Cryptographic Algorithms: Successes, Failures and Challenges

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

Information processing

- the Internet of things, ubiquitous computing, pervasive computing, ambient intelligence (10^{12})
- Internet and mobile (10^6)
- PCs and LANs (10^7)
- mainframe (10^5)
- mechanical processing (10^2)

Exponential growth

Ray Kurzweil, KurzweilAI.net
- Human brain: 10^{14} ... 10^{16} 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

HARDWARE
- DES, RSA, DH, CBC-MAC
- Provable security (PKC), ZK, ElGamal, ECC, stream ciphers
- MD4, MD5
- Provable security (SKC)
- Key escrow
- How to use RSA?
- Alternatives to RSA

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

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

TLS 1.0 Data Encapsulation Options (01/99)

<table>
<thead>
<tr>
<th>Integrity</th>
<th>144</th>
<th>160</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMAC-MD5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HMAC-SHA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Confidentiality

<table>
<thead>
<tr>
<th>key size</th>
<th>40</th>
<th>56</th>
<th>128</th>
<th>168</th>
</tr>
</thead>
<tbody>
<tr>
<td>algorithm options</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3DES_CBC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RC4_CBC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DES_CBC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Data Encapsulation Options (01/99)
- Cryptographic Algorithms
**Cryptographic Algorithms**

**Bart Preneel**

---

**TLS 1.0 Key Management Options (01/99)**

- **Anonymous**
  - Server authentication, no client authentication
  - DH_anon
- **Non-anonymous**
  - Server and client authentication
  - RSA
  - DH_DSS
  - DH_RSA
  - DHE_DSS
  - DHE_RSA

---

**IKE Algorithm Selection**

**Mandatory Algorithms**

<table>
<thead>
<tr>
<th>Algorithm Type</th>
<th>IKE v1</th>
<th>IKE v2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Payload Encryption</td>
<td>DES_CBC</td>
<td>3DES_CBC</td>
</tr>
<tr>
<td></td>
<td>(AES_128_CBC)</td>
<td></td>
</tr>
<tr>
<td>Payload Integrity</td>
<td>HMAC-MD5</td>
<td>HMAC-SHA1</td>
</tr>
<tr>
<td>DH Group</td>
<td>768 Bit</td>
<td>1024 (2048) Bit</td>
</tr>
<tr>
<td>Transfer Type 1</td>
<td>ENCR_DES_CBC</td>
<td>ENCR_3DES</td>
</tr>
<tr>
<td>(Encryption)</td>
<td>(ENCR_AES_128_CBC)</td>
<td></td>
</tr>
<tr>
<td>Transfer Type 2</td>
<td>PRF_HMAC_SHA1</td>
<td>PRF_HMAC_SHA1</td>
</tr>
<tr>
<td>(PRF)</td>
<td>[RFC2104]</td>
<td>[RFC2104]</td>
</tr>
<tr>
<td>Transfer Type 3</td>
<td>AUTH_HMAC_SHA1</td>
<td>AUTH_HMAC_SHA1</td>
</tr>
<tr>
<td>(Integrity)</td>
<td>[RFC2404]</td>
<td>[RFC2404]</td>
</tr>
</tbody>
</table>

---

**RFC 3268: AES Ciphersuites for TLS (06/02)**

<table>
<thead>
<tr>
<th>CipherSuite</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLS_RSA_WITH_AES_128_CBC_SHA</td>
</tr>
<tr>
<td>TLS_DH_DSS_WITH_AES_128_CBC_SHA</td>
</tr>
<tr>
<td>TLS_DH_RSA_WITH_AES_128_CBC_SHA</td>
</tr>
<tr>
<td>TLS_DHE_DSS_WITH_AES_128_CBC_SHA</td>
</tr>
<tr>
<td>TLS_DHE_RSA_WITH_AES_128_CBC_SHA</td>
</tr>
<tr>
<td>TLS_DH_anon_WITH_AES_128_CBC_SHA</td>
</tr>
<tr>
<td>TLS_RSA_WITH_AES_256_CBC_SHA</td>
</tr>
<tr>
<td>TLS_DH_DSS_WITH_AES_256_CBC_SHA</td>
</tr>
<tr>
<td>TLS_DH_RSA_WITH_AES_256_CBC_SHA</td>
</tr>
<tr>
<td>TLS_DHE_DSS_WITH_AES_256_CBC_SHA</td>
</tr>
<tr>
<td>TLS_DHE_RSA_WITH_AES_256_CBC_SHA</td>
</tr>
<tr>
<td>TLS_DH_anon_WITH_AES_256_CBC_SHA</td>
</tr>
</tbody>
</table>

