Curve41417: fast, highly secure and implementation-friendly curve

Chitchanok Chuengsatiansup

Technische Universiteit Eindhoven

June 12, 2014

Joint work with Daniel J. Bernstein and Tanja Lange

Existing deployment of Curve41417

What is the goal of new crypto?

- Example of old crypto:
  - OpenSSL `secp160r1` (security level only 2^{80})
    - least secure option supported by OpenSSL
    - $\approx$ **2.1 million** Cortex-A8 cycles (not constant time)

What is the goal of new crypto?

- Example of old crypto:
  - OpenSSL `secp160r1` (security level only 2^{80})
    - least secure option supported by OpenSSL
    - $\approx$ **2.1 million** Cortex-A8 cycles (not constant time)
- Best speed with acceptable security?
  - Curve25519 (security level 2^{125})
    - $\approx$ **0.5 million** Cortex-A8 cycles (constant time)
  - Kummer (hyperelliptic, security level 2^{125})
    - $\approx$ **0.3 million** Cortex-A8 cycles (constant time)

What is the goal of new crypto?

- Example of old crypto:
  - OpenSSL `secp160r1` (security level only 2^{80})
    - least secure option supported by OpenSSL
    - $\approx$ **2.1 million** Cortex-A8 cycles (not constant time)
- Best speed with acceptable security?
  - Curve25519 (security level 2^{125})
    - $\approx$ **0.5 million** Cortex-A8 cycles (constant time)
  - Kummer (hyperelliptic, security level 2^{125})
    - $\approx$ **0.3 million** Cortex-A8 cycles (constant time)
- Best security with acceptable speed?
  - Curve41417 (security level above 2^{200})
    - $\approx$ **1.8 million** Cortex-A8 cycles (constant time)

Design of Curve41417

- High-security elliptic curve (security level above 2^{200})
- Defined over prime field $\mathbf{F}_p$ where $p = 2^{414} - 17$
- In Edwards curve form

$$x^2 + y^2 = 1 + 3617x^2y^2$$

Design of Curve41417

- High-security elliptic curve (security level above 2^{200})
- Defined over prime field $\mathbf{F}_p$ where $p = 2^{414} - 17$
- In Edwards curve form

$$x^2 + y^2 = 1 + 3617x^2y^2$$

- IEEE P1363 criteria (large embedding degree, etc.)
- Large prime-order subgroup (cofactor 8)
- Twist secure (twist cofactor 8)
- 3617 is smallest value satisfying these criteria

Prime $2^{414} - 17$

- Extremely close to a power of 2
- Difference 17 has just two bits set
- $2^{414}x \mod p$ computed as $16x + x$ with single shift-and-add
- 414 is divisible by $9, 18, 23, 46$
- 416 (for $4p$) is divisible by $8, 13, 16, 26, 32, 52$
- With 32-bit words, wasted bandwidth under 1% ($13 \cdot 32 = 416$)
  allowing two extra bits for extension
  e.g., sign bit in a compressed point

Importance of prime choice

- NIST P-384
  - $p = 2^{384} - 2^{128} - 2^{96} + 2^{32} - 1$
  - reduction requires 4 additions for radix 2^{32}
  - for other radix, implementor has a choice:

Note: count subtraction as addition

Importance of prime choice

- NIST P-384
  - $p = 2^{384} - 2^{128} - 2^{96} + 2^{32} - 1$
  - reduction requires 4 additions for radix 2^{32}
  - for other radix, implementor has a choice:
    - slower and much more complicated
    - more complicated and much slower

Note: count subtraction as addition

Importance of prime choice

- NIST P-384
  - $p = 2^{384} - 2^{128} - 2^{96} + 2^{32} - 1$
  - reduction requires 4 additions for radix 2^{32}
  - for other radix, implementor has a choice:
    - slower and much more complicated
    - more complicated and much slower
- Curve41417
  - $p = 2^{414} - 17$
  - reduction requires 1 shift and 2 additions

Note: count subtraction as addition

Importance of curve choice

Curve	DBL	ADD	mADD
Short Weierstrass	8	16	11
Twisted Hessian	8	11	9
Twisted Edwards	7	8	7

Note: assuming best known coordinates

mADD = mixed addition

mDADD = mixed differential addition

Importance of curve choice

Curve	DBL	ADD	mADD	mDADD
Short Weierstrass	8	16	11	-
Twisted Hessian	8	11	9	-
Twisted Edwards	7	8	7	-
Montgomery	4	-	-	5

