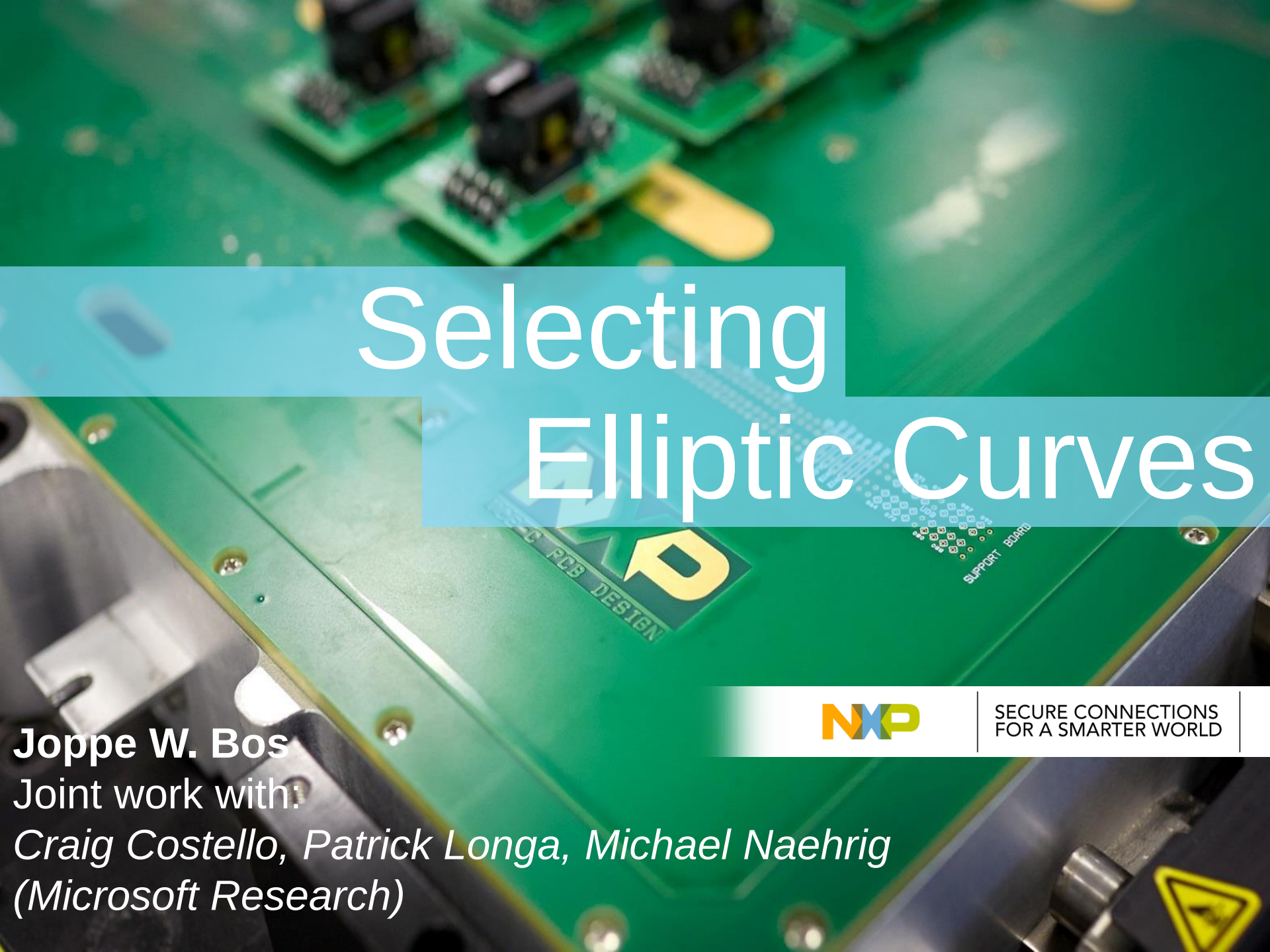


Selecting Elliptic Curves

NXP SECURE CONNECTIONS FOR A SMARTER WORLD

Joppe W. Bos
Joint work with:
Craig Costello, Patrick Longa, Michael Naehrig (Microsoft Research)

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 only)

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

2014
- IETF asks for recommendations for new elliptic curves in TLS

Motivation

SECURE CONNECTIONS
FOR A SMARTER WORLD

Motivation

- Public distrust against everything touched by NIST
- 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*"

Motivation

- Public distrust against everything touched by NIST
- 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

Forms of Elliptic Curves

NXP

SECURE CONNECTIONS FOR A SMARTER WORLD

Forms of Elliptic Curves

Weierstrass curves

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

- Most general form, all elliptic curves can be written as a Weierstrass curve

Forms of Elliptic Curves

Weierstrass curves

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

- Most general form, all elliptic curves can be written as a Weierstrass curve

"*One form to rule them all*"
Convenience
One API to implement them all

Forms of Elliptic Curves

Weierstrass curves

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

- Most general form, all elliptic curves can be written as a Weierstrass curve
- Prime order n possible

Forms of Elliptic Curves

Weierstrass curves

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

- Most general form, all elliptic curves can be written as a Weierstrass curve
- Prime order n possible

✓ Full security
✓ No small subgroups

Point validation

- Check $Q \neq \mathcal{O}$
- Check Q is on the curve
- Check $nQ = \mathcal{O}$

This can be omitted!

