Types of Cryptographic Attacks

Eric Conrad

Types of Cryptographic Attacks

Introduction

Cryptographic attacks are designed to subvert the security of cryptographic algorithms, and they are used to attempt to decrypt data without prior access to a key. They are part of Cryptanalysis, which is the art of deciphering encrypted data. Cryptanalysis and Cryptography (the art of creating hidden writing, or ciphers) form the science of Cryptology.

Cryptographic Attack Methods

There are six related cryptographic attack methods, including three plaintext-based methods and three ciphertext-based methods:

Plaintext-Based Attacks	Known Plaintext	Chosen Plaintext	Adaptive Chosen Plaintext
Ciphertext-Based Attacks	Ciphertext Only	Chosen Ciphertext	Adaptive Chosen Ciphertext

These methods are used as the foundation of cryptographic attacks.

Known Plaintext and Ciphertext-Only Attacks

A known plaintext attack is an attack where a cryptanalyst has access to a plaintext and the corresponding ciphertext and seeks to discover a correlation between the two.

A ciphertext-only attack is an attack where a cryptanalyst has access to a ciphertext but does not have access to corresponding plaintext. With simple ciphers, such as the Caesar Cipher, frequency analysis can be used to break the cipher.

Chosen Plaintext and Chosen Ciphertext Attacks

A chosen plaintext attack is an attack where a cryptanalyst can encrypt a plaintext of his choosing and study the resulting ciphertext. This is most common against asymmetric cryptography, where a cryptanalyst has access to a public key.

A chosen ciphertext attack is an attack where a cryptanalyst chooses a ciphertext and attempts to find a matching plaintext. This can be done with a decryption oracle (a machine that decrypts without exposing the key). This is also often performed on attacks versus public key encryption; it begins with a ciphertext and searches for matching publicly-posted plaintext data.

Adaptive Chosen Plaintext and Adaptive Chosen Ciphertext Attacks

In both adaptive attacks, a cryptanalyst chooses further plaintexts or ciphertexts (adapts the attack) based on prior results.

Side Channel Attacks

Side channel attacks leverage additional information based on the physical implementation of a cryptographic algorithm, including the hardware used to encrypt or decrypt data.

The cryptographic attack methods previously described assume that a cryptanalyst has access to the plaintext or ciphertext (sometimes both) and possibly the cryptographic algorithm: .A side channel attack leverages additional information, such as time taken (or CPU cycles used), to perform a calculation, voltage used, and so on.

Bruce Schneier wrote: "Some researchers have claimed that this is cheating. True, but in real-world systems, attackers cheat. Their job is to recover the key, not to follow some rules of conduct. Prudent engineers of secure systems anticipate this and adapt to it."[1]

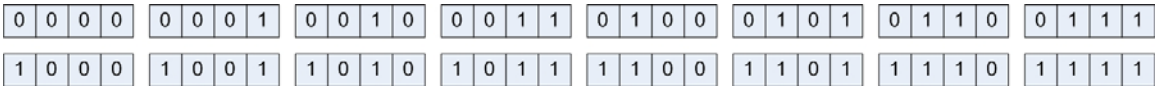
Many practical side channel attacks have been discovered. One example is the network-based attack versus OpenSSL.

OpenSSL uses two types of multiplication: one (called Karatsuba) for equal-sized words and normal multiplication for unequal-sized words. Karatsuba is faster, and the difference in speed can be detected via a network using an SSL TCP/IP connection.

The type of multiplication in use leaks information to an attacker. Researchers at Stanford University were able to launch a side-channel timing attack to recover the 1024-bit RSA key on an OpenSSL 0.9.7 server. The attack required one million queries and took two hours.[2]

Brute Force Attacks

A brute force attack systematically attempts every possible key. It is most often used in a known plaintext or ciphertext-only attack. Here is an example of a brute force attack on a 4-bit key:



Given a finite key length and sufficient time, a brute force attack is always successful. Encryption algorithms can become susceptible to brute force attacks over time as CPU speeds increase. Single DES encryption has an effective key length of 56-bits, and any key can be cracked within days using specialized hardware, such as the Electronic Frontier Foundation's Deep Crack.[3] Triple DES (168-bit key) was approved due to DES's weakness to brute force attacks, followed by the Advanced Encryption Standard (AES) in 2001. If a machine could crack one DES key per second, it would take "149

---

1 Crypto-Gram Newsletter June 15, 1998.

2 Brumley, David and Dan Boneh, *Remote Timing Attacks are Practical*. 12th Usenix Security Symposium. URL: http://www.usenix.org/events/sec03/tech/brumley/brumley_html/index.html.

