Selecting and Deploying Elliptic Curves in Security Protocols

Tolga Acar
Microsoft Research

Abstract

- The **development** and **adoption** of a cryptographic standard is a delicate endeavor with competing and conflicting actors, which becomes only harder with integration into security protocols some yet undefined
- Use of Elliptic Curves (EC) in a sliver of pervasive **security protocols**
- NIST-defined ECs, impact of new information made available in the past couple of years, and current attempts to alleviate sometimes unsubstantiated yet valid **concerns** over these curves
- An elliptic curve selection **algorithm** and its **analysis** from a performance and security perspective including rigid parameter generation, constant-time implementation, and exception-free scalar multiplication

Elliptic Curves in Cryptography

1985-1987
- Koblitz and Miller: **elliptic curves in cryptography**

2000
- Certicom: First curve standard **Standards for Efficient Cryptography**
- NIST: FIPS 186-2 **Digital Signature Standard**

2005
- ECC Brainpool: **Standard Curves and Curve Generation**

2006
- D. J. Bernstein: **Curve25519 (128-bit security)**

2013
- New York Times:
  "*the National Security Agency had written the standard and could break it*"

ECC in Protocols

- 2005 – 2006: ECC in TLS, Kerberos, CMS
- Cross-company interoperability with TLS
- NIST prime curves, only
- No GF(2) curves
- ECDH and ECDSA, only
- No point compression
- Named curves in X.509 certificates and CMS

What about Dual EC DRBG?

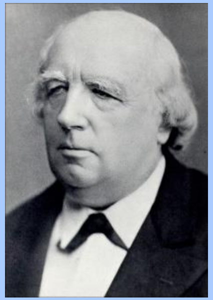
- φ : prime curve → integers
- φ $(x,y) = x$
- P, Q points on the curve (per SP800-90)
- Equations
  - r_i = φ(s_iP)
  - t_i = φ(r_iQ) Output
  - s_{i+1} = φ(r_iP) State update
  - Output 16 bits of t_i
- Is $e \in Z_q$ known, s.t. eQ = P

Forms of Elliptic Curves

Weierstrass curves

$$y^2 = x^3 + ax + b$$

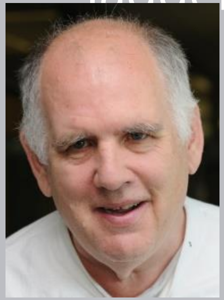
- Most general form
- Prime order possible
- Exceptions in group law
- NIST and Brainpool curves



Montgomery curves

$$By^2 = x^3 + Ax^2 + x$$

- Subset of curves
- Not prime order
- Montgomery ladder



Twisted Edwards curves

$$ax^2 + y^2 = 1 + dx^4y^4$$

- Subset of curves
- Not prime order
- Fastest arithmetic
- Some have complete group law

Problems

- New Elliptic Curves research and aging NIST curves
- Reduced customer confidence in NIST elliptic curves
- Need for efficient Perfect Forward Secrecy (PFS) to meet the demand
- Need for efficient ECDSA

Requirements

- Support existing protocols and standards
  - IETF: TLS *and* other protocols (IPsec, S/MIME (CMS), OAuth)
  - W3C: Web Crypto, XML-Dsig
- Support standard algorithms
  - ECDHE and ECDSA (with the same curve form?)
- Support standard security levels
  - 128 and 256 bit, 192 bit optional

Requirements

- Support standard EC point representation
  - Retain existing (x,y) coordinate encoding formats
  - Compression?
- Rigid parameter generation
  - Primes and curve constants
- Support standard group and field order bit length
  - Recommend alignment at CPU register boundary: 64-bit length alignment

Motivation

- Public distrust against everything touched by NIST (justified?)
- Dan Bernstein & Tanja Lange: *Security dangers of the NIST curves*
- Bruce Schneier: "*I no longer trust the constants. I believe the NSA has manipulated them through their relationships with industry*"

NIST curves are old curves designed for 32-bit platforms.
Many new techniques since 2000:

1) Faster modular arithmetic
2) Faster curve arithmetic (twisted Edwards)
3) Constant-time algorithms to protect against various types of side-channel attacks

Other Motivations

Nothing wrong from a mathematical perspective

Engineering considerations

- o more (complex) implementations
  - arithmetic on Montgomery *and* Edwards curves
  - conversion code
- o how does one transmit points in protocols?