Forms of Elliptic Curves

Weierstrass curves

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

- Most general form, all elliptic curves can be written as a Weierstrass curve
- Prime order n possible
- Exceptions in group law

Forms of Elliptic Curves

Weierstrass curves

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

- Most general form, all elliptic curves can be written as a Weierstrass curve
- Prime order n possible
- Exceptions in group law

$P + Q$, $P + \mathcal{O}$
$P + P$, $P + (-P)$
might require different formula / implementations

Complete formula exists: very slow

Forms of Elliptic Curves

Weierstrass curves

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

- Most general form, all elliptic curves can be written as a Weierstrass curve
- Prime order n possible
- Exceptions in group law
- NIST and Brainpool curves are prime-order Weierstrass curves with $a = -3$.

Forms of Elliptic Curves

Weierstrass curves

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

- Most general form, all elliptic curves can be written as a Weierstrass curve
- Prime order n possible
- Exceptions in group law
- NIST and Brainpool curves are prime-order Weierstrass curves with $a = -3$.

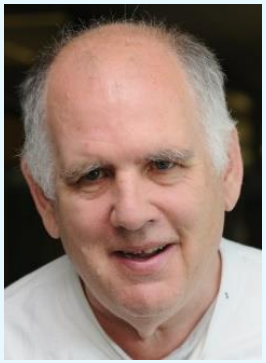
- ✓ Full security
- ✓ No small subgroups
- ✓ $a = -3$ for efficiency
- ✓ How were these curves chosen?

Forms of Elliptic Curves

Montgomery curves

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

- Subset of all curves



P. L. Montgomery (1987). *Speeding the Pollard and Elliptic Curve Methods of Factorization*

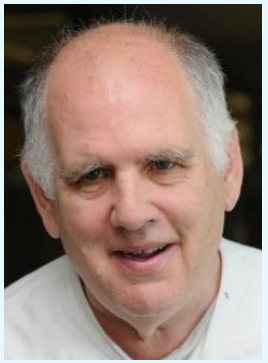
Forms of Elliptic Curves

Montgomery curves

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

- Subset of all curves

- Not all curves can be written as a Montgomery curve
- How many? Roughly 40%



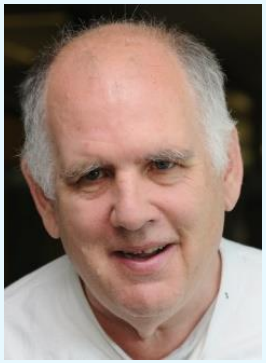
P. L. Montgomery (1987). *Speeding the Pollard and Elliptic Curve Methods of Factorization*

Forms of Elliptic Curves

Montgomery curves

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

- Subset of all curves
- **Not** prime order



P. L. Montgomery (1987). *Speeding the Pollard and Elliptic Curve Methods of Factorization*

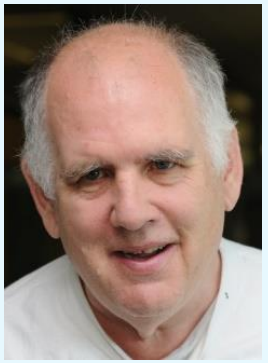
Forms of Elliptic Curves

Montgomery curves

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

- Subset of all curves
- **Not** prime order

- The order of a Montgomery curve is divisible by at least 4
- Lose <u>at least</u> one bit of security
- Small subgroup attacks



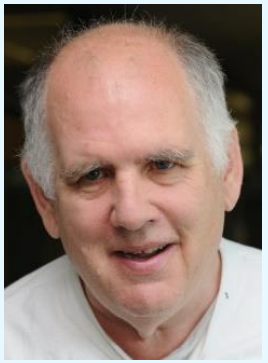
P. L. Montgomery (1987). *Speeding the Pollard and Elliptic Curve Methods of Factorization*

Forms of Elliptic Curves

Montgomery curves

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

- Subset of all curves
- **Not** prime order
- Montgomery ladder



P. L. Montgomery (1987). *Speeding the Pollard and Elliptic Curve Methods of Factorization*

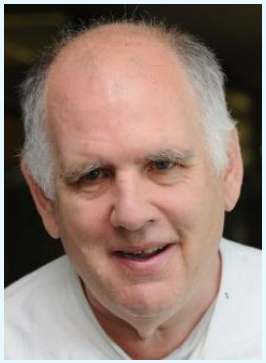
Forms of Elliptic Curves

Montgomery curves

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

- Subset of all curves
- **Not** prime order
- Montgomery ladder

✓ Very efficient **differential addition**
Given $P, Q, P - Q$ compute $P + Q$
✓ Inherently, constant-time



P. L. Montgomery (1987). *Speeding the Pollard and Elliptic Curve Methods of Factorization*

Forms of Elliptic Curves

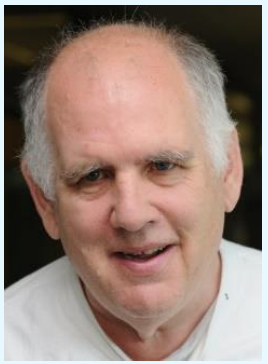
Montgomery curves

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

- Subset of all curves
- **Not** prime order
- Montgomery ladder

✓ Very efficient **differential addition**
Given $P, Q, P - Q$ compute $P + Q$
✓ Inherently, constant-time

- No **point addition**
  We can do ECDH, no ECDSA signature verification
- Single-coordinate system
  Twist-security comes into play



