ECRYPT

http://www.ecrypt.eu.org

Lightweight Crypto
An Overview

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

April 2010

1

Information processing

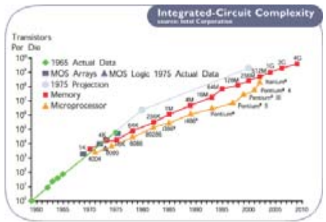
the Internet of things, ubiquitous computing, pervasive computing, ambient intelligence (10^{12})

Internet and mobile (10^{9})

PCs and LANs (10^{7})

mainframe (10^{5})

mechanical (10^{4})



2

Information processing

Everything is always connected everywhere

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

3

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

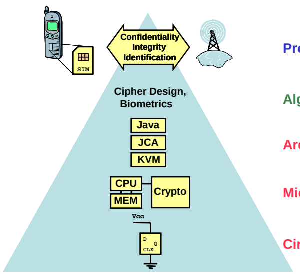
HARDWARE 70
Limited (govt+financial sector)
DES, 3DES 80

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
…

4

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 5

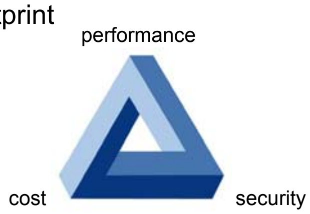
Lightweight crypto design

- 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

6

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

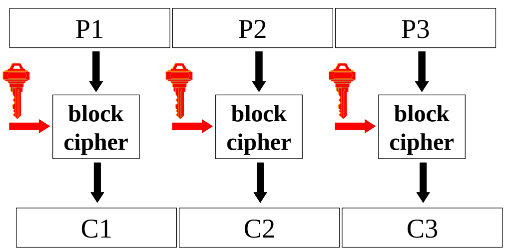
7

Outline

- Context
- Block ciphers
- Stream ciphers
- Hash functions
- MAC algorithms
- Public-key cryptography
- Secure implementations
- RFID protocols

8

Block cipher



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

9

Block ciphers

64-bit block

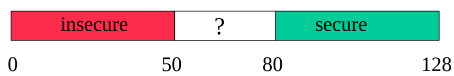
3-DES (112-168)
IDEA (128)
MISTY1 (128)
KASUMI (128 in 3G, 64 in 2G)
HIGHT (128)
PRESENT (80-128)
TEA (128)
mCRYPTON (128)
KATAN (80)

128-bit block

AES (128-192-256)
CAMELLIA
RC6
CLEFIA

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

Symmetric key lengths



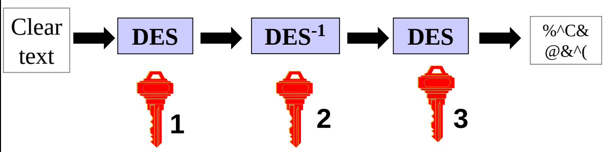
10

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)



11

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: 1267 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

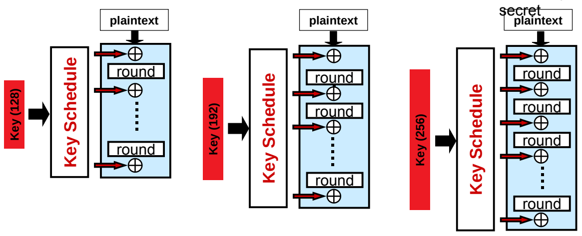
12

AES variants (2001)

- AES-128
- 10 rounds
- sensitive

- AES-192
- 12 rounds
- classified

- AES-256
- 14 rounds
- secret and top secret



Light weight key schedule, in particular for the 256-bit version

AES implementations: efficient/compact

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

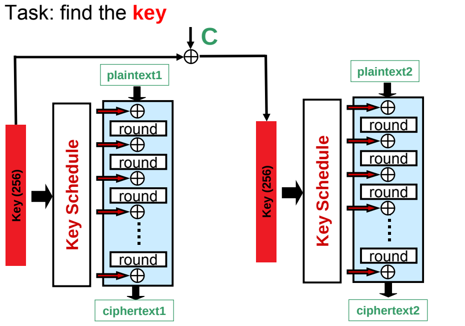
AES-256 security

- Exhaustive key search on AES-256 takes 2^{256} encryptions
  - 2^{64}: 10 minutes with $ 5M
  - 2^{80}: 2 year with $ 5M
  - 2^{120} : 1 billion years with $ 5B
- [Biryukov-Khovratovich'09] **related key attack on AES-256**
  - requires $\mathbf{2^{119}}$ encryptions with 4 related keys,
  - data & time complexity $\mathbf{2^{119} << 2^{256}}$
- Why does it work? Very lightweight key schedule

- Is AES-256 broken?
  No, only an academic "weakness" that is easy to fix
- No implications on security of AES-128 for encryption
- Do not use AES-256 in a hash function construction

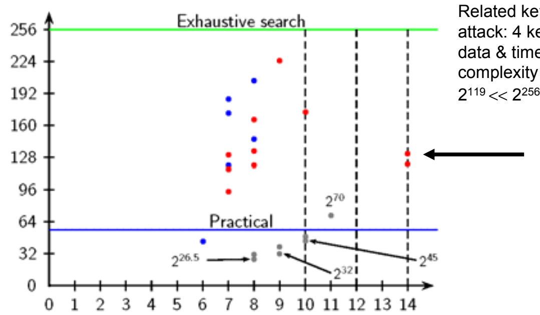
What is a related key attack?

