CANADIAN THESES

THÈSES CANADIENNES

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30.

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30.

THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED

LA THÈSE A ÉTÉ
MICROFILMÉE TELLE QUE
NOUS L'AVONS REÇUE

DATA ENCRYPTION BASED ON THE LOGARITHM PROBLEM

by

John W. Jones

A thesis
presented to the University of Ottawa
in partial fulfillment of the
requirements of the degree of
Master of Applied Science
in
the Department of Electrical Engineering

ISBN 0-315-33330-8

UNIVERSITÉ D'OTTAWA
UNIVERSITY OF OTTAWA

The University of Ottawa requires the signatures of all persons using or photocopying this thesis. Please sign below, and give your address and the date.

TABLE OF CONTENTS

ABSTRACT

An analysis of the Public Key Distribution System based on the discrete logarithm problem is made, the underlying idea being to see whether a chosen key is unexpectedly weak. First some background on the development of cryptography is given, and then some relevant concepts in number theory are introduced. A brief summary of the previous research done on the logarithm problem is then given. Following this, sets are defined that will partition all possible remainders modulo p, and it is shown that this can sometimes lead to a determination of the private key. These keys should be avoided. Analysis is done and examples are given. Further analysis and examples are given centering around another observation and around an approach using the Chinese remainder theorem.

ACKNOWLEDGEMENTS

The author would like to express his deepest and sincerest appreciation to Professor Saligram G.S. Shiva for his constant advice, guidance, and patience.

The author also acknowledges the financial support of the Natural Sciences and Engineering Research Council of Canada.

TABLE OF NOTATION USED

The following notation is used in this thesis. These symbols are defined when they are first introduced in the thesis, and are summarized here for convenience. The equation or section number following the definition shows where the symbol is introduced.

C a set formed by repeatedly cubing ρ (3.34)

E_k the smallest number such that ρ divides $\rho_k^{E_k} - 1$ §3.7

k the logarithm of q^k mod p, which is the secret key of the Diffie-Hellman public key distribution system (3.1)

m the cardinality or maximum size of the set S, which is the smallest integer such that ν divides $2^m - 1$ §3.2

p the size of the prime field $GF(p)$, over which the logarithms are calculated (3.1)

p_i the prime factors of $p - 1$ (3.41)

q the primitive element or primitive root of p (2.13)

r a positive integer signifying the distance of the exponent k or $p - 1 - k$ or $\frac{p-1}{2} \pm k$ from the easily determined exponent $\frac{p-3}{4}$ §3.5

S the set formed by repeatedly squaring ρ; these sets partition the set of all possible remainders or public keys (3.7)

S' the set formed by repeatedly squaring ρ^{-1}; each element of S' is the multiplicative inverse of the corresponding element in S (3.8)

$\bar{S}$ $S \cup S'$ §3.2

S_k the set of the exponents of the remainders in the set S (3.10)

S'_k the set of the exponents of the remainders in the set S' (3.10)

β the logarithm of 2; $q^\beta \equiv 2$ §3.7

γ_i $q^{\frac{p-1}{i}}$ §3.8

μ m or a divisor of m

ν $\frac{p-1}{2}$; in this thesis we only consider ν odd §3.2

ρ the remainder q^k mod p, which is the public key of the Diffie-Hellman public key distribution system (3.1)

Ψ another set that partitions the set of all possible remainders or public keys (3.11)

Ψ_k the set of the exponents of the remainders in the set Ψ (3.12)

CHAPTER I

§1.1 Introduction

Cryptography is the art or science of making a text unintelligible to an unauthorized reader by some sort of transformation. The original text is referred to as the *plaintext*, and it can consist of any kind of information, such as English text, computer code, electronic banking data, or numeric codes representing Chinese characters. A cryptographic *cipher*, or algorithm, will be used, which will transform the plaintext into *ciphertext*, the particular transformation depending on the value of a secret *key*, known only to the intended receiver and sometimes to the sender of the cryptogram.

If a ciphertext message is somehow intercepted by a third party—the "enemy", then a *cryptanalyst* can attempt to "break" the cipher, rendering the text readable. So *cryptanalysis* is the art or science of solving the cryptogram without knowledge of the key, and sometimes without knowledge of the algorithm used.

Cryptography and cryptanalysis are the two branches of the science of *cryptology*, which embraces both the invention and the solution of ciphers.

A deep understanding of the science of cryptology is needed by communications and computer engineers because it is they who are responsible for designing and implementing telecommunications and computer-communications systems, and there is an ever-increasing need for these systems to be secure from eavesdropping and from third parties changing the message being sent, or masquerading as another party. There is also a need to protect confidential stored data from being copied or altered by unauthorized people. Cryptography is the best means available for such protection.

Private businesses naturally want to protect proprietary and financial information from their competitors; banking institutions must be able to securely carry out electronic funds transfers, including those done by individual customers using automated teller machines; and governments have to be able to protect the ever-increasing amounts of confidential information they have on their citizens, especially in countries where individuals have the legal right to privacy. All of this is in addition to the traditional needs of the military establishment and the diplomatic community. Whereas governments can develop a pool of highly expert cryptographers through their security agencies, private companies must rely on outside expertise to help them with their cryptographic needs. Thus there is a definite need for cryptologic research in the public domain by engineers and scientists, and for publicly available information on the relative security of different cryptographic schemes.

Cryptographic ciphers can be placed into two categories — those whose security relies on the secrecy of a key known to the two people communicating; and those for which the enciphering key is public knowledge, but the deciphering key is known only to the intended recipient of the message. The first category is referred to as conventional or *symmetric* encryption because the same secret key is known to both the encrypter and the decrypter. The weakness of this system is that the key has to somehow be communicated secretly. The second category is referred to as *asymmetric* or *public key* encryption because two different keys are utilized; a publicly known one for encryption and a secret one for decryption.

One scheme that can be used both as a public key encryption system and as a scheme to exchange private keys over an insecure channel is based on the discrete logarithm problem. The efficient solution of this problem would render both systems obsolete.

§1.2 Outline of Thesis

In the first chapter of this thesis we briefly examine the historical background of cryptographic ciphers. We then introduce some of the number theory needed to understand public key systems, especially the logarithm problem-based one that this thesis examines. Finally we introduce public key systems.

In Chapter II we define the distribution system based on the logarithm problem and give a summary of the previous research that has been carried out into it. We then define the sets S and Ψ that can sometimes help in the determination of the private key given the public one. We develop this idea and show some examples to illustrate how this may be done. We then give an assertion for determining the private key in a certain number of trials using another approach; if the number of trials needed is low, then these primes p should be avoided. Finally, we develop the observation made by Pohlig and Hellman [29] that the Chinese remainder theorem can be used to solve for the private key. This leads to the conclusion that $\frac{p-1}{2}$ should be prime.

We end the thesis with some Concluding Remarks.

Part of the work in this thesis was presented at the IEEE International Symposium on Information Theory held in Brighton, England, June 24-28, 1985.

CHAPTER II

§2.1 Review of the Historical Evolution of Cryptography

Conventional cryptographic ciphers can generally be classified into two broad categories, *transposition ciphers* and *substitution ciphers*. If a cipher uses a mixture of transposition and substitution, then the cipher is labelled a *product cipher*.

In a transposition cipher the plaintext symbols are rearranged into a different order; they are quite literally scrambled. The idea behind transposition ciphers can be simply illustrated by means of the earliest known cipher, which used a device called a *scytale* [27]. Used by the ancient Greeks as early as 400 BC, the scytale was a cylinder around which was wrapped a long narrow band of papyrus. With the papyrus wrapped around the cylinder, a message would be written out horizontally line by line. The papyrus could then be separated from the cylinder and given to a messenger. If the messenger happened to be intercepted by someone, he would not be able to decipher the message without knowing the diameter of the cylinder, or without trying cylinders of every diameter possible.

With the scytale cipher the original letters were merely placed in a different order on the papyrus strip; in other words they were transposed. The key that determined the transposition mapping was the diameter of the cylinder.

The other broad category of ciphers is the substitution cipher. With a substitution cipher the original plaintext symbols are substituted for different symbols, without changing their order. An example of an early substitution cipher is the *Cæsar cipher*, first used by the Roman emperor to communicate with Marcus Tullius Cicero around 50 BC, probably saving his life and those of his troops. Let the 26 letters of

the English alphabet be symbolized by $m_0, m_1, m_2, \ldots, m_{25}$. With a Cæsar cipher each letter m_i is substituted for $m_{i+k \bmod 26}$, where k can be thought of as being the key for the cipher. In the cipher supposedly used by Julius Cæsar the key was $k = 3$. This of course is a very simple cipher, since there are only 26 possible keys.

However, even if any mapping at all from the 26 plaintext letters to the 26 ciphertext letters is allowed, with there being 26! different mappings possible, this cipher would be easy to solve. This is because in English, or any other written language, each letter occurs with a different average frequency. In English, the most common letter is E which occurs about 13% of the time, while the least common letter is Z which occurs less than 0.1% of the time. Table 2.1 shows the frequency of letters in English [24].

Table 2.1

One-Gram Probability Distribution

Letter	p	Letter	p
A	0.0856	*N*	0.0707
B	0.0139	*O*	0.0797
C	0.0279	*P*	0.0199
D	0.0378	*Q*	0.0012
E	0.1304	*R*	0.0677
F	0.0289	*S*	0.0607
G	0.0199	*T*	0.1045
H	0.0528	*U*	0.0249
I	0.0627	*V*	0.0092
J	0.0013	*W*	0.0149
K	0.0042	*X*	0.0017
L	0.0339	*Y*	0.0199
M	0.0249	*Z*	0.0008

So, given a ciphertext formed by this Cæsar substitution, one need only determine the frequency of occurence of each ciphertext letter to provide a valuable insight

into what each letter is.

Not only does each letter occur with different frequency, but also different pairs of letters occur together with different frequencies. For example, the two-gram TH occurs about 3% of the time (occurring in common words such as *the*, *this* and *there*), whereas the two-gram YR occurs only about 0.01% of the time. Given a ciphertext, the frequency of different two-grams can be compared to that of the English language in general to help decipher the text.

One can also utilize the relative frequencies of three-grams, model the plaintext as being generated by a Markovian process, or search for a particular word or phrase that one suspects could be occurring frequently in a given ciphertext. Also, if the traffic being deciphered is not standard English text, for instance if it is computer program source code or data on bank account transactions, then probability distribution tables for typical samples of this traffic can be compiled.

Even though these ciphers existed so long ago, they were used extremely rarely at that time in history, and the art of cryptanalysis did not at all exist, which meant that these ciphers were stronger than they now appear. It wasn't until the 14th century, when the Islamic civilization was bearing its finest fruits, that cryptanalysis came into being. At that time an Islamic scholar named Táj ad-Dín 'Alí ibn ad-Duraihim ben Muḥammad aṯ-Ṯa'álibí al-Mausilí described a number of ciphers and techniques to cryptanalyse them based on the frequency and distribution of letters in the Arabic alphabet. This was the first time that both substitution and transposition ciphers were considered together [20].

The Cæsar cipher described above is a *one*-gram substitution cipher, meaning that each letter is substituted for another one independent of the letters around it. As was shown, since individual letters differ appreciably in their probability of occurence, these letter frequencies can be used to help determine the mapping between the plaintext and ciphertext. However, if blocks of n letters are enciphered one block at a time, with each ciphertext letter depending on all n plaintext letters, then the

letter frequencies of Table 2.1 are not of as much help. The probability distributions of n-grams can be used, $n \geq 2$, but the probabilities of individual n-grams are far less unique than those of one-grams, so this does not help as much either. But these substitution ciphers are still easy to break. Famous examples of n-gram substitution ciphers are the two-gram *Playfair cipher* and the n-gram *Hill cipher*.

To prepare a text to be enciphered using Playfair encipherment, you first divide it into two-grams. Say the plaintext is

SEND SUPPLIES TOMORROW.

Dividing it into two-grams gives

SE ND SU PP LI ES TO MO RR OW.

Where there are two-grams consisting of the same letter, such as the PP above, a Q is inserted. This gives

SE ND SU PQ PL IE ST OM OR RO W.

Note that since adding the Q between the P's split up the R's so that they are now in separate two-grams, no Q need be inserted there. But we have an odd number of letters, leaving a single W at the end. This is padded with a Q to get

SE ND SU PQ PL IE ST OM OR RO WQ.

The plaintext is now ready for encipherment.

The key to this cipher is a word of any length, with no letters occurring more than once. A Playfair square of 25 letters (I and J are treated as the same letter) is then drawn up row by row, with the keyword first, followed by the rest of the alphabet. Say the keyword is CIPHER. This gives the following square:

$$S_{\text{Playfair}} = \begin{bmatrix} C & I & P & H & E \\ R & A & B & D & F \\ G & K & L & M & N \\ O & Q & S & T & U \\ V & W & X & Y & Z \end{bmatrix}$$

The plaintext is enciphered two-gram by two-gram, using the following set of rules.

— If the letters of the plaintext two-gram are in the same row but in different columns, then the two-gram is enciphered into a ciphertext two-gram consisting of the letters in the same row and in the column to the immediate right. (The lefthandmost column can be thought of as being to the right of the righthandmost column.)

— If the plaintext letters are in the same column but in different rows then they are enciphered into a ciphertext two-gram consisting of the letters in the same column and the row directly below. (The top row can be thought of as being below the bottom row.)

— If the plaintext letters are in different rows and different columns then they can be thought of as forming opposite corners of a square. The ciphertext two-gram letters will be the ones in the other two corners of the square, with the individual plaintext and ciphertext letters being in the same row.

Using these rules, our plaintext message becomes

UP MF TO IS BS PC TU TG VG GV IW.

Note that a square which is a cyclic rotation of columns and rows of S_{Playfair} will give the same encryption. An example of a square equivalent to S_{Playfair} above is:

$$S_{\text{equiv}} = \begin{bmatrix} T & U & O & Q & S \\ Y & Z & V & W & X \\ H & E & C & I & P \\ D & F & R & A & B \\ M & N & G & K & L \end{bmatrix}$$

The weakness of Playfair encipherment lies in the structural constraints set by the rules of encipherment. For example, if a plaintext two-gram p_1p_2 maps to c_1c_2, then p_2p_1 must map to c_2c_1. One can take an encrypted text and, knowing the rules of encipherment, determine what some of the entries in the Playfair square are, and find the corresponding plaintext.

With a long text we can also use our knowledge of the relative frequencies of different two-grams in order to determine some of the entries. For example, the most frequently occurring two-gram in English is TH, which occurs about 3% of the time, while its reversal HT occurs only about 0.1% of the time. If a certain ciphertext two-gram also occurs very frequently, but its reversal occurs much less often, then it could correspond to TH.

The n-gram Hill cipher is also a substitution cipher. We will show an example of a 3-gram Hill cipher. The 26 letters of the alphabet are represented during encipherment by the integer 0 to 25. The plaintext is divided into 3-grams which are then converted into their numerical representations. If the 3-gram is thought of as a 3 element array, then it can be multiplied by a 3×3 matrix T to give another 3 element array, with all the operations performed modulo 26. This new 3 element array is the numerical representation of the ciphertext.

In order to be able to decipher the ciphertext into a unique plaintext, we need T to be an invertible linear transformation. T is invertible iff $\det(T)$ is not divisible by p for any prime p that divides m, where m is the number of plaintext symbols (ie, the modulus for our arithmetic). In our case we are using $m = 26$.

Let the plaintext be

NO CIPHER IS UNBREAKABLE

which is true except for the one-time cipher described later on. We divide this into 3-grams

NOC IPH ERI SUN BRE AKA BLE.

Say the substitution matrix is

$$T = \begin{bmatrix} 14 & 19 & 7 \\ 1 & 8 & 17 \\ 9 & 24 & 10 \end{bmatrix}.$$

Then $\det(T) = -2211$, which is divisible by neither 2 nor 13. So the matrix is invertible.

NOC has numerical representation $(13, 14, 2)$.

$$T(13, 14, 2) = (20, 3, 5)$$

which corresponds to the letters UDF. If we continue in this manner we get the ciphertext

UDF ENI TQE VJS BXP TIY RBB.

When a small sample of plaintext and corresponding ciphertext is available the substitution matrix T is extremely easy to determine using Gaussian elimination. In the above example if both plaintext and ciphertext are known then we can form the following equations modulo 26 in order to solve for the top row of coefficients of T:

$$\begin{array}{rcrcrcl} 13a_{11} & + & 14a_{21} & + & 2a_{31} & = & 20 \\ 8a_{11} & + & 15a_{21} & + & 7a_{31} & = & 4 \\ 4a_{11} & + & 17a_{21} & + & 8a_{31} & = & 19 \end{array}$$

$$\begin{array}{rl} & a_{11} = 14 \\ \Longrightarrow & a_{21} = 19 \\ & a_{31} = 7 \end{array}$$

Another type of substitution cipher is *Vigenère encipherment*. We will illustrate how it works by means of a simple example. We will use the integer representation of the alphabet that we used for Hill encipherment.

A key of finite length is chosen. Let our key be TERVEISET which in integer representation is $(19, 4, 17, 21, 4, 8, 18, 4, 19)$, which is $r = 9$ letters long. We take our plaintext, which is "the logarithm problem is used to form a public key distribution system", and divide it into 9-grams:

THELOGARI	→	19	7	4	11	14	6	0	17	8
THMPROBLE	→	19	7	12	15	17	14	1	11	4
MISUSEDTO	→	12	8	18	20	18	4	3	19	14
FORMAPUBL	→	5	14	17	12	0	15	20	1	11
ICKEYDIST	→	8	2	10	4	24	3	8	18	19
RIBUTIONS	→	17	8	1	20	19	8	14	13	18
YSTEM	→	24	18	19	4	12				

The key numbers are added to the plaintext numbers modulo 26 to form the ciphertext. The first key number is added to the first plaintext number, the second key number to the second plaintext number, and so forth.

