CS 161 Fall 2005 — Computer Security — Joseph/Tygar/Vazirani/Wagner — Notes 8

1 Brief History of Cryptography

The word "cryptography" comes from the latin root crypt meaning secret, and graphia, meaning writing. So cryptography is literally the study of how to send secret messages. In the next few lectures we shall study encryption schemes as well as some other fundamental goals of cryptography: authentication and digital signatures.

Schemes for sending secret messages go back to antiquity - the Romans used the "Caesar cypher" which consists of permuting the alphabet by simply shifting each letter forward by a fixed amount. For example, Caesar used a shift by 3 so the message "cryptography" would be encoded as "fubswrjudskb". With the developement of the telegraph (wireless communication) at the end of the nineteenth century, the need for encryption in military and diplomatic communications became particularly important. The codes that were used during this "pen and ink" period were relatively simple since messages had to be decoded by hand. The codes were not very secure especially with the use of mathematical and statistical techniques.

The second phase of cryptography, the "mechanical era" was the result of an intense German project to create a mechanical device for encrypting messages in an unbreakable code. The resulting Enigma machine was a remarkable engineering feat. Even more remarkable was the massive British effort to break the code. The breaking of the Enigma code was an important event that determined the course of World War II, and according to most experts shortened the war by about a year. There were two important factors in the breaking of the Enigma code: first, the British managed to obtain a replica of a working Enigma machine from Poland, which had cracked a simpler version of the code. The second factor was scale and sophistication of the code breaking effort, first by the Poles, who employed a large contingent of mathematicians to crack the structure, and then by the British, whose massive effort included Alan Turing, one of the founding fathers of computer science.

Modern cryptography is distinguished by its reliance on mathematics and electronic computers. It has its early roots in the work of Claude Shannon following World War II. The analysis of the one-time pad later in this lecture is due to Shannon. The early seventies saw the the introduction of the cryptosystem DES by NIST (the National Institute for Standards in Technology). DES answered the growing need for digital encryption standards in banking and other business. The decade starting in the late seventies saw an explosion of work on a computational theory of cryptography.

The most basic problem in cryptography is one of ensuring the security of communications across an insecure medium. The two main members of the cast of characters in cryptography are Alice and Bob who wish to communicate securely as though they were in the same room or were provided with a dedicated, untappable line. In actual fact they have available a telephone line or an internet connection which can be tapped by an eavesdropping adversary, Eve. The goal is to design a scheme for scrambling the messages between Alice and Bob in such a way that Eve has no clue about the content of their exchange. In other words we wish to simulate the ideal commuication channel with the available insecure channel.

In a symmetric trust model or the shared-key model, Alice and Bob share a random key K that is unknown to Eve. Alice encrypts her message A based on the key K, and Bob decrypts the received ciphertext (to recover

the original message, the plaintext) using the same key K.

Let us now examine the threat model, which in this setting involves answering the question, how powerful is Eve? We will assume that Eve knows the encryption and decryption algorithms. The only information she is missing is the secret key K. There are several possibilities about how much access Eve has to the insecure channel:

1. Eve has managed to intercept a single encrypted message and wishes to recover the plaintext (the original message).
2. Eve has intercepted an encrypted message and also has some partial information about the plaintext.
3. Eve can trick Alice to encrypt messages of her choice (this might happen if Eve has access to the encryption program).
4. Eve can trick Bob into decrypting some ciphertexts.

2 One time Pad:

The one time pad is a simple and idealized encryption scheme that will help illustrate some important concepts. Alice and Bob share an n-bit secret key $K = k_1 \dots k_n$ where the bits $k_1, \dots k_n$ are picked at random (they are the outcomes of independent unbiased coinflips).

Suppose Alice wishes to send the n-bit message $M = m_1, \dots m_n$.

The desired properties of the encryption scheme are:

1. It should scramble up the message. i.e. map it to a ciphertext $C = c_1 \dots c_n$.
2. Given knowledge of the secret key K, it should be easy to recover M from C.
3. Eve, who does not know K, should get no information about M.

The encryption scheme is very simple: $c_j = m_j \oplus k_j$, where $\oplus$ is the xor or exclusive-or of the two bits (0 if the two bits are the same and 1 if they are different).

Decryption is equally simple: $m_j = c_j \oplus k_j$.

