Computer Science 308-547A
Cryptography and Data Security

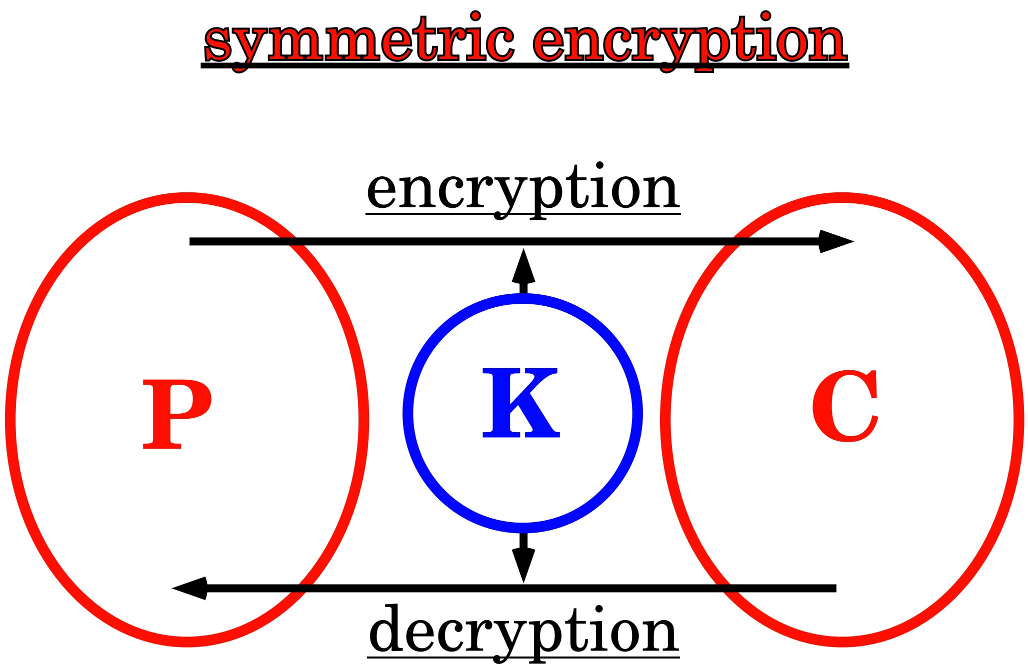
Claude Crépeau

These notes are, largely, transcriptions by Anton Stiglic of class notes from the former course *Cryptography and Data Security (308-647A)* that was given by prof. Claude Crépeau at McGill University during the autumn of 1998-1999. These notes are updated and revised by Claude Crépeau.

3 Introduction

3.1 Crypto system

Definition 3.1 *Let $\mathcal{P}$ denote a finite set of messages (also called plaintexts), $\mathcal{C}$ a finite set of ciphered texts and $\mathcal{K}$ a finite set of keys.*
For each $k \in \mathcal{K}$, we associate an encryption function $e_k : \mathcal{P} \to \mathcal{C}$ and a decryption function $d_k : \mathcal{C} \to \mathcal{P}$ such that $d_k(e_k(x)) = x$, for all $x \in \mathcal{P}$. The set of e_k's will be noted by $\mathcal{E}$ and $\mathcal{D}$ will designate the set of d_k's.
$(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ defines a cryptosystem.



3.2 Classic simple cryptosystems

Most of the following cryptosystems will be defined over $\mathbb{Z}_{26}$, so to correspond with the english alphabet of 26 symbols, but they can be generalized to $\mathbb{Z}_m$.

3.2.1 Shift cipher

Let $\mathcal{P} = \mathcal{C} = \mathcal{K} = \mathbb{Z}_{26}$.
For $0 \leq k \leq 25$ and $x, y \in \mathbb{Z}_{26}$ define

$$e_k(x) = x + k \; mod \; 26$$

and

$$d_k(y) = y - k \; mod \; 26$$

For the particular case where $k = 3$, the scheme is called the **Caesar Cipher**.

3.2.2 Substitution cipher

Let $\mathcal{P} = \mathcal{C} = \mathbb{Z}_{26}$.
$\mathcal{K} = \{\pi | \pi \; is \; a \; permutation \; over \; the \; symbols \; 0, 1, \ldots, 25 \; of \; \mathbb{Z}_{26}\}$
For $\pi \in \mathcal{K}$ and $x, y \in \mathbb{Z}_{26}$ define

$$e_\pi(x) = \pi(x)$$

and

$$d_\pi(y) = \pi^{-1}(y)$$

Note that the *shift cipher* is a special case of the *substitution cipher* in which only 26 of the possible 26! permutations are used.

