CRYPTOGRAPHY

Intro

Created by Alkalem, modified by Hanna3-14

06.06.2024

OVERVIEW

- security goals
- cryptographic ciphers/protocols
  - classical cryptography
  - randomness
  - symmetric cryptography
  - asymmetric cryptography
- typical vulnerabilities
- tools
- further topics
- tasks

SECURITY GOALS

- **confidentiality:** protecting personal data
- **integrity:** data has not been modified
- **authenticity:** exactly what is claimed

CLASSICAL CRYPTOGRAPHY

CAESAR CIPHER

- each letter is shifted by a fixed value K
- $encrypt_K(P) = (P + K)\ mod\ 26$
- $decrypt_K(C) = (C - K)\ mod\ 26$

Example for K = 23:

Plaintext:	T	H	E	Q	U	I	C	K	B	R	O	W	N	F	O	X	...
Ciphertext:	Q	E	B	N	R	F	Z	H	Y	O	L	T	K	C	L	U	...

CLASSICAL CRYPTOGRAPHY

VIGENÈRE CIPHER

- choose a key of multiple letters, shift each letter according to the key letter
- attacks: determine key length, frequency analysis, (Kasiski examination)

Example:

Plaintext:	A	T	T	A	C	K	A	T	D	A	W	N
Key:	L	E	M	O	N	L	E	M	O	N	L	E
Ciphertext:	L	X	F	O	P	V	E	F	R	N	H	R

CLASSICAL CRYPTOGRAPHY

ONE TIME PAD (OTP)

- use key with same length as plaintext
- **information-theoretically secure,** if key is chosen equally distributed at random and used only once
- key exchange needs to be done separately via a secure channel
- mostly not realizable

RANDOMNESS

- in most programming languages default pseudo-random number generators (PRNGs) are not cryptographically secure

  ⇛ state can be recovered

- cryptographically secure RNGs:
  - /dev/urandom
  - hardware RNGs
  - RNGs of the cryptographic libraries (e.g., secrets or Crypto.Random in python)

SYMMETRIC CRYPTOGRAPHY

STREAM CIPHERS

- pseudo-random key stream generated from key with an PRNG
- key stream is XORed with plaintext stream
- examples: RC4, SEAL, Salsa, CryptMT
- attacks:
  - known plaintext: calculate parts of the key stream when parts of plaintext are known
  - key reuse: two messages are encrypted with same key stream, difference (XOR) between plaintexts is observable

SYMMETRIC CRYPTOGRAPHY

BLOCK CIPHERS

- encrypt blocks of fixed length
- padding: extend messages to full block length
- examples: DES, IDEA, RC5, AES, Blowfish, ...
- modes of operation: (e.g. ECB, CBC, CTR, GCM)
- attacks:
  - against cipher: differential or linear cryptanalysis
  - different attacks against different modes of operation

RSA - KEY GENERATION

- choose large prime numbers p and q
- calculate the modulus $N = p * q$
- calculate $\phi(N) = (p-1) * (q-1)$
- choose e with $gcd(e,\ \phi(N)) = 1 \land 1 < e < \phi(N)$
- calculate d as inverse of e under modulus $\phi(N)$

$$e * d \equiv 1 \mod \phi(N)$$

- public key: $N,\ e$
- private key: d

RSA - ENCRYPTION AND DECRYPTION

- encryption: $c = m^e \mod N$
- decryption: $c^d = (m^e)^d = m^{ed \mod \phi(N)} = m^1 \mod N$
- RSA without special padding is homomorphic ($Enc(m_1, pk) * Enc(m_2, pk) = Enc(m_1 * m_2, pk)$) and deterministic
- use RSA-OAEP if that is problematic

RSA - ATTACKS

- factoring N has complexity of about $exp(log(N)^{\frac{1}{3}}(loglogN)^{\frac{2}{3}})$, infeasible for reasonable choice of N
- in some cases, attacks in polynomial time possible:
  - small private exponent d ($d < \frac{1}{3}N^{\frac{1}{4}}$): Wiener's attack
  - for small public exponent or partially known prime factor: Coppersmith's attack
  - $m < N^{\frac{1}{e}}$: calculate message as root of ciphertext
  - message sent to many recipients using same public exponent: Hastad's Broadcast Attack

ELLIPTIC CURVES

- elliptic curve equation:
- group: generator point G, point addition and multiplication with natural number
- cyclic EC group over $\mathbb{Z}_p, p > 3$
- point (x, y) on curve iff $y^2 \equiv x^3 + ax + b \mod p$, plus (imaginary) point at infinity O, $a, b \in \mathbb{Z}_p$ with $4a^3 + 27b^2 \not\equiv 0 \mod p$
- harder to attack, can use smaller keys for same security level

TYPICAL VULNERABILITIES

- implementation mistake: incorrect/vulnerable custom implementation, incorporated incorrectly into application
- conceptual mistake: incorrect use or not sufficient for use case
- theoretic mistake: violated condition for security, advanced maths or theoretic computer science necessary, "read the paper"
- well-known and documented attacks (e.g. length extension attack)
- oracles

TOOLS

- CyberChef
- https://factordb.com/
- sagemath (free open-source mathematics software system)
- Z3 (theorem prover)

FURTHER TOPICS

- post-quantum cryptography
- pairing-based cryptography
- zero knowledge

START PLAYING CRYPTO CTF

- https://intro.kitctf.de/
- other platforms:
  - https://cryptohack.org/ (Easy to hard, with good explanations)
  - https://cryptopals.com/ (Implement cryptosystems and attacks)
  - https://overthewire.org/wargames/krypton (Classical crypto)
  - https://imaginaryctf.org/ (not only crypto but also other CTF challenges)