

# Advanced Topics

Nigel Smart

nigel@cs.bris.ac.uk

January 27, 2009

# Outline

## The ECDLP

- Generic Algorithms

- Special Algorithms

## Counting Points

## Pairing Based Protocols

- Tripartite Key Agreement

- Short Signatures

- ID-Based Encryption

- ID-Based Signatures

# ECDLP

# How hard is the ECDLP?

**Black box group** means there is no information about the representation.

- ▶ Only **generic algorithms** are available.

**Theorem:** (V. Shoup)

In a black box group of prime order  $L$  it takes at least  $O(\sqrt{L})$  operations to solve the discrete logarithm problem.

**Compare to factoring where underlying problem is sub-exponential**

# Generic Algorithms

## Baby step giant step (Shanks)

Want to find  $0 \leq \lambda < L$  such that  $Q = \lambda P$  in  $E(\mathbb{F}_q)$ .

Put  $M = \lceil \sqrt{L} \rceil$  (or  $M = \lceil \sqrt{L/2} \rceil$ ).

Make a **list of baby steps**:

- ▶  $\mathcal{O}_E, P, 2P, \dots, MP$ .

Take **giant steps**  $Q - MP, Q - 2MP, \dots$  until find a match

$$Q - \lambda_1 MP = \lambda_0 P$$

with the list.

Then  $\lambda = \lambda_0 + M\lambda_1$ .

- ▶ **Time:**  $O(\sqrt{L})$ .
- ▶ **Memory:**  $O(\sqrt{L})$ .

# Pollard methods

Use deterministic random walks in  $E(\mathbb{F}_q)$ :

- ▶ Partition  $E(\mathbb{F}_q)$  into  $2^n$  sets  $G_1, \dots, G_{2^n}$ .
- ▶ Construct  $2^n$  random points  $P_i = \alpha_i P$ .
- ▶ Random walk  $X \mapsto (X + P_i \text{ if } X \in G_i)$ .

Method:

- ▶ Start at  $X = P$  and take  $O(\sqrt{L})$  steps in random walk
  - ▶ and store the final value  $Y = \alpha P$ .
- ▶ Start at  $X = Q$  and take steps in walk until hit  $Y$ .
- ▶ Have  $Q + \alpha' P = \alpha P$ .
  
- ▶ **Time:**  $O(\sqrt{L})$ .
- ▶ **Memory:**  $O(1)$ .

# Parallel Pollard (Van Oorschot-Wiener)

Distinguished point set  $\mathcal{D}$ , size  $\theta \# E(\mathbb{F}_q)$ .

**Method:** Suppose we have  $M$  processors in parallel.

- ▶ Each processor starts at a random point  $X = \alpha P + \beta Q$ 
  - ▶ and walks in the group.
- ▶ Every time a distinguished point  $X$  is encountered then send
  - ▶  $(X, \alpha, \beta)$  to the central server.
- ▶ When the server receives  $(X, \alpha, \beta)$  and  $(X, \alpha', \beta')$ 
  - ▶ then can solve for discrete logarithm.
  
- ▶ **Time:**  $\sqrt{\pi L/2}/M + L/(\theta \# E(\mathbb{F}_q))$ .
- ▶ **Server memory:**  $\theta \sqrt{L}$ .

In practice:  $L \sim 2^{100}$ ,  $M \sim 2^{10}$  and  $\theta = 2^{-30}$ .

## Equivalence classes (W-Z/G-L-V)

### Example :

- ▶ For  $P = (x, y)$  have  $-P = (x, -y)$ .
- ▶ Impose a canonical choice of representative for elements of the
  - ▶ set  $E(\mathbb{F}_q)/\langle \pm 1 \rangle$
- ▶ Define the random walk on this set instead.
- ▶ Pollard methods are faster by a factor of  $\sqrt{2}$ .

### Example:

- ▶ Consider a subfield curve  $E/\mathbb{F}_2$  with discrete logarithm
  - ▶ problem in  $E(\mathbb{F}_{2^l})$  (Koblitz curve)
- ▶ Action of  $\pm Frob_2$  gives equivalence classes of size  $2l$ .
- ▶ So method faster by factor  $\sqrt{2l}$ .

Koblitz curves are recommended in some standards eg ANSI, NIST etc

## Special Algorithms

# Special Attacks

There are a number of special attacks.