3.2.3 Affine cipher

Let $\mathcal{P} = \mathcal{C} = \mathbb{Z}_{26}$,
$\mathcal{K} = \{(a, b) \in \mathbb{Z}_{26} \times \mathbb{Z}_{26} \mid gcd(a, 26) = 1\}$.
For $K = (a, b) \in \mathcal{K}$ and $x, y \in \mathbb{Z}_{26}$ define

$$e_K(x) = ax + b \; mod \; 26$$

and

$$d_K(y) = a^{-1}(y - b) \; mod \; 26$$

The functions used are called *affine* functions, thus the name of the cryptosystem. Note that the *affine cipher* is a special case of the *substitution cipher* in which only $26 * 12$ (26 values of b and 12 values of a) of the possible 26! permutations are used. Notice that if $a = 1$, we have the *shift cipher*.

In the *substitution cipher*, once a key is chosen, each alphabetic character is mapped to a unique alphabetic character. These are called *monoalphabetic* ciphers. These ciphers are vulnerable to attacks in which we can use the frequency of certain letters of the language in use. In the next cipher we present the well known *Vigenère cipher*, which is a *polyalphabetic* cipher.

3.2.4 Vigenère Cipher

Let $\mathcal{P} = \mathcal{C} = \mathcal{K} = (\mathbb{Z}_{26})^m$, for some fixed $m \in \mathbb{Z}_{26}$.
For $K = (k_1, k_2, \ldots, k_m)$ define

$$e_K(x_1, x_2, \ldots, x_m) = (x_1 + k_1, x_2 + k_2, \ldots, x_m + k_m)$$

and

$$d_K(y_1, y_2, \ldots, y_m) = (y_1 - k_1, y_2 - k_2, \ldots, y_m - k_m).$$

All operations are performed in $\mathbb{Z}_{26}$.

3.2.5 Vernam's One-time pad

Let $n \geq 1$ and
let $\mathcal{P} = \mathcal{C} = \mathcal{K} = (\mathbb{Z}_2)^n$.
For $K \in (Z_2)^n$, define

$$e_K(x) = (x_1 + K_1, \ldots, x_n + K_n) \ mod\ 2$$

and

$$d_K(y) = (y_1 + K_1, \ldots, y_n + K_n) \ mod\ 2.$$

The famous *one-time pad* has unconditional perfect secrecy. If, when using the *Vigenère* cipher, we use a new random key for each encryption then we have perfect secrecy. This can be viewed as a generalization of the *One-time pad* from a binary to an arbitrary alphabet .

3.2.6 Hill cipher

Let m be a fixed integer and
let $\mathcal{P} = \mathcal{C} = (\mathbb{Z}_{26})^m$, $\mathcal{K} = \{m \times m \ invertible\ over\ \mathbb{Z}_{26}\}$.
For $K \in \mathcal{K}$, define

$$e_K(x) = x \cdot K$$

and

$$d_K(y) = y \cdot K^{-1}$$

Note that a *permutation cipher* is a special case of the *Hill cipher* in which only $m!$ (permutation matrices) of all the possible invertible $m \times m$ matrices are used. Such a cipher is permuting blocks of m letters in a fixed reversible way.

