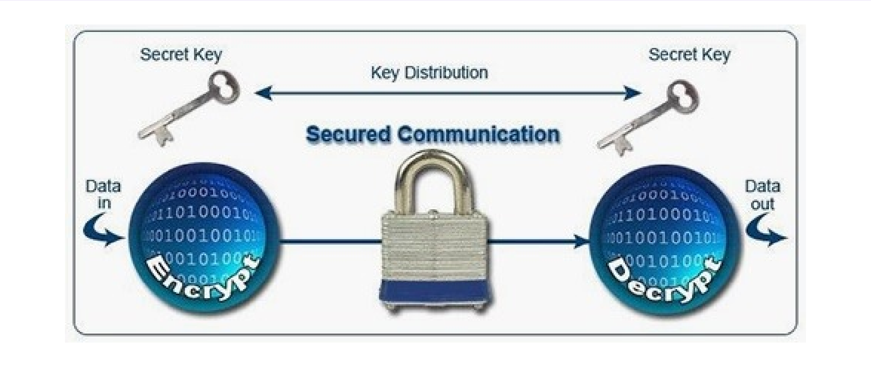
DOTA2024:4++ Defense of the Ancients Fourth topic - Crypto - extra math slides

Hugh Anderson

National University of Singapore
School of Computing

July, 2024

Keys...


Secret Key
Key Distribution
Secret Key
Secured Communication
Data in
Encrypt
Decrypt
Data out

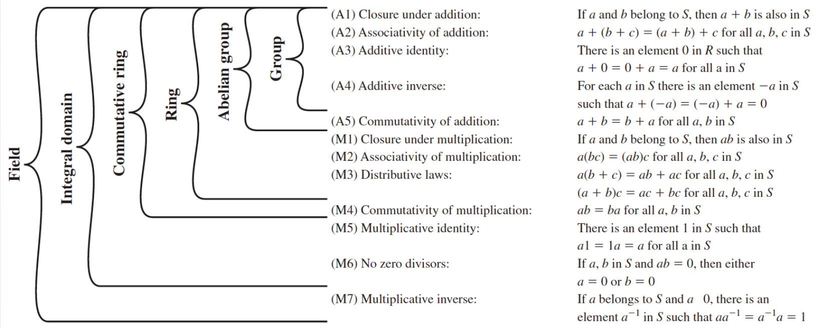
Outline

1. **Building blocks for “hardness” operations in $\mathbb{Z}_p$ and $\mathbb{Z}_N$**
   - Cyclic groups, math operations...
   - Algorithms, with complexity
   - Problems in $\mathbb{Z}_p$ and $\mathbb{Z}_N$

Outline

1. **Building blocks for “hardness” operations in $\mathbb{Z}_p$ and $\mathbb{Z}_N$**
   - Cyclic groups, math operations...
   - Algorithms, with complexity
   - Problems in $\mathbb{Z}_p$ and $\mathbb{Z}_N$

Abstract algebra concepts...



(Diagram from Stallings, a crypto book)

Fields, rings and groups

In crypto, we often work with finite, cyclic mathematical structures. These have been studied for centuries, and it is appropriate for us to know about the mathematical domain in which we must work.

I say *must*, because without knowing the hard facts/limits of the mathematical structures we must deal with, we have *nothing* (!)

Fermat's (little) theorem for $\mathbb{Z}_p$: (primes)

When p is a prime, if a is any non-zero number less than p, then

$$a^{p-1} \bmod p = 1$$

A table showing powers-of-a for $p = 11$:

a	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}
2	4	8	5	10	9	7	3	6	1
3	9	5	4	1	3	9	5	4	1
4	5	9	3	1	4	5	9	3	1
5	3	4	9	1	5	3	4	9	1
6	3	7	9	10	5	8	4	2	1
7	5	2	3	10	4	6	9	8	1
8	9	6	4	10	3	2	5	7	1
9	4	3	5	1	9	4	3	5	1
10	1	10	1	10	1	10	1	10	1

Fermat's little theorem

Observations

- For $p = 11$ the value is always 1 when the power gets to 10
- Sometimes the value gets to 1 earlier
- Lengths of runs are always numbers that divide evenly into 10
- A value of a for which the whole row is needed is called a *generator*. 2, 6, 7, and 8 are generators.

