Wireless communication Security
無線通訊安全

Lecture II-3

May 14, 2009

洪國寶

Outline

Part II:

(d) 橢圓曲線密碼技術
- 基本代數概念
- 橢圓曲線簡介
- 基本橢圓曲線密碼協定
- 橢圓曲線之其他性質與應用

(e) 無線感測網路安全
- 無線感測網路簡介
- 無線感測網路的安全議題
  - Key distribution/management
  - Secure routing

(f) 相關論文討論

Review of Lecture 2

- Galois Fields $GF(p^n)$
- Moe algebraic structures
  - Field extension, Algebraic number fields, Algebraic closure
- Elliptic curve
  - We usually need to specify that (why?)
    - The characteristic is not 2 or 3, and
    - $4a^3 + 27b^2 \neq 0$
    - Point at infinity O (or ∞)
    - If the characteristic of K is 2, than the elliptic curves have different forms.
  - What are j-invarant, n-torsion point, Weil pairing, and supersingular curves etc?

Torsion points

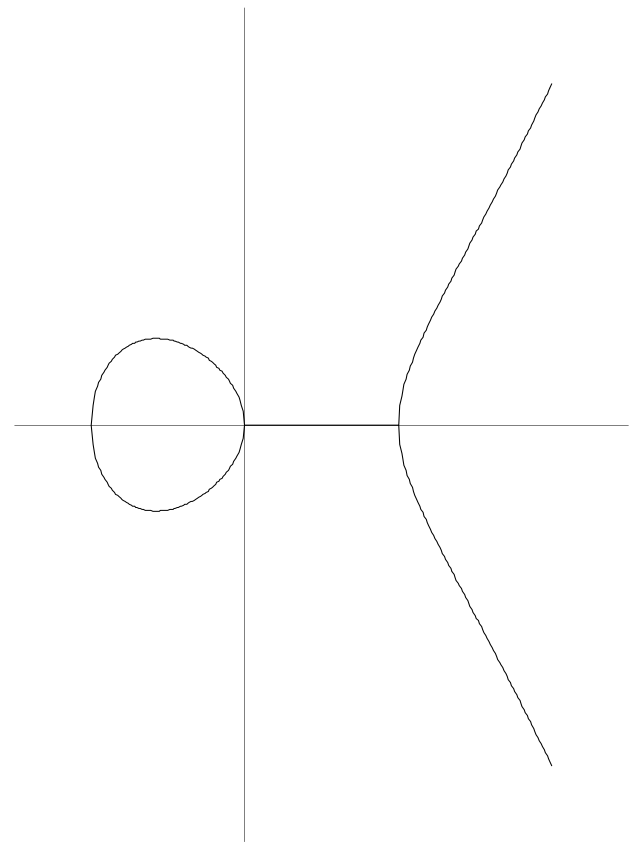
Supersingular curves

- An elliptic curve in characteristic p is called **supersingular** if $E[p] = \{O\}$.
- In other words, there are no points of order p, even with coordinates in an algebraically closed field.
- An **attractive feature** of supersingular curves is that computations involving an integer times a point can sometimes be done faster than might be expected. ■

Examples of Elliptic Curves

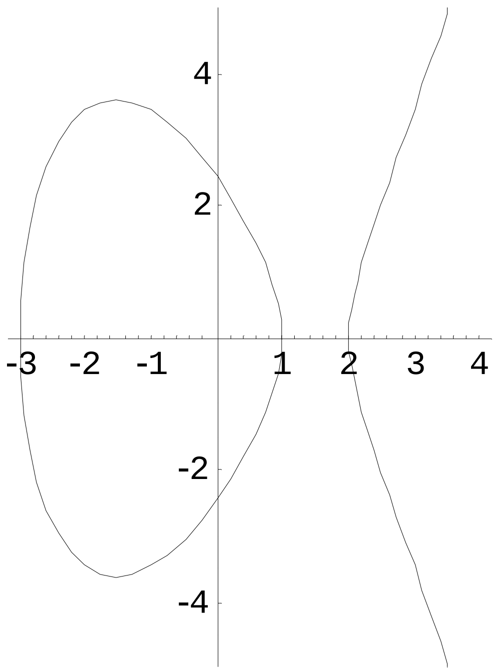
- Over the reals, the solutions form a curve with one or two components
- Example:

$$y^2 = x^3 - x$$

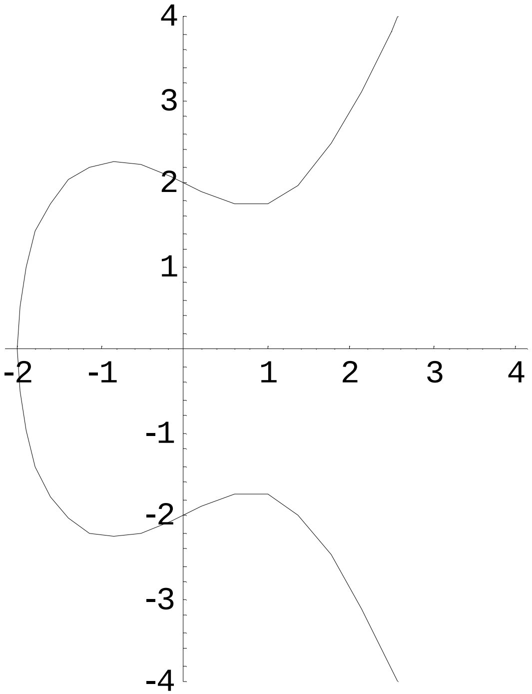


Examples of Elliptic Curves

- $y^2 = x^3 - 7x + 6$



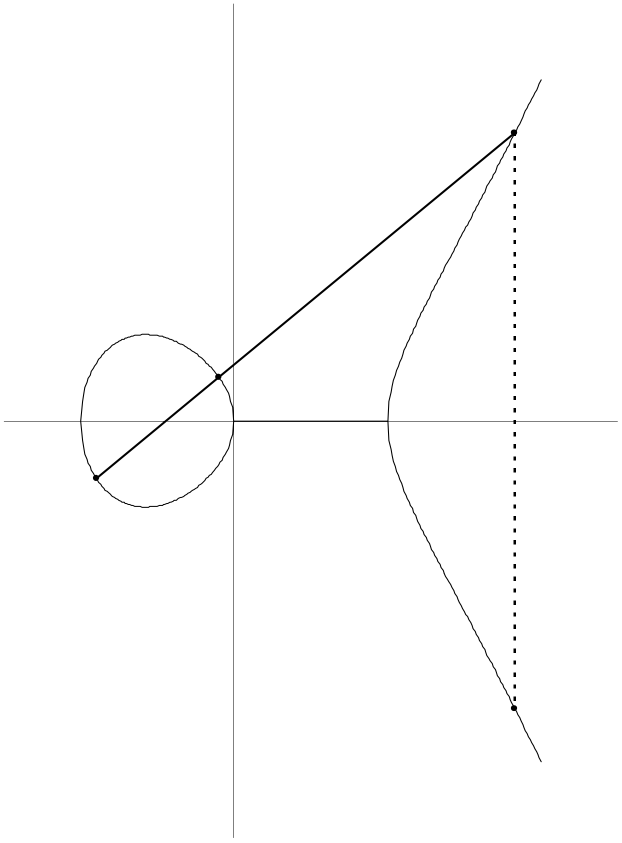
- $y^2 = x^3 - 2x + 4$



The graph of a non-singular curve has *two* components if its discriminant is positive, and *one* component if it is negative.

Elliptic Curve Arithmetic

- A group law may be defined where the sum of two points is the reflection across the *x*-axis of the third point on the same line
- "**Chords and tangents**"



Properties of "Addition" on E

Theorem: ***The addition law on E has the following properties*:**

a) $P + O = O + P = P$ for all $P \in E$.

b) $P + (-P) = O$ for all $P \in E$.