- Attacker chooses **plaintexts** and **key difference** C
- Attacker gets **ciphertexts**
- Task: find the **key**



AES-256

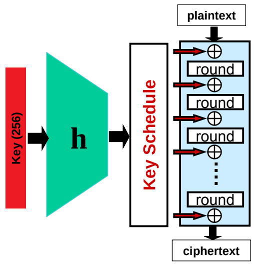
[Biryukov-Khovratovich'09]
[Biryukov-Dunkelman-Keller-Khovratovich-Shamir'09]



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

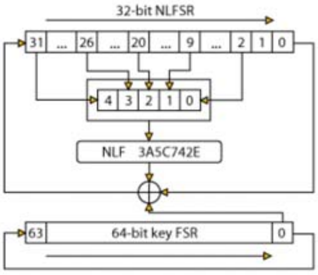
Should I worry about a related key attack?

- Very hard in practice (except 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



Keeloq [Smit+/-'85] aka the M$10 cipher

- block length: 32
- key length: 64
- rounds: 528

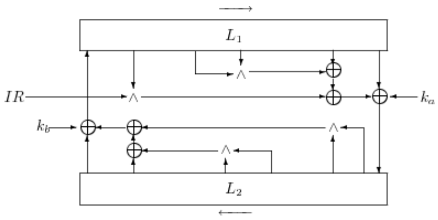


19

KATAN/KTANTAN

[De Cannière-Dunkelman-Knežević'09]
http://www.cs.technion.ac.il/~orrd/KATAN/

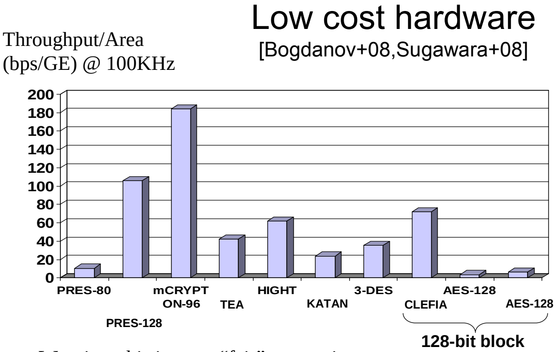
- block length: 32, 48, 64
- key length: 64
- rounds: 254



20

Low cost hardware

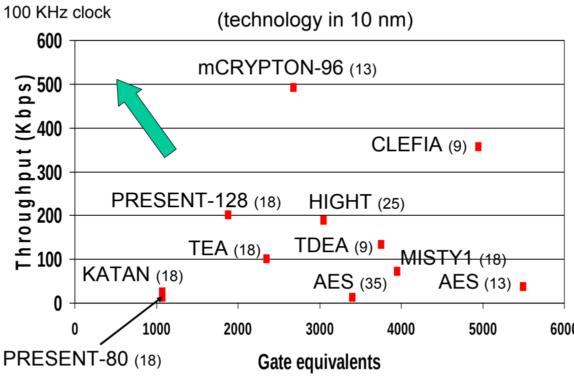
[Bogdanov+08,Sugawara+08]



Warning: this is not a "fair" comparison
- Technologies range from 90nm-350nm
- Power consumption could be real problem

21

Low cost hw: throughput versus area



22

Block ciphers: conclusions

- Several mature block ciphers available
- Security well understood
  – in particular against statistical attacks (differential, linear) and structural attacks
- More work:
  – algebraic attacks
  – related key attacks
  – understanding of structural tradeoffs
- What are the limitations for lightweight block ciphers?

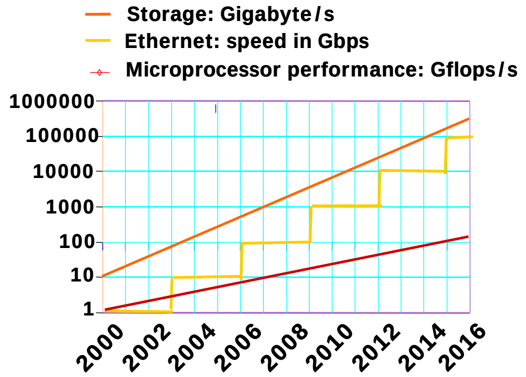
23

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
  – lack of standards and open solutions
  – standards: SNOW2.0, SNOW3G, MUGI, Rabbit, DECIM, K2,..

