CSC 580
Cryptography and Computer Security

Discrete Logarithms, Diffie-Hellman, and Elliptic Curves
(Sections 2.8, 10.1-10.4)

March 20, 2018

Overview

Today:

- Discuss homework 6 solutions
- Math needed for discrete-log based cryptography
- Diffie-Hellman and ElGamal
- Elliptic Curves - idea and translation of Diffie-Hellman to ECC

Next:

- Quiz on Thursday (based on HW6 & formal models)
- Graded Homework 2 due on Thursday!
- Read Chapter 11 (skip SHA-512 logic and SHA3 iteration function)
- Project project due in two weeks (April 3) - don't forget this!

The Discrete Log Problem

For every prime number p, there exists a primitive root (or "generator") g such that

$$g^1, g^2, g^3, g^4, \ldots, g^{p-2}, g^{p-1} \quad \text{(all taken mod } p\text{)}$$

are all distinct values (so a permutation of 1, 2, 3, ..., p-1).

Example: 3 is a primitive root of 17, with powers:

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
3^i mod 17	3	9	10	13	5	15	11	16	14	8	7	4	12	2	6	1

$f_{g,p}(i) = g^i \bmod p$ is a bijective mapping on {1,.., p-1}

g and p are global public parameters

$f_{g,p}(i)$ is easy to compute (modular powering algorithm)

Inverse, written $\text{dlog}_{g,p}(x) = f_{g,p}^{-1}(x)$, is believed to be difficult to compute

Diffie-Hellman Key Exchange (DHE)

Assume g and p are known, public parameters

Alice

$a \leftarrow$ random value from $\{1, \ldots, p\text{-}1\}$
$A \leftarrow g^a \bmod p$

Bob

$b \leftarrow$ random value from $\{1, \ldots, p\text{-}1\}$
$B \leftarrow g^b \bmod p$

Send A to Bob →

← *Send B to Alice*

$S_a \leftarrow B^a \bmod p$

$S_b \leftarrow A^b \bmod p$

In the end, Alice's secret (S_a) is the same as Bob's secret (S_b):

$$S_a = B^a = g^{ba} = g^{ab} = A^b = S_b$$

Eavesdropper knows A and B, but to get a or b requires solving the discrete logarithm problem!

ElGamal Encryption

The idea is simple:

Define "long term key" for one side of Diffie-Hellman

Key Generation (Bob):

- $b \leftarrow$ random value from $\{1, \ldots, p\text{-}1\}$
- $B \leftarrow g^b \bmod p$
- (B, g, p) is public key (i.e., encryption key) - b is private key

For Alice to send a message to Bob:

- Get (B, g, p) from Bob
- Pick $k \leftarrow$ random value from $\{1, \ldots, p\text{-}1\}$
- For message $M \in \{1, \ldots, p\text{-}1\}$, ciphertext $(C_1, C_2) = (g^k \bmod p,\ M \cdot B^k \bmod p)$

For Bob to decrypt ciphertext (C_1, C_2):

- $K \leftarrow C_1^{\,b} \bmod p$ // Same as B^k above
- $M \leftarrow C_2 \cdot K^{-1} \bmod p$ // Same as original plaintext (see DHE for similarity)

ElGamal Encryption

The idea is simple:

<u>Big Warning!!!!</u>

In ElGamal, only one side can be a long-term key!!!

Serious problems if sender re-uses k!

- Pick $k \leftarrow$ random value from $\{1, \ldots, p\text{-}1\}$
- For message $M \in \{1, \ldots, p\text{-}1\}$, ciphertext $(C_1,C_2) = (g^k \bmod p,\ M \cdot B^k \bmod p)$

For Bob to decrypt ciphertext (C_1,C_2):

- $K \leftarrow C_1^{\,b} \bmod p$ // Same as B^k above
- $M \leftarrow C_2 \cdot K^{-1} \bmod p$ // Same as original plaintext (see DHE for similarity)

Abstracting the Problem

There are many sets over which we can define powering.

Example: Can look at powers of $n \times n$ matrices (A^2, A^3, etc.)

Any finite set S with an element g such that $f_g: S \to S$ is a bijection (where $f_g(x) = g^x$ for all $x \in S$) is called a *cyclic group*

- Very cool math here - see Chapter 5 for more info (optional)

If f_g is easy to compute and f_g^{-1} is difficult, then can do Diffie-Hellman

"Elliptic Curves" are a mathematical object with this property

In fact: f_g^{-1} seems to be harder to compute for Elliptic Curves than $\boldsymbol{Z}_p$

- Consequence: Elliptic Curves can use shorter numbers/keys than standard Diffie-Hellman - so faster and less communication required!

Elliptic Curves

The basic idea...

