Cryptography: Handout 1

Based on notes by William Gasarch

We will use three characters: Alice and Bob who want to communicate secretly, and Eve who wants to see what they are talking about. Alice and Bob do not want Eve to be able to decode their messages.

1 Shift Cipher

Alice and Bob have wanted to exchange secret messages for the last 4000 years. One of the earliest techniques for this, called the *Caesar Cipher*, operates as follows.

First imagine all letters as numbers. A is 0, B is 1, C is 2, etc, Z is 25. Map every letter to the letter that is three higher (modulo 26). So, the the last three letters shift to the first three. So

A goes to D
B goes to E
⋮
V goes to Y
W goes to Z
X goes to A
Y goes to B
Z goes to C.

More generally, a *shift cipher* is a code where every letter shifts a constant amount.

Are shift ciphers good?

PROS

1. The scheme is easy to describe, easy to code, and easy to decode. So Alice and Bob can operate very fast.

2. Alice and Bob only have to agree on the shift. Since the shift is in $\{1, \ldots, 25\}$, they can easily communicate to each other which shift to use.

CONS

1. The scheme is easy so Eve may spot the pattern.

2. If Eve knows that it is a shift cipher then she can just try all 25 possible shifts.

3. Alice and Bob do have to meet privately once to agree on the shift. (Is this avoidable?)

2 Linear Cipher

We can represent a shift of s by $f(x) = (x+s) \bmod 26$. We can use a more complicated function. For example

$$f(x) = (3x + 4) \bmod 26$$

or

$$f(x) = (5x - 7) \bmod 26.$$

Are these codes good?

PROS

1. The scheme is easy to describe, easy to code, and easy to decode (once you know the trick). So Alice and Bob can operate very fast, though not as fast as with the shift cipher.

2. Alice and Bob only have to agree on the multiplier and the shift. This amounts to knowing two numbers from $\{1, \ldots, 25\}$. We represent the numbers in base 2. Each number is 5 bits long, so two numbers take 10 bits. This is small.

CONS

1. Not all choices of parameters lead to 1-1 functions (in which case they cannot be used for coding). However, determining which ones can be used is not hard.

2. The scheme is easy so Eve may spot the pattern, though it's not as easy as the Shift Cipher.

3. If Eve knows that it is a linear cipher then she can just try all 625 possible shifts. Notice that this is harder than for a shift cipher. (Actually the number of possibilities can be reduced since some of them do not yield 1-1 functions.)

4. Alice and Bob do have to meet privately to agree on the parameters. (Is this avoidable?)

3 Quadratic Cipher

One can look at even more complicated functions such as

$$f(x) = (2x^2 + 5x + 9) \bmod 26.$$

These are called quadratic ciphers. They have similar PROS and CONS to linear ciphers.

4 Any Permutation

Alice and Bob pick a *random* permutation of $\{A, \ldots, Z\}$.

Is this a good code?

PROS

1. It seems as though Eve has to try 26! possibilities.

CONS

1. The key Alice and Bob use is a list of the letters of the alphabet in some order. In base 2 this is $26 \times 5 = 130$ bits (though it can be done in somewhat less).
2. Alice and Bob do have to meet in secret to estabish the key. (Is this avoidable?)
3. This is the real problem: Eve doesn't actually have to go through all the possiblities. She can use frequency analysis. *Throughout history codes thought unbreakable were broken because the way to break them was unrelated to how they were derived.*

Frequency analysis uses the fact that we know how letters are distributed in English. For example e is the most common letter in the alphabet and th is the most common pair. Using this one can do a statistical analysis on a coded text and (if it is long enough) crack it.

5 Matrix Codes

Let A be the following matrix.

$$\mathbf{A} = \begin{pmatrix} 8 & 9 \\ 11 & 7 \end{pmatrix}$$

We can map *pairs of numbers* with this matrix as follows. The pair (x, y) will map to the pair you get by applying the matrix and reducing modulo 26, which is

$$((8x + 9y) \bmod 26, (11x + 7y) \bmod 26).$$

From start to finish: take a text, convert the letters to numbers, (assume it has an even number of letters), break the sequence of numbers into blocks of 2 numbers each, and apply the matrix to each pair to get an encoded pair.

Notice that this can be extended to 3×3 matrices or more generally $k \times k$ matrices.

Think about the PROS and CONS.

6 An Uncrackable Code: the One-Time Pad

Definition 6.1 If a and b are bits (0 or 1) then $\oplus$ (also written XOR and called "exclusive or") is defined as follows:

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

The following facts are easy to verify.

Fact 6.2 *Let a, b, c be bits.*

1. $(a \oplus b) \oplus c = a \oplus (b \oplus c)$.
2. *For all bits a, $a \oplus a = 0$.*
3. $a \oplus b \oplus b = a \oplus (b \oplus b) = a \oplus 0 = a$.

We now describe the one-time pad.

