inputs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



1A3FD4128A198FB3CA345932

34

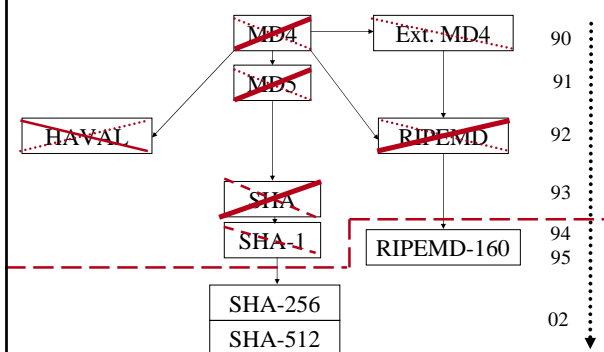
Security requirements (n-bit result)



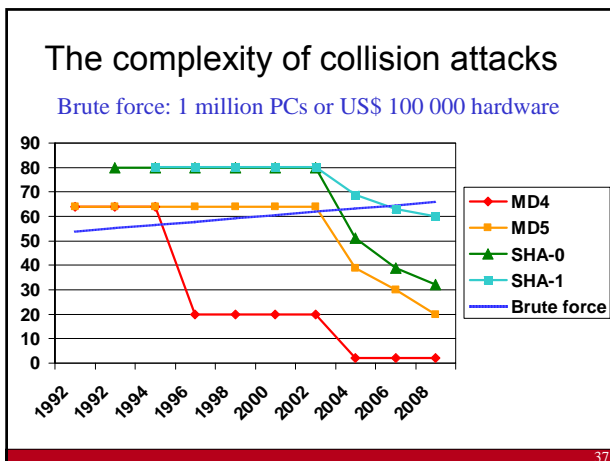
> 80% of all designs for collision resistant hash functions are broken

35

MDx-type hash function history



36



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

SHA-1

- SHA designed by NIST (NSA) in '93
- redesign after 2 years ('95) to SHA-1
- collisions for SHA-1 in 2^{69} [Wang+'05] and 2^{63} [Wang+'05 unpublished]
- collisions for SHA-1 in 2^{60} [Mendel+'08 - unpublished]

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

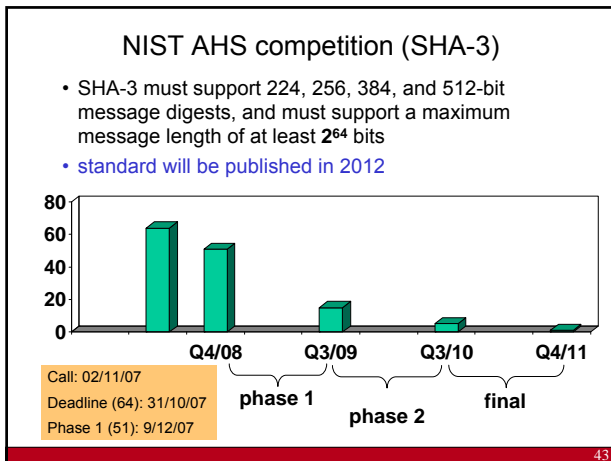
Hash function attacks:

cryptographic meltdown yet with limited impact

- collisions problematic for future
 - digital signatures for non-repudiation (cf. traffic tickets in Australia?)
- 2nd preimage only a problem for MD4
- HMAC-MD4 broken, HMAC-MD5 questionable for the long term
- RIPEMD-160 seems more secure than SHA-1 ☺
- use more recent standards (slower and larger)
 - SHA-256, SHA-512
 - Whirlpool
- upgrading MD5 and SHA-1 in Internet protocols:
 - it doesn't work: **algorithm flexibility is much harder than expected**

Hash function attacks: impact (2)

- [Sotirov-Stevens-Appelbaum-Lenstra-Molnar-Osvik-de Weger '08] **MD5 considered harmful today**
 - fake CA certificate
 - results in a rogue CA: its certificates are trusted by all common browsers
 - need to predict serial number + validity period
- 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



