Information Security:
An Introduction to Cryptography

University of Arizona

Jeffrey Sorrentino

Information Security:
An Introduction to Cryptography

Abstract

Throughout time, all forms of communication have been subject to some sort of eavesdropping, be it interception of a telegram, the over-hearing of a conversation, or the active attacks on digital communications and networks. The constant search to improve information security has historical significance in respect to both the government and the private sector. While this paper's main focus is on encryption and its applications, origins, and algorithms, the innermost technical workings of current algorithms have been omitted.

We will begin by discussing some known origins of cryptography, and then quickly move to more modern concepts of peer review and public algorithms, such as the Data Encryption Standard (DES) and the Advanced Encryption Standard (AES). The process by which algorithms are analyzed and selected as Encryption Standards are also within the focus of this paper.

Introduction

The need for encryption arises from the desire to keep information private, be it governmental communications, banking, or simply sending confidential messages to a colleague or friend. Securing digital or written data is an ever-changing, ongoing battle between those who desire secure communications, and those who would like to compromise them. Throughout history, encryption has played an important role within just about every major government since the rule of Caesar.

The Internet, which is currently the most common medium for the transfer of information, has brought with it prying eyes, and thus a reason for cryptography in almost all of its applications. SSL for browser communications, PGP for secure messaging, and SSH for remote connectivity are just a few examples of current cryptographic applications in use on the Internet today.

Origins

One of the first recorded cryptographic algorithms is called the Caesar Shift. This simple cipher is a type of letter substitution taking the Standard English twenty-six character alphabet and shifting the characters by some number *n* less than 26. Encryption is then performed by adding *n* to the index of the current plain text character modulo 26 to get the cipher character's index.

Example:

If $n = 7$, we shift the entire alphabet by 7 characters mod 26 to produce the key:

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

So, the plain text "INSECURE" would produce the ciphertext "PUZLJBYL",
and the ciphertext "ZLJBYL" would be decrypted to "SECURE".

The person receiving the message need only know what *n* and the original alphabet is to decrypt the message. This algorithm used is, by itself, trivial to break. Simple cryptanalysis **[1]** by a person or computer can be accomplished quite quickly using the fact that certain short word patterns occur more commonly than others. If common words like; "THE", "AND", and "AS", are present, then one would only need to map one or two letters to a key to be able to decrypt the entire message.

During World War II, both the Americans and the Germans had developed impressive cryptographic rotor machines. The rotor machine, being the first mechanical encryption device intended to automate cryptography **[2],** was easily the most important device of World War II, and remained dominant until at least the nineteen fifties **[3]**. A rotor machine is made up of a typewriter-style keypad and multiple rotors, with a complete desired alphabet imprinted around each rotor much like the old-fashioned typewriter balls. Each rotor is linked to the next, such that for each letter selection on a given rotor, a pseudo-random letter substitution is made of the alphabet onto the next rotor.

Example:

Each row represents a rotor in a four-rotor machine. So if "**D**" is selected from rotor 1, then the encryption sequence is **D->K->F->J,** after the 4th rotor is applied.

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

The true strength of the algorithm lies within the fact that, after each sequential letter substitution, the alignments of the rotors change to create a different substitution pattern. Because all the rotors rotate at different rates, the period for an *n*-rotor machine is 26^n **[2]**. Some early machines have a standard pattern by which the substitution permutation is performed and thus, once the machines were compromised, so were all past and future communications. The better of these machines accept a pass code or pass card that seeds the pseudo-random permutation sequence. The most effective rotor machine built was the German Enigma **[2]**. The Enigma was a very complex machine with three rotors chosen from five, and a reflective rotor that meant each rotor would operate on each plaintext character twice **[2]**. This machine was cracked by a team of Polish cryptographers and was continually studied by the British throughout the remainder of the war **[2]**.

Government agencies are historically the major contributors to the field of cryptography. This is partially due to their seemingly open-ended budgets, and their legal ability to classify intellectual properties as secret. As a result, the private sector has not had, until recently, strong algorithms and applications at their immediate disposal. Around 1974, IBM Research proposed a public encryption algorithm at the request of the National Bureau of Standards (NBS, now the

National Institute of Standards and Technology, NIST), which would soon become DES, the first public governmental Data Encryption Standard **[4]**.

