ECC mod 8^91+5

especially elliptic curve 2y^2=x^3+x for cryptography

Andrew Allen and Dan Brown, BlackBerry

CFRG, Prague, 2017 July 18

$$2y^2=x^3+x/GF(8^{91}+5)$$

Simplest secure and fast ECC ?

Benefits of Galois field size $8^{91}+5$ for ECC

Feature	Benefits
6 symbols: 8^91+5	Little room for trapdoor (low Kolmogorov complexity) Keep it simple, Occam's razor, only the essentials, security not obscurity, no sophism
Prime	No risk of subfield attacks [e.g. Teske 2003, or Petit-Quisquater] Fast in software, simple pre-university math
273 bits	Well over minimum (256-10) bits needed for ECC to protect 128-bit sym. keys (AES, HMAC-SHA-256, etc.) Multiplication with just five 64-bit words (and delayed carries)
Close to 2^m	Fast and simple modular reduction [Mohan-Adiga, 1985]
5 above 2^m	Fast and simple Fermat inversion (+ fast and simple square root checking and computation)

Simple and fast Fermat inversion mod $8^{91}+5$

$y=1/x=x^{p-2}=x^{8^{91}+3} \bmod p=8^{91}+5$

```
i inv(f y,f x)
{
  i j=272;f z;
  squ(z,x);
  mul(y,x,z);
  for(;j--;) squ(z,z);
  mul(y,z,y);
  return !!cmp(y,(f){});
}
```

Comparing 8^91+5 to other fields

Field [curve]	**Better than 8^91+5**	***Worse than 8^91+5***
[P-256=secp256r1]	[NSA], used~1999, 4int64, 32B	Suite B, many symbols, (inv., sqrt., red.?), <Pollard rho,
2^255-19 [Curve25519]	[DJB], used~2005?, 4int64, 5double, 10int32, 32B, less overflow risk?	**7 symbols (8^85-19), inv.?,sqrt.?, <Pollard rho, buggy 4int64?[?]**
[K-283=sect283k1]	5 symbols: 2^283, Zigbee, >Pollard rho	Risk of subfield attacks, slower software?, complex math?
[secp256k1]	Bitcoin~200?, 4int64, 32B	Bitcoin?, many symbols, red., <Pollard rho
[Brainpool@256]	[BSI], used~2003, 4int64, 32B, random?	Slower (farther to 2^m), <Pollard rho, MANY symbols, pi, SHA
(2^127-1)^2	Faster, 32B	Risk of subfield attacks, 11 symbols, <Pollard rho, inv.?
8^95-9	>Pollard rho, mul (uint)?	Inv., sqrt., red.?, longer scalar?
9^99+4	>>Pollard rho	Slower (far to 2^m, other?)
94!-1	5 symbols, >> Pollard rho,	Slower (far to 2^m, other?), uses extra symbol '!'
9*8^96+5	Leads to CM55 curve	More symbols, slower, etc.
8^81-9 (or smaller)	Faster, <32B	<<Pollard rho: too weak for AES, inv.?, sqrt.?
Larger than 2^320	>>Pollard rho	7+ symbols, slower (cannot fit in 5int64, longer exponent)

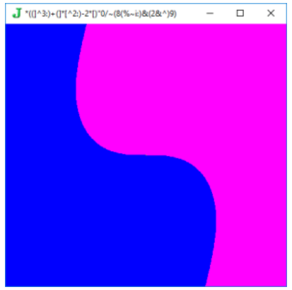
Decimal exponential complexity as an efficiency heuristic

- **Predictive** (true positive): Closer to a power of two (fast, simple) ~ shorter
  - Curve25519, base20, 6 symbols: 8^45+j, so small alt. bases fast too
- **Incomplete** (false negative): missed Curve25519, 2^263+9, Chung-Hasan, …
- **Fixable flaws** (false positives): 2^283, 9^99+4, … (easy to weed out)
- **Lucky**:
  - Base 10 gives has just **2** shortest secure and fast options 8^91+5 and 8^95-9
  - Unique prime of form 2^m+c for 240<m<320, c in {3,5,7} has 3|m, i.e. 8^91+5
  - ECC born in 1985 (little-endian 5891) , prime is 5+8^91 ☺
    - To be fair: -**19**+8^**85**

