Introduction to Mathematical Cryptography I

Tamir Enkhjargal and Stanislav Lyakhov

July 2017

1 Table of Contents

2 Shift Ciphers

2.1 Simple Shift Ciphers

Most, if not all ciphers take a *plaintext* message, and converts it into a *ciphertext* message.

The simple shift cipher, also most commonly known as the **Caesar Cipher**, uses two alphabets of same size and "converts" one letter to the other.

First Alphabet	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Second Alphabet	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e

Using the First Alphabet as our *plaintext* alphabet, and the Second Alphabet (shifted 5 letters) as our *ciphertext* alphabet, we can now convert any message into an encrypted message.

shift ciphers have been used for thousands of years now
xmnky hnumj wxmfa jgjjs zxjik twymt zxfsi xtkdj fwxst b

Note: It's common practice that encrypted messages be put in "blocks" of a set number of letters, to not give away word sizes.

We can confirm by shifting our ciphertext back 5 letters to get back our plaintext

Resources

- Cipher Encoder/Decoders - Rumkin.com/tools/cipher/
- Extremely Versatile Website for "All Your Needs" - Dcode.fr/tools-list
- SSTCTFs Dictionary-Attack Caesar Decoder - Github.com/SST-CTF/cipher-tools
- Another Bruteforce Decoder - Quipqiup.com

2.2 Vigenére Ciphers

The Vigenére Cipher, an extension of the practice of shift ciphers, does not require the two alphabets to be of the same size.

In fact, the "Second Alphabet" is replaced by the **Key**, and it is **repeated** when encrypting our plaintext

Plaintext	t	h	i	s	i	s	a	t	e	s	t	f	o	r	t	h	e	v	i	g	e	n	e	r	e
Key	s	s	t	c	t	f	s	s	t	c	t	f	s	s	t	c	t	f	s	s	t	c	t	f	s
Ciphertext	l	z	b	u	b	x	s	l	x	u	m	k	g	j	m	j	x	a	a	y	x	p	x	w	w

This works by using the number value of each letter to be the "shift" value. 'a' would shift a letter by 0, and 'b' by 1, and so on.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

This is the *Tabula Recta*, created in 1508, and was used to encrypt/decrypt keyed ciphers We can see one by one that each letter in our plaintext is shifted by s(18), t(19), c(2), or f(5).

2.2.1 Cryptanalysis of the Vigenére Cipher

The first method of cryptanalysis on Vigenére was to estimate the *period* or the *blocksize* of the key. Sometimes, at random, certain words line up with the same portion of the key, and show in the ciphertext. In a message long enough, the probability that the most common bigrams (two-letter combinations) and trigrams (three-letter combinations) appear increase. This method was called the Kasiski test. Imagine if you had to solve a Vigenére Cipher that was 6 letters long, with a 6 length key. It would be virtually impossible unless you knew a portion of the key or original plaintext.

Putting the Kasiski test to use, we will test for common trigrams that appear in the message:

```
zpgdl rjlaj kpylx zpyyg lrjgd lrzhz qyjzq repvm swrzy rigzh
zvreg kwivs saolt nliuw oldie aqewf iiykh bjowr hdogc qhkwa
jyagg emisr zqoqh oavlk bjofr ylvps rtgiu avmsw lzgms evwpc
dmjsv jqbrn klpcf iowhv kxjbj pmfkr qthtk ozrgq ihbmq sbivd
ardym qmpbu nivxm tzwqv gefjh ucbor vwpcd xuwft qmoow jipds
fluqm oeavl jgqea lrkti wvext vkrrg xani
```

Table 1: Table 2.2.1: Cryptanalysis of Ciphertext

Trigram	Location	Difference
avl	117 and 258	$141 = 3 \cdot 47$
bjo	86 and 121	$35 = 5 \cdot 7$
dlr	4 and 25	$21 = 3 \cdot 7$
gdl	3 and 24	$21 = 3 \cdot 7$
lrj	5 and 21	$16 = 2^4$
msw	40 and 138	$98 = 2 \cdot 7^2$
pcd	149 and 233	$84 = 2^2 \cdot 3 \cdot 7$
qmo	241 and 254	$13 = 13$
vms	39 and 137	$98 = 2 \cdot 7^2$
wpc	147 and 231	$84 = 2^2 \cdot 3 \cdot 7$
zhz	28 and 49	$21 = 3 \cdot 7$

Repeated Trigrams in ciphertext, and their estimated periodicity

We put the ciphertext at a Vigenére Decipher, with our key size guess being 7.