24

Moore's Law: computation/storage 2000-2020



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

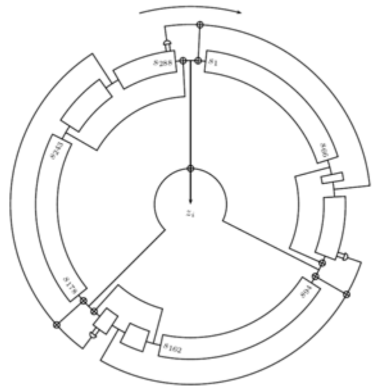
The eSTREAM Portfolio

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

Trivium



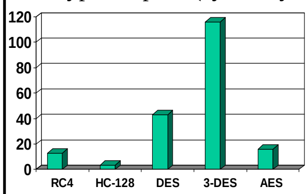
Cube attack [Dinur-Shamir'08]

- Exploits low degree equations in stream cipher
- Can break certain ciphers which could not be broken before
- …Media hype and controversy
  - Relation to higher order attacks (Lai) and AIDA (algebraic IV differential attack) (Vielhaber)
- Trivium:
  - key setup can be broken if number of rounds is reduced from 1024 to 793 (Aida) or 767 (cube)
  - attack can probably be further improved
  - solution: increase number of rounds to 2048

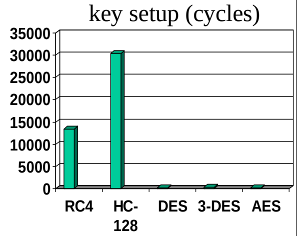
Performance reference data

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

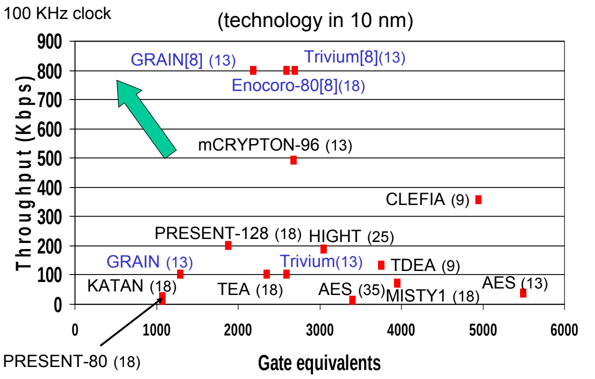
encryption speed (cycles/byte)



key setup (cycles)



Low cost hw: throughput versus area



31

Stream ciphers: conclusions

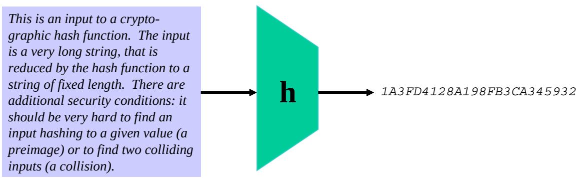
- Substantial progress made in last 5 years
  - concrete designs
  - data-time-memory tradeoffs
- 80-bit security implies 160-bit memory (seems to be a lower bound)
- Many designs still "at the edge" (quite risky)
- Expect further progress

32

Hash functions

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

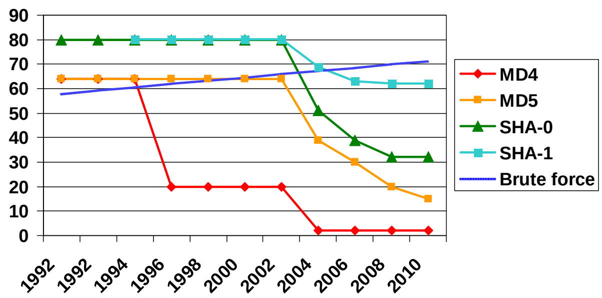
- collision resistance
- preimage resistance
- 2nd preimage resistance



33

The complexity of collision attacks

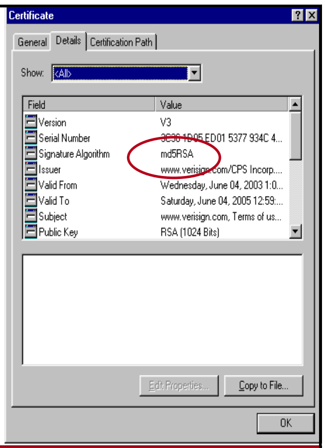
Brute force: 4 million PCs or US$ 100K hardware (1 year)



34

MD5

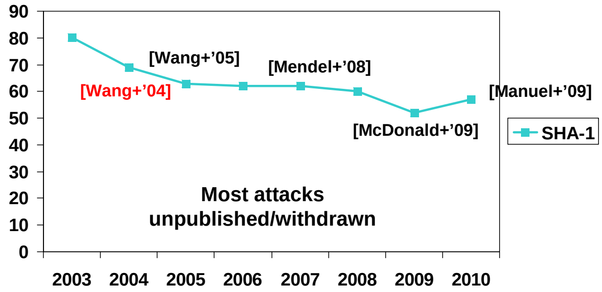
- Advice (RIPE since '92, RSA since '96): stop using MD5
- Largely ignored by industry (click on a cert...)
- Collisions for MD5
  - brute force (2^{64}): 1M$ 10 hours in '09
  - [Wang+'04] collision in 15 minutes on a PC
  - [Stevens+'09] collisions in **milliseconds**
- 2nd preimage:
  - [Sasaki-Aoki'09] 2^{123}



35

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

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?