Basic Concepts

Although many advances have been made recently by the private sector, the most important concept for improvement and algorithmic strength is peer review. Relying solely upon the premise that no one knows how an algorithm operates is a short-lived notion of security, and a recipe for disaster. If an algorithm has been created once, then it is reasonable to assume that someone else is not far behind. True security lies within public algorithms that have been subjected to, and withstood, years of scrutiny by the experts in the field.

The primary purpose of cryptography is to take a message or piece of data and render it unreadable by an intercepting party who does not possess the appropriate key. Although this is the basic idea for both digital and written communications, the remainder of our discussion will be limited to the techniques used for digital cryptography. Keys, obfuscation, and one-way functions are the basis of all cryptography. Keys can be composed from any of several different entities: pass phrases, prime numbers, random bits, and other events. Most of the time, however, a key is made up of some combination of these.

Compression is a very important concept within encryption, and in fact, strong encryption cannot be accomplished without compression. Successful cryptanalysis relies on the exploitation of redundancy in the plaintext. Since the basis of compression is to remove redundancy, this becomes a logical initial step for all encryption **[5]**. There are many complex compression techniques available today, however, the necessity remains only to remove redundancy. The following is a description of two simple compression techniques.

Null Suppression and Bit Mapping are some of the earliest and simplest forms of compression. Null Suppression is a pretty straightforward idea. As the name implies, one compresses the data by suppressing nulls, zeros, or some form of blank space, depending on the implementation. This technique takes advantage of the fact that most files can contain a large number of blanks and zeros. This method is applicable to datasets containing fixed-sized elements, such as words or bytes.

Example #1:

> Assume we are dealing with a dataset containing a large number of 0's, so that implementing Null Suppression is worthwhile, and each item in the collection is a fixed size.

Original: | data | 0 | 0 | 0 | 0 | 0 | 0 | data | 0 | 0 | 0 | data | 0 | 0 | 0 | 0 | 0 |

Compressed: | 1000000100010000 | data | data | data |

[6]

To implement the Bitmap, we prepended a key to the front of the collection. We then dropped all of the zeros, and turned on 1's corresponding to the data in the addresses to follow, and 0's where the groups of zeros originally were within the collection. By suppressing the zeros using the Bitmap, we went from size = 16 to size = 4. Null Suppression can also be implemented using the Run-length technique, where a special character is inserted along with an integer to replace a run of nulls.

Example #2:

Original Data: A10000X02500000N00000COST

Compressed Data: A1#4X025#5N#5COST

[6]

The choice of characters to use for replacement is quite trivial. Any character that is not commonly used is acceptable. If, while replacing nulls in a collection, the chosen character is found in the stream, it is simply replaced with a double instance of itself.

Example #3:

Original Data: A1#400000000034

Compressed Data: A1##4#934

Other stronger compression techniques exist and should be implemented, depending upon the medium to be encrypted. However, further discussion of this approach is outside the scope of this paper.

A basic algorithm for the function of encrypting bits of data consists of using the Exclusive OR operation (XOR). The idea is to take the bits of the plaintext and XOR them with some sort of key to obtain the ciphertext. To convert the ciphertext back to plaintext, you need only XOR the ciphertext with the original key. Single key encryption algorithms such as these are considered Symmetric Key Algorithms, because these algorithms use the same key for both encryption and decryption.

Simple implementations using only the XOR operation tend to be somewhat insecure, but the basic idea still remains quite useful.

Current Techniques

Currently, there are many algorithms used in mainstream cryptography. Some are regarded as being much more sound and secure than others. The only algorithm that has ever been proven one hundred percent secure is the One-Time Pad **[7]**.

One-Time Pad

The One-Time Pad, sometimes referred to as the one-time tape, is the simplest algorithm to implement, but it is also the least practical. One-Time Pads rely solely on a truly random set of bits, the same length as the plaintext or longer, which is used as a key. An Exclusive OR operation is then performed on the plaintext and the one-time pad to produce the ciphertext **[7]**. In order to recreate the plaintext from the ciphertext, simply XOR the ciphertext with the original pad. Since every plaintext message is equally possible, there is no way for the cryptanalyst to determine which plain text message is the correct one **[8]**.

