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

Moore’s Law: computation/storage 2000-2020

Information processing

Exponential growth
Ray Kurzweil, KurzweilAI.net
- Human brain: \(10^{14} \ldots 10^{15}\) 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$

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

Information processing

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

Everything is always connected everywhere

the Internet of things, ubiquitous computing, pervasive computing, ambient intelligence \((10^{12})\)
PCs and LANs \((10^9)\)
mainframe \((10^8)\)
mechanical processing \((10^9)\)
manual processing \((10^2)\)
[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.

Context (2)

wireless data


Vernam

OTP

rotor machines

LFSR

WLAN

PAN

3GSM

Context (3)

mobile phones

1980 1990 2000

AMPS

TDMA

GSM/TDMA

3GSM

LTE

WLAN

1997 2002 2004

WEP

WPA

WPA2

802.11i

attacks on A5, COMP128

PAN

1999 2004

Bluetooth

Bluetooth problems

Zigbee

Challenges for crypto

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

Performance of hash functions - Bernstein (cycles/byte) Intel Pentium D 2992 MHz (64)
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)
- mCRYPTON (128)
- KATAN (80)

128-bit block
- AES (128-192-256)
- CAMELLIA
- RC6
- CLEFIA

Symmetric key lengths

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


- 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 2001 after 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
- 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 \([\text{Käsper-Schwabe}'09]\)
- 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

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 \(2^{119} \ll 2^{256}\)

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

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 \(\oplus C\)) 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
Cryptographic Algorithms for Network Security - Failures, Success and Challenges

Bart Preneel

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).

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, 1MB 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

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

Most attacks unpublished/withdrawn

Hash function attacks: cryptographic meltdown yet with limited impact

- Problems: collision, preimage, etc.
- Use new standards (slower and larger)
- SHA-2 (SHA-256, SHA-224,...SHA-512)
- SHA-3?

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
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 2^44 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

MAC Algorithms

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

CBC-MAC based on AES (EMAC)

P1
P2
P3

AES
AES
AES

C1
C2
C3

Security level: $2^{64}$

NIST prefers CMAC

HMAC based on MDx, SHA

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

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

- Keys $K_i, K_j \in GF(2^{128})$
- Input $x$: $x_1, x_2, \ldots, x_n$, with $x_i \in GF(2^{128})$
- $g(x) = K_1 + \sum_{i=1}^{n} x_i \cdot (K_2)^i$
- 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

UMAC RFC 4418 (2006)

- Key $K, K_1, K_2, \ldots, K_{256} \in GF(2^{12})$ (1024 bytes)
- Input $x$: $x_1, x_2, \ldots, x_{256}$ with $x_i \in GF(2^{12})$
- $g(x) = prf_K(h(x))$
- $h(x) = \left( \sum_{i=1}^{256} (x_{2i-1} + k_{2i-1}) \mod 2^{12} \cdot (x_{2i} + k_{2i}) \mod 2^{12} \right) \mod 2^{64}$

- Properties:
  - Software performance: 1-2 cycles/byte
  - Forgery probability: $1/2^{102}$ (provable lower bound)
  - [Handschuh-Preneel08] full key recovery with $2^{40}$ verification queries (no nonce reuse needed!)
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

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

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

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

Is factoring hard?
Is the RSA problem, i.e., inverting \( f(x) = x^e \mod n \) as hard as factoring?
Can we show that forging a signature implies factoring (and this without the Random Oracle assumption)?
Cryptographic Algorithms for Network Security - Failures, Success and Challenges
Bart Preneel

Elliptic curve cryptography

Elliptic curve: \( E : y^2 = x^3 - 13x - 3 \)

Point multiplication:
\[ rP = P + P + \ldots + P \]

Edwards curve: \( E : x^2 + y^2 = 1 - 30x^2y^2 \)

Key lengths for confidentiality
http://www.ecrypt.eu.org

<table>
<thead>
<tr>
<th>duration</th>
<th>symmetric</th>
<th>RSA</th>
<th>ECC</th>
</tr>
</thead>
<tbody>
<tr>
<td>days/hours</td>
<td>50</td>
<td>512</td>
<td>100</td>
</tr>
<tr>
<td>5 years</td>
<td>73</td>
<td>1024</td>
<td>146</td>
</tr>
<tr>
<td>10-20 years</td>
<td>103</td>
<td>2048</td>
<td>206</td>
</tr>
<tr>
<td>30-50 years</td>
<td>141</td>
<td>4096</td>
<td>282</td>
</tr>
</tbody>
</table>

Assumptions: no quantum computers; no breakthroughs; limited budget

New computational models: quantum computers?

- exponential parallelism
- 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: \( x^2 \)
- hash sizes: unchanged for collisions, \( x^2 \) 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
\[ \text{choose } k \quad \text{RSA}_{PK}(k || t_A) \quad \text{decrypt with } SK_B \text{ to get } k \]

Signed Diffie-Hellman (STS)
\[ \text{choose } x \quad \alpha^x \quad \text{choose } y \quad \alpha^y \quad k = (\alpha^x)^y \quad \sqrt{\text{Sig}_B} \]
\[ \sqrt{\text{Sig}_A} \]

Signed Diffie-Hellman (STS)
\[ \text{choose } x \quad \alpha^x \quad \text{choose } y \quad \alpha^y \quad k = (\alpha^x)^y \quad \sqrt{\text{Sig}_B} \]
\[ \sqrt{\text{Sig}_A} \]

4-channel Varian spectrometer

11.7 T Oxford magnet, room temperature bore

15 = 5 \times 3

2001

grad students in sunny California...
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

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

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

Outline

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

Challenges for crypto

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

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

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

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)

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"

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
Cryptographic Algorithms for Network Security - Failures, Success and Challenges
Bart Preneel

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

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

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?