

Lecture 7

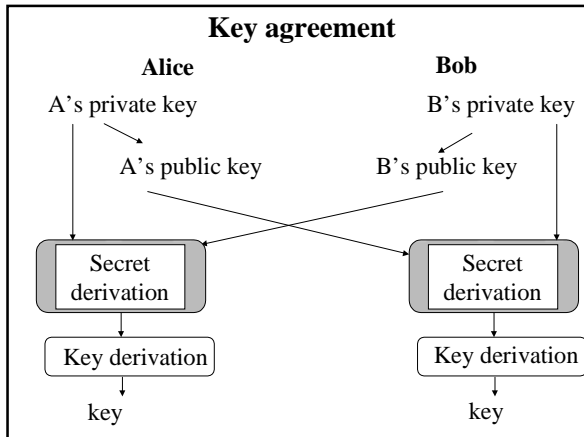
Survey of public key cryptosystems

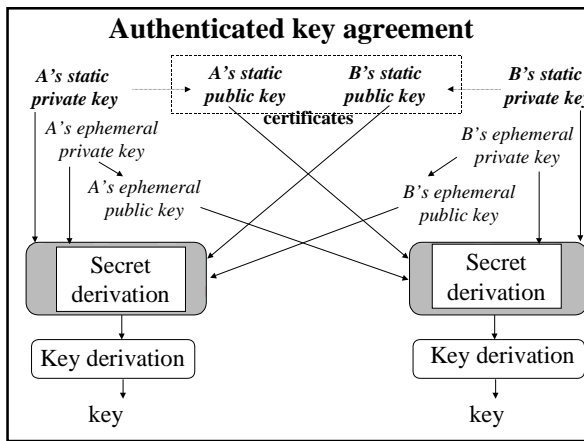
Bases of the public cryptosystems security

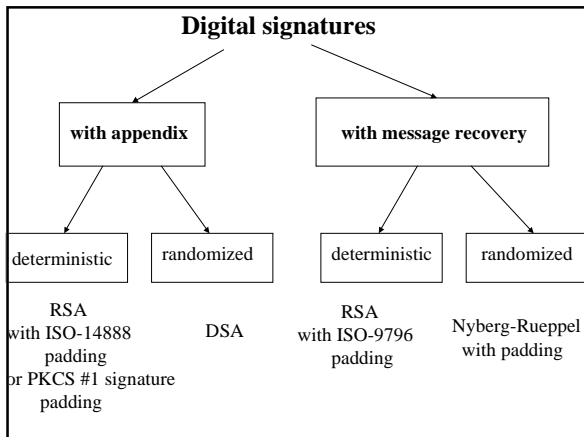
	Factorization	Discrete Logarithm	Elliptic Curve Discrete Logarithm
Given:	$N = p \cdot q$	$y = g^x \text{ mod } p =$ $= \underbrace{g \cdot g \cdot g \cdot \dots \cdot g}_{x \text{ times}}$ constants p, g	$Q = x \cdot P =$ $= \underbrace{P+P+\dots+P}_{x \text{ times}}$ P - point of an elliptic curve
Unknown:	p, q	x	x

Most known public key cryptosystems

	Based on the difficulty of		
	Factorization	Discrete logarithm	Elliptic curve discrete logarithm
Signature	RSA	DSA, N-R	EC-DSA
Encryption	RSA	El-Gamal	EC-El-Gamal
Key agreement	RSA	Diffie-Hellman (DH)	EC-DH







Genesis of DSS

- 1976 public key cryptography, Diffie-Hellman
- 1978 RSA (patent in 1983)
- 1982 NIST solicitation for a public key signature algorithm
- 1984 El Gamal algorithm (not patented)
- 1989 Schnorr algorithm (patent in 1991 in U.S. and many other countries)
- 1990 the primary candidate considered by NIST is RSA
- 1991 NIST announces DSA
- 1994 DSS published as FIPS PUB 186

Digital Signature Algorithm

System parameters

May be shared by a group of users or belong to a single user; known to everybody

q - 160-bit prime
p - L-bit prime, such that $q \mid p-1$
 where $L = 512 + 64 \cdot k$

$g = h^{(p-1)/q} \pmod p$ where $1 < h < p-1$,
 such that $g > 1$

From Fermat's theorem
 $g^q \pmod p = h^{p-1} \pmod p = 1$
 g - generator of the cyclic group of order q
 in Z_p^*

