Bar Ilan Winter School. Feb. 2013

Bilinear Pairings in Cryptography:

Basics of Pairings

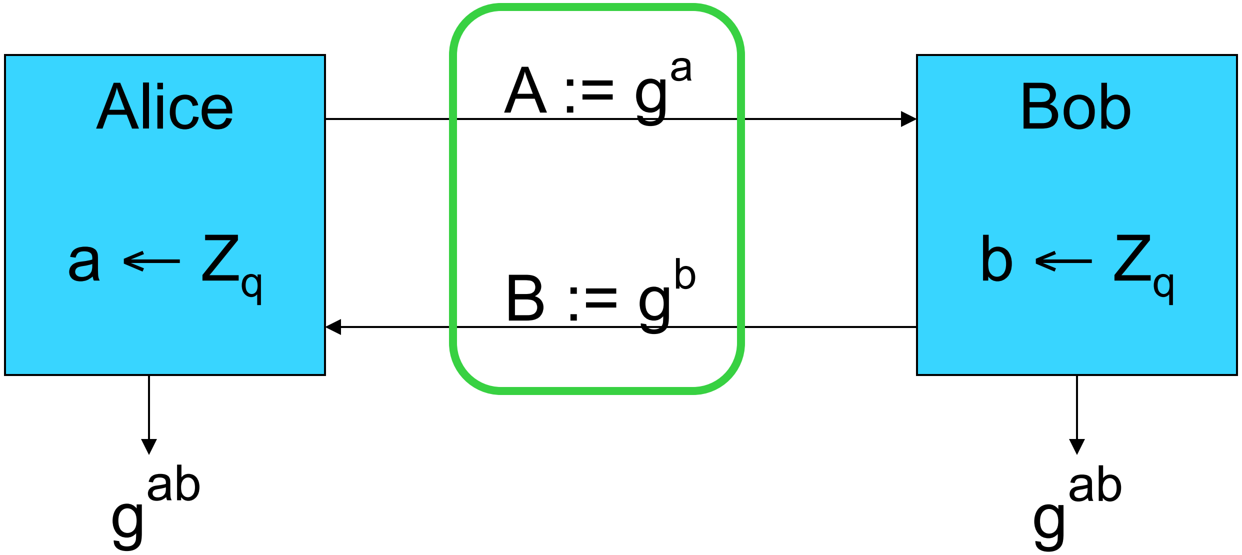
Dan Boneh

Stanford University

(1 hour)

Recall: Diffie-Hellman protocol

- G: group of prime order q ; $g \in G$ generator



- Security: Decision Diffie-Hellman assumption in G:

(g, A, B, g^{ab}) indist. from (g, A, B, g^{rand})

Standard complexity assumptions

- G: group of order q ; $1 \neq g \in G$; $x,y,z \leftarrow Z_q$
- **Discrete-log** problem: $g,\ g^x \Rightarrow x$

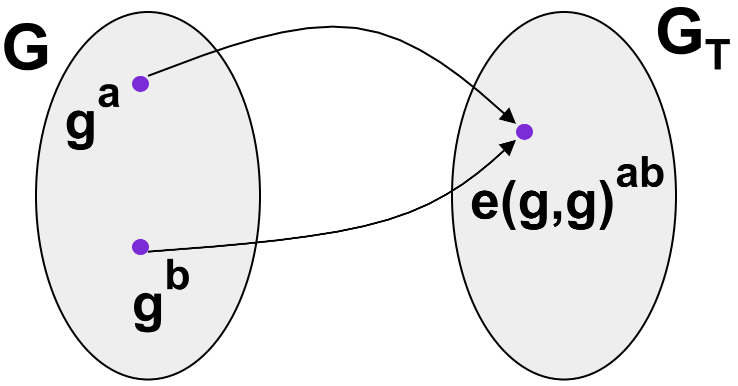
- **Computational Diffie-Hellman** problem (CDH):

$$g,\ g^x,\ g^y \Rightarrow g^{xy}$$

- **Decision Diffie-Hellman** problem (DDH):

$$g,\ g^x,\ g^y,\ g^z \Rightarrow \begin{cases} 0 & \text{if } z=xy \\ 1 & \text{otherwise} \end{cases}$$

Pairings



- G , G_T : finite cyclic groups of prime order q.

