ELLIPTIC CURVE CRYPTOSYSTEMS — READY FOR PRIME TIME

Alfred Menezes

University of Waterloo
ajmeneze@math.uwaterloo.ca

January 29, 1998

Outline

1. Introduction

- Discrete-log cryptographic protocols are usually described in the algebraic setting of the group $\mathbb{Z}_p^*$ (the multiplicative group of the integers modulo a prime p).
- These include Diffie-Hellman key agreement, ElGamal encryption, and the ElGamal signature scheme.
- They can also be described in the more abstract setting of a finite cyclic group G.

Diffie-Hellman key agreement

Objective: Alice and Bob establish a shared secret by communicating over an unsecured but authentic channel.

1. Public parameters: A prime p and a generator g of $\mathbb{Z}_p^*$.
2. Alice generates a random integer a, $1 \le a \le p-2$, and sends g^a to Bob.
3. Bob generates a random integer b, $1 \le b \le p-2$, and sends g^b to Alice.
4. Alice computes $K = (g^b)^a$.
5. Bob computes $K = (g^a)^b$.
6. The shared secret is $K = g^{ab}$.

Among the groups proposed:

- Multiplicative group of a finite field $\mathbb{F}_q$ (Diffie and Hellman, 1976).
- Group of points on an elliptic curve over a finite field (Koblitz, Miller, 1985).
- Class group of an imaginary quadratic number field (Buchmann, Williams, 1988).
- Subgroup of the multiplicative group of $\mathbb{Z}_p$ (Schnorr, 1989).
- Jacobian of a hyperelliptic curve defined over a finite field (Koblitz, 1989).

2. The Digital Signature Algorithm (DSA)

- Variant of ElGamal and Schnorr schemes.
- Proposed in 1991.
- US Federal Information Processing Standard (FIPS-180).
- Exploits small subgroups in $\mathbb{Z}_p^*$ in order to decrease the size of signatures.

DSA system parameter generation

1. Select a prime q such that $2^{159} < q < 2^{160}$.
2. Select a 1024-bit prime number p with the property that $q \mid p-1$.
3. (Select a generator g of the unique cyclic group of order q in $\mathbb{Z}_p^*$.)
   3.1. Select an element $h \in \mathbb{Z}_p^*$ and compute $g = h^{(p-1)/q} \bmod p$. (Repeat until $g \neq 1$.)
4. System parameters are p, q and g.

DSA key generation

Each entity A does the following:

1. Select a random integer x such that $1 \leq x \leq q-1$.
2. Compute $y = g^x \bmod p$.
3. A's public key is y; A's private key is x.

DSA signature generation

To sign a message m, A does the following:

1. Select a random integer k, $1 \leq k \leq q-1$.
2. Compute $r = (g^k \bmod p) \bmod q$.
3. Compute $k^{-1} \bmod q$.
4. Compute $s = k^{-1}\{h(m) + xr\} \bmod q$. If $s = 0$ then go to step 1.
5. The signature for the message m is (r, s).

- The signature is 320 bits in length.
- h is the hash function SHA-1.

DSA signature verification

To verify A's signature (r, s) on m, B should do the following:

1. Compute $w = s^{-1} \bmod q$ and $h(m)$.
2. Compute $u_1 = h(m)w \bmod q$ and $u_2 = rw \bmod q$.
3. Compute $v = (g^{u_1} y^{u_2} \bmod p) \bmod q$.
4. Accept the signature if and only if $v = r$.

Discrete logarithm problem

The security of DSA is based on the difficulty of the *discrete logarithm problem* (DLP):

Given a prime p, a generator g of $\mathbb{Z}_p^*$, and $y = g^x \bmod p$, find x.

