Defence Research and Development Canada
Recherche et développement pour la défense Canada

DEFENCE R&D DÉFENSE

Introduction to cryptography

F. Paquin
Nurun

M. Salois
DRDC Valcartier

Defence R&D Canada – Valcartier
Technical Memorandum
DRDC Valcartier TM 2006-797
February 2007

Canada

Introduction to cryptography

F. Paquin
Nurun

M. Salois
DRDC Valcartier

Defence R & D Canada – Valcartier

Technical Memorandum

DRDC Valcartier TM 2006-797

February 2007

Author

Martin Salois

Approved by

Yves van Chestein
Head/Information and Knowledge Management

Approved for release by

Christian Carrier
Chief Scientist

Abstract

The fields of software reliability, safety, and security are quite vast. One cannot hope to be an expert in every little aspect but should know enough to follow discussions and link the information from different fields. One aspect that always comes up in security is cryptography. Often misunderstood, always mysterious, cryptography is a complex subject and trying to explain it to the uninitiated is difficult. The purpose of this document is to give a quick overview of the essentials of this vast subject and, more importantly, point the reader to authoritative references.

Résumé

Les champs de la fiabilité, de la sûreté et de la sécurité des logiciels sont très vastes. Personne ne peut se prétendre expert dans tous leurs petits aspects. Par contre, cette même personne devrait en connaître assez pour suivre les discussions et faire les liens entre l'information de chacun des différents champs d'application. Un aspect qui revient toujours en sécurité est la cryptographie. Souvent mal comprise, toujours mystérieuse, la cryptographie est un sujet complexe et essayer de l'expliquer aux profanes est difficile. Le but de ce document est de présenter un bref survol des éléments essentiels de ce vaste sujet et, encore plus important, d'orienter le lecteur vers des références qui font autorité.

This page intentionally left blank.

Executive Summary

Working within a group that addresses software reliability, safety, and security, one often gets questions about cryptography. These questions vary from very simple (e.g. which tool to use to encrypt emails), to very complex ones (e.g. which algorithm is better in such and such circumstances). While not actively pursuing research in this field, the group must still be able to answer some of these questions and discuss them in general. Also, a few projects came up recently that required a basic knowledge of cryptography from their participants. The goal for this document is to serve as a starting ground for these projects and others that require a working knowledge of cryptography.

Searching the literature and the web for good introductions to cryptography yields many good finds. However, they are often much too complex and complete for an overview or, on the reverse, too simplistic to be of any real use. This document tries to stand between those extremes. It provides an overview of the basic concepts required to use cryptography, select the right tool, and discuss it at a non technical level. It also points the reader to more authoritative references and indicates which groups to keep an eye on.

Originally, the background work was performed for a 90 minute presentation on cryptography (PowerPoint included on the accompanying CD-ROM.) This turned out to be quite useful so it was decided to convert it into a technical memorandum. All references that are available electronically are also on the CD-ROM.

F. Paquin, M. Salois; 2007; Introduction to cryptography; DRDC Valcartier TM 2006-797; Defence R & D Canada – Valcartier.

Sommaire

En travaillant dans un groupe qui s'occupe de sécurité, de fiabilité et de sécurité logicielle, on se fait souvent poser des questions sur la cryptographie. Ces questions vont du très simple (p. ex., quel outil utiliser pour encrypter les courriels ?) au très complexe (p. ex., quel algorithme est meilleur dans telle ou telle circonstance ?). Tout en n'étant pas actif dans ce domaine de recherche, le groupe doit tout de même être capable de répondre à certaines de ces questions et pouvoir en discuter de façon générale. De plus, quelques projets ont récemment eu besoin d'une connaissance de base de la part de leurs participants. Le but de ce document est de servir de point de départ pour ces projets et d'autres qui nécessitent une connaissance pratique de la cryptographie.

Parcourir la littérature et le Web pour y trouver de bonnes introductions à la cryptographie produit une bonne récolte. Par contre, plusieurs trouvailles sont souvent trop complexes et complètes pour une vue d'ensemble ou, au contraire, sont trop simplistes pour être d'une réelle utilité. Ce document essaie de se situer entre ces deux extrêmes. Il présente un survol des concepts de base requis pour utiliser la cryptographie, sélectionner les bons outils et en discuter à un niveau non technique. Il renvoie aussi le lecteur vers des références qui font autorité et indique sur quels groupes garder un oeil.

À l'origine, le travail de base a été fait pour une présentation de 90 minutes sur la cryptographie (le PowerPoint en anglais est fourni sur le CD-ROM joint). Cette présentation ayant été très pratique, il fut décidé de la convertir en mémorandum technique. Toutes les références disponibles électroniquement sont également incluses sur le CD-ROM (en anglais pour la plupart, malheureusement).

F. Paquin, M. Salois; 2007; Introduction à la cryptographie; DRDC Valcartier TM 2006-797; R & D pour la défense Canada – Valcartier.

Table of Contents

List of Figures

List of Tables

This page intentionally left blank.

1 Introduction

With the increasing popularity of the internet and e-commerce, the security of communications and data storage is now an important factor to consider in software development. One way of protecting sensitive data is to encode it so no one but the intended recipient can see its content. The science that does this by encoding information so that it looks completely different from the original form is called cryptography.

Recently, the team at RDDC Valcartier encountered a few projects for which a basic knowledge of cryptography was required. Since questions about cryptography also often come up in discussion with clients, it was decided to produce a 90-minute introduction that could be used in such circumstances. The ensuing PowerPoint presentation is available on the accompanying CD-ROM. Finding the presentation useful, it was decided to write an accompanying document that could be used when the presenters are not available or when the audience wants more detail and wishes to read on the subject. This is the result.

This document is an introduction to the basic concepts of cryptography. This is a complex subject involving much mathematics and advanced concepts. The goal here is not to give all of these details but rather to provide a guide to external references. These four documents all provide a more complete overview [1, 2, 3, 4] but are quite long to read. David [5] provides a more digestible overview in the form of lecture slides.

For the avid reader who wants the complete lowdown, the following five books are recommended: [6, 7, 8, 9, 10].

2 What is Cryptography?

2.1 Basic Terminology

Originally, cryptography was the art of writing secret messages. *Crypto* means secret and *graphy* means writing. To read an encrypted message, the reader must know the secret key; otherwise the message looks like gibberish. This art evolved to become more of a science and it is now possible to encrypt not only messages but also files, pictures, and virtually any kind of digital document that needs protection.

As the field of cryptography advances, the dividing line for what is and what is not cryptography has becomes blurred. Following is a general discussion on the concepts of cryptography, as they were extracted and amalgamated from various sources. A quick read and a good starting point is available at Wikipedia [11].