Simplifying expressions

Because a to a power mod p always starts repeating after the power reaches $p - 1$, you can do this:

$$a^x \bmod p = a^{x \bmod (p-1)} \bmod p$$

Thus modulo p in the expression requires modulo $p - 1$ in the exponent. For $p = 13$, then

$$a^{29} \bmod 13 = a^{29 \bmod 12} \bmod 13 = a^5 \bmod 13$$

Another example result $= 7^{1215} \bmod 13$

$\mathbb{Z}_p$

a big number...

622470275067322737046556455907979268906239864832921913090207877109248699107274058 7
065198907810173838994978267934813009677708927826601313557773653614840447838008512 2
281739226134142137076240050702683456450161478881858016233581815507729190060733863 8
109858209984177537766703728681473967012031571239691400018482234035235590645515566 7
534102473964535413774125836762607063593310484032937790537046487710697641318654226 2
299505280557584280574185802694213299802280179325494560628948940739344482284649151 1
971411686989595879473202428574269018023244940256710105083114967356334295809219455 7
111911312469746271731112427925544533211650491453007724199618935729850860520678012 0
898808355252223419405145855673208684204238889320915704079986487190106499123086028 8
657545878548380319021099351102645038915441458725807478306222940669780470596980888 8
224976779404912792017633095411318555938776800816778624695807909497057871925962771 2
779630348778181410614737537090462719599558908727684699 43 mod 13 = 5

$\text{result} = 7^{1215} \bmod 13$

$$
\begin{aligned}
&= 7^{1215} \bmod 13 \\
&= 7^{1215 \bmod 12} \bmod 13 \\
&= 7^{3} \bmod 13 \\
&= 343 \bmod 13 \\
&= 5
\end{aligned}
$$

We can do BIG NUMBER maths without calculating BIG numbers!

Another example result $= 7^{1215} \bmod 13$

a big number...

6224702750673227370465564559079792689062398648329219130902078771092486991072740587
0651989078101738389949782679348130096777089278266013135577736536148404478380085122
2817392261341421370762400507026834564501614788818580162335818155077291900607338638
1098582099841775377667037286814739670120315712396914000184822340352355906455155667
5341024739645354137741258367626070635933104840329377905370464877106976413186542262
2995052805575842805741858026942132998022801793254945606289489407393444822846491511
9714116869895958794732024285742690180232449402567101050831149673563342958092194557
1119113124697462717311124279255445332116504914530077241996189357298508605206780120
8988083552522234194051458556732086842042388893209157040799864871901064991230860288
6575458785483803190210993511026450389154414587258074783062229406697804705969808888
2249767794049127920176330954113185559387768008167786246958079094970578719259627712
77963034877818141061473753709046271959955890872768469943 mod 13 = 5

result $= 7^{1215} \bmod 13$

$$
\begin{aligned}
&= 7^{1215} \bmod 13 \\
&= 7^{1215 \bmod 12} \bmod 13 \\
&= 7^{3} \bmod 13 \\
&= 343 \bmod 13 \\
&= 5
\end{aligned}
$$

We can do BIG NUMBER maths without calculating BIG numbers!

Another example result $= 7^{1215} \bmod 13$

$\mathbb{Z}_p$

a big number...