To sum up, the one-time pad (or any symmetric encryption scheme) is described by specifying three procedures:

Key generation: Alice and Bob pick a shared random key K. Encryption algorithm: $C = M \oplus K$. Decryption algorithm: $M = C \oplus K$.

Let us now analyze how much information Eve gets about the plaintext M by intercepting the ciphertext C. What is the correct measure of this information gained by Eve. It might be the case that Eve had some partial information about A to begin with. Perhaps she knew that the last bit of A is a 0 or that 90% of the bits of A are $1's$. We wish to show that after intercepting B, Eve gets no additional information about A.

We will show this by proving that no matter what the value of M is, the ciphertext C is a uniformly random n-bit string.

Another way of saying this is the following:

Consider the following experiment - suppose Alice has sent one of two messages M or M'. Eve tries to distinguish which by looking at the ciphertext. We will show that Eve's success probability is $1/2$, which is no different than it would be if she had not intercepted the ciphertext at all.

The proof is very simple: for a fixed choice of plaintext M, every possible value of the ciphertext C can be achieved by an appropriate and unique choice of the shared key K: namely $K = M \oplus C$. Since each such key value K is equally likely, it follows that C is also equally likely to be any n bit string. Thus Eve sees a uniformly random n bit string no matter what the plaintext message was, and thus gets no information about the plaintext.

The one time pad has a major drawback - as its name suggests, the shared key cannot be reused to transmit a new message M'. If the key K is reused, then Eve can take the xor of the two ciphertexts $C = M \oplus K$ and $C' = M' \oplus K$ to obtain $M \oplus M'$. This gives partial information about the two messages. In particular, if Eve happens to learn M, then she can deduce the other message M'. Actually in this case she can reconstruct the key K.

3 Symmetric Encryption Systems:

The Data Encryption Standard (DES) is an example of a block cipher. It was designed by IBM in 1974 in response to a request from NIST for an encryption algorithm that could be standardized.

In a block cipher, Alice and Bob share a k bit random key K, and use this to encrypt an n bit message into an n bit ciphertext. In mathematical notation this can be said as follows. There is an encryption function $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$. Once we fix the key K, we get a function mapping n bits to n bits: $E_K : \{0,1\}^n \to \{0,1\}^n$ defined by $E_K(M) = E(K,M)$. E_K is required to be a permutation on the n bit strings. The inverse mapping of this permutation D_K is the decryption algorithm. $D_K(E_K(M)) = M$.

DES uses a key length of $k = 56$ and a block length of $n = 64$. It was designed to have extremely fast VLSI implementations. In terms of security, DES has proved to be an impressively strong algorithm. After all these years, the best practical attack known is exhaustive key search (a symmetry in the key structure can be used to halve the search space) thus requiring 2^{55} computations. Thus DES behaves very differently than the one-time pad. Even given a very large number of plaintext, ciphertext pairs, there appears to be no effective way to decrypt any new ciphertexts. We will formalize these security properties of DES by saying that the function E_K for a randomly chosen key K "behaves like" a random permutation on the n bit strings.

Formally, we shall measure the security of the block cipher by performing the following experiment: the adversary, Eve, is given a box which contains either (I) the encryption function E_K with a random key K (II) a uniformly random permutation on n bits. Eve is allowed T steps in which to play with the box and guess whether it type I or type II. The advantage of the adversary Eve, is $\mathrm{Adv}Eve = |p - q|$, where p is the probability that the adversary guesses box I when the box she is given is actually of type I, and q is the probability that the adversary guesses box I when the box she is given is actually of type II. Informally the advantage of Eve is the chance that she can distinguish between the block cipher and a truly random permutation. If Eve's advantage is at most ε then we say that the block cipher is (T, ε)-secure. For DES, the above discussion says that if we want $\varepsilon = 1$ we need $T \geq 2^{55}$. In general there is a tradeoff between T and ε, and so we could say that $T/\varepsilon \geq 2^{55}$. In some sense $log\ T/\varepsilon$ is the effective key length of the block code.

In 1998 NIST announced a competition for a new block cipher. One motivation is that the block length of $n = 64$ is just too short to guarantee security. Another motivation is speed - DES was really designed for hardware implementation, and is quite slow in software. In the summer of 2001 NIST announced their choice - the Advanced Encryption Standard (AES) which was designed by Joan Daemen and Vincent Rijmen, both from Belgium. AES has a block length of $n = 128$ bits and a key length k that may be either

128, 192 or 256 bits.