For the first block of text:

key	TERVEISET	→	19	4	17	21	4	8	18	4	19
PT	THELOGARI	→	19	7	4	11	14	6	0	17	8
			12	11	21	6	18	14	18	21	1
			M	L	V	G	S	O	S	V	B

We continue in this manner to get a ciphertext that looks like:

MLVGSOSVB MLDKVWTPX FMJPWMVXH

YSIHEXMFE BGBZCLAWM KMSPXQGRL

RWKZQ

For a cryptanalyst to decipher a text encrypted using the Vigenère cipher, he must first determine the period r of the key, and then determine what the key is. He can use the *phi-test* to help determine the period of the key. The phi-value of a ciphertext is defined as being

$$\phi(y) = \sum_{0 \le t < m} N_t(y)[N_t(y) - 1] \tag{2.1}$$

where $N_t(y)$ is the number of occurrences of the letter t in the ciphertext y. $\phi(y)$ can be compared to reference values for different lengths r for typical text in order to

determine what r is. The reason why $\phi(y)$ is correlated with r is that as r increases, the distribution of different letters becomes more uniform.

If $r = 1$, which means Cæsar substitution, we know that the probability of each letter will be as in Table 2.1, even though each letter will be substituted for a different one. But if $r = 2$ then the distribution will be a bit more uniform. Say the first key symbol maps E to K and the second key symbol maps U to K. Then the average probability of a K occurring in the ciphertext would be about

$$\frac{P(\mathrm{E}) + P(\mathrm{U})}{2} = \frac{0.1304 + 0.0249}{2} = 0.07765$$

As r increases the probability of each ciphertext letter approaches $\frac{1}{26} \approx 0.03846$. The phi-test is a measure of the unevenness of this distribution, and so is a measure of r.

Once the period r is determined, the cryptanalyst can examine every rth letter in the ciphertext, all of which will be enciphered using the same rule, in order to determine the corresponding plaintext.

A special case of the Vigenère system is the *Vernam system*, invented in 1917 by Gilbert S. Vernam of AT&T. The key here is a binary one and is recorded on a paper tape. The 26 letters and 6 special characters are coded into 5-bit codewords using a code called the *Baudot code*, and then added bit by bit to the binary key.

For the Vigenère system of ciphers the enciphering used need not be the Cæsar cipher. Any rule that substitutes one symbol for another using a key will do.

A cipher that employs both transposition and substitution is called a *product cipher*. Such a hybrid cipher will in general be stronger than one that uses transposition or substitution alone.

An early example of a product cipher is the *ADFGVX cipher* used by the Germans in World War I. [27] It utilizes a 6 row by 6 column table containing 36 symbols: the letters A to Z and the digits 0 to 9. Both the columns and the rows are labelled by the letters A, D, F, G, V and X. The coordinates of a symbol are given by a pair of

these letters, with the row coordinate coming before the column coordinate. This pair forms an intermediary between the plaintext and the ciphertext. The intermediate text is then written down row by row, each row containing 7 letters. The letters are then read column by column, with a key giving the order in which the 7 columns are read. The key is a 7 letter word, and the letters in alphabetical order give the order in which columns are to be read.

We will show this with an example. Let our plaintext be "for centuries, cryptography has been a valuable asset of the military and diplomatic communities" [14].

The cipher table looks like:

	A	D	F	G	V	X
A	K	Z	W	R	1	F
D	9	B	6	C	L	5
F	Q	7	J	P	G	X
G	E	V	Y	3	A	N
V	8	0	D	H	O	2
X	U	4	I	S	T	M

The intermediate text will be:

```
AX VD AG DG GA GX XV XA AG XF GA XG DG AG GF FG
XV VD FV AG GV FG VG GF VG GV XG DD GA GA GX GV
GD GV DV XA GV DD DV GA GV XG XG GA XV VD AX XV
VG GA XX XF DV XF XV GV AG GF GV GX VF VF XF FG
DV VD XX GV XV XF DG DG VD XX XX XA GX XF XV XF
GA XG
```

Say our keyword is CIPHERS → 1 4 5 3 2 6 7. Then the ciphertext message will be:

```
AGXA GDFG GVXV XDGV AXFX DXXA
AXXG XGGG GVDV XVXV GFVV VAXD
GGDG AGXA GVGA XXXF VVXG XVXG
AXFF GGAG AGGA AXGV GGGX FXVA
AGFV VVGD GAGX XFGF DVDX XGGX
FAVG FDXD DXVV FGVX DXDG FDVG
GVVV DGVD GVGD VGFX FXXG
```

Being over 70 years old and thus workable by pencil and paper, this cipher is also very easy to solve. The only thing unknown is the order in which the columns are

selected. For instance, the ciphertext can be placed into 7 columns by the cryptanalyst and the letters for some common words searched for.

For example, an extremely common word in English is THE, which gives the intermediate letters XVVGGA. If we look for one or two rows that contain all these letters, with the first letter(s) being in the upper row and the rest in the row below it, and with no two letters in the same column, we find only two possibilities. The first is in rows 14 and 15, and the second is in rows 16 and 17. The first occurrence gives the following possible partial keywords, or cyclic shifts of partial keywords, with the missing number being anywheres:

3 2 6 7 1 4
3 6 2 7 1 4
5 2 6 7 1 4
5 6 2 7 1 4

Only the first possibility gives a readable text.

There are also other attacks that can be made on the ciphertext. Even if we use the unimaginative approach of key exhaustion, there are only $7! = 5040$ possible keys.

An excellent discussion of ciphers up to this point in history from an information theoretic background is given in Claude Shannon's landmark paper [31].

One class of product ciphers is ciphers generated by a *rotor system*. A rotor system consists of a series of wired rotors or codewheels, each of which implements a monoalphabetic substitution. The plaintext letter and ciphertext letter are connected together by an electric circuit. By rotating the individual rotors this mapping is changed.

The most famous example of a rotor system is the German *Enigma machine*. It was the main form of encipherment used by the Germans in World War II, and its reconstruction and analysis by Polish and British cryptographers helped the Allies immeasurably during the war. The Enigma machine had four rotors, three of which

could be rotated. An excellent account of how the Enigma cipher was broken is given in [25].

Other rotor systems used at that time included the Japanese Red and Purple machines [26], and the American Sigaba. The Japanese machines did not have Enigma-type wheels as rotors, instead having telephone selectors/wipers, and mechanically they worked differently from Enigma, but cryptographically they performed a similar function. The Purple machine, first built in 1937 and replacing the Red machine starting in 1939, had 25 different steps of substitution matrix encipherment. Both machines partitioned the alphabet into the 20 consonants and 6 vowels, with the vowels being enciphered only amongst themselves! This was a gift for the American cryptographers.

Part of the reason why the German and Japanese cipher systems were so quickly solved by the Allies was because of the introduction of electronic computers. By the 1960s computers were not only a powerful cryptanalytic tool, they had also become a source of new volumes of digital data. This data included both data held by businesses and personal data on individuals held by governments and offices like credit bureaus. Thus the need arose for a much stronger method of encryption. IBM started doing research to develop a strong product cipher in 1968, which around 1970 resulted in a system called LUCIFER [27]. When the National Bureau of Standards (NBS) in the United States solicited algorithms for a Data Encryption Standard (DES), which would serve as a Federal Information Processing Standard (FIPS), in 1973, IBM further developed the LUCIFER design and submitted it to the NBS. It was approved by the NBS, and became their Data Encryption Standard in 1977 [10].

The reason why a Data Encryption Standard was needed was so that U.S. government departments and agencies, as well as private corporations, would have a standard method of encryption that had been analysed and certified by the NBS and the National Security Agency (NSA). Whereas in the past cryptography was of interest to mainly the military and diplomatic corps, it was now of interest to everyone.

Individuals have the right to privacy, and with there being so many data banks of information on individuals, these must be protected from unauthorized use if this right is to be maintained. Also, with the rapid increased use of electronic banking, electronic crime is becoming a serious problem, and it is in everyone's interest to prevent it. Thus a strong, publicly known, government certified method of encryption was a necessity, in everyone's interest. Before the DES was published there were no encryption schemes that carried a government "guarantee". A commercial user of encryption equipment usually had no means of accertaining the cryptographic strength of the equipment he was buying, because most expertise was with the military and diplomatic communities. The marketplace was truly one where the warning caveat *emptor* stood. It was into this field that the US government stepped with its Data Encryption Standard. Since that time DES has indeed become a standard method of encryption, with a variety of products implementing it on the market.

The Data Encryption Standard is a product cipher consisting of 16 rounds of product encipherment. The data being enciphered is split into individual blocks of 64 bits each, and each block is put through these rounds of encipherment. The key used for the enciphering is a 64-bit key, k, of which 56 bits are used for the derived key for each round and 8 bits are used as parity check bits for error detection. For the details of DES encipherment, see [10]. For a good introduction on the background to DES and its intended purposes, see [11] and [28].

The result of the DES encipherment is an output block which to the best of our knowledge cannot be used to directly determine the input block without trying every possible key. The only known attack on DES that can possibly work is the brute force one of exhaustively testing for every possible key. Since the DES algorithm is a publicly known standard this attack will work, but would require a considerable amount of resources at present. Because the key is in effect 56 bits long, $2^{56} \approx 7.206 \times 10^{16}$ different keys are possible. Diffie and Hellman [14] in 1977 pointed

out that by using many DES chips in parallel one can reduce the time required to check every possible key in a known ciphertext and corresponding plaintext attack. They felt that by the time the DES standard comes into widespread use, meaning that inexpensive LSI chip implementations would be available, it would be feasible to mount such an attack. Since at this time DES appears to be immune from other forms of attack, they suggested that independent keys be used for each round of encipherment. This would make an exhaustive attack prohibitively expensive, but would not conform to the DES standard.

Ever since Diffie and Hellman started criticizing the DES algorithm, the NSA has insisted that they were being far too pessimistic and that DES is secure [33], until last year when Walter Deeley, deputy director for communications security at the NSA, announced that he is not going to reapprove DES for use as a standard in 1988 [23]. Instead the NSA will introduce new ciphers for use by both government and private industry. These new ciphers will be secret; the NSA will not divulge how they work, and the chips implementing the ciphers will be designed to prevent people from determining the algorithms from the chips themselves. Also, the NSA will not guarantee the security of these algorithms unless the Agency supplies the keys to them, or supplies instructions for creating these keys. The reason for this is that some keys could prove to be weak ones, but this will result in the NSA's being able to know what keys each customer uses, enabling them to easily decipher any communications using these ciphers.

After having reviewed all these cryptographic techniques, and having found weaknesses in all of them, the question arises as to whether a perfectly secret cipher can exist. The answer is yes. The only cipher known that is perfectly immune from cryptanalysis (if properly used) is the *one-time pad system*. With this cipher you add to your text a series of completely random (ie, independently and uniformly distributed) numbers and transmit this. If the receiver of the message has received through secure

channels (such as a courier) a copy of these random numbers then he can reconstruct the original message.

For example, if we use the ASCII representation of alphanumeric text, then the plaintext

ATTACK AT DAWN

is represented as

C1 D4 D4 C1 C3 CB C1 D4 C4 C1 D7 CE.

If the series of random hexadecimal numbers shared by the sender and receiver is

597A 6455 E5BE 0138 449A E24B

and the two streams are added digit by digit modulo 16, then the ciphertext is

1A4E 3816 A879 C20C 085B B909.

Characteristics of ASCII code, such as the fact that pure alphabetic text will always start with hexadecimal C or D, and characteristics of the language used, are completely masked by the random key. There is absolutely nothing for the cryptanalyst to go on.

But despite this one-time ciphers have been decrypted. Why is this possible? Because of the logistical problem of getting to the man in the field that random key which is at least as long as the message transmitted and which can be used only once. What was done instead sometimes was to issue books or pads of these random numbers and to reuse them. Typically, a message would begin with the page number and location on the page of where to begin using the pad. After awhile a number of messages using the same page of the random number pad would be intercepted. And once the cryptanalyst had more than one ciphertext that used the same random numbers he could start his work.

The one-time pad system is an excellent system for extremely short and important messages between two people who use the pad only once. But for high volume traffic it is totally inadequate.

§2.2 Some Results from Number Theory

Before going on to the next part of this introductory chapter, which describes public key distribution systems in general and the Diffie-Hellman scheme in particular, we briefly review some number theoretic concepts to give a background for this material [3].

In "normal" mathematics we define operations such as addition, multiplication, division and exponentiation over the set of real numbers, or perhaps over the set of integers, both of which are infinite sets. We can also define these operations over a finite set of integers. We can define the set of numbers

$$\{0, 1, 2, 3, 4, \ldots, n-1\} \tag{2.2}$$

as being a set in which closure exists under addition and multiplication, provided we perform these operations modulo n.

What we mean by the term "modulo n" is that if n divides the difference $a - b$, then

$$a = b \text{ modulo } n \tag{2.3}$$

This is often abbreviated as $a = b \bmod n$, or $a \equiv b$, where the "modulo" is understood, or has been previously stated.

To give an example, say $n = 14$. Then the field of 14 integers consists of

$$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13\}.$$

If we want to multiply the two numbers 3 and 7, we get $(3)(7) \equiv 21 \equiv 21 - 14 \equiv 7$ (modulo 14).

Say we want to find 12^{11}. If we were working in the set of positive integers N then this would be

$$12^{11} = 743\ 008\ 370\ 688$$

which is a large number relative to $n = 14$. But, modulo 14, we have

$$12^{11} \equiv 743\ 008\ 370\ 688 - 14(53\ 072\ 026\ 477)$$

$$\equiv 10$$

Therefore $12^{11} = 10$ modulo 14.

We can avoid using large numbers in our computations by taking the modulus of the intermediate results:

$$12^{11} \equiv ((12^2)^2)^2 12^3 \bmod 14$$

$$\equiv ((144)^2)^2 12^3$$

$$\equiv ((4)^2)^2 12^3$$

$$\equiv (16)^2 (1728)$$

$$\equiv (4)(6)$$

$$\equiv 10$$

10 is referred to as the *residue* or *remainder* of 12^{11}.

With $n = 14$ we have a notable difference between arithmetic in an arbitrary finite set and regular arithmetic in $\mathbf{N}$, namely that products and quotients are not unique. For instance, $(10)(4) \bmod 14 \equiv 12 = (3)(4)$.

$$\text{Therefore } (10)(4) \equiv (3)(4)$$

In order to avoid this sort of problem, we must carry out our computations in a *finite field*, which can also be called a *Galois field*. All finite fields must have the following properties: [8]

1. There exist two operations that can be used to combine elements, namely multiplication and addition.

2. When two elements are multiplied or added the result must be an element that is in the field.
3. The field must contain the multiplicative identity element 1 and the additive identity element 0. So for any element a, $a + 0 = a$ and $a \cdot 1 = a$.
4. For every element a there exists an additive inverse element $-a$ and a multiplicative inverse element a^{-1}, so that $a+(-a) \equiv 0$ and $a \cdot a^{-1} \equiv 1$. Note that the multiplicative inverse of the additive identity element need not exist.
5. The associative (2.4),(2.5), commutative (2.6),(2.7) and distributive (2.8) laws apply. This means, modulo n, that

$$a + (b + c) \equiv (a + b) + c \tag{2.4}$$

$$a \cdot (b \cdot c) \equiv (a \cdot b \cdot c) \tag{2.5}$$

$$a + b \equiv b + a \tag{2.6}$$

$$a \cdot b \equiv b \cdot a \tag{2.7}$$

$$a \cdot (b + c) \equiv a \cdot b + a \cdot c \tag{2.8}$$

where a, b, and c are elements of the field.

When $n = 14$ we do not fulfill property 4 for an inverse multiplicative element. Say $a = 2$. Then we want an a^{-1} such that $2a^{-1} \equiv 1$, i.e., $2a^{-1} = 14M + 1$, for some integer M.

$$a^{-1} = \frac{14M + 1}{2} = \text{integer}$$

Since the numerator is always odd and the denominator is even, such an a^{-1} does not exist.

So we see that these finite fields do not exist for all sizes n of the field. In general they exist iff n is a prime number or a power of a prime number. When n is prime

the field is called a *prime field* and when n is a power of a prime the field is called an *extension field* over a prime field.

In this thesis we work with prime fields, so let $n = p$ indicate that the field is prime. We can designate this finite or Galois field of integers Z_p using the notation $GF(p)$, where

$$GF(p) = \{0, 1, 2, 3, 4, 5, \ldots, p-1\} \tag{2.9}$$

which we do for the remainder of the thesis. With this prime field the multiplication and addition performed are ordinary multiplication and addition carried out modulo p.

When working with an extension field, which can be symbolized by $GF(p^m)$ where m is an integer ≥ 2, we cannot easily use simple integers to carry out mathematical operations in the field. The field elements are best denoted by all possible polynomials of degree $m-1$, where the polynomial coefficients are from the prime field $GF(p)$. We will not show the details of this here since it is not needed for the thesis.

Let (x, y) symbolize the greatest common divisor of x and y. Given an element a in $GF(n)$ where $(a, n) = 1$, we have the following important theorem:

Euler-Fermat theorem:

$$a^{\phi(n)} \equiv 1 \bmod n \tag{2.10}$$

where $\phi(n)$, the Euler totient function, signifies the number of integers less than n that are relatively prime to n. When we consider a prime field $GF(p)$, this theorem can be simplified. $(a, p) = 1$ always for p prime, and $\phi(p) = p - 1$.