Digital Signature Algorithm

Public and private key

Private key

x - arbitrary 160 bit number $0 < x < q$

Public key

$y = g^x \pmod p$ $0 < y < p$
 L - bit number

DSA: Signature generation

1. Choose random message private key $1 < k < q$ (secret, different for each message)
2. Compute message public key $r = (g^k \text{ mod } p) \text{ mod } q$
3. Compute hash value

Message M

↓

SHA

↓

SHA(M)
4. Compute

$$s = k^{-1} (\text{SHA}(M) + x \cdot r) \text{ mod } q$$

$$\text{SGN}(M) = r \parallel s$$

160 bit
160 bit
40 bytes

DSA: Signature verification

1. Compute hash value

Message M'

↓

SHA

↓

SHA(M')
2. Compute

$$w = (s')^{-1} \text{ mod } q$$
3. Compute $u1 = \text{SHA}(M') \cdot w \text{ mod } q$
4. Compute $u2 = r' \cdot w \text{ mod } q$
5. Compute $v = ((g^{u1} \cdot y^{u2}) \text{ mod } p) \text{ mod } q$
6. Compare

Signature invalid

N

↓

$v=r'$

Y

Signature valid

DSA vs. RSA

Functionality
DSS cannot be used for encryption

<i>Advantages</i>	<i>Disadvantages</i>
export rules much less restrictive certain countries do not allow encryption	additional algorithm must be standardized and implemented for key exchange

DSS can be combined with the Diffie-Hellman key exchange scheme

El-Gamal Encryption

System parameters

May be shared by a group of users or belong to a single user;
known to everybody

p - prime

g - generator of the group \mathbb{Z}_p^*

El-Gamal Encryption

Public and private key

Private key

x - arbitrary number $1 \leq x \leq p-2$

Public key

$y = g^x \bmod p$ $0 < y < p$

El-Gamal: Encryption

1. Choose random
message private key $1 \leq k \leq p-2$,
relatively prime with $p-1$
(secret, different for each message)

2. Compute
message public key
 $r = g^k \bmod p$

3. Compute

$c = y^k \cdot M \bmod p$

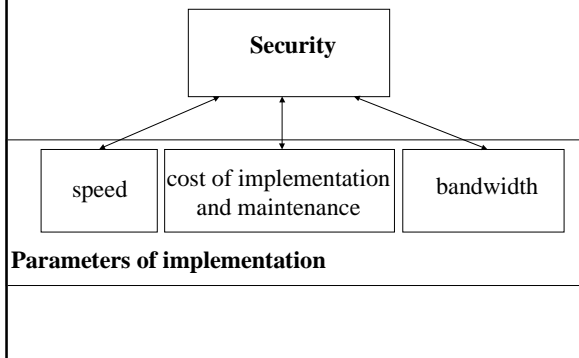
$C(M) = r \parallel c$

El-Gamal: Decryption

$$\begin{bmatrix} r \\ c \end{bmatrix} C(M)$$

$$M = c \cdot (r^x)^{-1} \text{ mod } p$$

Choice of a public key cryptosystem



Strategy of fair comparison

All algorithms have a **variable key length**

Best attacks specific for each cryptosystem

Security of various cryptosystems depends to a different extent on the key length

Comparison of implementation characteristics (in particular speed) under the assumption that selected key sizes guarantee the same security level

Best known attacks			
Basis of the cryptosystem security	Factorization	Discrete Logarithm	Elliptic Curve Discrete Logarithm
Best known attack	General Number Field Sieve	1. General Number Field Sieve 2. Parallel collision search	2. Parallel collision search
Complexity of the attack:	subexponential	1. subexponential 2. exponential	exponential

Best known attacks			
Basis of the cryptosystem security	Factorization	Discrete Logarithm	Elliptic Curve Discrete Logarithm
Cryptosystem	RSA	DSA, DH	EC-DSA EC-DH
Security parameter	Modulus N	1. Length of the modulus p 2. Size q of the subgroup generated by g	Size q of the subgroup generated by P
Typical lengths of the security parameter (in bits)	768-2048	1. 768-2048 2. 160 (for DSA)	140-200

Theoretical computational security of the best known attacks	
Basis of the cryptosystem security	Complexity of the best known attack
Factorization	subexponential $L_N[1/3, 1.92] = \exp((1.92 + o(1)) \cdot (\ln N)^{1/3}) \cdot (\ln \ln N)^{2/3}$
Discrete Logarithm	subexponential $L_p[1/3, 1.92] = \exp((1.92 + o(1)) \cdot (\ln p)^{1/3}) \cdot (\ln \ln p)^{2/3}$
Elliptic Curve Discrete Logarithm	exponential $(\pi \cdot q / 2)^{1/2/r}$ r - number of processors working in parallel