Example:

Encryption		**Decryption**	
Plaintext:	01100101	Cipher text:	11011000
One-time pad:	10111101	One-time pad:	10111101
	XOR		XOR
Ciphertext:	11011000	Plaintext:	01100101

The security inherent in this algorithm lies within its protocol. Each one-time pad, as implied by its name, can be used no more than once. The sender and the receiver must both have a copy of the pad used to encrypt the message. If the same one-time pad is used on more then one message, then a somewhat complicated cryptanalysis can be done on both the ciphertexts to recreate the pad used to encrypt them. Originally, these pads were distributed as books. When a new message arrived, one would just turn to the next page to decrypt it. Data tapes and Compact Disks have made this algorithm somewhat more feasible, but there still remains the problem of how to distribute the new pads quickly, efficiently and securely.

Symmetric Key Algorithms

Symmetric key algorithms are a partial solution to the problems which the one-time pad presents. Symmetric key encryption also uses only one key to both encrypt and decrypt the message. Unlike the one-time pad, where the key must be the same length as the plaintext, the

symmetric key is usually a fraction of the size of the message to be encrypted, usually on the order of 56-bits to 4096-bits in length. For serious security, 4096-bit keys should be used, giving a search space of 2^{4096} **[9]**. Symmetric key algorithms are sometimes referred to as Block Ciphers, since only a single block of plaintext, equal in length to the size of the key, is encrypted at once. Although the Exclusive OR operation is almost always used in Block Cipher algorithms, many other techniques, such as bit shifts and rotations, are used to help compensate for using the same key over multiple bits of plaintext.

Data Encryption Standard (DES)

The Data Encryption Standard (DES) was the first publicly available cryptographic algorithm that the government not only released to the general public, but also adopted as their encryption standard for unclassified data. When IBM Research proposed what would become DES around 1974, it was in almost the same form as it is today. IBM submitted the algorithm to the NSA for help with the S-Boxes, or Substitution-Boxes, and a review of its general performance and security **[10]**. DES is a block-cipher symmetric key algorithm which operates on a block size of 64-bits **[10]**. DES uses eight S-Boxes, which perform substitution operations on the XORed bits. These S-Boxes each have a 6-bit input and a 4-bit output **[10]**. The main weakness of the algorithm is that it is limited to using 56-bit keys. The fixed length key, however, makes it easy to implement the algorithm in hardware, which accelerates the encryption process. This same advantage, however, has led to hardware "cracking chips" which can very quickly break the 56-bit keys. Many variations of DES have been developed since its original release to help improve upon its weaknesses. The most lasting and accepted variation is Triple-DES, so named for its triple encryption using the DES algorithm. The concept of triple encryption was introduced by Tuchman **[11]**. Triple encryption uses two different keys on one block, three times. First, encrypt with the first key, then decrypt with the second key, and encrypt again with the first key, this operation is sometimes called encrypt-decrypt-encrypt (EDE) **[11]**.

Advanced Encryption Standard (AES)

After the thirty-plus years that DES has been in service, and after many successful attempts at cracking DES, the U.S-Government decided it was time to accept a new encryption standard to use for business and non-top secret documents. AES is a symmetric key algorithm like DES, and is intended to fully replace its predecessor. On January 2, 1997, the NIST announced the initiation of an effort to develop the AES, and made a formal call for algorithms on September 12, 1997 **[12]**. The AES search was public, where as the request for DES was only

made to a few research facilities. Allowing cryptographers from all over the world to fully submit and evaluate the proposed algorithms is an incredible demonstration of the importance that peer review holds in cryptography. The winning algorithm, originally called Rijndael, was created by two cryptographers from Belgium, Dr. Joan Daemen and Dr. Vincent Rijmen **[12]**.

Public Key Cryptography

Symmetric key algorithms pose the problem of secure key distribution. How to get each participating party a copy of the encryption key, and ensure that it has not been compromised, is an ongoing concern. Diffie and Hellman, and independently, Merkle, developed systems to allow key exchanges to take place in the open, on non-secure channels **[13]**. Examples of these systems are the Diffie/Hellman key exchange, and algorithms introduces by Ronald Rivest, Adi Shamir, and Leonard Adleman (RSA). Systems of this type are called public key cryptosystems **[13]**. At the heart of these systems are specialized encryption keys. Each user has two keys, a public key and a private key, which are inverse functions of each other **[13]**. In most cases, the public key is used for encryption, while the private key is reserved for decryption and Digital Signatures. The main advantage to this type of system is that a user's public key can be openly distributed without the repercussions of interception.