6224702750673227370465564559079792689062398648329219130902078771092486991072740587
0651989078101738389949782679348130096777089278266013135577736536148404478380085122
2817392261341421370762400507026834564501614788818580162335818155077291900607338638
1098582099841775377667037286814739670120315712396914000184822340352355906455155667
5341024739645354137741258367626070635933104840329377905370464877106976413186542262
2995052805575842805741858026942132998022801793254945606289489407393444822846491511
9714116869895958794732024285742690180232449402567101050831149673563342958092194557
1119113124697462717311124279255445332116504914530077241996189357298508605206780120
8988083552522234194051458556732086842042388893209157040799864871901064991230860288
6575458785483803190210993511026450389154414587258074783062229406697804705969808888
2249767794049127920176330954113185559387768008167786246958079094970578719259627712
7796303487781814106147375370904627195995589087276846994 3 mod 13 = 5

result $= 7^{1215} \bmod 13$

$$
\begin{aligned}
&= 7^{1215} \bmod 13 \\
&= 7^{1215 \bmod 12} \bmod 13 \\
&= 7^{3} \bmod 13 \\
&= 343 \bmod 13 \\
&= 5
\end{aligned}
$$

We can do BIG NUMBER maths without calculating BIG numbers!

Another example result $= 7^{1215} \bmod 13$

a big number...

6224702750673227370465564559079792689062398648329219130902078771092486991072740587
0651989078101738389949782679348130096777089278266013135577736536148404478380085122
2817392261341421370762400507026834564501614788818580162335818155077291900607338638
1098582099841775377667037286814739670120315712396914000184822340352355906455155667
5341024739645354137741258367626070635933104840329377905370464877106976413186542262
2995052805575842805741858026942132998022801793254945606289489407393444822846491511
9714116869895958794732024285742690180232449402567101050831149673563342958092194557
1119113124697462717311124279255445332116504914530077241996189357298508605206780120
8988083552522234194051458556732086842042388893209157040799864871901064991230860288
6575458785483803190210993511026450389154414587258074783062229406697804705969808888
2249767779404912792017633095411318555938776800816778624695807909497057871925962771
2779630348778181410614737537090462719599558908727684699 43 mod 13 = 5

result $= 7^{1215} \bmod 13$

$$
\begin{aligned}
&= 7^{1215} \bmod 13 \\
&= 7^{1215 \bmod 12} \bmod 13 \\
&= 7^{3} \bmod 13 \\
&= 343 \bmod 13 \\
&= 5
\end{aligned}
$$

We can do BIG NUMBER maths without calculating BIG numbers!

Another example result $= 7^{1215} \bmod 13$

a big number...

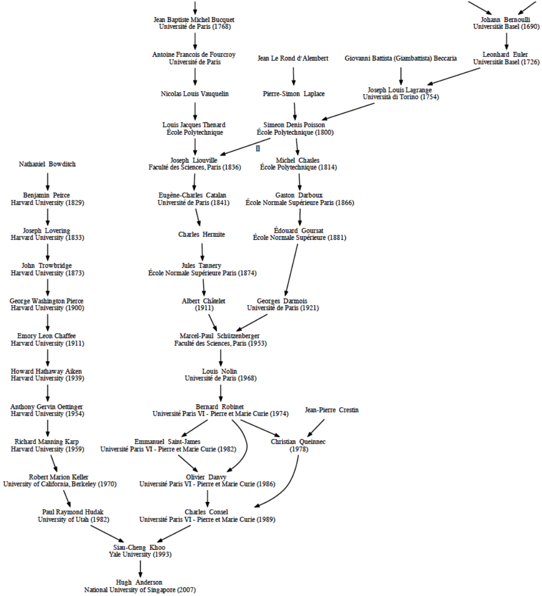
6224702750673227370465564559079792689062398648329219130902078771092486991072740587
0651989078101738389949782679348130096777089278266013135577736536148404478380085122
2817392261341421370762400507026834564501614788818580162335818155077291900607338638
1098582099841775377667037286814739670120315712396914000184822340352355906455155667
5341024739645354137741258367626070635933104840329377905370464877106976413186542262
2995052805575842805741858026942132998022801793254945606289489407393444822846491511
9714116869895958794732024285742690180232449402567101050831149673563342958092194557
1119113124697462717311124279255445332116504914530077241996189357298508605206780120
8988083552522234194051458556732086842042388893209157040799864871901064991230860288
6575458785483803190210993511026450389154414587258074783062229406697804705969808888
2249767794049127920176330954113185559387768008167786246958079094970578719259627712
779630348778181410614737537090462719599558908727684699943 mod 13 = 5

result $= 7^{1215} \bmod 13$

$$
\begin{aligned}
&= 7^{1215} \bmod 13 \\
&= 7^{1215 \bmod 12} \bmod 13 \\
&= 7^{3} \bmod 13 \\
&= 343 \bmod 13 \\
&= 5
\end{aligned}
$$

We can do BIG NUMBER maths without calculating BIG numbers!

