Principles of Computer Security

Alvin Lin

January 2018 - May 2018

The Role of Cryptography in Security

There are always ways to get around cryptography barriers and these methods have nothing to do with breaking codes. While systems may use cryptography to make sure that data is transmitted with perfect security, who's to ensure the integrity of the person who programs the computer?

Kerckhoff and Shannon

"A cryptographic system should be designed to be secure, even if all of the details, except for the key, are publicly known." - Auguste Kerckhoff, 1883
"One ought to design systems under the assumption that the enemy will immediately gain full familiarity with them." - Claude Shannon, 1949
These are also related to the Saltzer and Schroeder principles.

Shannon's Characteristics of Good Ciphers

1. The amount of secrecy needed should determine the amount of labor appropriate for encryption and decryption.
2. The set of keys and enciphering algorithm should be free from complexity.
3. The process implementation should be as simple as possible.
4. Errors in ciphering should not propagate and cause further corruption of message information.
5. The size of enciphered text should not be larger than the original message text.

Cryptographic Primitives

- Substitution: one set of bits is exchanged for another
- Transposition: rearranging ciphertext order to break repeating patterns in plaintext
- Confusion: an algorithm providing good confusion has a complex functional relationship between the plaintext and ciphertext, so that changing one character causes unpredictable changes to ciphertext
- Diffusion: distributes information from single plaintext characters over the entire ciphertext output

Properties of a Trustworthy Cryptosystem

- Must be based on sound mathematics
- Must have been analyzed by competent experts and found to be sound
- Mus thave stood the test of time
- Producing good cryptographic algorithms is a difficult task and should be left to those who know how to create such algorithms

Symmetric Encryption

Symmetric encryption is a standard technique for ensuring confidentiality for data at rest or in transit. It has two requirements: a strong encryption algorithm, and a secure secret key that is shared by the sender and receiver. Secure symmetric encryption has one point of failure. Once the key is compromised, the integrity of the message is compromised.

- Cryptanalytic Attacks: these attacks make use of some knowledge of general characteristics of the plaintext and use the nature fo the algorithm to figure out the specific plaintext or the key used. If this is successful, all future and past messages encrypted with that key are compromised.
- Brute-Force Attacks: these attacks try all possible keys on some ciphertext until an intelligible translation into plaintext is obtained. On average, half of all possible keys must be tried for success.

DES, 3DES, and AES

- Data Encryption Standard (DES): the DES algorithm uses 64-bit plaintext blocks and a 56-bit key to produce a 64-bit ciphertext block. It is no longer considered secure due to the inadequacy of 56-bit keys.
- Triple DES (3DES): the 3DES algorithm repeats DES three times using two or three unique keys. 168-bits help prevent brute-force attacks but it is a sluggish algorithm.
- Advanced Encryption Standard (AES): AES was developed as a replacement for 3DES that significantly improved efficiency without sacrificing security strength.

Typical symmetric encryption is applied to a unit of data larger than a single 64-bit or 128-bit block.

Average Time Required for Exhaustive Key Search

Key size (bits)	Cipher	Number of Alternative Keys	Time Required at 10^9 decryptions/s	Time Required at 10^13 decryptions/s
56	DES	$2^{56} \approx 7.2 \times 10^{16}$	1.125 years	1 hour
128	AES	$2^{128} \approx 3.4 \times 10^{38}$	5.3×10^{21} years	5.3×10^{17} years
168	3DES	$2^{168} \approx 3.7 \times 10^{50}$	5.8×10^{33} years	5.8×10^{29} years
192	AES	$2^{192} \approx 6.3 \times 10^{57}$	9.8×10^{40} years	9.8×10^{36} years
256	AES	$2^{256} \approx 1.2 \times 10^{77}$	1.8×10^{60} years	1.8×10^{56} years

Block and Stream Ciphers

Block Cipher:

- processes input, one block of elements at a time
- produces an output block for each input block
- can reuse keys
- more common

Stream Cipher:

- processes input elements continuously
- produces output one element at a time
- usually always faster, with less code
- encrypts plaintext one byte at a time
- pseudorandom stream is one that is unpredictable without knowledge of the input key

Message Authentication

Message authentication protects against active attacks and verifies that a received message is authentic. A message is authentic if its contents have not been altered, it is from an authentic source, and it is timely and in the correct sequence. For message authentication, hash functions with the following properties are used:

- applicable to a block of data of any size
- produces a fixed-length output
- one-way or pre-image resistant, computationally infeasible to find x such that $H(x) = h$
- computationally infeasible to find $y \neq x$ such that $H(y) = H(x)$
- collision resistant or strong collision resistance, computationally infeasible to find any pair (x, y) such that $H(x) = H(y)$

Secure hash functions can be attacked by cryptanalysis, which exploits logical weaknesses in the algorithm, or by brute force, which solely depends on the hash code length produced.

Public-Key Encryption

Public-key encryption is based and mathematicaly functions and was publicly proposed by Diffie and Hellman in 1976. It is asymmetric and uses two separate keys, a public and private key. Ideally, it should be computationally easy to create and distribute key pairs, but it should be computationally infeasible to determine the private key from the public key or otherwise recover the original message.

Asymmetric Encryption Algorithms

- **RSA (Rivest, Shamir, Adleman), 1977** is the most widely accepted and implemented PKE approach. It involves a block cipher in which the plaintext and ciphertext are integers are between 0 and $n - 1$ for n.
- The **Diffie-Hellman key exchange algorithm** enables two users to securely reach agreement about a shared secret for use as a secret key for subsequent symmetric encryption.
- The **Digital Signature Standard (DSS)** provides a digital signature function with SHA-1 and cannot be used for encryption or key exchange.
- **Elliptic Curve Cryptography (ECC)** has security like RSA, but with much smaller keys.

Random Numbers

Random numbers are used to generate keys for public-key algorithms, keys for symmetric stream ciphers, handshaking, or session keys. In order for a number sequence to be considered random, it must have uniform distribution, independence, and unpredictability. The frequency of each number occurrence should be approximately the same and no value in the sequence should be inferred from another. Each number should be statistically independent from other numbers in the sequence and attacks should not be able to predict future elements of the sequence based on earlier sequences.

Cryptographic applications typically use algorithms for random number generation. Since they are deterministic, they typically produce sequences that are not statistically random. Thus, they are likely to be predictable. True random number generators use some nondeterministic source to produce randomness, such as radiation, gas discharge, or leaky capacitors.

You can find all my notes at `http://omgimanerd.tech/notes`. If you have any questions, comments, or concerns, please contact me at alvin@omgimanerd.tech