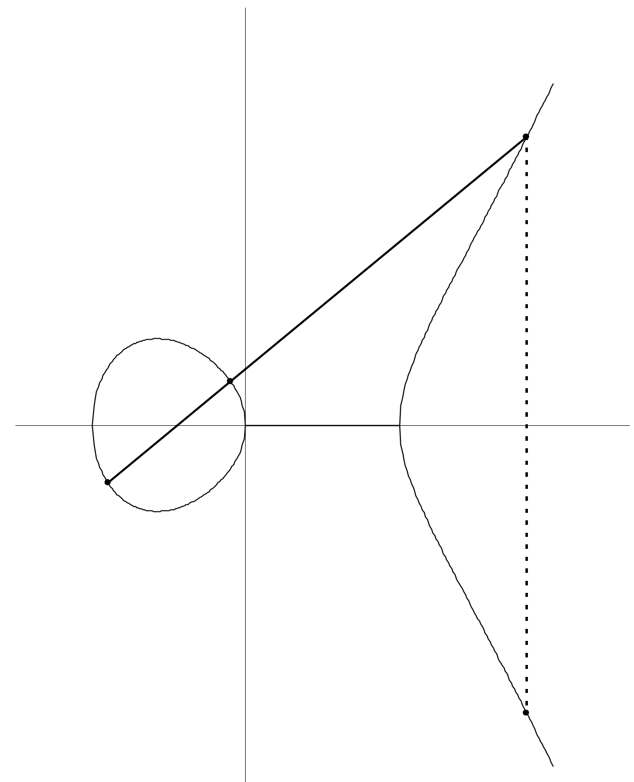
c) $(P + Q) + R = P + (Q + R)$ for all $P,Q,R \in E$.

d) $P + Q = Q + P$ for all $P,Q \in E$.

All of the group properties are trivial to check except for the **associative law** (c). The associative law can be verified by a lengthy computation using explicit formulas, or by using more advanced algebraic or analytic methods.

Group Law Axioms (recap)

- Closure
- Identity:
  $P + \boldsymbol{O} = \boldsymbol{O} + P = P$
- Inverse:
  $(x, y) + (x, -y) = \boldsymbol{O}$
- Associativity
- Commutativity



In other words, the addition law + makes the points of E into a **abelian group**.

Addition Formulae

- Now we can show the formulas for adding points.
  - Assume $P = (x_1, y_1)$ and $Q = (x_2, y_2)$
- If the characteristic of K is > 3 than
  - $-P = (x_1, -y_1)$
  - $P + Q = (\lambda^2 - x_1 - x_2, \lambda(x_1 - x_3) - y_1)$
    - $\lambda = (y_2 - y_1)/(x_2 - x_1)$, if $P \neq Q$

      $= (3 x_1^2 + a)/2y_1$, if $P = Q$

$\lambda =$

Addition Formulae

- If the characteristic of K is 2, than
  - If $j(E) \neq 0$:
    - $-P = (x_1, y_1 + x_1)$
    - $P+Q = (x_3, y_3)$

$$x_3 = ((y_1+y_2)/(x_1+x_2))^2 + (y_1+y_2)/(x_1+x_2) + x_1+x_2 + a, \ P \neq Q$$
$$= x_1^2 + b/ x_1^2, \ P = Q$$

$$y_3 = ((y_1+y_2)/(x_1+x_2))(x_1+x_3) + x_3 + y_1, \ P \neq Q$$
$$= x_1^2 + (x_1 + y_1/x_1)x_3 + x_3, \ P = Q$$

Addition Formulae

- If the characteristic of K is 2, than
  - If j(E) = 0:
    - $-P = (x_1, y_1 + c)$
    - $P+Q = (x_3, y_3)$