3. Background on elliptic curves

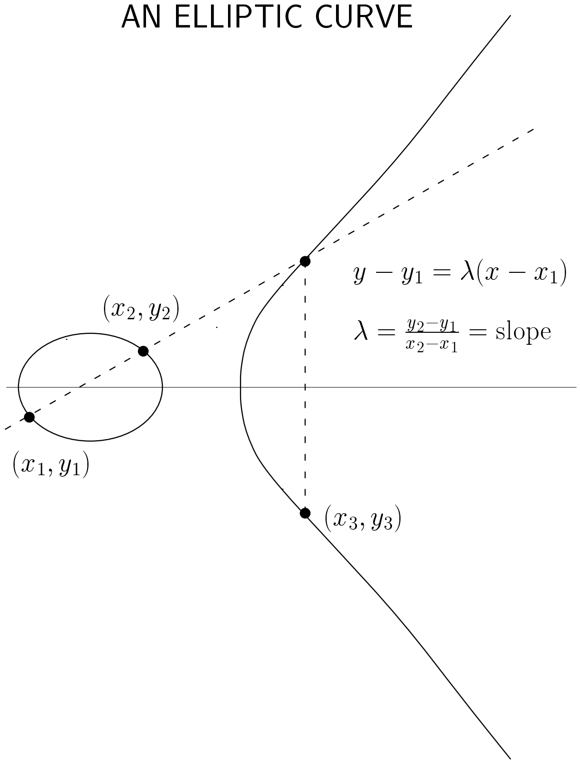
- Let $\mathbb{Z}_p$ be the set of integers modulo a prime p ($p > 3$).
- An *elliptic curve* E over $\mathbb{Z}_p$ is defined by an equation of the form
$$y^2 = x^3 + ax + b,$$
where $a, b \in \mathbb{Z}_p$, $(4a^3 + 27b^2) \not\equiv 0 \pmod{p}$, together with the *point at infinity* $\mathcal{O}$.
- The set $E(\mathbb{Z}_p)$ consists of all points (x, y), $x \in \mathbb{Z}_p$, $y \in \mathbb{Z}_p$, which satisfy the defining equation, together with $\mathcal{O}$.

An example over $\mathbb{Z}_{23}$

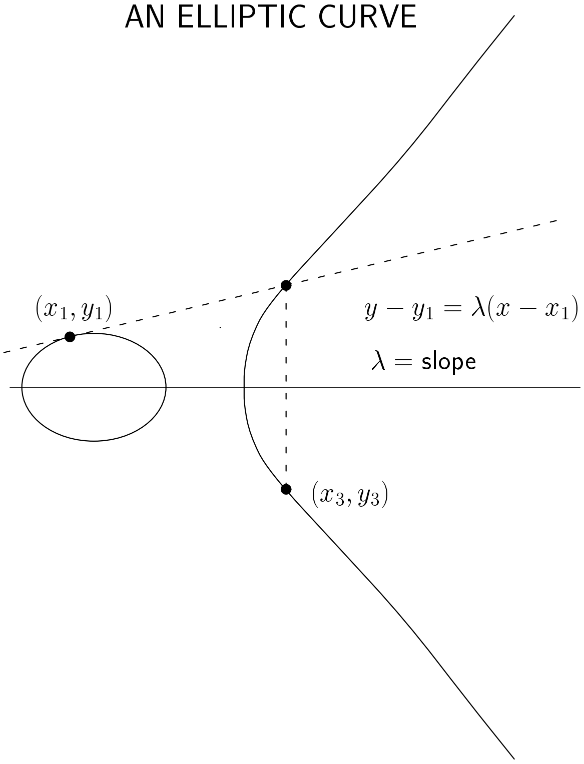
- Let $p = 23$.
- $y^2 = x^3 + x + 1$, (i.e. $a = 1$, $b = 1$).
- $E(\mathbb{Z}_{23}) = \{(x, y) : y^2 = x^3 + x + 1\} \cup \{\mathcal{O}\}$.
- Solutions to $y^2 = x^3 + x + 1$ over $\mathbb{Z}_{23}$:

(0,1)	(5,4)	(9,16)	(17,3)
(0,22)	(5,19)	(11,3)	(17,20
(1,7)	(6,4)	(11,20)	(18,3)
(1,16)	(6,19)	(12,4)	(18,20)
(3,10)	(7,11)	(12,19	(19,5)
(3,13)	(7,12)	(13,7)	(19,18)
(4,0)	(9,7)	(13,16)	

AN ELLIPTIC CURVE



AN ELLIPTIC CURVE



Addition formula

$E : y^2 = x^3 + ax + b$.