The protocol:

> Alice and Bob would like to send each other secure messages. Alice and Bob can openly email each other their public keys. When Alice wants to send Bob a secure message *M*, she simply encrypts *M* with Bob's public key.
> $E(K_1,M) = C$, where K_1 is Bob's public key and *C* is the ciphertext of *M*.
> Bob then receives the encrypted message and decrypts it with his private key.
> $D(K_2,C) = M$, where K_2 is Bob's private key and *M* is the original message in plaintext.

The downfall to a public key cryptosystem is that the encryption and decryption operations are incredibly expensive compared to that of symmetric key algorithms. The answer to this problem is to use both - symmetric algorithms and keys, in conjunction with public key cryptography.

Here is how it works:

> Alice encrypts *M* with a symmetric key algorithm like DES; she then encrypts only the symmetric key used to encrypt *M* with Bob's public key. The encrypted key is then

appended to the encrypted message and sent to Bob. Bob then decrypts the symmetric key with his private key, and follows to decrypt *M* with the symmetric key algorithm.

Systems like these have been in use for quit some time, and have been proven very strong. An example of this system is the application Pretty Good Privacy (PGP), created by Phil Zimmerman at the Massachusetts Institute of Technology (MIT) **[14]**.

Digital Signatures

Digital signatures are used to verify authenticity of electronic documents and transfers. Using Public Key cryptography along with one-way hash functions, digital signatures present greater authenticity than a normal hand-signed document. A signature can be used in conjunction with encryption, or as a standalone method of verification.

The signature algorithm is as follows: if John would like to sign a message and send it to Jane, he would first run a one-way hash function over the entire document to get the document's hash. The next step is to encrypt the hash of the document using his private key. Encrypting only the hash allows for a small addition to the size of the document, and since public/private key algorithms are very expensive, limiting the amount of data to be encrypted is very important. The encrypted hash is then appended to the end of the message that Jane is to receive. Once received, Jane decrypts the hash with John's public key and verifies that the hash is correct.

This accomplishes two things. The first is that this assures, to Jane, that John was the true author since, if she is able to decrypt the hash, then it was truly John's private key that signed it. The second is that since the original hash was encrypted, she can verify that no one has altered the document since it left John. If the message had been altered, then John's hash and Jane's hash would not match. The real power behind this scheme is the one-way hash function. There are many proven one-way hash function in use today, but the most common is the Secure Hash Algorithm (SHA).

Secure Hash Algorithm

The hash algorithm SHA was originally developed as a joint effort between NSA and NIST for use with the Digital Signature Standard **[15]**. SHA takes any message of length $< 2^{64}$ bits as input and produces a 160-bit hash of the message **[15]**. The basic steps of SHA include padding the message to be a multiple of 512-bits long by appending a one-bit and then as many zero-bits as are necessary, and then running each 512-bit block through four functions at twenty

rounds each. These functions consist of four constants and bit operations such as OR, AND, XOR, and NOT.

Example:

$$f(X,Y,Z) = (X \wedge Y) \vee ((\neg X) \wedge Z)$$
$$f(X,Y,Z) = (X \oplus Y \oplus Z)$$
$$f(X,Y,Z) = (X \wedge Y) \vee (X \wedge Z) \vee (Y \wedge Z)$$
$$f(X,Y,Z) = (X \oplus Y \oplus Z)$$

For a full description of SHA and its protocol see **[15].**

Political and Governmental Implications

The National Security Agency (NSA) holds the upper hand in cryptography and cryptanalysis in the United States. NSA, which only recently acknowledged its own existence publicly, has been developing cryptography for many years before the private sector became involved. NSA not only does research developing algorithms for securing governmental information, but also designs cryptanalytic techniques in order to listen in on non-U.S. communications **[16]**. Although DES is an open algorithm, it is widely speculated that NSA made changes to allow for easier cryptanalysis of messages encoded with DES.

Due to the nature of public cryptography, the U.S. government, acting on the advice of agencies such as NSA, has enacted strict export laws limiting which countries may have access to view and implement certain algorithms. A restriction on key size for international applications has also been imposed.