P. L. Montgomery (1987). *Speeding the Pollard and Elliptic Curve Methods of Factorization*

Forms of Elliptic Curves

Twisted Edwards curves

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

H. M. Edwards (2007), *A normal form for elliptic curves*
D. J. Bernstein, T. Lange (2007), *Faster addition and doubling on elliptic curves*

Forms of Elliptic Curves

Twisted Edwards curves

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

- Subset of all curves

- Not all curves can be written as a twisted Edwards curve
- How many? Roughly 40%

H. M. Edwards (2007), *A normal form for elliptic curves*
D. J. Bernstein, T. Lange (2007), *Faster addition and doubling on elliptic curves*

Forms of Elliptic Curves

Twisted Edwards curves

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

- Subset of all curves
- Not prime order

- The order of a twisted Edwards curve is divisible by at least 4
- Lose <u>at least</u> one bit of security
- Small subgroup attacks

H. M. Edwards (2007), *A normal form for elliptic curves*
D. J. Bernstein, T. Lange (2007), *Faster addition and doubling on elliptic curves*

Forms of Elliptic Curves

Twisted Edwards curves

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

- Subset of all curves
- Not prime order
- Fastest arithmetic

H. M. Edwards (2007), *A normal form for elliptic curves*
D. J. Bernstein, T. Lange (2007), *Faster addition and doubling on elliptic curves*

Forms of Elliptic Curves

Twisted Edwards curves

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

- Subset of all curves
- Not prime order
- Fastest arithmetic

✓ When $a = -1$, most efficient elliptic curve arithmetic known

H. Hisil, K. K. Wong, G. Carter, E. Dawson (2008), *Twisted Edwards Curves Revisited*

H. M. Edwards (2007), *A normal form for elliptic curves*
D. J. Bernstein, T. Lange (2007), *Faster addition and doubling on elliptic curves*

Forms of Elliptic Curves

Twisted Edwards curves

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

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

H. M. Edwards (2007), *A normal form for elliptic curves*
D. J. Bernstein, T. Lange (2007), *Faster addition and doubling on elliptic curves*

Forms of Elliptic Curves

Twisted Edwards curves

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

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

Efficient complete group law:

$$P + Q, \qquad P + \mathcal{O}$$
$$P + P, \qquad P + (-P)$$

However, slower than dedicated arithmetic

H. M. Edwards (2007), *A normal form for elliptic curves*
D. J. Bernstein, T. Lange (2007), *Faster addition and doubling on elliptic curves*

Forms of Elliptic Curves

Weierstrass curves

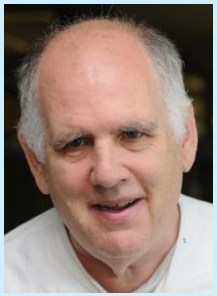
$$y^2 = x^3 + ax + b$$

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

Montgomery curves

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

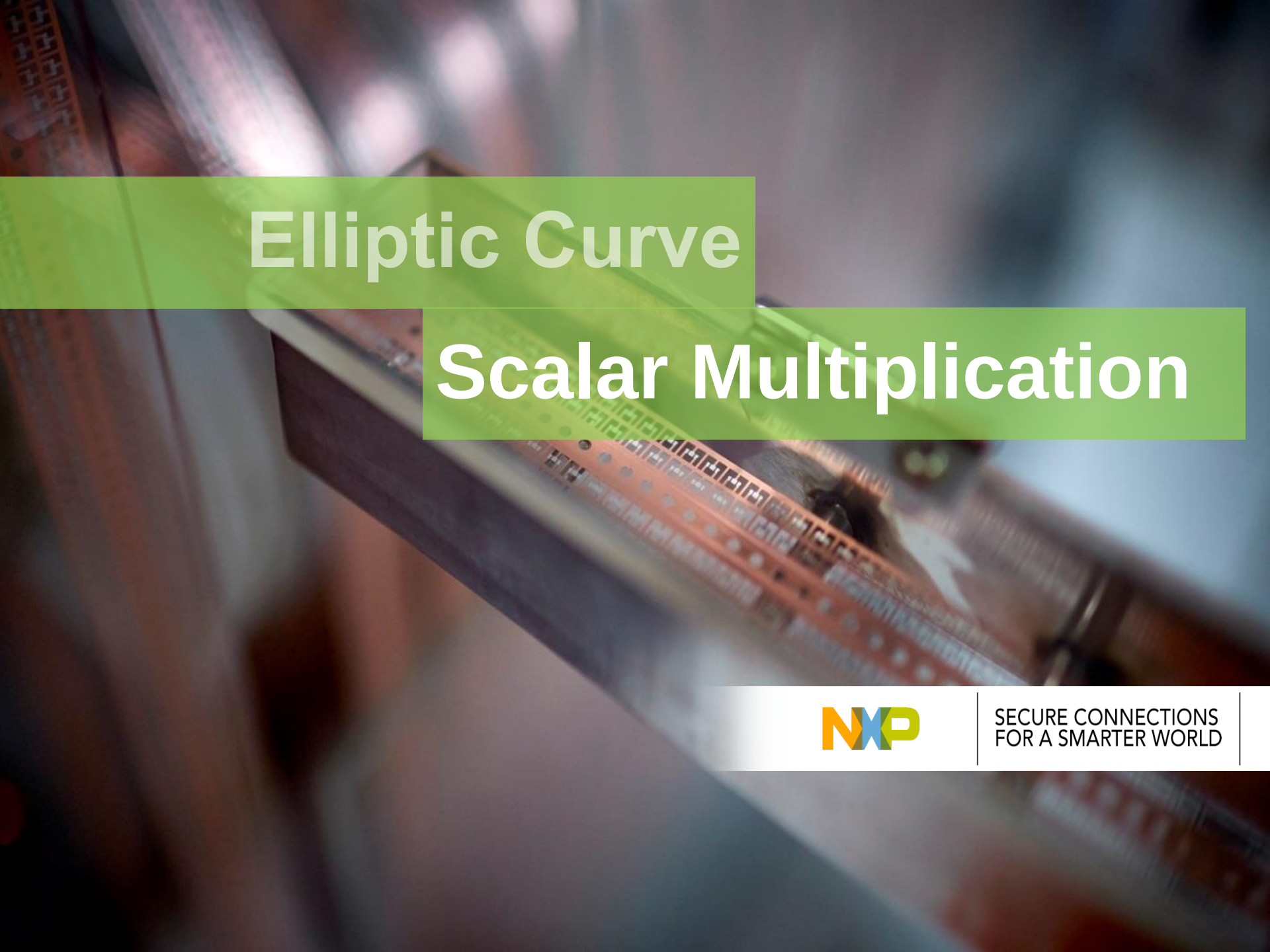
- Subset of curves
- Not prime order
- Montgomery ladder



