SIGPwny

FA2024 Week 08 • 2024-10-27

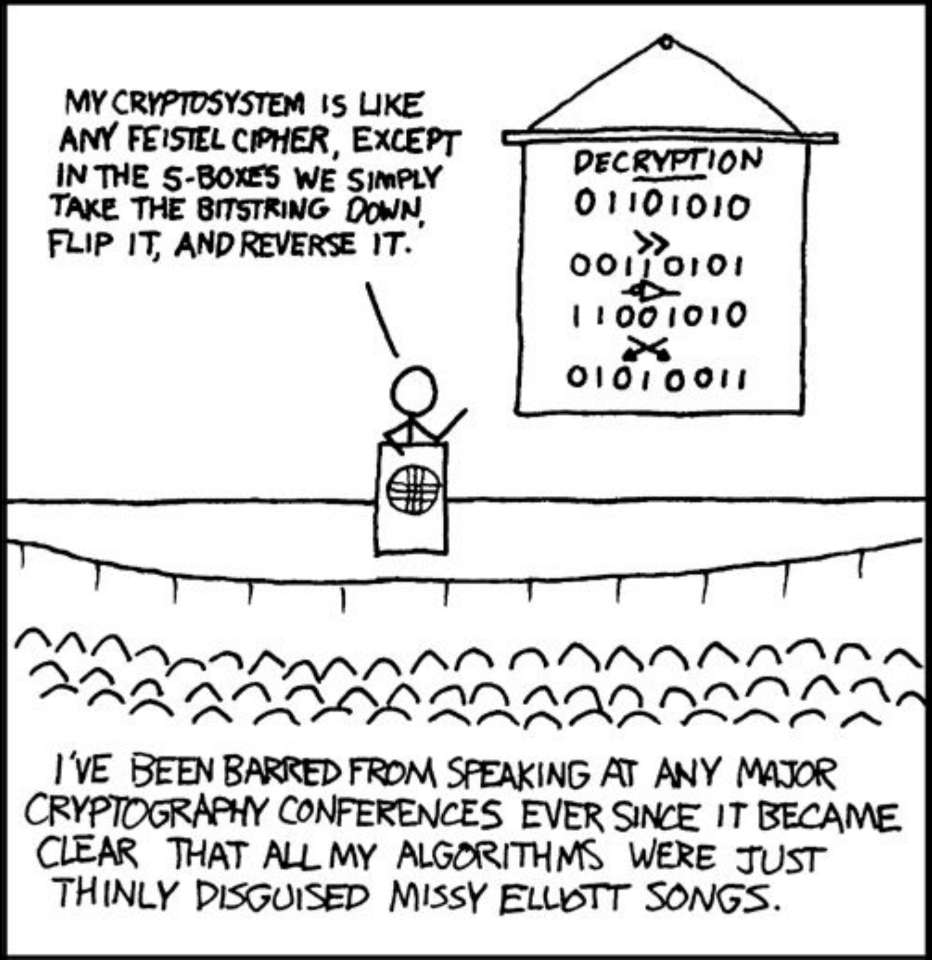
Cryptography II

Emma and Richard

Announcements

ctf.sigpwny.com

sigpwny{b1g_numb3r5_g0_brr}



Overview

- Modular Arithmetic
  - Chinese Remainder Theorem
  - Factoring
- RSA
  - Common attacks

Small vs Large n

- Modular arithmetic
  - Arithmetic mod n is the remainder after division with n
  - Information lost when finding values for mod n

Small vs Large n

- Suppose I ask you to find 4 * 4 mod 3
  - The result is 1, pretty straightforward if you're comfortable with modular arithmetic
- Now I tell you $x \equiv 1 \bmod 3$ and ask you to find x / 4
  - Much harder

Small vs Large n

- Now suppose I ask you to find 4 * 4 mod 20
  - The result is 16, also pretty straightforward
- Now I tell you x ≡ 16 mod 20 and ask you to find x / 4
  - Much easier by comparison!
- What can we do with this?

The Chinese Remainder Theorem

- Ancient theorem dating back to 3rd century
- Let's try to find x such that $0 \leq x \leq 105$. Additionally, we are given the following information

$$x \equiv 2 \pmod{3}$$

$$x \equiv 3 \pmod{5}$$

$$x \equiv 2 \pmod{7}$$

- According to the Chinese Remainder Theorem, $x \equiv 23 \ (\text{mod } 3 * 5 * 7 = 105)$

The Chinese Remainder Theorem

- More generally speaking, let's say we have:

$$x \equiv n_1 \pmod{p_1}$$
$$x \equiv n_2 \pmod{p_2}$$
$$\dots$$
$$x \equiv n_k \pmod{p_k}$$

