Elliptic Curve Diffie-Hellman

Notes for *Serious Cryptography* chapter 12

Jeffrey Goldberg

jeffrey@goldmark.org

March 11, 2022 (Revised May 18, 2024)

Review of Integer DHKE

Alice picks a secret, little a, and generates a public big A.

$$A = g^a \tag{1}$$

Bob does similarly

$$B = g^b \tag{2}$$

Never sending secrets

- Alice sends A to Bob.
- Alice never transmits a.
- Bob sends B to Alice.
- Bob never transmits b.

Computing keys

Alice knows a and B. She computes

$$\mathsf{k}_A = B^a \tag{3}$$

Bob knows b and A. He computes

$$\mathsf{k}_B = A^b \tag{4}$$

Common key $\mathsf{k}_{A,B}$

$$\begin{aligned} \mathsf{k}_A &= B^a = \left(g^b\right)^a = g^{ab} \\ \mathsf{k}_B &= A^b = \left(g^a\right)^b = g^{ba} \end{aligned} \tag{5}$$

$$\mathsf{k}_A = g^{ba} = g^{ab} = \mathsf{k}_B$$

In an elliptic curve group

- When defined appropriately an elliptic curve $\mathcal{E}_p$ with point addition forms an abelian cyclic group
- Repeated point addition (analogous to exponentiation in integer groups) can be computed efficiently
- The DLP is at least as hard in $\mathcal{E}_p$ as it is in $\mathbb{Z}_p^\times$

DLP in elliptic curve groups

Generalizing the DLP

The cryptographically useful properties of the discrete logarithm problem requires a finite cyclic group (with a few other conditions on the group).

The group does not need to have integer elements and modular multiplication. It can be constructed from other things if it has the right structure. The term “generalized DLP” (GDLP) is sometimes used to talk about this generalization of the the DLP.

Notation wars

Operation	$\mathbb{Z}_p^\times$	$\mathcal{E}_p$
Op. name	mod. multiplication	point addition
	ab	$P+Q$
Repeated	exponentiation	scalar multiplication
	x^n	nP
From generator	$A = g^a$	$P = dG$
Logarithm	Find a given A, g	Find p given P, G
	$\log_g A$	P/G

Table 1: Terms and notation for integer and elliptic curve groups

Notation wars

Operation	$\mathbb{Z}_p^\times$	$\mathcal{E}_p$
Op. name	mod. multiplication ab	point addition $P+Q$
Repeated	exponentiation x^n	scalar multiplication nP
From generator	$A = g^a$	$P = dG$
Logarithm	Find a given A, g $\log_g A$	Find p given P, G P/G

Table 1: Terms and notation for integer and elliptic curve groups

1. ECs have a distant history in geometry, and so points are typically referred to using capital letters like P and Q, while in integer groups p is often used for the prime modulous.
2. When talking abstractly about groups, G is often used to refer to a group, but with elliptic curves, it is often used to describe the generator or base point.
3. “Scalar” means ordinary number. n is *not* a point.

Penelope and Quintin pick a secrets

Penelope picks a secret, d_P, and generates a public P.

$$P = d_P G \tag{6}$$

Quintin does similarly

$$Q = d_Q G \tag{7}$$

2024-05-18

Penelope and Quintin pick a secrets

Penelope picks a secret, d_P, and generates a public P.

$$P = d_P G \tag{6}$$

Quintin does similarly

$$Q = d_Q G \tag{7}$$

1. d_P is Penolope's *decryption* secret. We don't use 'p' and 'q' because those have other meanings in defining an elliptic curve.
2. We use 'P' and 'Q' because 'A' and 'B' are used for other things when talking about elliptic curves.

Never sending secrets

- Penelope sends P to Quintin.
- Penelope never transmits d_P.
- Quintin sends Q to Penelope.
- Quintin never transmits d_Q.

Computing keys

Penelope knows d_P and Q. She computes

$$\mathsf{k}_P = d_P Q \tag{8}$$

Quintin knows d_Q and P. He computes

$$\mathsf{k}_Q = d_Q P \tag{9}$$

Common key $\mathsf{k}_{P,Q}$

$$\mathsf{k}_P = d_P Q = d_P(d_Q G) = d_P d_Q G$$
$$\mathsf{k}_Q = d_Q P = d_Q(d_P G) = d_Q d_P G \tag{10}$$

All the things we said about constructions based on Diffie-Hellman in $\mathbb{Z}_p^\times$ apply to $\mathcal{E}_p$, including, but not limited to

- Need to hash output to get things that work as keys
- Distinction between Decisional and Computational assumptions
- Vulnerability to Cat in the Middle attacks.
- Existence of $\mathsf{poly}(|p|)$ on suitable quantum computers

Point addition

It's a set of points

Definition (Elliptic curve)

An elliptic curves is the set of points (x, y) defined by

$$y^2 = x^3 + Ax + B \tag{11}$$

and a special point called '$\mathbf{0}$'.

It's a set of points

Definition (Elliptic curve)

An elliptic curves is the set of points (x, y) defined by

$$y^2 = x^3 + Ax + B \tag{11}$$

and a special point called '$\mathbf{0}$'.