However, more often than not, we will let the computer do a frequency statistical attack on the ciphertext. We can see that dcode's frequency attack is a lot more efficient and faster than the Kasiski test.

Note: there is also a mathematical formula that attempts to statistically predict the key size, letter frequency, or periodicity of the message.

3 The Substitution Cipher

3.1 Simple Substitution Ciphers

Simple substitution ciphers, also called **monoalphabetic ciphers**, instead of following an encryption system, it follows a replacement system[1]. For example, the **Atbash Cipher** follows the pattern,

Plaintext	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Ciphertext	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A

Therefore, any time there is an "A" in our plaintext message, we will see a "Z" in the ciphertext. Because each replacement is fixed (one-to-one), the ciphertext is susceptible to cryptanalysis through language infrequencies.

3.2 Frequency Analysis

The English language (and many others) use some letters more than others, obviously. Letters like "e" or "t" are more often used than "q" or "x".

Overall, letter frequencies follow this pattern:

By Decreasing Frequencies [2]				
E	13.11%	T	10.47%	} 39.73%
A	8.15%	O	8.00%	
N	7.10%	R	6.83%	} 26.38%
I	6.35%	S	6.10%	
H	5.26%	D	3.79%	} 15.36%
L	3.39%	F	2.92%	
C	2.76%	M	2.54%	} 9.75%
U	2.46%	G	1.99%	
Y	1.98%	P	1.98%	} 6.94%
W	1.54%	B	1.44%	
V	0.92%	K	0.42%	} 1.64%
X	0.17%	J	0.13%	
Q	0.12%	Z	0.08%	} 0.2%

We can see here that the English language has large variances in letter frequencies. We can say from these percentages, on average, 40% of words either have an E, A, T, or an O. This creates a system where we can see the letter frequency of the ciphertext, and start to predict them to these letter frequencies.

[1] In monoalphabetic cipher systems, where each letter/character is replaced one-to-one with another character
[2] Percentages taken from Hoffstein's *An Introduction to Mathematical Cryptography*

3.3 Common Bigrams

We can also do cryptanalysis through bigrams, or two-letter combinations. In many situations, the letter frequencies will not match the average letter frequencies, and will sometimes prove to be inefficient or slow. When trying to cryptanalysis using letter frequencies, there's a good possibility that many of the uncommon letters appear more often than even e, t, a, or o. That's why bigram cryptanalysis also exists.

The most common bigram frequency and usages are as such (per 1000 words)[3]:

th	he	an	re	er	in	on	at	nd	st	es	en	of	te	ed
168	132	92	91	88	86	71	68	61	53	52	51	49	46	46

Let's make a contest to see who can solve this first, and how long. Remember to look for bigrams, letter, and even trigram frequencies.

```
LOJUM YLJME PDYVJ QXTDV SVJNL DMTJZ WMJGG YSNDL UYLEO SKDVC
GEPJS MDIPD NEJSK DNJTJ LSKDL OSVDV DNGYN VSGLL OSCIO LGOYG
ESNEP CGYSN GUJMJ DGYNK DPPYX PJDGG SVDNT WMSWS GYLYS NGSKJ
CEPYQ GSGLD MLPYN IUSCP QOYGM JGCPL GDWWJ DMLSL OJCNY NYLYD
LJQLO DLCNL YPLOJ TPJDM NJQLO JWMSE JGGJG XTUOY EOOJO DQDMM
YBJQD LLOJV LOJTV YIOLU JPPES NGYQJ MOYVD GDNJE MSVDN EJM
```