37

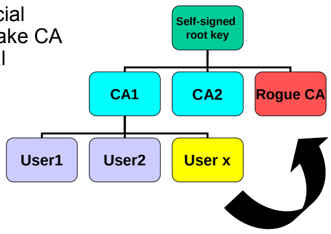
Hash function attacks: impact

- 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: cf. infra

38

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

39

Other ways to fool CAs

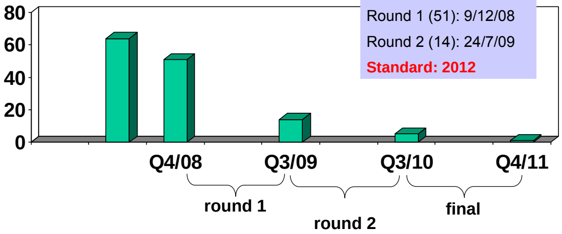
- [Moxie Marlinspike'09] Black Hat
  - browsers may accept bogus SSL certs
  - CAs may sign malicious certs
- certificate for www.paypal.com\0.kuleuven.be will be issued if the request comes from a kuleuven.be admin
- response by PayPal: suspend Moxie's account
  - http://www.theregister.co.uk/2009/10/06/paypal_banishes_ssl_hacker/

40

NIST AHS competition (SHA-3)

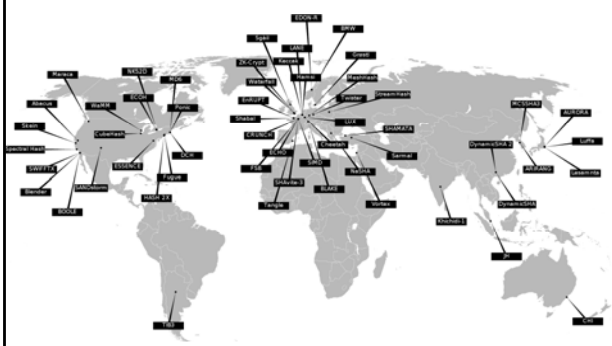
- SHA-3 must support 224, 256, 384, and 512-bit message digests, and must support a maximum message length of at least 2^{64} bits

Call: 02/11/07
Deadline (64): 31/10/08
Round 1 (51): 9/12/08
Round 2 (14): 24/7/09
Standard: 2012



41

The Candidates



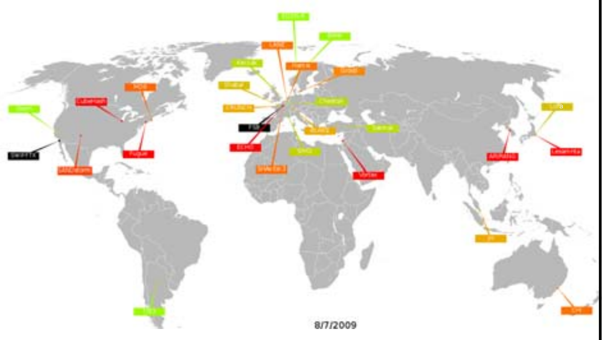
Slide credit: Christophe De Cannière

42

Preliminary Cryptanalysis

Slide credit: Christophe De Cannière

End of Round 1 Candidates



Slide credit: Christophe De Cannière

Round 2 Candidates

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
Cubehash8/1 (512)	7630	2	0.26
PRESENT-based (128)	4256	200	47.0

Block cipher-based designs require strong key schedule – otherwise risky

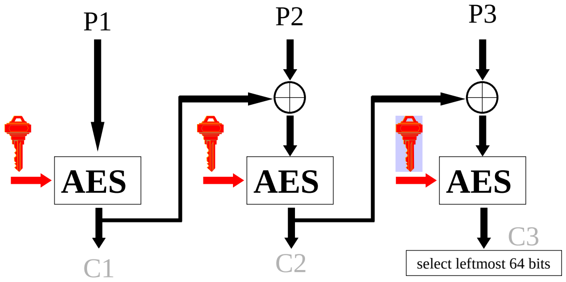
Hash functions: conclusions

- 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
- Today, our understanding has improved substantially, so probably it is likely that it will take > 20 years before we have a SHA-4 competition
- No really lightweight hash functions

MAC Algorithms

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

CBC-MAC based on AES (LMAC)



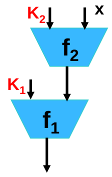
security level against forgery: 2^{64} text/MAC pairs

NIST prefers CMAC: requires only 1 block cipher key

49

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

50

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 $K_1 = AES_K(n)$ (CTR mode)

- properties:
  - lightweight and/or 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

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:
  - 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

- GCM
- CCM
- EAX

patented

52

MAC algorithms: conclusions

- can get better performance than encryption
- EMAC or OMAC (CBC-MAC) seems fine
- widely used choices lack robustness

- Modes for authenticated encryption today well understood but not yet widely deployed

53

Public key algorithms

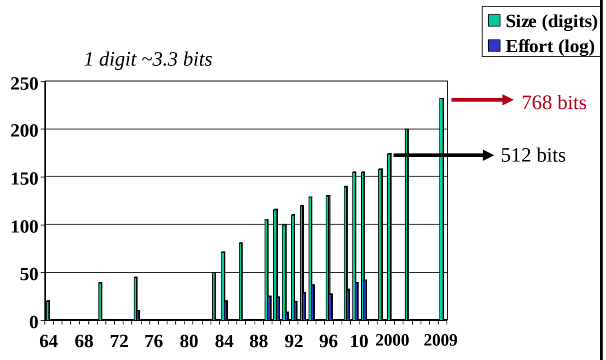
- RSA
- ECC/HECC
- NTRU

Slide credits: Lejla Batina, Junfeng Fan, Ingrid Verbauwhede

55

Factorisation records

2009: 768 bits or 232 digits



56

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

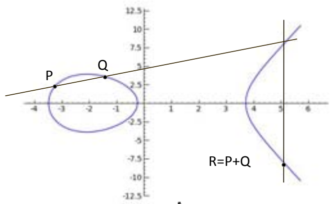
896-bit factorization in 2012, 1024-bit factorization in 2020?