Practical records			
Basis of the cryptosystem security	Factorization	Discrete Logarithm	Elliptic Curve Discrete Logarithm
Number of bits of the security parameter	512	283?	108
Challenges regarding breaking the cryptosystem	RSA Data Security Challenge, 1991-	—	Certicom challenge, 1997-

Practical implementations of attacks				
Factorization, RSA				
Year	Number of bits of N	Number of decimal digits of N	Method	Estimated amount of computations
1994	430	129	QS	5000 MIPS-years
1996	433	130	GNFS	750 MIPS-years
1998	467	140	GNFS	2000 MIPS-years
1999	467	140	GNFS	8000 MIPS-years

Challenge published in Scientific American	
Ciphertext:	<i>1977</i>
9686 9613 7546 2206 1477 1409 2225 4355 8829 0575 9991 1245 7431 9874 6951 2093 0816 2982 2514 5708 3569 3147 6622 8839 8962 8013 3919 9055 1829 9451 5781 5145	
Public key:	
N = 114381625757 88886766923577997614 661201021829672124236256256184293 570693524573389783059712356395870 5058989075147599290026879543541	
e = 9007	(129 decimal digits)
<i>Award 100 \$</i>	

Rivest estimation - 1977

The best known algorithm for factoring a 129-digit number requires:

**40 000 trillion years
= $40 \cdot 10^{15}$ years**

assuming the use of a supercomputer being able to perform

1 multiplication of 129 decimal digit numbers in 1 ns

Rivest's assumption translates to the delay of a single logic gate ≈ 10 ps

Estimated age of the universe: 100 bln years = 10^{11} years

Early records in factoring large numbers

Years	Number of decimal digits	Number of bits	Required computational power (in MIPS-years)
1974	45	149	0.001
1984	71	235	0.1
1991	100	332	7
1992	110	365	75
1993	120	398	830

How to factor for free?

A. Lenstra & M. Manasse, 1989

- **Using the spare time of computers, (otherwise unused)**

- **Program and results sent by e-mail (later using WWW)**

Breaking RSA-129

When: August 1993 - 1 April 1994, **8 months**
Who: D. Atkins, M. Graff, A. K. Lenstra, P. Leyland
+ 600 volunteers from the entire world
How: **1600 computers**
from Cray C90, through 16 MHz PC,
to fax machines

Only 0.03% computational power of the Internet

Results of cryptanalysis:

“The magic words are squeamish ossifrage”

An award of 100 \$ donated to Free Software Foundation

Elements affecting the progress in factoring large numbers

- computational power
1977-1993 increase of about 1500 times
- computer networks
Internet
- **better algorithms**

RSA Challenge

RSA-100
RSA-110
RSA-120
RSA-130
RSA-140

RSA-150
RSA-160
RSA-170
RSA-180
.....
RSA-450
RSA-460
RSA-470
RSA-480
RSA-490
RSA-500

Smallest unfactored number

RSA-150

Unused awards accumulate at a rate
of \$1750 / quarter

Factoring 512-bit number

512 bits = 155 decimal digits
old standard for key sizes in RSA

17 March - 22 August 1999

Group of Herman te Riele
Centre for Mathematics and Computer Science
(CWI), Amsterdam

First stage 2 months

168 workstations SGI and Sun, 175-400 MHz
120 Pentium PC, 300-450 MHz, 64 MB RAM
4 stations Digital/Compaq, 500 MHz

Second stage

Cray C916 - 10 days, 2.3 GB RAM

TWINKLE

“The Weizmann INstitute Key Locating Engine”

Adi Shamir, Eurocrypt, May 1999
CHES, August 1999

Electrooptical device capable to speed-up
the first phase of factorization from 100 to 1000 times

If ever built it would increase the size of the key
that can be broken from 100 to 200 bits

Cost of the device (assuming that the prototype was
earlier built) - \$5000

Recommended key sizes for RSA

Old standard:

Individual users

~~512 bits~~
(155 decimal digits)

New standard:

Individual users

768 bits
(231 decimal digits)

Organizations (short term)

1024 bits
(308 decimal digits)

Organizations (long term)

2048 bits
(616 decimal digits)

Practical implementations of attacks				
Discrete logarithm, DSA, DH				
Year	Number of bits of p	Number of decimal digits of p	Method	Estimated amount of computations
1990	191	57	NFS-COS	31 MIPS-years
1996	248	74	NFS-DL	
1998	283	85	NFS-COS	
1998	430	129 (p of the special form)	SNFS	