Benefits of curve equation $2y^2=x^3+x$

Feature	Benefit
Similar to $y^2=x^3-ax$ [Miller, 1985]	Essentially in first ECC paper.
Montgomery equation: $by^2=x^3+ax^2+x$	Fast doubling (P->2P) and differential addition (P-Q,P,Q)->(P+Q) 9 field multiplications per bit… [Montgomery, 1987]
Complex multiplication by i: (x,y) -> (-x,iy)	Fast: Gallant-Lambert-Vanstone multiplication, Bernstein 2-dimensional Montgomery ladder (7 field mults per bit) Compress by 1 extra bit (drop sign of x)
Similar to secp256k1	Used in BitCoin to protect high value of transactions
10 symbols: 2y^2=x^3+x	Little room for trapdoor (among CM+Montgomery equations)
Size 72n (over field 8^91+5)	Cofactor 72 resists small-subgroup attacks (+Edwards?) Prime n, ~266 bits, protects 128-bit AES against Pohlig-Hellman Speculation: further speedups? Hessian? tripling? quadrupling?
Large embedding degree	Avoids Menezes-Okamoto-Vanstone attack
Curve size not field size	Avoids Smart-Araki-Satoh-Semaev attack

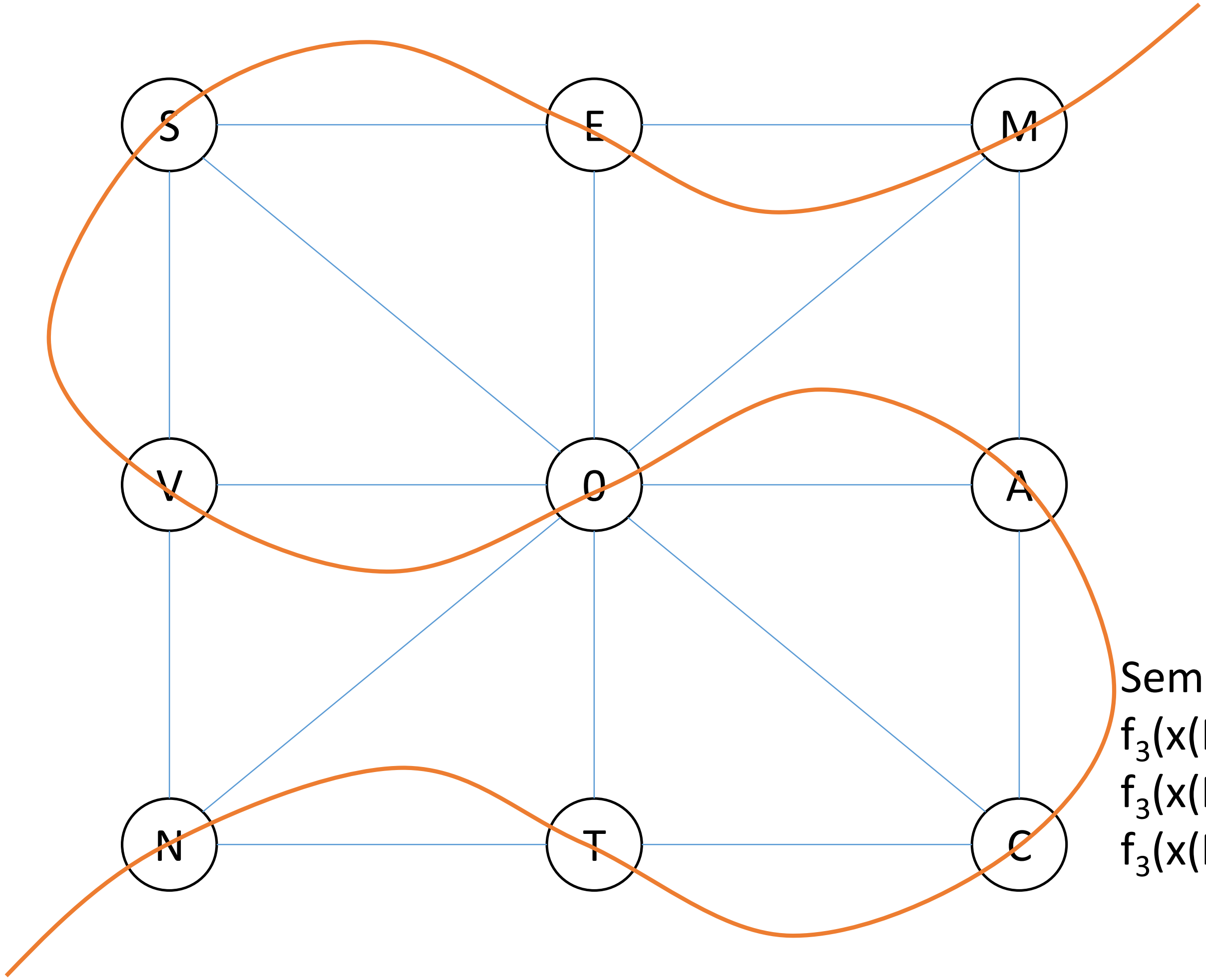
Aside: re-deriving differential addition (sketch)