1. This is the form of a Weierstrass elliptic curve. There are others.

Keeping $x, y \in \mathbb{R}$

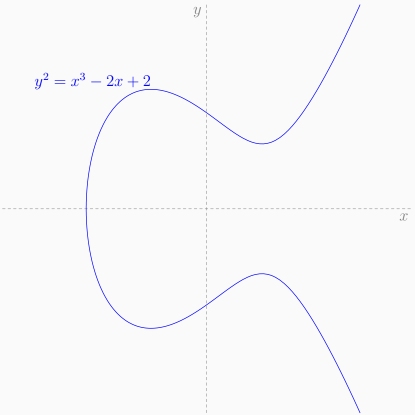


Figure 1: Elliptic curve over the reals

Keeping $x, y \in \mathbb{R}$

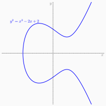


Figure 1: Elliptic curve over the reals

1. In order to draw pretty pictures explaining point addition, we will (for the time being) let x and y be real numbers. Here we will define curves “over the reals.”

A picture with points

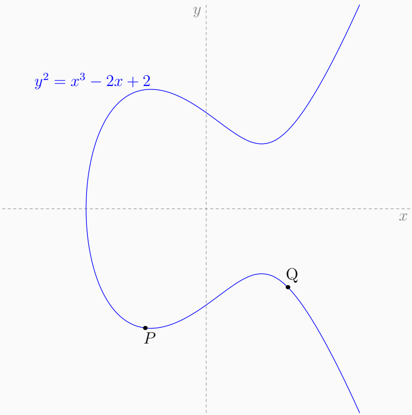


Figure 2: Same curve with points P and Q

A picture with points

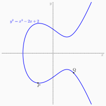


Figure 2: Same curve with points P and Q

1. These P and Q have nothing to do with Penelope or Quintin.

Rule of Three

Every straight line that intersects the curve at least twice must intersect it exactly three times.

Three point rule illustrated

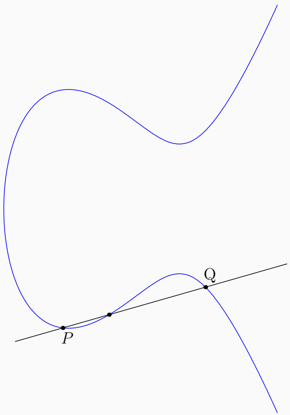


Figure 3: Line through P and Q crosses curve at another point

$P + Q$

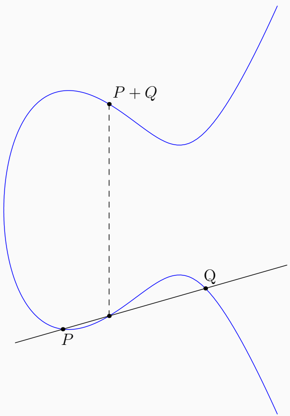


Figure 4: The vertical reflection of the third intersecting point is $P + Q$

Tangents count double

A line that is tangent to a point counts as intersecting that point twice.

Three point rule and tangents

A line that is tangent to a point on the curve must intersect the curve at exactly one other point.

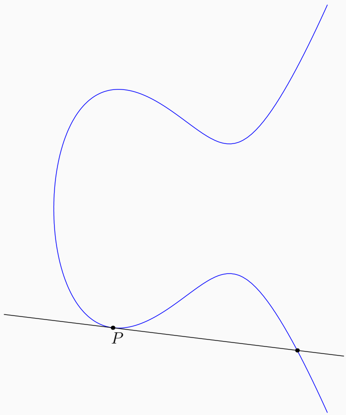


Figure 5: A line tangent to P intersects the curve at additional point

3 POINT RULE AND TANGENTS

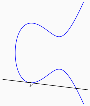


Figure 5: A line tangent to P intersects the curve at additional point

1. One way to think about this is that the line through P and P is the line tangent to the curve at P.

Point addition

$$P + P + \cdots + P$$

nP

We will want to add P to itself multiple times.

$$
\begin{aligned}
2P &= P + P \\
3P &= P + P + P \\
6P &= P + P + P + P + P + P \\
nP &= \underbrace{P + P + \cdots + P + P}_{\text{with } P \text{ appearing } n \text{ times}}
\end{aligned}
\tag{12}
$$

$2P$ (OR "POINT DOUBLING")