A simple but overarching definition of cryptography today might be the study of techniques and applications that depend on the existence of difficult problems to hide something from someone. To most people, cryptography is concerned with keeping communications private. Indeed, the protection of sensitive communications has been the emphasis of cryptography throughout much of its history. However, this is only one part of today's cryptography.

Encryption is the transformation of data into a form that is as difficult to read as possible without the appropriate knowledge (a *key*; see below). Its purpose is to ensure privacy by keeping information hidden from eavesdroppers, even those who have access to the encrypted data.

Decryption is the reverse of encryption. It is the transformation of encrypted data back into an intelligible form.

Encryption and decryption generally require the use of some secret information, referred to as a key. In modern cryptography, this is generally a series of numbers that is used as the seed for the cryptographic algorithms. A more basic key can be as simple as knowing which letters to substitute for which other letters. For some encryption mechanisms, the same key is used for both encryption and decryption; for others, different keys are used.

A *plaintext*, or *cleartext*, message is the original message that needs to be encrypted. The *ciphertext* is the encrypted message that needs to be decrypted to get back the original plain text. *Cipher*, by itself, is often used to refer to the actual pair of algorithm that encrypts/decrypts.

Cryptanalysis is the study of how to compromise (defeat) cryptographic mechanisms. That is, how to obtain the original message without the key. One common technique is to use the repetitive and patterned features of a language. For example, if one knows that the language used is English, then the most prominent letter is certainly 'e'. From there, some words can be guessed and the whole text can slowly be deciphered. Of course, this technique works only for the most basic of cryptographic algorithm...

Cryptology is the discipline of cryptography and cryptanalysis combined.

A *cryptosystem* is a piece of hardware or software that encrypts/decrypts back and forth between a plaintext and a ciphertext.

As mentioned, today's cryptography is more than just encryption and decryption. *Authentication*, or the art of making sure one is talking to whom he thinks he is talking to, is as fundamental as privacy. It is used almost on a daily basis. For example, when a student signs an exam or signs off for a credit card. As the world moves

more and more towards electronic communications, the need to have electronic techniques for providing authentication arises. Cryptography provides mechanisms for such procedures.

A *digital signature* binds a document to the possessor of a particular key, while a *digital timestamp* binds a document to its creation at a particular time. These cryptographic mechanisms can be used to control access to a shared disk drive, a high security installation, or a pay-per-view TV channel, for example.

Steganography is a related field that consists in hiding a secret message within another, seemingly harmless message. The main idea is that no one knows that there is a message.

Finally, the field of cryptography encompasses other uses as well. With just a few basic cryptographic tools, it is possible to build elaborate schemes and protocols. There are many examples, such as paying with electronic money or proving one knows certain information without revealing the information itself.

While modern cryptography is growing increasingly diverse, cryptography is fundamentally based on mathematical problems that are difficult to solve. A problem may be difficult because its solution requires some secret knowledge, such as decrypting an encrypted message or signing some digital document. The problem may also be hard because it is intrinsically difficult to complete, such as finding a message that produces a given hash value or finding really large prime numbers.

2.2 Purpose

Here are some properties that can be enforced using cryptography:

Confidentiality It is cryptography's traditional role to prevent an eavesdropper from understanding the content of a message or any other document. To send sensitive information over the Internet, for example, one might want to encrypt it because Internet is not a secure communication channel. It becomes secure after a proper cryptosystem is implemented.

Authentication and integrity It's often important to verify who sent a message and that the message was not been modified after being sent.

Let's suppose for example that one sends a message to one's stockbroker saying buy this or sell that, one wants to make sure nobody can alter the message before it gets to the broker. This ensures the *integrity* of the message. The broker also needs a means to verify who really sent the message. This is *authentication*. Public key cryptography along with hash functions perform these important tasks.

Non-repudiation This property ensures that the sender cannot deny having sent the message.

Coming back to the broker example, let's say that one's broker sends a message instructing to buy stocks from company XYZ. One can decide to put all of his money in this company and lose all his lifetime savings the next day when the company goes bankrupt. The broker could say that he never sent this message. It would be impossible to prove otherwise and sue the broker without the proper use of non-repudiation crypto-mechanisms. Thus, the legal requirements of many e-commerce applications are such that they require non-repudiation sufficiently robust for the recipient to prove to a third party such as a judge or jury that the sender's denial was false.

2.3 History

A very good summary of the history of cryptography is already provided by Gordon [2]:

> "Around 100 AD, the *Khama Sutra* of Vatsyayana described a series of 64 life-skills, or *yogas*, for the edification of young, well-to-do ladies in India. While many of these *yogas* were highly erotic, number 44 was the *Science of Writing in Secret Ciphers*.
>
> So cryptography is hardly new.
>
> Indeed, 150 years before this, Julius Caesar was adept at the use of ciphers, and modern textbooks still speak of the *Caesar cipher* he invented, and with which he secretly corresponded with Cicero.
>
> We can go back even further, for Cryptography is as old as writing itself. In 800 BC, Homer in the Iliad, made only one reference to writing - the story of Bellerophon - the first man in history or legend to carry an encrypted message. It was to be his doom.
>
> But to find the very earliest known examples, we must go back to around 2000 BC, to the tomb of Khnumhotep II, which was inscribed with mysterious, occult symbols, quite unlike the common, demotic hieroglyphs which the Egyptians used for their normal record keeping. These hieratic signs were used for secret and magical purposes by the priests.
>
> Cryptanalysis - the science of breaking ciphers - was certainly well established by 1412 when Qualqashandi wrote a 14-volume treatise on the routine breaking Arabic codes.
>
> Cryptography and cryptanalysis have been major forces in history. In sixteenth century England, Mary Queen of Scots was sentenced to death in 1586 and beheaded, on the evidence of the cryptanalyst, Thomas Phillippes.

She had been treasonably corresponding in a secret cipher with Philip II of Spain. In 1917, the breaking of the Zimmerman telegram was instrumental in bringing the United States into World War I. The course World War II was dramatically altered by the cryptanalysts at Bletchley Park in Buckinghamshire, who broke German high grade ciphers.

Today, cryptography is part of our everyday lives, being found in such commonplaces as gas meters, cash payment systems, franking machines, vehicle alarms, nautical charts, TV signal scramblers, the internet, and so forth."