57

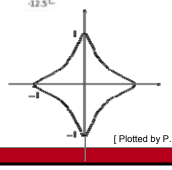
Elliptic curve cryptography

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

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



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



[Plotted by P. Schwabe]

58

Key lengths for confidentiality

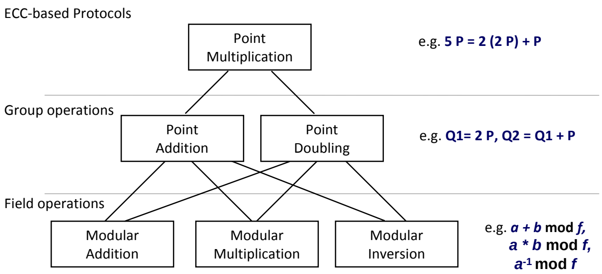
http://www.ecrypt.eu.org

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

59

Point multiplication - ECC

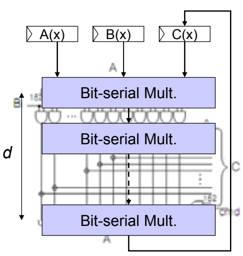


60

Multiplier

Algorithm 1: Modular Multiplication in GF(2^n)

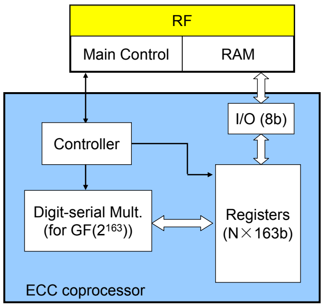
Input: $A(x)$, $B(x)$ and $p(x)$
Output: $A(x)B(x) \bmod p(x)$
1: $C(x) \leftarrow 0$
2: for $i=n-1$ to 0 do
3: $C(x) \leftarrow x(C(x) + c_n p(x) + b_i A(x))$
4: end for
Return $C(x)/x$



Bit-serial Mult.
Digit-serial Mult.

61

ECC processor

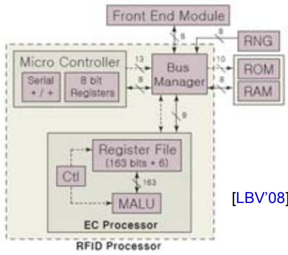


- Area
- Energy
- Security

62

Low footprint

- Curve parameters
  – ECC over binary fields, e.g. $GF(2^{163})$
  – Low weight p(x)
- Coordinates
  – Affine : P(x,y)
  – Projective : P(X,Y,Z)
  – López-Dahab : P(*x*, *z*)
- Only 6 registers!

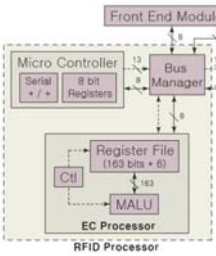


[LBV'08]

63

Low energy

- Energy = Power × Delay
- Reduce power
  – area
  – flip-flop toggling
  – clock frequency
- Reduce delay
  – cycle counts
  – memory accesses



[LBV'08]

64

Hyperelliptic curve Cryptography

- Definition

  Hyperelliptic curve *C* over field ***K*** is defined by

  $y^2 + h(x)y = f(x)$ where $h(x), f(x) \in K[x]$

  – $\deg(h(x)) < g$ and $\deg(f(x)) = 2g + 1$
  – No points also satisfy $2v + h(u) = 0$, $h'(u)v - f'(u) = 0$

- Divisor and Jacobian

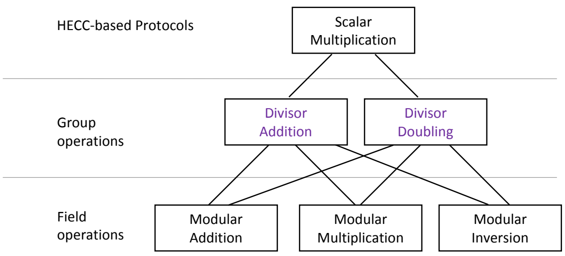
  A divisor *D* is a formal sum of points on *C*.