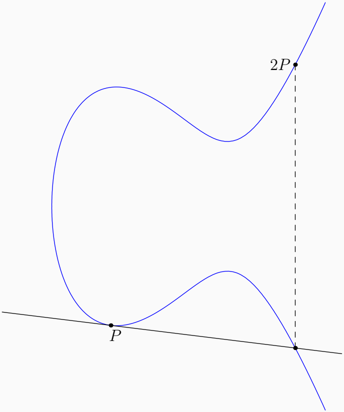


Figure 6: Point doubling: $P + P$

$3P = P + 2P$

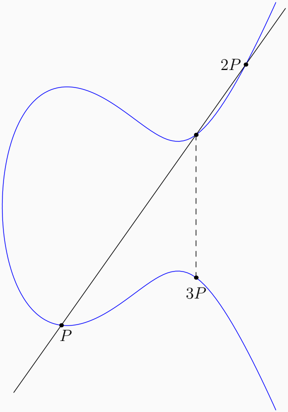


Figure 7: $3P = P + 2P$

$4P = P + 3P$

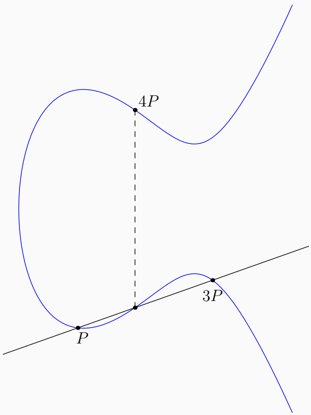


Figure 8: $4P = P + 3P$

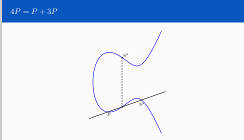


Figure 8: $4P = P + 3P$

1. It takes three point additions to get to $4P$ this way: $P + P$; $P + 2P$; and $P + 3P$.

Point addition

Speed-up

A FASTER WAY TO 4: $4P = 2P + 2P$

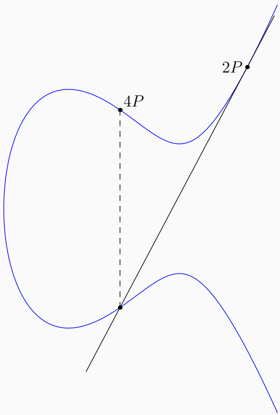


Figure 9: Computing $4P$ with only two additions: $P + P$ and $2P + 2P$

A faster way to 4: $4P = 2P + 2P$

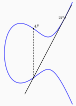


Figure 9: Computing $4P$ with only two additions: $P + P$ and $2P + 2P$

1. It takes two point additions to get to $4P$ this way.

$16P$

Computing $16P$ takes 4 point doublings

- $2P = P + P$
- $4P = 2P + 2P$
- $8P = 4P + 4P$
- $16P = 8P + 8P$

$25P$

Computing $25P$ takes 4 point doublings and two additions

- $2P = P + P$
- $4P = 2P + 2P$
- $8P = 4P + 4P$
- $16P = 8P + 8P$
- $24P = 16P + 8P$
- $25P = 24P + P$

$25P$

$$
\begin{aligned}
25P &= 16P + 8P + P \\
&= 2^4 P + 2^3 P + 2^0 P \\
&= 2^4 P + 2^3 P + 0 + 0 + 2^0 P \\
&= 1 \cdot 2^4 P + 1 \cdot 2^3 P + 0 \cdot 2^2 P + 0 \cdot 2^1 P + 1 \cdot 2^0 P \\
25 &= 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 \\
&= \texttt{0b11001}
\end{aligned}
$$

Double and Add

```
Class Point: ...
  def scaler_multiply(self, n: int) -> 'Point':
    """returns n * self"""
    sum = self.curve.PAI  # additive identity
    doubled = self
    for bit in lsb_to_msb(n):
      if bit == 1:
          sum += doubled
      doubled = doubled.double() # toil & trouble
    return sum
```

Double and Add

```
Class Point: ...
  def scaler_multiply(self, n: int) -> 'Point':
    """returns n * self"""
    sum = self.curve.PAI  # additive identity
    doubled = self
    for bit in lsb_to_msb(n):
      if bit == 1:
          sum += doubled
      doubled = doubled.double() # toil & trouble
    return sum
```

1. `PAI` is the Point at Infinity for the curve; `double()` doubles.
2. If the input has b bits, the algorithm does $b - 1$ doublings. For each 1 bit, it does a non-doubling addition.
3. I originally just wanted to write that one method for displaying here, but I ended up writing a whole EC calculator around it. See `double-and-add.py` in the source directory.
4. Do not use that `double-and-add.py` code for anything. Its only role is to be a context for illustrating the `scalar_multiply()` method.

Exponential speedup

$$g(d) = dG \qquad \text{Using repeated addition} \tag{13}$$

$$f(d) = dG \qquad \text{Using double-and-add algorithm} \tag{14}$$

$f(x)$ is exponentially faster than $g(x)$

$$\begin{aligned} g(x) &\in \mathcal{O}(x) &&= \mathcal{O}\left(2^{|x|}\right) \\ f(x) &\in \mathcal{O}(\log_2 x) &&= \mathcal{O}(|x|) \end{aligned} \tag{15}$$

Exponential speedup

$$g(d) = dG \qquad \text{Using repeated addition} \tag{13}$$

$$f(d) = dG \qquad \text{Using double-and-add algorithm} \tag{14}$$

$f(x)$ is exponentially faster than $g(x)$

$$\begin{aligned} g(x) &\in \mathcal{O}(x) &&= \mathcal{O}(2^{|x|}) \\ f(x) &\in \mathcal{O}(\log_2 x) &&= \mathcal{O}(|x|) \end{aligned} \tag{15}$$

1. $g(d)$ takes $d - 1$ point additions.
2. $f(d)$ takes less than $2 \log_2 d$ additions.