[3]Values taken from Hoffstein's *An Introduction to Mathematical Cryptography*

The solution to the substitution cipher, in block form, is:

```
THEWR ITERC LAIME DBYAM OMENT ARYEX PRESS IONAT WITCH OFAMU
SCLEO RAGLA NCEOF ANEYE TOFAT HOMAN ANSIN MOSTT HOUGH TSHIS
CONCL USION SWERE ASINF ALLIB LEASS OMANY PROPO SITIO NSOFE
ECLID SOSTA RTLIN GWOUL DHISR ESULT SAPPE ARTOT HEUNI NITIA
TEDTH ATUNT ILTHE YLEAR NEDTH EPROC ESSES BYWHI CHHEH ADARR
IDEDA TTHEM THEYM IGHTW ELLCO NSIDE RHIMA SANEC ROMAN CER
```

```
The writer claimed by a momentary expression, a twitch of a muscle or a glance of an
eye, to fathom a man's inmost thoughts.  His conclusions were as infallible as so many
propositions of Euclid.  So startling would his results appear to the uninitiated that
until they learned the processes by which he had arrived at them they might well consider
him as a necromancer.[4]
```

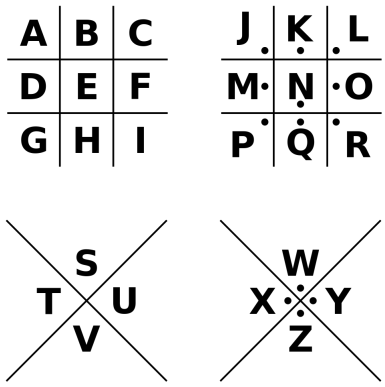
3.4 Common Variants of Substitution Ciphers

Polyalphabetic Substitutions:

Instead of **Monoalphabetic Substitutions**, which indicate a one-to-one substitution between two alphabets, polyalphabetic substitutions indicate multiple cipher alphabets to be used. The most famous and popular one is the **Vigénere** cipher, which we already went over. Others also include the autokey cipher, running key cipher, and by extension stream ciphers.

Symbolic Substitutions:

Also a very popular variant of the substitution cipher. Substitution ciphers are in essence the exact same as regular substitution ciphers. There is a plaintext alphabet (typically A-Z), and there is a ciphertext alphabet (anything you want).



The Pigpen Cipher, assigns each plaintext letter to a certain symbol. Of course, you can reassign any letter to any symbol, as it would make no difference when using cryptanalysis using letter frequencies and bigram frequencies.

Generally, simple symbol substitutions are one of the more simple problems, as the ciphertext alphabet either already exists, or can be solved through computer analysis.

[4] *A Study in Scarlet* (Chapter 2), Sir Arthur Conan Doyle

4 Advanced Encryption Standard (AES)

4.1 Overview

Advanced Encryption Standard (AES) is a symmetric 128 bit block cipher, which was submitted to the 2001 AES selection process by Vincent Rijmen and Joan Daemen. After intense testing and investigation, AES was selected by the U.S. National Institute of Standards and Technology (NIST) to be the new standard of encryption. As part of the requirement of NIST, the algorithm supports three different key-lengths: 128, 192, and 256 bit. The algorithm itself consists of rounds–functions with multiple layers that modify the text.

# rounds	key-length
10	128
12	192
14	265

Figure 1: Amount of rounds depending on key-length

Almost every round (excluding the first and last round) consist of four layers: Byte Substitution, Shift Row, Mix Column, and Key addition. For additional security, at the very beginning of the first round and very end of the last round, an additional subkey is XORed with the message. This technique, called key whitening, increases security and is done in most modern ciphers. Unlike all other rounds, the last round does not posses a MixCol layer.

