Generation, Verification, and Attacks on Elliptic Curves and their Applications in Signal Protocol

Tanay Dusane

Research Idea Ring
Department of Computer Science

March 19, 2020

Outline

1. Elliptic curves
2. Special types of elliptic curves
3. Arithmetic of EC, ECDLP
4. Usage in cryptography
5. What makes a secure curve?
6. Signal protocol
7. X3DH

Goal of the thesis

- G1: Generate curves based on the security specifications.
- G2: Verify the security of the generated curves.
- G3: Demonstrate attacks on weak curves.
- G4: Test the usability of the generated curves in protocols.
- G5: Can they be Quantum safe?

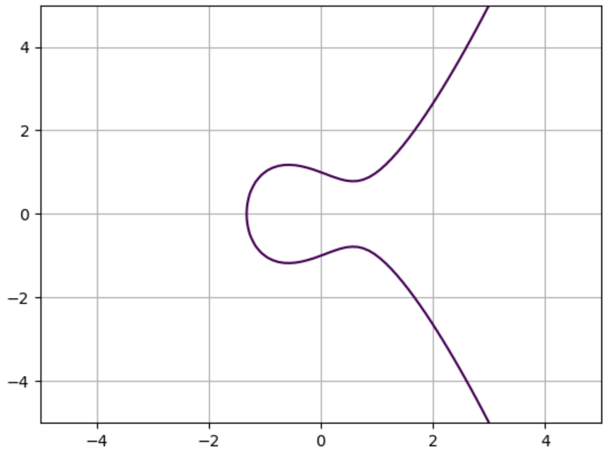
Weierstrass elliptic curves

- $a, b \in \mathbb{Z}_p$ such that $4a^3 + 27b^2 \neq 0$
- non-singular EC is a set E of solutions $(x, y) \in \mathbb{Z}_p$ to equation
$$y^2 = x^3 + ax + b \text{ (mod p)},$$
with a special point O.

Weierstrass elliptic curves

- $a, b \in \mathbb{Z}_p$ such that $4a^3 + 27b^2 \neq 0$
- non-singular EC is a set E of solutions $(x, y) \in \mathbb{Z}_p$ to equation
$$y^2 = x^3 + ax + b \text{ (mod p)},$$
with a special point O.

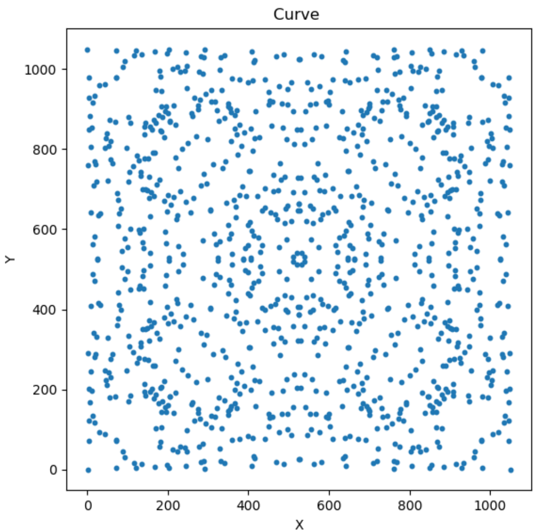
EC over reals:



Special types of elliptic curves

Edwards curve,

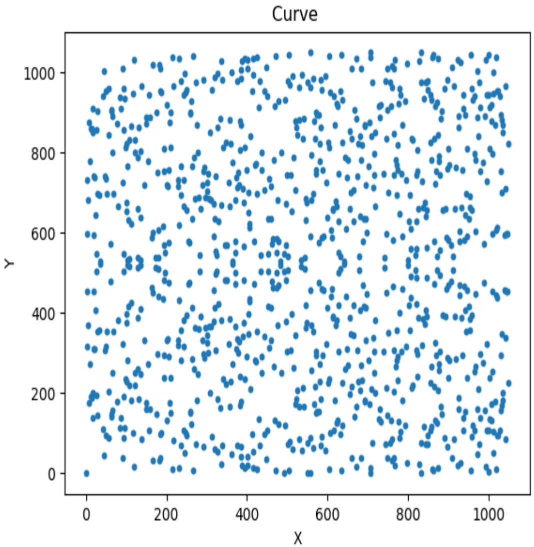
$x^2 + y^2 = 1 + dx^2y^2$



$x^2 + y^2 = 1 + 2x^2y^2 \pmod{1051}$

Montgomery curve,

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



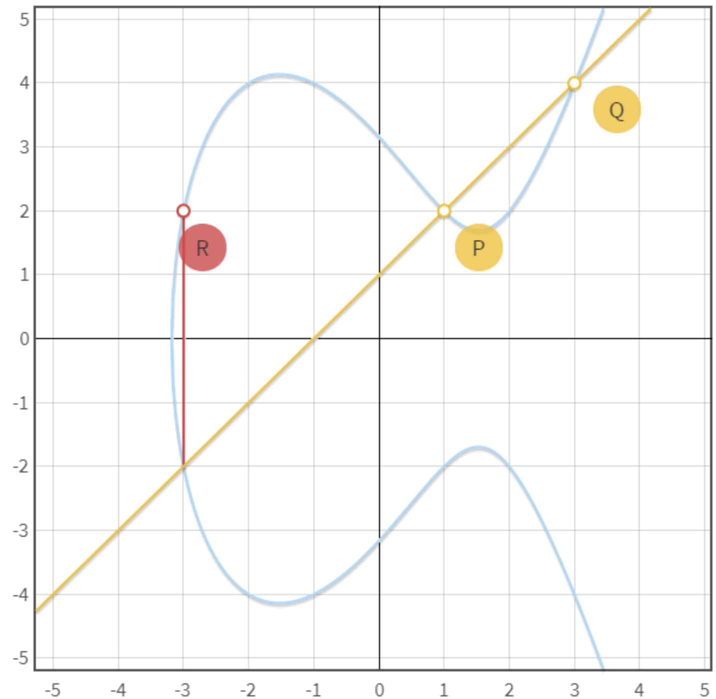
$y^2 = x^3 + 10x^2 + x \pmod{1051}$

Point addition

$P = (x_1, y_1)$
$Q = (x_2, y_2)$, where $x_1 \neq x_2$

$P + Q = R = (x_3, y_3)$,

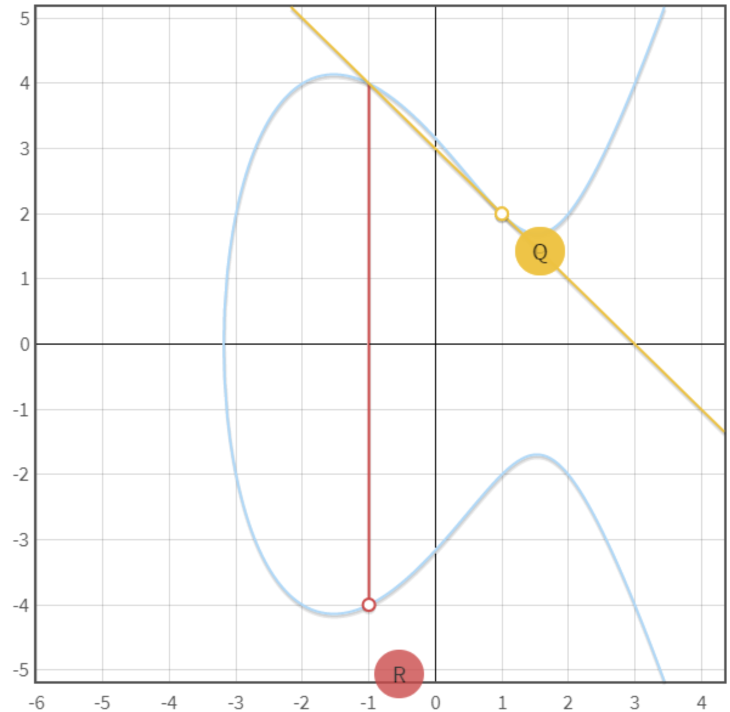
$x_3 = \lambda^2 - x_1 - x_2$,
$y_3 = \lambda(x_1 - x_3) - y_1$,
$where\ \lambda = (y_2 - y_1)(x_2 - x_1)^{-1}$.