	$\mathbb{Z}_p^\times$	$\mathcal{E}_p$
Group Operation	ab	$P + Q$
Repeated	x^n	nP
From generator	$A = g^a$	$P = dG$
Logarithm	$\log_g A$	P/G
Fast algorithm	square-and-multiply	double-and-add

Table 2: More terms and notation for integer and elliptic curve groups

Square and multiply

	$\mathbb{Z}_p^\times$	$\mathcal{E}_p$
Group Operation	ab	$P + Q$
Repeated	x^n	nP
From generator	$A = g^a$	$P = dG$
Logarithm	$\log_g A$	P/G
Fast algorithm	square-and-multiply	double-and-add

Table 2: More terms and notation for integer and elliptic curve groups

1. I'm sorry. There is nothing I can do about the fact that the group operation we need over integers is modular multiplication and the one for elliptic curves is point addition.
2. When thinking about both kinds of *groups*, it is useful to think of multiplication in the integer group as being like addition in the elliptic curve group.
3. When we get to integer *fields*, as we will, we will need to stop making that comparison.
4. I'm sorry. The only way to avoid all this "what is addition and what is multiplication" stuff would have been to present all of this at a higher level of abstraction, which would have introduced way more unfamiliar notation.

Point addition

Side channels are real

Leaks like a sieve

Figure 10: Leeks like a sieve

Leaks like a sieve

Figure 10: Leeks like a sieve

1. tbh, neither a spatter screen nor a colander are sieves, but they are what I had at home.

Power consumption jumps on 1 bits

- Each 0 bit in the secret leads to a doubling operation.
- Each 1 bit in the secret leads to a doubling operation and an adding operation.
- These operations take time and consume power.

Smart cards

- Smart cards are powered by their readers;
- Malicious readers are in a very good position to measure card power consumption;
- Malicious readers are in a very good position to choose plaintext or ciphertext;
- Laboratory studies in the 1990s demonstrating the ease at which readers could exact the secret keys from cards led to a redesign of cards.[Koc96; KJJ99; AK96]

Smart cards

- Smart cards are powered by their readers;
- Malicious readers are in a very good position to measure card power consumption;
- Malicious readers are in a very good position to choose plaintext or ciphertext;
- Laboratory studies in the 1990s demonstrating the ease at which readers could exact the secret keys from cards led to a redesign of cards.[Koc96; KJJ99; AK96]

1. Smartcards used RSA, and so the attack is on the analogous “square-and-multiply” algorithm for exponentiation. Remember that in a multiplicative integer group we use modular *multiplication* as the group operation, but in elliptic curve groups we use point addition as the group operation.

LIBGCRYPT LEAKED THROUGH WALLS

- Until 2016 libgcrypt used naive double-and-add;
- libgcrypt is used by GnuPG;
- The electromagnetic leak of from changes in power consumption could be detected through walls;
- Inexpensive equipment was able to recover most of the bits of a secret through a wall;
- Enough bits were recovered to allow for brute forcing the remaining bits. [Gen+16]

Key bits

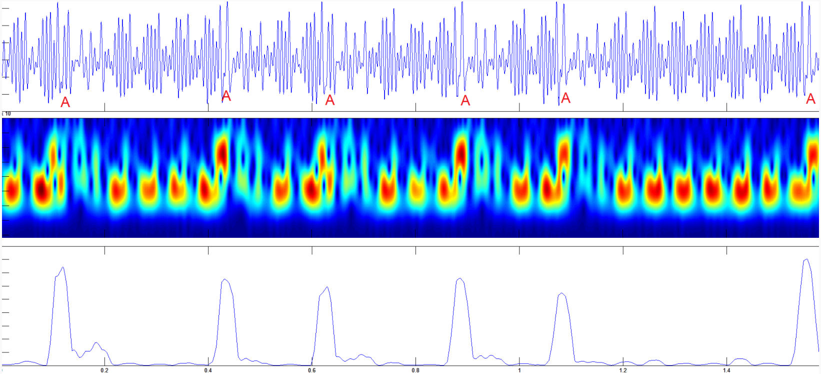


Figure 11: Figure 4 from [Gen+16]. Key bits captured in this 1.6 millisecond period are (probably) `100110110001`.

2024-05-18

Elliptic Curve Diffie-Hellman
└─Point addition
└─Side channels are real
└─Key bits

Key bits

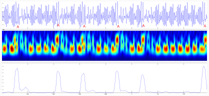


Figure 11: Figure 4 from [Gen+16]. Key bits captured in this 1.6 millisecond period are (probably) `100110110001`.

1. Well, it is a little less direct to get the actual key bits.
2. Algorithm works from the least significant end, so read those bits right to left.

Groupthink

Groupthink

Zero and infinity

Zero identity

- A group needs an identity element;
- Do vertical lines satisfy the three point rule?
- We call our operation “addition” so identity should be analogous to zero;
- Solution: We add a zero element, $\mathbf{0}$ (often written ‘$\mathcal{O}$’).

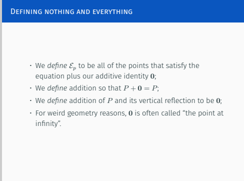
Zero identity

- A group needs an identity element;
- Do vertical lines satisfy the three point rule?
- We call our operation “addition” so identity should be analogous to zero;
- Solution: We add a zero element, $\mathbf{0}$ (often written ‘$\mathcal{O}$’).

