TOPIC: 6.857 – Computer & Network Security

DATE: 4/9/14

FILE UNDER:

PAGE: L17.1

Admin:

Pset #4: note elliptic curve bug in lecture notes for doubling a point

Quiz: Wed 4/16 in-class, open notes (no txts/phones/laptops)
Send in your questions!

Projects: Meet with TA's this week.

Talk schedule up real soon. (4/28, 4/30, 5/5, 5/7)

Today:

- ☐ CDH, DDH, gap groups
- ☐ Bilinear maps ($\Rightarrow$ DDH easy)
- ☐ Digital signatures (160 bits)
- ☐ IBE (identity-based encryption)
- ☐ 3-way key agreement

TOPIC

FILE UNDER

DATE

"Gap group" is one in which

- DDH is easy ("Decision Diffie Hellman")

  [Recall: given (g, g^a, g^b, g^c), to

  decide if $ab = c \pmod{\text{order}(g)}$

  ]

but

- CDH is hard ("Computational Diffie Hellman")

  [Recall: given (g, g^a, g^b), to

  compute g^{ab}

(Note that CDH easy $\Rightarrow$ DDH easy)

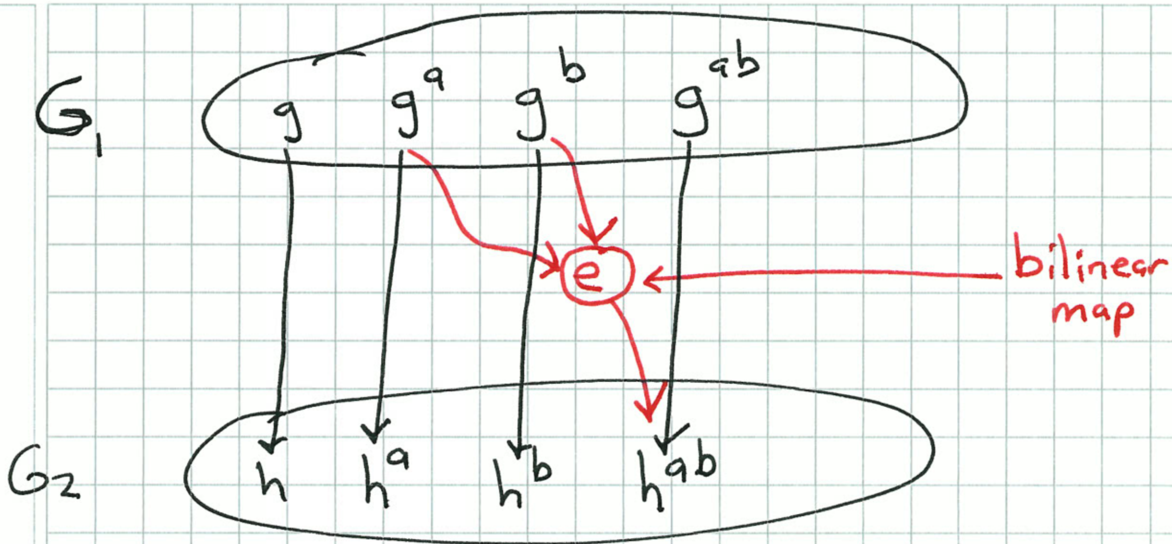
This difference in difficulty between DDH ("easy") and CDH ("hard") forms a "gap".

— How can one construct a "gap group"?

— What good would that be?

TOPIC: DATE:

FILE UNDER: PAGE:



"shadow group"

$|G_1| = |G_2| = q$ (prime)

g generates G_1

h generates G_2

CDH hard in G_1 & in G_2

DDH easy in G_1 (using e)

$$\boxed{e(g^a, g) = h^a}$$

computes "shadows"

TOPIC:

DATE:

FILE UNDER:

PAGE L19.3

Bilinear maps

"shadow group"

Suppose: G_1 is group of prime order q, with generator g

G_2 is group of prime order q, with generator h

see Figure (next page)

[we use multiplicative notation for both groups]

and there exists a (bilinear) map

$$e: G_1 \times G_1 \longrightarrow G_2$$

such that

$e(g,g)=h$

$$\boxed{(\forall a,b)\ e(g^a,g^b) = h^{ab}} \quad !!!$$

$$\begin{aligned}
&= e(g,g^{ab}) \\
&= e(g,g)^{ab} \\
&= e(g,g^b)^a \\
&= e(g,g^a)^b \\
&= e(g^b,g^a) \\
&\cdots
\end{aligned}$$

Bilinear maps also called "pairing functions"

They have an enormous number of applications.*

We are, of course, interested in efficiently computable bilinear maps.

* google: "The pairing-based crypto lounge"

Theorem:

If there is a bilinear map

$$e: G_1 \times G_1 \to G_2$$

between two groups of prime order q,

then DDH is easy in G_1

Proof:

Given (g, g^a, g^b, g^c) (elements of G_1)

then

$$c = ab \pmod q \iff \underbrace{e(g^a, g^b) = e(g, g^c)}_{\equiv\ \underbrace{h^{ab} = h^c}_{\equiv\ ab = c \pmod q}}$$

So: accept (g, g^a, g^b, g^c) iff $e(g^a, g^b) = e(g, g^c)$.

$\blacksquare$

Even though DDH is easy in G_1, CDH may still be hard; we may have a "gap group".

