Today:

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

**Gap groups**

A gap group is a group where

\[ \text{Decisional Diffie-Hellman} \]

- DDH is easy:

\[ (g, g^a, g^b, g^{ab}) \not\equiv (g, g^a, g^b, g^{ab}) \]

\[ \text{Computational Diffie-Hellman} \]

- CDH is hard

\[ g, g^a, g^b \xrightarrow{\text{hard}} g^{ab} \]

\[ \textbf{Note: } \text{CDH is easy } \Rightarrow \text{DDH is easy} \]

The difference between DDH being easy & CDH being hard forms a gap.
Q1: Why do we want a "gap group"?

G2: How can we construct a gap group?

Bilinear maps

Suppose: $G_1$, group of prime order $g$ with generator $g$,

$G_2 = \{ \ldots, h \}$

[We use multiplicative notation for both groups]

& there exists a bilinear map

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

$\forall a, b \in \mathbb{Z}_q \quad e(g^a, g^b) = e(g, g)^{a \cdot b} \begin{cases} = e(g^{a+b}, g) = \ldots \\ = e(g^b, g^a) = \ldots \end{cases}$

$e(g, g) = h$

non-degenerate.

Bilinear maps are also called pairing functions.

They have numerous applications!
Thm: If there exists a bilinear map 
\[ e: G_1 \times G_1 \rightarrow G_2 \]
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 elliptic curve crypto.

2000: First "good" use

[Joux]: 3-way key agreement

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

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

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

**Application 1**: 3-way key agreement

Recall DH:

\[
\begin{array}{c}
A \\
g^a \rightarrow \\
B \\
g^b \leftarrow \\
\text{key } g^{ab}
\end{array}
\]

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

\[
\begin{align*}
A & \rightarrow BC: \quad g^a \\
& \quad A \text{ computes } e(g^a, g^b)^a = e(g, g)^{abc} \\
B & \rightarrow AC: \quad g^b \\
& \quad B \quad e(g, g)^b = " \\
C & \rightarrow A B C: \quad g^c \\
& \quad C \quad e(g^a, g^b)^c = " \\
\text{key: } e(g, g)^{abc}
\end{align*}
\]
Secure assuming the Decisional Bilinear Diffie-Hellman (DBDH) assumption:

\[
(g, g^a, g^b, g^c, e(g,g)^{abc}) \equiv (g, g^a, g^b, g^c, e(g,g)^u)
\]

Computational BDH:

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

4-way key agreement ?? open!

Major open question: Construct a multi-linear map

\[
e: G_1 \times G_1 \times \ldots \times G_k \rightarrow G_2
\]

\[
e(g_1^{a_1}, \ldots, g_k^{a_k}) \mapsto e(g_1, \ldots, g_k)^{a_1 \cdot \ldots \cdot a_k}
\]

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 $e : G_1 \times G_1 \rightarrow G_2$
- $H$ hash function from msgs to $G_1$
  (modelled as random oracle).

KeyGen:

$sk : x \in \mathbb{Z}_q$

$pk : y = g^x \ (in \ G_1)$

Sign_{sk} (m):

$c = H(m)^x \ (in \ G_1)$

Verify (pk, m, c):

Check: $e(g, c) = e(g^x, H(m))$

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):

Published $G_1, G_2$ prime order groups of order $q$.

PP:

$g \in G_1$ generator

$y = g^a, \ a \in \mathbb{Z}_q$ 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".
Encrypted (y, name, m):

Choose r \in \mathbb{Z}_q

Output \((g^r, m \oplus H_2(g_A^r))\)

where \(g_A = e(H_1(name), y)\)

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

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

Alice's secret key. Needs to obtain it only once.

Note: TTP also knows it.

Compute:

\[ m = u \oplus \underbrace{H_2(e(H_1(name)^s, g^r))}_{\text{Alice's}} \]

\[ = u \oplus \underbrace{H_2(e(H_1(name)^r, y))}_{\text{Alice's secret key}} \]

\[ = u \oplus \underbrace{H_2(e(H_1(name), y)^r)}_{g_A} \]

Security: Semantically secure in ROM assuming

comp BDH

\(\left[ g^0, g^r, g^\text{HARD}, e(q, g)^{2r} \right]_{H_1(name)}\)