SIGPwny

General FA2025 • 2025-10-23

Cryptography II

Ahmed Alkhalawi

Ahmed Alkhalawi

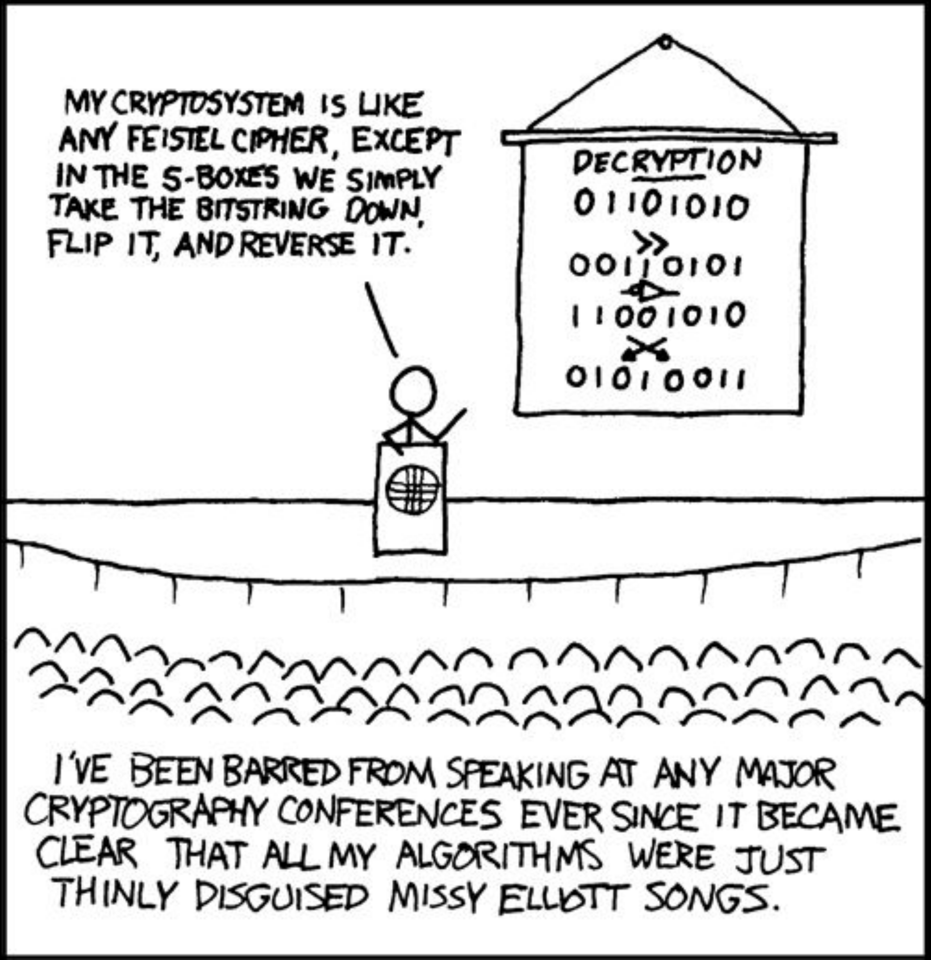
- Helper
- Math/CS major
- I like bouldering

Announcements

- We will be playing **osu!gaming ctf** tomorrow!
    - Likely Siebel CS 2406 as usual, may change based on availability!

ctf.sigpwny.com

sigpwny{R0n_w4s_wr0ng}


MY CRYPTOSYSTEM IS LIKE ANY FEISTEL CIPHER, EXCEPT IN THE S-BOXES WE SIMPLY TAKE THE BITSTRING DOWN, FLIP IT, AND REVERSE IT.
DECRYPTION
01101010
>>
00110101
11001010
01010011
I'VE BEEN BARRED FROM SPEAKING AT ANY MAJOR CRYPTOGRAPHY CONFERENCES EVER SINCE IT BECAME CLEAR THAT ALL MY ALGORITHMS WERE JUST THINLY DISGUISED MISSY ELLIOTT SONGS.

Asymmetric Encryption

- Public key
  - Intentionally broadcast for people to use
  - Anyone can use to encrypt a message to send us
- Private key
  - Keep to yourself
  - Use to decrypt other people's messages to us
- Examples: RSA, Diffie Hellman, Elliptic Curves

Quick Maths

Calculate 5^5 mod 13

5 ^ 2 mod 17 = 25 mod 17 = 8 mod 17

5 ^ 4 mod 17 = (5 ^ 2) ^ 2 mod 17 = 8 ^ 2 mod 17 = 13 mod 17

5 ^ 5 = 5 ^ 4 * 5 mod 13 = 13 * 5 mod 17 = 14 mod 17

In general, we can compute x^e mod n in log(e) time.

But how would we find c such that c ^ 5 = 14 mod 17?

Totients and Euler's Function

- $\phi(n)$ = the number of numbers ≥ 0 that share no factors with n
- We call $\phi(n)$ Euler’s “totient” function
- Euler’s Theorem: If a and n share no factors, then
  a^$\phi(n) \equiv 1 \pmod{n}$
- This theorem is the basis for the RSA cryptosystem

More Maths

Now we can solve our problem!

We want to solve x^5 = 14 mod 17

ϕ(17) = 16

By Euler’s theorem x^16=1

We know 5*13 mod 16 = 65 mod 16 = 1 mod 16

This means there exists k with 5*13 = 16*k + 1

Hence, x = (x^16)^k*x = x^(16k+1) = (x^5)^13 = 14^13 = 5 mod 17

What if we change 17?