Note: assuming best known coordinates

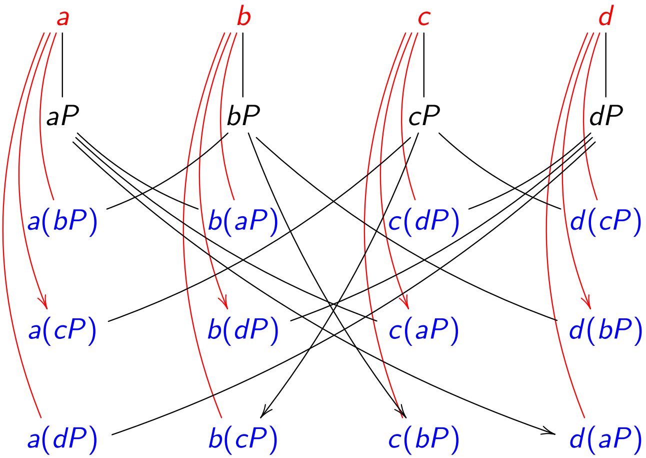
mADD = mixed addition

mDADD = mixed differential addition

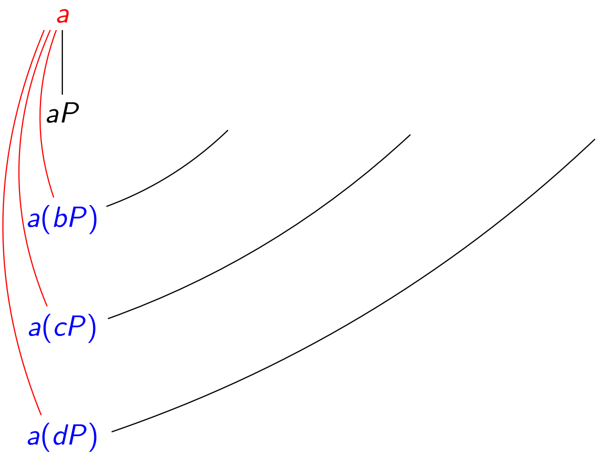
Diffie–Hellman Key Exchange

$$\begin{tikzcd}
a \arrow[d] \arrow[dd, bend right] & b \arrow[d] \arrow[dd, bend right] \\
aP \arrow[dr] & bP \arrow[dl] \\
a(bP) & b(aP)
\end{tikzcd}$$

Diffie–Hellman Key Exchange

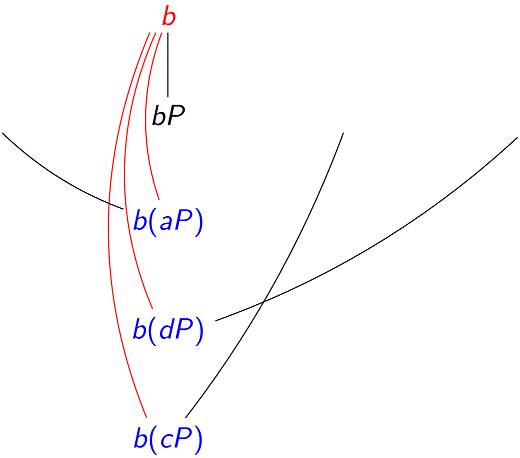


Diffie–Hellman Key Exchange



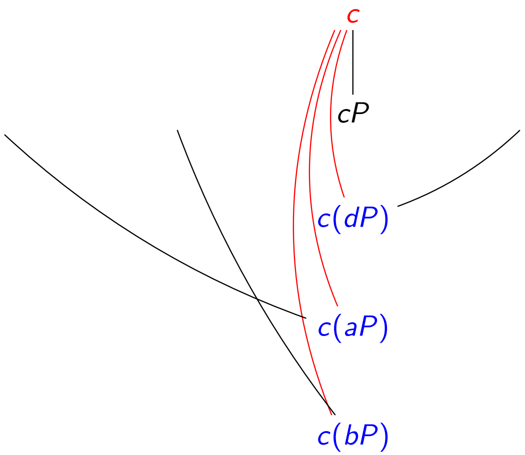
main DH challenge: make variable-base scalar mult as fast as possible