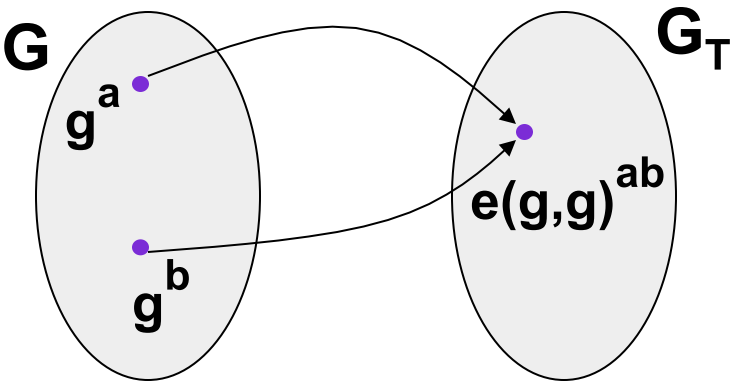
- Def: A **pairing** $e: G \times G \to G_T$ is a map:
  - Bilinear: $e(g^a, g^b) = e(g,g)^{ab}$ $\forall a,b \in Z, g \in G$
  - Poly-time computable and non-degenerate:

    g generates G ⇒ e(g,g) generates G_T

- Current examples: $G \subseteq E(\mathbf{F}_p)$, $G_T \subseteq (\mathbf{F}_{p\alpha})^*$

  (α = 1, **2**, 3, 4, **6**, 10, 12)

Pairings



- G , G_T : finite cyclic groups of prime order q.

$$e(g^{x}, h^{y}) = e(g^{y}, h^{x})$$

- Current examples: $G \subseteq E(\mathbf{F}_p)$, $G_T \subseteq (\mathbf{F}_{p\alpha})^*$

(α = 1, **2**, 3, 4, **6**, 10, 12)

Consequences of pairing

- **Decision Diffie-Hellman** (DDH) in G is easy: [J' 00, JN' 01]
  - input: $g,\ g^{\mathbf{x}},\ g^{y},\ g^{z} \in G$
  - to test if $z=xy$ do: $e(g, g^{\mathbf{z}}) \overset{?}{=} e(g^{\mathbf{x}}, g^{\mathbf{y}})$

- Dlog reduction from G to G_T : [MOV ' 93]

DLog in G $g,\ g^{a} \in G$ $\Rightarrow$ DLog in G_T $e(g,g),\ e(g,g^{a}) \in G_T$

Basic complexity assumptions in bilinear groups

- $e: G \times G \rightarrow G_T$; $1 \neq g \in G$; $x,y,z \xleftarrow{\checkmark} Z_q$
- **Discrete-log** problem: $g, \quad g^x \quad \Rightarrow \quad x$ ✓

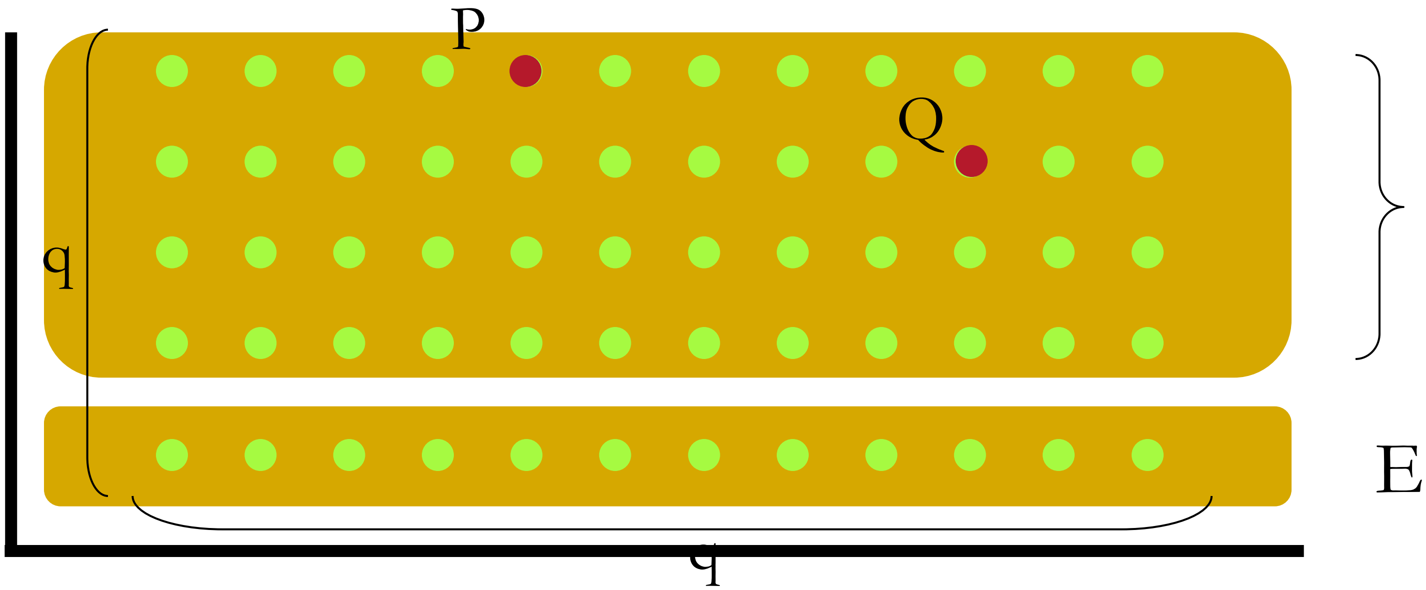
- **Computational Diffie-Hellman** problem (CDH): ✓

$$g, \quad g^x, \quad g^y \quad \Rightarrow \quad g^{xy}$$

- ***Bilinear*** **Decision Diffie-Hellman** problem (BDDH):

$$h, \quad g, \quad g^x, \quad g^y, \quad e(h, g)^z \quad \Rightarrow \quad \begin{cases} 0 & \text{if} \quad z=xy \\ 1 & \text{otherwise} \end{cases}$$