Leonhard Euler (1707-1783)



Euler's theorem: (composites)

If N is any positive integer and a is any positive integer less than N with no divisors in common with N, then

$$a^{\phi(N)} \bmod N = 1$$

where $\phi(N)$ is the *Euler phi (totient) function*:

$$\phi(N) = N(1 - 1/p_1)\ldots(1 - 1/p_m)$$

and $p_1, \ldots, p_m$ are all the prime numbers that divide evenly into N.

If N is a prime, then using the formula, we have

$$\phi(N) = N(1 - \frac{1}{N}) = N(\frac{N-1}{N}) = N - 1$$

We see that Fermat's result is a special case of Euler's:

$$a^{\phi(N)} \bmod N = a^{N-1} \bmod N = 1$$

Special cases

Special case #2 (RSA-style composites)

Another special case needed for RSA comes when the modulus is a product of two (different) primes: $N = pq$. Then

$$\phi(N) = N(1 - \frac{1}{p})(1 - \frac{1}{q}) = (p-1)(q-1)$$

and so we have

$$a^{(p-1)(q-1)} \bmod pq = 1$$

(if a has no divisors in common with pq and p, q prime)

On the next slide we illustrate Euler with $N = 15$

The table illustrates Euler's theorem for $N = 15 = 3 \times 5$, with

$$\phi(15) = 15(1 - \frac{1}{3})(1 - \frac{1}{5}) = (3-1)(5-1) = 8$$

Notice here that a 1 is reached when the power is 8, but only for numbers with no divisors in common with 15.

Euler table for $N = 15$, $\phi(N) = 8$

a	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}
2	4	8	1	2	4	8	1	2	4	8	1	2	4
3	9	12	6	3	9	12	6	3	9	12	6	3	9
4	1	4	1	4	1	4	1	4	1	4	1	4	1
5	10	5	10	5	10	5	10	5	10	5	10	5	10
6	6	6	6	6	6	6	6	6	6	6	6	6	6
7	4	13	1	7	4	13	1	7	4	13	1	7	4
8	4	2	1	8	4	2	1	8	4	2	1	8	4
9	6	9	6	9	6	9	6	9	6	9	6	9	6
10	10	10	10	10	10	10	10	10	10	10	10	10	10
11	1	11	1	11	1	11	1	11	1	11	1	11	1
12	9	3	6	12	9	3	6	12	9	3	6	12	9
13	4	7	1	13	4	7	1	13	4	7	1	13	4
14	1	14	1	14	1	14	1	14	1	14	1	14	1

Arithmetic in the exponent is taken mod $\phi(N)$

If a has no divisors in common with N,

$$a^x \bmod N = a^{x \bmod \phi(N)} \bmod N.$$

so for example $a^{28} \bmod 15 = a^{28 \bmod 8} \bmod 15 = a^4 \bmod 15$.

Quadratic residues

The square root of $x \in \mathbb{Z}_p$ is... (primes)

...a number $y \in \mathbb{Z}_p$ such that $y^2 \equiv x \bmod p$. Note that $\sqrt{2} \mod 7 = 3$, but $\sqrt{3} \mod 7$ doesn't exist.

$\mathbb{Z}_p^*$ is the multiplicative (sub) group and an element $x \in \mathbb{Z}_p^*$ is called a QR (Quadratic Residue) if its square root exists in $\mathbb{Z}_p$. Each $x \in \mathbb{Z}_p^*$ has either 0 or 2 square roots.