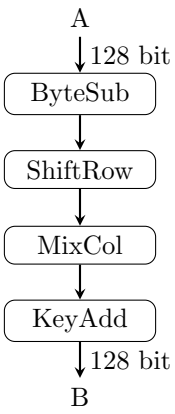


Figure 2: Layers of one round of AES

4.2 Internals

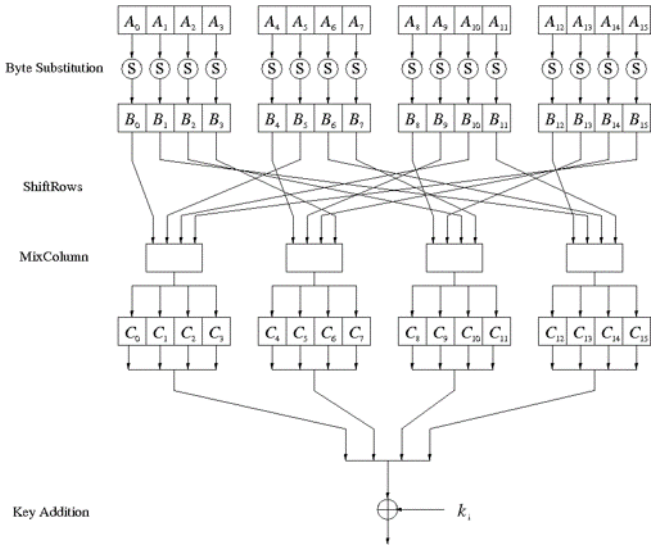


Figure 3: Internals of an AES round

In AES, 128 bit block input is divided into 16 bytes. Each goes through the first layer (S-Box) separately.

4.2.1 Byte Substitution Layer (S-Box Layer)

The S-Box Layer acts as the main source of *confusion* in the AES cipher. It consists of two parts: a look-up table and affine transformation. The look-up table is built non-linearly in order to withstand differential cryptanalysis attacks. This is done by determining the multiplicative inverse for every given number in a finite field $GF(2^8)$. In order to remove correlation to mathematical property, the bytes go through another transformation before exiting the S-Box.

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xa	xb	xc	xd	xe	xf
0x	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1x	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2x	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3x	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4x	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5x	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6x	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7x	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8x	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9x	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
ax	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
bx	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
cx	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
dx	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
ex	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
fx	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Figure 4: S-Box lookup table

4.2.2 Shift Rows Layer

The ShiftRow layer shuffles bytes out of the S-Box layer and groups them into the MixCol layer, 4 bytes per MixCol box. This adds extra *diffusion* to the system. The shuffling might seem random at first, but it is very systematic:

1. All 16 bytes are organized in a matrix. First four bytes in the first col, next four in second col etc.
2. The first row remains unchanged. The second row is shifted one to the left, the third row two to the left, and the fourth row three to the left.
3. Each column is sent forward (to the MixCol Layer) in respective order.

4.2.3 Mix Column Layer

The MixCol layer is the main source of *diffusion* in AES. The layer is simply a matrix multiplication: Every 4 bytes are grouped into a matrix, and multiplied by a constant 4 by 4 matrix. This means that one bit flip anywhere in the input means that that all 4 bytes are affected–a very strong *diffusion*.

$$\begin{pmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \times \begin{pmatrix} B_0 \\ B_5 \\ B_{10} \\ B_{15} \end{pmatrix}$$

$$C_0 = 02 \times B_0 + 03 \times B_5 + 01 \times B_{10} + 01 \times B_{15}$$

Note: All operations are performed in $GF(2^8)$

4.2.4 Key Addition

- The subkeys are different in every round and are generated using Rijndael's key schedule.
- An XOR operation between the state and the subkey.
- irreversible without knowing secret key.

5 RSA Cryptosystem

5.1 Overview

RSA, the most well known *asymmetric* crypto-system, was invented in 1977 by Rivest, Shamir, and Adleman. RSA allows two parties to communicate through an insecure channel without having to establish a secret key through a different, secure, method of communication. However, due to its slow speed of operation, RSA is often used as a method of exchanging secret keys for quicker *symmetric* algorithms like AES and 3DES.

5.2 The RSA Algorithm

5.2.1 Key Generation

Unlike symmetric algorithms, the RSA protocol requires the computation of a key-pair $(K_{public}, K_{private})$:

1. Choose two primes (p, q) greater than 2^{512}.
2. Compute $n = p \times q$.
3. Compute Euler's totient function: $\phi(n) = (p-1) \times (q-1)$.
4. Choose $e \in \{1, 2, \ldots, \phi(n)-1\}$ so that greatest common denominator of e and $\phi(n)$ is 1.
5. Compute d, the multiplicative inverse of e, so that $e \times d \equiv 1 \pmod{\phi(n)}$. This is done using the Extended Euclidean Algorithm.

$$K_{public} = (n, e), K_{private} = (d)$$

5.2.2 RSA Encryption

After generating a set of keys and distributing the public key, a message (x) can be encrypted in the following way:

$$y \equiv x^e \bmod n$$

where $x \in \mathbb{Z}_n = \{0, 1, ..., n-1\}$ and y is the encrypted message

5.2.3 RSA Decryption

To decrypt the message (y) a private key (d) is required.