Where pairings come from …



$E(\mathbf{F}_{p^\alpha})[q]$

$E(\mathbf{F}_p) = G$

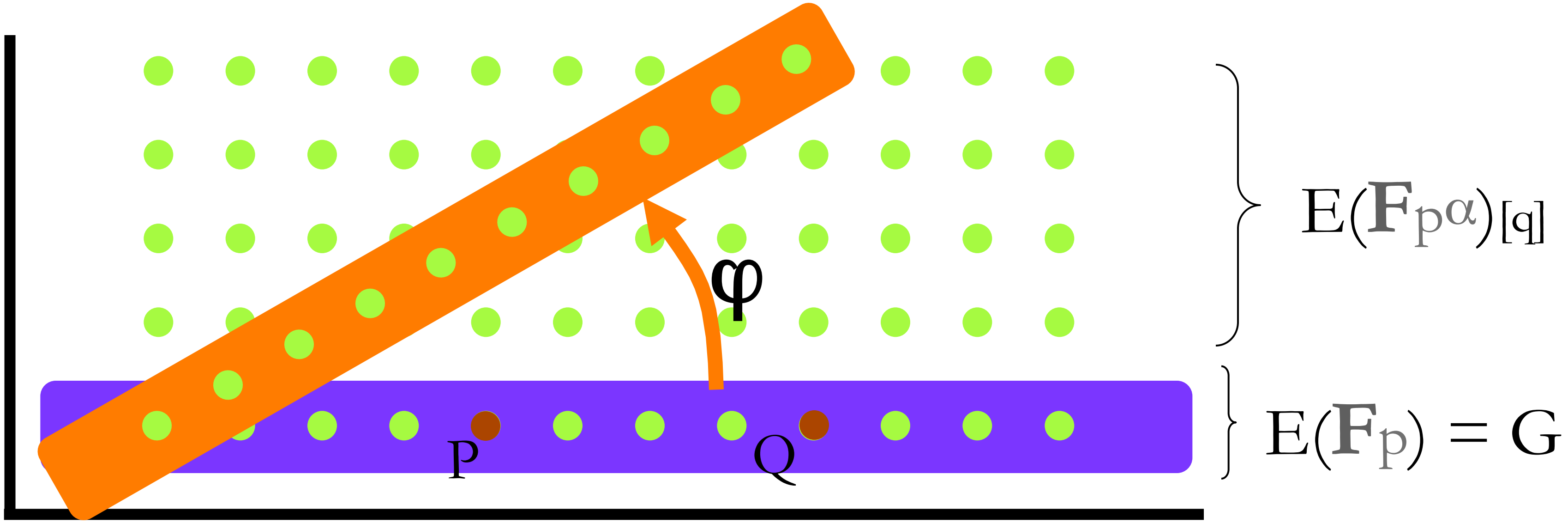
Tate pairing: $\boxed{e(P, Q) := f_P(Q)^{(p^\alpha - 1)/q}}$, $(f_P) = q\cdot(P) - q\cdot(O)$

V. Miller (84): f_P has a short straight line program

… but: $\forall\, P,Q \in G$: $e(P,Q) = 1$

Supersingular bilinear groups

Supersingular curves: (e.g. $y^2 = x^3 + x$, $p=3 \pmod 4$)