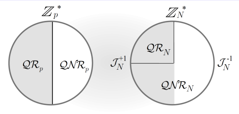
- If a is a square root of $x \mod p$, then so is $-a \mod p$.
- Exactly half of the elements in $\mathbb{Z}_p^*$ are QR.
- a is a QR $(\bmod p)$ iff $a^{\frac{p-1}{2}} \mod p = 1$ (Euler's criterion).

Computing square roots in $\mathbb{Z}_p^*$ is computationally possible. An $\mathcal{O}(n^4)$ randomized algorithm exists. Easier in some special cases of p, e.g. when $p = 3 \mod 4$, then

$$\boxed{\sqrt{x} \mod p = x^{\frac{p+1}{4}} \mod 4}$$

Given an $x \in \mathbb{Z}_p^*$, deciding whether x is a QR is computationally easy.

Identifying quadratic residues



The Legendre and Jacobi "symbol" notation

Legendre: *identifies* the QR in $\mathbb{Z}_p$; a function of a and p: $(\mathbb{Z}_p)$

$$\left(\frac{a}{p}\right) = a^{\frac{p-1}{2}} \mod p = \begin{cases} 1 & \text{if } a \text{ is a QR} \\ -1 & \text{if } a \text{ is not a QR} \\ 0 & \text{if } a = 0 \mod p \end{cases}$$

Jacobi: is a generalization of Legendre, extending it to $\mathbb{Z}_N$ (composites). $(\mathbb{Z}_N)$
Easy to compute given it's prime factorization. For example, given $N = p \times q$ with $p \neq q$, then

$$\left(\frac{a}{N}\right) = \left(\frac{a}{p}\right)\left(\frac{a}{q}\right)$$

Note that there is an algorithm that, given a and N, can find the Jacobi symbol without knowing the factorization of N. From the Jacobi symbol, we can't tell whether a is a QR, but no entry with -1 is a quadratic residue.

Outline

1. **Building blocks for "hardness" operations in $\mathbb{Z}_p$ and $\mathbb{Z}_N$**
   - Cyclic groups, math operations...
   - Algorithms, with complexity
   - Problems in $\mathbb{Z}_p$ and $\mathbb{Z}_N$

Complexity of basic arithmetic

Basic ideas...

We often work with (cyclic) groups in $\mathbb{Z}_p$ and $\mathbb{Z}_N$ with very large p and N.

Computation complexity: the following basic arithmetic operations can be efficiently computed in $\mathbb{Z}_p$ and $\mathbb{Z}_N$:

Operations	Complexity
Addition	$\mathcal{O}(m)$
Multiplication	$\mathcal{O}(m^2)$
Inverse	$\mathcal{O}(m^2)$
Exponentiation	$\mathcal{O}(m^3)$

Programming issues: Since our machines work with 32 or 64-bit integers, we have libraries such as the GMP (GNU Multiple Precision) library, and Java "BigInteger" classes.

On finding generators:

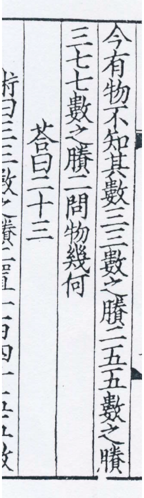

```
gens=[]
for s in facs:
    g=2
    generators=[]
    while g<s[0]:
        gen=1
        for j in s[1]:
            if (g**((s[0]-1)/j))%s[0]==1:
                gen=0
                break
        if gen:
            generators.append(g)
        g=g+1
    gens.append([s[0],generators])
```

Algorithm to find a generator modulo p:

1. Find prime factors $q_1, q_2, \ldots$ of $\phi(p)$.
2. Select a random candidate a.
3. For each factor q_i, calculate $a^{\frac{p-1}{q_i}} \bmod p$. If any one is 1, then back to step 2.
4. Otherwise a is a generator.