Euler-Fermat theorem for prime fields:

$$a^{p-1} \equiv 1 \bmod p \tag{2.11}$$

This is a variation of the Little Fermat theorem, which states that for any integer a and prime p we have $a^p \equiv a$ modulo p.

If the factors of n are known, then the Euler-totient function can be easily calculated. Letting $n = p_1^{e_1} p_2^{e_2} \cdots p_r^{e_r}$, where p_i are primes,

$$\phi(n) = \prod_{i=1}^{r} (p_i - 1) p_i^{e_i - 1} \tag{2.12}$$

However, if n is large and its factors are not known, then calculating $\phi(n)$ is difficult.

We can define the *exponent* of a modulo p as being the smallest integer e such that $a^e \equiv 1$ modulo p. From the Euler-Fermat theorem we know that $e \leq p - 1$. If $e = p - 1$ then a is called a *primitive element* in $GF(p)$ or a *primitive root* of p.

Primitive roots are very important because if an element q is primitive then the numbers

$$q, q^2, q^3, q^4, q^5, \ldots, q^{p-2}, q^{p-1} \tag{2.13}$$

modulo p are all distinct, taking on the values from 1 to $p - 1$. In other words, with respect to a given q, each of these remainders has a unique exponent and each exponent has a unique remainder; there is a one-to-one mapping between the exponents and the remainders. This is an essential property for the public key distribution system described in the next section.

For a prime p the number of primitive elements in $GF(p)$ is given by $\phi(p - 1)$. If one primitive element q is known, then the rest of the primitive elements are given by q^m where m are all the positive integers $\leq p - 1$ such that $(m, p - 1) = 1$. In other words, they consist of the following set:

$$\{q^m : \ 1 \leq m \leq p - 1 \text{ and } (m, p - 1) = 1\} \tag{2.14}$$

Clearly there are $\phi(p - 1)$ of them.

One property of primitive roots is that

$$q^{\frac{p-1}{2}} \equiv -1 \bmod p \tag{2.15}$$

always. The proof of this is simple. The Euler-Fermat theorem tells us that $q^{p-1} \equiv 1$. So $q^{\frac{p-1}{2}} \equiv (q^{p-1})^{1/2} \equiv (1)^{1/2}$. Since q is primitive, all the powers of q are distinct. This would lead one to guess that $(1)^{1/2} \equiv -1$, since we already have $1 \equiv q^{p-1}$. It is clearly true that $(-1)^2 \equiv (p-1)^2 \equiv p^2 - 2p + 1 \equiv 1$. Since each exponent maps to exactly one remainder and each remainder maps to exactly one exponent, we must have $q^{\frac{p-1}{2}} \equiv -1$.

In this thesis we will need to know how to find the primitive roots of large primes. Set (2.14) can be used once at least one primitive element has been found, but we should be able to find that original primitive root in an efficient manner, or be able to check to see if a certain element is primitive. Let us consider how to do this.

It is generally known that the order of any element modulo p must be a divisor of $p-1$ [3]. So we could factor $p-1$ into its prime factors and check all product combinations of these factors to see if any of them give a remainder of unity. If only $p-1$ gives the unity remainder then q must be primitive. This method takes at most $2^f - 2$ trials, where f is the number of prime factors of $p-1$.

However, we have found a way of shortening this method so that it takes up to f trials only. The shortcut centres around the following result.

Theorem: Given $\frac{p-1}{2} = \nu = p_1 p_2 p_3 ... p_u$, where p_i are the prime factors of ν, if $q^{\nu/p_i} \not\equiv -1$ for all p_i, then $q^f \not\equiv -1$, where f is any factor of ν. This means that q is primitive.

Proof: q^f can be equivalently expressed as $q^{\nu/f}$, where f is any factor of ν. From the discussion above, we know that if q is primitive then

$$q^{\nu} \equiv -1 \tag{2.16}$$

$$q^{\nu/p_i} \not\equiv -1 \text{ for all } p_i \tag{2.17}$$

$$(q^{\frac{\nu}{p_i}})^{\frac{1}{f'}} \not\equiv (-1)^{\frac{1}{f'}} \not\equiv (-1) \tag{2.18}$$

since all f' are odd, where f' is any factor not containing p_i.

Let $f = f' p_i$, so that f can be any factor of ν. Then (2.18) can be expressed as

$$q^{\frac{\nu}{f}} \not\equiv -1 \text{ for all } f. \tag{2.19}$$

Equivalently, $q^f \not\equiv -1$ for all f.

□

This statement leads naturally to the following simple procedure to find a primitive element.

Let α be our potential primitive element. Check to see if $\alpha^\nu \equiv -1$. If not, then try another α. Calculate $\alpha^{\frac{\nu}{p_i}}$ for all p_i. If none are congruent to -1, then α is primitive. If any are congruent to -1, then α is not primitive, and another α must be tried.

Example 2.1: Say $p = 2147483647$. Then

$$\nu = 1073741823 = (3)(3)(7)(11)(31)(151)(331)$$

Try $\alpha = 2$. Then $2^\nu \equiv 1$.

Try $\alpha = 3$. Then $3^\nu \equiv -1$ but $3^{\nu/3} \equiv -1$ also.

Try $\alpha = 7$.

$$7^\nu \equiv -1$$

$$7^{\nu/3} \equiv 1513477736 \not\equiv -1$$

$$7^{\nu/7} \equiv 610312904 \not\equiv -1$$

$$7^{\nu/11} \equiv 1327797538 \not\equiv -1$$

$$7^{\nu/31} \equiv 2146435071 \not\equiv -1$$

$$7^{\nu/151} \equiv 21925355 \not\equiv -1$$

$$7^{\nu/331} \equiv 164185298 \not\equiv -1$$

Therefore 7 is a primitive element.

□

The above method is highly suitable if we are only looking for one primitive element. But if, for some reason, a number of primitive elements are needed then we

only need to calculate powers of elements for prime elements. This is because of the obvious fact that given an exponent e and some elements α_i,

$$\alpha_1^e \alpha_2^e \dots \alpha_i^e \equiv (\alpha_1 \alpha_2 \dots \alpha_i)^e \bmod p. \tag{2.20}$$

If we determine all of the prime primitive elements using the above method, then the composite primitive elements can be determined using (2.20) by factoring the element into its prime factors. This is a shorter process, since it does not require any exponentiation.

Example 2.2: For $p = 1823$, $\nu = 911$ which is prime. Thus an element α is primitive iff $\alpha^\nu \equiv -1$. Let us assume that we know what all of the small prime primitive elements are.

Say $\alpha = 1365 = (3)(5)(7)(13)$.

$$\alpha^\nu \equiv 3^\nu 5^\nu 7^\nu 13^\nu$$

$$\equiv (1)(-1)(1)(1) \equiv -1$$

Therefore $\alpha = 1365$ is primitive.

Say $\alpha = 465 = (3)(5)(31)$.

$$\alpha^\nu \equiv (1)(-1)(-1) \equiv 1$$

Therefore $\alpha = 465$ is not primitive.

□

Also note that a perfect square can never be a primitive element since

$$(\alpha^\nu)^2 \equiv (\pm 1)^2 \not\equiv -1.$$

If q is primitive then for some element a in $GF(p)$ there is a unique integer k where $0 \leq k \leq p-2$ such that

$$a \equiv q^k \bmod p. \tag{2.21}$$

In number theory this integer k is called the *index* of a to the base q modulo p. Indices have properties that make them very similar to logarithms. Because of this they are often referred to as logarithms, which is what they are called in this thesis. Given (2.21) we can write

$$k = \text{ind}_q a \tag{2.22a}$$

or

$$k = \log_q a \tag{2.22b}$$

Letting $a_1 \equiv q^{k_1}$ and $a_2 \equiv q^{k_2}$ modulo p, we have the following properties modulo $p-1$:

$$\log(a_1 a_2) \equiv \log a_1 + \log a_2 \tag{2.23}$$

$$\log a_1^n \equiv n \log a_1 \quad \text{for } n \geq 1 \tag{2.24}$$

$$\log 1 \equiv 0 \text{ and } \log q \equiv 1 \tag{2.25}$$

$$\log(-1) \equiv \frac{p-1}{2} \tag{2.26}$$

If q' is also a primitive root of p then we have that

$$\log_{q'} a \equiv \log_q a \ \log_{q'} q \mod (p-1). \tag{2.27}$$

The following example illustrates some of these ideas.

Example 2.3: Let $p = 43$. Then

$$\frac{p-1}{2} = \nu = 21 = (3)(7).$$

$q = 3$ is a primitive root. This is so because

$$3^\nu \equiv 3^{21} \equiv -1 \mod 43$$

$$3^{\nu/3} \equiv 3^7 \equiv 37 \not\equiv -1$$

$$3^{\nu/7} \equiv 3^3 \equiv 27 \not\equiv -1$$

For small primes such as $p = 43$ the above exponentiation is easy to perform, but for large primes it is helpful to first calculate the remainders of q^{2^r}. We do this by first repeatedly squaring the primitive element modulo p up to the highest exponent less than p. In this case we have

$$3^2 \equiv 9$$

$$3^4 \equiv 38$$

$$3^8 \equiv 25$$

$$3^{16} \equiv 23$$

$$3^{32} \equiv 13$$

To determine 3^{21} we calculate

$$3^{21} \equiv 3^{16}3^43 \mod 43$$

$$\equiv (23)(38)(3)$$

$$\equiv -1$$

Since $q = 3$ is primitive, the numbers $q, q^2, q^3, \ldots, q^{p-1}$ will partition $GF(p)$. In this case (2.13) becomes

3,	9,	27,	38,	28,	41,	37,	25,	32,	10,	30,	4,	12,	36,
22,	23,	26,	35,	19,	14,	42,	40,	34,	16,	5,	15,	2,	6,
18,	11,	33,	13,	39,	31,	7,	21,	20,	17,	8,	24,	29,	1

If we want to find a second primitive root we can use (2.14). One positive integer m such that $(m, 42) = 1$ is $m = 11$.

$$q^{11} \equiv 3^{11} \equiv 3^8 3^3$$

$$\equiv (25)(27) \equiv 30$$

So a second primitive element is $q' = 30$.

Say we know the logarithms for the base $q = 3$ and want to find the logarithms for a different base, say $q' = 30$. Then we can use equation (2.27), which is

$$\log_{q'} a \equiv \log_q a \ \log_{q'} q \mod (p-1) \tag{2.27}$$

We first need to know $\log_{q'} q = \log_{30} 3 = x$, which is the same as finding $30^x \equiv 3 \mod 43$. By trying different values, we find that $x = 23$.

So for example, given $\log_3 37 = 7$, if we want to find $\log_{30} 37$, we use (1.21) to get

$$\log_{30} 37 \equiv \log_3 37 \log_{30} 3 \mod 42$$

$$\equiv (7)(23) \equiv 35$$

□

Western and Miller have published an entire book of primitive roots and indices [35], and their Introduction gives a fine summary of techniques used up to 1968 to determine these primitive roots and indices.

One interesting characteristic of the discrete logarithm function is that it is comparatively easy to exponentiate, but difficult to calculate logarithms. This disparity is clearer when a large prime p is used.

Example 2.4 Say $p = 4379\,327$. We first need to find a primitive element. In the same way as was done for Example 1.1, we can find one primitive element to be $q = 5$. In order to efficiently exponentiate, we first calculate the following:

$$\begin{aligned}
q^2 &\equiv 25 \\
q^4 &\equiv 625 \\
q^8 &\equiv 390\,625 \\
q^{16} &\equiv 3379\,291 \\
q^{32} &\equiv 128\,922 \\
q^{64} &\equiv 1336\,119 \\
q^{128} &\equiv 3227\,246 \\
q^{256} &\equiv 4203\,401 \\
q^{512} &\equiv 1253\,567 \\
q^{1024} &\equiv 695\,406 \\
q^{2048} &\equiv 2320\,861 \\
q^{4096} &\equiv 3123\,728 \\
q^{8192} &\equiv 1784\,090 \\
q^{16384} &\equiv 3436\,614 \\
q^{32768} &\equiv 2213\,605 \\
q^{65536} &\equiv 598\,417 \\
q^{131\,072} &\equiv 957\,772 \\
q^{262\,144} &\equiv 2715\,275 \\
q^{524\,288} &\equiv 3079\,296 \\
q^{1048\,576} &\equiv 1966\,467 \\
q^{2097\,152} &\equiv 2927\,819 \\
q^{4194\,304} &\equiv 3150\,999
\end{aligned}$$

Say we want to calculate $q^{4000\,000}$.

$$q^{4000\,000} \equiv q^{2097\,152} q^{1048\,576} q^{524\,288} q^{262\,144} q^{65536} q^{2048} q^{256}$$

$$\equiv (2927\,819)(1966\,467)(3079\,296)(2715\,275)(598\,417)(2320\,861)(4203\,401)$$

$$\equiv 1812\,360.$$

□

This method requires on the order of $\log_2 p$ multiplications.

But to calculate logarithms is much more difficult. For instance, given the remainder $\rho = q^x \equiv 1812\ 361$, to find x is very difficult. To try each possible value of x successively would require up to p exponentiations. The problem of finding x is called the *discrete logarithm problem.* It is a difficult problem. Mathematicians have been examining this problem in number theory for over a century [6], but even as late as 1973 [22] little progress had been made on finding an efficient solution to it. Since it is easy to compute $f \equiv q^x$ but hard to compute $f^{-1} = \log_q x$, the logarithm problem is known as a *one-way function.* [24] This makes it a good candidate for a public key distribution system [36].

§2.3 Public Key Systems

With a conventional or *symmetrical* cryptographic cipher a *key* is required, which determines exactly how the plaintext is to be transformed into ciphertext. The sender of the message encrypts it using the key, and the receiver of the ciphertext must have this key in order to unlock the original message. The two parties communicating with one another must agree upon this key before they start communicating, and must keep this key secret from others since the security of any cipher cannot be stronger than the secrecy of the key. This means that whereas the ciphertext message itself can be sent through any channel at all, the key must be sent by means of a secure channel, that is to say a channel that is secure from all means of eavesdropping. For example, if a national government wants to have secure communications with one of its embassies by means of the telephone network, then the keys can be exchanged secretly by means of diplomatic pouch.

The one problem with such exchanges of keys through secure channels is that it can be slow and awkward, and not all users of cryptographic equipment have such secure channels at their disposal. Today larger and larger volumes of valuable information are being transmitted in encrypted form, and the attacks against ciphers are becoming more sophisticated and the computers used in these attacks faster, necessitating frequent changes in the keys used. This means that more and more keys must be exchanged by secure means.

However, an alternative to this need for secure channels is available. It is a class of systems called public key systems or *asymmetrical* systems [2,17,32]. The general idea behind public key systems is that a user will have both a public key and a secret key. The public key is publicly known (for example it can be published in a directory), and if someone wants to send this user a message they encrypt it using this public key. The encryption scheme will be such that when the user receives this encrypted message he can then decrypt it using his private key. Thus for the system to be

strong it is necessary that it be computationally infeasible for someone to calculate the private key from the public one, which can enable them to listen in on this traffic. This means that the system must be based on a *one-way function*, which is a function f such that it is easy to compute f but hard to compute f^{-1}. In the last section it was seen that the logarithm problem provides such a one-way function, and it will be shown in the next chapter how the logarithm problem can be used to secretly exchange user keys.

A popular example of a public key system based on another one-way function is that invented by Rivest, Shamir and Adleman [30], called the *RSA public key system*. The RSA system has the following parameters. The letters p and q denote secret prime numbers known only by one user of the system. Each user has a different p and q. From these two primes can be formed the product $r = p \cdot q$ which, since it is difficult to factor large primes, can be made public without compromising the secrecy of its factors. (Note that if someday an algorithm is discovered that will efficiently factor large primes then the RSA scheme will automatically become insecure.) From r the Euler-totient function $\phi(r) = (p-1)(q-1)$ can be easily calculated by the user. This is kept secret. Since p and q are unknown to others, they cannot easily calculate $\phi(r)$.

The secret key k_s and the public key k_p must be multiplicative inverses modulo $\phi(r)$. In other words they must satisfy

$$k_p \cdot k_s \equiv 1 \mod \phi(r) \tag{2.28}$$

If the public key k_p is chosen arbitrarily, then its inverse k_s exists iff

$$(k_p, \phi(r)) = 1 \tag{2.29}$$

k_s can be found using the Euclidean algorithm [27].

Each user of the system has his own primes p and q, and his own keys k_s and k_p. A directory containing each user's r and k_p is published, as in Table 2.2. Say

Table 2.2

Public Key Directory for RSA Algorithm

r	k_p
r_1	k_{p_1}
r_2	k_{p_2}
⋮	⋮
r_n	k_{p_n}

someone wants to send a message to user 2. They can take their plaintext message and divide it into blocks of numbers modulo r. Say one block is the number X. They calculate and send

$$Y = X^{k_p} \mod r. \tag{2.30}$$

User 2 receives Y and calculates

$$\begin{aligned} Y^{k_s} \mod r &\equiv \left(X^{k_p}\right)^{k_s} \\ &\equiv X \quad (\text{recall Euler} - \text{Fermat theorem}) \end{aligned} \tag{2.31}$$

In this way an encrypted message is transmitted without the need for a secret exchange of keys.