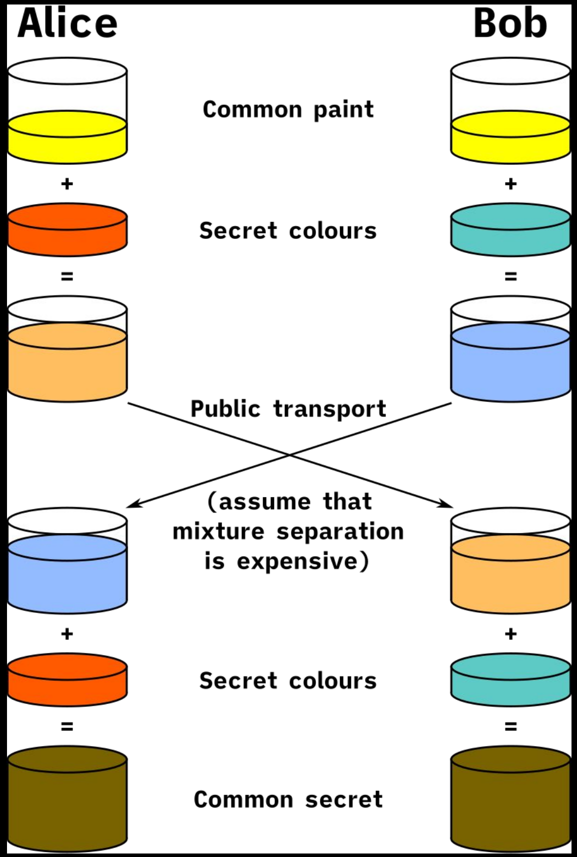
- Because p_i and p_j share no common factors whenever $i \neq j$, we have a unique solution for $x \pmod{p_1 p_2 \dots p_k}$

Why Should I Care?

- Cryptographic systems using modular arithmetic (many modern cryptographic systems) need to be careful with primes
- Smooth primes: primes p such that p - 1 has many small factors
  - Pohlig-Hellman algorithm
- CRT and Pohlig-Hellman used to attack TLS/SSL in 2015

Quick Refresher: Diffie-Hellman

- Alice and Bob arrive at a shared secret using their private secrets
- Works because of the discrete logarithm problem
- Diffie-Hellman used to share keys for symmetric encryption schemes
  - What about asymmetric encryption?



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
- RSA is one example we will go into

Totients and Euler's Function

- We call $\phi(n)$ Euler's "totient" function
- $\phi(n)$ = the number of numbers ≥ 0 that share no factors with n
- 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

Activity: Quick Maths

- Multiply 7 and 7
  - 49
- Multiply 2048 and 3
  - 6144
- Factors of 49
  - 1, 7, 49
- Factors of 6144
  - 1, 2, 3, 4, 6, 8, 12, 16, 24, 32, 48, 64, 96, 128, 192, 256, 384, 512, 768, 1024, 1536, 2048, 3072, 6144
- Factors of 32138210943
  - Uhhhhhhhhhhhhhhhhhhhhhhhhhhhhh

The Hard Problem in RSA

- Multiplication is easy
- Factoring is hard
- Let p and q be large primes. If $n = p * q$, then $\phi(n) = (p - 1) * (q - 1)$
- Given n, since p and q are large, factoring is hard!
- Therefore, finding $\phi(n)$ is hard

