EFFICIENT EPHEMERAL ELLIPTIC CURVE CRYPTOGRAPHIC KEYS

Andrea Miele, Arjen K. Lenstra

MOTIVATION

- ECC: use of fixed elliptic curve, finite field from public standards "standard material/technology to manufacture keys":

1. Need to trust standard's (material/technology) designer(s)

2. Use of same standard (material/technology) incentive to attackers

- Alternatively one can generate their own curves…

IDEA

- **We want personalized real time curve selection for ECDH key-exchange, ideally a unique curve per session**
- Interference of third parties on parameter choice, exposure to cryptanalysis and attack window/payoff are all minimized

PROBLEM

(from http://stlbuyerguide.com)

- Two parties want to agree on a unique secure "ephemeral" pair elliptic curve equation, prime field for an ECDH key-exchange
- Question: can parties generate secure, unique, unpredictable, ephemeral ECC parameters in real time on their smartphones?

GENERATING ELLIPTIC CURVES FOR ECC (PRIME FIELDS)

1. For $\approx$**k** bits of security: select random **2k**-bit (recall rho's run time…) prime. Then pick a random curve $\mathbf{E_{a,b}(F_p)}$ until $\mathbf{\#E_{a,b}(F_p)}$ (quasi-)prime

2. Compute order with point-counting (SEA) (**too slow for real-time!**)

- Additionally (twist-security) search until $\mathbf{\#\tilde{E}}$ also (quasi-)prime
  For a prime **p**, $\mathbf{\#E_{a,b}(F_p) = p+1-t}$ with $\mathbf{|t| \leq 2\sqrt{p}}$, quadratic twist's order $\mathbf{\#\tilde{E} = p+1+t}$ where $\mathbf{\tilde{E} = E_{r^2a,r^3b}}$ with **r** any non-square in $\mathbf{F_p}$

POINT COUNTING

Currently, too slow for real time

MAGMA on Intel Core i7-3820QM 2.7GHz

	80-bit security	112-bit security	128-bit security
ECC	12s	47s	120s
twist-secure ECC	6m	37m	83m

COMPLEX MULTIPLICATION METHOD

1. Select a CM curve first (a subset of cryptographically interesting curves…)

2. Find a prime of a particular form

3. **Compute order in a cheap way!**

The Q-curve of Costello, Longa (Microsoft Research) is CM curve…

CM METHOD STEPS

1. Pick a square-free positive integer **$d \neq 1,3$**, compute the Hilbert class polynomial **$H_d(X)$** of **$\mathbb{Q}(\sqrt{-d})$** (degree **h_d**) assume (**$d \equiv 3 \bmod 4$**)

2. Find integers **u,v: $u^2+dv^2 = 4p$** such that **p** is prime

3. Solve **$H_d(X) \equiv 0 \bmod p$** to find root **j** then $(a,b) = \left(\frac{-27j}{4(j-12^3)}, \frac{27j}{4(j-12^3)}\right) \in \mathbb{F}_p^2$

   defines **$E_{a,b}(F_p)$** with **$\#E_{a,b} = p+1 \pm u$** and **$\#\tilde{E} = p+1 \mp u$**

REAL TIME CM

- **CM for small h_d still too slow… but for "very small" h_d (<5):**

  Solve $H_d(X)$ by radicals to get root **j**, store **d** and **(a,b)** in a table

- **[Lenstra99]**: table for $h_d=1$ (**8** curves):

start: Select random positive integers u, v_0
for $i=0$ to $L-1$
 $v=v_0+i$
 for each **d** in the table
   if p: $u^2+dv^2=4p$ is prime and $p+1\pm u$ (orders) are (quasi-)prime
    return **p** and **(a,b)** reduced modulo **p**
goto start

OUR CONTRIBUTIONS

- We extended the subset with **11** more equations
- We improved method by **sieving** for prime **p** and (quasi-)prime orders
- We implemented extra options, e.g. twist security, Montgomery-friendly
- **C** implementation based on **GMP** for PCs and Android (JNI/NDK)

SIEVING IDEA

- Base alg: fix **u**, try all **v** in **[v_0,v_0+L)** until **p_j=(u^2+d_jv^2)/4**, and orders are prime for a curve **E_j** in our table (**j<C**)

- Idea: write **p_j**, curve and twist orders as polynomials in **v** (as below)

- We can quickly identify values of **v** such that **p_j(v)**, **ord_j(v)** and **$ordT_j$(v)** are divisible by primes less than fixed bound **B** (therefore composite): avoid useless primality tests!

SIEVE

$$A[0] := \underbrace{"11\ldots1"}_{C} \quad A[1] := \underbrace{"11\ldots1"}_{C} \quad \cdots \quad A[L-1] := \underbrace{"11\ldots1"}_{C}$$

for each prime **q<B**

for j=0 to C-1 (i.e., for each **curve** E_j in the table)

find roots of $p_j(v)$, $ord_j(v)$ and $ordT_j(v)$ modulo **q**

for each root **r**

for each $i \equiv (r - v_0) \bmod q$ and $0 \leq i < L$: $A[i] := "11\ldots\underbrace{0}_{j}\ldots1"$

At the end bit-positions containing **1** are further inspected!

128-BIT SECURITY: TIMINGS

OS X 10.9.2, Intel Core i7-3820QM 2.7GHz

Prime order		
Twist security	Basic	Sieve (B, V)
No	0.009s	0.008s (100, 2^{11})
Yes	0.18s	0.05s (800, 2^{16})

Android, Samsung Galaxy S4, Snapdragon 600 1.9GHz

Prime order		
Twist security	Basic	Sieve (B,V)
No	0.065s	0.053s (200, 2^{12})
Yes	1.43s	0.39s (750, 2^{15})

EPHEMERAL CURVE DH

- Exchange hash-commitments of random seeds
  Exchange seeds, XOR them to obtain shared seed
  OR
  Use verifiable random beacon [LW15] to select shared seed (combined with identities, time, …)

- Use shared seed to initialize generation process

CONCLUSION

- We described a method to generate real time ephemeral ECC parameters for ECDH

- Future (more choice of curves):
  Faster point counting for random curve generation?

THANKS FOR YOUR ATTENTION!