In addition to public key ciphers there also exist public key distribution systems that can be used by two users to determine a private key to be used with a conventional cipher, such as DES. The Diffie-Hellman scheme, based on the logarithm problem, seems to be the most secure of proposed distribution systems to date. It is explained in detail in the next chapter.

CHAPTER III

§3.1 The Logarithm Problem and Previous Research into It

As mentioned in the last chapter, one of the one-way functions that has been used for public key distribution systems is the discrete logarithm function. It was first proposed for a public key distribution system by Diffie and Hellman in 1976 [13]. Given a prime p, a primitive element q in $GF(p)$, and a private key k, the problem of computing the remainder (public key)

$$\rho = q^k \text{ modulo } p \tag{3.1}$$

is an easy one, which need take no more than $log_2 p$ multiplications. However, given p,q and ρ, the problem of computing the private key k is computationally difficult. For large p, a brute-force approach is unacceptable, for on average it will take $\frac{p}{2}$ trials (assuming that k is chosen at random). A polynomial solution has been found, but it is only of mathematical interest since it is much slower than the other methods, slower even than the brute-force approach [34]. The best methods of solution found so far, published by Adleman [1] and by Pohlig and Hellman [29], are still slow enough that the logarithm problem seems to be a good one-way function when used modulo a large prime p. (In the next section we will see that computing logarithms over $GF(p^m)$, where m is an integer, has been proposed, implemented, and seriously attacked.) These methods require on the order of $\sqrt{q}$ steps. But of course there is always the possibility that a sufficiently efficient solution can be found to render the Logarithm Problem useless for cryptographic applications.

To use (3.1) to distribute keys, we let each user (say in a computer network)

choose his own secret private key k and compute his public key ρ. The public keys of the different users can then be listed in a directory, as follows:

Table 3.1

Public Key Directory for Diffie-Hellman Scheme

User	Private key (secret)	Public Key (mod p)
1	k_1	q^{k_1}
2	k_2	q^{k_2}
⋮	⋮	⋮
n	k_n	q^{k_n}

Say users 1 and 2 wish to establish an initial secret key. User 1 looks up the value of q^{k_2} in the directory, and raises it to the power of his secret k_1 (modulo p), obtaining $q^{k_1 k_2}$ mod p. User 2 looks up q^{k_1} and raises it to k_2 also obtaining $q^{k_1 k_2}$ mod p.

A third party, knowing only the public keys in the directory (along with p and q of course) cannot obtain $q^{k_1 k_2}$ mod p, and so the key is secret, known only to users 1 and 2. This assumes that the directory cannot be tampered with by the third party, enabling him to masquerade as another user.

There are a fair number of papers analysing the logarithm problem (see the next Section), especially for when logarithms are computed over $GF(p^m)$, where m is an integer, but these papers all consider algorithms that can perform logarithms for all k. They do not consider the fact that some keys may be considerably "weaker" than others; that is to say, some keys may perhaps be relatively easily determined using methods that are not meant to render solutions for all k.

Diffie and Hellman [13] in their original paper state the obvious fact that the private keys chosen should be "independent random number[s]". Pohlig and Hellman [29] in their paper state that "restrictions might be imposed on k (eg., $k \neq 1$) to avoid simple but improbable transformations ".

In this thesis we examine the strategy of partitioning all the possible remainders ρ into sets in order to determine the private key k. If k can be easily determined in this manner, then q^k and all other elements in its set should be discarded as potential public keys. For a cryptographically large p, these keys will be very few; however, such an analysis of potential keys is, we feel, necessary, since if one of these weak keys does in fact occur, then the security of its user is compromised.

There is also an actual cipher or cryptosystem based on the discrete logarithm problem. It was proposed by Pohlig and Hellman [29], and works as follows.

Let the plaintext message, which is represented as a positive integer less than the prime p, be denoted by the symbol M. Given a key k, a ciphertext message

$$C = M^k \mod p \tag{3.2}$$

can be easily calculated.

If $(k, p-1) = 1$, then k will have a multiplicative inverse $D \equiv k^{-1}$ modulo p. This means that

$$C^D \mod p \equiv (M^k)^D \mod p \equiv M \tag{3.3}$$

so that D is the decryption key. Note that k can never be even.

Since D can be easily determined from k and vice versa, both must be kept secret and exchanged by secure means, so we can regard D and k as simply being two different representations of the same key, and consider this cipher to be a symmetrical one. It is not a public key cipher.

Example 3.1: Let $p = 65543$, which is ideally suited for enciphering 16 bit blocks of data. Say the enciphering key is $k = 47913$, which is relatively prime to $p-1$. The

deciphering key can be found to be

$$D \equiv k^{-1} \equiv 29345.$$

Now say the plaintext message starts with the letters HELP. In ASCII representation the first two letters are C8 C5. The decimal notation for this 16-bit block is $M = 51397$. The ciphertext block will be

$$\begin{aligned} C &\equiv M^k \mod p \\ &\equiv (51397)^{47913} \mod 65543 \\ &\equiv 39369 \end{aligned}$$

The receiver of the cryptogram C will decipher it as follows

$$\begin{aligned} M &\equiv C^D \mod p \\ &\equiv (39369)^{29345} \mod 65543 \\ &\equiv 51397 \end{aligned}$$

to end up with the original message.

□

ElGamal [15] has described a **public key** cryptosystem based on the discrete logarithm problem. It works as follows.

Two users, A and B, want to communicate. User B has a private key x_B, and he has published his public key $y_B \equiv q^{x_B} \mod p$. So user A can randomly choose his secret key k. He computes

$$K = y_B^k \mod p. \tag{3.4}$$

Given a plaintext message m where $0 \leq m \leq p-1$, he can compute

$$c_1 \equiv q^k \mod p \tag{3.5a}$$

and

$$c_2 \equiv Km \mod p \tag{3.5b}$$

and sent c_1, c_2 as the ciphertext message.

User B receives c_1, c_2 and recovers

$$K \equiv (q^k)^{x_B} \equiv c_1^{x_B} \mod p. \tag{3.6}$$

He then divides c_2 by K and recovers the original plaintext message m.

Solving the discrete logarithm problem is equivalent to breaking the above two ecryption schemes.

We now conclude this section with a summary of the previous research that has been done on the discrete logarithm problem.

The logarithm problem has been known for a long time [6], but it wasn't until Diffie and Hellman published their seminal paper [13] in 1976, proposing a public key distribution system based on it, that the earnest attention of mathematicians, computer scientists and engineers has been focused on it.

In 1978 Pohlig and Hellman published a paper devoted entirely to the use of the logarithm problem in cryptography [29], pointing out the weakness of using primes p such that $p-1$ has only small prime factors. They showed that the logarithm problem can be used as an encryption scheme as well as a public key distribution system, which was seen in the last chapter. They also noted that these schemes can be implemented in $GF(p^m)$, where m is an integer, as well as in $GF(p)$.

In 1979, Adleman [1] developed a subexponential algorithm for calculating logarithms modulo a prime, which ran in time $O(e^{(\sqrt{\log_e(p)\log_e\log_e(p)})})$. It was based on an algorithm by Merkle, and in the Hellman and Reyneri paper is referred to as the Merkle-Adleman algorithm. It was based on the important idea (which is expanded upon in this thesis through the use of the sets S and Ψ) that if for some small but not negligible fraction of remainders calculating logarithms is easy, then it must also

be easy for the rest of the remainders. This is because if a non-negligible fraction ϵ of the exponents can be determined, and we have a remainder ρ which is not one of these, then we can randomly (ie, uniformly and independently) pick keys k and calculate $\rho' \equiv q^k \rho$. Since ϵ is not negligible, one of these ρ's has to be easy to solve for.

If the logarithms of the first M primes are known, where the Mth prime is relatively small with respect to p, then any remainder that factors completely into these small primes can have its logarithm easily calculated. This is another important idea that is used in this thesis. Adleman labelled such remainders that factor completely into small primes as being "smooth" numbers.

To compute the logarithms of small primes, Merkle proposed that exponents be randomly chosen from $GF(p)$ and exponentiated until a remainder is found that is smooth. The following equations can then be formed:

$$q^k \equiv \rho = \prod_{i=1}^{N} p_N^{b_i} \mod p \qquad (3.7a)$$

and

$$k \equiv \sum_{i=1}^{N} b_i e_i \mod (p-1) \qquad (3.7b)$$

where p_i are the small primes, b_i are the powers of the primes, and e_i are the exponents such that $p_i \equiv q^{e_i} \mod p$. This process is continued until M invertible independent equations of the form in (3.7b) are obtained. This system of M independent equations is then solved to determine the M exponents e_i. Since the system of equations must be invertible, more than M smooth remainders are needed. Since it cannot be proved that this method can be exponential, Adleman modified Merkle's method by eliminating the need for an invertible set of equations. In this way he was able to obtain a subexponential algorithm.

Then, in the May 1979 issue of IEEE Communications Magazine, Berkovits et al [4] wrote that MITRE Corporation was going to incorporate the logarithm problem

based public key distribution system in an end-to-end encrypted data network. In order to easily implement the algorithm on a microprocessor with binary logic, they decided to have the algorithm perform its operations in a finite field of 2^m elements, namely $GF(2^{127})$, using the prime $p = 2^{127} - 1$.

This announcement was then followed by a number of papers that examined the calculation of logarithms in a $GF(2^m)$.

In 1981 Herlestam and Johannesson [19] published a new method of computing logarithms over $GF(2^p)$, where p is prime, which was based on the interdependent relations

$$f_{r,s}(t) = t^{-2^r} f(t)^{2^s} \tag{3.8a}$$

and

$$\log f_{r,s}(t) = -2^r + 2^s \log f(t) \tag{3.8b}$$

where f and $f_{r,s}$ are polynomials over $GF(2)$. For their algorithm they estimated a mean running time of cp^a, where $a = 9$ for their $p = 31$ that they used.

In 1983 Brickell and Moore [7] published a paper entitled "Some remarks on the Herlestam-Johannesson algorithm for computing logarithms over $GF(2^p)$", in which they confirmed that this algorithm could possibly be a threat to the Diffie-Hellman PKDS. Herlestam and Johannesson showed their algorithm to be polynomial in p over $GF(2^{31})$. For this to be a threat, their algorithm would also have to be polynomial over $GF(2^p)$ for higher p, such as the $p = 127$ used by the MITRE Corporation. Further work would have to be done to determine whether or not this was the case.

At the same time Hellman and Reyneri [18] extended the subexponential Merkle-Adleman algorithm to the case when computation is carried out in $GF(p^m)$, where prime p is fixed and m grows.

In 1984 Blake et al [5] published a paper in which they showed that an algorithm very similar to that of Adleman could feasibly be used to perform logarithms in $GF(2^{127})$. Their approach made use of much precalculation to build up a substantial

but feasible database. Though very similar in philosophy to the Adleman algorithm, theirs is about 100 times as efficient in terms of the number of iterations. However, for a sufficiently large field of characteristic two they still felt that the Diffie-Hellman public key distribution system was a viable technique.

Also in 1984, Coppersmith published a paper [9] in which he showed a still faster way of calculating logarithms in $GF(2^m)$. With his method, the computation of logarithms in a field as large as $GF(2^{400})$ is barely possible, and computations in $GF(2^{127})$ is quite easy. To quote Coppersmith,

> "Throughout this paper we will use for our example the field of $GF(2^{127})$.... To build the database necessary to take logarithms in this field, Adleman's algorithm seems to take two weeks; a modification due to Blake, Fuji-Hara, Mullin, and Vanstone takes about nine hours, and the present scheme [that of Coppersmith] takes eleven minutes. (These estimated timings for an IBM 3081K assume 250 microseconds for a "smoothness test.")"

He concludes that the proposal, made by Diffie and Hellman and implemented by MITRE Corporation, to base the Logarithm Problem PKDS on a field of the form of $GF(2^m)$ was "ill-advised", and that using $GF(p)$ is a far safer approach (as far as we know today).

In addition to this, in 1985 ElGamal [16] developed an algorithm similar to the Merkle-Adleman algorithm to compute logarithms over $GF(p^2)$ which remains sub-exponential in time as $p \nrightarrow \infty$. It uses quadratic fields as the appropriate algebraic structure. At the same time he published a paper [15] suggesting a public key cryptosystem and a signature scheme based on the logarithm problem.

A sub-exponential time algorithm has not yet been discovered for computing logarithms over $GF(p^m)$, where $p^m \rightarrow \infty$ in an arbitrary manner. However, much progress has been made.

So in a sense research on the logarithm problem has come full circle in the past six years. The "traditional" use of $GF(p)$ seems to have been proven the strongest,

though perhaps this is only temporary, since the MITRE Corporation announcement spurred a lot of research into $GF(2^m)$, with researchers momentarily abandoning $GF(p)$. In any case, computing logarithms in $GF(p)$ is the present area where more research is needed, and is the area on which this thesis focuses.

§3.2 Definition of Sets

With p a prime number, let q be a primitive number in the $GF(p)$. As defined in equation (3.1), if we are given p, q and the remainder

$$\rho_k = q^k \mod p, \tag{3.1}$$

where $1 \leq k \leq p-1$, the problem of finding k is called the logarithm problem, and forms the basis for a public key distribution system first proposed by Diffie and Hellman [13]. With $p = 2^u\nu + 1$, ν odd, and a_i as the coefficient of 2^i in the radix-2 representation of k, the coefficients $a_0, a_1, a_2, \ldots, a_{u-1}$ can be determined easily, using recursively the fact that

$$\rho_k^{\frac{p-1}{2}} \equiv 1 \mod p \tag{3.9a}$$

for $a_0 = 0$, and

$$\rho_k^{\frac{p-1}{2}} \equiv -1 \mod p \tag{3.9b}$$

for $a_0 = 1$, regardless of q. Because of this fact, in this thesis we consider the case of $p = 2\nu + 1$ in order to minimize the amount of information about k that can be found in this way. Specifically, we draw attention to certain facts which are useful in choosing values of k which are cryptographically strong.

Our point of view is one of pre-analysing those keys to see if they are weak vis-à-vis these methods of attack.

With $p = 2\nu+1$, ν odd, let m be the smallest number such that ν divides $2^m - 1$. Then $\rho_k^{2^m} \equiv q^{k2^m} \equiv q^{k(2^m-1)}q^k$. Here $q^{k(2^m-1)} = q^{kc\nu} = q^{kc(p-1)/2}$, where c is given by $c = \frac{2^m-1}{\nu}$. Since c is odd, $q^{kc(p-1)/2} \equiv 1$ for even k and -1 for odd k. Thus we have $q^{k2^m} \equiv q^k$ for even k and $q^{k2^m} \equiv -q^k \equiv q^{k+(p-1)/2}$ for odd k. This means that, for a given $\rho_k \equiv q^k$, the size of the set S, which consists of the numbers $\rho_k^{2^i}$ modulo p and $p - \rho_k^{2^i}$ modulo p, is a divisor of $2m$. Furthermore, let S' represent

the set consisting of the numbers $\rho_k^{-2^i}$ modulo p and $p - \rho_k^{-2^i}$ modulo p, and let $\bar{S}$ be the union of S and S'. With $\rho_k \equiv q^k$ (modulo p), we have $-\rho_k \equiv q^{k+(p-1)/2}$, $\rho_k^{-1} \equiv q^{-k} \equiv q^{p-1-k}$ and $-\rho_k^{-1} \equiv q^{-k+(p-1)/2} \equiv q^{p-1-k+(p-1)/2}$. This means that, given a ρ_k, if we can find the exponent corresponding to any one number in S, then k can be determined easily.

Without loss of generality, we can assume that k is even. In this case the set S looks like:

$$S = \begin{Bmatrix} \rho_k & \rho_k^2 & \rho_k^4 & \dots & \rho_k^{2^{\mu-1}} \\ p-\rho_k & p-\rho_k^2 & p-\rho_k^4 & \dots & p-\rho_k^{2^{\mu-1}} \end{Bmatrix} \tag{3.10}$$

where μ is a divisor of m. S' looks like:

$$S' = \begin{Bmatrix} \rho_k^{-1} & \rho_k^{-2} & \rho_k^{-4} & \dots & \rho_k^{-2^{\mu-1}} \\ p-\rho_k^{-1} & p-\rho_k^{-2} & p-\rho_k^{-4} & \dots & p-\rho_k^{-2^{\mu-1}} \end{Bmatrix} \tag{3.11}$$

The numbers in S' can be determined by using the basic relation

$$\rho_k^{-1} = \frac{1+Mp}{\rho_k} \quad \text{for some } M \in GF(p). \tag{3.12}$$

Clearly a chosen k is cryptographically weak if the relevant sets S or S' contain a number whose corresponding exponent can be determined easily.

We can also define the corresponding sets S_k and S'_k as the sets containing the exponents or keys of the remainders in S and S'.

$$S_k = \begin{Bmatrix} k & 2k & 4k & \dots & 2^{\mu-1}k \\ k+\frac{p-1}{2} & 2k+\frac{p-1}{2} & 4k+\frac{p-1}{2} & \dots & 2^{\mu-1}k+\frac{p-1}{2} \end{Bmatrix} \tag{3.13a}$$

$$S'_k = \begin{Bmatrix} p-1-k & p-1-2k & p-1-4k & \dots & p-1-2^{\mu-1}k \\ \frac{p-1}{2}-k & \frac{p-1}{2}-2k & \frac{p-1}{2}-4k & \dots & \frac{p-1}{2}-2^{\mu-1}k \end{Bmatrix} \tag{3.13b}$$

All exponents here are calculated modulo $p-1$. Note that the keys in the top row of S_k and S'_k are all even, while those in the bottom row are all odd.