1. I am using ‘$\mathbf{0}$’ to avoid confusion with the big-O notation.

- We *define* $\mathcal{E}_p$ to be all of the points that satisfy the equation plus our additive identity $\mathbf{0}$;
- We *define* addition so that $P + \mathbf{0} = P$;
- We *define* addition of P and its vertical reflection to be $\mathbf{0}$;
- For weird geometry reasons, $\mathbf{0}$ is often called "the point at infinity".

2024-05-18



1. "Weird geometry" is projective geometry. It really does all make sense in projective geometry.

$P + -P$

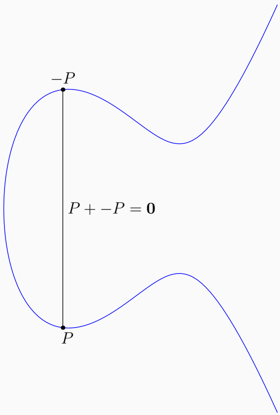


Figure 12: P plus its vertical reflection, $-P$, is $\mathbf{0}$.

Group laws

An elliptic curve, $\mathcal{E}$, including the point at infinity $\mathbf{0}$, with addition suitably defined is an abelian group because for all $P, Q, R \in \mathcal{E}$:

- It is **closed**. $P + Q \in \mathcal{E}$;
- It is **abelian** (commutative). $P + Q = Q + P$;
- There is an **identity element**, $\mathbf{0}$ such that $P + \mathbf{0} = P$;
- Every element has an **inverse**; There is an element, which we will write '$-P$', such that $P + -P = \mathbf{0}$;
- It is **associative** $(P + Q) + R = P + (Q + R)$.

Groupthink

Toward finiteness

Fields

- All fields are groups. Not all groups are fields;
- Fields have two operations, which at the moment we will call “addition” and “multiplication”;
- Both have identity elements; The additive identity element is 0. The multiplicative identity is 1;
- Both operations are closed, and both are associative;
- Every element of the field has an additive inverse, and (almost) every element has a multiplicative inverse.

Definition (Multiplicative inverse in Fields)

All elements of a field must have a multiplicative inverse except for the additive identity, 0. That is, if a is a member of the field and $a \neq 0$, then there must exist an a^{-1} in the field such that $aa^{-1} = 1$.

Definition (Distributive properties)

Multiplication distributes over addition. That is for all a, b, c in the field, $a(b + c) = ab + ac$.

Multiplication properties

Definition (Multiplicative inverse in Fields)
All elements of a field must have a multiplicative inverse except for the additive identity, 0. That is, if a is a member of the field and $a \neq 0$, then there must exist an a^{-1} in the field such that $aa^{-1} = 1$.

Definition (Distributive properties)
Multiplication distributes over addition. That is for all a, b, c in the field, $a(b + c) = ab + ac$.

1. The “except for the additive indentity” clause is just saying that you can’t divide by zero.
2. These are the only two things – that the additive identity does not have a multiplicative inverse, and that multiplication distributes over addition – in the definition of a field that distinguish between the one we call “addition” and the one we call “multiplication”.

Reminder: elliptic curves are groups

Elliptic curves are groups. The have only one operation, point addition. Point addition over properly defined elliptic curves satisfies all of the group properties.

Why talk of fields?

Elliptic curves, which are groups, are defined over fields.

Why talk of fields?

Elliptic curves, which are groups, are defined over fields.

1. Hopefully this will become clearer shortly

A rational field

The set of rational numbers, $\mathbb{Q}$, are those numbers which are ratios of whole numbers. Ordinary addition and multiplication over $\mathbb{Q}$ is a field.

- Additive identity is 0. Multiplicative identity is 1.
- Addition and multiplication are closed.
- Every a has an additive inverse, $-a$.
- Every a other than 0 has a multiplicative inverse, $\frac{1}{a}$ or a^{-1}.
- Multiplication distributes. $a(b + c) = ab + ac$.

Integers are not a field

The set of integers, $\mathbb{Z}$, with ordinary addition and multiplication

- Is not a field;
- Not every integer has a multiplicative inverse. There is no integer a such that $2a = 1$;
- $\mathbb{Z}$ with ordinary addition is a group.

The set of integers with all addition and multiplication done modulo some prime p is a field.

- It is written '$\mathbb{Z}_p$';
- It is a finite cyclic field.

Bad notation: $\mathbb{Z}_p^\times \neq \mathbb{Z}_p$

$\mathbb{Z}_p^\times$ is not a field

- $\mathbb{Z}_p^\times$ is a group with a single operation.
- That single group operation in $\mathbb{Z}_p^\times$ just happens to be modular multiplication.
- $\mathbb{Z}_p^\times$ does not contain 0.

$\mathbb{Z}_p$ is a field

- $\mathbb{Z}_p$ is a field with both modular addition and modular multiplication;
- The multiplication operation $\mathbb{Z}_p$ distributes over addition;
- The identity element of addition, 0, does not have a multiplicative inverse.

Bad notation: $\mathbb{Z}_p^\times \neq \mathbb{Z}_p$

$\mathbb{Z}_p^\times$ is not a field

- $\mathbb{Z}_p^\times$ is a group with a single operation.
- That single group operation in $\mathbb{Z}_p^\times$ just happens to be modular multiplication.
- $\mathbb{Z}_p^\times$ does not contain 0.