    $x_3 = ((y_1+y_2)/(x_1+x_2))^2 + x_1+x_2$, $P \neq Q$

    $= (x_1^4 + a^2)/ c^2$, $P = Q$

    $y_3 = ((y_1+y_2)/(x_1+x_2))(x_1+x_3) + c + y_1$, $P \neq Q$

    $= ((x_1^2 + a)/c)(x_1+x_3) + c + y_1$, $P = Q$

Elliptic Curves over Finite Fields

- An elliptic curve may be defined over any finite field **GF(*q*)** (char. of GF(q) > 3)

$$y^2 = x^3 + ax + b$$

- For **GF(2^m)**, the curve has a different form:

$$y^2 + xy = x^3 + ax^2 + b$$

  where $b \neq 0$

- Addition formulae are similar to those over **R**.
  - ■

Example

- $E : Y^2 = X^3 - 5X + 8$ modulo 37

{ O, (1,2), (1,35), (5,16), (5, 21), (6,3), (6,34), (8,6), (8,31), (9,10), (9, 27), (10,12), (10,25), (11,10), (11, 27), (12,14), (12,23), (16,18), (16,19), (17, 10), (17,27), (19,1), (19,36), (20,8), (20, 29), (21,5), (21,32), (22,1), (22,36), (26, 8), (26,29), (28,8), (28,29), (30,12), (30, 25), (31,9), (31,28), (33,1), (33,36), (34, 12), (34,25), (35,11), (35,26), (36,7), (36, 30) }

- Let $P_1 = (6,3)$ and $P_2 = (9,10)$. Then $P_1 + P_2 = (11,10)$.

(see next slide for more details)

Example

Let $\lambda = \frac{y_2 - y_1}{x_2 - x_1}$ if $P_1 \neq P_2$ and $\lambda = \frac{3x_1^2 + a}{2y_1}$ if $P_1 = P_2$.

Then $P_1 + P_2 = (\lambda^2 - x_1 - x_2, -\lambda^3 + 2\lambda x_1 + \lambda x_2 - y_1)$.

- $P_1 = (6, 3)$, $P_2 = (9, 10)$
- **λ**= (10-3)/(9-6) = 7/3 = 7×25 = 27 **mod 37**
- $(27^2 - 6 - 9, -27^3 + 324 + 243 - 3) =$
  (714, -19119) =
  (11, 10) **mod 37**

Elliptic Curves over Finite Fields

- Let #E(F_q) denote the number of points on an elliptic curve $E(F_q)$, including ***O***
- Hasse bound: #E(F_q) = q+1-t, where

$$|t| \leq 2\sqrt{q}$$

- The group of points is either cyclic or a product of two cyclic groups ■

Example

- $y^2 = x^3 + 1$ / GF(5)

z	0	1	2	3	4
z^2	0	1	4	4	1

x	1	2	3	4	5
y	± 1	?	± 2	?	0

$E(\mathbf{F}_5) = \{\infty, (0, \pm 1), (2, \pm 2), (4, 0)\}$. Hence $\# E(\mathbf{F}_5) = 6$.

Is E($\mathbf{F}_5$) cyclic?

Anomalous curves

- An elliptic curve is called **anomalous** if #E[**F**q] = q.
- The discrete log problem for the group E(**F**q) can be solved quickly.
- An **attractive feature** of anomalous curves is that they permit a speed-up in certain calculation in E($\overline{\mathbf{F}}$q). ■

Scalar Multiplication

- *Scalar multiplication* is repeated group addition:

$$cP = P + \cdots + P \quad (c \text{ times})$$

where c is an integer

- For all $P \in E(F_q)$,

$$nP = \boldsymbol{O}$$

where $n = \#E(F_q)$

Analogy with Multiplicative Groups

Elliptic Curve Group	Multiplicative Group
point addition	multiplication
scalar multiplication	exponentiation
elliptic curve discrete logarithm	discrete logarithm

Outline

Part II:

(d) 橢圓曲線密碼技術

– 基本代數概念
– 橢圓曲線簡介
– 基本橢圓曲線密碼協定
– 橢圓曲線之其他性質與應用

(e) 無線感測網路安全

– 無線感測網路簡介
– 無線感測網路的安全議題
  - Key distribution/management
  - Secure routing

(f) 相關論文討論

Elliptic Curve Cryptography

- ECDLP (EC discrete logarithm problem)
- Related issues
  - Restrictions, Domain Parameters, Selecting curves
- Elliptic Curve Cryptographic Schemes
  - ECDH
  - ECMQV
  - ECIES
  - ECDSA
- ECC Advantages and Disadvantages
- Standardization Efforts

EC Discrete Logarithm Problem

- **Problem:** Given two points *W*, *G*, find *s* such that $W = sG$
  - first suggested by Miller 1985, Koblitz 1987
- With appropriate cryptographic restrictions, this is believed to take ***exponential*** *time*
  - $O(\sqrt{r})$ time, where *r* is the order of *W*
- There is a way to reduce the log problem over elliptic curve to the log problem over F_{q^k}
  - The reduction only works for some special curves that are called **supersingular**
  - Why do you care about this? ■

EC Discrete Logarithm Problem

- By comparison, factoring and ordinary discrete logarithms can be solved in ***subexponential*** time
- ECC thus offers much shorter key sizes than other public-key cryptosystems ■

Elliptic Curve Cryptography

- ECDLP (EC discrete logarithm problem)
- **Related issues**
  - Restrictions, Domain Parameters, Selecting curves
- Elliptic Curve Cryptographic Schemes
  - ECDH
  - ECMQV
  - ECIES
  - ECDSA
- ECC Advantages and Disadvantages
- Standardization Efforts ■

Typical Cryptographic Restrictions

- $\#E(F_q) = kr$ for large prime r
  - k is *cofactor*
- GCD $(k, r) = 1$
- "**Anomalous**" condition: $r \neq q$
- **MOV** condition: r does not divide q^i-1 for small i ■

Domain Parameters

- **Common values** shared by a group of users from which key pairs may be generated
- User or trusted party may *generate* domain parameters
- Anyone may *validate* domain parameters ■

EC Domain Parameters

- Finite field F_q
- Elliptic curve $E(F_q)$ with cryptographic restrictions
- Prime divisor r of $\#E(F_q)$
- Cofactor k (usually 1,2, or 4)
- Base point $G \in E(F_q)$ of order r $\blacksquare$

Generating EC Domain Parameters

1. Select a prime power q
2. Select an elliptic cuve E over F_q with cryptographic restrictions
   – order $\#E(F_q) = kr$
3. Generate a point G of order r
4. Output F_q, $E(F_q)$, r, k, G ■

Selecting an Elliptic Curve

- **Random method**
- Complex multiplication method
- Subfield method

- Methods provide tradeoff between speed, “structure” in curves
  - less structure = more conservative in assumptions about security ■

Random Method

1. Generate a random curve
2. Count the number of points $\#E(F_q)$
3. If restrictions not met, goto 1

- No structure, but step 2 may be slow
  - (Schoof 1985, etc.) ■

Generating a Point of Order r

1. Generate a point $H \in E(F_q)$
2. Compute $G = kH$
3. If $G = \boldsymbol{O}$, goto 1
4. Output G ■

Validating EC Domain Parameters

1. Check that q is a prime power
2. Check that E is an elliptic curve over F_q with cryptographic restrictions
   – order $\#E(F_q) = kr$, where r is prime
3. Check that G is a point on $E(F_q)$ of order r
4. Output *valid* if all checks pass, *invalid* otherwise ■

Key Pairs

- Pairs of public, private values with which users may perform cryptographic operations
- User or trusted third party may *generate* key pair
- Anyone may *validate* public key ■

EC Key Pairs

- Public key $W \in E(F_q)$
- Private key $s \in [1, r\text{-}1]$
  - where $W = sG$ ■