In passing we note that forming the set S containing a remainder ρ provides an algorithm to find its two square roots. This algorithm takes $\mu - 1$ steps, and if m is not too large it may be a good one to use.

Now we consider another type of set of numbers. We denote this set by the symbol Ψ. The set Ψ looks like:

$$\Psi = \{\rho_k \quad \rho_k^2 \quad q\rho_k^2 \quad q^2\rho_k^4 \quad q^3\rho_k^4 \quad \dots\}, k \text{ odd} \tag{3.14a}$$

$$\Psi = \{\rho_k \quad q\rho_k \quad q^2\rho_k^2 \quad q^3\rho_k^2 \quad q^6\rho_k^4 \quad \ldots\}\ , k \text{ even} \tag{3.14b}$$

The remainders in Ψ can be thought of as occuring in pairs, one remainder with an even exponent and one with an odd exponent. Consider the case when k is odd. The remainders in the set (3.14a) occur in pairs of the form

$$q^{2^i-1}\rho_k^{2^i} \quad , \quad q^{2^{i+1}-2}\rho_k^{2^{i+1}}$$

where $i \geq 0$. Let m be the smallest integer such that ν divides $2^m - 1$, as defined before. The mth pair of remainders to be formed is

$$q^{2^m-1}\rho_k^{2^m} \quad , \quad q^{2^{m+1}-2}\rho_k^{2^{m+1}}$$

Since ν divides $2^m - 1$ and both are odd, then $c\nu = 2^m - 1$, where c is some odd constant. This means that $q^{2^m-1} \equiv q^{c\nu} \equiv q^{\frac{p-1}{2}c} \equiv (-1)^c \equiv -1$. So the mth remainder with odd exponent is

$$\begin{aligned} q^{2^m-1}\rho_k^{2^m} &\equiv q^{2^m-1}q^{k2^m} \\ &\equiv q^{c\nu}q^{k(c\nu+1)} \\ &\equiv q^{(k+1)c\nu}q^k \\ &\equiv q^{(\text{even})c\nu}q^k \\ &\equiv q^k \end{aligned}$$

So after m pairs we return to ρ_k. The same can be shown for even k since it is the same set, except that we multiply by q first and then square the remainder. This means that the size of the set Ψ is $2m$ or a divisor of $2m$.

It is clear that if the exponent of any one number in Ψ is known, then that of any other number in Ψ can be computed. Thus, a chosen k is cryptographically weak if the corresponding set Ψ contains a number whose exponent is known or can be computed comparatively easily.

Corresponding to Ψ we can define Ψ_k as the set containing the exponents of the numbers in Ψ. It looks like:

$$\Psi_k = \{k \quad 2k \quad 2k+1 \quad 4k+2, \quad \ldots \quad 2^{\mu-1}k + 2^{\mu-1} - 1 \quad 2^\mu k + 2^\mu - 2\},\ k \text{ odd} \tag{3.15a}$$

$$\Psi_k = \{k \quad k+1 \quad 2k+2 \quad 2k+3 \quad \ldots \quad 2^{\mu-1}k + 2^\mu - 2 \quad 2^{\mu-1}k + 2^\mu - 1\},\ k \text{ even} \tag{3.15b}$$

§3.3 Forming Congruences in S_k and Ψ_k

Suppose by examining the remainders in S or Ψ we can find a congruence relation consisting of the product of some of these remainders which are raised to some power that can be a function of k and of perhaps some remainders with known exponents. If these powers are represented by $n_i(k)$, then we have

$$\rho_{k_0}^{n_0(k)} = (\rho_{k_1}^{n_1(k)})(\rho_{k_2}^{n_2(k)})(\rho_{k_3}^{n_3(k)}) \ldots (\rho_{k_a}^{n_a(k)}) q^b \text{ modulo } p \tag{3.16}$$

where $n_i \geq 1$,$a \geq 1$, and $b \geq 0$. q^b represents the remainders with known exponents. Since all of the $n_i(k)$'s can be expressed in terms of one unknown key k, we can define a congruence relation in S_k or Ψ_k, with k being the only unknown:

$$n_0(k) = n_1(k) + n_2(k) + n_3(k) + \ldots + n_a(k) + b \text{ modulo } (p-1). \tag{3.17}$$

We note that if m is large relative to p, then it is not always necessary to form the entire set S or Ψ. As soon as a congruence relation is found we can stop forming additional remainders.

The larger m is, the more numbers there are in a set, and the better the chances are of finding a relationship. However, there is a memory and computation constraint, and for a cryptographically large p, if m is also large relative to p, as is often the case when $\frac{p-1}{2} = \nu$ is prime, then a relationship realistically cannot be searched for in the entire set. Only a portion of the set, starting with $\rho \equiv q^k$, can be used.

On the other hand, if m is very small, then with so few numbers in a set relationships are unlikely. But the two cases balance out in that if m is small, then different related sets can be combined for the search, for example by forming the sets containing ρ^3 or ρ^5.

We now illustrate this with the following examples.

Example 3.2: Suppose $p = 683$ and $q = 5$, which is one possible primitive element. To see how $GF(683)$ is partitioned into sets S, see Appendix I. Say, $\rho_k \equiv q^k \equiv 681$

(modulo p).

$$S_1 = \begin{Bmatrix} 681 & 4 & 16 & 256 & 651 & 341 & 171 & 555 & 675 & 64 \\ 2 & 679 & 667 & 427 & 32 & 342 & 512 & 128 & 8 & 619 \end{Bmatrix}$$

We see that $512 \equiv 2^9$. This is our congruence expression.

$$2 \equiv q^{k+341} \bmod p.$$

$$512 \equiv q^{64k+341} \equiv \left(q^{k+341}\right)^9 \mod p \text{ in the format of (3.15)}$$

$$64k + 341 \equiv 9k + 9(341) \bmod (p-1)$$

$$682M = 55k$$

$$k \in \{62, 124, 186, 248, 310, 372, 434, 496, 558, 620, 682 \equiv 0\}$$

By calculating q^k for each of the 11 possible keys, we find that $q^{558} \equiv 681$.

Therefore $k = 558$

□

Example 3.3: Let us again suppose that $p = 683$ and $q = 5$. Say we want to solve for $q^k \equiv 43$.

$$S_2 = \begin{Bmatrix} 483 & 386 & 102 & 159 & 10 & 100 & 438 & 604 & 94 & 640 \\ 200 & 297 & 581 & 524 & 67 & 583 & 245 & 79 & 589 & 43 \end{Bmatrix}$$

From the previous example we know that $2 \equiv q^{558+341} \equiv q^{217}$. We will use that fact here.

Examining S_2 we see that $q^{2k+341} \equiv 200$ and $q^{64k} \equiv 100$. Thus we have

$$q^{2k+341} \equiv q^{64k} 2 \equiv q^{64k+217} \text{ modulo } 683$$

which is our congruence expression. We obtain

$$2k + 341 \equiv 64k + 217 \text{ modulo } 682$$

or

$$62k = 682M + 124$$

or

$$k = 11M + 2.$$

Since q^k appears in the bottom row of S_2, k is odd, so we have $k = 13 + 22M$, $M \in \{0, 1, 2, 3, 4, 5, \ldots, 29, 30\}$. By computing q^{13+22M} modulo p for the different possible values of M, we find that $q^{497} \equiv 43$ for $M = 22$.

Therefore $k = 497$.

On the other hand, we have from S_1 that

$$640 \equiv q^{k+341} \equiv (128)(5)$$

$$\equiv 2^7 q$$

$$\equiv (q^{217})^7 q$$

$$\equiv q^{156}$$

Therefore $k = 497$.

□

Note that here we are able to directly determine the unique key k needed, whereas for the first solution we are left with a set of possible values of k, and have to try each one in turn. This is due to the fact that S is constructed independent of the value of q, so a number of possible keys are obtained. But if in the solution we utilize the actual value of q then a unique solution may be obtained. If we use the set Ψ instead of S, then a unique solution can be directly obtained because q is used in the construction of Ψ.

The following example shows how we can combine two sets to obtain a solution.

Example 3.4: With $p = 683$ and $q = 5$ as in the previous two examples, suppose $\rho_k \equiv q^k \equiv 67$. Then the corresponding S_1 would be:

$$S_1 = \begin{Bmatrix} 67 & 391 & 572 & 27 & 46 \\ 616 & 292 & 111 & 656 & 637 \end{Bmatrix}$$

In S_1 there are no numbers between which we can establish a relationship of the kind possible in Example 3.2. However, we have the number 27, which is 3^3. Constructing the set S_2 for 3, we have

$$S_2 = \begin{Bmatrix} 3 & 9 & 81 & 414 & 646 \\ 680 & 674 & 602 & 269 & 37 \end{Bmatrix}$$

Let $q^{k_1} \equiv 3$. We see that $111 \equiv (3)(37)$ whereas from S_1 we have $111 \equiv q^{4k+341}$ and from S_2 we have $37 \equiv q^{16k_1+341}$. This gives $q^{4k+341} \equiv q^{k_1} q^{16k_1+341}$.

$$4k \equiv 17k_1 \pmod{682}$$

Since k and k_1 are related by $27 \equiv q^{8k} \equiv q^{3k_1}$ or $8k \equiv 3k_1 \bmod 682$, we can combine the two to get

$$31k_1 = 682M$$

So k_1 is a multiple of 22.

Computing q^{M22} modulo 682, $M = 1, 2, 3, \ldots$, we find $q^{330} \equiv 3$. This means for S_1 that $q^k \equiv 67 \equiv 27^4 \equiv (q^{330})^1 2 \equiv q^{550}$.

Therefore $k = 550$.

□

The following example shows the same procedure for the set Ψ.

Example 3.5: Suppose again that $p = 683$ and $q = 5$. Let $\rho_k = 499$. Since $\rho_k^{(p-1)/2} \equiv -1$, we conclude that k is odd. We find the set Ψ to be:

$$\Psi = \{499, 389, 579, 571, 123, 103, 515, 221, 422, 504, 471, 549,$$
$$13, 169, 162, 290, 84, 226, 447, 373\}$$

Examining the numbers of Ψ we find

$$504 \equiv q^{32k+30}$$

$$162 \equiv q^{128k+127}$$

$$84 \equiv q^{256k+255}$$

or

$$q^{32k+30} \equiv (8)(7)(9)$$

$$q^{128k+127} \equiv (2)(81)$$

$$q^{256k+255} \equiv (4)(3)(7)$$

Manipulation of these three congruences leads to

$$q^{1024k+314} \equiv 1$$

after assuming that $q^{217} \equiv 21$. This means that

$$1024k + 314 = M682$$

or

$$342k = M682 + 368 \tag{3.18}$$

This gives

$$0 \equiv M169 + 26(\text{mod}171)$$

or

$$0 \equiv M13 + 2(\text{mod}171)$$

which in turn gives

$$M = 13M_1 + 171$$

Using this in (3.18) we have

$$2k = 682M_1 + 54$$

This implies, since k is odd, that

$$k = 27.$$

□

As mentioned in §3.1, Adleman [1] showed that knowing the logarithms for small prime remainders allows the easy calculation of logarithms of smooth numbers, that is to say, numbers that factor completely into these small primes. Such smooth remainders make extremely weak public keys.

Here we extend this observation to include use of the sets S and S'. If a remainder is in a set which also includes a smooth remainder, then this remainder is also unsatisfactorily weak (unless, of course, m is large with respect to p).

Below is an example where the logarithms of a few small primes are calculated, and then used to determine the exponent of an arbitrary non-smooth potential public key.

Example 3.6: In this example we will use $p = 20347$. Its cardinality is $m = 226$. A few of the possible primitive elements are

$$q = 3, 5, 12, \ldots, 17432, 17435, 17439, \ldots, 20341.$$

Let $q = 17439$. We first want to get the logarithms of some small primes. The set S_1 used is contained in Appendix II.

Let us examine the set S_1 containing q, since the exponents of all its elements can be so easily calculated. It contains the remainders 40 and 160. $40 \equiv q^{2^{89}} \mod p$ and

$$2^{89} \bmod (p-1) \equiv 2^{64}2^{16}2^9 \equiv 19970$$

Therefore $40 \equiv q^{19970}$

$160 \equiv q^{2^{213}} \bmod p$ and $2^{213} \bmod (p-1) \equiv 2^{128}2^{64}2^{16}2^5 \equiv 16664$ Therefore $160 \equiv q^{16664}$

Let $q^{k_2} \equiv 2$. $160 = (2)^2(40)$

$$q^{16664} \equiv q^{2k_2}q^{19970} \bmod p$$

and $16664 \equiv 2k_2 + 19970 \bmod (p-1)$ giving $k_2 = 8520$ or 18693.

$$q^{18693} \equiv q^{16384}q^{2048}q^{256}q^4q$$

$$\equiv (6166)(12819)(2989)(19765)(17439) \equiv 2$$

$$\text{Therefore } q^{18693} \equiv 2 \tag{3.19}$$

Let $q^{k_5} \equiv 5$. $q^{19970} \equiv 40 \equiv (2)^3(5) \equiv q^{3(18693)}q^{k_5}$

$19970 \equiv 3(18693) + k_5 \bmod (p-1)$ giving $k_5 = 4583$

$$\text{Therefore } q^{4583} \equiv 5 \tag{3.20}$$

One of the remainders is $12960 = (2)^5(3)^4(5) \equiv q^{2^{172}} \bmod p$.

$2^{172} \bmod (p-1) \equiv 2^{128}2^{32}2^{12} \equiv (19204)(8080)(4096) \equiv 12382$

Let $q^{k_3} \equiv 3$. $12960 \equiv q^{12382} \equiv (2)^5(3)^4(5) \equiv q^{5(18693)}q^{4k_3}q^{4583}$

$12382 \equiv 5(18693) + 4k_3 + 4583 \mod 20346$ giving $k_3 = 4016$ or 14189.

$$q^{14189} \equiv q^{8192}q^{4096}q^{1024}q^{512}q^{256}q^{64}q^{32}q^8q^4q \equiv 3$$

$$\text{Therefore } q^{14189} \equiv 3 \tag{3.21}$$

Another remainder is $420 = (2)^2(3)(5)(7) \equiv q^{2^{156}}$.

$$2^{156} \bmod (p-1) \equiv 2^{128}2^{16}2^{12} \equiv 13324$$

Let $q^{k_7} \equiv 7$. $420 \equiv q^{13324} \equiv (2)^2(3)(5)(7) \equiv q^{2(18693)}q^{14189}q^{4583}q^{k_7}$

$$13324 \equiv 2(18693) + 14189 + 4583 + k_7 \mod 20346$$

$20346M = 2142 + k_7$, giving $k_7 = 18204$.

$$\text{Therefore } q^{18204} \equiv 7 \tag{3.22}$$

Another remainder is $4851 = (3)^2(7)^2(11) \equiv q^{2^{104}}$.

$$2^{104} \bmod (p-1) \equiv 2^{64}2^{32}2^8 \equiv 8908$$

Let $q^{k_{11}} \equiv 11$. $4851 \equiv q^{8908} \equiv (3)^2(7)^2(11) \equiv q^{2(14189)}q^{2(18204)}q^{k_{11}}$ $8908 \equiv 2(14189) + 2(18204) + k_{11} \mod 20346$, giving $k_{11} = 5160$.

$$\text{Therefore } q^{5160} \equiv 11 \tag{3.23}$$

To summarize, we have determined, using the set S_1 containing q, the following exponents:

$$2 \equiv q^{18693}$$

$$3 \equiv q^{14189}$$

$$5 \equiv q^{4583}$$

$$7 \equiv q^{18204}$$

$$11 \equiv q^{5160}$$

If we wanted to, we could continue and find the logarithms of more small primes. But, given even these few logarithms, the computation of the exponents of arbitrary remainders is much easier.

Say we have $\rho \equiv q^k \equiv 6221$, which is not smooth. We form the set containing ρ and find that it contains

$$4200 = (2)^3(3)(5)^2(7)$$

$$\equiv q^{3(18693)} q^{14189} q^{2(4583)} q^{18204}$$

$$\equiv q^{16254}$$

$$4200 \equiv q^{2^{61}k}$$

$$2^{61} \equiv 2^{32}2^{16}2^{13} \bmod 20346$$

$$\equiv 2054$$

$$4200 \equiv q^{2054k} \equiv q^{16254}$$

$$2054k = 16254 + 20346M$$

$$k = \frac{8127 + 10173}{1027} = 4089 \text{ or } 14262$$

Since $1027 = (13)(79)$, we can find that $M = 9 + 13M_1$. Trying successive values of M_1, we arrive at $q^{14262} \equiv 6221$.

Therefore $k = 14262$

□

§3.4 Lower Limit on Cardinality m

Given p, it is possible to determine a lower limit on the cardinality m. Let q be an arbitrary primitive element in $GF(p-1)$. We know that q^i is unique for $i = 1, 2, 3, ...p-1$.

The easiest way to calculate q^i for an arbitrary i is to first precalculate q^2, q^4, q^8,..., q^{2^j}, ..., q^{2^N}, where N is the largest integer such that $2^N < p-1$. (2^N cannot equal $p-1$ because this would make $\frac{p-1}{2}$ even.) q^i is the product of terms whose exponents give the radix-2 representation of i. In order for each q^i to be unique, each q^{2^j} must also be unique, up to $j = N$.

Now, the set S containing q must also contain these unique q^{2^j}'s. Since q^1 has an odd exponent, it is positioned in the bottom row. The smallest possible set S would occur if $q^{2^N} \equiv p - q$, placing q^{2^N} directly above q.