$\mathbb{Z}_p$ is a field

- $\mathbb{Z}_p$ is a field with both modular addition and modular multiplication;
- The multiplication operation $\mathbb{Z}_p$ distributes over addition;
- The identity element of addition, 0, does not have a multiplicative inverse.

1. I'm so sorry. I have tried to protect you some of unfortunate notation we encounter. But there isn't anything I can do about this one.
2. When we get to RSA, we will see that we can create multiplicative groups for when the modulus is not prime.
3. I am strongly implying that the modulus for a finite field must be prime. It's not entirely true, but we don't need to go into extension fields unless we were to dive into the internals of AES.

Definition

An elliptic curve is the $\mathbf{0}$ and the points (x, y) satisfying

$$y^2 = x^3 + Ax + B \tag{11}$$

And where x, y are treated as members of a field.

$\mathcal{E}_p$

The elliptic curves used in cryptography are defined over a finite field. All of the arithmetic (addition, multiplication, inversion) for computing the x and y values of particular points is performed modulo p. The result is an abelian finite cyclic group with some very nice properties.

Groupthink

Example additions

Curve $\mathcal{E}_{191}$

Definition ($\mathcal{E}_{191}$)

The finite curve from Chapter 12 is the point at infinity and the set of points which satisfy the curve equation modulo 191.

$$\mathcal{E}_{191} = \{\mathbf{0}\} \cup \left\{(x, y) \mid y^2 = x^3 - 4x + 0 \pmod{191}\right\}$$

where $x, y \in \mathbb{Z}$.

The points of $\mathcal{E}_{191}$

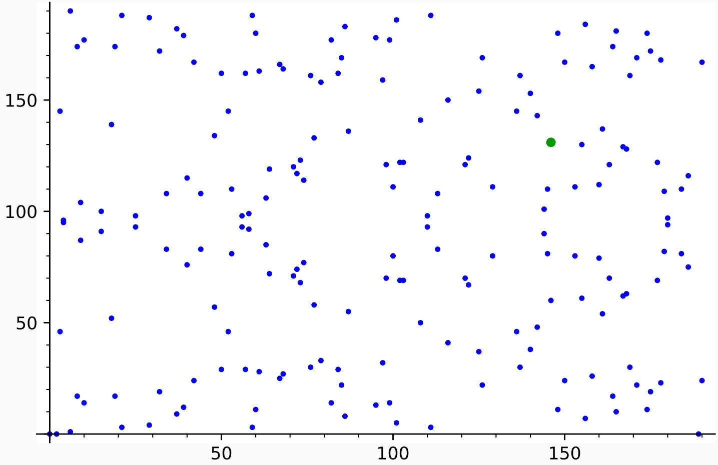


Figure 13: All the points of $\mathcal{E}_{191}$ except for $\mathbf{0}$. The generator $G = (146, 131)$.

