4/9/18
Lec 16

Today:

- Gap groups & bilinear maps
- BLS (Boneh-Lynn-Shacham) signatures.
- 3-way key agreement (Joux)
- ~~Identity~~ Identity-based encryption

Gap groups

A gap group is a group where

- DDH is easy: (Decisional Diffie-Hellman)

  i.e. $(g, g^a, g^b, g^{ab}) \not\approx (g, g^a, g^b, g^{ab})$

- CDH is hard (Computational Diffie-Hellman)

  i.e. $g, g^a, g^b \xrightarrow{\text{hard}} g^{ab}$

Note: CDH is easy $\Rightarrow$ DDH is easy

The difference between DDH being easy & CDH being hard forms a gap

Q1: Why do we want a "gap group"?

Q2: How can we construct a gap group?

Bilinear maps

Suppose: G_1 group of prime order q with generator g

G_2 " " " " " " " h

[We use multiplicative notation for both groups]

& there exists a bilinear map

$e : G_1 \times G_1 \longrightarrow G_2$ s.t.

$$\forall a, b \in \mathbb{Z}_q \quad e(g^a, g^b) = e(g, g)^{a \cdot b} \quad \left(\begin{array}{l} = e(g^{a \cdot b}, g) = \ldots \\ = e(g^b, g^a) = \ldots \end{array} \right)$$

$$e(g, g) = h$$

non-degenerate.

Bilinear maps are also called pairing functions.

They have numerous applications!

Thm: If there exists a bilinear map

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

then DDH is easy in G_1

Proof: Given (g, g^a, g^b, g^c)

check if $e(g^a, g^b) = e(g, g^c)$

If so output "$c = a \cdot b$".

& o.w. output "c is random" ∎

Note: Even though DDH is easy in G_1,

CDH may still be hard.

I.e. we may still have a gap group.

How to construct a gap group w. bilinear map?

This is not simple!

G_1 is an elliptic curve (w. certain properties)

e (the bilinear map) is a "Weil pairing" or a "Tate pairing".

1993 : Used to try to break eliptic curve crypto.

2000 : First "good" use

[Joux] : 3-way key agreement

(extention of Diffie-Hellman 2-way key agreement).

2001 : [Boneh-Lynn-Shacam] : short signatures

2001 : [Boneh-Fanklin] : Identity based encryption.

Application 1 : 3-way key agreement

Recall DH :

$$A \qquad\qquad B$$

$$\xrightarrow{\ g^a\ }$$

$$\xleftarrow{\ g^b\ }$$

key g^{ab}

3-way : Let G_1, G_2 be prime order groups w. bilinear map $e : G_1 \times G_1 \longrightarrow G_2$ & let g be generator of G_1

$A \longrightarrow BC$:	g^a	A computes	$e(g^b, g^c)^a = e(g,g)^{abc}$
$B \longrightarrow AC$:	g^b	B "	$e(g^a, g^c)^b =$ "
$C \longrightarrow AB$:	g^c	C "	$e(g^a, g^b)^c =$ "

Key : $e(g,g)^{abc}$

Secure assuming the Decisional Bilinear Diffie-Helman (DBDH) assumption:

$$\left(g, g^a, g^b, g^c, e(g,g)^{abc}\right) \cong \left(g, g^a, g^b, g^c, e(g,g)^u\right)$$

Computational BDH:

$$g, g^a, g^b, g^c \xrightarrow{\text{HARD}} e(g,g)^{abc}$$

4-way key agreement ?? open!

<u>Major open question</u>: Construct a multi-linear map

$$e : \underbrace{G_1 \times G_1 \times \ldots \times G_1}_{k} \longrightarrow G_2$$

$$e(g^{a_1}, \ldots, g^{a_k}) \longmapsto e(g, \ldots, g)^{a_1 \cdots a_k}$$

Implies obfuscation!

Application 2: Short digital signatures

[Boneh-Lynn-Shacham 2001]

Each signature consists of only 160 bits!

Public Params:

- Groups G_1, G_2 of prime order q, $g \in G_1$ generator
- Pairing function $e: G_1 \times G_1 \longrightarrow G_2$
- H hash function from msgs to G_1 (modelled as random oracle).

KeyGen: SK: $x \leftarrow \mathbb{Z}_q$

PK: $y = g^x$ (in G_1)

$\text{Sign}_{SK=x}(m)$: $\sigma = H(m)^x$ (in G_1)

$\text{Verify}(\underset{y}{PK}, m, \sigma)$: Check: $e(g, \sigma) \stackrel{?}{=} e(g^x, H(m))$

$= e(y, H(m))$.

Thm: BLS sig scheme is existentially unforgeable against adaptive chosen msg attacks in ROM, assuming CDH is hard in G_1

Application 3: Identity-Based Encryption (IBE)

[Boneh–Franklin 2001]

IBE: Encryption scheme where my pk can be my name (or email address).

Trusted third party (TTP):

Publishes PP: G_1, G_2 prime order groups of order q.

$g \in G_1$ generator

$y = g^s$ $\quad s \leftarrow \mathbb{Z}_q$ is master secret key.

Let H_1 be hash function (modelled as RO) mapping names (eg. alice@mit.edu) to elements in G_1

Let H_2 be hash function (modelled as RO) mapping G_2 to msg space.

Goal: Enable anyone to encrypt a msg for Alice, knowing only PP & Alice's "name".

Encrypt$(\overset{g^s\ (=PP)}{\overset{\shortparallel}{y}}, \text{name}, m)$:

Choose $r \leftarrow \mathbb{Z}_q$

Output $\left(g^r,\ m \oplus H_2(g_A^{\,r})\right)$

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

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

Alice obtains $d_A = \left(H_1(\text{name})\right)^s$ from TTP.

Alic's secret key.
Needs to obtain it only once.
Note: TTP also knows it.

Compute :

$$m = v \oplus H_2\left(e(\overbrace{H_1(\text{name})^s}^{d_A}, g^r)\right)$$

$$= v \oplus H_2\left(e(H_1(\text{name})^r, y)\right)$$

$$= v \oplus H_2\left(\underbrace{e(H_1(\text{name}), y)}_{g_A}{}^r\right)$$

Security : Semantically secure in ROM assuming comp BDH

$$\left[\ g^a, g^r, \underset{H_1(\text{name})}{\overset{\shortparallel}{Q}} \xrightarrow{\text{HARD}} e(Q, g)^{ar}\ \right]$$