CRT - the Chinese Remainder Theorem

$\mathbb{Z}_N$


今有物不知其數三三數之賸二五五數之賸
三七七數之賸二問物幾何
荅曰二十三

Problem posed by Wei/Jin mathematician Sun Tzu 1700 years ago:

There are certain things whose number is unknown. Repeatedly divided by 3, the remainder is 2; by 5 the remainder is 3; and by 7 the remainder is 2. What will be the number?

The Chinese remainder theorem - nowadays

Two simultaneous congruences $n = n_1 \bmod m_1$ and $n = n_2 \bmod m_2$ are only solvable when $n_1 = n_2 \bmod (\gcd(m_1, m_2))$. The solution is unique modulo $\operatorname{lcm}(m_1, m_2)$.

It demonstrates to us that a number less than the product of two primes can be uniquely identified by its residue modulo those primes. It is useful in RSA.

CRT - the Chinese Remainder Theorem

$\mathbb{Z}_N$

Worked example

The original problem:

$$\begin{aligned} x &= 2 \bmod 3 & (1)\\ x &= 3 \bmod 5 & (2)\\ x &= 2 \bmod 7 & (3) \end{aligned}$$

From (1), we know that $x = 3n + 2$ for some n. Substituting this in (2) gives $3n = 1 \bmod 5$. This reduces to a simpler equation $n = 2 \bmod 5$ which is equivalent to $n = 5m + 2$ for some m. Substituting this back into $x = 3n + 2$ gives us $x = 15m + 8$. Substituting this in (3) gives $15m + 8 = 2 \bmod 7$, or $m = 1 \bmod 7$, i.e. $m = 7o + 1$.

From this we can see that $x = 105o + 23$. Note that $105 = \mathrm{lcm}(3, 5, 7)$ and the solutions are $23, 128$ (and so on).

Euclidean algorithm

Some theory to compute the multiplicative inverses.

To find the **gcd** of two numbers (say 2394 and 154), use the prime factorization:

$$\begin{aligned} 2394 &= 2*3*3*7*19 \\ 154 &= 2*7*11 \\ \therefore \gcd(2394,154) &= 2*7 = 14 \end{aligned}$$

Euclidean algorithm

Unfortunately, it is not easy to find the prime factors of integers. The **gcd** of two integers can however be found by repeated application of division, using the Euclidean algorithm. You repeatedly divide the divisor by the remainder until the remainder is 0. The **gcd** is the last non-zero remainder.

Example:

$$\begin{aligned} \text{dividend} &= \text{quotient} * \text{divisor} + \text{remainder} \\ 2394 &= 15*154+84 \\ 154 &= 1*84+70 \\ 84 &= 1*70+14 \\ 70 &= 5*14 \\ \therefore \gcd(2394,154) &= 14 \end{aligned}$$

Extended Euclidean algorithm

$\mathbb{Z}_N$

An interesting property

If the $\gcd(a, b) = r$ then there exist integers m and n so that $ma + nb = r$.
Use to calculate the multiplicative inverse of an element x modulo n.

The extended Euclidean algorithm

1. We begin by dividing n by x, and as we carry out each step i of the Euclidean algorithm discovering the quotient q_i, we also calculate an extra number x_i. For the first two steps $x_0 = 0$ and $x_1 = 1$.
2. For the following steps, calculate $x_i = x_{i-2} - x_{i-1}q_{i-2} \bmod n$.
3. If the last non-zero remainder occurs at step k, then if this remainder is 1, x has an inverse and it is x_{k+2}.

The inverse of 15 modulo 26, showing the method:

0 :	26	=	$1 * 15 + 11$	$x_0 = 0$		
1 :	15	=	$1 * 11 + 4$	$x_1 = 1$		
2 :	11	=	$2 * 4 + 3$	$x_2 = 0 - 1 * 1 \bmod 26$	=	25
3 :	4	=	$1 * 3 + 1$	$x_3 = 1 - 25 * 1 \bmod 26$	=	2
4 :	3	=	$3 * 1 + 0$	$x_4 = 25 - 2 * 2 \bmod 26$	=	21
				$x_5 = 2 - 21 * 1 \bmod 26$	=	7

Outline

1. Building blocks for “hardness” operations in $\mathbb{Z}_p$ and $\mathbb{Z}_N$
   - Cyclic groups, math operations...
   - Algorithms, with complexity
   - Problems in $\mathbb{Z}_p$ and $\mathbb{Z}_N$

Easy problems in $\mathbb{Z}_p$

(...assuming p is very large)

Basic math: Generating a random element, adding and multiplying elements, finding inverse, exponentiation.