Point doubling

$$Q + Q = 2Q = R = (x_3, y_3),$$

$$x_3 = \lambda^2 - x_1 - x_2,$$
$$y_3 = \lambda(x_1 - x_3) - y_1,$$
$$where\ \lambda = (3x_1^2 + a)(2y_1)^{-1}.$$



Discrete Log Problem for Elliptic curves (G1,G3)

- DLP: $z = x^y$ mod p, find y

Discrete Log Problem for Elliptic curves

(G1,G3)

- DLP: $z = x^y$ mod p, find y
- ECDLP: DLP on EC.

Discrete Log Problem for Elliptic curves (G1,G3)

- DLP: $z = x^y \bmod \mathrm{p}$, find y
- ECDLP: DLP on EC.

$P, Q \in E$, find d, where $d \in < 1, < Q >>$, such that

$$Q = dP.$$

Discrete Log Problem for Elliptic curves (G1,G3)

- DLP: $z = x^y$ mod p, find y
- ECDLP: DLP on EC.

$P, Q \in E$, find d, where $d \in <1, <Q>>$, such that

$$Q = dP.$$

ECDLP is a good candidate for one-way function.

- State of art DL solver: up to 750 bits.
- State of art ECDL solver: up to 110 bits.

Usage in cryptography

Mainly used in public key cryptography.

Usage in cryptography

Mainly used in public key cryptography.

- Provide smaller keys compared to `RSA`.

Usage in cryptography

Mainly used in public key cryptography.

- Provide smaller keys compared to `RSA`.
- Key exchange: Elliptic Curve Diffie Hellman (ECDH)
- Digital signatures algorithms:
  Elliptic Curve Digital Signature Algorithm (ECDSA)
  Edwards Curve Digital Signature (EDDSA)

What makes a secure curve?

(G1,G2,G3,G5)

It's the parameters!

What makes a secure curve?

(G1,G2,G3,G5)

It's the parameters!

- ECDL problem should be hard to solve under defined parameters.

$$|\#E| = \underbrace{c}_{\text{small cofactor}} \cdot \underbrace{p}_{\text{large prime}}$$

What makes a secure curve?

(G1,G2,G3,G5)

It's the parameters!

- ECDL problem should be hard to solve under defined parameters.

  $$|\#E| = \underbrace{c}_{\text{small cofactor}} \cdot \underbrace{p}_{\text{large prime}}$$

- Families of curves with particular properties.

  Curves with fast endomorphism and pairing friendly curves.

What makes a secure curve?

(G1,G2,G3,G5)

It's the parameters!

- ECDL problem should be hard to solve under defined parameters.

  $|\#E| = c \cdot p$

  small cofactor   large prime

- Families of curves with particular properties.

  Curves with fast endomorphism and pairing friendly curves.

- Normality of the curve.

What makes a secure curve?

(G1,G2,G3,G5)

It's the parameters!

- ECDL problem should be hard to solve under defined parameters.

  $$|\#E| = \underbrace{c}_{\text{small cofactor}} \cdot \underbrace{p}_{\text{large prime}}$$

- Families of curves with particular properties.

  Curves with fast endomorphism and pairing friendly curves.

- Normality of the curve.
- Should be convenient to implement the curve.

  Point compression and fast base field arithmetic.

- Implementation dependent security.

What makes a secure curve?

(G1,G2,G3,G5)

It's the parameters!

- ECDL problem should be hard to solve under defined parameters.

  $$|\#E| = c \cdot p$$

  $$\begin{array}{cc} \swarrow & \searrow \\ \text{small cofactor} & \text{large prime} \end{array}$$

- Families of curves with particular properties.