  $D = \Sigma m_P P$

  – $\deg D = \Sigma m_P$
  – Jacobian is defined as $J = Div_0 / PrinD$

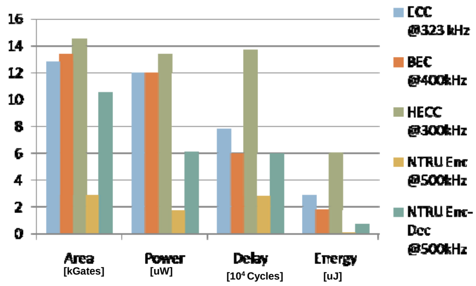
65

Point multiplication - HECC



66

Public key performance comparison



[LBV'08]
[Preprint]
[FBV'08]
[ABFVO'08]

* ECC/BEC over $GF(2^{163})$
* HECC over $GF(2^{83})$
* NTRU parameters: N=167, q=128, p=3

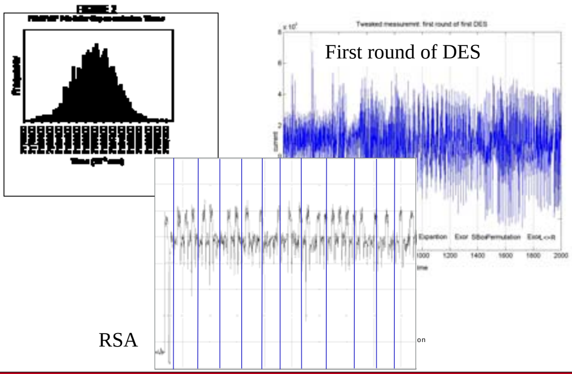
67

Models and reality



68

Implementations: side channel attacks



69

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

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

70

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:
  - implementations that are 2-3x slower
  - special cache for crypto algorithms
- Cache attacks apply to any cryptographic algorithm that uses tables

71

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

72

Side channel attacks on unprotected implementations?

73

RFID technology


1. Passive tag
2. Battery assisted (BAP)
3. Active tag with onboard power source

Slide credits: Lejla Batina, Junfeng Fan, Dave Singelée, Ingrid Verbauwhede

74

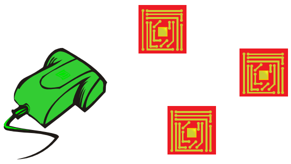
RFID applications

- Asset tracking
- Barcode replacement
- RFID passports
- Mobile credit card payment systems
- Transportation payment systems
- Sporting events (timing / tracing)
- Animal identification
- …

75

RFID security problems (1/2)

- Impersonation attacks
  - Genuine readers
  - Malicious tags



=> **Tag-to-server authentication**

76

RFID security problems (2/2)

- Eavesdropping
- Replay attacks
- Person-in-the-middle attacks
- Cloning
- Side-channel attacks
- …

77

RFID privacy problems (1/5)

Mr. Jones in **2015**

[A. Juels. RSA Laboratories]

78

RFID privacy problems (2/5)

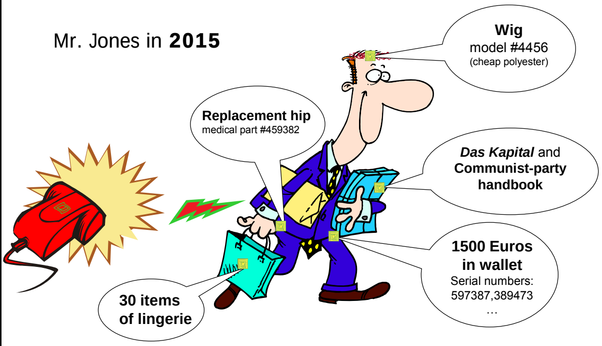
Mr. Jones in **2015**

[A. Juels. RSA Laboratories]

79

RFID privacy problems (3/5)

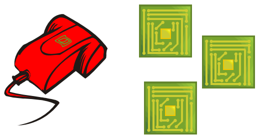
Mr. Jones in **2015**



[A. Juels. RSA Laboratories]

RFID privacy problems (4/5)

- RFID Privacy problem
  - Malicious readers
  - Genuine tags



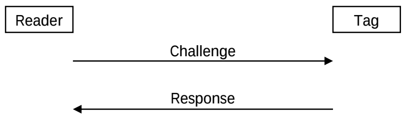
=> **Untraceability**

RFID privacy problems (5/5)

- Untraceability
  - Inequality of two tags: the (in)equality of two tags must be impossible to determine
- Theoretical framework [Vaudenay'07]
  - Narrow versus wide privacy
    - Wide attacker has access to result of verification (accept/reject) at reader side
  - Weak versus strong privacy
    - Strong attacker can extract secret key from tag and reuse it

Cryptographic authentication protocol

- Tag proves its identity using challenge response
  - Security (entity authentication)
  - Privacy



Technological requirements

- Scalability
- Implementation issues
  - Low-cost implementation
    - Memory
    - Gate area
  - Lightweight
  - Efficient

=> **Influence on cryptographic building blocks**

Implementation cost

- Symmetric encryption
  - AES: **3-4 kgates**
- Cryptographic hash function
  - SHA-3: **8 – 30 kgates** [ECRYPT II: SHA-3 Zoo]
- Public-key encryption
  - Elliptic Curve Cryptography (ECC): **11-15 kgates**

Public key cryptography is suitable for RFID

Symmetric protocols

- Fixed Access Control (AC)
  - fixed response from a tag
  - easily tracked
- Randomized AC with a shared key
  - can clone tags by hacking a single tag
- Randomized AC w/o a shared key
  - not scalable
- Randomized AC by updating a key
  - vulnerable against the Denial of Service Attack