TOPIC

FILE UNDER:

DATE

How to construct gap groups (with bilinear maps):

- This is not simple! We give just a sketch.
- G_1 will be "supersingular" elliptic curve

  e.g. elliptic curve defined by points on

  $$y^2 = x^3 + ax + b \pmod{p}$$

  where $p = 2 \pmod{3}$, $p \geq 5$

  $a = 0$

  $b \in \mathbb{Z}_p^*$ (can choose $b = 1$)
- G_2 is finite field $\mathbb{F}_{p^k}$ for some small k

  (can use subgroups of G_1 & G_2 by choosing generators of order $\approx 2^{160}$ say...)

- e (bilinear map) is implemented as a "Weil pairing" or a "Tate pairing".

TOPIC:

DATE:

FILE UNDER:

Application 1: Digital signatures

(Boneh, Lynn, Shacham (2001))

Signatures are <u>short</u> (e.g. 160 bits)!

Public: groups G_1, G_2 of prime order q

pairing function $e: G_1 \times G_1 \to G_2$

g = generator of G_1

H = hash fn (c.R.) from messages to G_1

Note use of multiplicative notation here.

Secret key: x where $0 < x < q$

Public key: $y = g^x$ (in G_1)

To sign message M:

Let $m = H(M)$ (in G_1)

Note: Signature may be <u>short</u>! Just one element of G_1.

To represent a point on an elliptic curve, really just need to give x, and then one bit more to say which y is wanted (there are 2 square roots)

Output $\sigma = \sigma_x(M) = m^x$ (in G_1)

To verify (y, M, σ):

Check $e(g, \sigma) \stackrel{?}{=} e(y, m)$ where $m = H(M)$

$e(g, m)^x$ in both cases

Theorem: BLS signature scheme secure against existential forgery under chosen message attack in ROM assuming CDH is hard in G_1.

Application 3: Identity-based encryption (IBE) [Boneh, Franklin '01]

TTP (trusted third party) publishes

G_1, G_2, e (bilinear map), g (generator of G_1), y

where $y = g^s$ & s is TTP's master secret.

Let H_1 be random oracle mapping names (e.g. "alice@mit.edu") to elements of G_1

Let H_2 be random oracle mapping G_2 to $\{0,1\}^*$ (PRG).

Want to enable anyone to encrypt message for Alice knowing only TTP public parameters & Alice's name

Encrypt(y, name, M):

$r \xleftarrow{R} Z_q^*$ (here prime $q = |G_1| = |G_2|$)

$g_A = e(Q_A, y)$ where $Q_A = H_1(\text{name})$

output $(g^r, M \oplus H_2(g_A^r))$

TOPIC:

DATE:

FILE UNDER:

Decrypt ciphertext $c = (u, v)$:

- Alice obtains $d_A = Q_A^s$ from TTP (once is enough) where $Q_A = H_1(\text{name})$.

  This is Alice's decryption key.

  Note that TTP also knows it!

  Note that message may be encrypted before Alice gets d_A.

- Compute $v \oplus H_2(e(d_A, u))$

$$= v \oplus H_2(e(Q_A^s, g^r))$$
$$= v \oplus H_2(e(Q_A, g)^{rs})$$
$$= v \oplus H_2(e(Q_A, g^s)^r)$$
$$= v \oplus H_2(e(Q_A, y)^r)$$
$$= v \oplus H_2(g_A^r)$$
$$= M$$

TOPIC:

DATE:

FILE UNDER:

Application 2: Three-way key agreement (Joux, generalizing DH)

Recall DH: $A \to B : g^a$
$B \to A : g^b$
$\text{key} = g^{ab}$

Joux: Suppose G_1 has generator g
Suppose $e : G_1 \times G_2$ is a bilinear map.

$A \to B, C : g^a$

$B \to A, C : g^b$

$C \to A, B : g^c$

A computes $e(g^b, g^c)^a = e(g,g)^{abc}$

B computes $e(g^a, g^c)^b = e(g,g)^{abc}$ =

C computes $e(g^a, g^b)^c = e(g,g)^{abc}$

$\text{key} = e(g,g)^{abc}$

Secure assuming "BDH" $\equiv$

given g, g^a, g^b, g^c, e

hard to compute $e(g,g)^{abc}$

Four-way key agreement is open problem!

(maybe... see Garg/Gentry/Halevi Proc. Eurocrypt '13)

multilinear maps!

TOPIC:

DATE:

FILE UNDER:

PAGE:

note use of additive notation

ID-based signature (Hess 2002; Dutta survey §4.10)

master secret $= s$
master public $= P_{pub} = sP$ (P generates G_1)

$H_1 : \{0,1\}^* \to G_1$

$H : \{0,1\}^* \times G_2 \to \mathbb{Z}_q^*$

Extract: user gives ID. Public id $= H_1(ID) = Q_{ID}$
secret key $= s \cdot Q_{ID} = S_{ID}$

Sign (S_{ID}, m): $P_1 \in_R G_1^*$

$k \in_R \mathbb{Z}_q^*$

$r = e(P_1, P)^k$

$\left.\begin{array}{l} v = H(m,r) \\ u = v S_{ID} + k P_1 \end{array}\right\} = \text{signature}$

Verify: $(Q_{ID}, m, (u,v))$:

$r = e(u,P) \cdot e(Q_{ID}, -P_{pub})^v$

accept iff $v = H(m,r)$

Secure against existential forgery in ROM under adaptive chosen message attack assuming weak-DH problem is hard.

given (P, Q, sP) for $P, Q \in G_1$

output sQ