  Curves with fast endomorphism and pairing friendly curves.
- Normality of the curve.
- Should be convenient to implement the curve.

  Point compression and fast base field arithmetic.
- Implementation dependent security.

  Resistance against side channel attacks.

  Microsoft Curveball vulnerability.

Signal curves

(G1)

`Signal` protocol is an end-to-end encryption protocol used in

- `WhatsApp`
- `Facebook Messenger`
- `Signal Messenger`

Two elliptic curves are used in `Signal`:

- Curve25519, Montgomery curve, Dan Bernstein, 2005
  128 bit security.

- Curve448, Edwards curve, Mike Hamburg, 2015
  224 bit security.

`Signal` uses extended triple Diffie-Hellman.

Diffie-Hellman Key Exchange (G3,G4)

DH

Alice
choose $Pr_A = a \in \{2, 3, \ldots, p-2\}$

Bob
choose $Pr_B = b \in \{2, 3, \ldots, p-2\}$

$$\xrightarrow{Pub_A = g^a \pmod{p} = A}$$

$$\xleftarrow{Pub_B = g^b \pmod{p} = B}$$

computes $B^a \pmod{p}$

computes $A^b \pmod{p}$

Shared secret $B^a = (g^b)^a = g^{ab} = A^b$

Diffie-Hellman Key Exchange (G3,G4)

DH

Alice
choose $Pr_A = a \in \{2, 3, ..., p-2\}$

Bob
choose $Pr_B = b \in \{2, 3, ..., p-2\}$

$$\xrightarrow{Pub_A = g^a \pmod{p} = A}$$

$$\xleftarrow{Pub_B = g^b \pmod{p} = B}$$

computes $B^a \pmod{p}$

computes $A^b \pmod{p}$

Shared secret $B^a = (g^b)^a = g^{ab} = A^b$

ECDH

Alice
choose $Pr_A = a \in \{2, 3, ..., \#E-1\}$

Bob
choose $Pr_B = b \in \{2, 3, ..., \#E-1\}$

$$\xrightarrow{Pub_A = aP = A = (x_A, y_A)}$$

$$\xleftarrow{Pub_B = bP = B = (x_B, y_B)}$$

compute $aB = T_{AB}$

compute $bA = T_{AB}$

Shared secret $T_{AB} = (x_{AB}, y_{AB})$

Curve25519

Equation of the curve is : $y^2 = x^3 + 486662x^2 + x$
Specified parameters as per RFC 7748:

Name	**Definition**
$field$	$2^{255} - 19$
$order$	2^{252} + 0x14def9dea2f79cd65812631a5cf5d3ed
$cofactor$	8

Base point $< P >$
(x(P), y(P)) = (9,14781619447589544791020593568409986887264606134616475288964881837755586237401).

Curve448

The equation of the curve is : $x^2 + y^2 = 1 - 39081x^2y^2$
Specified parameters as per RFC 7748:

Name	**Definition**
$field$	$2^{448} - 2^{224} - 1$
$order$	2^{446} - 0x8335dc163bb124b65129c96fde933d8d723a70aadc873d6d54a7bb0d
$cofactor$	4

Base point $< P >$
(x(P), y(P)) = (224580040295924300187604334099896036246789641632
564134246125461686950415467406032909029192869357953282578032075146446173674602635247710,
298819210078481492676017930443930673437544040154080242095928241372331506189835876003536878655418784733982303233503462500531545062832660)

Extended Triple Diffie-Hellman

(G3,G4)

- Designed for asynchronous settings.

Extended Triple Diffie-Hellman (G3,G4)

- Designed for asynchronous settings.
- Provides forward secrecy and cryptographic deniability.

Extended Triple Diffie-Hellman

(G3,G4)

- Designed for asynchronous settings.
- Provides forward secrecy and cryptographic deniability.

The keys used in X3DH are elliptic curve public keys.

Extended Triple Diffie-Hellman (G3,G4)

- Designed for asynchronous settings.
- Provides forward secrecy and cryptographic deniability.

The keys used in X3DH are elliptic curve public keys.