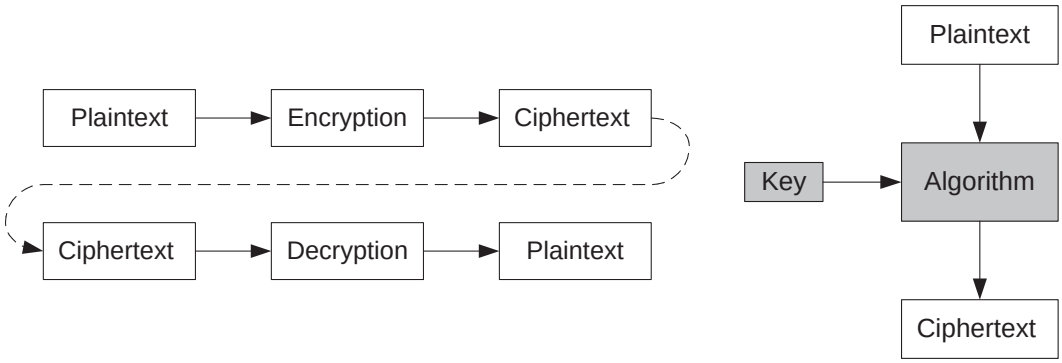
In part 7 of his lectures on network security, David [5] provides more detail on the Caesar cipher:

- Caesar Cipher - Julius Caesar (49-60 BC)
  - Around 50 B.C. Julius Caesar used a substitution cipher to transmit messages. This cipher is MONOALPHABETIC because only one alphabet was used. This cipher involved shifting the alphabet three letters and substituting those letters. It is known as $C3$: $Zi = Cn(Pi)$, where
    - Zi = ciphertext characters
    - Cn = Monoalphabetic substitution where n is the number of letters shifted
    - Pi = Plaintext
  - Julius Caesar (100-44 BC) used a simple substitution with the normal alphabet (just shifting the letters a fixed amount) in government communications.
  - This cipher isn't strong, but in a day when few people read in the first place, it was good enough. He also used transliteration of Latin into Greek letters and a number of other simple ciphers

As is often the case, Wikipedia [11] provides a good starting point in its section on the history of cryptography. Ellison [12] also provides a good summary of a cryptography timeline. Much of the material is taken from Kahn's book [10]. Another good summary is available in Goldwasser and Bellare's notes on cryptography [3, p. 11].

3 How Does it Work?

This section explores the general concepts of cryptography. The most common techniques are explained in some details in the following subsections but let's first start with the general process.



(a) From plaintext to ciphertext and back

(b) Inputting a key into the cryptographic algorithm

Figure 1: *The generic process of encryption/decryption*

Here is how the story goes: a principal has a plain text that he wishes to protect with cryptography. This could be an email message or any kind of digital document. He encrypts it using an encryption algorithm, which results in a ciphertext that is unreadable unless decrypted using the right key. This is illustrated in Figure 1(a), copyright unknown.

Getting the plaintext back from the ciphertext is merely the reverse process as shown above. The diagram in Figure 1(b) is a more accurate way of representing the encryption process because the key is a fundamental part of the cryptosystem. This is usually the only secret piece of the puzzle as a cryptosystem which relies on the secrecy of the algorithm is usually compromised as soon as a third party knows the sequence of operations, no matter how big the key is. This is why all of the modern algorithms are open standards. They rely only on the length of the key.

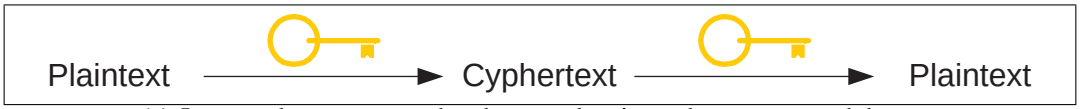
Therefore the process relies on two important concepts [5]:

Algorithm The steps executed and rules associated with the method of performing the encryption.

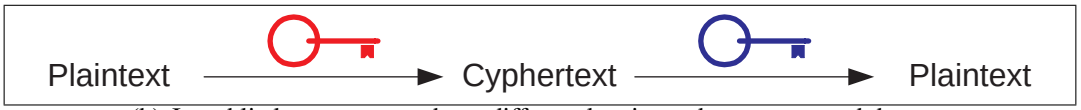
Key The unique attribute that controls the result of the encryption algorithm, and which must be known to reverse the process (decryption).

Of course, many different algorithms have been developed and each use their own types of key, sometimes more than one. However, it could be said that there are three basic types of cryptography:

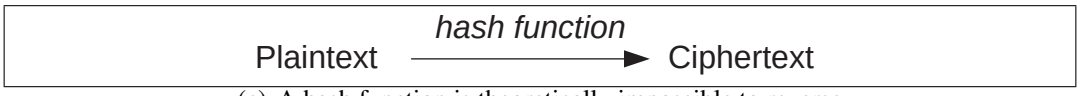
Secret key Also called *symmetric*, this technique uses the same key for encryption and for decryption.



(a) In secret key cryptography, the same key is used to encrypt and decrypt



(b) In public key cryptography, a different key is used to encrypt and decrypt



(c) A hash function is theoretically impossible to reverse

Figure 2: *3 types of cryptography*

Public key Also called *asymmetric*, this technique uses a different key for encryption and for decryption. The *public* key is distributed to everyone and only one principal has the *secret* key that can decrypt something that was encrypted by the public key. The process also works in reverse, meaning that the owner of the secret key is the only one who can encrypt a message that can be decrypted with the public key. Thus, this can be used as a signature to identify the owner.

Hash functions These are a special type of keyless cryptographic algorithms because it is impossible to retrieve the plaintext from the ciphertext. It is similar to a checksum and it is used in digital signatures, for instance, to make sure that the text was not altered. One computes the hash and compares it to the original; if they match, one can be almost certain[1] that the text was not altered.

In general, secret key cryptosystems are much faster than public key systems and are harder to break when using a large key size. This is due to the fact that they use the same key on both ends and this makes the algorithms involved simpler.

The three types are illustrated in Figure 2. More details are given in the next three subsections and complementary or more complex techniques are presented in the remaining subsections.

3.1 Secret Key Cryptosystems

This system requires that both parties know the key before each transmission, and of course, the secret key cannot be sent over an unsecured communication channel. This creates a problem with key management: how to send the key to the intended recipient while making sure that he is the only one to get it? One way

[1] In these extreme mathematical concepts, there is always a possibility...

Encrypt:	Plaintext	10101100	10010110	00001000
	Key	10110110	10110110	10110110
	Cipher	00011010	00100000	10111110
Decrypt:	Key	10110110	10110110	10110110
	Cipher	00011010	00100000	10111110
	Plaintext	10101100	10010110	00001000

Figure 3: *A simple stream cipher*

is to physically distribute the key but this rapidly becomes useless if there are too many recipients since a unique key is required for each. This is why this system is generally used in combination with a public key cryptosystem (subsection 3.2) that handles the session key exchange.