Twisted Edwards curves

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

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


Elliptic Curve
Scalar Multiplication
SECURE CONNECTIONS
FOR A SMARTER WORLD

Elliptic Curve Scalar Multiplication

Core of all ECC based protocols

Given $P \in E(\mathbf{F}_p)$ of order n and $k \in \mathbf{Z}_{>0}$ compute kP

scalar multiplication	k	P
Variable-base	Random	Random
Fixed-base	Random	Fixed

Fixed-base: pre-computation can lead to performance speedups

Elliptic Curve Scalar Multiplication

Core of all ECC based protocols

Given $\mathrm{P} \in E(\mathbf{F}_p)$ of order n and $k \in \mathbf{Z}_{>0}$ compute kP

scalar multiplication	k	P
Variable-base	Random	Random
Fixed-base	Random	Fixed

Fixed-base: pre-computation can lead to performance speedups

Most popular ECC protocols (used in e.g. TLS):

- ECDSA signature generation = fixed-base scalar multiplication
- ECDSA signature verification = double-scalar multiplication
  - compute $k_1 P + k_2 Q$ such that k_1 and k_2 random, P fixed and Q random
- ECDHE = variable + fixed base scalar multiplication
  - Ephemeral ECDH (temporary session keys)
  - Perfect forward secrecy
  - Compute $k_1(k_2 P)$ such that k_1, k_2 random and P fixed

Example: Curve25519

Cryptographic curve providing 128-bit security

Montgomery Curve

$$y^2 = x^3 + 486662x^2 + x$$

Twisted Edwards curve

$$x^2 + y^2 = 1 + \frac{121665}{121666}x^2y^2$$

Fast ECDH →
Montgomery ladder

Fast ECDSA →
twisted Edwards arithmetic

D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, B.-Y. Yang (2011), *High-speed high-security signatures*
D. J. Bernstein (2006), *Curve25519: New Diffie-Hellman Speed Records*

Example: Curve25519

Cryptographic curve providing 128-bit security

Montgomery Curve

$$y^2 = x^3 + 486662x^2 + x$$

Twisted Edwards curve

$$x^2 + y^2 = 1 + \frac{121665}{121666}x^2y^2$$

Fast ECDH →
Montgomery ladder

Fast ECDSA →
twisted Edwards arithmetic

Practical Issues

How to compute ECDHE?

1) Montgomery ladder, Montgomery ladder
   (fixed-base) (variable-base)
2) Twisted Edwards, convert, Montgomery ladder
   (fixed base) (variable-base)
3) Twisted Edwards, twisted Edwards
   (fixed base) (variable-base)

D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, B.-Y. Yang (2011), *High-speed high-security signatures*
D. J. Bernstein (2006), *Curve25519: New Diffie-Hellman Speed Records*

Example: Curve25519

Cryptographic curve providing 128-bit security

Montgomery Curve

$$y^2 = x^3 + 486662x^2 + x$$

Twisted Edwards curve

$$x^2 + y^2 = 1 + \frac{121665}{121666}x^2y^2$$

Fast ECDH →
Montgomery ladder

Fast ECDSA →
twisted Edwards arithmetic

Practical Issues

How to compute ECDHE?