- ▶ Only apply to certain curves
- ▶ Analogous to weak RSA keys

Unlike weak RSA keys, weak ECC keys are easily detected by any user

- ▶ i.e. can be detected by anyone and not just the person who makes the key.

# Menezes-Okamoto-Vanstone/Frey-Rück

Construct group homomorphism

$$E(\mathbb{F}_q) \longrightarrow \mathbb{F}_{q^k}^*$$

where  $k$  is the smallest integer such that the exponent of  $E(\mathbb{F}_q)$  divides  $q^k - 1$ .

Can solve discrete logarithm problem in  $\mathbb{F}_{q^k}^*$  using an index calculus algorithm of subexponential complexity (in  $q^k$ ).

$E$  supersingular implies  $k \leq 6$ .

General case  $k \sim q$ .

- ▶ But are some special cases for ordinary curves.

# Menezes-Okamoto-Vanstone/Frey-Rück

There is a pairing, the (modified) Tate pairing, such that for supersingular curves

$$t : \begin{cases} E(\mathbb{F}_q) \times E(\mathbb{F}_q) & \longrightarrow \\ (P, Q) & \longmapsto f_{n,P}(Q)^{(q^k-1)/n} \end{cases} \mathbb{F}_{q^k}^*$$

where

- ▶  $(f_{n,P}) = n(P) - n(\mathcal{O})$
- ▶  $n = \#E(\mathbb{F}_q)$
- ▶  $t(P, Q)$  is bilinear
- ▶ If  $P, Q \neq \mathcal{O}$  then  $t(P, Q) \neq 1$

# Menezes-Okamoto-Vanstone/Frey-Rück

To solve

$$Q = \lambda P$$

Compute

- ▶  $g = t(P, P)$
- ▶  $h = t(Q, P)$

Try to solve, in the finite field,

$$\begin{aligned} h &= t(Q, P) \\ &= t(\lambda P, P), \\ &= t(P, P)^\lambda, \\ &= g^\lambda. \end{aligned}$$

# Menezes-Okamoto-Vanstone/Frey-Rück

For general curves the modified Tate pairing is defined as

$$t : \begin{cases} E(\mathbb{F}_q) \times \overline{E}(\mathbb{F}_{q^e}) & \longrightarrow \\ (P, Q) & \longmapsto f_{n,P}(Q)^{(q^k-1)/n} \end{cases} \mathbb{F}_{q^k}^*$$

where

- ▶  $(f_{n,P}) = n(P) - n(\mathcal{O})$
- ▶  $n = \#E(\mathbb{F}_q)$
- ▶  $t(P, Q)$  is bilinear
- ▶ If  $P, Q \neq \mathcal{O}$  then  $t(P, Q) \neq 1$
- ▶  $e = k/d$  where  $d$  is the largest possible twist
- ▶  $\overline{E}$  is a  $d$ -th twist of  $E$ , which is a curve defined over  $\mathbb{F}_{q^e}$ .

## Semaev/Smart/Araki-Sato

Suppose  $E(\mathbb{F}_p)$  has exactly  $p$  points.

Construct a group homomorphism

$$E(\mathbb{F}_p) \longrightarrow \mathbb{F}_p^+.$$

### Methods:

- ▶ Using  $p$ -adic logarithm and  $p$ -adic lift.
- ▶ Take a function  $f$  such that  $(f) = p(P) - p(\mathcal{O})$  and consider the holomorphic differential  $\omega = \frac{1}{f} df$ .

Discrete logarithm problem in  $\mathbb{F}_p^+$  solved using Euclid's algorithm.

# Weil Descent

Only (currently) applies to fields of characteristic two.

If curve defined over  $\mathbb{F}_{q^n}$  for a small value of  $n$  can reduce ECDLP to a HCDLP.

For some values of  $n$ , eg  $n = 4$  this weakens the curve.

- ▶ Work of Frey, Galbraith, Gaudry, Hess and Smart

Values of  $n$  of 4, 5, 6 sometimes chosen for efficiency reasons.

Note, only standard which uses such curves is IPsec (I think)

## New Techniques

The most successful of the more modern techniques (post 2005) have been those in the Semaev/Gaudry/Diem family.

- ▶ Use a combination of index calculus, division polynomials and Groebner basis.