$2G$

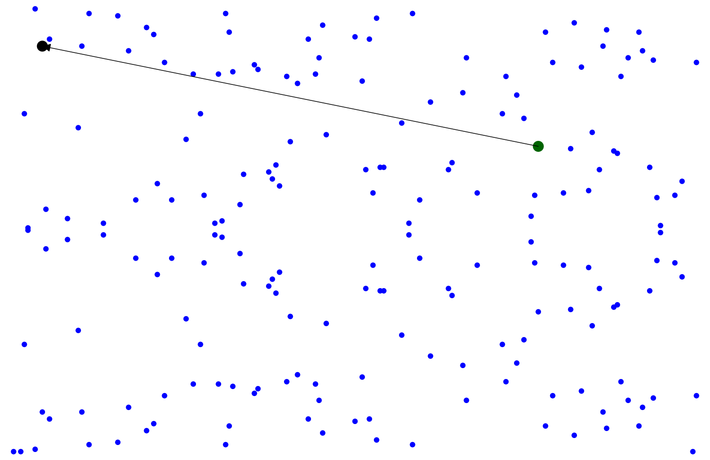


Figure 14: G to $2G$

$3G$

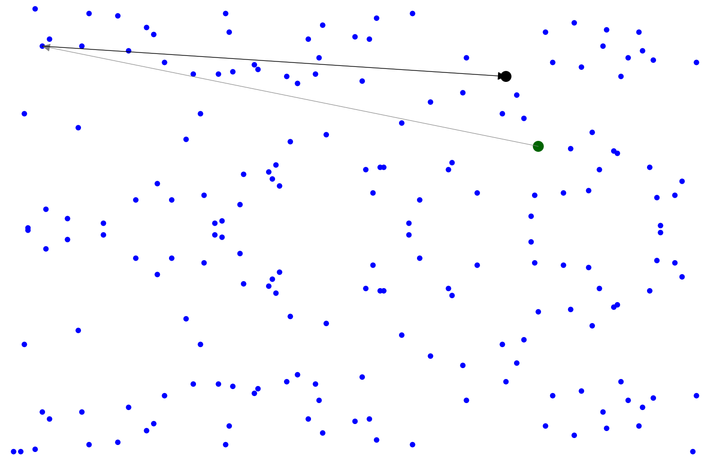


Figure 15: G to $3G$

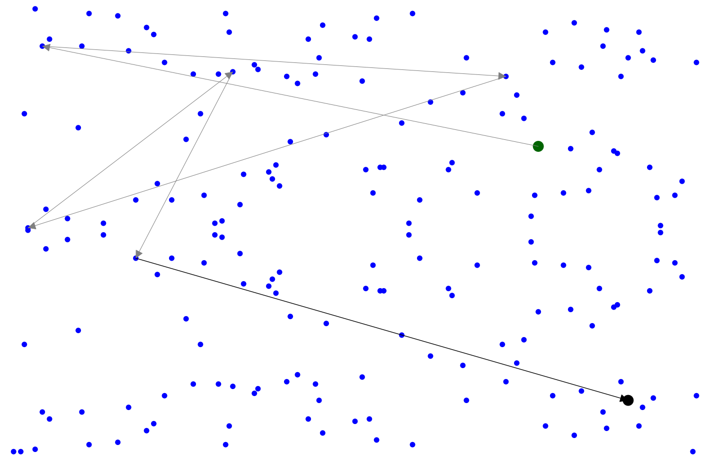


Figure 16: G to $7G$

$(|G| - 1)G$

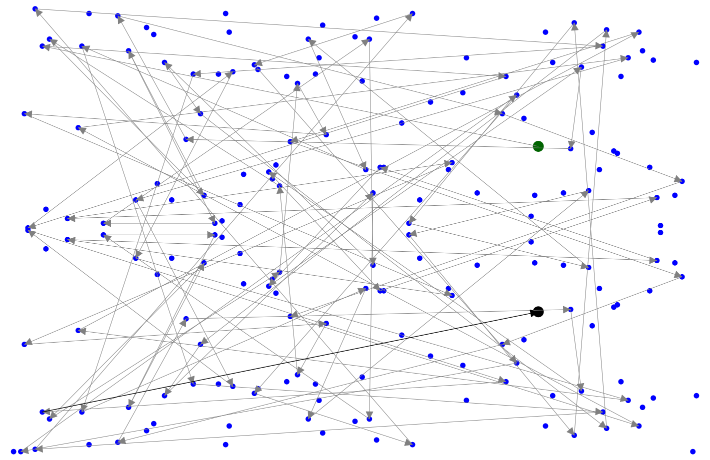


Figure 17: $95G = (146, 60) = -G$. So $96G$ would be $\mathbf{0}$. Not coincidentally, $96 = (p + 1)/2$.

$(|G|-1)G$

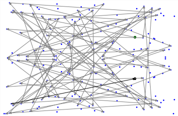


Figure 17: $95G = (146, 60) = -G$. So $96G$ would be $\mathbf{0}$. Not coincidentally, $96 = (p+1)/2$.

1. I point this out as a kind of foreshadowing. It isn't anything to actually learn or understand at this point.
2. In the context where a is a member of a group, $|a|$ is the "order of a". It is the number time times the group operation is peformed on a that brings you to the identity element.
3. I can't really draw an arrow to $\mathbf{0}$. So I stopped at 95.

Why elliptic curves

Isn't this a lot of trouble and complexity when we already can do what we need in a multiplicative integer group, $\mathbb{Z}_p^\times$?

Smaller keys

We can get 128-bit security with 256-bit keys over elliptic curves, while we need at least 3072-bit keys over $\mathbb{Z}_p^\times$.

Smaller keys

We can get 128-bit security with 256-bit keys over elliptic curves, while we need at least 3072-bit keys over $\mathbb{Z}_p^\times$.

1. This is the most visible and salient advange of ECC, but it really isn't the most important.

Fewer attacks on DLP

- Pollard's ρ (rho) is a non-polynomial time attack on the discrete logarithm;
- Pollard's ρ works both in $\mathcal{E}_p$ and $\mathbb{Z}_p^\times$;
- There are faster (but still non-polynomial) solutions to the DLP in $\mathbb{Z}_p^\times$.
- Pollard's ρ is believed to the the fastest way to solve the DLP in $\mathcal{E}_p$.

The right level of abstraction

- The better than Pollard's ρ approaches exploit the fact that there are relationships among integers that are not part of the group operation.
- Those relationships don't get translated to relationships between points on an elliptic curve.
- It would be possible to abstract a finite cyclic group to its pure form, but that makes the memory requirement needed to perform group operations at $\mathcal{O}(p^2)$.

2024-05-18

- The better than Pollard's ρ approaches exploit the fact that there are relationships among integers that are not part of the group operation.
- Those relationships don't get translated to relationships between points on an elliptic curve.
- It would be possible to abstract a finite cyclic group to its pure form, but that makes the memory requirement needed to perform group operations at $\mathcal{O}(p^2)$.

1. An untrue example is that with two numbers we can say which is larger, but that kind of relationship doesn't exist among points. (We can't actually say which number is larger.)
2. There might be some faster way, but I am describing the best algorithm off of the top of my head.