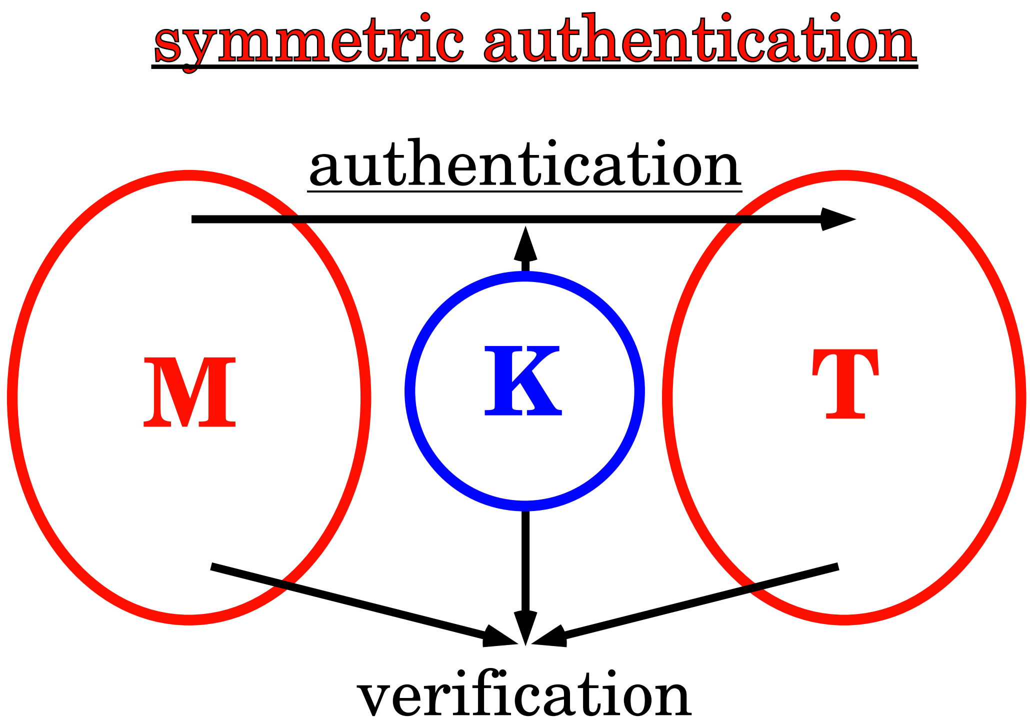
3.3 Cryptanalysis: classes of attacks

There are 4 basic classes of attacks on a cryptosystem. In every case, the encryption-decryption scheme is known to everyone and an attacker *Oscar* is interested in recovering the plaintext corresponding to a specific ciphertext, or even more drastically, deduce the decryption key of the scheme in use. The four classes of attacks are presented in the following table:

Class	description
ciphertext-only	Oscar tries to deduce the plaintext or decryption key using only the ciphertext.
known plaintext	Oscar has access to a series of ciphertext-plaintext pairs.
chosen plaintext	Oscar is given the ciphertexts corresponding to the plaintexts of his choice.
chosen ciphertext	Oscar is given the plaintexts corresponding to the ciphertexts of his choice.

4 Authentication Codes

A message authentication code (MAC) is essentially a scheme where *Alice* may append a tag (a MAC) to a message in such a way that *Bob* may verify the tag so as to convince himself that *Alice* was in fact the one that sent the message.



In this section, we present an authentication code that is unconditionally secure. The main results in this section is from [**?**].

Formally, an Authentication Code is defined as follows:

Definition 4.1 *Let $\mathcal{M}$ be a finite set of messages and $\mathcal{T}$ a finite set of authentication tags such that for each $k \in \mathcal{K}$, there is an authentication algorithm aut_k and a corresponding verification algorithm ver_k such that $aut_k : \mathcal{M} \rightarrow \mathcal{T}$ and $ver_k : \mathcal{M} \times \mathcal{T} \rightarrow \{true, false\}$ are polynomial-time computable functions and*

$$ver_k(m, t) = \begin{cases} true & : \quad if\ t = aut_k(x) \\ false & : \quad if\ t \neq aut_k(x) \end{cases}$$

The security of an authentication code is defined in terms of probability for an adversary to predict a proper tag t corresponding to a message m

he has never seen authenticated before. In the rest of this section we build authentication codes that are based on the notion of *Strongly Universal$_2$* hash functions.

Definition 4.2 (*Strongly Universal$_2$*) *Let H is a set of hash functions from set A to B.*
H is Strongly Universal$_2$ if for all a_1, a_2, distinct elements of A, and all b_1, b_2, elements (not necessarily distinct) of B, we have

$$|\{h \in H : h(a_1) = b_1, h(a_2) = b_2\}| = |H|/|B|^2$$

Remark: An equivalent definition is the following, H is *Strongly Universal$_2$* if for any h picked randomly (uniformly) from H we have that

1. $\forall_{a \in A, b \in B}\ Pr[h(a) = b] = 1/|B|$
2. $\forall_{a_1, \neq a_2 \in A, b_1, b_2 \in B} Pr[h(a_1) = b_1, h(a_2) = b_2] = 1/|B|^2$
3. $\forall_{a_1, \neq a_2 \in A, b_1, b_2 \in B} Pr[h(a_2) = b_2 | h(a_1) = b_1] = 1/|B|$

We need 1 and 3 for perfect authentication.