---

**Implications 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

---

**Disclaimer:**

**Cryptography ≠ security**

- crypto is only a tiny piece of the security puzzle
  - but an important one
- 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

---

**[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.
Challenges for crypto

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

Outline

- Block ciphers
- Hash functions
- Stream ciphers
- Public-key cryptology
- Protocols
- Implementations issues
- Research challenges

Outline

- Block ciphers
  - 3-DES (112-168)
  - AES (128-192-256)
  - KASUMI (128 in 3G, 64 in 2G)
  - IDEA (128)

Federal Register, July 24, 2004

- SUMMARY: The Data Encryption Standard (DES), currently specified in Federal Information Processing Standard (FIPS) 46-3, was evaluated pursuant to its scheduled review. At the conclusion of this review, NIST determined that the strength of the DES algorithm is no longer sufficient to adequately protect Federal government information. As a result, NIST proposes to withdraw FIPS 46-3, and the associated FIPS 74 and FIPS 81. Future use of DES by Federal agencies is to be permitted only as a component function of the Triple Data Encryption Algorithm (TDEA).

DES (1977)

- 56-bit key length is too short
- 25/10/99: DES reaffirmed for the 4th time as FIPS 46-3
- 2007: $1 million search machine: 20 seconds
  - cost per key: less than $0.50
- 2007: 500 PCs at night: 1 month
  - Cost per key: essentially 0 (+ some patience)

Block cipher

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

Symmetric key lengths

- insecure
- secure

Federal Register, July 24, 2004

- SUMMARY: The Data Encryption Standard (DES), currently specified in Federal Information Processing Standard (FIPS) 46-3, was evaluated pursuant to its scheduled review. At the conclusion of this review, NIST determined that the strength of the DES algorithm is no longer sufficient to adequately protect Federal government information. As a result, NIST proposes to withdraw FIPS 46-3, and the associated FIPS 74 and FIPS 81. Future use of DES by Federal agencies is to be permitted only as a component function of the Triple Data Encryption Algorithm (TDEA).

Block cipher

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

3-DES: NIST Spec. Pub. 800-67
(May 2004)

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

AES (2001)

- open competition: 1997-2000
- FIPS 197 published on December 2001
- mandatory for sensitive US govt. information

- fast adoption in the market
  - > 1000 products
  - July 2008: 835 AES product certifications by NIST
  - standardization: ISO, IETF, IEEE 802.11,…
- slower adoption in financial sector
- mid 2003: AES-128 also for classified information and AES-192/-256 for secret and top secret information!

AES may well be the last block cipher

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

AES: rich mathematical structure

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

Modes of Operation for AES

- encryption: ECB/CBC/CFB/OFB;
  - CTR mode allows for pipelining (‘01)
- data authentication: CMAC (‘05), EMAC
- applications need authenticated encryption:
  - GCM Galois Counter Mode (final draft: July 07)

Issues:
- associated data
- parallelizable
- on-line
- provable security
  - IAPM
  - XECB
  - OCB
  - GCM
  - EAX
  - CCM

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 526 simple rounds
**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-Indesteegeh-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-ManzuriShalmaniPaar 08]:
  - Side channel attack allows to recover master key

> in 2010 cryptographers will drive expensive cars

**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

---

**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).

---

**Security requirements (n-bit result)**

<table>
<thead>
<tr>
<th>preimage</th>
<th>2nd preimage</th>
<th>collision</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="h(x) ≠ h(x')" /></td>
<td><img src="image" alt="h(x) = h(x')" /></td>
<td>![h(x) ≠ h(x')] (2^n)</td>
</tr>
</tbody>
</table>