Too abstract

$\circ$	0	p	w	q	r	f	g	h	v	j
0	0	p	w	q	r	f	g	h	v	j
r	r	f	g	h	v	j	0	p	w	q
p	p	w	q	r	f	g	h	v	j	0
j	j	0	p	w	q	r	f	g	h	v
f	f	g	h	v	j	0	p	w	q	r
q	q	r	f	g	h	v	j	0	p	w
h	h	v	j	0	p	w	q	r	f	g
g	g	h	v	j	0	p	w	q	r	f
v	v	j	0	p	w	q	r	f	g	h
w	w	q	r	f	g	h	v	j	0	p

Figure 18: Group operation table with operation $\circ$ and identity element 0. The only way to compute $X \circ Y$ is to consult the table.

Extras

Point at infinity

Point at infinity

Figure 19: The additive identity, $\mathbf{0}$ (or $\mathcal{O}$), is known as the point at infinity

Point at infinity

Figure 19: The additive identity, **0** (or $\mathcal{O}$), is known as the point at infinity

1. Pointing hand src: https://www.needpix.com/photo/825401/
2. At first I wanted to an Infinity Motors™ vehical, but I didn't want to spend more time on image editing.

Definition

Projective geometry offers useful and coherent ways of talking about points at infinity. It had its start in how to map things in the three dimensional space into two dimensions.

A new perspective

Figure 20: Perugino's *Delivery of the Keys*, c1481

Figure 20: Perugino's *Delivery of the Keys*, c1481

1. Higher resolution availabe, but I didn't want the slides to get to big.
2. Jesus is deliverying the keys to Peter. It is very important that the keys not be at the point at infinity.

Punto del Centro

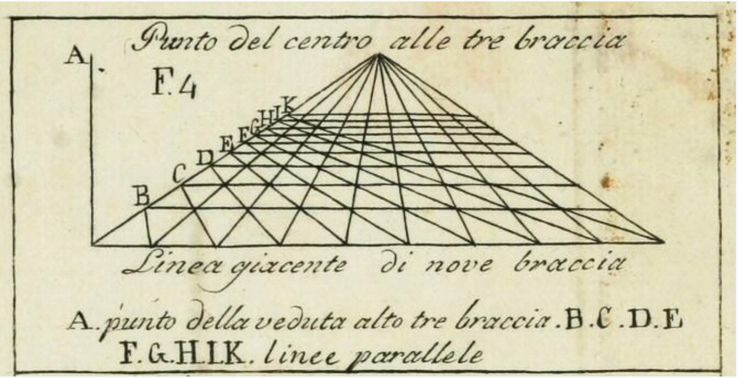


Figure 21: From Leon Battista Alberti's *De Pintura*, 1435

Punto del Centro

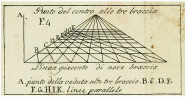


Figure 21: From Leon Battista Alberti's *De Pintura*, 1435

1. Alberti literally wrote to book on linear perspective and the geometry of using vanishing points.

Also a cryptographer

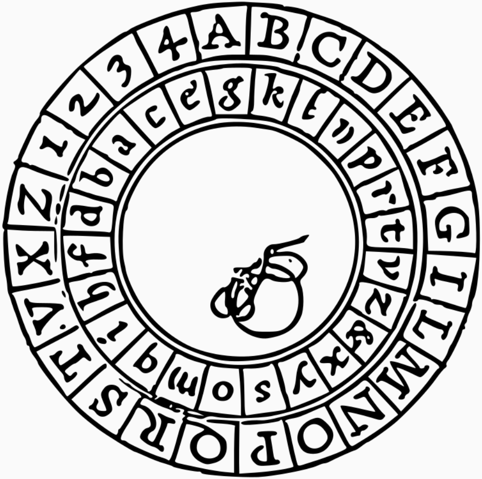


Figure 22: Alberti cipher disk. Alberti developed one of the first polyalphabetic ciphers.

Also a cryptographer

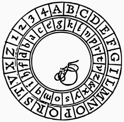


Figure 22: Alberti cipher disk. Alberti developed one of the first polyalphabetic ciphers.

1. Points at infinity and elliptic curves came into cryptography in the late 20th century. But both had been around for a while.

Resources

Resources

- These slides
- Sources

References I

[AK96] Ross Anderson and Markus Kuhn. **"Tamper resistance — a cautionary note."** In: *Proceedings of the second Usenix workshop on electronic commerce*. Vol. 2. 1996, pp. 1–11. URL: http://www.cl.cam.ac.uk/users/mgk25/tamper.pdf.

[Gen+16] Daniel Genkin et al. ***ECDH Key-Extraction via Low-Bandwidth Electromagnetic Attacks on PCs.*** Cryptology ePrint Archive, Report 2016/129. 2016. URL: http://eprint.iacr.org/2016/129.

[KJJ99] Paul Kocher, Joshua Jaffe, and Benjamin Jun. **"Differential Power Analysis."** In: *Advances in Cryptology — CRYPTO' 99*. Ed. by Michael Wiener. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 388–397. URL: https://link.springer.com/chapter/10.1007/3-540-48405-1_25.

References II

[Koc96] Paul C. Kocher. **"Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems."** In: *Advances in Cryptology – CRYPTO '96*. Vol. 1109. Lecture Notes in Computer Science. Springer, 1996, pp. 104–113. DOI: 10.1007/3-540-68697-5_9.