3 URL: http://www.eff.org/Privacy/Crypto/Crypto_misc/DESCracker/.

thousand-billion (149 trillion) years to crack a 128-bit AES key."[4]\

One challenge associated with a ciphertext-only brute force attack is determining when it is successful. If a 15-byte plaintext of "This is secure" is encrypted with a one-time pad, a brute force attack reveals the plaintext, but it also reveals many additional possible plaintexts, such as "This is purple."

Meet-in-the-Middle Attack

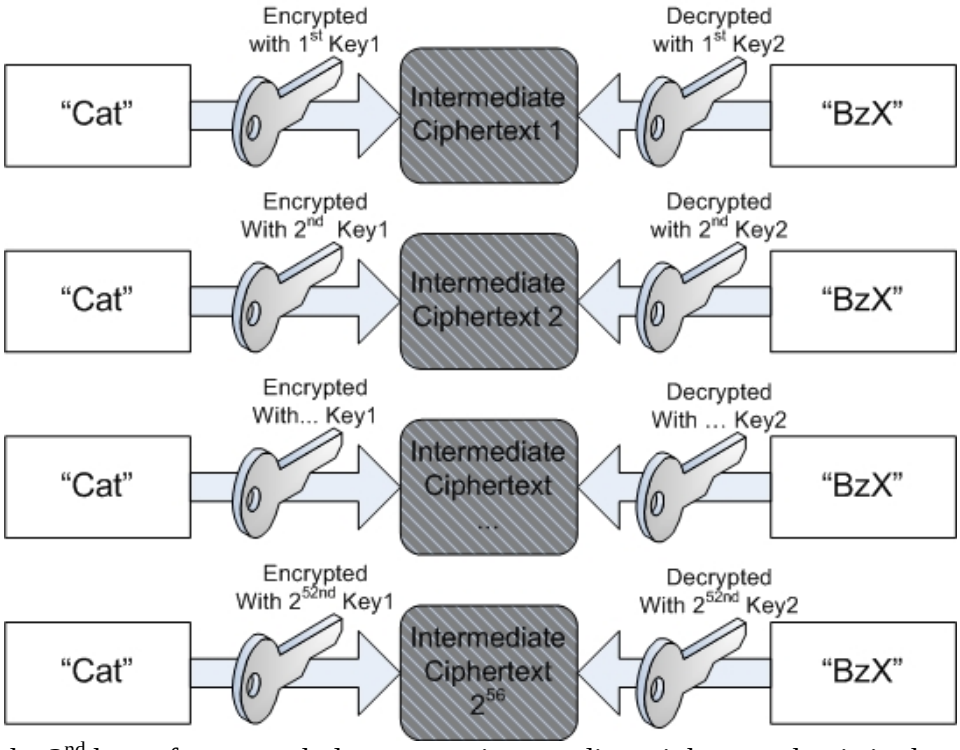
Meet-in-the-middle attacks can be used against cryptographic algorithms that use multiple keys for encryption. An example of a successful meet-in-the-middle attack is the attack versus Double DES.

To improve the strength of 56-bit DES, Double DES (two rounds of DES encryption using two different keys, for a total key length of 112 bits) was suggested.

The meet-in-the-middle attack is a known plaintext attack; the cryptanalyst has access to both the plaintext and resulting ciphertext. In this example, assume the plaintext is "Cat," and the resulting double DES ciphertext is "BzX." The cryptanalyst wants to recover the two keys (called Key1 and Key2) used for encryption.

The cryptanalyst first conducts a brute force attack on Key1 using all 2^{56} different Single-DES keys to encrypt the plaintext of "Cat" and saves each key and the resulting intermediate ciphertext in a table. The analyst then brute forces Key2, decrypting "BzX" up to 2^{56} times.

[4] *AES Fact Sheet.* URL: http://csrc.nist.gov/CryptoToolkit/aes/aesfact.html.



When the 2nd brute force attack decrypts an intermediate ciphertext that is in the table, the attack is complete and both keys are known to the cryptanalyst. The attack takes 2^{56} plus at most 2^{56} attempts, or a maximum of 2^{57} total attempts. This is far easier than 2^{112} attempts.