- Identity keys

  IK_A - Alice's Identity Key.

  IK_B - Bob's Identity Key.

Extended Triple Diffie-Hellman

(G3,G4)

- Designed for asynchronous settings.
- Provides forward secrecy and cryptographic deniability.

The keys used in X3DH are elliptic curve public keys.

- Identity keys
  - IK_A - Alice's Identity Key.
  - IK_B - Bob's Identity Key.
- Ephemeral keys
  - EK_A - Alice's Ephemeral Key.
  - EK_B - Bob's Ephemeral Key.

Extended Triple Diffie-Hellman (G3,G4)

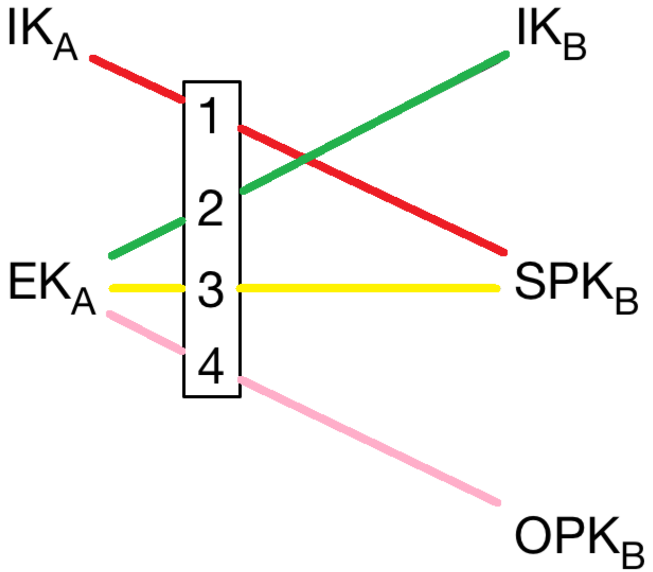
- Designed for asynchronous settings.
- Provides forward secrecy and cryptographic deniability.

The keys used in X3DH are elliptic curve public keys.

- Identity keys
  IK_A - Alice's Identity Key.
  IK_B - Bob's Identity Key.
- Ephemeral keys
  EK_A - Alice's Ephemeral Key.
  EK_B - Bob's Ephemeral Key.
- Signed prekeys
  SPK_B - Bob's signed prekey.
  OPK_B - Bob's one time prekey.

X3DH

1. $DH1 = DH(IK_A, SPK_B)$
2. $DH2 = DH(EK_A, IK_B)$
3. $DH3 = DH(EK_A, SPK_B)$
4. If OPK_B exists,
   $DH4 = DH(EK_A, OPK_B)$



Secret Key $SK = KDF(DH1||DH2||DH3||DH4)$

Goals

- G1: Generate curves based on the security specifications. (March, 2020)
- G2: Verify the security of the generated curves. (March, 2020)
- G3: Demonstrate attacks on weak curves. (April, 2020)
- G4: Test the usability of the generated curves in protocols. (May, 2020)
- G5: Can they be Quantum safe? (May, 2020)

References

- J.-P. Flori, J. Plût, J.-R. Reinhard, and M. Ekerå, "Diversity and transparency for ecc", Cryptology ePrint Archive, no. 659, 2015.
- H. Edwards, A normal form for elliptic curves, Bulletin of the American Mathematical Society, `https://bit.ly/2NQylSy`, 2007.
- D. R. Stinson and M. B. Paterson, *Cryptography: Theory and Practice, Fourth Edition*. 2019.
- D. J. Bernstein, P. Birkner, M. Joye, T. Lange, and C. Peters, Twisted Edwards Curves, Cryptology ePrint Archive, `https://eprint.iacr.org/2008/013`.
- E. M. Barnard, *Tutorial of twisted edwards curves in elliptic curve cryptography*, UC SANTA BARBARA, CS 290G, FALL 2015, `https://koclab.cs.ucsb.edu/teaching/ecc/project/2015Projects/Barnard-Paper.pdf`, 2015.