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

---

**MDx-type hash function history**

- MD4
- Ext.-MD4
- MD5
- SHA-0
- SHA-1
- SHA-256
- SHA-512
- RIPEMD
- RIPEMD-160

---

**Collision attacks**

<table>
<thead>
<tr>
<th>Brute force: 1 million PCs or 100.000$ hardware</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Graph" /></td>
</tr>
</tbody>
</table>

> MD4, MD5, SHA-0, SHA-1, Brute force
### 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 found for SHA-0 in $2^{51}$ [Joux+’04]
- Reduced to $2^{39}$ [Wang+’05] and $2^{32}$ [Rechberger+’07]
- Collisions for SHA-1 in $2^{63}$ [Wang+’05]
- Collisions for SHA-1 found for 70 out of 80 rounds [De Cannière-Mendel-Rechberger’07] in $2^{44}$
- Prediction: collision for SHA-1 in 2009; complexity estimate is $2^{60}$ [Rechberger+’07]

---

### Hash function attacks: impact
- Cryptographic meltdown yet with limited impact
- Collisions problematic for future
  - Digital signatures for non-reputation (cf. Traffic tickets in Australia?)
- $2^{64}$ 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)
  - SHA-256, SHA-512
  - Whirlpool
- Upgrading MD5 and SHA-1 in Internet protocols:
  - It doesn’t work: Algorithm flexibility is much harder than expected
- NIST will run an open competition from 2008 to 2012

### Model of a practical stream cipher

---

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

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

In July 2008 this information was still available on MSDN
**Stream ciphers**

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

---

**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

---

**Lightweight crypto**

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

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

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

RSA: factorisation records

Key lengths for confidentiality

http://www.ecrypt.eu.org

Elementary

Factorisation

- New record in May 2005: 663 bits (or 200 digits) using NFS
- New record in May 2007: 2^{1039}-1 (313 digits) using SNFS

- hardware factoring machine: TWIRL [TS’03]
  (The Weizmann Institute Relation Locator)
  - initial R&D cost of ~$20M
  - 512-bit RSA keys can be factored with a device costing $5K in about 10 minutes
  - 1024-bit RSA keys can be factored with a device costing $10M in about 6 weeks

- ECRYPT statement on key lengths and parameters
  http://www.ecrypt.eu.org

  768-bit factorization in 2008 and 896-bit factorization in 2010

Key lengths for confidentiality

http://www.ecrypt.eu.org

- symmetric
- RSA
- ECC

<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

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

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

60 01 ff ... ff 00 HashID H Magic

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

60 01 ff ... ff 00 HashID H
Protocols (1)
- key transport (email)
- authenticated key agreement (TLS, SSH, GSM, UMTS)
- time-stamping
- notarisation
- credentials (TPM)
- anonymous communication
- e-cash
- voting
- auctions
- threshold cryptography
- robust networking

Protocols (2)
- multi-party computation
- threshold crypto
- privacy protecting data mining
- social and group crypto
decryption based on location and context
distance bounding

Models and reality

Implementation: side channel attacks
- First round of DES
- RSA

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

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

L.R. Knudsen: “It is not cryptanalysis, it is vandalism”
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: $x^2$
- Hash sizes: $x^{1.5}$
- Alternatives: McEliece, HFE, NTRU,…
- So far it seems very hard to match performance of current systems while keeping the security level against conventional attacks

News on 13 Sept. 2007

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

News on 19 Dec. 2007

- Optical quantum computer (team led by Daniel James, University of Toronto)
- Factored 15

Layers

- Applications
- Protocols
- Primitives
- 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
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
- Performance
- Cost
- Security

The power challenge:

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

The demand in applications:

- high
  - hash functions
  - public key operations
  - block ciphers
  - stream ciphers
  - simple protocols
- low
  - sophisticated protocols

Life cycle of a cryptographic algorithm:

- idea
- mathematical analysis
- publication
- public evaluation
- RIP
- hw/sw implementation
- standardization
- industrial products
- take out of service

Challenges for advanced crypto:

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

Conclusions:

- The "security problem" is not solved
- Many challenging problems ahead…
- Make sure that you can upgrade your crypto algorithm and protocol
- Bring advanced cryptographic protocols to implementations

When will the IACR hold its elections on-line?
When will everyone pay with e-cash?
Can we reconcile privacy, DRM and data mining?