There are two basic approaches with secret key ciphers [5]:

Block cipher A transformation of plain text into ciphertext [of the same length] using a secret key. For example, the plain text may be transformed in blocks 64 bits in size.

Popular block ciphers are: 3-Way, AES, Akelarre, Blowfish, Camellia, CAST-128, CAST-256, CMEA , ,CS-Cipher, DEAL, DES, DES-X, FEAL, FOX, FROG, G-DES, GOST, ICE, IDEA, Iraqi, KASUMI, KHAZAD, Khufu and Khafre, LOKI89/91, LOKI97, Lucifer, MacGuffin, Madryga, MAGENTA, MARS, MISTY1, MMB, NewDES, Noekeon, RC2, RC5, RC6, REDOC, Red Pike, S-1, SAFER, SEED, Serpent, SHACAL, SHARK, Skipjack, Square, TEA, Triple DES, Twofish, XTEA.

Stream cipher The transformation works at the bit level, rather than on blocks of data, and is generally faster than using a block cipher technique. It is accomplished through the use of a message stream and a key stream to produce a ciphertext stream. It is generally an exclusive-ORing of each bit of the key with each bit of the plaintext message. A simple example from [5] is shown in Figure 3.

Popular stream ciphers are: A5/1, A5/2, Chameleon, FISH, Grain, Helix, ISAAC, LEVIATHAN, MUGI, Panama, Pike, QUISCI, RC4, Salsa20, SEAL, SOBER, SOBER-128, Trivium, VEST, WAKE.

Block ciphers are extremely fast and depend on what is called a *mode of operation*. A mode of operation describes how to manage plaintext that is larger than the block size, as is usually the case. The process is illustrated below with two modes of operation, namely ECB and CBC. More detail on secret key cryptography can be found in [13], [3, p. 51], and [2, p. 11].

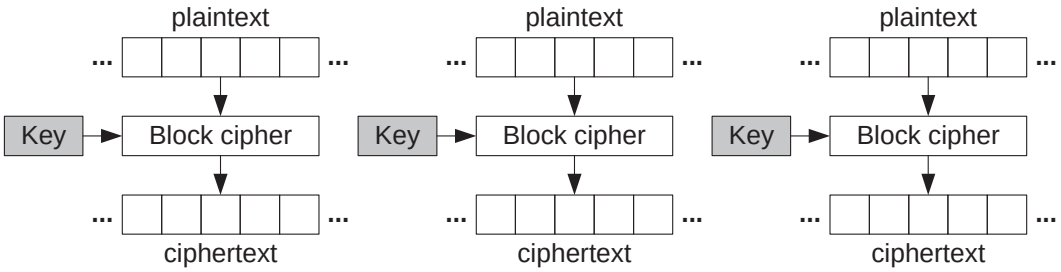


Figure 4: *The ECB mode encryption*

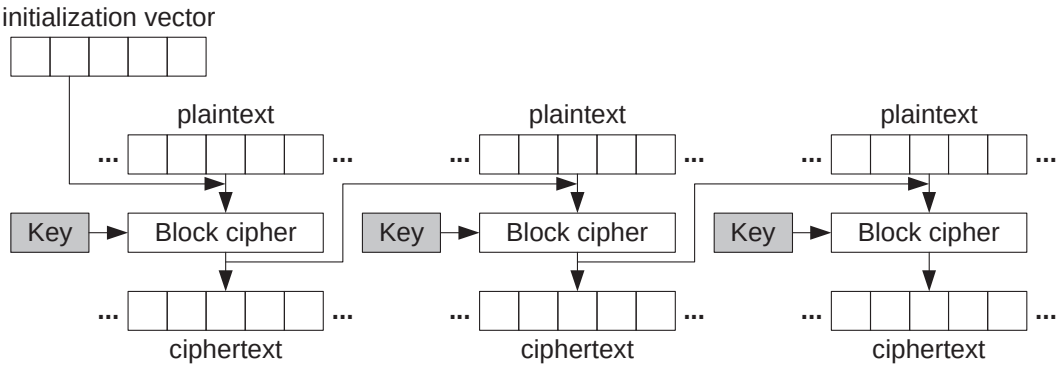


Figure 5: *The CBC mode encryption*

3.1.1 Electronic Code-Book (ECB)

In this mode of operation, each block of plaintext is encrypted separately and the ciphertext is the same length as the plaintext. ECB is not used in practice because it does not hide the patterns that are present in the plain text. Cryptanalysis is then pretty simple since language patterns are still present, especially if the attacker has a piece of plaintext and its corresponding ciphertext. The process is illustrated in Figure 4, based on [11].

3.1.2 Cipher Block Chaining (CBC)

Unlike in the ECB mode of encryption, this mode is not length preserving. An initialization vector, usually random, is chosen and ends up being part of the cipher. The idea is that the plaintext is fed into the algorithm along with the cipher of the previous block; hence, the need for an initialization vector. This approach is powerful because the cipher for a specific block will not always be the same since it depends on its surrounding. This makes cryptanalysis much more difficult as the language patterns tend to disappear. It also contributes to making this mode of encryption the most popular, used pervasively in practice.