$2z=x^3+xz^2$

0=(0:1:0)->(0,0)

Old x(P)->inverse slope of line through 0 and P



A = S-M

T = S+M

Semaev summation poly $f_3(-,-,-)$

$f_3(x(N),x(T),x(C)) = f_3(x(M),x(E),x(S)) = 0$

$f_3(x(N),x(A),x(C)) = f_3(x(M),x(A),x(C)) = 0$

$f_3(x(M),X,x(S))=a(X-x(S-M))(X-x(S+M))$

Curve criteria ceded by $2y^2=x^3+x$

Criterion	Adherents	Non-adherents	Benefit	Cost
Twist-secure	Curve25519	P-256, Brainpool	Securer [Bernstein] (bug-proof), (faster?)	Big curve spec, (e.g. 19+ symbols), unneeded for ephemeral DH, sigs, etc.
Cofactor 1	P256, Brainpool	Curve25519	Securer [Lim-Lee, weakly]	Slower (no Montgomery), big curve spec [expected]
Cofactor 2^m	Almost all	Hessian …	Securer [Bleichenbacher]	Extra curve spec (+?), unneeded for ephemeral DH, workarounds…
Ordinary: no fast complex multiply	P-256, Brainpool, Curve25519	Bitcoin, Koblitz (K-283), Galbraith-Lin-Scott	Securer [Miller, conjectured]	Slower, counting, riskier? (lose non-std. conjecture, isogenies similar to [Kob.-Kob.-Men.])
Randomized (j-invariant)	P-256, Brainpool	Curve25519, Bitcoin, K-283, GLS	Securer [Various, arguable]	Very BIG curve spec, riskier [proof/consensus of randomization]
Genus >=2,	Kummer	Elliptic curves	Faster?	Riskier (sub-exp. attacks?), big spec
Compact n	CM55, ???	Most	Securer?	Other criteria suffer
Tight DHP	CM55	Almost all	Securer [den Boer,…]	Big curve spec, riskier?
Cheon-safe	(New*SEC1)	Almost all	Securer [Gallant,…]	Big curve spec, riskier?

Counterarguments: Fudd and Bugs ☺

Screenshot (from Wikipedia) of *Hare Brush* , Freleng, Foster, Bonnicksen, Davis, Chiniquy, Pratt, Wyner, 1955.

Miller, 1985

Instead of using the Schoof algorithm, when searching for a good p, I have taken the following approach: Choose the curve to be:

$$E:\ y^2 = x^3 - ax$$

where a is not a perfect square. This curve has complex multiplication by $\sqrt{-1}$, and there is an exact formula for N_p (see [10]). In the case $p = 3 \bmod 4$ we have $N_p = p + 1$. This is the so-called "supersingular" case. In this case we know even more. It is well known (see [1]) that any field

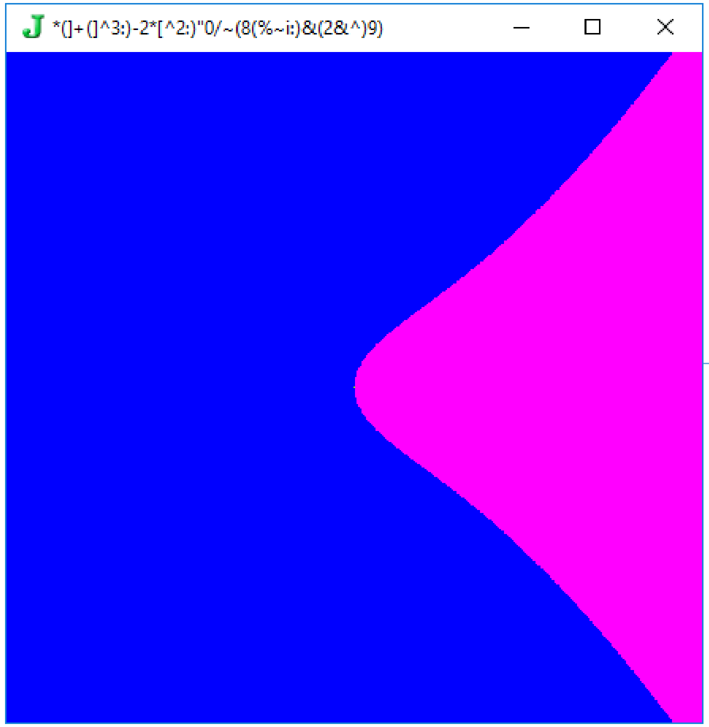
The above choice of curve was taken for convenience in calculation. However, it may be prudent to avoid curves with complex multiplication because the extra structure of these curves might somehow be used to give a better algorithm.