This sets a lower limit on the cardinality of p as being $m \geq N$, where N is the largest integer such that $N < log_2(p-1)$. So $N + 1 > log_2(p-1)$, and

$$m > log_2(p-1) - 1 \tag{3.24}$$

Consider the case of q^{2^N} being directly above q. In this case $2^N = 1 + \frac{p-1}{2}$.

$$p = 2^{N+1} - 1 \tag{3.25}$$

This makes p a Mersenne prime, with $m \geq N$.

However, by definition, m is the smallest integer such that $2^m - 1$ divides $\frac{p-1}{2}$. With a Mersenne prime, $\frac{p-1}{2} = 2^N - 1$, giving $m = N$.

So for a Mersenne prime, $p = 2^{m+1} - 1$.

$$m = log_2(p+1) - 1 \tag{3.26}$$

$$m \approx log_2(p-1) - 1 \tag{3.27}$$

for large Mersenne primes.

Mersenne primes have the lowest possible cardinality m relative to their magnitude p.

The following table shows some typical values of m for $p \approx 2436000$. Also included is the atypical Mersenne prime $p = 2^{31} - 1$.

Table 3.2

Cardinality of some primes

p	m	$\log_2(p-1)$	$m \geq$
		-1	
2437571	30272	20.22	21
2437583	43212	20.22	21
2437607	8688	20.22	21
2437619	303645	20.22	21
2437639	1596	20.22	21
2437663	192438	20.22	21
2437667	237270	20.22	21
2437691	121884	20.22	21
2437751	69300	20.22	21
2437763	24534	20.22	21
$2^{31} - 1$	30	almost 30	30

§3.5 An Analysis in S_k

The "brute force" approach to solving for k is to calculate

$$q^{i+1} = (q^i)(q) \mod p \tag{3.28}$$

until we arrive at $q^k \equiv q_k$.

By making p cryptographically large, this method becomes useless for all but the smallest of exponents.

However, when the sets $\tilde{S}$ and Ψ are formed, we can have as many as $6m$ remainders with different but related exponents. If even one of these exponents is small enough that the "brute force" approach will work on it, then all of the exponents should be discarded as potential keys.

The following example illustrates this approach:

Example 3.7 Let $p = 30059$ and $q = 7$. Suppose we have $\rho_k = 1744$. Since $1744^{(p-1)/2} \equiv 1$, k is even. The first few terms of the top two rows of S are:

$$S = \left\{ \begin{array}{cccccccc} 1744 & 5577 & 21923 & 4578 & 6961 & 413 & 20274 & 8310 \\ 28315 & 24482 & 8136 & 25481 & 23098 & 29646 & 9785 & 21749 \end{array} \right.$$

$$\left. \begin{array}{ccccc} 10577 & 23390 & 18300 & 2681 & 3660 \quad \ldots \\ 19482 & 6669 & 11759 & 27378 & 26399 \quad \ldots \end{array} \right\}$$

If for each elements of S we use the "brute force" approach up to say q^{100}, we will find that $q^{37} \equiv 11759$.

$$q^{1024k+15029} \equiv q^{37} \mathrm{mod} p$$

$$1024k + 15029 \equiv 37 \mathrm{mod}(p-1)$$

$$1024k + 14992 = 30058M$$

$$512k + 7496 = 15029M$$

M must be even and give an integer k for

$$k = \frac{15029M - 7496}{512}$$

For $M = 296$, $k = 8674$.

□

Just out of a matter of interest, we now state the following assertion concerning the upper bound for the smallest exponent in an arbitrary set $\bar{S}_k = S_k \cup S'_k$.

Assertion 3.1: If $p-1$ is not divisible by 6, then the set $\bar{S}_k$ for any k contains a number $\leq \frac{p+11}{12}$. If $p-1$ is divisible by 6 then the set contains a number $\leq \frac{p-1}{6}$.

Proof of Assertion 3.1

The four rows of the set $\bar{S}_k$ consist of numbers of the form $2^i k$, $2^i k + \frac{p-1}{2}$, $p-1-2^i k$ and $\frac{p-1}{2} - 2^i k$, all numbers computed modulo $p-1$. For a given i, each of these four numbers falls in a different interval among the four intervals

$$\left[1, \frac{p-3}{4}\right], \left[\frac{p+1}{4}, \frac{p-1}{2}\right], \left[\frac{p+1}{2}, \frac{3p-5}{4}\right] \text{ and } \left[\frac{3p-1}{4}, p-1\right]. \tag{3.29}$$

This means that the set $\bar{S}_k$ contains numbers in the first interval of the form $\frac{p-3}{4} - r$, and in the second interval of the form

$$\frac{p-1}{2} - \frac{p-3}{4} + r = \frac{p+1}{4} + r\ ,$$

where $0 \leq r \leq \frac{p-7}{4}$. Since $\bar{S}_k$ contains $\frac{p-3}{4} - r$, it must also contain

$$2\left(\frac{p-3}{4} - r\right) = \frac{p-3}{2} - 2r = \frac{p-1}{2} - 2r - 1.$$

This in turn means that $\bar{S}_k$ must contain

$$\frac{p-1}{2} - \left(\frac{p-1}{2} - 2r - 1\right) = 2r + 1.$$

Thus the set $\bar{S}_k$ contains numbers of the form

$$\frac{p-3}{4} - r \ , \ \frac{p+1}{4} + r \ , \ 2r+1. \tag{3.30}$$

With reference to (3.30), the lowest number will be either $\frac{p-3}{4} - r$ or $2r+1$. If $p-1$ is divisible by 6 then we can set $\frac{p-3}{4} - r_0 = 2r_0 + 1$ to get an upper bound on the lowest number. In this case $r_0 = \frac{p-7}{12}$, so that $2r_0 + 1 = \frac{p-1}{6}$. This leads to the following assertion:

Assertion 3.1.1: If $p-1$ is divisible by 6, then for any k the set $\bar{S}_k$ contains a number $\leq \frac{p-1}{6}$.

To discuss the case of $p-1$ not being divisible by 6, let $n_0 - 1$ represent the largest odd integer $< \frac{p-1}{6}$. This number $n_0 - 1$ corresponds to the situation when $\frac{p-3}{4} - r$ is just greater than $2r+1$, implying with reference to (3.30) that $\bar{S}_k$ contains numbers of the form $n_0 + 1 - j$, $n_0 + j$, $n_0 - 1 - 2j$, where $0 \leq j \leq \frac{n_0-2}{2}$. Now let us consider the array of numbers shown below.

A	B	C
$n_0 + 1$	n_0	$n_0 - 1$
n_0	$n_0 + 1$	$n_0 - 3$
$n_0 - 1$	$n_0 + 2$	$n_0 - 5$
$\vdots$	$\vdots$	$\vdots$
$n_0 + 1 - j$	$n_0 + j$	$n_0 - 1 - 2j$
$\vdots$	$\vdots$	$\vdots$
$n_0 + 1 + \frac{n_0-2}{2}$	$n_0 + \frac{n_0-2}{2}$	1

Case 1: If $n_0 + 1 - j$ in A is even, then $\frac{n_0+1-j}{2}$ also belongs to $\bar{S}_k$ and we have

$$\frac{n_0 + 1 - j}{2} \leq \frac{n_0 + 1}{2}. \tag{3.31}$$

But $n_0 - 1$ is very nearly equal to, but less than, $\frac{p-1}{6}$. Therefore,

$$\frac{n_0 + 1 - j}{2} \leq \frac{p+11}{12} \tag{3.32}$$

Case 2: If $n_0 + 1 - j$ in A is odd, then $n_0 + j$ in B is even. If $n_0 + j$ is divisible by 4, then $\frac{n_0+j}{4}$ belongs to $\bar{S}_k$ and

$$\frac{n_0 + j}{4} \leq \frac{n_0 + \frac{n_0-2}{2}}{4} = \frac{3n_0 - 2}{8} < \frac{3\left(\frac{p+5}{6}\right) - 2}{8} = \frac{3p + 15 - 12}{48} = \frac{p+1}{16}. \tag{3.33}$$

Case 3: If $n_0 + j$ in B is even, but not divisible by 4, then $p - 1 - (n_0 + j)$ is divisible by 4, since $\frac{p-1}{2}$ and $\frac{n_0+j}{2}$ are odd. Therefore $\frac{p-1-(n_0+j)}{4}$ belongs to $\bar{S}_k$ and we have

$$\frac{p - 1 - \left(n_0 + \frac{n_0-2}{2}\right)}{4} \leq \frac{p - 1 - (n_0 + j)}{4} \leq \frac{p - 1 - n_0}{4}. \tag{3.34}$$

Here we recall that $n_0 - 1 \leq \frac{p-1}{6}$. Using this fact we have

$$\frac{p - 1 - \left(\frac{p+5}{6} + \frac{p-7}{12}\right)}{4} \leq \frac{p - 1 - (n_0 + j)}{4} \leq \frac{p - 1 - \left(\frac{p+5}{6}\right)}{4} \tag{3.35a}$$

or

$$\frac{3p - 5}{16} \leq \frac{p - 1 - (n_0 + j)}{4} \leq \frac{5p - 11}{24}. \tag{3.35b}$$

Here $\frac{3p-5}{16} \geq \frac{p+5}{6} = n_0$ if $p \geq 55$ and $\frac{5p-11}{24} \leq \frac{p-7}{4} = n_0 + \frac{n_0-2}{2}$ if $p \geq 53$. This means that $\frac{p-1-(n_0+j)}{4}$ is in column B. If $\frac{p-1-(n_0+j)}{4}$ is odd, then the corresponding number in A is even and the situation belongs to Case 1. If $\frac{p-1-(n_0+j)}{4}$ is divisible by 4, then we have Case 2. If $\frac{p-1-(n_0+j)}{4}$ is even, but not divisible by 4, then $\frac{p-1-(n_0+j)}{8}$ is in $\bar{S}_k$ and we have

$$\frac{p - 1 - \left(n_0 + \frac{n_0-2}{2}\right)}{8} \leq \frac{p - 1 - (n_0 + j)}{8} \leq \frac{p - 1 - n_0}{8} \tag{3.36a}$$

or

$$\frac{3p - 5}{32} \leq \frac{p - 1 - (n_0 + j)}{8} \leq \frac{5p - 11}{48}. \tag{3.36b}$$

Recalling that $n_0 - 1 \leq \frac{p-1}{6}$, we have

$$\frac{5p-11}{48} \leq \frac{5p-11}{6} \approx n_0 + 1$$

and

$$n_0 + 1 + \frac{n_0 - 2}{2} = \frac{3n_0}{2} \approx 3\left(\frac{p+5}{12}\right) = \frac{p+5}{12} \geq \frac{3p-5}{32}.$$

Thus, $\frac{p-1-(n_0+j)}{8}$ is in A and it is odd, indicating that we have Case 2.

In the early part of the discussion under Case 3 we had to stipulate the condition that $p \geq 55$. For all primes $p < 55$, $p = 2\nu + 1$, ν odd, $p - 1$ not divisible by 6, we find by actually forming the sets $\bar{S}_k$ that in every case there is a number $< \frac{p+11}{12}$.

In view of this fact and Cases 1,2 and 3, we can now make the following assertion:

Assertion 3.1.2: If $p - 1$ is not divisible by 6, then the set S_k for any k contains a number $\leq \frac{p+11}{12}$.

□

§3.6 Other Possible Sets

The sets S and Ψ are not the only possible sets that can be formed in $GF(p)$ and have the property of the exponents of the elements being related to one another.

For example, we can form a set by repeatedly cubing the remainder ρ. Let us call this set C, and the set of the corresponding exponents C_k.

$$C = \{q^k, q^{3k}, q^{9k}, q^{27k}, \ldots, q^{3^{\mu-1}k}\} \tag{3.37}$$

where μ is a divisor of m, which is the cardinality of the set C. Using the same argument as for the set S, it can be shown that m is the smallest number such that μ divides $3^m - 1$.

When a set C_1 containing ρ is formed, a second set C_2 containing $-\rho$ can be formed. The nth element of C_1 will be the negative of the nth element of C_2. This is because

$$\begin{aligned}(-\rho)^3 \equiv (p-\rho)^3 &\equiv (p-\rho)(p^2 - 2p\rho + \rho^2) \\ &\equiv p^3 - 2p^2\rho + p\rho^2 - \rho p^2 + 2p\rho^2 - \rho^3 \\ &\equiv -\rho^3\end{aligned}$$

Since we know the relationships between the exponents of both sets, we can use both when searching for congruences.

Example 3.8: Let $p = 683$ and $q = 5$. Say $\rho \equiv q^k \equiv 37$. Then

$$C = \left\{\begin{matrix} 37 & 111 & 265 & 607 & 193 & 482 & 269 & 292 & 372 & 455 \\ 430 & 336 & 602 & 616 & 440 & 240 & 80 & 433 & 674 & 637 \\ 333 & 325 & 545 & 112 & 680 & 656 & 124 & 371 & 316 & 579 \end{matrix}\right\}$$

Since $111 = (3)(37)$ and $333 = (3^2)(37)$, a congruence relation can be found.

$$111 \equiv q^{3k} \equiv (3)(37) \equiv (3)q^k \Longrightarrow 3 \equiv q^{2k}$$

$$333 \equiv q^{3^2 0k} \equiv (3^2)(37) \equiv (3^2)q^k \equiv (q^{4k})q^k$$

$$3^{20}k \equiv 4k + k \bmod 682$$

$$67k \equiv 5k$$

$$62k \equiv 682M$$

$$k = 11M \quad \text{(62 possible values)}$$

We find that $q^{165} \equiv 37$.

Therefore $k = 165$.

□

Other sets can be formed by repeatedly raising the remainder to some other power. In this thesis we have not examined these sets, but they can be used in the same manner, and if the exponent of a remainder can be found easily using one of these sets, then it should not be used as a cryptographic key.

§3.7 Some Other Observations

With p any prime, let $\rho_k = q^k$ modulo p be any remainder. Let E_k be the smallest number such that p divides $\rho_k^{E_k} - 1$. This means that $q^{kE_k} \equiv 1$ (modulo p), implying that

$$kE_k = M(p-1) \tag{3.38}$$

since $q^{p-1} \equiv 1$ (modulo p).

Now, $\rho_k^{E_k} \bmod p = 1$, so that E_k is the order of ρ_k, which we recall is an element of $GF(p)$. If ρ_k is a primitive element in $GF(p)$, then $E_k = p - 1$, while if ρ_k is not a primitive element then E_k should divide $p - 1$, since the order of any element is a divisor of $p - 1$.

So we can rewrite (3.38) as

$$k = \frac{M(p-1)}{E_k} \tag{3.39}$$

which leads to the following assertion.

Assertion 3.2: k can be obtained in at most E_k trials by computing $q^{M(p-1)/E_k}$, $M = 1, 2, 3, 4, \ldots, E_k$ until the remainder modulo p is ρ_k.

Note that since $\rho_k^{E_k} - 1 \geq p$,

$$E_k \geq \log_{\rho_k}(p+1) \tag{3.40}$$

To make this approach as difficult as possible, E_k should be large, ideally $p - 1$ or $\frac{p-1}{2}$, as is often the case.

Also note that $\rho_k^{E_k} - 1$ grows slowest for small ρ_k. As shown in §3.3 the exponents of remainders that are small primes helps with calculating logarithms of arbitrary remainders. This method works best for these small primes, which is good. It works more poorly for typical values of ρ_k.

For the special case when $\frac{p-1}{2}$ is prime, Assertion 3.2 yields no information. This is because if $E_k = p-1$ then we are using the "brute force" approach, while if $E_k = \frac{p-1}{2}$ then we are only determining if ρ_k is even or odd, which can be done much more easily by using (3.9).

The following two examples illustrate this Assertion. The first involves a Mersenne prime, where this method is extremely powerful for $\rho_k \equiv 2$.

Example 3.9: $p = 2147483647 = 2^{31} - 1$ and $\rho_k \equiv 2$. So $E_k = 31$ and at most 30 trials are needed.

$$q^{M(p-1)/E_k} \equiv q^{M_1 69273666}$$

For $q = 7$, we find that $7^{(7)(69273666)} \equiv 2$.

Therefore $k = 484915662$.

□

Example 3.10: $p = 151$. What is k for $\rho_k \equiv 2$?

$$p - 1 = 150 = (2)(3)(5)(5)$$

$$E_k \geq \log(p+1)$$

$$\geq 7.2479$$

$$\geq 8$$

$$E_k \in \{10, 15, 25, 30, 50, 75, 150\}$$

$$\frac{2^{10}-1}{151} \approx 6.77$$

$$\frac{2^{15}-1}{151} = 217 = \text{integer} \rightarrow E_k = 15$$

$$k = \frac{M(p-1)}{E_k} = 10M$$

Using Assertion 3.2, we compute q^{10M}, $M = 1,2,3,4,\ldots,15$, until we arrive at $q^{70} \equiv 2 \equiv \rho_k$.

Therefore $k = 70$

□

Suppose that by using this result we are able to determine β for $q^\beta \equiv 2$ modulo p. Then we can make the following remark:

Remark: Suppose for a moment that the set S_1 contains two numbers ρ and $\rho 2^j$, where

$$\rho = q^b \text{modulo } p \tag{3.41a}$$

and

$$\rho 2^j = q^{b2^i + \alpha(\frac{p-1}{2})} \text{ modulo } p \tag{3.41b}$$

Then i and j are related by

$$b(2^i - 1) = M(p-1) + \beta j \; - \alpha(\frac{p-1}{2}) \tag{3.42}$$

where $\alpha = 0$ or 1.