Almost all are non-practical but they obtain sub-exponential complexity for infinite families of curves over fields of the form

$$\mathbb{F}_{p^n}$$

where  $p$  and  $n$  lie in certain regions.

- ▶ Sort of “medium characteristic” fields.

# Gaudry's Method

For curves over  $\mathbb{F}_{q^n}$

For fixed  $n$ , but with  $q$  tending to infinity, Gaudry obtains a complexity of

$$O(q^{2-2/n}).$$

Comparing to Pollard rho of  $O(q^{n/2})$  we see that for  $n = 4$  this is more efficient.

- ▶ But is totally impractical

## Diem's Method

If  $a > 2 + \epsilon$  and  $(2 + \epsilon)n^2 \leq \log_2(q) \leq an^2$  then obtain

$$\exp(O(1)) \cdot (\log(q^n))^{2/3}$$

i.e.  $L_{q^n}(1/3, c \cdot \sqrt{a})$  for some constant  $c$ .

This generalises Gaudry's result.

We essentially obtain a polynomial algorithm in  $q$  as long as  $\log_2(q)$  is larger than  $2n^2$ .

- ▶ Hence if  $q$  is subexponential in  $q^n$  then we get a subexponential algorithm.

Again the method is totally impractical, even for small values of  $n$  and  $q$ .

## How big?

To compare ECC against other technologies we use the following table provided by NIST

Block Cipher Key Size	Example Block Cipher	ECC Key Size	RSA Key Size
80	SKIPJACK	163	1024
128	AES (small)	283	3072
192	AES (medium)	409	7680
256	AES (large)	571	15360

ECC at 571 bits is usable, RSA at 15360 bits is not.

Although 571 is really huge, conservative managers may want to go for the **highest level of security possible**.

## Counting Points

# Counting points

To use an elliptic curve in a real system we first need to know

$$\#E(\mathbb{F}_q).$$

For some curves this is easy

- ▶ Koblitz curves

For others we need to be more clever to compute this number

## Frobenius endomorphism

$$\pi : (x, y) \mapsto (x^q, y^q).$$

## Characteristic polynomial

$$\pi^2 - t\pi + q = 0.$$

Then have  $\#E(\mathbb{F}_q) = q + 1 - t$ .

Hasse:  $|t| \leq 2\sqrt{q}$ .

Computing  $\#E(\mathbb{F}_q)$  is equivalent to computing  $t$ .

## Idea:

Compute the value of  $t$  modulo small primes (or prime powers) /  
Recover  $t$  using the Chinese remainder theorem and the bound  
 $|t| \leq 2\sqrt{q}$ .

Very complicated algorithm.

- ▶ Can be made to be very efficient using ideas of Elkies and Atkin
- ▶ Can compute  $\#E(\mathbb{F}_q)$  for a given curve in a matter of seconds for most interesting values of  $q$ .

# Satoh

Satoh in 1998 invented a new method which is better than Schoof for fields  $\mathbb{F}_{p^n}$  of small characteristic  $p$ , eg characteristic two,

Lifts the curve to a  $p$ -adic extension.

Applies a  $p$ -isogeny  $n$  times, to obtain an isogeny cycle.

Use this to write down  $t$  modulo  $p^n$ .

Note : A 2-isogeny is given by the Arithmetic-Geometric mean as any standard textbook on elliptic integrals (or computing  $\pi$ ) will tell you.

- ▶ Thus Satoh's algorithm gives rise to the AGM method of Harley-Gaudry from 2001
- ▶ AGM method only useful for characteristic two.

# Introduction

# Pairing Based Cryptography

Recently attention has focused again on curves which admit efficient pairings, as they allow a new class of protocols to be defined.

Recall these curves admit the Tate [pairing](#)...

$$t : \begin{cases} E(\mathbb{F}_q) \times \overline{E}(\mathbb{F}_{q^e}) & \longrightarrow & \mathbb{F}_{q^k}^* \\ (P, Q) & \longmapsto & f_{n,P}(Q)^{(q^k-1)/n} \end{cases}$$

where

- ▶  $(f_{n,P}) = n(P) - n(\mathcal{O})$
- ▶  $n = \#E(\mathbb{F}_q)$
- ▶  $t(P, Q)$  is bilinear
- ▶ If  $P, Q \neq \mathcal{O}$  then  $t(P, Q) \neq 1$
- ▶  $e = k/d$  where  $d$  is the largest possible twist
- ▶  $\overline{E}$  is a  $d$ -th twist of  $E$ , which is a curve defined over  $\mathbb{F}_{q^e}$ .