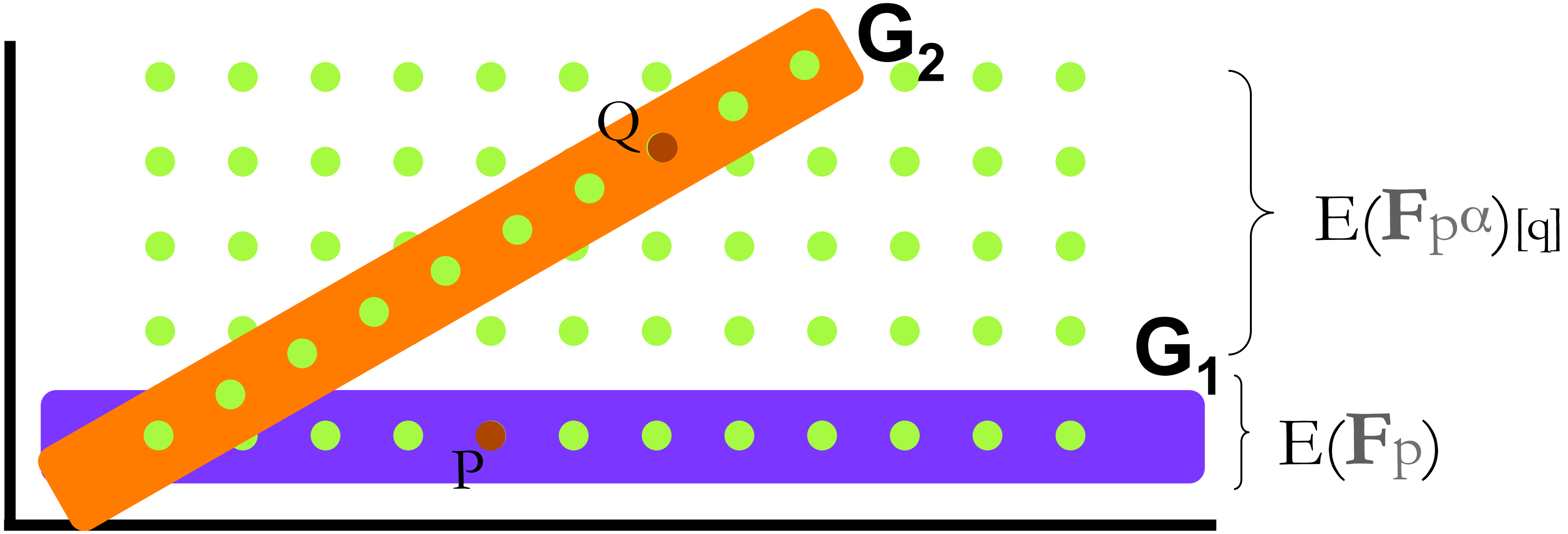
$$\bar{e} : G \times G \rightarrow G_T$$

Def: $\bar{e}(P, Q) = e(P, \varphi(Q))$

Possible α: α=2,3,4,6 or "α"=7.5 [RS '02]