Generating an EC Key Pair

1. Randomly generate $s \in [1, n\text{-}1]$
2. Compute $W = sG$
3. Output (W, s) ■

Validating an EC Public Key

- Assume valid domain parameters

1. Check that W is a point on $E(F_q)$ of order r
2. Output *valid* if so, *invalid* otherwise ■

Elliptic Curve Cryptography

- ECDLP (EC discrete logarithm problem)
- Related issues
  - Restrictions, Domain Parameters, Selecting curves
- **Elliptic Curve Cryptographic Schemes**
  - ECDH
  - ECMQV
  - ECIES
  - ECDSA
- ECC Advantages and Disadvantages
- Standardization Efforts ■

Cryptographic Schemes

- Following general model from IEEE P1363, a ***scheme*** is a set of related operations providing the building blocks for a *protocol*
  - Examples:
    - Key agreement
    - Signature with appendix
    - Encryption
- A (cryptographic) scheme consists of an unambiguous specification of a set of transformations that are capable of providing a (cryptographic) service when properly implemented and maintained. (NIST)
  - **A scheme is a higher level construct than a primitive and a lower level construct than a protocol.** ■

Scheme Operations

- Depending on the scheme, related operations may include:
  - domain parameter generation, validation
  - key pair generation, public-key validation
  - one or more scheme-specific operations ■

Key Agreement Scheme

- *Key agreement operation* derives a shared secret key from a private key, another's public key, and key derivation parameters
- Multiple secret keys can be obtained by varying parameters ■

Elliptic Curve Diffie-Hellman

- Key agreement scheme based on Diffie-Hellman protocol
- Underlying function:
  - KDF: key derivation function ■

ECDH Key Agreement

- *Input:* private key s, other's public key W^*, key derivation parameters P
- *Output:* shared secret key K

1. Compute $Z = sW^*$
2. Compute $K =$ **KDF** (Z, P)
3. Output K ■

Key Agreement Modes

- Each key pair may be ephemeral, authenticated, or a combination, depending on security goals
- Examples of protocol modes:
  - anonymous
  - static-static
  - signed ephemeral-ephemeral
  - ephemeral-static ■

Elliptic Curve Diffie-Hellman Key Exchange

Public Knowledge: A group $E(\mathbb{F}_p)$ and a point P of order n.

BOB		ALICE
Choose secret $0 < b < n$		Choose secret $0 < a < n$
Compute $Q_{Bob} = bP$		Compute $Q_{Alice} = aP$
Send Q_{Bob}	→	to Alice
to Bob	←	Send Q_{Alice}
Compute bQ_{Alice}		Compute aQ_{Bob}

Bob and Alice have the shared value $bQ_{Alice} = abP = aQ_{Bob}$

Presumably(?) recovering abP from aP and bP requires solving the elliptic curve discrete logarithm problem.

ECMQV

- MQV is short for Menezes-Qu-Vanstone, the names of the authors of this protocol.
- MQV offers attributes—such as key-compromise impersonation resilience and unknown key-share resilience—that are not found with DH.
  - This allows protocols that use MQV for key agreement to offer stronger authentication and ensure malicious entities cannot masquerade as a third party to the entity whose key was compromised.
- MQV also has many desirable performance attributes, including
  - the dominant computational steps are not intensive
  - has low communication overhead,
  - is role-symmetric, non-interactive and
  - does not use encryption or time-stamping. ■

Encryption Scheme

- *Encryption operation* computes a ciphertext from a message with a public key
- *Decryption operation* recovers a message from a ciphertext with a private key
- A*ugmented* encryption scheme also binds *control information* to message ■

Elliptic Curve Integrated Encryption Scheme (ECIES) - Encryption

- **Input:** Public key (static) W in E, message M.
- **Output:** Ciphertext (R,S,A).
- **Actions:**

1. Set R = rG for random r in [1,n-1].
2. Set (u,a) = KDF(x(rW)).
3. Set S = Encrypt(u,M) and A = MAC(a,S).

- **Note:** (R,r) **ephemeral** public-private key pair. ■

ECIES - Decryption

- **Input:** Private key s, ciphertext (R,S,A).
- **Output:** Invalid; or valid and message M.
- **Actions:**