- $\mathcal{O} + \mathcal{O} = \mathcal{O}$.
- $(x, y) + \mathcal{O} = (x, y)$ for all $(x, y) \in E$.
- $(x, y) + (x, -y) = \mathcal{O}$ for all $(x, y) \in E$. (i.e. $-(x, y) = (x, -y)$).
- Let $P = (x_1, y_1)$, $Q = (x_2, y_2)$, $P \neq \pm Q$. Then $P + Q = (x_3, y_3)$, where
  $x_3 = \lambda^2 - x_1 - x_2$
  $y_3 = \lambda(x_1 - x_3) - y_1$ and

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & P \neq Q \\ \dfrac{3x_1^2 + a}{2y_1} & P = Q. \end{cases}$$

Examples of addition in $E(\mathbb{Z}_{23})$.

1. $P_1 = (3,10)$, $P_2 = (9,7)$,
   $P_1 + P_2 = (x_3, y_3)$.

$$\lambda = \frac{7-10}{9-3} = \frac{-3}{6} = \frac{-1}{2} = 11 \in \mathbb{Z}_{23}.$$

$x_3 = 11^2 - 3 - 9 = 6 - 3 - 9 = 17$,
$y_3 = 11(3 - (-6)) - 10 = 11(9) - 10$
$= 89 = 20 = 20$.
Therefore $P_1 + P_2 = (17, 20)$.

2. $P_1 = (3,10)$, $2P_1 = (x_3, y_3)$,

$$\lambda = \frac{3(3^2)+1}{20} = \frac{5}{20} = \frac{1}{4} = 6.$$

$x_3 = 6^2 - 6 = 30 = 7$,
$y_3 = 6(3-7) - 10 = -24 - 10 = 12$.
Therefore $2P_1 = (7, 12)$.

Basic facts

- There are about $2p$ "different" elliptic curves over $\mathbb{Z}_p$.
- $E(\mathbb{Z}_p)$ is an abelian group with identity $\mathcal{O}$.
- The number of points on the elliptic curve is $\#E(\mathbb{Z}_p) = p + 1 - t$, where $|t| \leq 2\sqrt{p}$. Hence, $\#E(\mathbb{Z}_p) \approx p$.
- $\#E(\mathbb{Z}_p)$ can be computed in polynomial time using Schoof's algorithm.
- The above results are also true if $\mathbb{Z}_p$ is replaced by any finite field $\mathbb{F}_q$.

Example

- $E : y^2 = x^3 + x + 1$ over $\mathbb{Z}_{23}$.
- $\#E(\mathbb{Z}_{23}) = 28$.
- $E(\mathbb{Z}_{23})$ is a cyclic group, and $P = (0,1)$ is a generator:

P=(0, 1)	$15P$=(9, 7)
$2P$=(6,19)	$16P$=(17,3)
$3P$=(3,13)	$17P$=(1,7)
$4P$=(13,16)	$18P$=(12,19)
$5P$=(18,3)	$19P$=(19,5)
$6P$=(7,11)	$20P$=(5,4)
$7P$=(11,3)	$21P$=(11,20)
$8P$=(5,19)	$22P$=(7 12)
$9P$=(19,18)	$23P$=(18,20)
$10P$ =(12,4)	$24P$=(13,7)
$11P$=(1,16)	$25P$=(3,10)
$12P$=(17,20)	$26P$=(6,4)
$13P$=(9,16)	$27P$=(0,22)
$14P$=(4,0)	$28P$=$\mathcal{O}$

$\mathbb{Z}_p^*$ and $E(\mathbb{F}_q)$ correspondence

Group	$\mathbb{Z}_p^*$	$E(\mathbb{F}_q)$
Group elements	Integers $\{1, 2, \ldots, p-1\}$	Points (x, y) on E plus $\mathcal{O}$
Group operation	multiplication modulo p	addition of points
Notation	Elements: g, h Multiplication: $g \cdot h$ Inverse: g^{-1} Division: g/h Exponentiation: g^a	Elements: P, Q Addition: $P + Q$ Negative: $-P$ Subtraction: $P - Q$ Multiple: aP
Discrete Logarithm Problem	Given $g \in \mathbb{Z}_p^*$ and $h = g^a \bmod p$, find a	Given $P \in E(\mathbb{F}_q)$ and $Q = aP$, find a.

