Bar Ilan Winter School. Feb. 2013

Bilinear Pairings in Cryptography:

School Overview

Dan Boneh

Stanford University

(25 minutes)

In the beginning …

… בראשית

Crypto in F_p^* :

(dim 0)

F_p

Lots of amazing applications:

- Diffie-Hellman key exchange, pub-key encryption, digital signatures, …

But discrete log problem in F_p^* is only sub-exp hard

GNFS: $\exp(\approx \log^{1/3}(p))$, record = 530-bit prime

Discrete log in other finite groups?

Lots of other groups can be used for crypto:

extension fields, matrix groups, class groups, ….

But either sub-exp Dlog or inefficient group operation

Elliptic curves over F_p : [Miller'85, Koblitz]

- no known sub-exp Dlog algorithm, and
- efficient group operation

Security Comparison	Symmetric	F_p*	ECC
	128 bits	3092 bits	256 bits

$y^2 = x^3 + 5x + 17$

Elliptic Curve Crypto: Day 1

Cryptosystems in F_p^* generally translate to elliptic curves.

Wide deployment:

Elliptic curve Diffie-Hellman

Pairings: additional structure on elliptic curves

(Some) elliptic curves have surprising structure [Weil 1940]

Pairing e: $E(F_p) \times E(F_p) \longrightarrow F_{p^\alpha}$

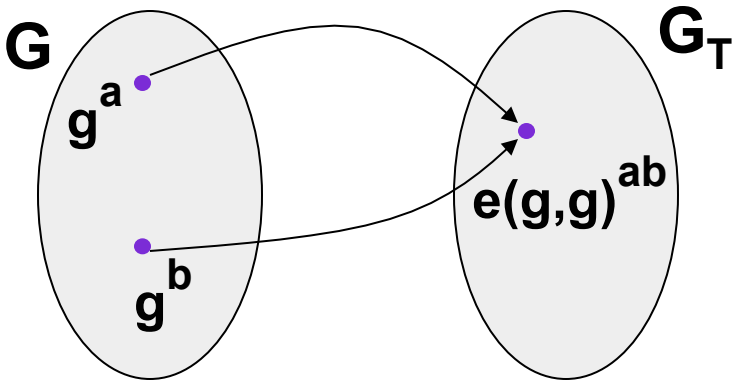
2 points on curve

finite field extension

s.t. $\forall\ P \in E(F_q)$, $a,b \in Z$: **$e(aP, bP) = e(P,P)^{ab}$**

and e(P,Q) is <u>efficiently</u> computable [Miller'86]

More abstractly: bilinear groups



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

- Def: A **pairing** $e: G \times G \rightarrow G_T$ 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 $\Rightarrow$ 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-based crypto: days 2,3,4

Encryption schemes with new properties:

Identity-based, attribute-based, functional,

Broadcast, BGN-Homomorphic, Searchable, CCA, …

Signature systems with new properties:

Short, Aggregate, Append-only, VRF,

Short group sigs, e-cash, anon. credentials …

Protocols: 3-way DH, efficient NIZKs, SNARGs, …

Simplest example: BLS signatures

[B-Lynn-Shacham'01]

KeyGen: sk = rand. x in Z_q , pk = $g^x \in G$

Sign(sk, m) $\longrightarrow$ $H(m)^x \in G$ $\quad e(g, H(m)^x) = e(g^x, H(m))$

Verify(pk, m, sig) $\longrightarrow$ accept iff $\quad e(g, \text{sig}) \stackrel{?}{=} e(pk, H(m))$

Thm: Existentially unforgeable under CDH in the RO model

New properties: (unknown with F_p*)

- **Short:** signature is a single element in G
- **Aggregatable** [BGLS'02]

Aggregating BLS signatures [BGLS'02]

user 1: $pk_1 = g^{x_1}$, $m_1 \longrightarrow s_1 = H(m_1)^{x_1}$

⋮

user n: $pk_n = g^{x_n}$, $m_n \longrightarrow s_n = H(m_n)^{x_n}$

$S \leftarrow s_1 \cdots s_n$

Aggregate S convinces verifier that msgs were signed by users 1, …, n.

Applications: cert. chains, bitcoins, SBGP

Verifying an aggregate signature: (incomplete)

$$\prod_{i=1} e\left(H(M_i), h^{x_i}\right) \stackrel{?}{=} e(S, h)$$

$$\prod_{i=1} e\left(H(M_i)^{x_i}, h\right) = e\left(\prod_{i=1} H(M_i)^{x_i}, h\right)$$

How pairings work: Day 4

Miller's algorithm and optimizations

Basis of several pairings implementations:

- PBC, jPBC , TinyPBC, MIRACL

Beyond bilinear maps [BS'03, GGH'12]

k-linear map $e: \underbrace{G \times G \times \cdots \times G}_{k} \longrightarrow G_k$

non-degenerate, efficient, hard Dlog in G

Even more applications:

- Optimal broadcast encryption,
  optimal traitor tracing, ABE for circuits [SW'12], …

Can they be constructed?

k-linear maps: a recent breakthrough

S. Garg, C. Gentry, S. Halevi

Properties: (informal)

- "randomized" representation of group elements
- Representation of $g \in G$ is $O(k)$ bits
- More than k-linear map: **gradation**

$e_1: G \times G \longrightarrow G_2$

$e_2: G \times G_2 \longrightarrow G_3$

$\vdots$

$e_k: G \times G_k \longrightarrow G_{k+1}$



The future

- Lots of work to do to extend and enhance

  bilinear constructions using k-linear maps

- Many (but not all) bilinear techniques translate to lattices
  - On going effort -- more on this tomorrow
  - Many open questions: 3-way DH, broadcast enc., …

… but first need to understand bilinear techniques