1. Set (u,a) = KDF(x(sR)).
2. Valid if A=MAC (a,S) else invalid.
3. If valid, set M = Decrypt(u,S). ■

Signature Scheme

- *Signature generation operation* computes a signature on a message with a private key
- *Signature verification operation* verifies a signature with a public key ■

Elliptic Curve Digital Signature Algorithm

- Signature scheme based on NIST FIPS 186-1 DSA
- Underlying function
  - Hash: collision-resistant hash function ■

ECDSA Signature Generation

- *Input:* private key s, message M
- *Output:* signature (c,d)

1. Compute f = Hash (M)
2. Generate a one-time key pair (u, V)
3. Compute c = int (x_V) mod r
4. Compute $d = u^{-1}(f + sc) \bmod r$
5. If $c = 0$ or $d = 0$, goto 2
6. Output (c,d) ■

ECDSA Signature Verification

- *Input:* signer's public key W, message M, signature (c,d)
- *Output: valid* or *invalid*

1. Compute f = Hash (M)
2. Check that $1 \leq c,d \leq r\text{-}1$
3. Compute $h = d^{-1} \bmod r$
4. Compute $P = fhG + chW$
5. Check that $P \neq \boldsymbol{O}$
6. Check that $c = \text{int}(x_P) \bmod r$
7. If all checks pass, output *valid*, otherwise output *invalid* ■

Some Observations

- In these schemes, only one or two steps are EC operations, some are modular arithmetic, the rest are Hash, KDF, Encrypt, MAC
  - the additional operations help provide provable security
- Schemes are readily adapated to multiplicative groups ■

Elliptic Curve Cryptography

- ECDLP (EC discrete logarithm problem)
- Related issues
  - Restrictions, Domain Parameters, Selecting curves
- Elliptic Curve Cryptographic Schemes
  - ECDH
  - ECMQV
  - ECIES
  - ECDSA
- **ECC Advantages and Disadvantages**
- Standardization Efforts ■

Key Size Comparison

- Today, three families of public-key techniques are prominent
- Following P1363, named according to the hard problem:
  - DL: (ordinary) discrete logarithms
  - EC: elliptic curve discrete logarithms
  - IF: integer factorization
- Each has its own advantages ■

Key Size Comparison

- Key size is length in bits of:
  - DL: field order q
    - also consider group order r
  - EC: group order r
  - IF: modulus n
- Key sizes can be compared based on running time for solving hard problem with current methods
  - other factors to consider ■

Comparable Key Sizes
(Based on Running Time)

EC	DL, IF	Symmetric
112	512	56
160	1024	80
224	2048	112

Advantages

- Alternative hard problem
- Speed
- Data size
- New types of schemes
- Many options ■

Alternative Hard Problem

- EC Discrete Logarithm Problem is very different than DL, IF hard problems
  - does not appear feasible to apply DL, IF approaches to solve it
- Thus, it is an effective alternative against advances in methods for other problems ■

Speed

- EC operations are generally faster than DL, IF counterparts at comparable key sizes
  – GF(2^m) arithmetic affords further speedups
- Key pair generation is much faster than for IF ■

Data Size

- EC data are shorter than DL, IF counterparts
- Intermediate values are shorter
- Keys are shorter
  - benefit depends on certificate content
- Signatures with appendix are same size as for DL, shorter than IF ■

New Types of Schemes

- EC family, like DL, has great flexibility due to the availability of common domain parameters
- Multiple schemes can be combined efficiently, e.g.:
  - signature + encryption
  - signature / key agreement + certification ■

Many Options

- EC family affords many choices:
  - field type, size, representation
  - curve formula
  - group order
  - base point
  - cryptographic scheme
- Appropriate choices can meet varying security and implementation objectives ■

Disadvantages

- Alternative hard problem
- Curve generation
- Many options ■

Alternative Hard Problem

- ECDLP has not been studied as long as DL, IF hard problems, and even a modest improvement in methods could have great impact
- However, the focus on this area has grown considerably over the past few years, with increased confidence ■