# Pairing Based Cryptography

We require

- ▶  $\#E(\mathbb{F}_q)$  is divisible by a large prime  $> 2^{160}$
- ▶  $q^k > 2^{1024}$

so as to obtain acceptable security levels.

In practice more likely to use the [Ate pairing](#)...

$$t : \begin{cases} E(\mathbb{F}_q) \times \overline{E}(\mathbb{F}_{q^e}) & \longrightarrow \mathbb{F}_{q^k}^* \\ (P, Q) & \longmapsto f_{T,Q}(P)^{(q^k-1)/n} \end{cases}$$

where

- ▶  $(f_{T,Q}) = T(Q) - T(\mathcal{O})$
- ▶  $T \approx \sqrt{n}$

# Pairing Based Cryptography

Some early protocols given in terms of symmetric pairings...

$$t : \begin{cases} E(\mathbb{F}_q) \times E(\mathbb{F}_q) & \longrightarrow & \mathbb{F}_{q^k}^* \\ (P, Q) & \longmapsto & f_{n,P}(Q)^{(q^k-1)/n} \end{cases}$$

where

- ▶ Here  $E$  must be a supersingular curve

# Pairing Based Cryptography

For asymmetric pairings we often write

$$t : \{ \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{F}_{q^k}^*$$

and define generators by

$$\mathbb{G}_1 = \langle P_1 \rangle \text{ and } \mathbb{G}_2 = \langle P_2 \rangle.$$

Probably the best curves to use are ones of the form

$$Y^2 = X^3 + D$$

- ▶ Have a sextic twist to aid efficiency
- ▶ Can use GLV type curve multiplications
- ▶ Easy to generate high security parameters via the Barreto-Naehrig algorithm, to get  $k = 12$ .

## Tripartite Key Agreement

# Tripartite Diffie-Hellman

Invented by Joux, ANTS-IV, 2000, uses symmetric pairings

Suppose three parties,  $A$ ,  $B$  and  $C$ , want to agree on a shared secret.

- ▶  $A$  computes  $aP$  and broadcasts it
- ▶  $B$  computes  $bP$  and broadcasts it
- ▶  $C$  computes  $cP$  and broadcasts it

They can now all compute the shared secret  $K$

- ▶  $A$  computes  $K = t(bP, cP)^a = t(P, P)^{abc}$
- ▶  $B$  computes  $K = t(aP, cP)^b = t(P, P)^{abc}$
- ▶  $C$  computes  $K = t(aP, bP)^c = t(P, P)^{abc}$

## Short Signatures

# Required Hash Functions

The following three schemes assume the following cryptographic hash functions:

- ▶  $H_1 : \{0, 1\}^* \longrightarrow E(\mathbb{F}_q) = \mathbb{G}_1$
- ▶  $H_2 : \{0, 1\}^* \longrightarrow \mathbb{F}_l$
- ▶  $H_3 : \mathbb{F}_{q^k} \longrightarrow \{0, 1\}^*$
- ▶  $H_4 : \mathbb{F}_l \longrightarrow \{0, 1\}^t$

where

- ▶  $k$  is the MOV security parameter
- ▶  $l$  is the large prime factor of  $\# E(\mathbb{F}_q)$
- ▶  $t$  is the key size of a DEM

## Short Signatures

Recall that EC-DSA had a signature size of around  $2 \log_2 q$   
In 2001 Boneh, Lynn and Shacham gave a signature algorithm whose size was  $\log_2 q$  using the Tate pairing.

**Private Key** :  $x \in \mathbb{F}_1$

**Public Key** :  $R = xP_2 \in \mathbb{G}_2$

**Signature** : To sign  $m$  compute

$$V = xH_1(m) \in \mathbb{G}_1$$

**Verification** : On input of  $m$  and  $V$  check whether

$$\begin{aligned} t(V, P_2) &= t(xH_1(m), P_2) \\ &= t(H_1(m), xP_2) \\ &= t(H_1(m), R) \end{aligned}$$

Signature is short since only a single point  $V$  is transmitted, which could be compressed.

