University of Illinois, Urbana Champaign
CS 598DK Special Topics in Cryptography

Instructor: Dakshita Khurana
Scribe: Joshua Reynolds, Amit Agarwal
Date: August 28, 2019

LECTURE
1

Introduction, Overview, One Time Pad

1.1 Introduction

This introductory lecture introduces some fundamental goals of cryptography and defines the basic requirement for a useful private key (symmetric) encryption scheme. A basic problem in Cryptography is that of encryption. The encryption problem is that two parties wish to communicate a message without an eavesdropper being able to learn their secret, or even any information about that secret. This problem can be solved with private key (symmetric) encryption under the assumption that both parties already have a shared secret key. This problem can also be solved with public key (asymmetric) key encryption under various other assumptions.

These primitives are among the building blocks of more complicated cryptography systems and a useful medium to explore the assumptions made in cryptographic systems. Encryption, in particular, is important because it allows us to communicate secrets safely through an untrusted medium like the Internet. Focusing first on private key encryption, we define the property of correctness. The second necessary property of encryption is security and is covered in part 2 of these lecture notes.

1.2 Notation

This set of scribe notes will use the following notation and definitions:

- A set will be denoted by a capital letter
  - M is the set of all possible messages
  - K is the set of all possible keys
  - C is the set of all possible ciphertexts
- Set subtraction will use the /operator and |S| means the cardinality of set S
- An item from a set will be denoted by a lower case letter

  - m is a message in plaintext (not encrypted)
  - ct is a ciphertext (encrypted message)
  - sk is a secret key used as an input in symmetric encryption and decryption

- A statistical method of sampling from a set will be denoted with an arrow combined with a description of how the selection is done. For example, $\xleftarrow{\text{unif.}} K$ means sampling uniformly at random from key space K

- The concept of the distribution of samples from a set taken in a specific way will be called a "distribution" and denoted by an italic capital letter. Distributions are equivalent if (not iff) their underlying sets are identical and the method of sampling is identical.

- A uniformly random sample of length λ will be represented as $\xleftarrow{\$} \{0,1\}^\lambda$

- $\Pr(i)$ Denotes the probability of i being true

1.3 Encryption

Encryption is a method whereby two parties can communicate without an eavesdropper learning anything about the content of their communication. The names "Alice" and "Bob" are commonly used in cryptography as generic names for two parties wishing to securely communicate or collaborate. "Eve" is a name denoting an eavesdropper who tries to learn Alice's and Bob's secrets, but does not try to trick them. We will describe two ways in which Alice can send a message to Bob without Eve learning what message says. This will be true even when Eve gets to read every communication between Alice and Bob. For now, we will not consider Eve as capable of preventing message deliveries or altering messages—she may only listen.

In the first scheme, we assume that Alice and Bob have a shared secret encryption string of bits called the secret key (sk). When Alice wants to send a message to Bob, she first applies a transformation to the message which can only be reversed by someone who knows the secret key. Since Bob does know the secret key, he is able to take that transformed message (called a ciphertext) and apply an inverse function to recover Alice's message. Eve, on the other hand, does not know the secret key and only sees the ciphertext and will not learn anything about the message contents[1]. This scheme is called "Private Key Encryption" in this class and is illustrated in Figure 1.1.

In the second scheme, we no longer assume that Alice and Bob have a shared secret key. In exchange for this relaxed constraint, we must make several other assumptions which will be covered in the lecture on asymmetric encryption. Alice and Bob each have a pair of encryption keys—one of which is publicly announced (pubkey), and the other of which is private (privkey). Encryption performed with a public key can only be reversed by applying decryption using the secret key as input [2]. This scheme is called "Public Key Encryption" in this class and is illustrated in Figure 1.2.

In this lecture, we will take a closer look at private key encryption.