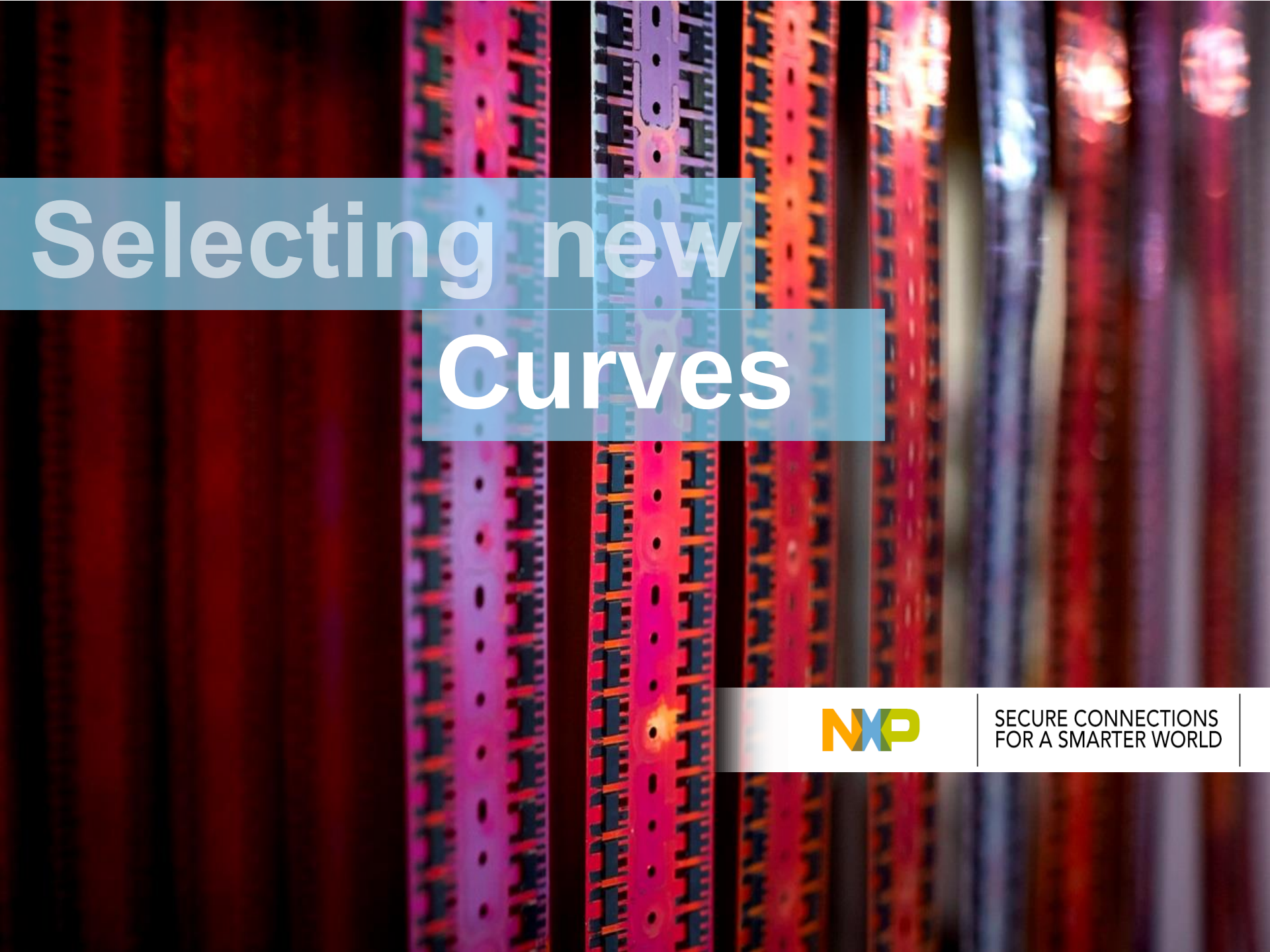
1) Montgomery ladder, Montgomery ladder
   (fixed-base) (variable-base)
2) Twisted Edwards, convert, Montgomery ladder
   (fixed base) (variable-base)
3) Twisted Edwards, twisted Edwards
   (fixed base) (variable-base)

How to transfer points?

Coordinate system:

- Montgomery
- Twisted Edwards
- Weierstrass

D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, B.-Y. Yang (2011), *High-speed high-security signatures*
D. J. Bernstein (2006), *Curve25519: New Diffie-Hellman Speed Records*


Selecting new
Curves
SECURE CONNECTIONS
FOR A SMARTER WORLD

Deterministic Prime Selection

Minimize room for manipulation

Given an even security level s do

$$c = 0; \text{do } p = 2^{2s} - c\text{++}; \text{while}(p \neq \text{prime or } p \not\equiv 3 \bmod 4);$$

Deterministic Prime Selection

Minimize room for manipulation

Given an even security level s do

$$c = 0; \text{do } p = 2^{2s} - c\text{++}; \text{while}(p \neq \text{prime or } p \not\equiv 3 \bmod 4);$$

Security level	Prime
128	$2^{256} - 189$
192	$2^{384} - 317$
256	$2^{512} - 569$

Deterministic Prime Selection

Minimize room for manipulation

Given an even security level s do

$$c = 0; \text{do } p = 2^{2s} - c\text{++}; \text{while}(p \neq \text{prime or } p \not\equiv 3 \bmod 4);$$

Why special prime shape? Performance!

$$0 \leq x, y < 2^{2s} - c$$

$$x \cdot y \bmod 2^{2s} - c$$

Security level	Prime
128	$2^{256} - 189$
192	$2^{384} - 317$
256	$2^{512} - 569$

$$\begin{aligned} x \cdot y &= z_h\ 2^{2s} + z_\ell \\ &\equiv z_h\ 2^{2s} + z_\ell - z_h(\ 2^{2s} - c) \pmod{2^{2s} - c} \\ &= z_\ell + c \cdot z_h \end{aligned}$$

Deterministic Prime Selection

Minimize room for manipulation

Given an even security level s do

$$c = 0; \text{do } p = 2^{2s} - c\text{++}; \text{while}(p \neq \text{prime or } p \not\equiv 3 \bmod 4);$$

Why special prime shape? Performance!