The RSA Cryptosystem

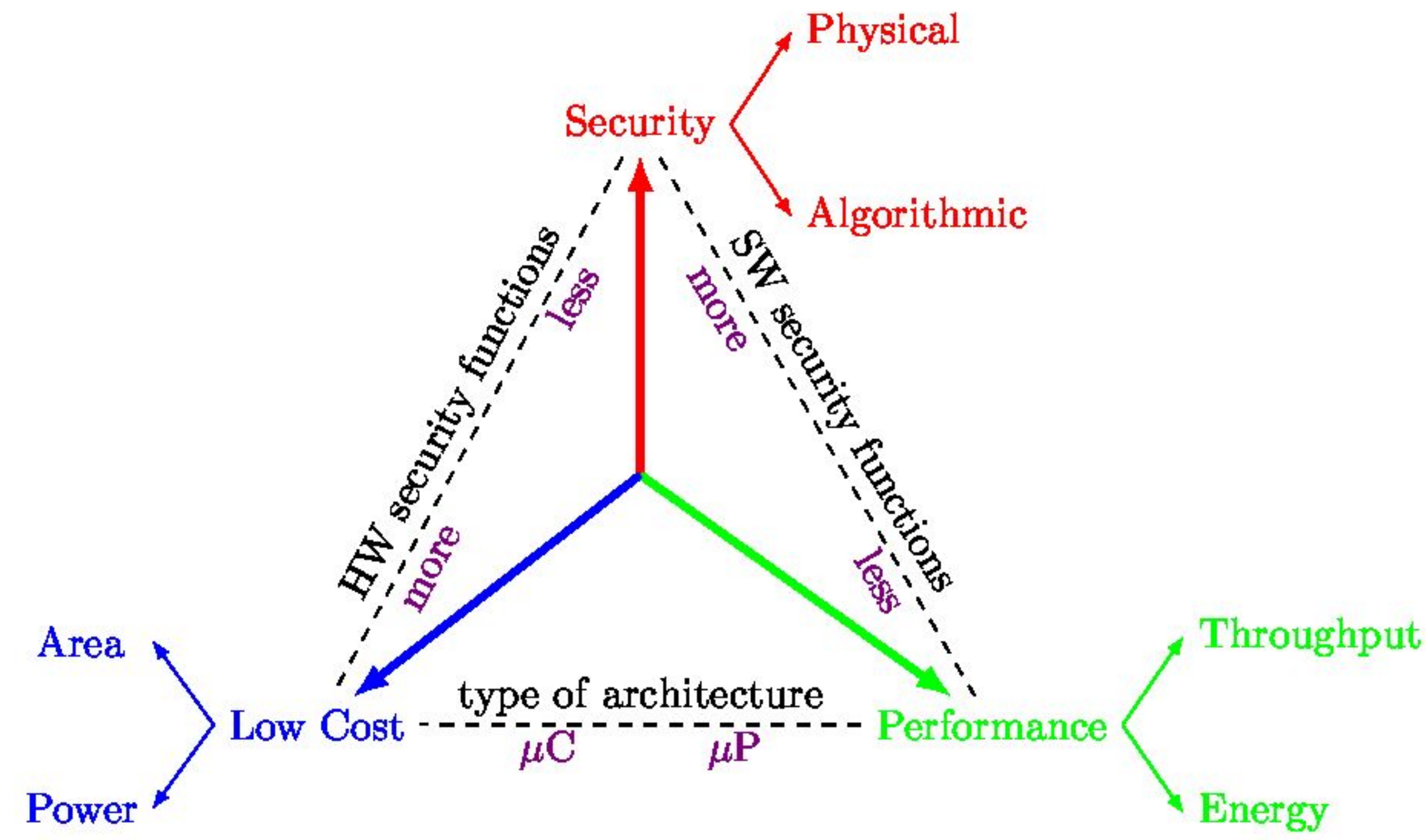
- Let e be a public exponent, usually $e = 2^{16} + 1 = 65537$
- Alice generates large (> 256 or even > 512 bits) secret primes p, q
- Alice then calculates $n = p * q$ and releases it as a public key. Then they calculate $\phi(n) = (p - 1) * (q - 1)$ as a private key.
- Knowing $\phi(n)$, compute d 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 $c^d \equiv m \pmod{n}$