4. ECDSA

- The Elliptic Curve Digital Signature Algorithm (ECDSA) is the elliptic curve analogue of the DSA.
- Under consideration by IEEE, ANSI, FIPS, ISO as signature standards.

ECDSA system parameter generation

1. Select an elliptic curve E defined over $\mathbb{F}_q$.
2. $\#E(\mathbb{F}_q)$ should be divisible by a large prime n.
3. Select a point P of order n in $E(\mathbb{F}_q)$.
4. The system parameters are E, P and n.

ECDSA key generation

Each entity A does the following:

1. Select a random integer d in the interval $[1, n-1]$.
2. Compute $Q = dP$.
3. A public key is Q; A's private key is d.

DSA and ECDSA notation correspondence

DSA notation	ECDSA notation
q	n
g	P
x	d
y	Q

ECDSA signature generation

To sign a message m, A does the following:

1. Select a random integer k, $1 \leq k \leq n-1$.
2. Compute $kP = (x_1, y_1)$ and $r = x_1 \bmod n$. If $r = 0$ then go to step 1.
3. Compute $k^{-1} \bmod n$.
4. Compute $s = k^{-1}\{h(m) + dr\} \bmod n$. If $s = 0$ then go to step 1.
5. The signature for the message m is (r, s).

- If n is a 160-bit prime, then the signature is 320 bits in length.
- h is the hash function SHA-1

ECDSA signature verification

To verify A's signature (r, s) on m, B should do the following:

1. Verify that r and s are integers in the interval $[1, n-1]$.
2. Compute $w = s^{-1} \bmod n$ and $h(m)$.
3. Compute $u_1 = h(m)w \bmod n$ and $u_2 = rw \bmod n$.
4. Compute $u_1P + u_2Q = (x_1, y_1)$ and $v = x_1 \bmod n$.
5. Accept the signature if and only if $v = r$.

Elliptic Curve Discrete logarithm problem

The security of ECDSA is based on the difficulty of the *elliptic curved discrete logarithm problem* (ECDLP):

Given an elliptic curve E defined over $\mathbb{F}_q$,, a point P of order n, $Q = dP$, find d.

5. The RSA signature scheme

RSA key generation

Each entity A does the following:

1. Select large random primes p and q.
2. Compute $n = pq$ and $\phi = (p-1)(q-1)$.
3. Select an integer e, $1 \le e \le \phi - 1$, such that $\gcd(e, \phi) = 1$.
4. Compute the integer d, $1 \le d \le \phi - 1$, such that $ed \equiv 1 \pmod{\phi}$.
5. A's public key is (n, e); A's private key is d.

RSA signature generation

To sign a message M, A does the following:

1. Compute $m = H(M)$.
2. Compute $s = m^d \bmod n$.
3. The signature for M is s.

RSA signature verification

To verify A's signature s on M, B should do the following:

1. Compute $m = H(M)$.
2. Compute $m' = s^e \bmod n$.
3. Accept the signature if and only if $m = m'$.

Integer factorization problem

The security of RSA is based on the difficulty of the *integer factorization problem* (IFP):

Given an integer n that is a product of two distinct primes p and q, find p and q.

6. Evaluation criteria

- (Perceived) security.
- Key lengths.
- Signature size.
- Speed.
- Storage (precomputation?).
- Complexity of implementation (code size, gate count, power consumption, etc.).
- Platforms (hardware, software, firmware).
- Industry/government standards.
- Patent coverage.
- Licensing terms.