Curve Generation

- EC curve generation is complex, not readily implemented
- However, implementers can rely on third parties for curves, which can be validated
  - e.g., NIST curves ■

Many Options

- ECC affords many options, so interoperability is challenging:
  - no conversion between GF(2^m), GF(p)
  - hardware optimizations may be specific to one set of domain parameters
- However, much of this will be settled by standards and industry practice ■

Elliptic Curve Cryptography

- ECDLP (EC discrete logarithm problem)
- Related issues
  - Restrictions, Domain Parameters, Selecting curves
- Elliptic Curve Cryptographic Schemes
  - ECDH
  - ECMQV
  - ECIES
  - ECDSA
- ECC Advantages and Disadvantages
- **Standardization Efforts** ■

Standardization Efforts

- Elliptic curves are parts of standards being developed by several groups:
  - ANSI X9F1
  - IEEE P1363
  - ISO JTC1 SC27
  - SECG
  - U.S. NIST ■

U.S. NIST

- Information processing for U.S. government
- FIPS 186 (Digital Signature Standard) to add support for ANSI X9.62
- Eventual ANSI X9.63 support likely
- Reference elliptic curves published
- csrc.nist.gov/fips ■

NSA Suite B Cryptography

- Required cryptographic algorithms for all US non-classified and classified (SECRET and TOP-SECRET) needs
  - Except a small area of special-security needs (e.g. nuclear security) – guided by Suite A (definition is classified)
- **Encryption: AES**
  - FIPS 197 (with keys sizes of 128 and 256 bits)
- **Digital Signature: Elliptic Curve Digital Signature Algorithm**
  - FIPS 186-2 (using the curves with 256 and 384-bit prime moduli)
- **Key Exchange: Elliptic Curve Diffie-Hellman or ECMQV**
  - Draft NIST Special Publication 800-56 (using the curves with 256 and 384-bit prime moduli)
- **Hashing: Secure Hash Algorithm**
  - FIPS 180-2 (using SHA-256 and SHA-384) ■

NIST standards

- NIST has proposed a specific set of elliptic curves for cryptography purposes (**DRAFT FIPS PUB 186-3**)
- Elliptic curves are defined for prime fields GF(p) and binary

Bit Length of n	Prime Field	Binary Field
161 – 223	$\text{len}(p) = 192$	$m = 163$
224 – 255	$\text{len}(p) = 224$	$m = 233$
256 – 383	$\text{len}(p) = 256$	$m = 283$
384 – 511	$\text{len}(p) = 384$	$m = 409$
≥ 512	$\text{len}(p) = 521$	$m = 571$

Curve P-192 (a = -3)

p = 6277101735386680763835789423207666416083908700390324961279

n = 6277101735386680763835789423176059013767194773182842284081

b = `64210519 e59c80e7 0fa7e9ab 72243049 feb8deec c146b9b1`

G_x = `188da80e b03090f6 7cbf20eb 43a18800 f4ff0afd 82ff1012`

G_y = `07192b95 ffc8da78 631011ed 6b24cdd5 73f977a1 1e794811`

ECC recap

- ECC offers an attractive alternative to other public-key cryptosystems
  - new hard problem
  - smaller key size
- Many standards are emerging
- Number theory continues to be useful ■

Elliptic Curve Research Areas

- EC over finite fields has been an increasing focus of research

1. Efficient elliptic curve arithmetic, scalar multiplication
   - including finite field arithmetic
2. Efficient curve generation
3. Cryptographic properties ■

Some Interesting Applications

- Factoring (Lenstra 1985)
  - running time of Elliptic Curve Method (ECM) depends on size of prime factors of a number, ideal for "smooth" numbers
- Primality proving (Goldwasser-Kilian 1986)
  - under number-theory assumptions, method for *proving* primality in random polynomial time
- **Fermat's Last Theorem** ■