Asymmetric pairings $\mathbf{e: G_1 \times G_2 \rightarrow G_T}$

Non-supersingular curves: (1st case) $(G_1 \neq G_2)$

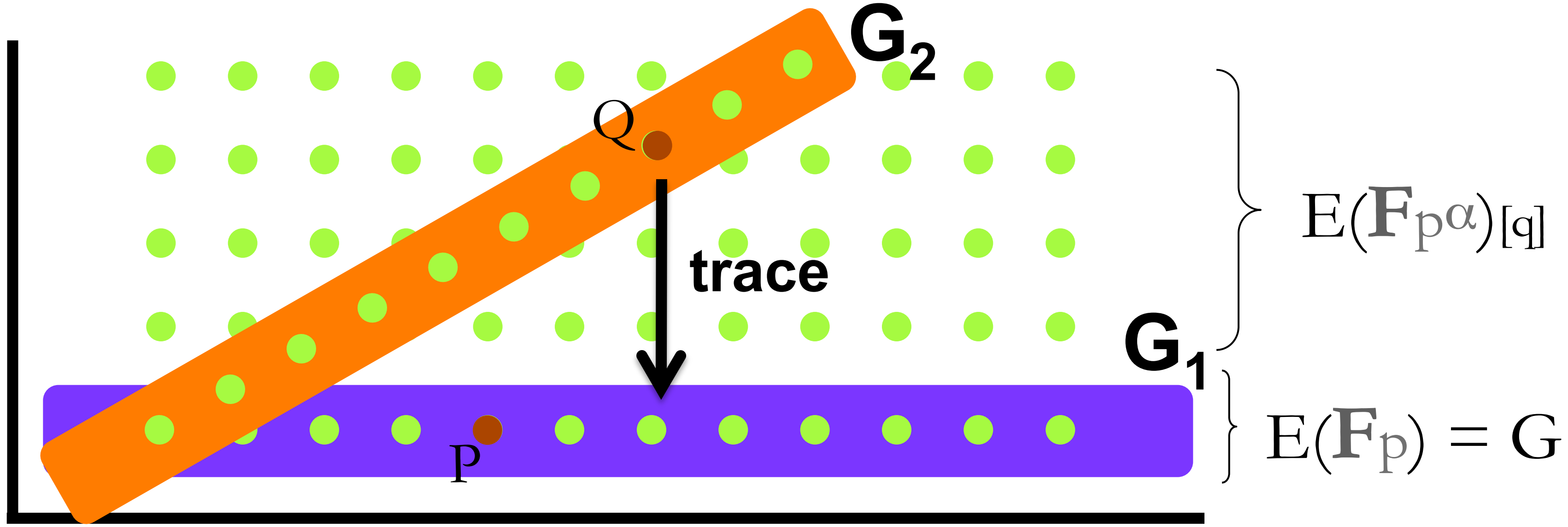


No mapping φ out of $E(F_p)$

$$\boldsymbol{e} : G_1 \times G_2 \rightarrow G_T$$

Asymmetric pairings $e: G_1 \times G_2 \rightarrow G_T$

Non-supersingular curves: (1st case) $(G_1 \neq G_2)$



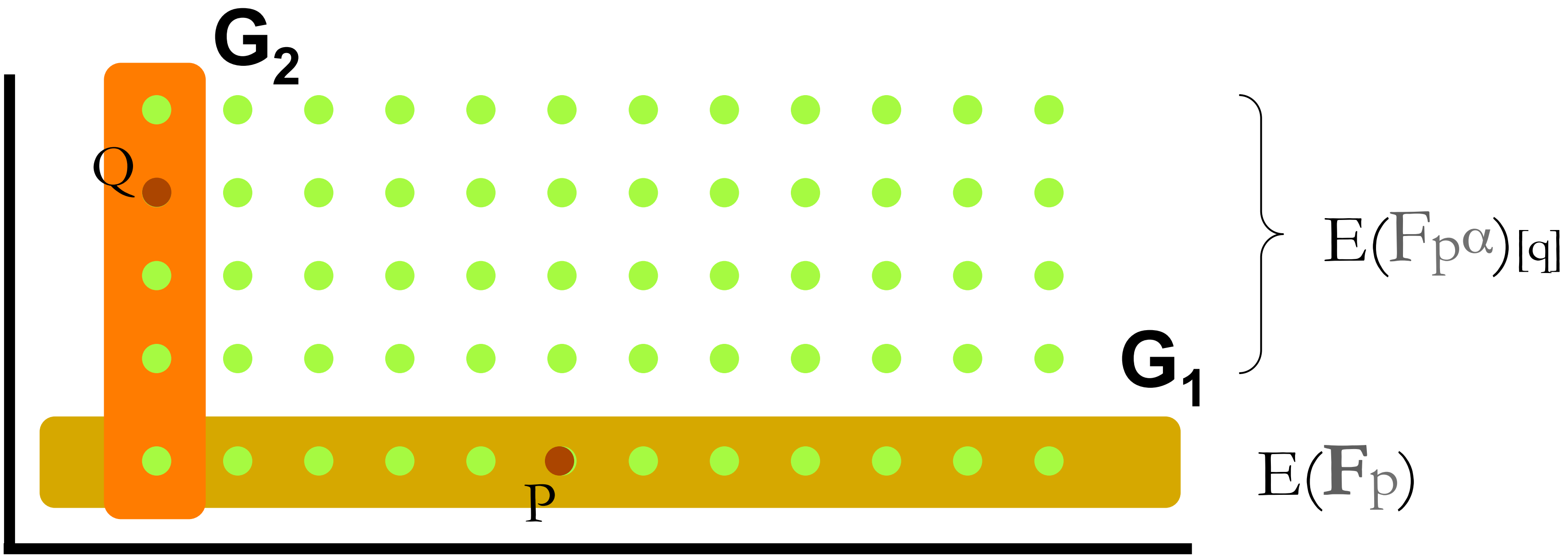
Projection map tr: $G_2 \longrightarrow G_1$ $\Rightarrow$

Symmetric pairing on G_2 $\Rightarrow$ easy DDH in G_2

… but no (known) DDH algorithm in G_1

Asymmetric pairings $\mathbf{e: G_1 \times G_2 \rightarrow G_T}$

Non-supersingular curves: (2nd case)



No projection map ⇒ no known DDH algorithm in G_1 or G_2

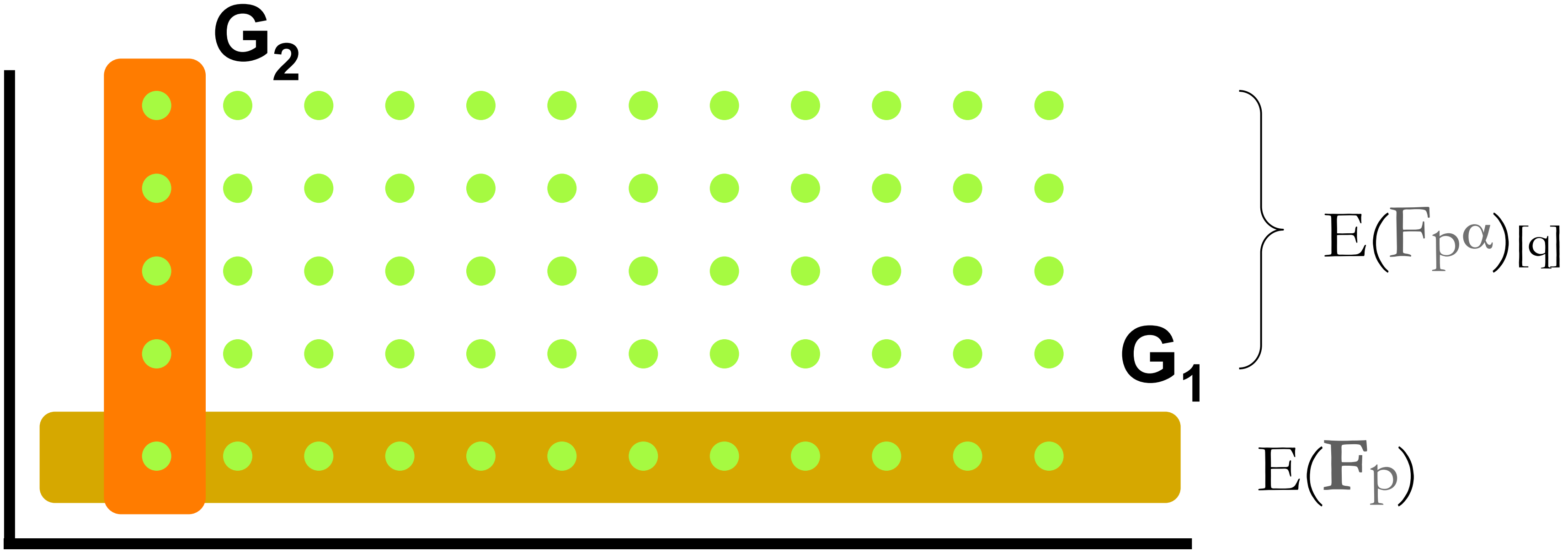
SXDH assumption: DDH hard in G_1 and G_2