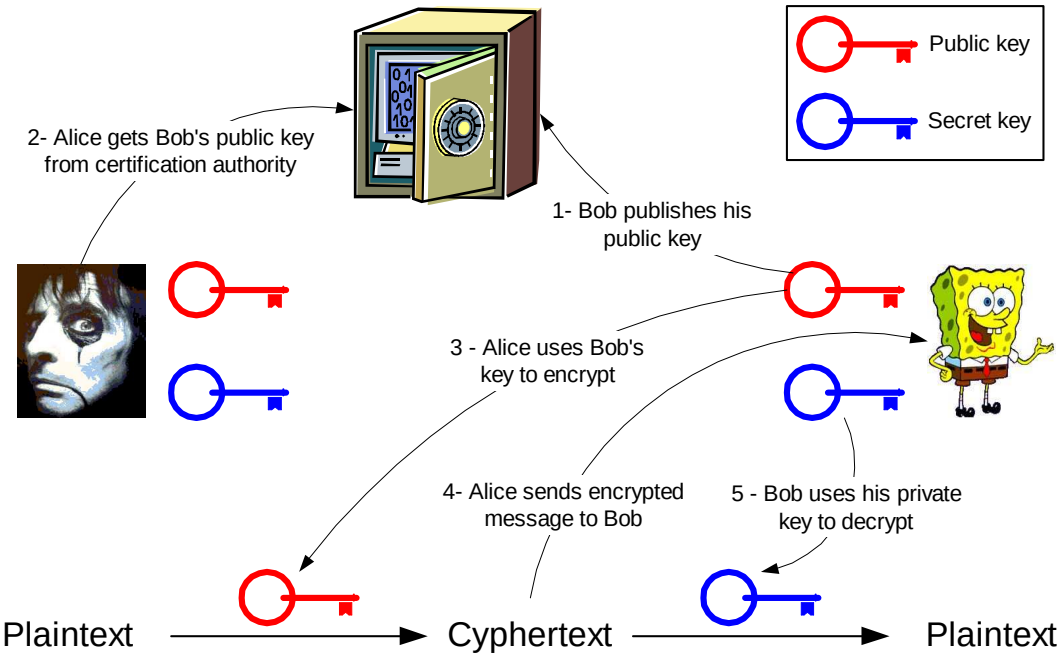


Figure 6: *A simplified view of a public key cryptosystem*

3.2 Public Key Cryptosystems

The previous subsection has shown that secret key cryptography requires that both the sender and the receiver possess the same key. It has also been said that this creates a huge management overhead as one must make sure that only the intended recipient gets the key and no one else. It also requires a different key for every pair of principals that wishes to communicate secretly. A solution to this was developed by Whitfield Diffie and Martin E. Hellman in the mid 70s: the asymmetric public key cryptosystem [14]. The principle has since become ubiquitous. Here is a simple summary of how it works.

Let's say Alice wants to send a private message to Bob.

The first thing that Alice needs to do is to get Bob's public key. Alice can ask Bob directly for his key or can look it up in a server called a certification authority, a kind of certified phone directory for such things. Once Alice has Bob's public key, he encrypts his private message using this key. From now on, only Bob can decrypt the ciphertext with his private key and retrieve Alice's original message. All Alice has to do now is to send the ciphertext to Bob. This is illustrated in Figure 6.

This technique tends to be much slower than a secret key protocol as the mathematics involved are much more complex (the factorization of very large integer). Consequently, it is often used to distribute a secret key, created for the occasion, be-

tween two principals. The whole architecture that support this (servers, certificates, registration, keys, etc.) is called a *public key infrastructure*.

Unfortunately, the original protocol developed by Diffie and Hellman was later found to be vulnerable to a man-in-the-middle attack. This happens if an attacker can somehow get in between Alice and Bob and substitute his own public key to Bob's and Alice's. The attacker can then pretend to be Bob to Alice (and vice versa) and can decrypt all the communication. This illustrates the complexities involved in cryptography. This is also why one should always rely on tried-and-tested cryptosystems that were developed by professionals. A solution to this particular vulnerability was later found in 1992 by Diffie, van Oorschot, and Wiener [15].

More detail about public key infrastructure can be found in [1, p. 81], [2, p. 29], [3, p. 120], and [4, p. 73].

Popular public key cryptosystems are: RSA, EIGamal, Diffie-Hellman (key exchange protocol) and Elliptic Curve Cryptography (ECC) (discussed in 3.5.2).

3.3 Hash Functions

A hash function is simply a transformation that takes as input some text (or a binary string) and produces a unique fixed-size string called a *hash value*. This is used in many areas, not just in cryptography. For example, this is used by Windows to make sure that the files of the operating systems are the originals.

In order to be considered secure, a hash function used in cryptography must make it computationally very difficult (almost infeasible) to create a message that produces the same hash value as another message. As always in cryptography, the point here is "almost infeasible", as in there is always a possibility but it is remote enough to be practically impossible.

A series of popular hash functions is known as the Message Digest (MD) algorithms. These algorithms are byte-oriented and produce hash values of 128 bits. MD5 is the most well known and is used by many web site, for example, to ensure the integrity of downloaded files. Weaknesses were found in the protocol in 1996 [16] and newer functions exist to address this issue but are not as ubiquitous yet.

Other popular hash functions are: Secure Hash Algorithm (SHA), RIPEMD, HAVAL.

More detail about hash functions can be found in [1, p. 154] and [2, p. 32].

3.4 Digital Signatures

A *digital signature* binds a message to a unique private key. This ensures that the message can not be tampered with by a third party. This property is known as *non-repudiation*. It means that the owner cannot pretend no to have sent the message (unless he can prove that his key was stolen).

The process and the use are similar to a hash function, except that public/private keys are involved. For the sender, a simple calculation involving his secret key and the message is performed. To verify the integrity of the message, the receiver performs another calculation that involves the digital signature, the message, and the sender's public key.

More detail about digital signatures can be found in [1, p. 137], [2, p. 30], [3, p. 164], and [4, p. 99].

3.5 Other Techniques

Here are relatively recent variant techniques that have appeared in the field of cryptography. They are not mainstream yet (and might never be!).

3.5.1 Message Authentication Code (MAC)

Similar to a digital signature, a *message authentication code* (MAC) binds a message to a unique private key. The difference is that, in this case, the same key is used to verify the integrity of the message. The process is thus much faster as discussed at the beginning of the section. However, it is also weaker and can not provide third party authentication and non-repudiation as at least two persons need to share the secret key.

FIPS PUB 198 [17], a documented standard from the United States' Department of Commerce provides more detail.

3.5.2 Elliptic Curve Cryptography (ECC)

As discussed, public key cryptosystems rely on complex mathematical problems (namely the factorization of very large integer). This requires very large key to make sure that the problem to solve is not trivial. *Elliptic curve* cryptography uses a different set of difficult problems related to the elliptic curve discrete logarithm problem (which is quite out of the scope of this document).

There are two alternatives using this technique. The first one is analog to the RSA system and does not offer much of an advantage over it since it is slower. The sec-

ond one shows potential as algorithms for solving its equation are currently much less efficient than those used for integer factorization. Therefore, much shorter keys could be used with all the optimization that this entails.

More detail about ECC can be found in [1, p. 124], [2, p. 34], [3, p. 264], and [4, p. 101]. Complete and gory details can be found in [18] and [19].

3.5.3 Quantum Cryptography

The golden fleece of cryptography, *quantum cryptography* uses quantum mechanics to detect eavesdroppers with 100% certainty. This would be the perfect solution to exchanging throw-away secret keys in a symmetric cryptosystem.

Because of its inherent properties, a quantum random number generator would also be the perfect tool for any algorithm that relies on random numbers (and there are many!)

Such systems are quite complex and many are sceptical of real-world application as it relies on a perfect system and errors always happen in the real world. A more technical discussion on this subject is well beyond the scope of this document.