From this remark it is clear that if β is known, and a set contains two numbers ρ and $\rho 2^j$, then the search to find these numbers is easier, since only some values of i are allowed. Having found them, b, and from it the keys to all the numbers in S_1, can be easily determined. We can find the value of b by calculating q^b modulo p for each b that satisfies (3.42). An upper bound to the number of trials needed is

$$T = \text{GCD}(2^{i-1}, p-1) \tag{3.43}$$

This means, from a cryptographic point of view, that if

$$p - 1 = p_1^{n_1} p_2^{n_2} \ldots p_r^{n_r}, \tag{3.44}$$

where each p_i is prime, then at least some p_i's should be large. † Also, $2^i - 1$ should not be a multiple or factor of $p - 1$.

We will now illustrate this remark with an example.

Example 3.11: $p = 8191$, for which $m = 12$. One possible primitive element is $q = 2739$. What is k for $q^k \equiv 51$?

First let us find β for $q^\beta \equiv 2$. Since $p = 8191 = 2^{13} - 1$, $E_k = 13$.

$$q^{M(p-1)/E_k} \equiv q^{630M} \bmod 8191$$

$$q^{630} \equiv q^{512}q^{64}q^{32}q^{16}q^4q^2$$

$$\equiv (8121)(394)(3554)(7130)(990)(7356)$$

$$\equiv 256$$

$$q^{(2)(630)} \equiv (256)^2 \equiv 8$$

$$q^{(3)(630)} \equiv (256)^3 \equiv 2048$$

$$q^{(4)(630)} \equiv (8)^2 \equiv 64$$

$$q^{(5)(630)} \equiv (64)(256) \equiv 2 \equiv q^\beta$$

Therefore $\beta \equiv 3150$

$$b(2^{i-1}) = M_1(p-1) + \beta j - \alpha(\frac{p-1}{2}) \tag{3.45}$$

$$b(2^{i-1}) = 8190M_1 + 3150j - 4095\alpha$$

$$8190 = (2)(3)^2(5)(91)$$

$$3150 = (2)(3)^2(5)^2(7)$$

$$4095 = (3)^2(5)(91)$$

† This is also needed for the reason given by Pohlig & Hellman in [29] (see §3.8)

The numbers $3, 5, 15$ and 63 all divide all three coefficients. This means that we can have $i = 2, 4, 6$.

So if the pairs exist, then they will be spaced with an odd number of other elements between them. We will now form the set S_1 containing $q^k \equiv 51$.

$$S_1 = \left\{ \begin{array}{cccccccccccc} 51 & 2601 & 7626 & 7967 & 1030 & 4261 & 4865 & 4426 & 4795 & 8079 & 4353 & 2826 \\ 8140 & 5590 & 565 & 224 & 7161 & 3930 & 3326 & 3765 & 3396 & 112 & 3838 & 5365 \end{array} \right\}$$

This set has a pair, the numbers 112 and 224. $\alpha = 1$, $i = 6$ and $j = 1$. So (3.45) becomes

$$b(2^6 - 1) = M(p-1) + \beta - (\frac{p-1}{2})$$

$$63b = 8190M - 945$$

Our upper bound on the number of trials is $T = \text{GCD}(63, 8190) = 63$. Note that T is small because $2^6 - 1$ is a factor of $p - 1$. In this case we can divide through by 63. $b = 130M - 15$, where $M \in [1, 63]$. By calculating q^b for each value of M, we find that $q^{2585} \equiv 112$.

$$q^k \equiv (112)^8 \equiv (q^{2585})^8 \equiv q^{4300}$$

Therefore $k = 4300$.

□

The following Theorem shows that when the primitive element chosen meets a certain property, then the chances are 50-50 that the private key associated with the public keys $\frac{p-1-2q}{2}$ or $\frac{p+1+2q}{2}$ can be easily determined. These keys should not be used.

THEOREM: If $4p$ divides $p^2 + 1 + 4q^2 - 6p - 4pq + 8q$, then

$$q^{\frac{p+1}{4}} \equiv \frac{p-1-2q}{2} \text{ or } \frac{p+1+2q}{2} \text{ modulo } p. \qquad (3.46)$$

Proof: Since $q^{\frac{p-1}{2}} \equiv -1$ (modulo p), we have $q^{\frac{p+1}{2}} \equiv p - q$, implying that

$$q^{\frac{p+1}{4}} \equiv \sqrt{Mp + p - q} \text{ or } p - \sqrt{Mp + p - q} \tag{3.47}$$

where M is such that $Mp + p - q$ is a square. If $4p$ divides $p^2 + 1 + 4q^2 - 6p - 4pq + 8q$, suppose we set $M = \frac{p^2+1+4q^2-6p-4pq+8q}{4p}$. Using this value of M we have

$$Mp + p - q = \frac{p^2 + 1 + 4q^2 - 2p - 4pq + 4q}{4} = \left(\frac{p - 1 - 2q}{2}\right)^2 \tag{3.48}$$

which means, with reference to (3.46), that

$$q^{\frac{p+1}{4}} \equiv \frac{p - 1 - 2q}{2} \text{ or } \frac{p + 1 + 2q}{2}. \tag{3.49}$$

Example 3.12: Suppose $p = 47$ and $q = 5$. Then we have

$$\frac{p^2 + 1 + 4q^2 - 6p - 4pq + 8q}{4p} = 6 = M.$$

With $M = 6$ we have $\sqrt{Mp + p - q} = \sqrt{324} = 18$ *or* -18. Thus $q^{\frac{p+1}{4}} \equiv 18$ or 29. By actual computation we find that $q^{\frac{p+1}{4}} \equiv 18$.

□

§3.8 An Analysis Using the Chinese Remainder Theorem

In [29] Pohlig and Hellman note that the Chinese remainder theorem can be used to determine the private key k, given the public key ρ. If the factors of $p-1$ are all small then the computing of a logarithm requires only on the order of $\sum p_i^2$ operations, where p_i is defined in (3.44), if some tables are computed beforehand.

$$p-1 = p_1^{n_1} p_2^{n_2} \cdots p_r^{n_r} \tag{3.44}$$

The following example illustrates how this is done.

Example 3.13: Let $p = 30059$ and $q = 7$. We want to find k such that $q^k \equiv \rho \equiv 649$. $p-1$ factors into $p-1 = (2)(7)(19)(113)$, so k mod p_i for $p_i = 2, 7, 19$ and 113 can be determined as follows:

In radix-2 representation k is

$$\begin{aligned} k =& a_0 + 2a_1 + 4a_2 + 8a_3 + 16a_4 + 32a_5 + 64a_6 + 128a_7 + 256a_8 + 512a_9 \\ &+ 1024a_{10} + 2048a_{11} + 4096a_{12} + 8192a_{13} + 16384a_{14}, \end{aligned}$$

where $a_i \in [0,1]$.

To determine a_0 one simply uses equation (3.6):

$$\begin{aligned} \rho^{\frac{p-1}{2}} &\equiv 7^{15029} \\ &\equiv 7^{8192}\ 7^{4096}\ 7^{2048}\ 7^{512}\ 7^{128} 7^{32}\ 7^{16}\ 7^{4}\ 7 \\ &\equiv (7078)(11180)(9214)(1315)(6659)(22312)(8126)(2401)(7) \\ &\equiv -1 \quad \longrightarrow \quad k \text{ is odd} \quad (a_0 = 1) \end{aligned}$$

In radix-7 representation k is

$$k = b_0 + 7b_1 + 49b_2 + 343b_3 + 2401b_4 + 16807b_5$$

where $b_i \in [0,6]$. If we define

$$\gamma_7 \equiv q^{\frac{p-1}{7}} \equiv q^{4294}$$
$$\equiv q^{4096} q^{128} q^{64} q^{6}$$
$$\equiv (11180)(6659)(18245)(117\ 649)$$
$$\equiv 7572$$

then $\gamma_7^{b_0} \equiv \rho^{\frac{p-1}{7}} \equiv 9122$. If the 7 powers of γ_7 have been pre-calculated and stored in memory, then b_0 can be immediately determined:

$$\gamma_7^2 \equiv 12671$$
$$\gamma_7^3 \equiv 26543$$
$$\gamma_7^4 \equiv 9122 \longrightarrow b_0 = 4$$
$$\gamma_7^5 \equiv 26261$$
$$\gamma_7^6 \equiv 8007$$

We continue in this way. In radix-19 representation k is

$$k = c_0 + 19c_1 + 361c_2 + 6859c_3$$

$$\gamma_{19} \equiv q^{\frac{p-1}{19}} \equiv q^{1582} \equiv 5704$$

$$\rho^{\frac{p-1}{19}} \equiv 22175 \equiv \gamma_{19}^{c_0} \longrightarrow c_0 = 14$$

In radix-113 representation k is

$$k = d_0 + 113d_1 + 12769d_2$$

$$\gamma_{113} \equiv q^{\frac{p-1}{113}} \equiv q^{266} \equiv 29828$$

$$\rho^{\frac{p-1}{113}} \equiv 19022 \equiv \gamma_{113}^{d_0} \longrightarrow d_0 = 28$$

So now we have the following:

$$k = 1 \mod 2$$

$$k = 4 \mod 7$$

$$k = 14 \mod 19$$

$$k = 28 \mod 113$$

Let us define $a_0 = \delta_1$, $b_0 = \delta_2$, $c_0 = \delta_3$ and $d_0 = \delta_4$. The Chinese remainder theorem can now be used to combine these results to get k. If for each of the prime factors p_i we define a y_i such that

$$\left(\frac{p-1}{p_i}\right) y_i \equiv 1 \mod p_i \tag{3.46}$$

and have determined these y_i beforehand, then we can write

$$k = \sum_{i=1}^{r} \left(\frac{p-1}{p_i}\right) y_i \delta_i \mod p-1. \tag{3.47}$$

In this case we have

$$15029y_1 \equiv 1 \mod 2 \longrightarrow y_1 = 1$$

$$4294y_2 \equiv 1 \mod 7 \longrightarrow y_2 = 6$$

$$1582y_3 \equiv 1 \mod 19 \longrightarrow y_3 = 18$$

$$266y_4 \equiv 1 \mod 113 \longrightarrow y_4 = 110$$

So

$$\begin{aligned} k &= 15029(1)(1) + 4294(5)(4) + 1582(4)(14) + 266(65)(28) \mod 30058 \\ &\equiv 673\,621 \mod 30058 \\ &\equiv 12345 \end{aligned}$$

□

So apparently using the Chinese remainder theorem provides a solution for k provided we precalculate and store the different powers of γ_i and the values of y_i. When $p-1 = p_1 p_2 p_3 \cdots p_r$, the number of powers of the different γ_i to be stored is

$$\sum_{i=1}^{r} p_i,$$

while the number of values of y_i to be stored is r.

The total number of operations needed to determine the powers of γ_i is $\sum_{i-1}^{r} p_i$ (where an operation is defined as one exponentiation), and the maximum number of operations needed to determine the y_i's is also $\sum_{i=1}^{r} p_i$ (where an operation is defined as one modulo $p-1$ multiplication). So there are $O(\sum_{i=1}^{r} p_i)$ operations and $O(\sum_{i=1}^{r} p_i)$ numbers to be stored.

When $p-1$ factors into small primes then the amount of computation and storage required is relatively small enough that such p should not be used in a public key distribution system. For $p = 30059$, $\sum i = 1^r p_i = 141$, which means that k is easily determined.

However, if ν is prime where $p = 2\nu + 1$, then $\sum_{i=1}^{r} p_i = 2 + \frac{p-1}{2} \approx \nu$. For the prime $p = 30203$, $\nu = \frac{p-1}{2} = 15101$ is also prime, so $\sum_{i=1}^{r} p_i = 15103$. In this case the Chinese remainder theorem method is equivalent to the exhaustive brute force method of determining k, for once (3.9) is used to see if k is even or odd, an exhaustive attack requires ν trials. So the best defence against a Chinese remainder theorem based attack is to have ν prime.

From the point of view of pre-analysing potential private keys k to see if they are suitable, one can ensure that δ_i and y_i are not too small, but caution is needed, for if the cryptanalyst can find a pattern in the choice of δ_i or y_i then his job is made easier.

CHAPTER IV

Concluding Remarks

In this thesis we examined the discrete logarithm problem from a cryptographic point of view. That is to say, instead of trying to find an algorithm to efficiently perform logarithms for all possible remainders, as other researchers have done, we have developed strategies that may easily yield a solution in some cases. This approach is important for two reasons. One is that it is important for the designer of the public key distribution system to be able to note which keys are "weak" in this sense, so as to avoid them. The other reason is that if the solution of the logarithm problem is easy enough for a non-negligibly small fraction ϵ of the remainders, then it becomes easy for the rest as well. This is because if for a certain remainder $\rho \equiv q^k$ the key k is not "weak" then the cryptanalyst can multiply q^k by a randomly chosen power of q, say q^a, where a is known to him, and try these strategies for the new remainder $\rho' \equiv q^{k+a}$. He can continue in this manner until he hits upon a solution.

In the thesis we examined the "traditional" discrete logarithm problem, i.e., the one performed over the prime field $GF(p)$. We limited ourselves to the case where $\nu = \frac{p-1}{2}$ is odd, since if it were even then at least one additional bit of the key would be easily determined.

The first two chapters of the thesis are introductory and background ones. The third chapter contains our research on the logarithm problem.

In §3.1 we defined the discrete logarithm problem, showed its cryptographic use, and reviewed the previous research done on it. As mentioned above, most of this research was aimed at finding a general solution to efficiently calculating logarithms.

Also, much of it was for logarithms in the extention fields $GF(p^m)$, not in the prime field. Sufficiently efficient algorithms have been found for the logarithm problem in these extention fields to render it inadequate for cryptographic purposes.

In the next two sections §3.2 and §3.3 we defined the sets S and Ψ, and showed how their use can sometimes yield an easy solution to the logarithm problem by forming congruence relations in S_k or Ψ_k. We ended by considering the powerful strategy of calculating the logarithms of small prime remainders. When p is known in advance, the cryptanalyst can calculate these in advance of having ρ, and when he is given a ρ of which he has to take the logarithm, all he has to do is find a related remainder that factors into small prime factors.

In §3.4 we briefly considered a lower limit on the cardinality of the set S.

In §3.5 we considered another strategy that can be used by the cryptanalyst, that of a limited "brute force" or exhaustive search for the key for each element of S. If m is small with respect to p, then the designer of the public key distribution system can easily calculate S_k and ensure that no elements are too close to 1, $p-1$, or $\frac{p-1}{2}$.

In §3.6 we briefly discussed the possibility of forming sets other than S and Ψ. Any sets that have the property that the exponents of the set elements are related in a known manner could possibly be used. It is quite possible that using these sets might show more exponents to be inadequately weak as possible keys. We did not examine this possibility, but it is a possible area of future research.

In §3.7 we considered some other results for calculating logarithms.

In §3.8 we considered the strategy of determining k modulo each prime factor of $p-1$, and combining these using the Chinese remainder theorem to get k.

The strategies developed in this thesis can be used in two different ways. One use is for the cryptanalyst, given $\rho \equiv q^k$, to try to determine k. He can try these methods of attack and, if the key was poorly chosen, will determine k. The other

use is for the cryptographer who implements the public key distribution system; he wants to avoid keys that will yield to such an attack.

Another area of possible future research could be to implement the strategy of forming congruence relations in S_k on a computer, and seeing how often and how easily elements of cryptographically large primes p yield a solution.