Practical implementations of attacks					
Elliptic curve discrete logarithm problem, ECC-DSA, DH					
Year	Curve	Number of bits of q	Number of decimal digits of q	Method	Number of group operations
II.1998	ECC2-89	89	27	ρ -Pollard	1.8×10^{13}
I.1998	ECCp-89	89	27	ρ -Pollard	3.0×10^{13}
V.1998	ECC2K-95	95	29	ρ -Pollard	2.2×10^{13}
III.1998	ECCp-97	97	30	ρ -Pollard	2.0×10^{14}
IX.1999	ECC2-97	97	30	ρ -Pollard	1.0×10^{14}
IV. 2000	ECC2K-108	108	33	ρ -Pollard	2.0×10^{15}

IEEE P1363			
	factorization	discrete logarithm	Elliptic curve discrete logarithm
encryption	RSA with OAEP		
signature	RSA & R-W with ISO-14888 or ISO 9796	DSA, NR with ISO 9796	EC-DSA, EC-NR with ISO 9796
key agreement		DH1, DH2 and MQV	EC-DH1, EC-DH2 and EC-MQV

IEEE P1363a			
	factorization	discrete logarithm	Elliptic curve discrete logarithm
encryption	RSA with OAEP	new scheme	new scheme
signature	RSA & R-W with ISO-14888 or ISO 9796	DSA, NR with ISO-9796	EC-DSA, EC-NR with ISO 9796
key agreement	new scheme	DH1 DH2 & MQV	EC-DH1 EC-DH2 & EC-MQV

IEEE P1363

Working group of IEEE including representatives of major cryptographic companies and university centers from USA, Canada and other countries

Part of the Microprocessors Standards Committee

Modern, open style

Quarterly meetings + multiple teleconferences +
+ discussion list + very informative web page with the draft versions of standards

IEEE P1363

Combined standard including the majority of modern public key cryptography

Several algorithms for implementation of the same function

Tool for constructing other, more specific standards

Specific applications or implementations may determine a profile (subset) of the standard

PKCS
Public-Key Cryptography Standards
Informal Industry Standards

developed by RSA Laboratories

in cooperation with
Apple, Digital, Lotus, Microsoft, MIT, Northern
Telecom, Novell, Sun

First, except PGP, formal specification of RSA
and formats of messages.

ANSI X9
American National Standards Institute

Work in the subcommittee X9F
developing standards for **financial institutions**

Standards for the wholesale
(e.g., interbank)
and retail transactions
(np. bank machines, smart card readers)

ANSI represents U.S.A. in **ISO**

NIST FIPS
National Institute of Standards and Technology
Federal Information Processing Standards

American Federal Standards

Required in the government institutions

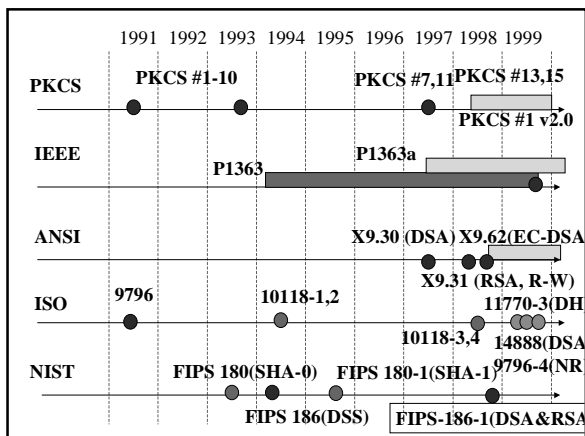
Original algorithms developed in cooperation
with the National Security Agency (NSA)

Secure key sizes

	factorization	Discrete logarithm	Elliptic curve discrete logarithm
PKCS			
IEEE P1363			
ANSI X9	≥ 1024	≥ 1024	≥ 160
NIST FIPS		≥ 1024	
ISO			

Padding schemes

	encryption	Signatures with appendix	Signatures with message recovery
PKCS	OAEP PKCS #1	PKCS #1	
IEEE P1363	OAEP	ISO 14888	ISO 9796
ANSI X9	OAEP	ISO 14888	ISO 9796
NIST FIPS			
ISO		ISO 14888	ISO 9796



Elliptic Curve Cryptosystems - ECC

Advantages

- first true alternative for RSA
- several times shorter keys
- fast and compact implementations, in particular in hardware
- a family of cryptosystems, instead of a single cryptosystem

Elliptic Curve Cryptosystems - ECC

Disadvantages

- complex mathematical description
- short period of research on the cryptanalysis

Elliptic Curve Cryptosystems vs. RSA

Certicom	RSA Data Security Inc.
ECC	RSA ECC
Security Builder	BSAFE
Efficient software and hardware implementations	Efficient software implementations
ECC - "cryptography of the XXI century"	ECC - strong low-risk cryptography

Fact or myth?

