COMP47860: Ethical Computer Hacking Broken Cryptography

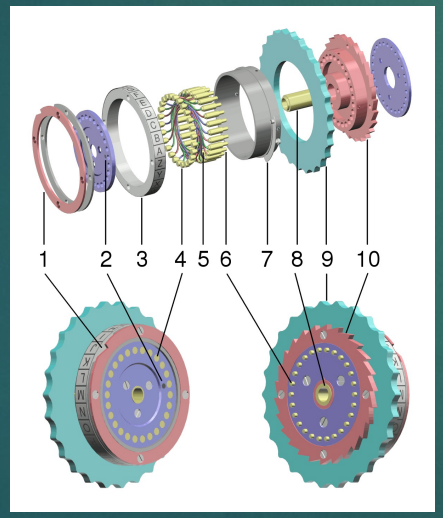
ASSOC. PROF. MARK SCANLON

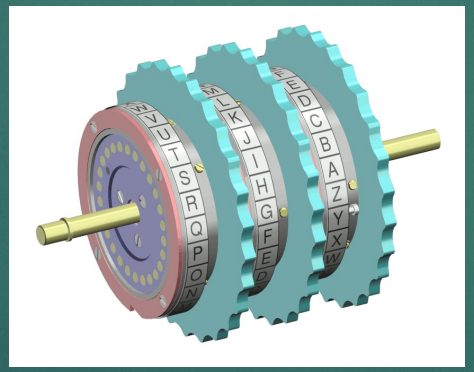
Agenda

- Overview of cryptography and encryption techniques
- How public and private keys are generated
- Overview of MD5, SHA, RC4, Blowfish algorithms
- Overview of digital certificates
- Broken cryptography
- Sensitive data exposure
- Use of broken or risky cryptography algorithms
- Mitigations

Enigma Machine

- Enigma machines were a series of electro-mechanical rotor cipher machines, developed and used in World War 2, for hiding military secret information.





- Online Emulator:

https://www.101computing.net/enigma-machine-emulator/

Cryptography - Hashing

- One way function, meant to protect or keep data such as passwords or file secure.
- Hashing is the transformation of a string of characters into a usually shorter fixed-length value or key that represents the original string.
- It is used to index and retrieve items in a database.
- For example, index a list of books in a library

Cryptography - Encryption

- Encryption is the process of transforming information using an algorithm to make it unreadable, only to anyone who expect to know the secret key.
  - e.g., Enigma machine, WhatsApp

Cryptography - Encoding

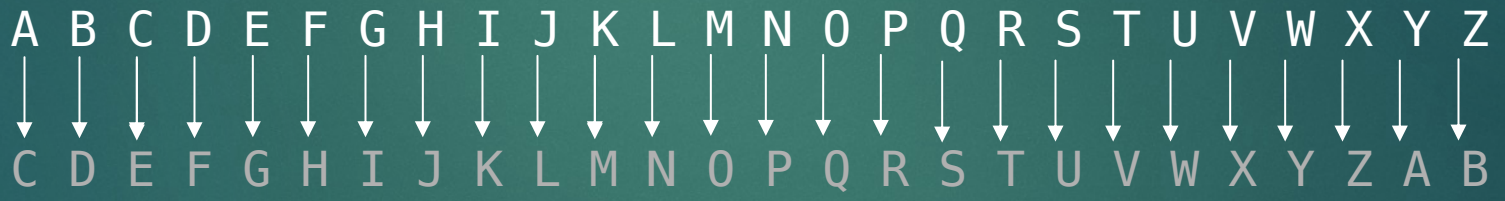
- Encoding transforms data into another format using a scheme that is publicly available, can be easily be reversed.
- e.g., Base64: contains A-Z, a-z, 0-9, +,/, =

SGF2ZSB5b3Ugbm90aGluZyBiZXR0ZXIgdG8gYmUgYXQ/

Note: Encoding is not encryption, it is easily reversed.

Caesar cipher

- Caesar cipher (a type of shift cipher), is one of the simplest forms of encryption.
- Crack the cipher algorithm, crack the cipher; no matter the key.