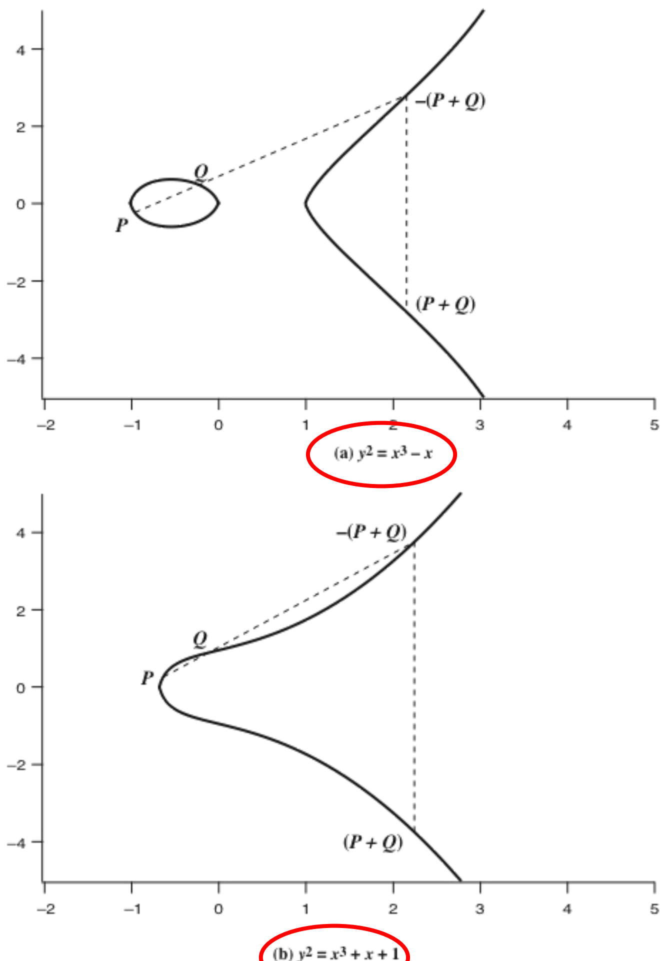

Figure 10.4 Example of Elliptic Curves

Key ideas:

- Formula with x and y defines a set of points (x,y).
- Formula is quadratic in y, cubic in x
- Since quadratic in, symmetric around x axis

Define “addition of two points”:

- Draw a line through the two points
- Where else does it hit curve
    - 3 places because cubic in x
- Reflect around x axis

Elliptic Curves over Finite Fields

General formula for "Elliptic Curves over Z_p" (p is prime):

$E_p(a,b)$ is the set of points (x,y) satisfying $y^2 \equiv x^3+ax+b \pmod p$

Technical requirement for a and b: $4a^3 + 27b^2 \not\equiv 0 \pmod p$

Squares in Z_5

$0^2 = 0$
$1^2 = 1$
$2^2 = 4$
$3^2 = 4$
$4^2 = 1$

Points in $E_5(2,1)$ ($y^2 \equiv x^3+2x+1 \pmod 5$)

x = 0: $y^2 = x^3+2x+1 \bmod 5 = 1$
y = 1 or 4 (see table on left)

x = 1: $y^2 = x^3+2x+1 \bmod 5 = 1+2+1 = 4$
y = 2 or 3

x = 2: $y^2 = x^3+2x+1 \bmod 5 = 8+4+1 = 3$ (no sol'n)

x = 3: $y^2 = x^3+2x+1 \bmod 5 = 27+6+1 = 4$
y = 2 or 3

x = 4: $y^2 = x^3+2x+1 \bmod 5 = 64+8+1 = 3$ (no sol'n)

Points

O
(0,1)
(0,4)
(1,2)
(1,3)
(3,2)
(3,3)

Elliptic Curves over Finite Fields

General formula for "Elliptic Curves over Z_p" (p is prime):

$E_p(a,b)$ is the set of points (x,y) satisfying $y^2 \equiv x^3+ax+b \pmod{p}$

Technical requirement for a and b: $4a^3 + 27b^2 \not\equiv 0 \pmod{p}$

Important points

- Can add points as before (no sensible picture, however)
- For a point P, can calculate
  - 2*P = P+P
  - 3*P = P+P+P
  - 4*P = P+P+P+P
  - ...

  (eventually repeats → P generates a cyclic group)
- Notation is multiplying rather than powering, but can do Diffie-Hellman!

Important: Discrete logs seem to be harder to compute for Elliptic Curves than $\boldsymbol{Z}_p$

- Consequence: Elliptic Curves can use shorter numbers/keys than standard Diffie-Hellman - so faster and less communication required!

Revisiting Key Sizes
From NIST publication 800-57a

Issue: PK algorithms based on mathematical relationships, and can be broken with algorithms that are faster than brute force.

We spent time getting a feel for how big symmetric cipher keys needed to be
➔ How big do keys in a public key system need to be?

From NIST pub 800-57a:

Table 2: Comparable strengths

Security Strength	Symmetric key algorithms	FFC (e.g., DSA, D-H)	IFC (e.g., RSA)	ECC (e.g., ECDSA)
≤ 80	2TDEA[21]	$L = 1024$ $N = 160$	$k = 1024$	$f = 160\text{-}223$
112	3TDEA	$L = 2048$ $N = 224$	$k = 2048$	$f = 224\text{-}255$
128	AES-128	$L = 3072$ $N = 256$	$k = 3072$	$f = 256\text{-}383$
192	AES-192	$L = 7680$ $N = 384$	$k = 7680$	$f = 384\text{-}511$
256	AES-256	$L = 15360$ $N = 512$	$k = 15360$	$f = 512+$