The definition can be generalized

Definition 4.3 (*Strongly Universal$_n$*) *Let H is a set of hash functions from set A to B. H is Strongly Universal$_n$ if for all $a_1, a_2, ..., a_n$, distinct elements of A, and all $b_1, b_2, ..., b_n$, elements (not necessarily distinct) of B, we have*

$$|\{h \in H : h(a_1) = b_1, ..., h(a_n) = b_n\}| = |H|/|B|^n$$

Remark: If H is *Strongly Universal$_n$* then it is *Strongly Universal$_{n-1}$*

Example 4.1 *Let $A = B$ be a finite field. Let H be the class of polynomials of degree less than n. H is Strongly Universal$_n$ since given any n distinct elements of A and corresponding elements of B, there is exactly one polynomial of degree less than n which interpolates through the designated pairs.*

We can create an authentication system that is unbreakable with certainty p. To do this, we simply choose $\mathcal{T}$ such that $|\mathcal{T}| \geq 1/p$ and F to be a *Strongly Universal$_2$* class of hash functions from $\mathcal{M}$ to $\mathcal{T}$. Someone who sees $m \in \mathcal{M}$ and $t = f(m)$ knows only that $f \in H'$ where

$H' = \{g \in H \mid g(m) = t\}$, so then, by definition of *Strongly Universal$_2$*, guessing a correct function that maps an $m' \neq m \in \mathcal{M}$ happens with probability $\leq p$ since a fraction $1/|\mathcal{T}|$ of all the functions from H' agree with $g(m') = t'$.

The problem with this protocol is that all know *Strongly Universal$_2$* sets are rather large (see [**?**] for such sets), and specifying a certain function from these sets requires a key at least as long as the original message. A second problem is that only one message can be sent with a certain key, knowledge of two message-tag pairs may give information on the values of the function of some third message.
We can do better than this. We will first show a protocol that solves the first problem, and then one that solves the second, both come from [**?**].

We first define the following class:

$$H_2 = \{h : \mathcal{F}_q \to \mathcal{F}_q | h(a) = ia + j \; for some \, i, j \in \mathcal{F}_q\}$$

here, $A = B = \mathcal{F}_q$, $|A| = |B| = q$, $|H| = q^2$.

Theorem 4.4 *H_2 is Strongly Universal$_2$.*

Proof. Consider $a \neq a'$, and two outputs b, b',

$$\begin{array}{rr} & ia + j = b \\ - & \underline{ia' + j = b'} \\ & i(a - a') = (b - b') \end{array}$$

$\Rightarrow i = (b - b')(a - a')^{-1}$ (we are in a field: $(a - a')$ exists and is unique) and so $j = b - ia = b - (b - b')(a - a')^{-1}a$.

These values of i and j define a unique h such that $h(a) = b, h(a') = b'$. We thus have that

$$\forall_{a \neq a', b, b'} |\{h : \; h(a) = b, h(a') = b'\}| = 1 = |H_0|/|B|^2$$

Unfortunately, the key to authenticate a message is twice as big as the message itself. Moreover, if we send very long messages it is not necessary to have probability $1/|B|$ of defeating the authentication. We may be happy with probability, say, $1/2^{50}$. In this case we use the following class instead:

$$H_{cut} = \{h : \mathcal{F}_{p^m} \to \mathcal{F}_{p^n} | h(a) = (ia + j)_{[last \, n \, symbols]}, \; i, j \in \mathcal{F}_{p^m}\}$$

here $A = \mathcal{F}_{p^m}, |A| = p^m$ and $B = \mathcal{F}_{p^n}, |B| = p^n$

Theorem 4.5 *H_{cut} is Strongly Universal$_2$.*

Proof. We leave the proof to the reader.