## ID-Based Encryption

# Identity Based Cryptography

In 1983 Shamir invented identity based cryptography.  
Here a users identity becomes their public key

- ▶ No need for certificates
- ▶ No need to transfer the public key

However, Shamir only gave an ID based signature scheme.

An ID based encryption scheme which was efficient and secure did not come along until Boneh and Franklin, CRYPTO 2001.

# Boneh-Franklin Encryption Scheme

## Trust Authority Private Key

- ▶  $s \in \mathbb{F}_l$

## Trust Authority Public Key

- ▶  $R = sP_2 \in \mathbb{G}_2$

## Users Public Key

- ▶  $Q_{ID} = H_1(ID) \in \mathbb{G}_1$

- ▶ Hence anyone can obtain the public key, if they know the identity  $ID$ .

## Users Private Key

- ▶  $S_{ID} = sQ_{ID} \in \mathbb{G}_1$

- ▶ Obtained by the user from the TA
  - ▶ To enable greater security the TA secret can be split using standard techniques.

# Boneh-Franklin Encryption Scheme

To encrypt a message  $m$  to the user with identity  $ID$  the sender computes

- ▶  $r \in \mathbb{F}_l$  chosen at random
- ▶  $U = rP_2 \in \mathbb{G}_2$
- ▶  $V = m \oplus H_3(t(rQ_{ID}, R))$
- ▶ Transmit  $(U, V)$

To decrypt the owner of  $S_{ID}$  can compute

$$\begin{aligned}V \oplus H_3(t(S_{ID}, U)) &= V \oplus H_3(t(sQ_{ID}, rP_2)) \\ &= V \oplus H_3(t(rQ_{ID}, sP_2)) \\ &= V \oplus H_3(t(rQ_{ID}, R)) \\ &= m\end{aligned}$$

To obtain CCA security need to complicate the scheme slightly using the Fujisaki-Okamoto transform.

# SK-KEM

Boneh-Franklin is very inefficient due to the need to hash ID's to points.

The most efficient scheme, in the ROM, is the SK-KEM/DEM scheme.

The encryption is performed by a DEM, so we only need to define an identity based KEM

## Trust Authority Private Key

- ▶  $s \in \mathbb{F}_l$

## Trust Authority Public Key

- ▶  $R = sP_1 \in \mathbb{G}_1$

# SK-KEM

## Users Public Key

- ▶  $Q_{ID} = R + H_1(ID) \cdot P_1 \in \mathbb{G}_1$
- ▶ Hence anyone can obtain the public key, if they know the identity  $ID$ .

## Users Private Key

- ▶  $S_{ID} = (1/(x + H_1(ID))) \cdot P_2 \in \mathbb{G}_2$
- ▶ Obtained by the user from the TA

Note, for a valid key pair we have

$$t(Q_{ID}, S_{ID}) = t(P_1, P_2).$$

# SK-KEM

## Encapsulate

- ▶  $r \in \mathbb{F}_l$
- ▶  $k = H_4(r)$ .
- ▶  $U = r \cdot Q_{ID}$
- ▶  $C = r + H_2(t(P_1, P_2)^r)$ .
- ▶ Output  $(k, (U, C))$ .

## Decapsulate

- ▶  $r = C - H_2(t(U, S_{ID}))$
- ▶  $U' = r \cdot Q_{ID}$
- ▶  $k = H_4(r)$
- ▶ If  $U' = U$  then output  $k$ , otherwise output  $\perp$ .

## ID-Based Signatures

# Hess' ID-Based Signature Scheme

On publication of the Boneh-Franklin scheme a number of pairing based ID signature schemes were given.

One, whose security rests on CDH rather than some weaker assumption, and which is relatively efficient is due to Hess.

It uses same TA and user keys as the Boneh-Franklin encryption scheme.

- ▶ There are more efficient schemes now for ID-based signatures

# Hess' ID-Based Signature Scheme

To sign  $m$  compute

- ▶  $r = t(P_1, P_2)^k$  for random  $k \in \mathbb{F}_l$
- ▶  $h = H_2(m \| r)$
- ▶  $U = hS_{ID} + kP \in \mathbb{G}_1$

Output  $(U, h)$  as the signature

To verify a signature compute

- ▶  $r = t(U, P_2) \cdot t(Q_{ID}, -R)^h$

and check whether

- ▶  $h = H_2(m \| r)$ .