Diffie–Hellman Key Exchange

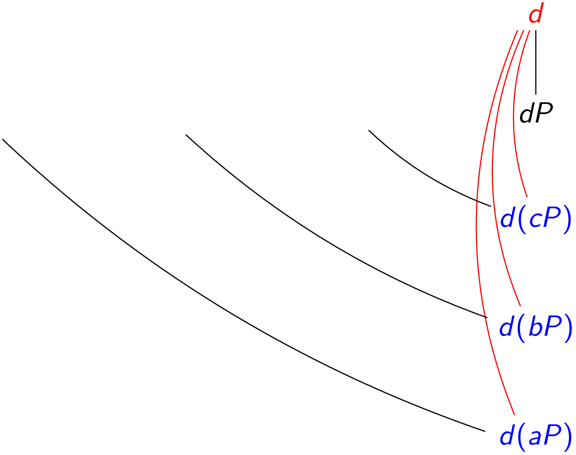


main DH challenge: make variable-base scalar mult as fast as possible

Diffie–Hellman Key Exchange



main DH challenge: make variable-base scalar mult as fast as possible



main DH challenge: make variable-base scalar mult as fast as possible

Side-Channel Attack

- Prevent software side-channel attack:
  - constant-time
  - no input-dependent branch
  - no input-dependent array index
- Constant-time table-lookup:
  - read entire table
  - select via arithmetic
    if c is 1, select tbl[i]
    if c is 0, ignore tbl[i]

$$t = (t \quad \cdot \quad (1-c)) \quad + \quad (\mathrm{tbl[i]} \quad \cdot \quad (c)\ )$$
$$t = (t \ \texttt{and}\ (c-1)) \ \texttt{xor}\ (\mathrm{tbl[i]}\ \texttt{and}\ (-c))$$

ECC Arithmetic

- Mix coordinate systems:
  - doubling: projective X, Y, Z
  - addition: extended X, Y, Z, T

  (See https://hyperelliptic.org/EFD/)

- Scalar multiplication:
  - signed fixed windows of width $w = 5$
    Example: $2345 = \underline{10}\ \underline{01001}\ \underline{01001}\ _2$
  - precompute $0P, 1P, 2P, \ldots, 16P$
    also multiply $d = 3617$ to T coordinate
  - compute T only before addition

Understanding ECC scalability to large primes

- Example: scaling from 255-bit to 414-bit scalar multiplication

Understanding ECC scalability to large primes

- Example: scaling from 255-bit to 414-bit scalar multiplication
- Schoolbook field multiplication
  expected scalar multiplication scaling $(414/255)^3 \approx 4.3$

Understanding ECC scalability to large primes

- Example: scaling from 255-bit to 414-bit scalar multiplication
- Schoolbook field multiplication
  expected scalar multiplication scaling $(414/255)^3 \approx 4.3$
- 2-level reduced refine Karatsuba
  actual performance scaling $(1.8/0.5) \approx 3.6$

Curve41417

- Very fast
  - $\approx$ 1.6 million cycles on FreeScale i.MX515
  - $\approx$ 1.8 million cycles on TI Sitara
- Very high security (above 2^{200})
  - also twist-secure
- Very flexible radix
  - support different sizes of limbs
- Very easy modular reduction

Curve41417

- Very fast
  - $\approx$ 1.6 million cycles on FreeScale i.MX515
  - $\approx$ 1.8 million cycles on TI Sitara
- Very high security (above 2^{200})
  - also twist-secure
- Very flexible radix
  - support different sizes of limbs
- Very easy modular reduction
- Real world deployment
  - "Blackphone has been added to the permanent collection at the world-renowned International Spy Museum in the gallery Weapons of Mass Disruption"

Cost Comparison (Karatsuba)

Level	Mult.	Add 64-bit	Add 32-bit	Cost
0-level	256	15	0	$256+\ 8+\ 0=264$
1-level	192	59	16	$192+30+\ 4=226$
2-level	144	119	40	$144+60+10=214$
3-level	108	191	76	$108+96+19=223$

Note: use multiply-add instructions

Recall:

1 cycle per multiplication

0.5 cycle per 64-bit addition

0.25 cycle per 32-bit addition

Cost Comparison (refined Karatsuba)

Level	Mult.	Add		Cost
		64-bit	32-bit	
0-level	256	15	0	$256+\ 8+\ 0=264$
1-level	192	52	16	$192+26+\ 4=222$
2-level	144	103	40	$144+52+10=206$
3-level	108	166	76	$108+83+19=210$

Note: use multiply-add instructions

Recall:

1 cycle per multiplication

0.5 cycle per 64-bit addition

0.25 cycle per 32-bit addition

Cost Comparison (reduced refined Karatsuba)

Level	Mult.	Add		Cost
		64-bit	32-bit	
0-level	256	15	0	$256+\ 8+\ 0 = 264$
1-level	192	45	16	$192+23+\ 4 = 219$
2-level	144	96	40	$144+48+10 = 202$
3-level	108	159	76	$108+80+19 = 207$

Note: use multiply-add instructions

Recall:

1 cycle per multiplication

0.5 cycle per 64-bit addition

0.25 cycle per 32-bit addition