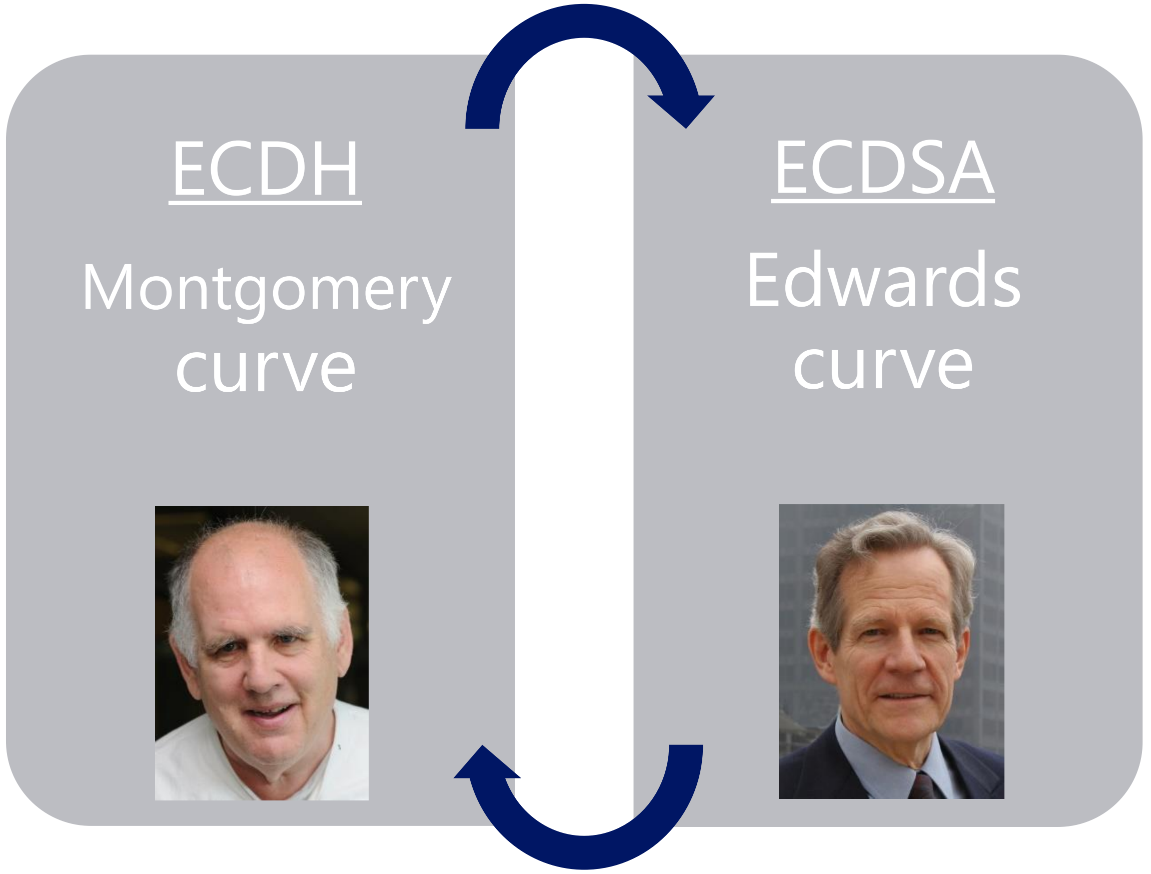
We choose a different approach

Goals

- rigorously analyze curves from a security and efficiency perspective
- abandoning a set of standard curves demands a judicious selection of new curves

Example: Curve25519

- ✓ This is a Montgomery curve (efficient plain **ECDH**)
- ▪ For **ECDHE** with **PFS:** part of the computation on Edwards, other part on Montgomery curve
- ▪ For **ECDSA**: on Edwards



Our Research

- Comprehensive analysis
  - Curve forms and their arithmetic
  - Prime forms
  - Performance in protocols
  - Constant-time and exception-free implementation
- Implementation available
  - http://research.microsoft.com/en-us/projects/nums/default.aspx

NUMS Curves
Nothing Up My Sleeves

- Rigidly generated primes and constants
- Simplified parameter generation criteria
  - Largest prime
  - Smallest curve constant
- Match standard security levels