Algorithm: Shift by 2

`Yja uq ugtkqwu?` ➡️ `Why so serious?`

Symmetric Key Encryption

- Both sender and receiver share the same key.
- Fast and good for bulk encryption.
  - e.g., Share your house key with someone

- No secure way to share the key between multiple systems.
  - e.g., DES

Asymmetric Key Encryption

- Asymmetric key encryption was created to address the weakness of symmetric key management and distribution.
- Uses two related, but different keys, public key and private key.
- A public key is made available to anyone who might want to send you an encryption message.
- A private key is kept secret. e.g., mailbox

How public and private keys are generated

- Digital certificates: used for transferring public key, a certificate is file of information that identifies a user or a server, contains the organization name, the user's email address, country and public key.
- Certificate authorities: issues digital certificates, Symantec, GeoTrust…
  - e.g., Certificate generation and destruction
- Key management: key storage (where you store the key) and key length (what key length is adequate for the data protection)

Overview of of MD5, SHA, RC4, and Blowfish algorithms

- Algorithms vary in key length from 40 bits to 448 bits. The longer key length, the stronger the encryption algorithm.
- Brute force a key of 40 bits -> 1.4 minutes to 0.2 seconds, 64-bit key -> 50 years to 37 days
- 256 bits is considered uncrackable for now, but probably not in the future.
- MD5 (hashing)– a hashing algorithm that uses a random - length input to generate a 128-bit digest, the length of MD5 is fixed

MD5("please do not change me.") -> `6A84D15544B770AC43BAC9491A329556`
MD5("Please do not change me.") -> `57B9EC80859E5D9A56D2EE36253804B4`

- SHA (Secure Hash Algorithm) – used for digital signature algorithm.
- RC4 – encryption: a symmetric key algorithm
- Blowfish– encryption: a 64-bit block cipher

What is Broken Cryptography?

- OWASP Top 10 2021 (no. 2) (previously Sensitive Data Exposure)
  - Sensitive data that requires extra protection but has not been protected
  - Data stored in clear text long term, including backups of data.
  - Data transmitted in clear text, internally or externally. Internet traffic is especially dangerous.
  - Old / weak cryptographic algorithms being used.
  - Weak crypto keys generated, or proper key management or rotation is missing.
  - Browser security directives or headers missing when sensitive data is provided by / sent to the browser.

Sensitive Data Exposure

- Threat Agents
  - Who could gain access to sensitive data and any backups. Including data at rest, in transit, and even in a users browsers.
- Attack Vectors
  - Attackers typically don't break crypto directly. They break something else, such as steal keys, do man-in-the-middle attacks, or steal clear text data off the server, while in transit, or from the user 's browser.
- Security Weakness
  - Simply not encrypting sensitive data. Then weak key generation and management, and weak algorithm usage, particularly weak password hashing techniques. Browser weaknesses are very common and easy to detect, but hard to exploit on a large scale.

Sensitive Data Exposure

- Technical Impacts
  - Failure frequently compromises all data that should have been protected. Typically, this information includes sensitive data such as health records, credentials, personal data, credit cards, etc.
- Business Impacts
  - Consider the business value of the lost data and impact to your reputation. What is your legal liability if this data is exposed? Also consider the damage to your reputation.
- Examples:
  - Data stored in plain text, e.g. 100 million passwords in text from vk.com
  - Lack of HTTPS on authenticated pages
  - Hashed passwords with lack of salt, making the password easily cracked.
  - Tokens disclosed in public source code, e.g. slack bot token leakage

Obfuscating or Hiding Data

- Encoding is often mistakenly used as a mean of encrypting data. For example the use of base64.
- Storing the cryptographic key on the client side such as a cryptographic function in JavaScript.
- The use of XOR to try obfuscate the plain text.
- Storing obfuscated coupon code on the client side. JavaScript deobfuscators can be utilised to view how coupons are generated.