- Used for anonymous IBE, circular insecure enc., …

[D'10] [ABBC'10, CGH'12]

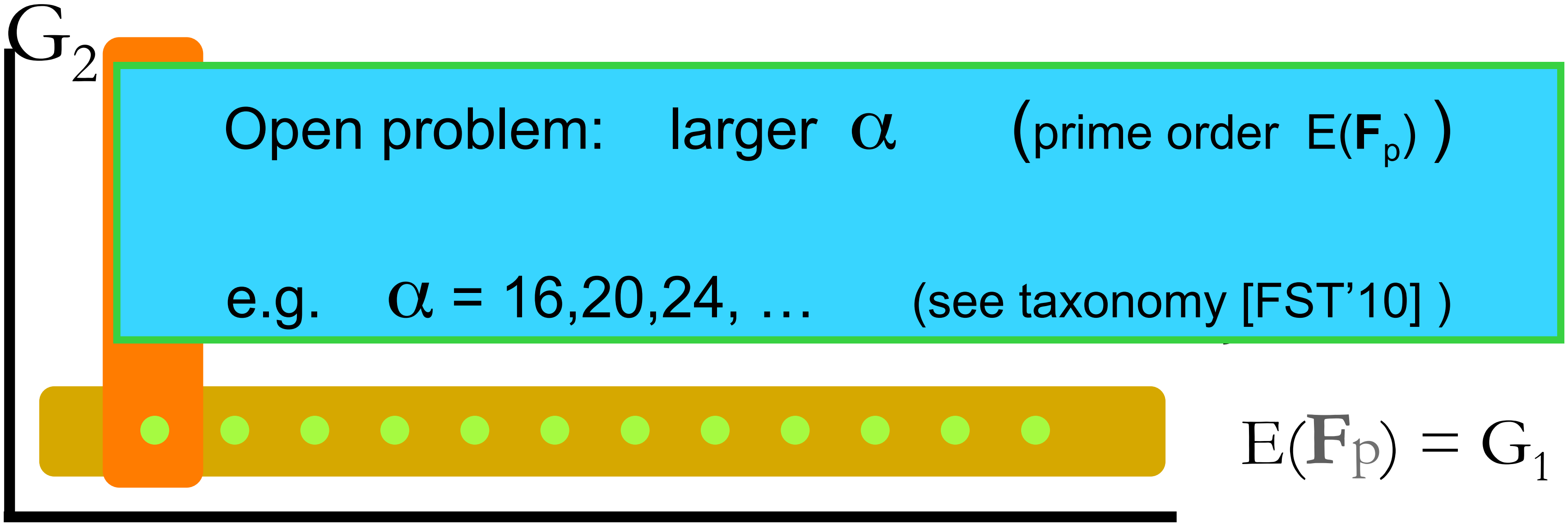
Asymmetric pairings $\mathbf{e: G_1 \times G_2 \rightarrow G_T}$

Non-supersingular curves: (2nd case)



Most efficient implementations

MNT and BN groups: asymmetric pairings



Open problem: larger α (prime order $E(\mathbb{F}_p)$)