86

Asymmetric protocols

- Conventional public-key authentication
  - Schnorr or Okamoto
  - vulnerable to tracking
- GPS
  - variant of Schnorr protool
  - secure transfer of a tag's ID is not solved
- Rabin Encryption
  - large key size and transmission
  - compact architecture [Feldhofer-Oren'09]

- Wide–Weak Privacy–Preserving RFID Authentication Protocols," *Int. Conf. on Mobile Lightweight Wireless Systems* [Lee-Batina-Singelée-Verbauwhede'10]

87

System parameters
[Lee-Batina-Singelée-Verbauwhede'10]

System Parameters	y : Server's private key $Y(=yP)$: Server's public-key x_1 : Tag's ID $X_1(=x_1P)$: Tag's ID-verifier P : Base point in the EC group whose order is a prime n : Prime order of P
Server's storage	y, X_1, x_1, _ P, n
Tag's storage	x_1, _ Y, P, n
Attacker's storage	Y, P, n : Publicly known information

88

Schnorr based on ECC


- Server's input: y
- Tag's input: x_1, $Y(=yP)$

Verifier(Server) — Prover(Tag)

1) Prover: $r_{t1} \in_R \mathbb{Z}$, $T_1 \leftarrow r_{t1}P$; T_1 sent to Verifier
2) Verifier: $r_{s1} \in_R \mathbb{Z}$; r_{s1} sent to Prover
3) Prover: $v \leftarrow r_{t1} + r_{s1}x_1$; v sent to Verifier
4) $(y^{-1}vP - T_1)r_{s1}^{-1} = x_1P$

Problem: tag's public key can be computed from the exchanged messages

89

Secure ID Transfer
(first attempt)

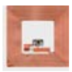

- Server's input: y
- Tag's input: x_1, $Y(=yP)$

Verifier(Server) — Prover(Tag)

1) Prover: $r_{t1} \in_R \mathbb{Z}$, $T_1 \leftarrow r_{t1}P$; T_1 sent to Verifier
2) Verifier: $r_{s1} \in_R \mathbb{Z}$; r_{s1} sent to Prover
3) Prover: $T_2 \leftarrow (r_{t1} + r_{s1}x_1)Y$; T_2 sent to Verifier
4) $(y^{-1}T_2 - T_1)r_{s1}^{-1} = x_1P$

90

Comparison with Schnorr

- Server's input: y
- Tag's input: x_1, $Y(=yP)$

Verifier(Server) — Prover(Tag)

1) Prover: $r_{t1} \in_R \mathbb{Z}$, $T_1 \leftarrow r_{t1}P$; T_1 sent to Verifier
2) Verifier: $r_{s1} \in_R \mathbb{Z}$; r_{s1} sent to Prover
3) Prover: $T_2 \leftarrow (r_{t1} + r_{s1}x_1)Y$, $v \leftarrow r_{t1} + r_{s1}x_1$; v T_2 sent to Verifier
4) $(y^{-1}T_2 - T_1)r_{s1}^{-1} = x_1P$

$(y^{-1}vP - T_1)r_{s1}^{-1} = x_1P$

91

ECC-based authentication protocols

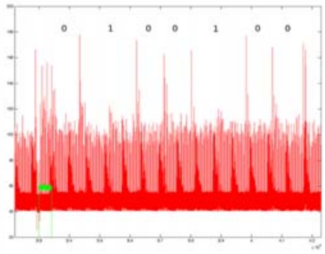
- **rely exclusively on ECC**
- first attempt is vulnerable to person-in-the-middle attack [Deursen-Radomirovic'09] but has been repaired [LBSV'10] to give strong and wide privacy
- Protocols (not shown but paper is online)
  - ID-transfer scheme
  - password transfer scheme
  - scalable search protocol

92

Side-channel attacks and countermeasures

- Unprotected method

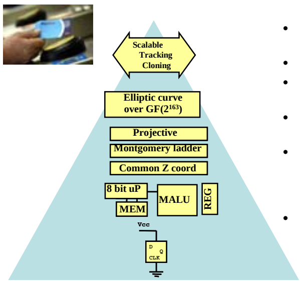
```
for i=n-1 to 0
   Q← 2Q
   if k_i=1
      Q ← Q+P
end for
```



- Countermeasure
  - Unified PA/PD
  - Window method
  - Montgomery ladder

93

Challenge: low power public key



- **Protocol** : asymmetric (most work for the reader)
- **Algorithm**: Elliptic curve (163 bits)
- **Field Operation**: Binary and not Prime fields: easier field operations
- **Projective** coordinate system: (X, Y, Z) instead of (x,y): no field inversions
- **Special coordinate system**: no need to store Y coordinates (Lopez-Dahab) and common Z (only one Z coordinate)
- **Minimize storage**: Only 5 registers (with mult/add/square unit) or 6 registers (with mult/add-only unit)

Address at all abstraction levels!