In general, n = p ^ a * q ^ b ….

Then ϕ(n) = (p-1)*(q-1) * …. * p ^ (a-1) * q^ (b-1) ….

In particular, if n = p*q, ϕ(n) = (p-1) * (q-1)

In other words, to compute ϕ(n) you have to factor n.

But factoring is hard!

We need ϕ(n) to solve x ^ e = c mod n, so solving it is hard.

The RSA Cryptosystem

- Say Bob wants to send the message ‘FLAG’ to Alice
- First he turns the message into a number m
- Alice chooses a public exponent e, usually $e = 2^{16} + 1 = 65537$
- Alice generates large (> 256 or even > 512 bits) secret primes p, q
- Alice then calculates $n = p * q$ and releases it as a public key. Then they calculate $\phi(n) = (p - 1) * (q - 1)$ as a private key.
- Knowing $\phi(n)$, compute d such that $ed \equiv 1 \pmod{\phi(n)}$
- If you know $\phi(n)$, this is fast using the Extended Euclidian Algorithm
- Bob computes $c = m^e$ and sends it to Alice
- Then Alice can compute $c^d \equiv m \pmod{n}$
- $c^d \equiv (m^e)^d \equiv m^{ed} \equiv m^{1+k\cdot\phi(n)} \equiv m * (m^{\phi(n)})^k \equiv m * 1^k \equiv m \pmod{n}$

GCDs

Recall gcd(a, b) is the largest number which divides a and b

If we could find t such that $p \mid t$, $q \nmid t$, then gcd(n, t) = p

This is useful because we can compute gcds quickly with the Euclidean Algorithm: gcd (a, b) = gcd(a-b, b)

Example:
gcd(807, 1443)

= gcd (807, 1443 - 807) = gcd(807, 636)

= gcd(171, 636) = gcd(171, 123) = gcd(123, 48) = gcd(48, 27)

= gcd(21, 27) = gcd(21, 6) = gcd(3, 6) = 3

GCDs

In 2012, researchers collected 11 million RSA public keys from the web, and took the gcd of each two.

They found that 21419 had repeated primes, and hence were able to easily factor them to break the encryption

Pollard p-1 Algorithm

If p - 1 is powersmooth (all of its prime power divisors are smaller than some B), then this lets us factor n=pq for any q

We take M to be the product of the largest prime powers less than B for each prime.

Then ϕ(p) = p - 1 divides M, so a^M = (a^k)^ϕ(p) = 1 mod p

This means that p will divide gcd(a^M - 1, n) so it will either be p or n

If it is n then we can reduce B or exponents in M

A 2016 paper showed that 4% of keys generated by the Infineon JCOP 80K are susceptible.

Polynomial GCDs

We can do the same thing to compute GCDs of Polynomials!

For example, the gcd of x^2-2x+1 = (x-1)^2 and x^2-1 = (x-1)(x+1) is x-1.

Or, gcd(x^2 - 2x +1, x^2 - 1) = gcd(x^2 - 2x +1 - (x^2 - 1), x^2 - 1)

= gcd(-2x + 2, x^2 - 1) = gcd(-2x + 2, x^2 - 1 - (-2x+2) * (-x/2))

= gcd(-2x + 2, x - 1) = x - 1

We already know x^e - c has m as a root mod n, so if we find another such polynomial we can immediately obtain m.

Related Message Attack.

Say we know the ciphertexts c_1, c_2 of 2 messages m, am+b

then we can decrypt m by taking the GCD of x^e - c_1, (ax+b)^e-c2

For example, this could occur if we encrypt a message followed by the time it was sent, and Bob sends the same message twice.

The Coppersmith short-pad attack strengthens this, so that if a short padding is added to the end of a message and e is small you can decrypt it even without knowing the padding.

Mitigation: Padding

- Modern systems usually introduce randomized padding before encrypting a message.
- Usually XOR a the padding with the message
- This stops algebraic attacks that don’t factor n
- They also make it impossible to guess the message then verify it

More Attacks

- Number Field Sieve
    - Fastest way to factor large numbers
    - The record is a 250 digit modulus being factored in 2020

- p-1 attack variants
    - Williams p+1 algorithm works if p+1 is smooth
    - Elliptic curve factoring is a generalization which tries to find any smooth number near p. It’s time relies only on the smallest prime so can be better than NFS.

More Attacks

- Coppersmith Method
  - If p is a b-bit prime this lets you factor pq if you know the the b/4 most or least significant bits of p
  - Used in 2017 Roca vulnerability which caused Estonia to recall 750000 national IDs
- Wiener's Attack
  - If d is small, can be used to obtain d without factoring n
- Side channels
  - Timing attacks

Challenges

- Cryptohack!

Learn with fantastic lessons and challenges, and earn points on PwnyCTF while you're at it!

ctf.sigpwny.com/challenges#Meetings/CryptoHack

Next Meetings

`2025-10-24` • **This Friday**

- osu!gaming ctf
- Meet us in Siebel CS 2406 (may change, see Discord) for osu!gaming ctf!

`2025-10-26` • **This Sunday**

- Python Jails
- Learn about escaping python sandboxes to get arbitrary code execution on the main machine!

ctf.sigpwny.com

sigpwny{R0n_w4s_wr0ng}

Meeting content can be found at sigpwny.com/meetings.

SIGPwny