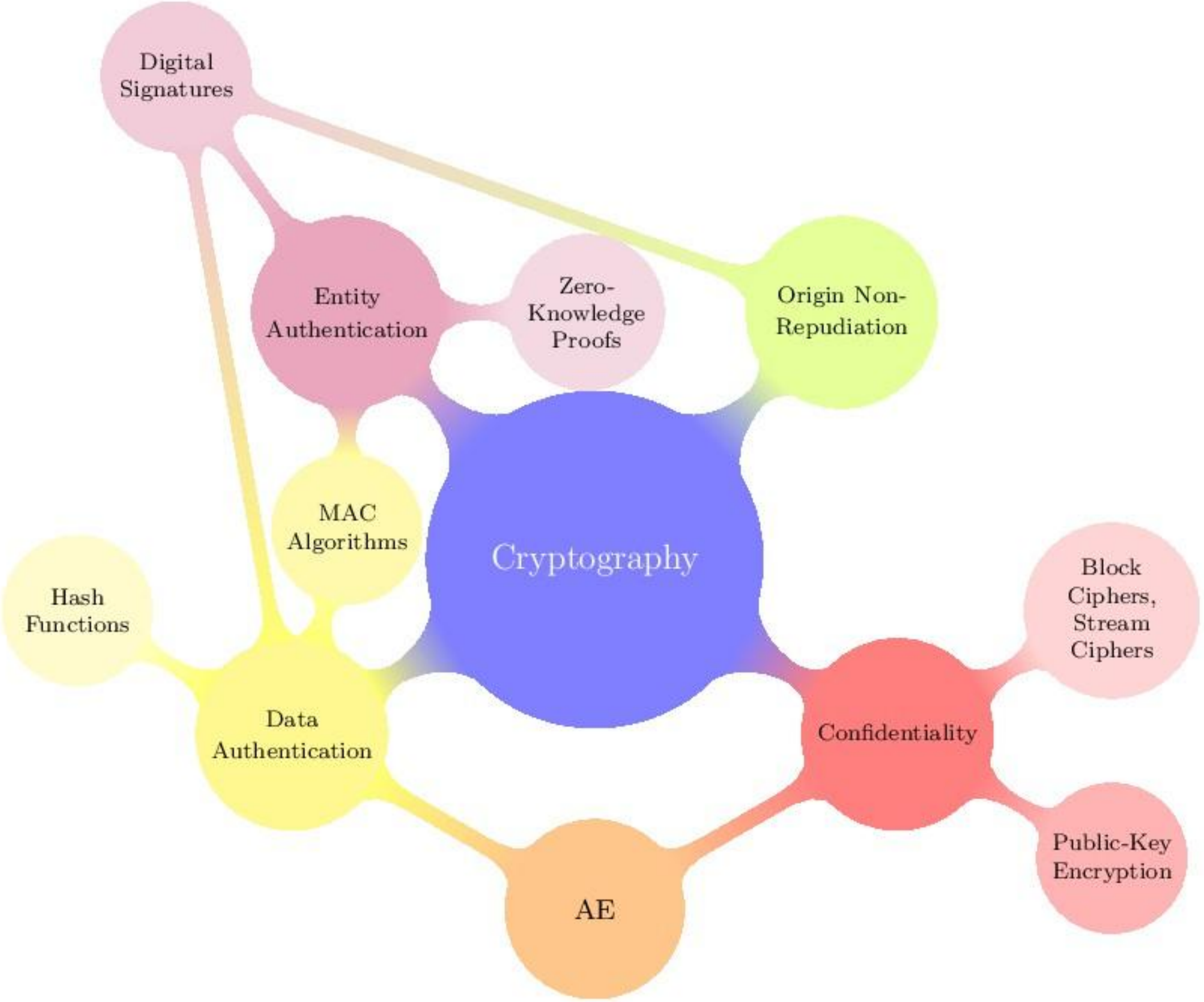
Correctness

- Remember, modular arithmetic is arithmetic using remainders
- So if $a \equiv b \pmod{n}$ then we should have that $a = b + kn$ for some k.
- $ed \equiv 1 \pmod{\phi(n)}$. So $ed = 1 + k \cdot \phi(n)$ for some k
- $c^d \equiv (m^e)^d \equiv m^{ed} \equiv m^{1+k \cdot \phi(n)} \equiv m * (m^{\phi(n)})^k \equiv m * 1^k \equiv m \pmod{n}$

Attacks

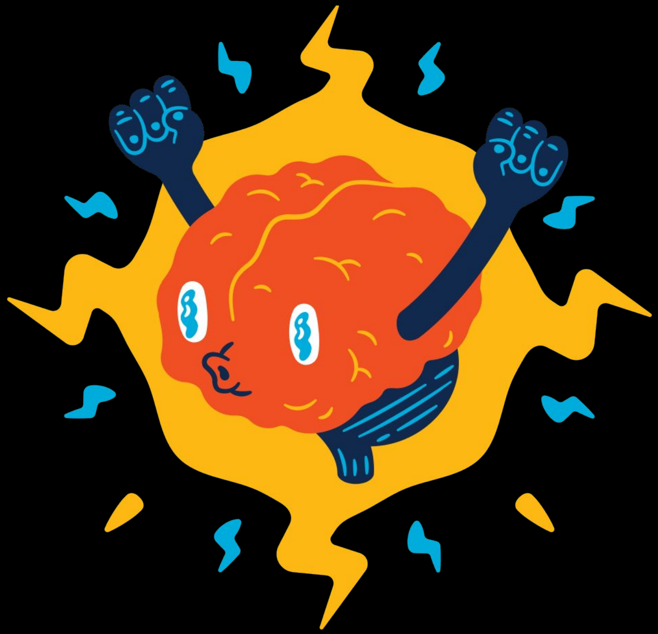
- Small primes: brute forceable
- Smooth primes: Chinese Remainder Theorem
- Large public n or small ϕ(n): Weiner's Attack
- Oracles: Get your pen and paper, do the algebra!
- Ducks (Protip: Don't use pastebin.com as secret storage)
- etc... (Google is your best friend)


Physical
Security
Algorithmic
HW security functions
less
more
SW security functions
more
less
Area
Low Cost
Power
type of architecture
μC
μP
Performance
Throughput
Energy


Digital Signatures
Entity Authentication
Zero-Knowledge Proofs
Origin Non-Repudiation
MAC Algorithms
Cryptography
Hash Functions
Block Ciphers, Stream Ciphers
Data Authentication
Confidentiality
Public-Key Encryption
AE

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

2024-10-31 • Next Thursday

- Halloween 👻

2024-11-03 • Next Sunday

- pwn II (format string attacks, control flow hijacking) with Sam ctf.sigpwny.com

sigpwny{b1g_numb3r5_g0_brr}

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

SIGPwny