[1] Except the fact that the two parties did communicate and an upper bound on the amount of useful information transferred

[2] Note that we must assume Alice can be sure she is using Bob's public key. Remember, Eve cannot change, block, or misdirect messages

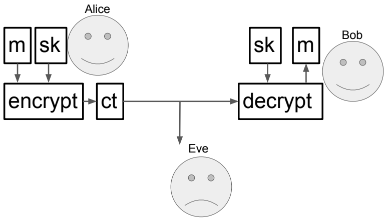

Figure 1.1: Private Key Encryption

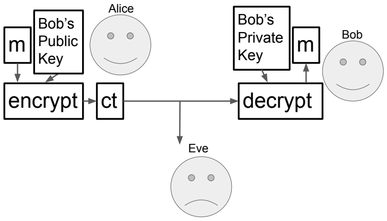

Figure 1.2: Public Key Encryption

1.4 Private Key Encryption

Private key encryption is a scheme which allows two parties with a shared secret key to send each other messages over a channel that may be intercepted without danger of an eavesdropper learning about the message.

Definition 1.1. Private Key Encryption is a scheme allowing two parties to send a message over an observed channel without revealing the message to the observer. It requires both parties to have a shared secret key sk before the scheme begins. It consists of three functions:

- KeyGen(n; r) is a function which takes a source of randomness as input (r) as well as a desired length and returns a bitstring of length n chosen uniformly from random.
- Encrypt(sk, m) is a function which uses a secret key to reversibly hide a message m in a ciphertext.
- Decrypt(sk, m) is a function which uses a secret key to compute the inverse of Encrypt on a ciphertext to recover m.

Private key encryption must satisfy the properties of security and correctness (to be explained shortly).

Private key encryption requires three functions: a key creation algorithm, encryption, and decryption as described in Definition 1.1. Alice and Bob will first generate a shared secret key (sk) using a source of randomness with KeyGen(n; r). We will discuss later what length of secret key is necessary based on various assumptions. Then when Alice wishes to send a message to Bob, she will hide her message m using Encrypt(sk, m) to generate ciphertext ct. When Bob receives ct from Alice, he applies Decrypt(sk, ct) to recover Alice's message m. Eve, who only ever saw ciphertext ct, never learns about the contents of message m.

Private key encryption must satisfy two essential properties to be secure—correctness and security. A secure protocol that communicates the wrong message is not useful to Alice and Bob. Neither is a correct protocol that communicates insecurely and reveals secrets to Eve. The following sections will provide concrete definitions of these properties.

An example of a protocol (called a One-Time-Pad) that simply accomplishes these two properties is given in Example 1.2

Example 1.2. In the One Time Pad scheme for symmetric encryption, we assume Alice and Bob generate a shared secret that is longer than the message they wish to send. The encryption transformation from a message to ciphertext is accomplished by XORing the secret key with the message. Decryption is also accomplished by XORing the ciphertext with the secret key (XOR is its own inverse). We will later show that this scheme is both secure and correct (when used one time).

1.5 Correctness

An essential property of an encryption scheme is correctness and means that information is correctly transferred. Without correctness, there is no value in having the communication to begin with. This means, essentially, that Decrypt must be able to undo Encrypt. Perfect correctness means that every time Alice and Bob follow the protocol to transmit a message their counterpart will receive the exactly correct message.

Definition 1.3. A private key encryption scheme satisfies the "perfect correctness" property iff $\forall\ m \in M$ and $\forall\ k \in K$, Decrypt(k,Encrypt(k,m)) $= m$.

Correctness may also be satisfied with a scheme wherein Alice and Bob are able to communicate their message after a reasonable number of tries with high probability. This weaker form of correctness will be useful later in the course.

Definition 1.4. (Informal) A private key encryption scheme satisfies the "mostly correct" property iff $\forall\, m \in M$ and $\forall\, k \in K$, $\exists$ a "reasonable" number of times Decrypt(k,Encrypt(k,m)) would need to be tried to be almost certain to satisfy Decrypt(k,Encrypt(k,m)) $= m$ at least once.

Example 1.2 is perfectly correct because Encrypt and Decrypt are both XOR operations, which are mathematical inverses. Therefore $\forall\, m \in M$ and $\forall\, k \in K$ Decrypt(k,Encrypt(k,m)) $= m$.