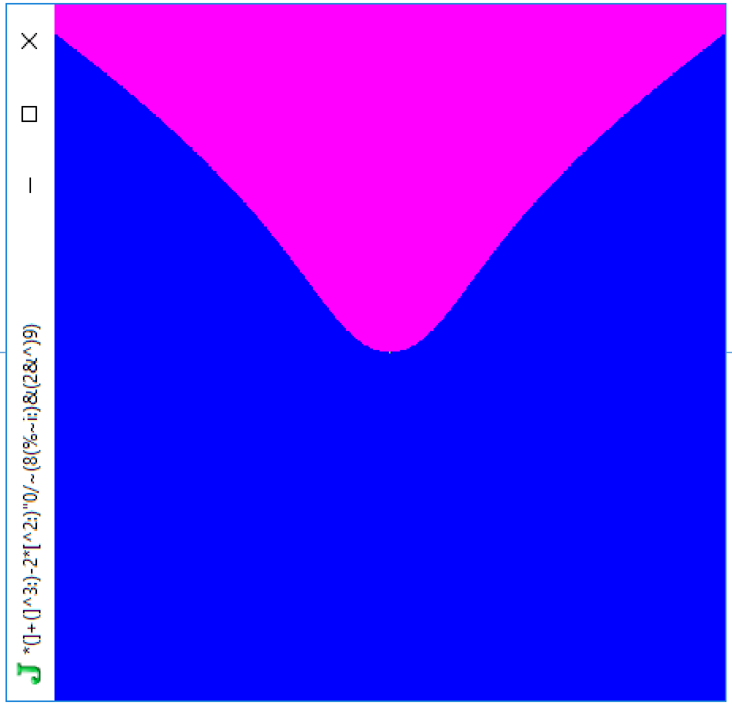
Was it "**prudent**"?

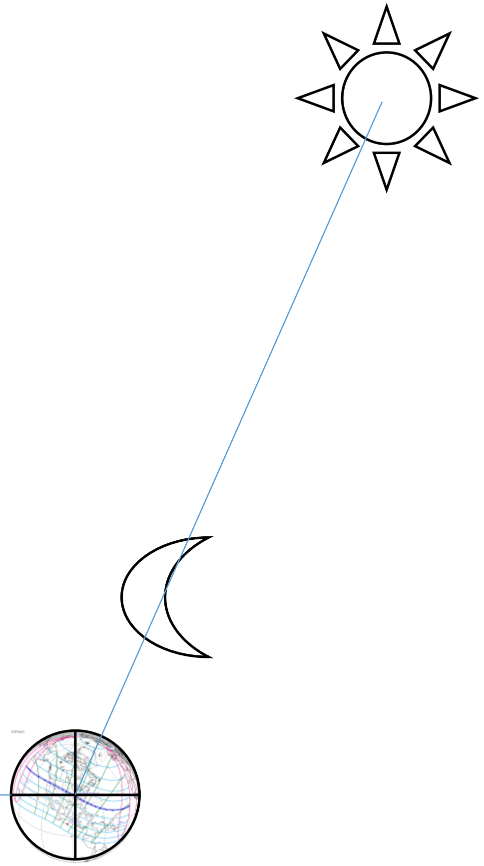
- Supersingular: YES [Menezes-Okamoto-Vanstone attack 1993]
  - Miller 8 years ahead of the curve
- Complex multiplication curves: NO (no published attacks yet, Bitcoin, qed.)
  - Prescient about a "better algorithm" ☺

Happy 32nd birthday ECC

… soon, this August?







Courtesy NASA/JPL-Caltech.