7. Security

History of math problems

	IFP	DLP and ECDLP
$\approx$ 1920s	Random squares (Kraitchik)	Index-calculus (Kraitchik)
1975	Continued fraction	
1976		DLP proposed for use in cryptography
1977	RSA proposed	
1979		Index-calculus
1982	Quadratic sieve	

	IFP	DLP and ECDLP
1985		ECDLP proposed for use in cryptography
1990	Number field sieve	Number field sieve for DLP
1991		Reduction for supersingular curves (for ECDLP)
1994		Subexponential-time algorithm for high-genus hyperelliptic curves
1995		Trace 1 curves are weak (Semaev) (for ECDLP)
1998	?	?

Some questions to ponder:

1. Has the integer factorization problem indeed been *seriously* studied by thousands of mathematicians for hundreds of years?
2. Has the integer factorization problem been more carefully studied than the discrete logarithm problem?
3. Is the research on the discrete logarithm problem prior to 1985 of any relevance/significance to the elliptic curve discrete logarithm problem?

4. Have there been many more research papers on the security and/or implementation of RSA than that of elliptic curve cryptosystems?
5. Is it true that the elliptic curve discrete logarithm problem is not well understood due to the *abstruse nature* of elliptic curves? (As compared, say, to the number field sieve.)

My opinion:

While it is true that the integer factorization problem has been more heavily scrutinized than the elliptic curve discrete logarithm problem, the difference in the efforts these problems have received has been exaggerated.

Attacks on underlying math problems

1. Integer factorization problem (IFP):
   - Number field sieve $L_n[\frac{1}{3}, 1.923]$ (*subexponential-time* algorithm).
   - Easily parallelized in software.
   - Record: 130-decimal digit, 500 MIPS years.
   - Challenge: 512-bit RSA number.

2. Discrete logarithm problem (DLP):
   - Number field sieve (for $\mathbb{Z}_p$)
   - $L_p[\frac{1}{3}, 1.923]$ (*subexponential-time* algorithm).
   - Easily parallelized in software.
   - Record: 75-decimal digit (248 bits).
   - Challenge: $p = 2 \cdot 739 \cdot \frac{(7^{149}-1)}{6} + 1$ (427 bits).

3. Elliptic curve discrete logarithm problem (ECDLP): ($\#E(\mathbb{F}_q)$ divisible by large prime n)
   - Pollard-ρ
     Expected running time: $\sqrt{\frac{\pi n}{2}}$.
   - Distributed version (van Oorschot/Wiener)
     m processors: $\sqrt{\frac{\pi n}{2}}/m$
     (*fully exponential-time* algorithm).
   - Easily parallelized in hardware.
   - Certicom ECC Challenge – www.certicom.com.
     Challenges: 109, 131, 163, 191, 239, and 359-bits.

Factoring estimates (software)

(A. Odlyzko – CryptoBytes, Summer 1995)

Size of n (in bits)	MIPS years
512	3×10^4
768	2×10^8
1024	3×10^{11}
1280	1×10^{14}
1536	3×10^{16}
2048	3×10^{20}

Computing power available (MIPS years):

	covert attack	open project
2004	10^8	2×10^9
2014	$10^{10} - 10^{11}$	$10^{11} - 10^{13}$

(2014: 10^{10} people, 10 computers/person, typical computer rated at $10^4 - 10^5$ MIPS.)

EC discrete logarithm estimates

- **Software**
  Assumption: a 1 MIPS machine can perform 4×10^4 EC additions/sec, or 2^{40} EC additions/year.

Size of n ($\approx q$) (in bits)	MIPS years
160	9.6×10^{11}
186	7.9×10^{15}
234	1.3×10^{23}
354	1.5×10^{41}

- **Hardware** (van Oorschot/Wiener,1994)
  - \$10 million
  - 325,000 processors
  - $n \approx 2^{120}$
  - 1 logarithm in 35 days

8. Comparison

- Since no subexponential-time algorithm is known for the general ECDLP, a smaller underlying finite field $\mathbb{F}_q$ can be chosen (compared to traditional discrete log systems).
- A smaller field results in the following benefits of elliptic curve systems:
  - smaller key sizes (and certificates)
  - smaller signature sizes
  - bandwidth savings
  - smaller hardware processors
  - low power requirements
  - efficient implementations.