1.6 Security

Defining security in a formal way is a tricky business. One might come up with different possible definitions for security but we must be careful in analyzing our security definition from the adversary's perspective i.e., we must try thinking of ways in which an adversary might be successful in breaking our cryptographic protocol even though the protocol satisfies our proposed definition of security.

One possible way to think about the security of a private key encryption scheme is to think about the following question: *How much information does an arbitrary ciphertext leak about the plaintext from which it was generated.* The intuition behind such a notion of security is that an adversary who can eavesdrop on the communication channel is only able to observe the ciphertext, and tries to glean information about the possible plaintext from it. To make it difficult for such an adversary to break our encryption scheme, we would like to minimize the amount of information that a ciphertext leaks about its plaintext.

If the amount of information leakage is exactly zero, an adversary will have no advantage in finding the correct plaintext after seeing a ciphertext i.e., from the adversary's perspective, the distribution of plaintexts before and after seeing a ciphertext is exactly the same. This idea can be formally represented by Equation 1.1.

$$\forall m \in M, ct \in C;\ \Pr(m|ct) = \Pr(m) \tag{1.1}$$

where the probabilities are over the randomness used to sample the secret key.

The above idea is also mathematically equivalent to saying that the distributions of plaintexts and ciphertexts are independent. Therefore, we can also restate Equation 1.1 in an alternative way using Equation 1.2. We'll refer to this definition of security as **perfect security**

$$\forall m_0, m_1 \in M, ct \in C;\ \Pr(ct|m_0) = \Pr(ct|m_1) \tag{1.2}$$

where the probabilities are over the randomness used to sample the secret key.

1.7 One Time Pad is perfectly secure

Now that we have mathematically quantified our definition of security, we will prove that the One Time Pad in Example 1.2 satisfies Equation 1.2.

Proof. Because the secret key is uniformly sampled,

$$\Pr(ct|m) = \frac{Number\ of\ keys\ sk\ such\ that\ Encrypt(sk, m) = ct}{Total\ size\ of\ keyspace} \tag{1.3}$$

For a One Time Pad, there is always a unique key which maps a given plaintext to a given ciphertext and is given by Equation 1.4.

$$sk = m \oplus ct \tag{1.4}$$

Therefore, for the One Time Pad, for any m, the following relation holds:

$$\Pr(ct|m) = \frac{1}{Total\ size\ of\ keyspace} \tag{1.5}$$

Since $\Pr(ct|m)$ is a constant value, it implies that Equation 1.2 holds for One Time Pad. □

1.8 An impossibility result

We claim that any private key encryption scheme which has a keyspace smaller than the plaintext space will not be perfectly secure. The intuition is that in such a case, the space of possible decryptions for a given ciphertext is strictly smaller than the space of all possible messages. This enables an adversary to glean extra information about the possible plaintext by looking at a given ciphertext.

The formal proof for the same is as follows:

THEOREM 1.5. *Any private key encryption scheme where $|M| < |K|$ cannot satisfy Definition 1.3.*

Proof. Let $m \xleftarrow{\text{unif.}} M$ and $sk \xleftarrow{\text{unif.}} K$

We pick $ct \in C$ such that $P(ct = Encrypt(sk, m)) > 0$

This ct can decrypt to $|K|$ possible plaintexts. Let's denote this set of possible plaintexts as M'. Now we'll choose two different plaintexts, m_0 and m_1, such that $m_0 \in M'$ and $m_1 \in M/M'$.

Since $m_0 \in M'$, it implies that:

$$P(ct|m_0) > 0 \tag{1.6}$$

Since $m_1 \in M/M'$, it implies that:

$$\Pr(ct|m_1) = 0 \tag{1.7}$$

Therefore, we can say that:

$$\exists m_0, m_1 \textit{ such that } \Pr(ct|m_0) \neq P(ct|m_1) \tag{1.8}$$

Equation 1.8 clearly violates our definition of perfect security given by Equation 1.2, and therefore any private key encryption having $|K| < |M|$ is not perfectly secure. □

Acknowledgement

These scribe notes were prepared by editing a light modification of the template designed by Alexander Sherstov.