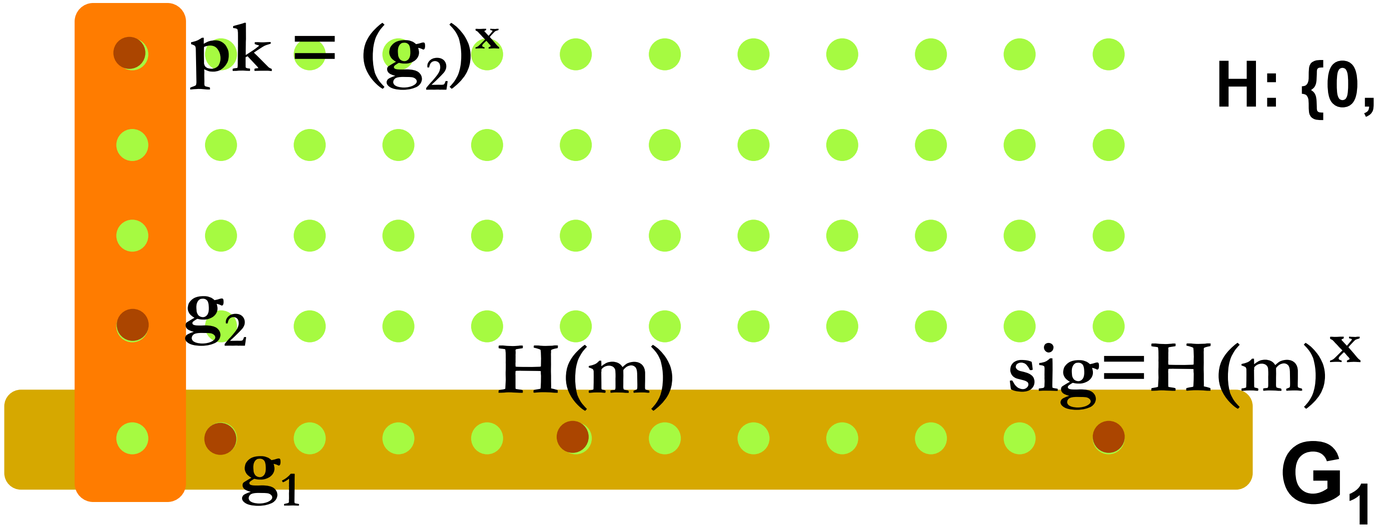
e.g. α = 16,20,24, … (see taxonomy [FST'10])

$$\mathbf{e} : G_1 \times G_2 \rightarrow G_T$$

- MNT '01 Curves: α=2,3,4,6
- BN '05, F'05 Curves: α=10, 12

} not supersingular curves

Example: BLS sigs. using asymmetric pairings



$H: \{0,1\}^* \longrightarrow G_1$

KeyGen: output $[g_1,\ g_2,\ pk=(g_2)^x]$, $sk \leftarrow x$

Sign(sk, m): output $sig \leftarrow H(m)^x \in G_1$

Verify(pk, m, s): accept iff $e(\,H(m)\,,\,pk\,) \overset{?}{=} e(\,sig,\,g_2\,)$

Security: EUF-CMA assuming aCDH (in RO model)

$$g_2,\ g_2^x,\quad g_1,\ g_1^x,\ g_1^y \quad \not\Rightarrow \quad g_1^{xy}$$

More complexity assumptions in bilinear groups

The decision linear assumption (DLIN) [BBS'04]

The **k-DLIN** assumption in G: (prime order q)

$$\begin{bmatrix} g_1 , & g_2 , \ldots, & g_k, & g_{k+1} \\ g_1^{x_1}, & g_2^{x_2}, \ldots, & g_k^{x_k} , & (g_{k+1})^{\sum x_i} \end{bmatrix} \approx_p \begin{bmatrix} g_1 , & g_2 , \ldots, & g_k, & g_{k+1} \\ g_1^{x_1}, & g_2^{x_2}, \ldots, & g_k^{x_k} , & (g_{k+1})^{y} \end{bmatrix}$$

Hierarchy: $\underbrace{\text{DDH} \equiv \text{1-DLIN}}_{\text{“easiest” to break}} \geq \text{2-DLIN} \geq \ldots \geq \underbrace{\text{k-DLIN}}_{\text{“harder” to break}} \geq \ldots$

Fact: (k+1)-linear map in G ⇒ k-DLIN is false (homework)

Assumption: k-DLIN holds even if k'-linear map in G for k'≤k

The decision linear assumption (DLIN)

- Many bilinear constructions can be based on 2-DLIN
- A useful implication: $g \in G$ order q

k-DLIN $\Rightarrow$ $(\ k < n,m\ )$

$$\boxed{\begin{array}{l} A \xleftarrow{R} (Z_q)^{n\times m} \\ \text{output} \quad g^A \end{array}} \approx_p \boxed{\begin{array}{l} B \xleftarrow{R} (Z_q)^{n\times m} \ , \ \text{rank}(B)=k \\ \text{output} \quad g^B \end{array}}$$

The "master" assumption [BBG'04]

Let {**f**}, $\mathbf{F = \{f_0=1, f_1, f_2, \ldots, f_n\} \subseteq F_q[\, x_1, \ldots, x_m \,]}$

such that $\mathbf{f \notin span_{F_q}\left(\{f_i \cdot f_j / f_k\}_{i,j,k} \right)}$ (*)