$$0 \leq x, y < 2^{2s} - c$$

$$x \cdot y \bmod 2^{2s} - c$$

Security level	Prime
128	$2^{256} - 189$
192	$2^{384} - 317$
256	$2^{512} - 569$

$$\begin{aligned} x \cdot y &= z_h\ 2^{2s} + z_\ell \\ &\equiv z_h\ 2^{2s} + z_\ell - z_h(\ 2^{2s} - c) \pmod{2^{2s} - c} \\ &= z_\ell + c \cdot z_h \end{aligned}$$

Constant time modular inversion

$$a^{-1} \equiv a^{p-2} \pmod{p}$$

Constant time modular square root

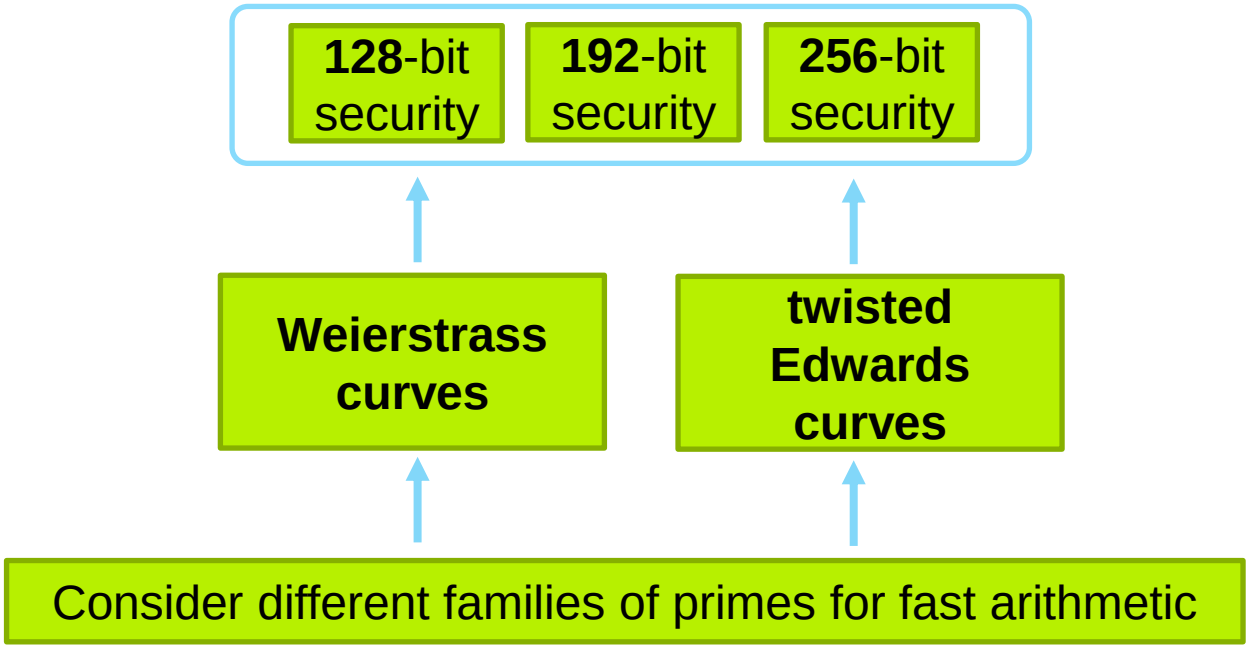
$$\sqrt{a} \equiv a^{(p+1)/4} \pmod{p}$$

Deterministic Curve Selection

	Curve	Quadratic twist	parameter	order
Weierstrass	$y^2 = x^3 - 3x + b$	$y^2 = x^3 - 3x - b$	$b \in \mathbf{F}_p \backslash \{\pm 2, 0\}$	prime
Twisted Edwards	$-x^2 + y^2 = 1 + dx^2y^2$	$-x^2 + y^2 = 1 + {}^1/_d\, x^2y^2$	$d \in \mathbf{F}_p \backslash \{0, 1\}$	$4 \times$prime

Deterministic Curve Selection

	Curve	Quadratic twist	parameter	order
Weierstrass	$y^2 = x^3 - 3x + b$	$y^2 = x^3 - 3x - b$	$b \in \mathbf{F}_p \backslash \{\pm 2, 0\}$	prime
Twisted Edwards	$-x^2 + y^2 = 1 + dx^2y^2$	$-x^2 + y^2 = 1 + {}^1/_d\, x^2y^2$	$d \in \mathbf{F}_p \backslash \{0, 1\}$	$4 \times$prime



Computing scalar multiplications

We have curves defined over these primes.

How to compute ECSM?

- Dedicated formulas
- Complete formulas

What formula should be used in practice?

Complete

- Works for everything
- No exceptions possible
- Performance price

Dedicated

- Faster
- Analyze exceptions

Computing scalar multiplications

We have curves defined over these primes.

How to compute ECSM?

- Dedicated formulas
- Complete formulas

What formula should be used in practice?

Complete

- Works for everything
- No exceptions possible
- Performance price

Dedicated

- Faster
- Analyze exceptions

Weierstrass	Dedicated	Complete	New "Complete"
double	4**M** + 4**S**		
jac + aff → jac	7**M** + 4**S**		8**M** + 3**S**
jac + jac → jac	11**M** + 5**S**	25**M**	12**M** + 4**S**

Twisted Edwards	Dedicated	Complete
double	4**M** + 3**S**	
proj + aff → proj	7**M**	8**M**
proj + proj → proj	8**M**	9**M**

Algorithm 19 Complete (projective) addition using masking and Jacobian coordinates on prime-order Weierstrass curves E_b.