$$x \equiv y^d \bmod n$$

5.3 Proof of Correctness

In order to prove correctness of RSA, we need to prove that $x^{ed} = x \bmod n$
Given that $ed = 1 \bmod \phi(n)$:

$$x^{ed} = x^{1+k\phi(n)} = x(x^{\phi(n)})^k$$

Using Euler's Theorem: $a^{\phi(n)} \equiv 1 \bmod n$

$$x(x^{\phi(n)})^k \equiv x(1)^k \bmod n \equiv x \bmod n$$

5.4 Example

Alice wants to share hey secret key (4) with Bob. Bob generates an RSA key-pair which is used by Alice to securely deliver the message:

Alice:		Bob:
$x = 4$		1. $p = 3, q = 11$
		2. $n = 33$
		3. $\phi(n) = 20$
	$\xleftarrow{K_{pub} = (33, 3)}$	4. choose $e = 3$
		5. d = $e^{-1} \equiv 7 \bmod 20$
$y = 4^3 \equiv 31 \bmod 33$	$\xrightarrow{\text{y} = 31}$	
		$x = y^d = 31^7 \equiv 4 \bmod 33$

Using RSA, Alice was able to share her secret message (4) with Bob through an insecure channel.

Note: In order for the exchange to be truly secure, the p and q numbers need to be significantly larger. The small values are used for the sake of simplicity.

5.5 Security

The only known way of calculating the inverse of the public key, e, is by using the Extended Euclidean Algorithm. The algorithm, in turn, requires knowing $\phi(n)$, which is very difficult to compute without the prime factors of n: p and q. Assuming n is large enough($\geq 2^{1024}$), finding p and q by factoring would take hundreds of years even with today's strongest computers.

Important: if p and q are very close to $\sqrt{n}$ or p is significantly smaller than q (or vice versa) it is possible to factor n significantly faster. Many CTF problems involving RSA require you to exploit this mistake.

6 Diffie-Hellman Key Exchange

6.1 Overview

The Diffie-Hellman(D-H) Key exchange is a public-key protocol first published in 1976 by Whitfield Diffie and Martin Hellman. D-H is exclusively used to securely establish a secret key between two parties through an insecure channel.

Note: Due to the process of key generation, D-H can not be used to share secret messages directly, and instead relies on other algorithms once a key was established. This will be discussed later in the paper.

6.2 The Diffie-Hellman Problem (DHP)

Using a multiplicative cyclic group, for example $\mathbb{Z}_p^*$, where p is a prime and 0 is omitted from the group $\{1, 2, \ldots, p-1\}$, we can create a computationally difficult problem.

Given p, $\beta \in \mathbb{Z}_p^*$ as well as the primitive element α, find x so that:

$$\alpha^x \equiv \beta \bmod p$$

The only known way of solving this problem is by solving a difficult Discrete Logarithm Problem (DLP):

$$x \equiv log_\alpha \beta \bmod p$$

6.3 The Protocol

Given a multiplicative group $\mathbb{Z}_p^*$ with p, a known prime number, and α, a known primitive element of the group:

Alice:		Bob:
$a = K_{privateA} \in \mathbb{Z}_p^*$		$b = K_{privateB} \in \mathbb{Z}_p^*$
$A = \alpha^a \bmod p = K_{pubA}$		$B = \alpha^b \bmod p = K_{pubB}$
	$\xrightarrow{K_{pubA} = (A)}$	
	$\xleftarrow{K_{pubB} = (B)}$	
$K_{AB} \equiv B^a \bmod p$		$K_{AB} \equiv A^b \bmod p$

Figure 5: The D-H Key Exchange

After this procedure, both Alice and Bob end up with the secret key K_{AB}. They can now use the key in a symmetric algorithm (e.g. AES) in order to communicate securely.

6.4 Proof of Correctness

In the final step of the key exchange, Alice computes $B^a \bmod p$. B, the public key of Bob, is equivalent to α^b. Substituting it for B we get $K_{AB} = B^a \equiv \alpha^{ab} \bmod p$.
Bob completes a similar calculation except with Alice's public key A. So $K_{AB} = A^b \equiv \alpha^{ab} \bmod p$.
Since both parties arrive at the same result, this exchange can be deemed as functional.