37181 DISCRETE MATHEMATICS

Lecture 17: the RSA cryptosystem

PLAN

- RSA

WARM-UP

I have a box, and a padlock. I put a secret message in the box (not a pigeon), put my padlock on it, keep the key, and send the box in the post.

(what happens next?)

Security assumption:

if you are given $n \in \mathbb{N}$ and told it is the product of two primes, how hard is it to find the two primes?

[1]you can stop when you hit $\lfloor \sqrt{n} \rfloor$. Why?

Security assumption:

if you are given $n \in \mathbb{N}$ and told it is the product of two primes, how hard is it to find the two primes?

- compute $\varphi(n)$ (recall end of Lectures 16 - how hard is it to compute $\varphi(n)$?)
- divide by $2, 3, 5, 7, \dots$ [1]

[1] you can stop when you hit $\lfloor \sqrt{n} \rfloor$. Why?

Security assumption:

if you are given $n \in \mathbb{N}$ and told it is the product of two primes, how hard is it to find the two primes?

- compute $\varphi(n)$ (recall end of Lectures 16 - how hard is it to compute $\varphi(n)$?)
- divide by $2, 3, 5, 7, \ldots$ [1]
- Quantum computer?
  `https://en.wikipedia.org/wiki/Shors_algorithm`
  `https://en.wikipedia.org/wiki/Integer_factorization_records`

[1] you can stop when you hit $\lfloor \sqrt{n} \rfloor$. Why?

Idea: Alice (pronouns she/her) and Bob (pronouns he/him) want to communicate over open channels (the internet)

so that at the end, they have a *shared secret* nobody else knows

even though everybody can see their communication.

Alice:

- chooses p, q two large distinct (and keeps them secret).
- She computes $n = pq$.
- She publishes n on her webpage.
- She secretly computes $\varphi(n) = (p-1)(q-1)$ (that's easy for her)

Alice:

- chooses p, q two large distinct (and keeps them secret).
- She computes $n = pq$.
- She publishes n on her webpage.
- She secretly computes $\varphi(n) = (p-1)(q-1)$ (that's easy for her)
- Then she chooses (any) $e \in \mathbb{Z}^*_{\varphi(n)}$.
- She computes the multiplicative inverse mod $\varphi(n)$: that is, she computes d so that $de \equiv 1 \bmod \varphi(n)$.

Alice:

- chooses p, q two large distinct (and keeps them secret).
- She computes $n = pq$.
- She publishes n on her webpage.
- She secretly computes $\varphi(n) = (p-1)(q-1)$ (that's easy for her)
- Then she chooses (any) $e \in \mathbb{Z}^*_{\varphi(n)}$.
- She computes the multiplicative inverse mod $\varphi(n)$: that is, she computes d so that $de \equiv 1 \bmod \varphi(n)$.
- Since Alice worked hard on the worksheets and lectures 15 and 16, all of that is easy for her to do.

Alice:

- chooses p, q two large distinct (and keeps them secret).
- She computes $n = pq$.
- She publishes n on her webpage.
- She secretly computes $\varphi(n) = (p-1)(q-1)$ (that's easy for her)
- Then she chooses (any) $e \in \mathbb{Z}^*_{\varphi(n)}$.
- She computes the multiplicative inverse mod $\varphi(n)$: that is, she computes d so that $de \equiv 1 \bmod \varphi(n)$.
- Since Alice worked hard on the worksheets and lectures 15 and 16, all of that is easy for her to do.
- Finally, she publishes e on her webpage as well.

Public: n, e Secret: p, q, d.

RSA

Bob:

- wants to send Alice a secret message which will be a number in $\mathbb{Z}_n$. He knows n since it is public.
- He picks m relatively prime to n.

Bob:

- wants to send Alice a secret message which will be a number in $\mathbb{Z}_n$. He knows n since it is public.
- He picks m relatively prime to n.
- He computes $[m^e]_n$ (he might need repeated squaring). Remember e, n are both public.
- He sends the number $c = [m^e]_n$ to Alice over the internet. Assuming raising to a really high power e then reducing mod n "mixes up" the number m, it should not be obvious what m is when anyone else sees c in the open channels.

