Deterministic Generation of Elliptic Curves (a.k.a. "NUMS" Curves)

Brian LaMacchia
Craig Costello

Motivation

- Reduced customer confidence in NIST-standardized curves (FIPS 186-3)
- Industry moving to Perfect Forward Secrecy (PFS) ciphersuites (e.g. ECDHE)
- We need new curves that have independently-verifiable provenance and also perform better for the standard ECC algorithms and protocols

Our Requirements (1 of 2)

- New curves must support standard security levels
  - 128-bit and 256-bit mandatory, 192-bit desired
- New curves generated deterministically from the security level
  - Rigid parameter generation for primes and curve constants
- New curves must work with the existing ECC protocol infrastructure
  - Must support standard ECDHE and ECDSA algorithms
  - Must work with TLS 1.2, X.509v3/PKIX, CMS (both for S/MIME and code signing)

Our Requirements (2 of 2)

- New curves must have good performance for both key agreement and digital signatures
- New curves must support standard EC point representations
  - Retain existing (x,y) coordinate encoding formats
- New curves must support standard group and field order bit length
  - Recommend alignment at CPU register boundary: 64-bit length alignment

Our EC Research

- Comprehensive analysis
  - Curve forms and their arithmetic
  - Prime forms
  - Performance in protocols
  - Constant-time and exception-free implementation
  - Full paper at http://eprint.iacr.org/2014/130

- Open source implementation
  - http://research.microsoft.com/en-us/projects/nums/default.aspx

Findings -- Curve Form Pros & Cons

Curve Family	Pros	Cons
Weierstrass	• Prime order • Widely deployed in existing infrastructure	• Slower than T-Edwards • Harder constant-time implementation
Montgomery	• Easier constant-time implementation • x-coordinate only	• Slower ECDHE than T-Edwards • Can't be used with ECDSA • Not prime order
Twisted Edwards	• Fastest overall performance • Easier constant-time implementation	• Not prime order

Twisted Edwards represents the best overall option

NUMS Curves -- "Nothing Up My Sleeves"

- NUMS parameter generation algorithm:
  1. Start with security level s (e.g. s = 128)
  2. Find smallest c>0 such that $p = 2^{2s} - c$ is prime and $p \equiv 3 \bmod 4$
  3. Given this p
     - For Weierstrass, find smallest |b| such that #E(GF(p)) and #E'(GF(p)) are prime, choose $\pm b$ based on smaller group order
     - For T-Edwards, find smallest d>0 such that #E(GF(p))=4q and #E'(GF(p))= 4q' where q, q' prime, q < q'
- For standard security levels, resulting primes and curves are:

Security Level	Prime (p)	Weierstrass (b) E: $y^2=x^3-3x+b$	T-Edwards (d) E: $-x^2+y^2=1+dx^2y^2$
128	$2^{256}-189$	152961	15342
192	$2^{384}-317$	-34568	333194
256	$2^{512}-569$	121243	637608

NUMS Benchmarks: Scalar Multiplication

Security Level	Prime (p)	Scalar Multiplication (in 10^3 cycles)			
		Weierstrass		T-Edwards	
		Fixed base	Variable base	Fixed base	Variable base
128	2^{256}-189	107	270	82	216
192	2^{384}-317	252	714	201	588
256	2^{512}-569	488	1504	391	1242

Results for scalar multiplication on an Intel Core i7-2600K (Sandy Bridge) processor running Linux (Ubuntu). Compilation tool: GNU GCC.

NUMS Benchmarks: ECDHE

Security Level	Prime (p)	ECDHE Cost (in 10^3 cycles)	
		Weierstrass	T-Edwards
128	2^{256}-189	379	300
192	2^{384}-317	968	791
256	2^{512}-569	1993	1638

Results for ECDHE on an Intel Core i7-2600K (Sandy Bridge) processor running Linux (Ubuntu). Compilation tool: GNU GCC.

- Gueron-Krasnov (2013): an implementation of the NIST curve P-256, computes ECDHE in 490,000 cycles
- ECDHE cost: 1 fixed base cost + 1 variable base cost + ε overhead

Recommendations to CFRG

- The requirements on Slides 3 & 4 should form the basis for defining new ECC curves for the IETF.
  - While TLS is the first group to ask for new curves, the CFRG's process and recommendations here will establish precedent for future requests from other WGs.
- Our Weierstrass-form curves are suitable "drop-in" replacements for the NIST curves that provide significantly improved performance.
- Our twisted Edwards curves provide even greater performance and are compatible with ECDHE, ECDSA, TLS 1.2, PKIX, CMS, …

Questions?

{bal,craigco}@microsoft.com