Input: $P, Q \in E_b(\mathbf{F}_p)$ such that $P = (X_1, Y_1, Z_1)$ and $Q = (X_2, Y_2, Z_2)$ are in Jacobian coordinates.

Output: $R = P + Q \in E_b(\mathbf{F}_p)$ in Jacobian coordinates. Computations marked with [*] are implemented in constant time using masking.

1. $T[0] = \mathcal{O}$ $\quad \{T[i] = (\tilde{X}_i, \tilde{Y}_i, \tilde{Z}_i)$ for $0 \le i < 5\}$
2. $T[1] = Q$
3. $T[4] = P$
4. $t_2 = Z_1^2$
5. $t_3 = Z_1 \times t_2$
6. $t_1 = X_2 \times t_2$
7. $t_4 = Y_2 \times t_3$
8. $t_3 = Z_2^2$
9. $t_5 = Z_2 \times t_3$
10. $t_7 = X_1 \times t_3$
11. $t_8 = Y_1 \times t_5$
12. $t_1 = t_1 - t_7$
13. $t_4 = t_4 - t_8$
14. index $= 3$
15. **if** $t_1 = 0$ **then** [*]
16. index $= 0$ $\quad \{R = \mathcal{O}\}$
17. **if** $t_4 = 0$ **then** index $= 2$ $\quad \{R = 2P\}$ [*]
18. **if** $P = \mathcal{O}$ **then** index $= 1$ $\quad \{R = Q\}$ [*]
19. **if** $Q = \mathcal{O}$ **then** index $= 4$ $\quad \{R = P\}$ [*]
20. mask $= 0$
21. **if** index $= 3$ **then** mask $= 1$
    {case $P + Q$, else any other case} [*]
22. $t_3 = X_1 + t_2$
23. $t_6 = X_1 - t_2$
24. **if** mask $= 0$ **then** $t_2 = Y_1$ **else** $t_2 = t_1$ [*]
25. $t_5 = t_2^2$
26. **if** mask $= 0$ **then** $t_7 = X_1$ [*]
27. $t_1 = t_5 \times t_7$
28. $\tilde{Z}_2 = Z_1 \times t_2$
29. $\tilde{Z}_3 = Z_2 \times \tilde{Z}_2$
30. **if** mask $\neq 0$ **then** $t_3 = t_2$ [*]
31. **if** mask $\neq 0$ **then** $t_6 = t_5$ [*]
32. $t_2 = t_3 \times t_6$
33. $t_3 = t_2/2$
34. $t_3 = t_2 + t_3$
35. **if** mask $\neq 0$ **then** $t_3 = t_4$ [*]
36. $t_4 = t_3^2$
37. $t_4 = t_4 - t_1$
38. $\tilde{X}_2 = t_4 - t_1$
39. $\tilde{X}_3 = \tilde{X}_2 - t_2$
40. **if** mask $= 0$ **then** $t_4 = \tilde{X}_2$ **else** $t_4 = \tilde{X}_3$ [*]
41. $t_1 = t_1 - t_4$
42. $t_4 = t_3 \times t_1$
43. **if** mask $= 0$ **then** $t_1 = t_5$ **else** $t_1 = t_8$ [*]
44. **if** mask $= 0$ **then** $t_2 = t_5$ [*]
45. $t_3 = t_1 \times t_2$
46. $\tilde{Y}_2 = t_4 - t_3$
47. $\tilde{Y}_3 = \tilde{Y}_2$
48. $R = T[\text{index}]$ $\quad (= (\tilde{X}_{\text{index}}, \tilde{Y}_{\text{index}}, \tilde{Z}_{\text{index}}))$ [*]
49. **return** R

Computing scalar multiplications

Use dedicated formulas when possible
Proof that no exceptions can occur

$P + P, \quad P + \mathcal{O}$
$\mathcal{O} + P, \quad P + (-P)$

Weierstrass

Variable base

- Given a window size $w, 2 \leq w < 10$.
- Scalar recoding to odd non-zero digits $\{\pm 1, \pm 3, \ldots, \pm(2^{w-1} - 1)\}$
- Compute ECSM with dedicated point additions
- Except the last addition which should be complete

Computing scalar multiplications

Use dedicated formulas when possible
Proof that no exceptions can occur

$P + P$, $P + \mathcal{O}$
$\mathcal{O} + P$, $P + (-P)$

Twisted Edwards

Variable base

- Given a window size $w, 2 \leq w < 10$.
- Clear torsion $Q = 4P$
- Validation: Check $Q \neq \mathcal{O}$
- Scalar recoding to odd non-zero digits $\{\pm 1, \pm 3, \ldots, \pm(2^{w-1} - 1)\}$
- Compute ECSM with dedicated point additions
- Except the last addition which should be complete

Computing scalar multiplications

Use dedicated formulas when possible
Proof that no exceptions can occur

$P + P, \quad P + \mathcal{O}$
$\mathcal{O} + P, \quad P + (-P)$

Weierstrass / Twisted Edwards

Fixed base

Offline

- Clear torsion $Q = 4P$
- Validation: Check $Q \neq \mathcal{O}$
- Precompute $v > 0$ different tables of 2^{w-1} points each

Online

- Compute ECSM with complete point additions only

Computing scalar multiplications

Use dedicated formulas when possible
Proof that no exceptions can occur

$P + P, \quad P + \mathcal{O}$
$\mathcal{O} + P, \quad P + (-P)$

Weierstrass / Twisted Edwards

Fixed base

Offline

- Clear torsion $Q = 4P$
- Validation: Check $Q \neq \mathcal{O}$
- Precompute $v > 0$ different tables of 2^{w-1} points each

Online

- Compute ECSM with complete point additions only

Are other fixed base implementations correct for all inputs?