Public: $c(= [m^e]_n)$ Secret: m.

Alice:

- She knows d, p, q so she takes the number c from Bob and computes
$$c^d \equiv (m^e)^d = m^{ed} \bmod n$$
- Remember $ed \equiv 1 \bmod \varphi(n)$ so $ed = 1 + s\varphi(n)$
- So $m^{ed} = m^{1+s\varphi(n)} = m.(m^{\varphi(n)})^s$ but by Euler's theorem $m^{\varphi(n)} \equiv 1 \bmod n$.
- So Alice just computed $m = [m^{ed}]_n$.

Practice:

(You will need a copy of the RSA procedure in front of you to follow this.)

1. Alice constructs an RSA system by choosing $n = 74$ and $e = 7$. What is her corresponding value for d?

Practice:

(You will need a copy of the RSA procedure in front of you to follow this.)

1. Alice constructs an RSA system by choosing $n = 74$ and $e = 7$. What is her corresponding value for d?

2. Then Bob wants to send $m = 21$. What c does he send?

Practice:

(You will need a copy of the RSA procedure in front of you to follow this.)

1. Alice constructs an RSA system by choosing $n = 74$ and $e = 7$. What is her corresponding value for d?

2. Then Bob wants to send $m = 21$. What c does he send?

3. Then do the steps Alice would do to decode c.

Practice:

(You will need a copy of the RSA procedure in front of you to follow this.)

1. Alice constructs an RSA system by choosing $n = 74$ and $e = 7$. What is her corresponding value for d?

2. Then Bob wants to send $m = 21$. What c does he send?

3. Then do the steps Alice would do to decode c.

$\varphi(74) = \varphi(2)\varphi(37) = 36, 36 = 5.7 + 1, 1 = 36 - 5.7$ so $d = -5 = 31$.
$c = 21^4 21^2 21^1 = 9(-3)21 = 9.(-63) = 9.11 = 99 = 25$. Alice computes
$25^d = 25^{31} = 25^{16} 25^8 25^4 25^2 25 = 21$

Alice constructs an RSA system and publishes $n = 77$ and $e = 43$. Bob then sends Alice the encoded message $c = 23$. What was Bob's intended message?

WAYS TO CHEAT

https://www.mtholyoke.edu/courses/quenell/s2003/ma139/js/powermod.html

mtholyoke.edu/courses/quenell/s2003/ma139/js/powermod.html

MATH 139

PowerMod Calculator

Computes (base)$^{\text{(exponent)}}$ mod (modulus) in log(exponent) time.

Base: 21	Exponent: 7	Modulus: 74
Compute	b^e MOD m =	25

The program is written in JavaScript, and runs on the client computer. Most implementations seem to handle numbers of up to 16 digits correctly.

mtholyoke.edu/courses/quenell/s2003/ma139/js/powermod.html

MATH 139

PowerMod Calculator

Computes (base)$^{\text{(exponent)}}$ mod (modulus) in log(exponent) time.

Base: 25	Exponent: 31	Modulus: 74
Compute	b^e MOD m =	21

The program is written in JavaScript, and runs on the client computer. Most implementations seem to handle numbers of up to 16 digits correctly.

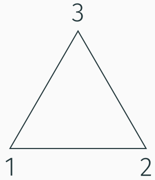
Define a relation $\mathcal{T}$ on the set of all finite length (including 0) binary strings by

$$a\mathcal{T}b \quad \text{if} \quad a \text{ is a factor of } b.$$

(b) Draw the *Hasse diagram*[2] for $\mathcal{T}$ on the set of all binary strings of length $0, 1, 2$ and 3.

[2]this was in Lecture 9.

HOMEWORK SHEET 9: TRIANGLE PROBLEM



(a) Is $*(\sigma, \tau)$ the same motion as $*(\tau, \sigma)$?

(b) What is $*(\tau, \tau)$?

(c') What is $*(\sigma, \sigma)$?

(d') Prove that τ, σ both have inverses.

NEXT

- Graph theory