- RSA: 1024-bit modulus n.
- DSA: 1024-bit p, 160-bit q.
- ECDSA: 160-bit n (so q is $160+\epsilon$ bits).

Parameter sizes

	ECDSA (160-bit q)	RSA (1024-bit n, $e=2^{16}+1$)	DSA (1024-bit p, 160-bit q)
System params	a, b, P, n 640 (bits)	— 0	p, q, g 2208
Public key	Q 161	n 1024	g^x 1024
Private key	d 160	d 1024	x 160

Software comparison (very rough)

Assumptions:

- 1 EC addition = 10 field multiplications.
- 40 160-bit field multiplications = 1 1024-bit modular multiplication.

	ECDSA	RSA $e=2^{16}+1$, CRT	DSA
Signing time	(kP) 60	$(m^d \bmod n)$ 384	$(g^k \bmod p)$ 240
Verifying time	(2 exps) 120	$(s^e \bmod n)$ 17	(2 exps) 480

(# of 1024-bit modular multiplications)

Hardware comparison

- James Dworkin, Motorola, April 1997.
- 20 MHz.
- RSA: Montgomery multiplication, CRT, 64-bit e, 16×16 bit multiplier.
- No other assumptions were stated.

	160-bit EC	1024-bit RSA	210-bit EC	2048-bit RSA
Signature speed (ms)	5.3	85.7	7.1	657.3
Verification speed (ms)	10.5	24.1	14.2	94.4
Silicon area (mil/side)	72	73	86	83
Energy to sign (mW/s)	.095	2.228	.214	22.611
Energy to verify (mW/s)	.190	.626	.427	3.249

9. Industry/Government standards

Goals:

- Facilitate widespread use of cryptographically sound and well-accepted techniques.
- Promote interoperability.

Draft standards

1. ANSI X9.62 (The Elliptic Curve Digital Signature Algorithm (ECDSA))
   - Goals: high security and interoperability.
   - Elliptic curves over $\mathbb{Z}_p$.
   - Elliptic curves over $\mathbb{F}_{2^m}$ (polynomial bases, optimal normal bases).
   - Security constraint: $n > 2^{160}$.
   - (Optional) method for generating random curves *verifiably* at random.
   - Ballot date: December 1997.

2. ANSI X9.63 (Elliptic Curve Key Agreement and Transport Protocols)
   - Key agreement: two Diffie-Hellman variants (MQV, unified model).
   - Key transport.
   - (Hoped) ballot date: Fall 1998.

3. IEEE P1363 (Standard Specification for Public-key Cryptography)
   - RSA, discrete logs, elliptic curves.
   - Programmers reference guide, rather than an interoperability standard.
   - Lots of options. (e.g. arbitrary polynomial or normal bases for $\mathbb{F}_{2^m}$).
   - No minimal security requirements.
   - Elliptic curve protocols: ECDSA, Nyberg-Rueppel signature scheme, MQV and unified-model key agreement.
   - http://stdsbbs.ieee.org/
   - (Hoped) ballot date: Summer 1998.

4. Internet OAKLEY (variant of Diffie-Hellman)
   - http://www.ietf.cnri.reston.va.us/
5. ISO 14888 (digital signatures with appendix)
   - High-level description of elliptic curve signature algorithms.
6. ATM Forum
   - Elliptic curve signature schemes.
   - Elliptic curves over $\mathbb{Z}_p$ and $\mathbb{F}_{2^m}$.
7. Widespread support for inclusion of elliptic curve algorithms in SET 2.0.

8. US government FIPS
   - May 13 1997 Federal Registry announcement: NIST seeks comments on the possibility of allowing government agencies to use additional public-key based digital signature algorithms, such as the RSA and elliptic curve techniques.
   - Plans to develop a federal standard for public-key based cryptographic key agreement and exchange. The notice asked for comments on such techniques as RSA, Diffie-Hellman and elliptic curve.

10. Conclusions

- Elliptic curve cryptosystems are the next generation of public-key technology.
- They have been accepted by many as a mature technology.
- ECC will see widespread deployment in the coming years.