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

NUMS Curves -- Nothing Up My Sleeves

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

New curve selection

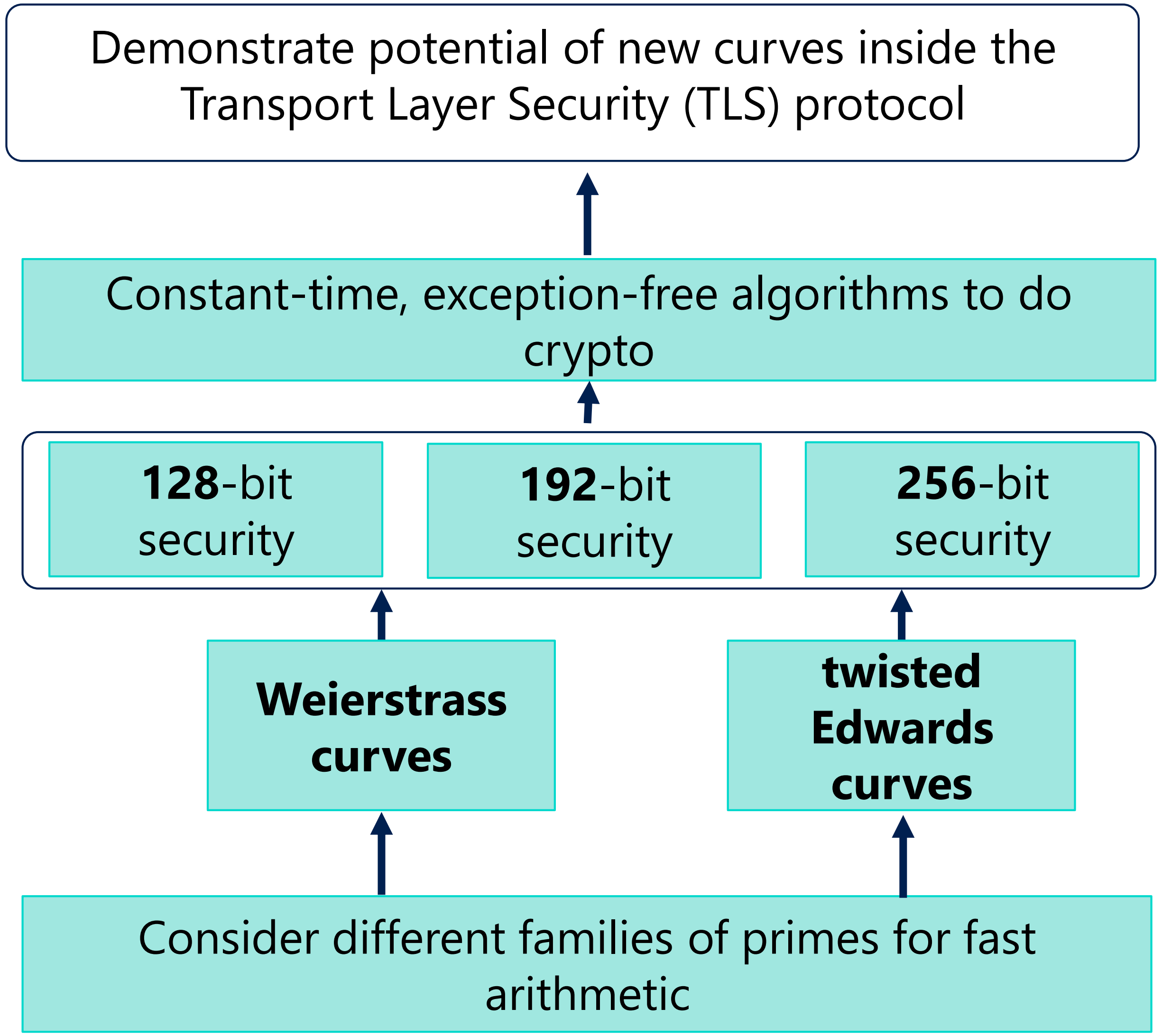
- Faster arithmetic compared to NIST primes
- Designed for arithmetic fast on both *embedded devices* (8-bit) and future *high-end platforms* (128-bit?)
- All curves are chosen deterministically after fixing security/efficiency constraints
  ➡ no room for manipulation?

Weierstrass

- fully backwards compatible with current software implementations (NIST curves)

Twisted Edwards curve

- fastest curve arithmetic
- no conversions
- no Montgomery curves used


Demonstrate potential of new curves inside the Transport Layer Security (TLS) protocol
Constant-time, exception-free algorithms to do crypto
128-bit security
192-bit security
256-bit security
Weierstrass curves
twisted Edwards curves
Consider different families of primes for fast arithmetic

Results

new Weierstrass	versus NIST curves	$\geq$ 1.4 times faster
new Edwards	versus NIST curves	$\geq$ 1.9 times faster

	NIST	Brainpool	Curve25519	New Weierstrass	New twisted Edwards
Backdoors	Unknown	Most likely not	Deterministic	Deterministic	Deterministic
ECDLP Security	Best	Best	Good	Best	Good
Efficiency	Lower	Lower	Highest	High	Highest

Curve Forms

	pros	cons	open questions
Weierstrass	Prime order, Widely deployed, General representation	Slower than TE, Cumbersome formulas, Harder constant time implementations	Keep in case?
Montgomery	Compact formulae, x coordinate only	Lower performance for DHE, No signatures, Limited deployment	Composite order concerns valid?
Twisted Edwards	Fastest, One set of simple formulas, Easier constant time implementations	Limited deployment	Composite order concerns valid?

Twisted Edwards represents the best trade-off?

NUMS Benchmarks: ECDHE

Security	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.

- Results on Windows OS are slightly slower
- Gueron-Krasnov 2013: an implementation of the NIST curve P-256 computes ECDHE in 490,000 cycles (in constant time?)
- ECDHE cost: variable base + fixed base cost

NIST VCAT Report

- Bart Preneel (excerpts from ECC section)
  - In the interest of transparency, it would have been desirable that the algorithm and full source code for the generation of these curves would have been made public, so that experts can verify the selection criteria and the outcome.
  - NIST should consider the publication of a standard algorithm and corresponding software to generate additional elliptic curves and should consider to use this tool to also publish some new curves.

Conclusion

- New curves are coming
- PFS is a big driver

- Ask: Participate in the IRTF CFRG