RSA is much more secure because the factorization problem was studied much longer than elliptic curve discrete logarithm problem

Factorization problem studied intensively since the end of 70's	Studies on factorization before the era of computers and computer networks is irrelevant
Elliptic curve discrete logarithm problem studied intensively since the beginning of 90's	Studies on attacks against discrete logarithms in GF(p) conducted earlier. Many of these attacks apply to the elliptic curve discrete logarithms.

Progress in algorithms for solving the discrete logarithm problem

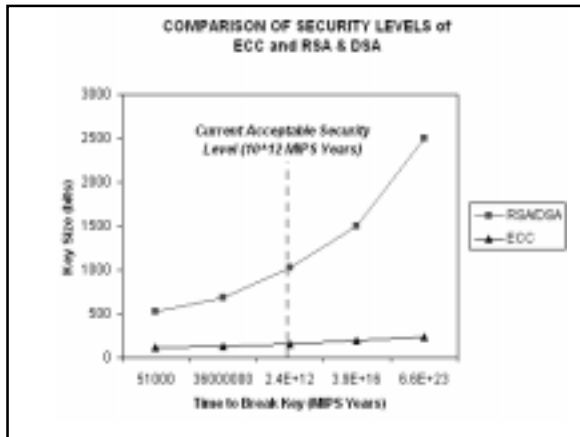
1997 *N. Smart*
1997 *T. Satoh, K. Araki*
Fast algorithm for a special class of curves
7.04.98 *R. Gallant, R. Lambert, S. Vanstone*; Certicom
8.04.98 *M. Wiener i R. Zuccherato*; Entrust
Algorithm speeding up computations $\sqrt{2}m$ times for Koblitz curves over $GF(2^m)$
For a randomly selected curve, neither attack applies

Workshops on Elliptic Curve Cryptography, since 1997
Sponsors: MasterCard, Mondex, etc.

Fact or myth?

Key length necessary to obtain the same level of security for RSA and Elliptic Curve Cryptosystems grows faster for RSA

True, if one takes into account only the number of operations necessary to conduct the attack	Untrue , if one takes into account much larger memory requirements for attacks against RSA
------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------



**RAM requirements
in the NFS factorization method**

Number of bits of N	Memory in the first phase of the algorithm (clients)	Memory in the second phase of the algorithm (server)
428	64 MB	2 GB
512	160 MB	20 GB
1024	256 GB	~100 TB

Equivalent key sizes
according to Robert Silverman, RSA Inc., 1999

Assumption: The same amount of arithmetic operations

RSA/DSA	ECC	Symmetric ciphers	Number of arithmetic operations
512	119	56	$1,7 \times 10^{19}$
768	144	69	$1,1 \times 10^{23}$
1024	163	79	$1,3 \times 10^{26}$
2048	222	100	$1,5 \times 10^{35}$

Equivalent key sizes
according to Michael Wiener, Entrust Technologies

Basic assumption: *The same number of instructions in MIPS-years*

RSA/DSA	ECC		Number of instructions w MIPS-years
Software attack	Software attack	Hardware attack	
1024	138	170	3×10^{11}

Equivalent key sizes
according to Michael Wiener

Detailed assumptions (1)

Hardware attack based on ASICs:

- clock frequency 64 MHz
- 70 levels of pipelining
- cost \$16

Equivalent key sizes
according to Michael Wiener

Detailed assumptions (2)

Number of PCs, 300 MHz, necessary to break RSA-1024

2^{30} PC-years

Number of ASICs necessary to break ECC-k

$2^{k/2 - 51}$ ASIC-years

Equivalent key sizes
according to Michael Wiener

Detailed assumptions (3)