Our Implementation Choices

Performance

✓ Use dedicated formulas when possible

Security

✓ Use only algorithms which can be implemented in constant-time (branch-free)

Weierstrass	***Twisted Edwards***
✓ Backward compatible with existing APIs implementing NIST curves	✓ Use Edwards arithmetic only (no Montgomery ladder)
	✓ Reduce implementation burden and code size
✓ Highest security	✓ Performance at least as fast


Performance
Results
SECURE CONNECTIONS
FOR A SMARTER WORLD

Performance Results

Costs estimates for the TLS handshake using the ECDHE-ECDSA cipher suite on a 3.4GHz Intel Core i7-2600 Sandy Bridge processor

128-bit security	ECDHE	ECDSA sign	ECDSA ver	Constant time
NIST P-256	490	90 (150 KB)	530	?
numsp256d1	379	107 (6 KB)	288 (2 KB)	✓

S. Gueron, V. Krasnov (2013), *Fast prime field elliptic curve cryptography with 256 bit primes*

Performance Results

Costs estimates for the TLS handshake using the ECDHE-ECDSA cipher suite on a 3.4GHz Intel Core i7-2600 Sandy Bridge processor

128-bit security	ECDHE	ECDSA sign	ECDSA ver	Constant time
NIST P-256	490	90 (150 KB)	530	?
numsp256d1	379	107 (6 KB)	288 (2 KB)	✓

Old Weierstrass versus New Weierstrass

- **ECDHE** 1.3x faster
- **ECDSA sign** 1.2x slower, 25x reduced look-up table, more secure: side-channel resistant + cache-attack resistant
- **ECDSA ver** 1.8x faster

S. Gueron, V. Krasnov (2013), *Fast prime field elliptic curve cryptography with 256 bit primes*

Performance Results

128-bit security	ECDHE	ECDSA sign	ECDSA ver	Constant time
NIST P-256	490	90 (150 KB)	530	?
numsp256d1	379	107 (6 KB)	288 (2 KB)	✓
Curve25519	398*	69 (30 KB)	225 (3.8 KB)	✓
numsp256t1	300	82 (9 KB)	231 (3 KB)	✓

Twisted Edwards versus twisted Edwards

- **ECDHE** 1.3x faster
- **ECDSA sign** 1.2x slower, 3x reduced look-up table, curve25519 implements EdDSA (not ECDSA)
  ECDSA ver 1.03x slower

S. Gueron, V. Krasnov (2013), *Fast prime field elliptic curve cryptography with 256 bit primes*
D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, B.-Y. Yang (2011), *High-speed high-security signatures*
D. J. Bernstein (2006), *Curve25519: New Diffie-Hellman Speed Records*

Performance Results

128-bit security	ECDHE	ECDSA sign	ECDSA ver	Constant time
NIST P-256	490	90 (150 KB)	530	?
numsp256d1	379	107 (6 KB)	288 (2 KB)	✓
Curve25519	398*	69 (30 KB)	225 (3.8 KB)	✓
numsp256t1	300	82 (9 KB)	231 (3 KB)	✓

Weierstrass versus twisted Edwards

- Compared to old (NIST) curves, twisted Edwards is for **ECDHE** 1.6x and **ECDSA sign** 2.4x faster
- Compared to our new curves, twisted Edwards is for **ECDHE** 1.3x and **ECDSA sign** 1.3x faster

S. Gueron, V. Krasnov (2013), *Fast prime field elliptic curve cryptography with 256 bit primes*
D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, B.-Y. Yang (2011), *High-speed high-security signatures*
D. J. Bernstein (2006), *Curve25519: New Diffie-Hellman Speed Records*

Performance Results

128-bit security	ECDHE	ECDSA sign	ECDSA ver	Constant time
NIST P-256	490	90 (150 KB)	530	?
numsp256d1	379	107 (6 KB)	288 (2 KB)	✓
Curve25519	398*	69 (30 KB)	225 (3.8 KB)	✓
numsp256t1	300	82 (9 KB)	231 (3 KB)	✓

Source code available (open source apache license v2.0)
http://research.microsoft.com/en-us/projects/nums/

S. Gueron, V. Krasnov (2013), *Fast prime field elliptic curve cryptography with 256 bit primes*
D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, B.-Y. Yang (2011), *High-speed high-security signatures*
D. J. Bernstein (2006), *Curve25519: New Diffie-Hellman Speed Records*

Conclusions

New set of elliptic curves for the 128-, 192-, and 256-bit security levels

Performance
- ❖ Use dedicated formulas: provably no exceptions
- ❖ New Weierstrass curves significantly faster than NIST curves

Security
- ❖ Use algorithms which can be implemented in constant-time
- ❖ Primes and curves are chosen deterministically (minimize room for backdoors)

Practical

Weierstrass curves
- ❖ Backward compatible with existing APIs
- ❖ No changes in protocols required

Twisted Edwards / Montgomery curves
- ❖ Use twisted Edwards arithmetic only (no Montgomery ladder) Reduce maintenance cost and code size (no restrictions)
- ❖ Better performance (1.3x) but protocol adjustments required

IETF proposal online

Elliptic Curve Cryptography Nothing Up My Sleeve Curves and Curve Generation

http://tools.ietf.org/html/draft-black-numscurves-01