APPENDIX I

Sets S for $p = 683$

$$S_1 = \begin{Bmatrix} 1 \\ 682 \end{Bmatrix}$$

$$S_2 = \begin{Bmatrix} 4 & 16 & 256 & 651 & 341 & 171 & 555 & 675 & 64 & 681 \\ 679 & 667 & 427 & 32 & 342 & 512 & 128 & 8 & 619 & 2 \end{Bmatrix}$$

$$S_3 = \begin{Bmatrix} 9 & 81 & 414 & 646 & 3 \\ 674 & 602 & 269 & 37 & 680 \end{Bmatrix}$$

$$S_4 = \begin{Bmatrix} 25 & 625 & 632 & 552 & 86 & 566 & 29 & 158 & 376 & 678 \\ 658 & 58 & 51 & 131 & 597 & 117 & 654 & 525 & 307 & 5 \end{Bmatrix}$$

$$S_5 = \begin{Bmatrix} 36 & 613 & 119 & 501 & 340 & 173 & 560 & 103 & 364 & 677 \\ 647 & 70 & 564 & 182 & 343 & 510 & 123 & 580 & 319 & 6 \end{Bmatrix}$$

$$S_6 = \begin{Bmatrix} 49 & 352 & 281 & 416 & 257 & 481 & 507 & 241 & 26 & 676 \\ 634 & 331 & 402 & 267 & 426 & 202 & 176 & 442 & 657 & 7 \end{Bmatrix}$$

$$S_7 = \begin{Bmatrix} 100 & 438 & 604 & 94 & 640 & 483 & 386 & 102 & 159 & 10 \\ 583 & 245 & 79 & 589 & 43 & 200 & 297 & 581 & 524 & 673 \end{Bmatrix}$$

$$S_8 = \begin{Bmatrix} 121 & 298 & 14 & 196 & 168 & 221 & 348 & 213 & 291 & 672 \\ 562 & 385 & 669 & 487 & 515 & 462 & 335 & 470 & 392 & 11 \end{Bmatrix}$$

$$S_9 = \begin{Bmatrix} 144 & 246 & 412 & 360 & 513 & 214 & 35 & 542 & 74 & 12 \\ 539 & 437 & 271 & 323 & 170 & 469 & 648 & 141 & 609 & 671 \end{Bmatrix}$$

$$S_{10} = \begin{Bmatrix} 169 & 558 & 599 & 226 & 534 & 345 & 183 & 22 & 484 & 670 \\ 514 & 125 & 84 & 457 & 149 & 338 & 500 & 661 & 199 & 13 \end{Bmatrix}$$

$$S_{11} = \begin{Bmatrix} 225 & 83 & 59 & 66 & 258 & 313 & 300 & 527 & 431 & 668 \\ 458 & 600 & 624 & 617 & 425 & 370 & 383 & 156 & 252 & 15 \end{Bmatrix}$$

$$S_{12} = \begin{Bmatrix} 289 & 195 & 460 & 553 & 508 & 573 & 489 & 71 & 260 & 666 \\ 394 & 488 & 223 & 130 & 175 & 110 & 194 & 612 & 423 & 17 \end{Bmatrix}$$

$$S_{13} = \begin{Bmatrix} 324 & 477 & 90 & 587 & 337 & 191 & 282 & 296 & 192 & 665 \\ 359 & 206 & 593 & 96 & 346 & 492 & 401 & 387 & 491 & 18 \end{Bmatrix}$$

$$S_{14} = \begin{Bmatrix} 361 & 551 & 349 & 227 & 304 & 211 & 126 & 167 & 569 & 19 \\ 322 & 132 & 334 & 456 & 379 & 472 & 557 & 516 & 114 & 664 \end{Bmatrix}$$

$$S_{15} = \begin{Bmatrix} 400 & 178 & 266 & 407 & 363 & 633 & 451 & 550 & 614 & 663 \\ 283 & 505 & 417 & 276 & 320 & 50 & 232 & 133 & 69 & 20 \end{Bmatrix}$$

$$S_{16} = \begin{Bmatrix} 441 & 509 & 224 & 317 & 88 & 231 & 87 & 56 & 404 & 662 \\ 242 & 174 & 459 & 366 & 595 & 452 & 596 & 627 & 279 & 21 \end{Bmatrix}$$

$$S_{17} = \begin{Bmatrix} 529 & 494 & 205 & 362 & 591 & 268 & 109 & 270 & 502 & 660 \\ 154 & 189 & 478 & 321 & 92 & 415 & 574 & 413 & 181 & 23 \end{Bmatrix}$$

$$S_{18} = \begin{Bmatrix} 576 & 521 & 290 & 91 & 85 & 395 & 301 & 445 & 638 & 659 \\ 107 & 162 & 393 & 592 & 598 & 288 & 382 & 238 & 45 & 24 \end{Bmatrix}$$

$$S_{19} = \begin{Bmatrix} 46 & 67 & 391 & 572 & 27 \\ 637 & 616 & 292 & 111 & 656 \end{Bmatrix}$$

$$S_{20} = \begin{Bmatrix} 101 & 639 & 570 & 475 & 235 & 585 & 42 & 398 & 631 & 655 \\ 582 & 44 & 113 & 208 & 448 & 98 & 641 & 285 & 52 & 28 \end{Bmatrix}$$

$$S_{21} = \begin{Bmatrix} 217 & 645 & 78 & 620 & 554 & 249 & 531 & 565 & 264 & 30 \\ 466 & 38 & 605 & 63 & 129 & 434 & 152 & 118 & 419 & 653 \end{Bmatrix}$$

$$S_{22} = \begin{Bmatrix} 278 & 105 & 97 & 530 & 187 & 136 & 55 & 293 & 474 & 652 \\ 405 & 578 & 586 & 153 & 496 & 547 & 628 & 390 & 209 & 31 \end{Bmatrix}$$

$$S_{23} = \begin{Bmatrix} 406 & 233 & 332 & 261 & 504 & 623 & 185 & 75 & 161 & 650 \\ 277 & 450 & 351 & 422 & 179 & 60 & 498 & 608 & 522 & 33 \end{Bmatrix}$$

$$S_{24} = \begin{Bmatrix} 473 & 388 & 284 & 62 & 429 & 314 & 244 & 115 & 248 & 34 \\ 210 & 295 & 399 & 621 & 254 & 369 & 439 & 568 & 435 & 649 \end{Bmatrix}$$

$$S_{25} = \begin{Bmatrix} 155 & 120 & 57 & 517 & 236 & 373 & 480 & 229 & 533 & 644 \\ 528 & 563 & 626 & 166 & 447 & 310 & 203 & 454 & 150 & 39 \end{Bmatrix}$$

$$S_{26} = \begin{Bmatrix} 234 & 116 & 479 & 636 & 160 & 329 & 327 & 381 & 385 & 40 \\ 449 & 567 & 204 & 47 & 523 & 354 & 356 & 302 & 318 & 643 \end{Bmatrix}$$

$$S_{27} = \begin{Bmatrix} 315 & 190 & 584 & 239 & 432 & 165 & 588 & 146 & 143 & 642 \\ 368 & 493 & 99 & 444 & 251 & 518 & 95 & 537 & 540 & 41 \end{Bmatrix}$$

$$S_{28} = \begin{Bmatrix} 255 & 140 & 476 & 503 & 299 & 611 & 403 & 538 & 535 & 48 \\ 428 & 543 & 207 & 180 & 384 & 72 & 280 & 145 & 148 & 635 \end{Bmatrix}$$

$$S_{29} = \begin{Bmatrix} 77 & 465 & 397 & 519 & 259 & 147 & 436 & 222 & 108 & 53 \\ 606 & 218 & 286 & 164 & 424 & 536 & 247 & 461 & 575 & 630 \end{Bmatrix}$$

$$S_{30} = \begin{Bmatrix} 184 & 389 & 378 & 137 & 328 & 353 & 303 & 287 & 409 & 629 \\ 499 & 294 & 305 & 546 & 355 & 330 & 380 & 396 & 274 & 54 \end{Bmatrix}$$

$$S_{31} = \begin{Bmatrix} 306 & 65 & 127 & 420 & 186 & 446 & 163 & 615 & 526 & 61 \\ 377 & 618 & 556 & 263 & 497 & 237 & 520 & 68 & 157 & 622 \end{Bmatrix}$$

$$S_{32} = \begin{Bmatrix} 548 & 467 & 212 & 549 & 198 & 273 & 82 & 577 & 308 & 610 \\ 135 & 216 & 471 & 134 & 485 & 410 & 601 & 106 & 375 & 73 \end{Bmatrix}$$

$$S_{33} = \begin{Bmatrix} 312 & 358 & 443 & 228 & 76 \\ 371 & 325 & 240 & 455 & 607 \end{Bmatrix}$$

$$S_{34} = \begin{Bmatrix} 253 & 490 & 367 & 138 & 603 \\ 430 & 193 & 316 & 545 & 80 \end{Bmatrix}$$

$$S_{35} = \begin{Bmatrix} 408 & 495 & 511 & 215 & 464 & 151 & 262 & 344 & 177 & 594 \\ 275 & 188 & 172 & 468 & 219 & 532 & 421 & 339 & 506 & 89 \end{Bmatrix}$$

$$S_{36} = \begin{Bmatrix} 453 & 309 & 544 & 197 & 561 & 541 & 357 & 411 & 220 & 590 \\ 230 & 374 & 139 & 486 & 122 & 142 & 326 & 272 & 463 & 93 \end{Bmatrix}$$

$$S_{37} = \begin{Bmatrix} 571 & 250 & 347 & 201 & 104 \\ 112 & 433 & 336 & 482 & 579 \end{Bmatrix}$$

$$S_{38} = \begin{Bmatrix} 350 & 243 & 311 & 418 & 559 \\ 333 & 440 & 372 & 265 & 124 \end{Bmatrix}$$

APPENDIX II

Set S_1 for Example 3.6

Top half:

1660	16625	17324	2726	4421	12121	13301	19783	12891	3932
17251	1779	11056	10707	4851	11069	13474	12742	9851	7358
17144	4321	12842	4529	2065	11702	1494	14213	4353	5552
19346	4998	14235	19799	15446	10341	12796	5307	4001	15259
6360	20111	15002	1837	17314	2245	14316	12872	2763	4044
15295	7566	8245	698	19223	1862	8054	680	14766	16651
7579	1660	8755	2876	10494	6072	420	13624	8042	10998
13436	7512	7913	7850	11784	14728	14964	2561	6987	5716
15721	15179	12960	17462	1302	6403	19551	2859	14734	8613
18954	7484	15312	19210	10908	15555	11848	1151	2246	18807
11348	941	10560	12040	9572	643	6509	4627	4085	2685
6387	18381	19573	9013	8945	8621	14397	19067	10640	19239
6844	1542	17512	160	5253	3477	3411	16784	18788	9188
19988	6799	18364	5318	19141	9799	2908	12459	19765	13172
2715	5611	6512	2996	2989	1788	2465	12819	4389	15059
6166	11360	8926	14971	8636	8741	2096	18611	2340	2257
7299	6955	7206	892	2131	3780	4806	3791	6699	11466
7189	341	14546	18010	8573	2965	1321	15546	16797	7707
4956	3107	8971	6456	9280	9896	705	8697	8010	6009
12503	19355	7408	2605	10474	13999	10044	1510	1236	1671
4702	11962	9340	8011	1683	4256	4706	8900	19476	5802
9266	14763	9452	16974	3156	10653	11190	662	10957	8549
19324	8832	14173	8345	11591	40				

Bottom half:

18747	3722	3023	17621	15926	8226	7046	564	7456	16415
3096	18568	9291	9640	15496	9278	6873	7605	10496	12989
3203	16026	7505	15818	18282	8645	18853	6134	15994	14795
1001	15349	6112	548	4901	10006	7551	15040	16346	5088
13987	236	5345	18510	3033	118102	6031	7475	17584	16303
5052	12781	12102	19649	1124	18485	12293	19667	5581	3696
12768	18687	11592	17471	9853	14275	19927	6723	12305	9349
6911	12835	12434	12497	8563	5619	5383	17786	13360	14631
4626	5168	7387	2885	19045	13944	796	17488	5613	11734
1393	12863	5035	1137	9439	4792	8499	19196	18101	1540
8999	19406	9787	8307	10775	19704	13838	15720	16262	17662
13960	1966	774	11334	11402	11726	5950	1280	9707	1108
13503	18805	2835	20187	15094	16870	16936	3563	1559	11159
359	13548	1983	15029	1206	10548	17439	7888	582	7175
17632	14736	13835	17351	17358	18559	17882	7528	15958	5288
14181	8987	11421	5376	11711	11606	18251	1736	18007	18090
13048	13392	13141	19455	18216	16567	15541	16556	13648	8881
13158	20006	5801	2337	11774	17382	19026	4801	3550	12640
15391	17240	11376	13891	11067	10451	19642	11650	12337	14338
7844	992	12939	17742	9873	6348	10303	18837	19111	18676
15645	8385	11007	12336	18664	16091	15641	11447	871	14545
11081	5584	10895	3373	17191	9694	9157	19685	9390	11798
1023	11515	6174	12002	8756	20307				

BIBLIOGRAPHY

[1] L.M. Adleman,"A subexponential algorithm for the discrete logarithm problem with applications to cryptography", *Proc. IEEE 20th Annual Symposium on Foundations of Computer Science*, pp. 55-60, 1979.

[2] L.M. Adleman and R.L. Rivest, "The use of public key cryptography in communication system design", *IEEE Communications Magazine*, vol.16, No.6, pp.20-23, Nov.1978.

[3] T.M. Apostal, *Introduction to Analytic Number Theory*, Springer-Verlag, New York, 1976.

[4] S. Berkovits, J.Kowalchuk and B. Schanning, "Implementing Public Key Scheme", *IEEE Communications Magazine*, 17, pp.2-3, May 1979.

[5] I.F.Blake, R.Fuji-Hara, R.C.Mullin and S.A.Vanstone, "Computing logarithms in finite fields of characteristic two", *SIAM J. Alg. Disc. Meth.*, vol.5, No.2, pp.276-285, June 1984.

[6] M.Bouniakowsky, "Sur les congruences binômes exponentielles à base 3 et sur plusieurs nouveaux théorèmes relatifs aux résidus et aux racines primitives", *Bulletin de l'Académie Impériale des Sciences de Saint-Pétersbourg*, **14**, pp.356-381, 1870.

[7] E.F.Brickell and J.H.Moore, "Some remarks on the Herlestam-Johannesson algorithm for computing logarithms over $GF(2^p)$", *Advances in Cryptology: Proceedings of Crypto 82*, pp.15-19, Plenum Press, New York, 1983.

[8] G.C. Clark and J.B. Cain, *Error-Correction Coding for Digital Communications*, Plenum Press, New York, 1981.

[9] D.Coppersmith, "Fast evaluation of logarithms in fields of characteristic two", *IEEE Trans. Inform. Theory*, vol. IT-30, pp.587-594, July 1984.

[10] *Data Encryption Standard*, Federal Information Processing Standard (FIPS) Publication 46, National Bureau of Standards, U.S. Department of Commerce, Washington, DC, January 1977.

[11] R.M. Davis, "The data encryption standard in perspective", *IEEE Communications Magazine*, vol.16, No.6, pp.5-9, Nov.1978.

[12] W. Diffie, "Cryptographic technology: fifteen-year forecast", in *Secure Communications and Asymmetric Cryptosystems*, pp.301-327, Gustavus J. Simmons, Editor, Westview Press, Boulder, Colorado, 1982.

[13] W.Diffie and M.E.Hellman, "New directions in cryptography", *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 644-654, Nov. 1976.

[14] W.Diffie and M.E.Hellman, "Exhaustive Cryptanalysis of the NBS Data Encryption Standard", *IEEE Computer Magazine*, pp. 74-84, June 1977.

[15] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms", *IEEE Trans. Inform. Theory*, vol. IT-31, pp. 469-472, July 1985.

[16] T. ElGamal, "A subexponential-time algorithm for computing discrete logarithms over $GF(p^2)$", *IEEE Trans. Inform. Theory*, vol. IT-31, pp. 473-481, July 1985.

[17] M.E. Hellman, "An overview of public key cryptography", *IEEE Communications Magazine*, vol.16, No.6, pp.24-32, Nov.1978.

[18] M.E.Hellman and J.M.Reyneri, "Fast computation of discrete logarithms in $GF(q)$", *Advances in Cryptology: Proceedings of Crypto 82*, pp.3-13, Plenum Press, New York, 1983.

[19] T.Herlestam and R.Johannesson, "On computing logarithms over $GF(2^p)$", *BIT* **21**, pp. 326-334.

[20] D.Kahn, *the Codebreakers: The Story of Secret Writing*, Macmillan, New York, 1967.

[21] D.E. Knuth, *the Art of Computer Programming, Volume 2/ Seminumerical Algorithms*, Addison-Wesley, Reading, Mass., 1969

[22] D.E. Knuth, *the Art of Computer Programming, Volume 3/ Sorting and Searching*, Addison-Wesley, Reading, Mass., 1973

[23] G. Kolata "NSA to provide secret codes", *Science*, the American Association for the Advancement of Science, Vol.230, No.4721,pp.45-46, 4 Oct. 1985.

[24] A.G.Konheim, *Cryptography: A Primer*, John Wiley & Sons, New York, 1981.

[25] W. Kozaczuk, *Enigma: How the German Machine Cipher Was Broken, and How It Was Read by the Allies in World War Two*, Arms and Armour Press, London, 1984.

[26] R. Lewin, *The American Magic: Codes, Ciphers and the Defeat of Japan*, Farrar Straus Giroux, New York, 1982.

[27] C.H.Meyer and S.M.Matyas, *Cryptography: A New Dimension in Computer Data Security—a Guide for the Design and Implementation of Secure Systems*, John Wiley & Sons, New York, 1982.

[28] R. Morris, "The data encryption standard—retrospective and prospects", *IEEE Communications Magazine*, vol.16, No.6, pp.11-14, Nov.1978.

[29] S.C.Pohlig and M.E.Hellman, "An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance", *IEEE Trans. Inform. Theory*, vol. IT-24, pp.106-110, Jan. 1978.

[30] R.L. Rivest, A. Shamir, L.M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", *Communications of the ACM*, **21**, No.2, pp.120-126, 1978.

[31] C.E.Shannon, "Communication theory of secrecy systems", *Bell System Technical Journal*, **28**, pp.656-715, 1949.

[32] G.J. Simmons, "Symmetric and assymmetric encryption", in *Secure Communications and Asymmetric Cryptosystems*, pp.241-298, Gustavus J. Simmons, Editor, Westview Press, Boulder, Colorado, 1982.

[33] United States Senate Select Committee on Intelligence, "Unclassified Summary: Involvement of NSA in the Development of the Data Encryption Standard", *IEEE Communications Magazine*, vol.16, No.6, pp.53-55, Nov.1978.

[34] A.L.Wells, "A polynomial form for logarithms modulo a prime", *IEEE Trans. Inform. Theory*, vol. IT-30, pp.845-846, Nov. 1984.

[35] A.E. Western and J.C.P. Miller, *Tables of Indices and Primitive Roots*, Royal Society Mathematical Tables, Vol. 9, Cambridge University Press, Cambridge, 1968.

[36] H.C. Williams, "Computationally "hard" problems as a source for cryptosystems", in *Secure Communications and Asymmetric Cryptosystems*, pp.11-39, Gustavus J. Simmons, Editor, Westview Press, Boulder, Colorado, 1982.

[37] "A new method for realizing public-key cryptosystem", *Cryptologia*, Vol.9, No.4 pp.360-?, Oct. 1985.