Lightweight (?) hash functions

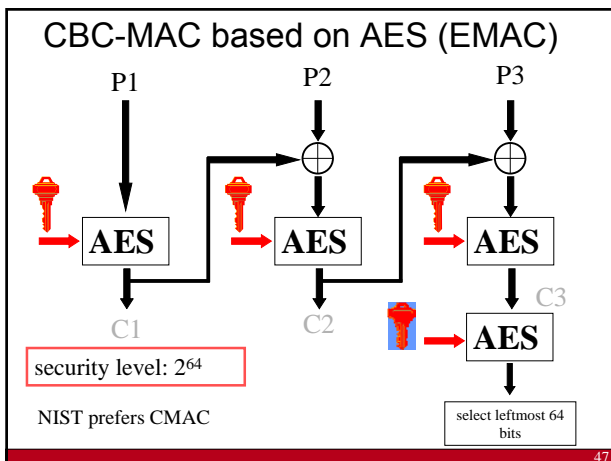
	Area (GE)	Throughput Kbps (@100 KHz)	Throughput/Area (bps/GE)
SHA-256	10900	45	4.1
MAME (256)	8100	267	33.0
PRESENT-based (128)	4256	200	47.0
Lane (256)	17400	37	2.1
Luffa (256)	17500	5216	298

Block cipher-based designs may have an advantage

44

- ### Hash functions: conclusions
- Cryptographic meltdown but fortunately implications so far limited
 - Designers often too optimistic (usually need 2x more rounds)
 - Our understanding has improved substantially
 - Not lightweight
- 45

- ### MAC Algorithms
- CBC-MAC: EMAC and CMAC
 - HMAC
 - GCM and GMAC
 - UMAC
 - Authenticated encryption
- 46



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

48

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

49

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!)
 - Similar attack applies to WMAC polynomial variant

50

MAC for RFID

- SQUASH [Shamir'07,'08]
 - MAC algorithm for authentication in RFID chips
 - claim 500 gates; probably 3000-10000 depending on security level
 - substantially slower than AES
 - security is related to modular squaring (Rabin cryptosystem)

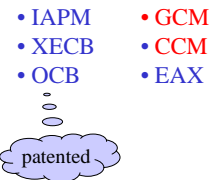
51

Authenticated encryption

- Default modes: ECB/CBC/CFB/OFB and CTR
- 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**



52

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

53

Outline

- Context
- Block ciphers
- Stream ciphers
- Hash functions
- MAC algorithms
- Public-key cryptology
- Implementations issues
- Research challenges

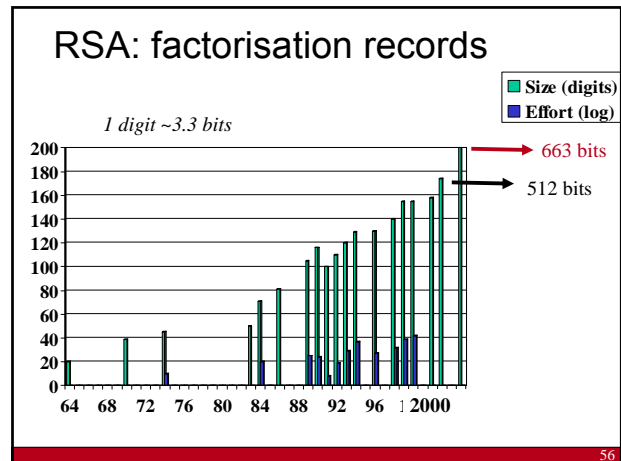
54

RSA

- 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?

55



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 2009 and 896-bit factorization in 2010

57

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

58

(H)ECC MALU for pervasive security

- Implements bit/digit serial modular multiplication and addition in a binary field
- Fixed irreducible polynomial
- ECC over $GF(2^p)$, ECC over composite fields and Hyper Elliptic Curves (HECC)
- Resource sharing of both modular operations
- No separate squaring unit or inverter
 - resistant to simple side-channel attacks

K. Sakiyama, L. Batina, B. Preneel, I. Verbauwhede, "Superscalar Coprocessor for High-speed Curve-based Cryptography," *CHES 2006*, LNCS 4249, Springer-Verlag, pp. 415-429, 2006.

59

Performance: ECC/HECC

PKC – bits of security	# gates w/o RAM	f [kHz]	T [ms]	P [μ W]
ECC - 131	8104	200	265	< 12
ECC - 163	7256	200	400	< 15
ECC-comp - 134	6103	200	210	< 13
HECC - 134	7652	500	546	< 17

60

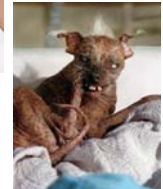
Performance NTRU

[Atici-Batina-Gierlichs-Verbauwhede'08]

- N is degree of polynomials, p, q = coefficient size
- Implementation based on (N, p, q) = (167, 128, 3) claimed to have moderate security, but it turns out that larger parameters are needed
- Encryption-only
 - 2.8 kgates, 1.72 μ W P_{dyn}, 56.44 ms latency @500kHz
- Encryption-decryption
 - 10.5 kgates, 6 μ W P_{dyn}, 56.8 ms/119.2 ms latency @500 kHz for encryption and decryption (but clever choice of f speeds up decryption x2 and reduces area roughly by 2)

