Proposal of addition of new cipher suites to TLS to support Camellia, EPOC, and PSEC

Shiho Moriai

shiho@isl.ntt.co.jp

NTT Laboratories

128-bit Block Cipher *Camellia*

Kazumaro Aoki[*] Tetsuya Ichikawa[†]
Masayuki Kanda[*] Mitsuru Matsui[†]
Shiho Moriai[*] Junko Nakajima[†]
Toshio Tokita[†]

* NTT
† Mitsubishi Electric Corporation

What's Camellia?

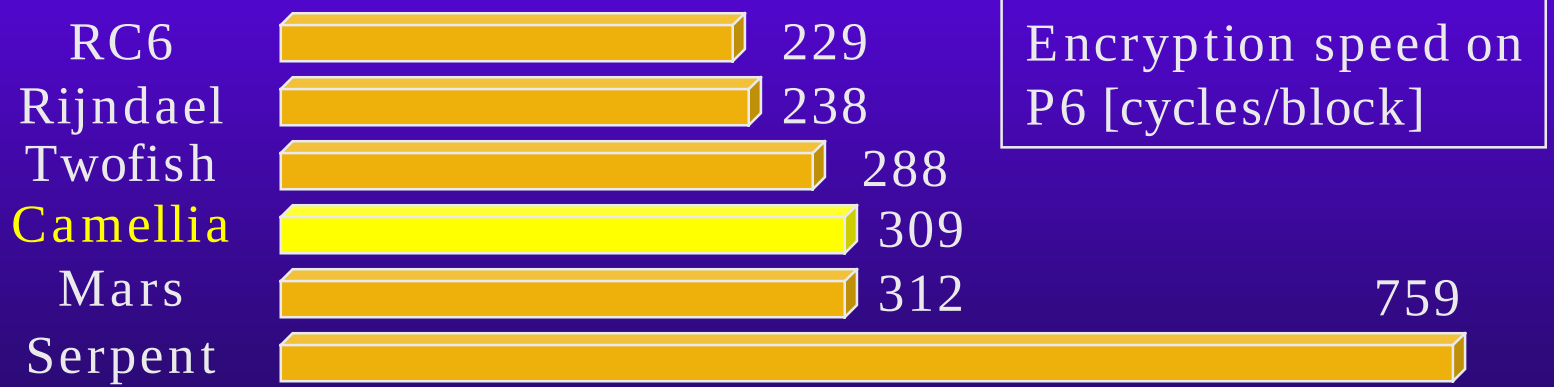
- 128-bit Block Cipher
  - Jointly developed by NTT and Mitsubishi
  - Designed by experienced cryptanalysists and programmers
- Supports 128-, 192-, 256-bit keys
  - Same interface as Advanced Encryption Standard (AES)
  - Offer more security against exhaustive key search

Design Goals

- High level of security
  - State-of-the-art cipher analysis technology
- Efficiency on multiple platforms
  - Software : 8-bit, 32-bit, 64-bit processors
  - Hardware : compact and high-performance

Software Performance (128-bit keys)

- On a Pentium III
  - 309 cycles/block (Assembly)
    = 469Mbps (1.13GHz)
- Much faster than DES
- Comparable speed to the AES finalists



*The programs are written in assembly language by Aoki, Lipmaa, and Osvik. Each figure is the fastest as far as we know.

Hardware (128-bit keys)

- ASIC (0.35μm CMOS)
  - Small Size Hardware 11KGates
    - Smallest among existing 128-bit block ciphers
  - High Performance Hardware

	Area [Kgates]	Throughput [Mbit/s]
MARS	2,936	226
RC6	1,643	204
Rijndael	613	1,950
Serpent	504	932
Twofish	432	394
Camellia	273	1,171
DES*	54	1,161

*DES is a 64-bit block cipher.

The above data (except Camellia) are presented by Ichikawa et al. at the 3rd AES conference.

Security Consideration

- Camellia provides strong security against differential and linear cryptanalysis.
  - Moreover, Camellia was designed to offer security against other advanced cryptanalytic attacks:
    - truncated differential attacks,
    - higher order differential attacks,
    - interpolation attacks,
    - related-key attacks, ...

For more information...

- Camellia Home Page

  http://info.isl.ntt.co.jp/camellia/

  - Specification & Sample code
  - Technical papers on design rationale, performance, software implementation techniques, and security evaluation

  - Internet-Draft on description of Camellia will be available soon!

Public Key Algorithms

EPOC and PSEC

Tatsuaki Okamoto
Shigenori Uchiyama
Eiichiro Fujisaki

NTT

Provable Security of Public Key Algorithms

- *Flaw in RSA with PKCS #1 Ver.1*
  - Importance of security against adaptively chosen ciphertext attacks
- EPOC & PSEC
  - Developed by Okamoto et al. (NTT)
  - Provably secure under the random oracle model in the strongest sense (i.e., non-malleable against adaptively chosen ciphertext attacks)

EPOC (Efficient Probabilistic Public-Key Encryption Scheme)

- Novelty
  - Essentially different from any other previous schemes including RSA-Rabin and Diffie-Hellman
- Security
  - Provably as secure as factoring in the strongest sense
- Efficiency
  - Compared with RSA(PKCS#1 Ver.2) with small e ($2^{16}+1$), encryption speed is slower, but decryption speed is faster.

PSEC (Provably Secure Elliptic Curve Encryption Scheme)

- Security
  - Provably as secure as elliptic-curve Diffie-Hellman problem in the strongest sense
- Efficiency
  - Almost as efficient as most common ECC, elliptic-curve ElGamal (Diffie-Hellman) scheme

Toward International Standards

- EPOC
  - IEEE P1363a (royalty free if selected)
- Camellia
  - ISO/IEC JTC 1/SC27
  - NESSIE (New European Schemes for Signature, Integrity, and Encryption)

Sample Code

- Camellia
  - http://info.isl.ntt.co.jp/camellia/
- EPOC & PSEC
  - http://www.nttmcl.com/sec/

Conclusion

- Camellia is a 128-bit block cipher with high security and performance
  - suitable for bulk encryption
- PSEC and EPOC are public-key algorithms with provable security and efficiency
  - suitable for key exchange and authentication

Conclusion (Cont.)

- Add them to Transport Layer Security!!

  enum { null, rc4, rc2, des, 3des, des0, idea, …, camellia } BulkCipherAlgorithm

  enum { rsa, diffie-hellman, epoc, psec } KeyExchangeAlgorithm

  enum { anonymous, rsa, dsa, epoc, psec } SignatureAlgorithm