1. Alice and Bob have to meet (or communicate over a secure channel that Eve cannot listen to) and agree on a randomly generated sequence of bits – a VERY long sequence. Say it's
$$r_1 r_2 \cdots r_N.$$
This is called *the key*. They then part.

2. If (later) Alice wants to send
$$a_1 a_2 a_3 \cdots a_m$$
she sends
$$(r_1 \oplus a_1)(r_2 \oplus a_2) \cdots (r_m \oplus a_m).$$
When Bob gets this string, which he sees as
$$s_1 \cdots s_m$$
he can decode it by taking
$$\begin{aligned}(r_1 \oplus s_1)(r_2 \oplus s_2) \cdots (r_m \oplus s_m) &= (r_1 \oplus (r_1 \oplus a_1))(r_2 \oplus (r_2 \oplus a_2)) \cdots (r_m \oplus (r_m \oplus a_m)) \\ &= ((r_1 \oplus r_1) \oplus a_1)((r_2 \oplus r_2) \oplus a_2)) \cdots ((r_m \oplus r_m) \oplus a_m)) \\ &= a_1 a_2 \cdots a_m\end{aligned}$$

3. If either Alice or Bob wants to send another message they will start with r_{m+1}.

PROS: This is impossible to crack! Since the original key was random, if Eve sees the message

$$s_1 s_2 \cdots s_m$$

it will look random to her.

CONS: N is LARGE! They have to meet and exchange A LOT of information. In fact, if they plan to later communicate N bits they need to have a key of length N.

PROBLEM: Can Alice and Bob use a shorter key?

PROBLEM: Can Alice and Bob agree on a secret key (e.g., $r_1 r_2 \cdots r_N$) without having to meet?

Cryptography: Handout 2

Based on notes by William Gasarch

7 Our Goal

The following problem plagues all of the systems we have considered: Alice and Bob must meet in secret to establish a key.

Is there a way around this? Is there a way for Alice and Bob to NEVER meet, and yet establish a secret key? That is, can they, by talking *in public* establish a shared secret key?

The answer will be yes, assuming that whoever is listening in has some limits on what they can compute.

8 Needed Math

We'll use multiplication modulo p in the set $Z_p = \{1, 2, \ldots, p-1\}$, where p is a prime number. It will be useful to find an element $g \in Z_p$, called a "generator", for which the sequence $g^0, g^1, g^2, \ldots, g^{p-2}$, taken modulo p, contains all of the elements of Z_p.

Let's look at $p = 11$. Notice that

$$\begin{aligned}
2^0 &\equiv 1 \pmod{11} \\
2^1 &\equiv 2 \pmod{11} \\
2^2 &\equiv 4 \pmod{11} \\
2^3 &\equiv 8 \pmod{11} \\
2^4 &\equiv 5 \pmod{11} \\
2^5 &\equiv 10 \pmod{11} \\
2^6 &\equiv 9 \pmod{11} \\
2^7 &\equiv 7 \pmod{11} \\
2^8 &\equiv 3 \pmod{11} \\
2^9 &\equiv 6 \pmod{11}
\end{aligned}$$

These calculations are not hard if you use that $2^n \equiv 2 \times 2^{n-1} \pmod{11}$. Notice that $\{2^0 \bmod 11, 2^1 \bmod 11, \ldots, 2^9 \bmod 11\} = \{1, 2, \ldots, 10\}$.

Do all elements of Z_{11} generate the entire set? No:

$$\begin{array}{lll}
5^0 & \equiv 1 & \pmod{11} \\
5^1 & \equiv 5 & \pmod{11} \\
5^2 & \equiv 3 & \pmod{11} \\
5^3 & \equiv 4 & \pmod{11} \\
5^4 & \equiv 9 & \pmod{11} \\
5^5 & \equiv 1 & \pmod{11} \\
5^6 & \equiv 5 & \pmod{11} \\
5^7 & \equiv 3 & \pmod{11} \\
5^8 & \equiv 4 & \pmod{11} \\
5^9 & \equiv 9 & \pmod{11}
\end{array}$$

Notice that $\{5^0 \bmod 11, 5^1 \bmod 11, \ldots, 5^9 \bmod 11\} = \{1, 3, 4, 5, 9\}$. This is NOT all of Z_{11}.

Convention 8.1 We will be using a prime p. We will assume that p is LARGE but that $\log p$ is not too large. Hence if Eve needs a computation of p steps to crack a code we will consider it a good code. Even if Eve needs a computation of $\sqrt{p}$ steps (or p^ϵ steps where $\epsilon > 0$) this is a long time and we will consider it a good code. Also, if Alice and Bob have to do operations that take $\log p$ steps, that's okay, they can do that. Even if they have to take $(\log p)^2$ (or some larger polynomial in $\log p$) thats okay, they can do that.