More detail about quantum cryptography can be found in [20] and [21].

4 Existing Tools

As seen in the previous section, cryptography can serve a variety of purposes, such as protecting the content of a storage device, securing a communication channel, and authenticating a document, just to name a few. A good way of getting familiar with cryptography is by playing around with tools that implement cryptosystems or cryptographic algorithms. Now that the basics of cryptography have been illustrated, the reader can understand and appreciate what existing tools can do and evaluate their strength. Securing data requires choosing the right tools and be aware of their limitations as far as security is concerned.

One can even develop his own tool using libraries of cryptographic algorithms. A zealous user can also craft his own algorithms. However, this is something better left to professional as the underlying theories are quite complex and the smallest error will leave the door wide open to other, more malicious, professionals. Remember that the strength of a cryptosystem should never reside in the secrecy of the algorithm but in the key used. So using a thoroughly documented well-proven-in-service cryptographic tool is often the best option for securing data or communications.

Therefore, Table 1 presents a collection of links to popular tools implementing cryptosystems or cryptographic algorithms. Some of these products are shareware or open source projects while others are commercial tools using either software, hardware, or both. As DRDC's current projects are for the vast majority Microsoft Windows related, so are most of the tools presented, although many of them have versions for other platforms as well.

Table 1: Cryptographic tools

Product: PGP	Source: Open source
Country: U.S.A. (original)	Web: See below

Description:

- The International PGP Home Page
  http://www.pgpi.org
- MIT Distribution Site for PGP
  http://web.mit.edu/network/pgp.html
- The OpenPGP Alliance
  http://www.openpgp.org

Product: Various	Source: Bokler Software
Country: U.S.A.	Web: http://www.bokler.com

Description:

From the web site:

- DESciphe^{TM} (single-DES only)
- HASHcipher^{TM} (SHA-1 & MD5)
- TDESciphe^{TM} (triple-DES & single-DES)
- B64codec^{TM} (Base64 encoding)
- CipherLockSDA^{TM} (Self-Decrypting Archives)
- RDESciphe^{TM} (AES / Rijndael)

Product: Various	Source: Microsoft
Country: U.S.A. Web: http://msdn2.microsoft.com/en-us/library/aa380259.aspx	

Description:

From the website:

> Cryptography tools provide command-line tools for code signing, signature verification, and other cryptography tasks.

Table 1: Cryptographic tools (continued)

Product: Crypto++ Library — Source: Wei Dai
Country: Web page in U.S.A. — Web: http://www.eskimo.com/~weidai/cryptlib.html

Description:

From the website:

> Crypto++ Library is a free C++ class library of cryptographic schemes.

Product: Cryptlib — Source: Peter Gutmann/University of Auckland
Country: New Zealand — Web: http://www.cs.auckland.ac.nz/~pgut001/cryptlib/index.html

Description:

From the website:

> Cryptlib is a powerful security toolkit that allows even inexperienced crypto programmers to easily add encryption and authentication services to their software. The high-level interface provides anyone with the ability to add strong security capabilities to an application in as little as half an hour, without needing to know any of the low-level details that make the encryption or authentication work. Because of this, cryptlib dramatically reduces the cost involved in adding security to new or existing applications.

The 'half-hour' is probably more like 'half a week' but the product sounds serious.

Product: AES — Source: NIST
Country: U.S.A. — Web: http://csrc.nist.gov/CryptoToolkit/aes/rijndael

Description:

From the website:

> Information on the AES algorithm (Rijndael) is available, including test values, intellectual property (IP) statements, and specifications.
>
> Please note that these pages are included for historical reference only. As these pages refer to a development effort, they may contain errors. Refer to FIPS 197 for the actual specification of AES.

Table 1: Cryptographic tools (continued)

Product: RSA	Source: RSA Security (part of EMC)
Country: U.S.A.	Web: http://www.rsasecurity.com
Description: The site of RSA Data Security Inc, creators of the RSA encryption technology used in Netscape Navigator, Quicken, Lotus Notes, and hundreds of other products.	
Product: TurboCrypt	Source: PMC Ciphers
Country: Germany (& U.S.A.)	Web: http://www.pmc-ciphers.com/index.php
Description: From the website: PMC Ciphers is a data security company that specializes in encryption technology. The backbone of PMC is our patented Polymorphic Encryption Technology. This technology allows us to create ultra-fast and ultra secure ciphers that are extremely adaptable. By combining speed, strength and adaptability, PMC Ciphers is able to provide the best ciphers on the market.	

Finally, Table 2 presents groups and projects in cryptography that are worth keeping an eye on.

Table 2: Active groups and projects

Cambridge University Computer Security Group	http://www.cl.cam.ac.uk/Research/Security
Centre for Computer Security Research, University of Wollongong, NSW, Australia	http://www.cs.uow.edu.au/ccsr
Center for Cryptography Computer and Network Security (University of Wisconsin, Milwaukee)	http://www.cs.uwm.edu/~cccns
Crypto group at Katholieke Universiteit Leuven	http://www.esat.kuleuven.ac.be/cosic/cosic.html
Cryptography Research	http://www.cryptography.com
École Normale Supérieure: Département d'Informatique	http://www.di.ens.fr/CryptoTeam.html
Eiji Okamoto	http://grampus.jaist.ac.jp:8080/index.html
Hidden Field Equations (HFE)	http://www.minrank.org/hfe/

Table 2: Active groups and projects (continued)

Information & Communications Security Laboratory, Sung Kyun Kwan University, Korea	http://dosan.skku.ac.kr
Information Security and Telecommunications Laboratory, Pohang, Korea	http://wooly.postech.ac.kr
Information Security Group, Royal Holloway, University of London	http://www.isg.rhul.ac.uk
KRyptoGate (Korean cRYPTOgraphers' GATEway)	http://www.cryptogate.com
Naval Research Lab Group	http://www.itd.nrl.navy.mil/ITD/5540/projects/crypto.html
Quantum Cryptography at Los Alamos	http://p23.lanl.gov/Quantum/quantum.htm
Quantum Cryptography in Norway	http://www.fysel.ntnu.no/Optics/qcr
SCSI	http://www.ulb.ac.be/di/scsi/defscsi.html

5 International Standards and Laws

Modern cryptography has been developed independently by several civil and military organizations all around the world. With the increasing need for secure communications worldwide, it soon became mandatory to establish international standards. These standards are guidelines to set up a cryptosystem with a security level that complies with the requirements of the international organizations that adopted the standards.