4.1 Multiple Messages

Using the above method, if an adversary sees two message-tag pairs, he may be able to determine more such pairs (by solving linear equations). One way around the problem is to use *Strongly Universal$_n$* functions, so that we can send $n-1$ messages. But there is a more elegant way: Let F be a *Strongly Universal$_2$* set of functions from A to B. To each message in M that we send, we append an unique number i between 1 and n. The sender (*Alice*) randomly chooses a $f \in F$ and randomly chooses n, $\lg|B|$ sized, one-time pads $b_1, b_2, ..., b_n$. She secretly shares these values $(f, b_1, b_2, ..., b_n)$ with the receiver (*Bob*). To create a tag t_i for message $i\|m$ (a message with i appended in front of it), *Alice* computes $f(i\|m) \oplus b_i$. When *Bob* receives a message $i\|m$ with a tag t_i, he accepts it iff $t_i \oplus b_i = f(i\|m)$.

Now the difference with before is that an adversary never sees a pair of message-tag; the tags he sees are always encrypted with a one-time-pad...

Theorem 4.6 *In the context of the above protocol, an adversary knowing only the set F and n pairs $(m_1, t_1), (m_2, t_2), ..., (m_n, t_n), m_i \neq m_j$ for $i \neq j$, cannot create a tag t'_i for a different message m'_i (containing i as a prefix) with probability of success greater than $1/|B|$*

Proof. The proof is left to the reader.

Theorem 4.7 *In the context of the above protocol, an adversary knowing only the set F and n pairs $(m_1, t_1), (m_2, t_2), ..., (m_n, t_n), m_i \neq m_j$ for $i \neq j$, cannot create a set of n valid message-tag pairs $(m'_1, t'_1), (m'_2, t'_2), ..., (m'_n, t'_n)$, $m'_i \neq m'_j$ for $i \neq j$ (and each m'_i containing i as a prefix) with probability of success greater than $1/|B|^k$ if k of the n pairs are distinct from the originals.*

Proof. The proof is left to the reader.

This stronger theorem works only because we appended the index i of each message in front of each m_i. Otherwise, if two messages m_i and m_j were identical the probability of substituting two message-tag pairs $(m_i, t_i), (m_j, t_j)$ by a different $(m'_i, t'_i), (m'_j, t'_j)$ for $m'_i = m'_j$ is at least $1/|B|$ by setting $t'_j = t'_i \oplus t_i \oplus t_j$. This follows from the fact that if t'_i happens to be correct, so will t'_j.

5 Identification Schemes

An identification scheme allows *Alice* to prove knowledge of a common secret key k in such a way that *Bob* may verify k if he already knows it, but will fail with high probabiilty to learn k if he does not already know it. This is typically used for password or PIN verification. In this first section we consider simple one-time identification schemes and show their (in)security.

Let $k = k_1 k_2 ... k_t$ be the binary representation of k.

5.1 PIN model

Let $\mathcal{K} = \{0, 1\}^t$.
Alice reveals k to *Bob* who accepts if k is valid.

Theorem 5.1 *This system has no security whatsoever. If Bob does not know k he learns it from Alice and then can use it at will.*

5.2 broken PIN model

Let $\mathcal{K} = \{0, 1\}^t$.
Alice reveals $k_1 ... k_{t/2}$ to *Bob* who accepts if they are valid.
Bob reveals $k_{t/2+1} ... k_t$ to *Alice* who accepts if they are valid.

Theorem 5.2 *This system has no security whatsoever. If Bob does not know k he learns $k_1 ... k_{t/2}$ from Alice and then can use them at will as Alice.*

5.3 interactive PIN model

Let $\mathcal{K} = \{0, 1\}^t$.
for $i := 1$ **to** $t/2$
Alice reveals k_i to *Bob* who accepts if it is valid.
Bob reveals $k_{t/2+i}$ to *Alice* who accepts if it is valid.
If an invalid bit is found then *Alice* or *Bob* aborts.

Theorem 5.3 *If Bob does not know k he will learn more than $\ell \leq t$ bits from Alice with probability only $2^{-\ell}$.*

5.4 hybrid PIN model

Let $\mathcal{K} = \{0,1\}^t$.
Bob picks a random subset S of indices such that $|S| = t/2$ and announces it to *Alice*.
Alice reveals k_S to *Bob* who accepts if it is valid.
Bob reveals $k_{\bar{S}}$ to *Alice* who accepts if it is valid.
If an invalid bit is found then *Alice* or *Bob* aborts and should report her (his) key stolen.

Theorem 5.4 *If Bob does not know k he may learn $t/2$ bits from Alice but will be able to answer a challenge issued by a third party Bill with probability roughly $2^{-t/4}$.*