Convention 8.2 For the rest of this document when we say "roughly p" we will mean p^ϵ for some ϵ, $\epsilon > 0$. When we say "roughly $\log p$" we will mean $(\log p)^a$ for some $a \in N$.

Theorem 8.3 *For every prime p there is a g such that $\{g^0 \bmod p, g^1 \bmod p, \ldots, g^{p-2} \bmod p\} = Z_p = \{1, \ldots, p-1\}$. There is an algorithm which will, given p, find such a generator g in roughly $\log p$ steps.*

We have already seen that $+, -, \times$, and (if p is prime) division can be done modulo p. We now have a way to do LOGARITHMS modulo p.

Definition 8.4 Let p be a prime and g be a generator of Z_p. Let $x \in Z_p$. The *Discrete Logarithm of x with base g* is the $y \in \{0, \ldots, p-2\}$ such that $g^y \equiv x \pmod{p}$. We denote this $DL_g(x)$.

Example 8.5 We rewrite the table above for $p = 11$ and add to it. The Discrete Logarithm lines follow from the prior line. We assume $g = 2$ and denote DL_2 by just DL.

$$
\begin{aligned}
2^0 &\equiv 1 \pmod{11} \\
DL(1) &= 0 \\[1em]
2^1 &\equiv 2 \pmod{11} \\
DL(2) &= 1 \\[1em]
2^2 &\equiv 4 \pmod{11} \\
DL(4) &= 2 \\[1em]
2^3 &\equiv 8 \pmod{11} \\
DL(8) &= 3 \\[1em]
2^4 &\equiv 5 \pmod{11} \\
DL(5) &= 4 \\[1em]
2^5 &\equiv 10 \pmod{11} \\
DL(10) &= 5 \\[1em]
2^6 &\equiv 9 \pmod{11} \\
DL(9) &= 6 \\[1em]
2^7 &\equiv 7 \pmod{11} \\
DL(7) &= 7 \\[1em]
2^8 &\equiv 3 \pmod{11} \\
DL(3) &= 8 \\[1em]
2^9 &\equiv 6 \pmod{11} \\
DL(6) &= 9
\end{aligned}
$$

COMMON BELIEF: It is believed that the problem of computing the discrete logarithm *requires* roughly p steps. This is a long time, so we assume Eve cannot do this.

Fact 8.6

1. *Given p, finding a generator for Z_p can be done in roughly* $\log p$ *steps.*
2. *Given L, finding a prime of size around L can be done in roughtly* $\log L$ *steps.*
3. *Given p, $a \in \{0, 1, \ldots, p-1\}$, and m, determining* $a^m \bmod p$ *takes roughly* $\log m$ *steps. (This is by repeated squaring.)*

9 Diffie Helman Key Exchange

We can USE this mathematics to have Alice and Bob exchange information in public and in the end they have a shared secret key.

1. Alice generates a large prime p and a generator g (this takes roughly $\log p$ steps) and sends it to Bob over an open channel. So now Alice and Bob know p, g but so does Eve.

2. Alice generates a random $a \in \{0, \ldots, p-2\}$. Bob generates a random $b \in \{0, \ldots, p-2\}$. They keep these numbers private. Note that even Alice does not know b, and even Bob does not know a.

3. Alice computes $g^a \bmod p$. Bob computes $g^b \bmod p$. Both use repeated squaring so it takes roughly $\log p$.

4. Alice sends Bob $g^a \bmod p$ over an open channel. Notice that Eve will NOT be able to compute a if computing DL_g is hard (which is the common belief). Even Bob won't know what a is.

5. Bob sends Alice $g^b \bmod p$. Notice that Eve will NOT be able to compute b if computing DL_g is hard. Even Alice won't know what b is.

6. RECAP: Alice now has a and g^b. SHE DOES NOT HAVE b. Bob has b and g^a. HE DOES NOT HAVE a. Eve has g^a and b^b. SHE DOES NOT HAVE a OR b.

7. Alice computes $(g^b)^a \bmod p = g^{ab} \bmod p$. Bob computes $(g^a)^b \bmod p = g^{ab} \bmod p$. They both use repeated squaring so this is fast.

8. SO at the end of the protocol they BOTH know $g^{ab} \bmod p$. This is their shared secret key. Eve likely does NOT know g^{ab} since she only gets to see g^a and g^b.

This scheme LOOKS good but we must be very careful about what is known about it.

1. Alice and Bob can execute the scheme quickly.

2. If Eve can compute DL_g quickly then she can crack the code.

3. There MIGHT BE other ways for Eve to crack the code. That is, being able to compute DL_g quickly is sufficient to crack this scheme, but might not be neccesary.

4. This scheme can be used for Alice and Bob to establish a secret key without meeting. This can then be used in other schemes such as the one-time pad.

5. Reality: This scheme is used in the real world for secret key exchange. lThe RSA algorithm is used for Public Key Cryptography (which is similar).