Many countries today still do not have regulations concerning the use of cryptography. The United States is probably the only western country that still has restriction policies. This can probably be explained by the fact that they their security agencies are responsible for a large part of the international cryptographic policies. However, it seems like it has been a while since these policies have been actively enforced (circa 1998). Standards for cryptography in the United States are issued by the National Institutes of Standards and Technology (NIST). Many in the private sector worldwide use these standards as well. In fact, the standards are published as Federal Information Processing Standards (FIPS) and are not restricted to the United States. In January 1997, DES, described in FIPS-46 became the de facto standard throughout the United States. The complete list of FIPS publications can be found at http://www.itl.nist.gov/fipspubs/by-num.htm.

In Canada, there are no laws restricting the private use of cryptography. There are some restrictions on exportation but they seem pretty relaxed. A report from 2000 by the Electronic Privacy Information Center (EPIC) [22] contains a very thorough summary of the policies for almost every country on the planet.

As many algorithms are patented or copyrighted, standard international business laws also apply to some cryptographic systems.

6 Breaking Cryptography

As stated earlier, cryptanalysis is the science that analyzes cryptographic systems in an effort to break them. This science is probably only a day younger then cryptography itself. Originally, cryptanalysis is the process of reverse engineering a cryptosystem or a cryptographic algorithm to identify vulnerabilities or flaws. The ultimate goal is to understand the content of an encrypted message without knowing the key.

With modern cryptographic algorithms using large size keys, a brute force attack is often impossible to achieve as it would take hundreds of years with current computers. A *brute-force attack* consist in trying out all keys and looking at the output to see if the result makes sense. Such a simple-minded attack usually does not retrieve the password, however, since the hash-function is one-way. But it will eventually retrieve the encrypted content. Modern systems rely on the fact that this technique could potentially take hundreds of year to run on top-of-the-line systems.

It is also possible to try to retrieve the password by using the cryptography algorithm with every possible password. Over the years, this technique has been refined to look first at more plausible inputs. For example, if the system only allows alphanumeric characters than it eliminates all other characters (e.g. “.,:;!%") from the input space and, thus, considerably reduce it. Since users often use words for password, another extremely fast technique is to use all the words from a dictionary. Variations are to use combination of words and capitalization. This is known as a *dictionary attack*.

Much more advanced techniques will look at word distribution and language statistics to uncover the hidden text. For example, in the English language, the letter ‘e’ is much more present than any other letters. Therefore, the corresponding cipher would also be the most popular and, from there, it is possible to start guessing words and work from there. The reader may have guessed that this technique is made at least an order of magnitude more complex by modern cryptographic algorithm... More detail is out of the scope of this document.

A scan of a United States Army field manual about basic cryptanalysis is available on the Internet [23] and covers many of these techniques in detail.

As computers get ever more powerful, the requirements for bigger keys will keep increasing. For example, the proponents of DES assumed it would require millions of dollar to make a computer (or a network of computers) that could break DES in a reasonable time and issued a public challenge stating so. A team of researched quickly built a $250,000 supercomputer that was able to break a 56-bit key in less than three days [24]!

On the other hand, so much effort is often not even required. Although modern cryptosystems are rather hard to break in theory, their implementations often contain flaws that allow an attacker to bypass the algorithms altogether. In fact, Schneier [25] identifies 8 vectors of attack against cryptosystems:

Attacks against cryptographic designs If the algorithms are not strong to begin with or if they use a predictable random number generator, then the cryptosystem can be broken. The same applies if the system is not installed properly.

Attacks against implementations As is the case in most software systems, programmers tends to make the same mistakes over and over again. Leaving plaintext or keys around memory or on disk is a fatal flaw. Programmers also take shortcuts or do trade-offs to enhance ease of use or make things simpler. These programmers are often not even aware of the consequences because they do not understand the complexities of cryptography. Also, "backup keys" are sometimes simply left in an unsecured location.

Attacks against passwords User-generated passwords are often weak and easy to guess. Cryptosystems that rely too much on such password are quite often easy to break.

Attacks against hardware Tamper-resistant hardware is rarely as resistant as the vendors would like you to believe. Hoping that bad guys will never have access to the inner workings of the system is called "security through obscurity" and is wishful thinking. If it is valuable enough, bad guys will find a way.

Attacks against trust models Knowing who to trust and when is key to many cryptosystems. Designer often make assumptions about usage that are not verified in real life. Consumers (and merchants!) can collude to get at information that they could not get to alone. A basic assumption, for example, is often that the cryptosystem will run in a secure environment, which is rarely the case. Most systems will eventually run on the Internet and on various other networks and unsecured computers.

Attacks on the users Users can always circumvent security, often unwittingly. It has been demonstrated many times that users will give their password easily enough under the right conditions. Users also often reuse the same passwords all over the place. A bad guy can break into an easy system to get a password and get access to stronger system with the same password.

Attacks against failure recovery Cryptosystems can be left in insecure conditions after an update or an upgrade. Systems that default to "no security" are the norm rather than the exception. If the bad guy can cause a system crash, there is a good change that the system will revert to a less secure version.

Attacks against the cryptography The science of cryptography is very difficult and some people will just get it wrong. Most proprietary encryption algorithms are thus often very weak. A funny quote from [25] says: " The system for DVD encryption took a weak algorithm and made it weaker"! The bottom-line is to leave this the professionals.

7 Conclusion

As mentioned at the beginning of this document, the purpose of this document is to provide an overview of the basic concepts of cryptography. This has been achieved with the help of summaries, the reproduction of seminal figures, and pointers to more detailed references.

Finally, the interested reader who wants to keep up to date with the world of cryptography might want to keep an eye on this Wikipedia*esque* resource about cryptography: http://www.cryptodox.com. It contains a good portion of the documents referenced here and a lot of others.

References

1. Paar, Christof. Applied Cryptography and Data Security. http://www.crypto.ruhr-uni-bochum.de/en_lectures.html.

2. Gordon, John (1998). Introduction to Cryptography. Concept Laboratories, United Kingdom. http://www.conceptlabs.co.uk. Read the PDF.

3. Goldwasser, Shafi and Bellare, Mihir (2001). Lecture Notes on Cryptography. MIT, Cambrigde, and University of California, United States. http://www.cse.ucsd.edu/users/mihir/crypto-lecnotes.html. Read the PDF. Lecture notes.

4. RSA Laboratories (2000). Frequently Asked Questions About Today's Cryptography. RSA Laboratories. http://www.rsasecurity.com/rsalabs/node.asp?id=2152. Read the PDF.

5. David, Robert G. (2004). CEN-4540: Intro to Computer & Network Security. Computer Science Department, University of West Florida, United States. http://www.cs.uwf.edu/~rdavid/CEN4540/CEN4540.html. Read the PDF. Lecture slides.