The (F,f) assumption: in a bilinear group G of order q

$$\boxed{g^{f_1(\bar{x})}, \ldots, g^{f_n(\bar{x})}, g^{\mathbf{f(\bar{x})}}} \approx_p \boxed{g^{f_1(\bar{x})}, \ldots, g^{f_n(\bar{x})}, g^{\mathbf{y}}}$$

Thm (informal): $\forall$(F,f) satisfying (*) and poly. degree,

the (F,f) assumption holds in a **generic** bilinear group

Composite order groups

Bilinear groups of order N=pq [BGN' 05]

- G: group of order N=pq. **(p, q) – secret**

  bilinear map: e: $G \times G \rightarrow G_T$

$$G = G_p \times G_q . \qquad g_p = g^q \in G_p \quad ; \quad g_q = g^p \in G_q$$

- <u>Facts</u>: $e(g_p , g_q) = e(g^q , g^p) = e(g,g)^N = 1$

  $e(g_p , g_p^x \cdot g_q^y) = e(g_p,g_p)^x$

An example: BGN encryption [BGN'05]

- KeyGen(λ): generate bilinear group G of order $N=p\cdot q$

  $pk \leftarrow \big(G, N, g, g_p\big)$; $sk \leftarrow p$

- Enc(pk, m) : $r \leftarrow Z_N$, $\mathbf{C \leftarrow g^m (g_p)^r \in G}$

- Dec(sk, C) : $C^p = [g^m]^p \cdot [g_p{}^r]^p = (g_q)^m \in G_q$

  Output: $\mathrm{Dlog}_{g_q}(C^p)$

- Note: decryption time is $O(\sqrt{m})$

  $\Rightarrow$ require small message space (e.g. {0,1})

Homomorphic Properties

$C_1 \leftarrow g^{m_1} (g_p)^{r_1}$, $C_2 \leftarrow g^{m_2} (g_p)^{r_2} \in G$

- ❑ Additive hom: $E(\mathbf{m_1+m_2}) = C_1 \cdot C_2 \cdot (g_p)^s$
- ❑ **<u>One</u>** mult hom: $\hat{E}(\mathbf{m_1 \cdot m_2}) = e(C_1,C_2) \cdot e(g_p,g_p)^s$

More generally: **$E(m_1), \ldots, E(m_n) \rightarrow \hat{E}\big(F(m_1,\ldots,m_n)\big)$**

For any $F \in Z_N[X_1,\ldots,X_n]$ of <u>total degree 2</u>

Example: matrix-matrix product of encrypted matrices [AW' 07]

(becomes fully homomorphic with a k-linear map, for suff. large k)

Security: the subgroup assumption

Subgroup assumption: $\mathbf{G \approx G_p}$

Distribution $\mathbf{P_G(\lambda)}$**:**

$(G,g,p,q) \leftarrow \text{GroupGen}(\lambda)$

$N \leftarrow p \cdot q$

$s \leftarrow Z_N$

Output: $(G, g, N, \ \mathbf{g^s})$

Distribution $\mathbf{P_p(\lambda)}$**:**

$(G,g,p,q) \leftarrow \text{GroupGen}(\lambda)$

$N \leftarrow p \cdot q$

$s \leftarrow Z_N$

Output: $(G, g, N, \ \mathbf{(g_p)^s})$

For any poly-time A:

$$\left| \Pr[A(X) : X \leftarrow \mathbf{P_G}(\lambda)] - \Pr[A(X) : X \leftarrow \mathbf{P_p}(\lambda)] \right| < \text{neg}(\lambda)$$

Thm: BGN is semantically secure under the subgroup assumption

From composite order to prime order

A general conversion: [F’10, L’12]

composite order bilinear groups system

⇒ prime order bilinear group based on 2-DLIN

- Resulting systems are often more efficient

  (since group size is smaller)

  but are technically more complex

Final note: pairings mod N

Consider elliptic curve E: $y^2 = x^3 + ax + b \pmod{N}$

where $N=p\cdot q$ is an RSA modulus

Then $E(\mathbb{Z}/N\mathbb{Z}) = E(F_p) \times E(F_q)$

- Finding size of $E(\mathbb{Z}/N\mathbb{Z})$ is as hard as factoring N
  - ⇒ cannot compute pairings on E
  - ⇒ no known algorithm for DDH on $E(\mathbb{Z}/N\mathbb{Z})$

- But DDH becomes easy given p , q

⇒ trapdoor DDH group

Early work on pairings in crypto

- Miller 1986
- Menezes-Okamoto-Vanstone attack (IEEE ’ 93)
- Joux (ANTS ’ 00),
- Sakai-Ohgishi-Kasahara (SCIS ’ 00)
- B-Franklin (Crypto ‘01)

… and many many others since

THE END