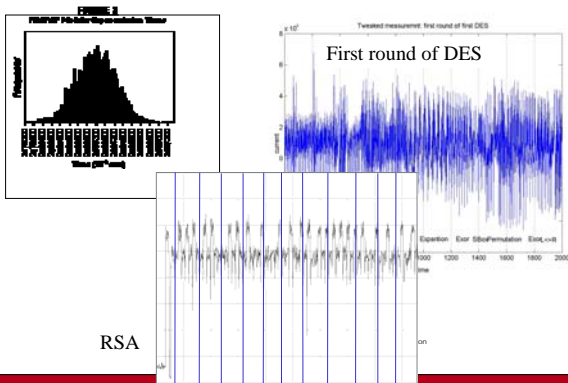
61

Models and reality



62

Implementations: side channel attacks



63

Implementation attacks

Sun Tzu, The Art of War:

In war, avoid what is strong and attack what is weak

- measure: time, power, electromagnetic radiation, sound
- introduce faults (even in CPUs – bug attacks)
- combine with statistical analysis and cryptanalysis
- software: API attacks
- major impact on implementation cost
- move towards security by obscurity

L.R. Knudsen: "It is not cryptanalysis, it is vandalism"

64

Timing attacks on AES software implementations

- Variable execution time typically associated with "if then else", rotations, multiplications
- Due to cache effects, several fast software implementations of AES can be broken
 - e.g., Open SSL: 65 milliseconds
- Fixes:
 - manage cache (2-3x slower)
 - bit slicing
 - hardware implementations
- Cache attacks apply to any cryptographic algorithm that uses tables

65

New side channel attack Bug attack [Biham-Carmeli-Shamir'08]

- Introduce a bug in a multiplier such that it produces the wrong result for a single input pair
 - Example: Pentium FDIV bug '94
- Results in key recovery for RSA-CRT, ECC
- Requires no local access (as a fault attack); only needs chosen texts
- If 64x64: impossible to detect by testing
- Risk of outsourcing the manufacturing

66

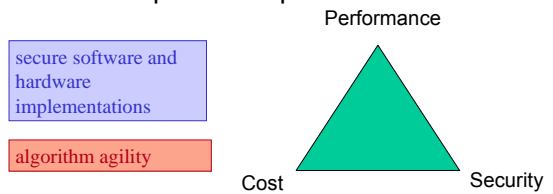
“New” cool crypto stuff

- Most applications use basic crypto technology as known for 30 years (1978)
- What about
 - Credentials
 - Accumulators
 - Pairings
 - Identity based encryption
 - Oblivious transfer
 - Zero-knowledge proof of X

67

Challenges for crypto

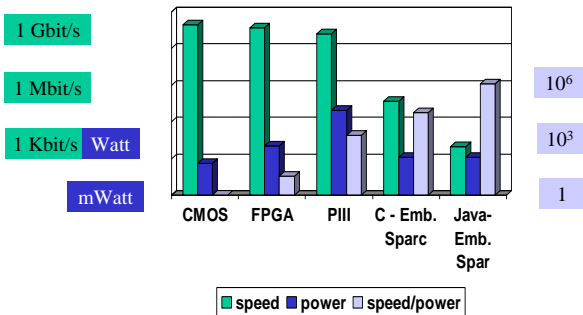
- security for 50-100 years
- authenticated encryption of Terabit/s networks
- ultra-low power/footprint



68

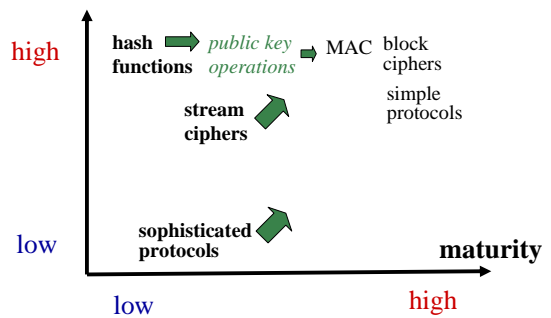
The power challenge:

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



69

demand in applications



70

Conclusions

- Major challenges remain in cryptographic algorithm design
- Lightweight crypto has many dimensions
 - no single optimal solution
 - pushing the edge for all aspects

71

The end



Thank you for your attention

72