6. Stallings, William (2005). Cryptography and Network Security, 4th ed. Prentice Hall. ISBN: 0131873164.

7. Stinson, Douglas R. (2002). Cryptography: Theory and Practice, 2nd ed. Chapman & Hall/CRC. ISBN: 1584882069.

8. Alfred J. Menezes, Scott A. Vanstone, Paul C. van Oorschot (1996). Handbook of Applied Cryptography, 1st ed. CRC. ISBN: 0849385237.

9. Schneier, Bruce (1995). Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2nd ed. Wiley. ISBN: 0471117099.

10. Kahn, David (1996). The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet, 2 ed. Scribner. ISBN: 0684831309.

11. Wikipedia. Cryptography: From Wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/Cryptography.

12. Ellison, Carl (2004). Cryptography Timeline. http://world.std.com/~cme/html/timeline.html. Read the PDF.

13. Dworkin, Morris (2001). Recommendation for Block Cipher Modes of Operation: Methods and Techniques. (Special publication 800-38A). National Institute of Standards and Technology (NIST). United States Department of Commerce. http://csrc.nist.gov/publications/nistpubs.

14. Diffie, Whitfield and Hellman, Martin E. (1976). New Directions in Cryptography. *IEEE Transactions on Information Theory*, **IT-22**(6), 644–654. htpp://citeseer.ist.psu.edu/diffie76new.html. Read the PDF.

15. Diffie, Whitfield, van Oorschot, Paul C., and Wiener, Michael J. (1992). Authentication and Authenticated Key Exchanges. *Designs, Codes and Cryptography*, **2**(2), 107–125. http://citeseer.ist.psu.edu/diffie92authentication.html. Read the PDF.

16. Dobbertin, Hand (1996). The Status of MD5 After a Recent Attack. *RSA Laboratories' CryptoBytes*, **2**(2), 1–6. http://www.rsasecurity.com/rsalabs/node.asp?id=2149. Read the PDF.

17. Federal Information Processing Standards (FIPS) (2002). The Keyed-Hash Message Authentication Code (HMAC). (Standard Document 198). FIPS. http://www.itl.nist.gov/fipspubs/by-num.htm. Read the PDF.

18. Certicom (2004). An Elliptic Curve Cryptography (ECC) Primer. White paper. http://www.certicom.com/whitepapers. Read the PDF.

19. Saeki, Mugino (1997). Elliptic Curve Cryptosystems. Master's thesis. McGill University. Montréal, Canada. http://citeseer.ist.psu.edu/saeki97elliptic.html. Read the PDF.

20. Gisin, Nicolas, Ribordy, Gregoire, Tittel, Wolfgang, and Zbinden, Hugo (2001). Quantum Cryptography. http://www.citebase.org/abstract?id=oai:arXiv.org:quant-ph/0101098. Read the PDF.

21. id Quantique (2005). Understanding Quantum Cryptography. White paper. http://www.idquantique.com/products/vectis.htm. Read the PDF.

22. EPIC (2000). Cryptography and Liberty 2000: An International Survey of Encryption Policy. Internet report. http://www2.epic.org/reports/crypto2000. Read the PDF.

23. (1990). Basic Cryptanalysis. Department of the Army. Washington DC, United States. http://www.umich.edu/~umich/fm-34-40-2. Read the PDF. FIELD MANUAL NO 34-40-2.

24. Weil, Nancy (1998). U.S. govt.'s encryption standard cracked in record time. *Network World*. http://www.networkworld.com/news/0720des.html. Read the PDF. Electronic version.

25. Schneier, Bruce (1998). Security Pitfalls in Cryptography. Essay, Counterpane Systems. http://www.schneier.com/essay-028.html. Read the PDF.

List of Acronyms

AESAdvanced Encryption Standard

CBCCipher Block Chaining

DESData Encryption Standard

DRDCDefence Research & Development Canada

ECBElectronic Code-Book

ECCElliptic Curve Cryptography

EPICElectronic Privacy Information Center

FIPSFederal Information Processing Standards

MDMessage Digest

MITMassachusetts Institute of Technology

NISTNational Institutes of Standards and Technology

PGPPretty Good Privacy

RDDCRecherche et Développement pour la Défense Canada

RSARivest, Shamir and Adleman

SHASecure Hash Algorithm

This page intentionally left blank.

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.)

Defence R & D Canada – Valcartier
2459 Pie-XI Blvd North, Québec, QC, Canada

2. SECURITY CLASSIFICATION (overall security classification of the document including special warning terms if applicable).

UNCLASSIFIED

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C,R or U) in parentheses after the title).

Introduction to cryptography

4. AUTHORS (Last name, first name, middle initial. If military, show rank, e.g. Doe, Maj. John E.)

Paquin, F. ; Salois, M.

5. DATE OF PUBLICATION (month and year of publication of document)	6a. NO. OF PAGES (total containing information. Include Annexes, Appendices, etc).	6b. NO. OF REFS (total cited in document)
February 2007	35	25

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered).

Technical Memorandum

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include address).

Defence R & D Canada – Valcartier
2459 Pie-XI Blvd North, Québec, QC, Canada

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Specify whether project or grant).	9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written).
15BP02	

10a. ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique.)	10b. OTHER DOCUMENT NOs. (Any other numbers which may be assigned this document either by the originator or by the sponsor.)
DRDC Valcartier TM 2006-797	

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

(X) Unlimited distribution
( ) Defence departments and defence contractors; further distribution only as approved
( ) Defence departments and Canadian defence contractors; further distribution only as approved
( ) Government departments and agencies; further distribution only as approved
( ) Defence departments; further distribution only as approved
( ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution beyond the audience specified in (11) is possible, a wider announcement audience may be selected).

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

The fields of software reliability, safety, and security are quite vast. One cannot hope to be an expert in every little aspect but should know enough to follow discussions and link the information from different fields. One aspect that always comes up in security is cryptography. Often misunderstood, always mysterious, cryptography is a complex subject and trying to explain it to the uninitiated is difficult. The purpose of this document is to give a quick overview of the essentials of this vast subject and, more importantly, point the reader to authoritative references.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title).

cleartext, cryptanalysis, cryptography, cryptology, cryptosystem, decryption, digital signature, encryption, plaintext, public key, secret key, hash, asymmetric, symmetric

Defence R&D Canada

Canada's Leader in Defence
and National Security
Science and Technology

R & D pour la défense Canada

Chef de file au Canada en matière
de science et de technologie pour
la défense et la sécurité nationale

WWW.drdc-rddc.gc.ca