Risk factors

- Algorithm Problems
  - Insecure algorithm such as DES, MD5, SHA1, AES, Blowfish
  - Choosing the wrong algorithm
  - Inappropriate use of an algorithm - Use hash function for encryption. Use encryption algorithm for hashing.
  - Implementation errors - Use non-standard cryptographic implementations/libraries.
- Key management problems
  - Weak keys
  - Key disclosure
  - Key updates
- Random number generator problems

Example Attack Scenarios

- Scenario #1: An application encrypts credit card numbers in a database using automatic database encryption. However, this means it also decrypts this data automatically when retrieved, allowing an SQL injection flaw to retrieve credit card numbers in clear text. The system should have encrypted the credit card numbers using a public key, and only allowed back-end applications to decrypt them with the private key.

- Scenario #2: A site simply doesn't use SSL for all authenticated pages. Attacker simply monitors network traffic (like an open wireless network), and steals the user's session cookie. Attacker then replays this cookie and hijacks the user's session, accessing the user's private data.

- Scenario #3: The password database uses unsalted hashes to store everyone's passwords. A file upload flaw allows an attacker to retrieve the password file. All of the unsalted hashes can be exposed with a rainbow table of precalculated hashes.

Password Cracking

- Step one get access to passwords from data at rest (e.g., in a database) or data in transit (e.g., data transmitted over a network)
- The purpose is to gain unauthorised access to a computer system, in cases to recover a forgotten password or for testing purposes to test the password strength to gauge if an attacker could break your password
- Password cracking can be a repetitive process where a computer script attempts different combinations of passwords until one matches.
- Techniques include – Brute Force, GPU and CUDA

Brute Force Technique

- Also known as a brute force attack it consists of a guessing passwords w here a program will draw from a large number of password combinations. Some notable password files are rockyou.txt or simply look up worst passwords of 2021.
- Usually the technique is employed by hackers when there is no chance of taking advantage of encryption weaknesses or by pentesters to determine the security of the system. If the password is not in a list it won't be cracked and the longer more complex the password the longer it'll take to crack.
- Time can also be effected by the network speed and the speed of the machine doing the brute forcing.

GPU Technique

- Graphic Processing Unit (GPU) cracking using the power of GPUs to run a password through a one way hashing function and then compare that hash against the target hash.
- GPUs are used as they can perform mathematical functions in parallel utilising the hundreds of cores.
- GPUs are much faster than CPUs at this type of calculation - hence why they are used.

CUDA Technique

- Utilises a parallel computing platform and API called Compute Unified Device Architecture (CUDA)
- Created by Nvidia for graphic processing
- Hackers adopted the technology to crack passwords using GPUs running in parallel

Rainbow Tables

- Lookup table used to recover the plaintext password from a ciphertext generated by a one- way hash.
- Tables are specific to the hash function they were created for, e.g., MD5 tables can only crack MD5 hashes.
- The tool RainbowCrack was later developed to generate and use rainbow tables for a variety of character sets and hashing algorithms, including LM hash, MD5, SHA1, etc.
- Other tools include Ophcrack and Cain and Abel

Other Tools

- Cain and Abel - Windows (Multi-functional)
- John the Ripper : Multi-platform (Password cracking)
- Aircrack : Wireless cracking tool
- Hydra: Network Cracker
- Much more…

Considerations

- Considering the threats you plan to protect this data from (e.g., insider attack, external user), make sure you encrypt all sensitive data at rest and in transit in a manner that defends against these threats.
- Don't store sensitive data unnecessarily. Discard it as soon as possible. Data you don't have can't be stolen.
- Ensure strong standard algorithms and strong keys are used, and proper key management is in place. Consider using FIPS 140 validated cryptographic modules.
- Ensure passwords are stored with an algorithm specifically designed for password protection, such as bcrypt, PBKDF2, or scrypt.
- Disable autocomplete on forms collecting sensitive data and disable caching for pages that contain sensitive data.

Mitigations

- Use strong, up-to-date algorithms to encrypt the data while to store or transmit sensitive data
- Do not use the weak or risky algorithms
- Proper key management
- Stored passwords with an algorithm specifically designed for password protection
- Do not develop custom or private algorithms

Further Reading

- Cryptography Engineering - Design Principles and Practical Applications
- OWASP Top 10 A6 Sensitive Data Exposure
- Cryptographic Storage Cheat Sheet
- Password Storage Cheat Sheet
- Transport Layer Protection Cheat Sheet
- CWE 327