QR: Testing if an element is a QR and computing its square root mod p if it is QR.

DDH: Decisional Diffie-Hellman problem[a]: Let g be a generator of a cyclic group $\mathcal{G}$. Given a positive integer 3-tuple (a, b, z), distinguish with probability greater than 0.5 whether the three outputs are from process A or B:

A: Output (g^a, g^b, g^z)
B: Output (g^a, g^b, g^{ab})

[a] For the group $\mathbb{Z}_p^*$, DDH is not hard (...use the Jacobi symbol).

Believed to be hard problems in $\mathbb{Z}_p$

But anything can change...

DL: Discrete log problem: Let g be a generator of a cyclic group. Given x, find r such that $x = g^r$.

CDH: Computational Diffie-Hellman problem: Let g be a generator of a cyclic group. Given g^a, g^b, find g^{ab}.

Cryptosystems based on the hardness of these problems:

1. Diffie-Hellman key exchange
2. Elgamal public key encryption
3. DL-based digital signatures like Elgamal, DSA, DSS...

Easy problems in $\mathbb{Z}_N$

(...assuming $N = p \times q$, and p, q are large primes)

Basic math: Generating a random element, adding and multiplying elements, finding inverse, exponentiation.

Jacobi: Finding the Jacobi symbol.

Hard problems in $\mathbb{Z}_N$

These get easier if factorization is known...

QR: Testing if an element is a QR in $\mathbb{Z}_N$.
(Computing the Jacobi symbol cannot solve the problem).

SQRT: Computing the square root of a QR in $\mathbb{Z}_N$.
Provably as hard as factoring N.
Rabin cryptosystems based on this hardness.

$\phi(N)$: Computing $\phi(N)$ - Provably as hard as factoring N. If you can efficiently compute $\phi(N)$, then you can also factor N.

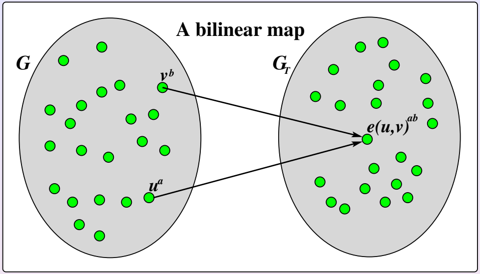
Roots: Computing the e-th roots mod N when $gcd(e, N) = 1$.
RSA cryptosystems based on this hardness.

These do not get easier if factorization is known...

DL: Discrete log problem With g being a generator of $\mathbb{Z}_N^*$, then given $x \in \mathbb{Z}_N^*$, find an r such that $x = g^r \mod N$.

CDH: Computational Diffie-Hellman problem With g being a generator of $\mathbb{Z}_N^*$, then given $g^a, g^b \in \mathbb{Z}_N^*$, find $g^{ab} \mod N$.

Bilinear maps/pairings



Pairings...

A (symmetric style) bilinear map is a pairing between two elements of a group to a second (same order) group: $G \times G \rightarrow G_T$. There are also (asymmetric) pairings like $G_1 \times G_2 \rightarrow G_T$.

Such a symmetric mapping (if it can be efficiently computed) can be used to solve DDH: $z = ab$ iff $e(g, g^z) = e(g^a, g^b)$. Originally this was the focus of bilinear maps in crypto, and this is why CDH is considered *harder* than DDH.

The other thing is that it can allow you to reduce the discrete log problem in G to a (perhaps simpler) discrete log problem in G_T: $e(g, g^a) = e(g, g)^a$.

A little more on bilinear maps/pairings

Can we have “simple” pairings?