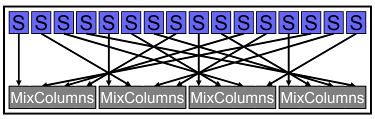
94

Performance results

Circuit Area (Gate Eq.)	14,566
Cycles for EC point multiplication	59,790
Frequency	700 KHz
Power	13.8 μW
Energy for EC point multiplication	1.18 μJ

95

Lightweight symmetric cryptography: lessons learned (1)



Non-linear layer

Linear layer

- How can we save?
  - Non-linear layer can be reduced from 1280 gates (AES) to 32 gates (KATAN) or even 3 gates (Trivium)
  - Linear layer can be reduced from 396 gates (AES) to 0 gates, e.g. bit permutation (KATAN/PRESENT)
- In both cases, this requires more rounds for block ciphers (and thus more energy)

96

Lightweight symmetric cryptography: lessons learned (2)

- If non-linear and linear layers are heavily optimized, the cost is dominated by memory for key (k bits, k=80-128) and by memory for state (n bits)
  - Block cipher: n bits – can encrypt at most $2^{n/2-10}$ plaintexts
  - Stream cipher: n ≥ k bits needed (in practice often 2k)
  - Hash function: n ≥ 2k bits needed for 2^k collision resistance (but no key!)
  - MAC: can be based on block cipher
- Hardware: how many gates does it cost to store 1 bit?
  - technology dependent: between 2 and 8
- Software: RAM usage is critical factor
  - 256 bytes on low-end 8-bit processor (such as PIC10-16, RS08TM, HC08TM, COP8, 80C51TM)

97

Lightweight public-key cryptography: lessons learned (1)

- Compact public key crypto (ECC, HECC, NTRU) is feasible but requires
  - hardcoded designs
  - context-dependent optimization for area, power, energy, speed on multiple abstraction layers
- Concerns: side channel attacks and long term security
- Case: RFID

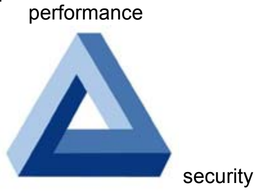
98

Challenges for crypto

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

secure software and hardware implementations

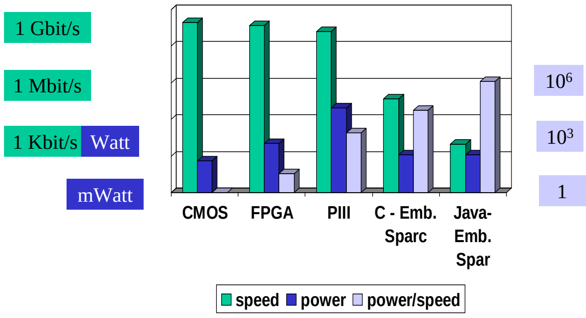
algorithm agility



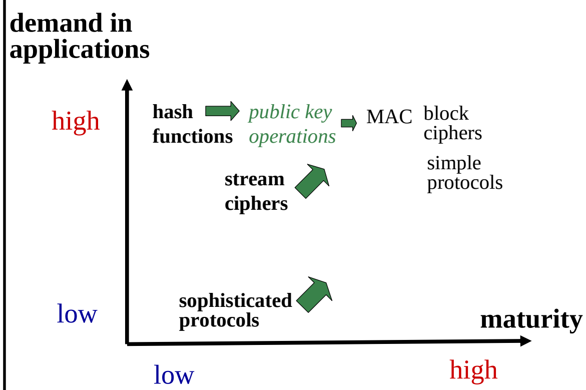
99

The power challenge:

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

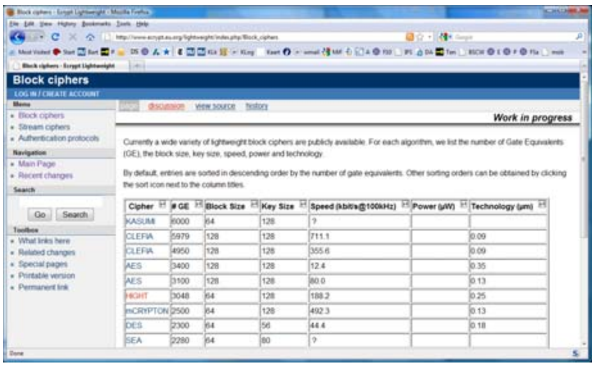


100



101

http://www.ecrypt.eu.org/lightweight/


Block ciphers

Work in progress

Currently a wide variety of lightweight block ciphers are publicly available. For each algorithm, we list the number of Gate Equivalents (GE), the block size, key size, speed, power and technology.

By default, entries are sorted in descending order by the number of gate equivalents. Other sorting orders can be obtained by clicking the sort icon next to the column titles.

Cipher	# GE	Block Size	Key Size	Speed (kbit/s@100kHz)	Power (µW)	Technology (µm)
KASUMI	6000	64	128	?		
CLEFIA	5979	128	128	711.1		0.09
CLEFIA	4950	128	128	355.6		0.09
AES	3400	128	128	12.4		0.35
AES	3100	128	128	80.0		0.13
HIGHT	3048	64	128	188.2		0.25
mCRYPTON	2500	64	128	492.3		0.13
DES	2300	64	56	44.4		0.18
SEA	2280	64	80	?		

102

Conclusions

- Major challenges remain in cryptographic algorithm design
- Lightweight crypto has many dimensions
  - no single optimal solution for RFID, sensor nodes, co-processor for 8-bit CPU,…
  - pushing the edge for all aspects
- Symmetric crypto with less than 1000 gates is feasible
- Public key crypto with less than 15,000 gates is feasible
- Side channel resistance remains a challenge

103