As a result of the meet-in-the-middle attack, Double DES has not been widely used.

Linear Cryptanalysis and Differential Cryptanalysis

Differential cryptanalysis and linear cryptanalysis are related attacks used primarily against iterative symmetric key block ciphers. An iterative cipher (also called a product cipher) conducts multiple rounds of encryption using a subkey for each round. Examples include the Feistel Network used in DES and the State rounds used in AES. In both attacks, a cryptanalyst studies changes to the intermediate ciphertext between rounds of encryption. The attacks can be combined, which is called differential linear cryptanalysis.

A goal of strong encryption is to produce ciphertexts that appear random where a small change in a plaintext results in a random change in the resulting ciphertext. This quality is called diffusion,[5] and any changed ciphertext bit should have a 50% chance of being a 1 or a 0. Both attacks seek to discover non-randomness (cases where the 50% rule is broken) in an effort to discover potential subkeys.

Linear Cryptanalysis

Linear cryptanalysis is a known plaintext attack that requires access to large amounts of

[5] Term was coined by Claude Shannon in 1949 in *Communication Theory of Secrecy Systems*. URL: http://netlab.cs.ucla.edu/wiki/files/shannon1949.pdf.

plaintext and ciphertext pairs encrypted with an unknown key. It focuses on statistical analysis against one round of decryption on large amounts of ciphertext.

The cryptanalyst decrypts each ciphertext using all possible subkeys for one round of encryption and studies the resulting intermediate ciphertext to seek the least random result. A subkey that produces the least random intermediate cipher[6] for all ciphertexts becomes a candidate key (the most likely subkey).

Differential Cryptanalysis

Differential cryptanalysis is a chosen plaintext attack that seeks to discover a relationship between ciphertexts produced by two related plaintexts. It focuses on statistical analysis of two inputs and two outputs of a cryptographic algorithm.

A plaintext pair is created by applying a Boolean exclusive or (XOR) operation to a plaintext. For example, XOR the repeating binary string 10000000 to the plaintext. This creates a small difference (hence the term differential cryptanalysis) between the two. The cryptanalyst then encrypts the plaintext and its XORed pair using all possible subkeys, and it seeks signs of non-randomness in each intermediate ciphertext pair. The subkey that creates the least random pattern becomes the candidate key.

Birthday Attack

The birthday attack is an attack that can discover collisions in hashing algorithms. It is based on the Birthday Paradox, which states that if there are 23 people in a room, the odds are slightly greater than 50% that two will share the same birthday.[7]

The odds might appear counterintuitive. The key to understanding the attack is remembering that it is the odds of any two people (out of the 23) sharing a birthday, and it is not the odds of sharing a birthday with a specific person.[8]

Alice is in a room with 23 people and has 22 chances to share a birthday with anyone else (there are 22 pairs of people). If she fails to match, she leaves, and Bob has 21 chances to share a birthday with anyone else. If he fails to match, Carol has 20 chances, and so on. Twenty-two pairs, plus 21 pairs, plus 20… plus one pair equals 253 pairs. Each pair has a 1/365 chance of having a matching birthday, and the odds of a match cross 50% at 253 pairs.[9]

The birthday attack is most often used to attempt discover collisions in hash functions, such as MD5 or SHA1.

Summary

Understanding cryptographic attacks is important to the science of cryptography, and it

---

[6] The intermediate cipher that produces the largest violation of the 50% rule.

[7] Assuming that birthdays are distributed evenly over 365 days, with no leap years.

[8] For one person, 253 people (or pairs) are required for the odds to exceed 50%.

[9] The odds of failing to find a match with 253 pairs are (364/365)^253, or .4995. Odds of success are thus .5005.

serves to improve cryptographic algorithms. One example is the design of the Data Encryption Standard (DES) algorithm first published in 1977. IBM was aware of the technique of differential cryptanalysis in 1974, though the technique was not publicly documented until the late 1980s[10]. IBM used this knowledge to improve the DES algorithm, adding substitution boxes to thwart differential cryptanalysis.

[10] Coppersmith, D. *The Data Encryption Standard (DES) and its strength against attacks*, URL: http://www.research.ibm.com/journal/rd/383/coppersmith.pdf.