This is easy, but unfortunately such "simple" maps are not useful for crypto. A simple map based on (say) integers might be the additive integer group $G = \langle \mathbb{Z}, + \rangle$, and $G_T = \langle \mathbb{Z}, * \rangle$, with

$$e(u, v) = 2^{uv}$$

For the additive group, u^2 means $u + u$, and it is sufficient to show that $e(u + u, v) = e(u, v)^2$ and $e(u, v + v) = e(u, v)^2$. Lets try an example:

$$e(3, 4) = 2^{12}$$

and

$$\begin{aligned} e(3+3, 4) &= e(3,4)^2 \\ &= e(3,4) \times e(3,4) \\ &= 2^{12} \times 2^{12} \\ &= 2^{24} \\ &= e(3, 4+4) \end{aligned}$$

A little more on bilinear maps/pairings

$G \rightarrow$

+	a^0	a^1	a^2	a^3	a^4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

$G_T \rightarrow$

a	a^1	a^2	a^3	a^4	a^5	a^6	a^7
2	2	4	8	5	10	9	7
3	3	9	5	4	1	3	9
4	4	5	9	3	1	4	5
5	5	3	4	9	1	5	3
6	6	3	7	9	10	5	8
7	7	5	2	3	10	4	6

"Simple" pairings with finite groups?

The groups $G = \mathbb{Z}_5^+$ and $G_T = \langle 3 \rangle_{11}$, with $e(u, v) = 3^{uv} (\bmod 11)$:

$$\begin{aligned} e(4,3) &= 3^{12} \bmod 11 \\ &= 9 \end{aligned}$$

and

$$\begin{aligned} e(4+4,3) &= e(4,3)^2 \\ &= e(4,3) \times e(4,3) \\ &= 3^{24} \bmod 11 \\ &= 4 \\ &= e(4,3+3) \end{aligned}$$

This sort of map is not too useful because the groups involving simple integer style math tend to be invertible - knowing u and $e(u, v)$ you can determine v.

The big picture..

In practice? G is an elliptic curve, G_T is a finite field...

There are getting to be more uses of pairing based crypto, including IBE (this week), simple encryption, signatures (next week) and so on. Note that

$$e(u^a, v^b) = e(u^b, v^a)$$

(one for encryption, one for decryption).

In a talk by Boneh, the above diagram showed crypto history, from one-way functions, discrete logs, pairings, LWE (Learning with Errors), through to multilinear maps for indistinguishability obfuscation. What comes next?

Presented as *one-way* functions

Three examples. Assume $N = p \times q$ (two large primes)

DL: Discrete Logarithm: With prime p and an element $g \in \mathbb{Z}_p^*$ of large order, $f(x) = g^x \mod p$. It is linear: Given $a \in \mathbb{Z}$, $f(x)$ and $f(y)$, it is easy to compute $f(ax)$ and $f(x + y)$. Its one-wayness is essential for the Diffie-Hellman key exchange protocol and Elgamal public key systems.

RSA: Let e be an integer such that $gcd(e, N) = 1$, $f(x) = x^e \mod N$. Note that given the factorization of N, the function f can be inverted efficiently. The one-wayness property is essential for the RSA public key system.

Rabin: $f(x) = x^2 \mod N$ This function is one-way if there is no efficient algorithm to factor N. The function can be inverted efficiently if the factorization of N is known. The one-wayness property is essential for Rabin's scheme.

Summary

The framework for “simple” asymmetric systems...

- Multiplicative inverses may not exist in modulo arithmetic. For a modulus N and a number x, x^{-1} exists iff $gcd(x, N) = 1$.
- CRT gives a mapping from $\mathbb{Z}_N$ to $\mathbb{Z}_p \times \mathbb{Z}_q$.
- Some problems are easy in $\mathbb{Z}_p$ but the same problems become difficult in $\mathbb{Z}_N$ if the factorization of N is unknown.
- Some problems, in particular Discrete Log, are hard in both.
- Factorization and Discrete Log are the two main types of problems. Many cryptosystems depend on their “hardness”.