Conclusion

The development of cryptography and information security is an ever-changing, ongoing battle between those who desire privacy and security, and those who would like to compromise them. There are many considerations involved when deciding which form of cryptography to implement, if any, and when to implement it. Budget, level of risk, and overall goals are important questions to consider when planning a cryptographic system. It is also important to thoroughly assess the budget and perseverance of the adversary, since those factors can affect the outcome.

Today, many algorithms and applications are available for the private sector to choose from. Some of them are quite good and have been field-tested for many years, while the security

and strengths of others rely solely upon secrecy, and the (false) hope that no one else understands the algorithm.

Security in information transfer is of great political and social importance. Cryptography allows us to transfer money from our bank accounts, shop with credit cards, send private e-mails, and many other daily events. Although cryptography in the private sector is a necessity, serious questions remain: Should everyone be allowed to use it? What if it falls into the wrong hands and is used for nefarious purposes instead of good? Our government is constantly wrestling with these questions, and one can only expect the debate to intensify as a result of the events of September 11, 2001.

Endnotes

[1] Cryptanalysis refers to the science and field of code breaking.

[2] SCHNEIR, BRUCE, Applied Cryptography (New York: Wiley, 1996), 12-13.

[3] DIFFIE, WHITFIELD AND HELLMAN, MARTIN, Privacy and Authentication: An Introduction to Cryptography (Proceedings of the IEEE, Vol. 67, No. 3, March 1979), 409.

[4] SCHNEIR, BRUCE, Applied Cryptography (New York: Wiley, 1996), 265-266.

[5] SCHNEIR, BRUCE, Applied Cryptography (New York: Wiley, 1996), 226.

[6] ARONSON, J, Data Compression: A Comparison of Methods (NBS Special Publications 500-9, US Department of Commerce, National Bureau of Standards, 1977).

[7] DIFFIE, WHITFIELD AND HELLMAN, MARTIN, Privacy and Authentication: An Introduction to Cryptography (Proceedings of the IEEE, Vol. 67, No. 3, March 1979), 399-400.

[8] SCHNEIR, BRUCE, Applied Cryptography (New York: Wiley, 1996), 16.

[9] TANENBAUM, ANDREW, Modern Operating Systems (New Jersey: Prentice-Hall, 2001), 588.

[10] SCHNEIR, BRUCE, Applied Cryptography (New York: Wiley, 1996), 265-280.

[11] SCHNEIR, BRUCE, Applied Cryptography (New York: Wiley, 1996), 356-357.

[12] JAMES NECHVATAL, ELAINE BARKER, LAWRENCE BASSHAM, WILLIAM BURR, MORRIS DWORKIN, JAMES FOTI, EDWARD ROBACK, Report on the Development of the Advanced Encryption Standard, (NIST, October 2, 2000), 7.

[13] DIFFIE, WHITFIELD AND HELLMAN, MARTIN, Privacy and Authentication: An Introduction to Cryptography (Proceedings of the IEEE, Vol. 67, No. 3, March 1979), 400-402.

[14] PGP is offered as completely public cryptosystem. For more information on PGP, visit www.pgp.com. For international and freeware users; visit www.pgpi.com.

[15] SCHNEIR, BRUCE, Applied Cryptography (New York: Wiley, 1996), 442-444.

[16] SCHNEIR, BRUCE, Applied Cryptography (New York: Wiley, 1996), 598.

Bibliography

SCHNEIR, BRUCE, Applied Cryptography (New York: Wiley, 1996).

ARONSON, J, Data Compression: A Comparison of Methods (NBS Special Publications500-9, US Department of Commerce, National Bureau of Standards, 1977).

DIFFIE, WHITFIELD AND HELLMAN, MARTIN, Privacy and Authentication: An Introduction to Cryptography (Proceedings of the IEEE, Vol. 67, No. 3, March 1979).

TANENBAUM, ANDREW, Modern Operating Systems (New Jersey: Prentice-Hall, 2001).

JAMES NECHVATAL, ELAINE BARKER, LAWRENCE BASSHAM, WILLIAM BURR, MORRIS DWORKIN, JAMES FOTI, EDWARD ROBACK, Report on the Development of the Advanced Encryption Standard, (NIST, October 2, 2000).