Cost of access to a PC
\$250

Cost of an ASIC
\$16

1 PC-year \approx 16 ASIC-years

k=170

Digital Signature Timings
Pentium Pro, 200 MHz, Michael Wiener, Entrust

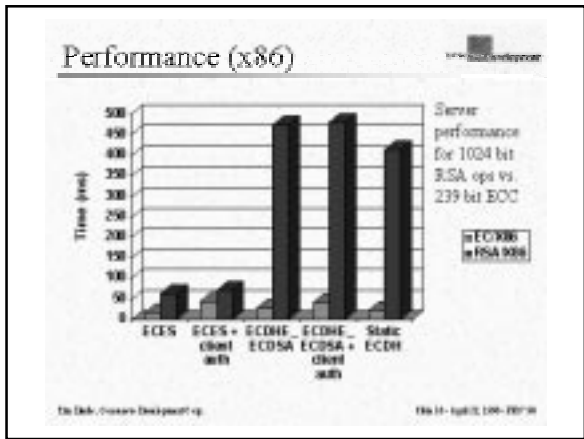
	RSA-1024 (e=3)	DSA-1024	ECDSA-170
Signature generation	43 ms	7 ms	5 ms
Signature verification	0.6 ms	27 ms	19 ms
Key generation	1100 ms	7 ms	7 ms

Digital Signature Timings
Pentium Pro, 180 MHz, Scott Contini, RSA DSI

	RSA-1024 (e=3)	DSA-1024	ECDSA-170
Signature generation	47 ms	28 ms	6 ms
Signature verification	1 ms	52 ms	30 ms

Key Agreement Timings
Pentium Pro, 200 MHz, BSDi 4.0 Unix
Certicom

	DH-1024 exponent 1024 bits	DH-1024 exponent 160 bits	EC-DH-163
Secret sharing	108 ms	17 ms	5 ms
Key generation	92 ms	14 ms	5 ms



Key Agreement Timings
Pentium Pro, 180 MHz
Scott Contini, RSA DSI

	RSA-1024 (e=3)	DH-1024 exponent 160 bits	EC-DH-170
Sender (client)	1 ms	52 ms	28 ms
Receiver (server)	47 ms	26 ms	22 ms

Binary code size			
	RSA	DSA	EC-DSA
Generation of system parameters	N/A	small	very large
Key generation	medium	very small	very small
Core operations	small	small	medium

Which cryptosystem is the best? (1)

Secure electronic mail

- speed of operations is not critical, security and trust of customers are more important
- message encrypted using a symmetric key cryptosystem

A key for a symmetric key cryptosystem encrypted once for each receiver

All operations performed by a sender

A key for a symmetric key cryptosystem decrypted separately by each receiver

Load distributed among receivers

Advantage: RSA

Which cryptosystem is the best? (2)

Use in public key certificates

- each certificate and CRL are signed only once but verified hundreds of times

Advantage: RSA

Which cryptosystem is the best? (3)

Wireless communication

- large cost of transmission
- shorter keys in ECCs
- shorter signatures and certificates in ECCs and DSA
- shorter messages in the key agreement schemes based on ECCs

Advantage: ECC

Which cryptosystem is the best? (4)

Hardware implementation

- small area of integrated circuits implementing ECC, in particular ECCs over $GF(2^m)$
- faster decryption and key generation

Advantage: ECC

Which cryptosystem is the best? (5)

Smart cards

ECCs

- smaller EEPROM requirements
- do not require an arithmetic coprocessor (at least for a class of curves over $GF(2^m)$)
- smaller requirements on the interface with a card reader
- allow to generate a key on the card

Advantage: ECC

American Standards			
	RSA	DSA, DH	EC-DSA EC-DH
Federal		FIPS 186	
Banking	X9.31	X9.30 X9.42	X9.62 X9.63
Industry	IEEE P1363 PKCS-1	IEEE P1363 PKCS-2	IEEE P1363 PKCS-13

Standard Internet Protocols	
Secure e-mail	
S/MIME v.2	RSA
v.3	RSA, DSA, DH
Secure WWW	
SSL v. 3.0	RSA, DSA, DH, proposed extension with ECCs
Secure payment card protocols	
SET	RSA, proposed extension with ECCs
Virtual Private Networks	
IPSec	DH, EC-DH

Patents - only U.S. and Canada		
RSA	DSA, DH	EC-DSA, EC-DH
Patent expired in 2000	DH Patent expired in 1997	No patents for cryptosystems themselves. Over 40 patent petitions regarding implementation details, <i>Certicom Inc.</i>

Summary

- RSA in common use, ECC struggle to enter the market
- New standards will support all three types of cryptosystems
- ECC particularly advantages in environments with limited bandwidth and storage (e.g., cellular telephones, pagers, smart cards)
- If there is no breakthrough in cryptanalysis the market will be shared among two (or three) classes of cryptosystems
