Cryptography and Encryption

KOSTAS ZOTOS, ANDREAS LITKE

Dept. of Applied Informatics,
University of Macedonia
54006 Thessaloniki, GREECE
{zotos, litke}@uom.gr

Abstract.
In cryptography, encryption is the process of obscuring information to make it unreadable without special knowledge. This is usually done for secrecy, and typically for confidential communications. Encryption can also be used for authentication, digital signatures, digital cash e.t.c. In this paper we are going to examine and analyse all these topics in detail.

Key words: cryptography; ciphers; encryption

1. Introduction

The fundamental objective of cryptography is to enable two people, usually referred to as Alice and Bob, to communicate over an insecure channel in such a way that an opponent, Oscar, cannot understand what is being said. This channel could be a telephone line or computer network, for example. The information that Alice wants to send to Bob, which we call "plaintext," can be English text, numerical data, or anything at all — its structure is completely arbitrary. Alice encrypts the plaintext, using a predetermined key, and sends the resulting ciphertext over the channel. Oscar, upon seeing the ciphertext in the channel by eavesdropping, cannot determine what the plaintext was; but Bob, who knows the encryption key, can decrypt the ciphertext and reconstruct the plaintext.

2. Ciphers

A cipher is an algorithm for performing encryption (and the reverse, decryption) — a series of well-defined steps that can be followed as a procedure. An alternative term is encipherment. The original information is known as plaintext, and the encrypted form as ciphertext. The ciphertext message contains all the information of the plaintext message, but is not in a format readable by a human or computer without the proper mechanism to decrypt it; it should resemble random gibberish to those not intended to read it.

Ciphers are usually parameterised by a piece of auxiliary information, called a key. The encrypting procedure is varied depending on the key which changes the detailed operation of the algorithm. Without the key, the cipher cannot be used to encrypt, or more importantly, to decrypt[3].

In non-technical usage, a "cipher" is the same thing as a "(secret) code"; however, in technical discussions they are distinguished into two concepts: codes work at the level of meaning; that is, words or phrases are converted into something else, while ciphers

work at a lower level: the level of individual letters, or small groups of letters — or in modern ciphers, individual bits.

Historically, cryptography was split into a dichotomy of codes and ciphers, and coding had its own terminology, analogous to that for ciphers: "encoding, code text, decoding" and so on. However, codes have a variety of drawbacks, including susceptibility to cryptanalysis and the difficulty of managing a cumbersome codebook. Because of this, codes have fallen into disuse in modern cryptography, and ciphers are the dominant paradigm.

3. Types of cipher

There are a variety of different types of encryption. Algorithms used earlier in the history of cryptography are substantially different to modern methods, and modern ciphers can be classified according to how they operate and whether they use one or two keys.

Encryption methods can be divided into symmetric key algorithm. A symmetric-key algorithm is an algorithm for cryptography that uses the same cryptographic key to encrypt and decrypt the message. Actually, it is sufficient for it to be easy to compute the decryption key from the encryption key and vice versa. In cryptography, an asymmetric key algorithm uses a pair of different, though related, cryptographic keys to encrypt and decrypt. The two keys are related mathematically; a message encrypted by the algorithm using one key can be decrypted by the same algorithm (e.g., RSA), there are two separate keys: a public key is published and enables any sender to perform encryption, while a private key is kept secret by the receiver and enables him to perform decryption. Common asymmetric encryption algorithms available today are all based on the Diffie-Hellman key agreement algorithm.

Symmetric key ciphers can be distinguished into two types, depending on whether they work on blocks of symbols usually of a fixed size (block ciphers), or on a continuous stream of symbols (stream ciphers).

4. A postal analogy

An analogy which can be used to understand the advantages of an asymmetric system is to imagine two people, Alice and Bob, sending a secret message through the public mail. In this example, Alice has the secret message and wants to send it to Bob, after which Bob sends a secret reply.

With a symmetric key system, Alice first puts the secret message in a box, and then locks the box using a padlock to which she has a key. She then sends the box to Bob through regular mail. When Bob receives the box, he uses an identical copy of Alice's key (which he has somehow obtained previously) to open the box, and reads the message. Bob can then use the same padlock to send his secret reply.

In an asymmetric key system, Bob and Alice have separate padlocks. Firstly, Alice asks Bob to send his open padlock to her through regular mail, keeping his key to himself. When Alice receives it she uses it to lock a box containing her message, and sends the locked box to Bob. Bob can then unlock the box with his key and read the message from Alice. To reply, Bob must similarly get Alice's open padlock to lock the box before sending it back to her. The critical advantage in an asymmetric key system is that Bob and Alice never need send a copy of their keys to each other. This substantially reduces the chance that a third party (perhaps, in the example, an corrupted postal worker) will copy a key while is in transit, allowing said third party to spy on all future messages sent between Alice and Bob. In addition, if Bob were to be careless and allow someone else to copy his key, Alice's messages to Bob will be

compromised, but Alice's messages to other people would remain secret, since the other people would be providing different padlocks for Alice to use.

Fortunately cryptography is not concerned with actual padlocks, but with encryption algorithms which aren't vulnerable to hacksaws, bolt cutters, or liquid nitrogen attacks[4].

Not all asymmetric key algorithms operate in precisely this fashion. The most common have the property that Alice and Bob own two keys; neither of which is (so far as is known) deducible from the other. This is known as public-key cryptography, since one key of the pair can be published without affecting message security. In the analogy above, Bob might publish instructions on how to make a lock ("public key"), but the lock is such that it is impossible (so far as is known) to deduce from these instructions how to make a key which will open that lock ("private key"). Those wishing to send messages to Bob use the public key to encrypt the message; Bob uses his private key to decrypt it.

Of course, there is the possibility that someone could "pick" Bob's or Alice's lock. Unlike the case of the one-time pad or its equivalents, there is no currently known asymmetric key algorithm which has been proven to be secure against a mathematical attack. That is, it is not known to be impossible that some relation between the keys in a key pair, or a weakness in an algorithm's operation, might be found which would allow decryption without either key, or using only the encryption key. The security of asymmetric key algorithms is based on estimates of how difficult the underlying mathematical problem is to solve. Such estimates have changed both with the decreasing cost of computer power, and with new mathematical discoveries.

Weaknesses have been found for promising asymmetric key algorithms in the past. The 'knapsack packing' algorithm was found to be insecure when an unsuspected attack came to light. Recently, some attacks based on careful measurements of the exact amount of time it takes known hardware to encrypt plain text have been used to simplify the search for likely decryption keys. Thus, use of asymmetric key algorithms does not ensure security; it is an area of active research to discover and protect against new and unexpected attacks[8].

Another potential weakness in the process of using asymmetric keys is the possibility of a 'Man in the Middle' attack, whereby the communication of public keys is intercepted by a third party and modified to provide the third party's own public keys instead. The encrypted response also must be intercepted, decrypted and re-encrypted using the correct public key in all instances however to avoid suspicion, making this attack difficult to implement in practice.

The first known asymmetric key algorithm was invented by Clifford Cocks of GCHQ in the UK. It was not made public at the time, and was reinvented by Rivest, Shamir, and Adleman at MIT in 1976. It is usually referred to as RSA as a result. RSA relies for its security on the difficulty of factoring very large integers. A breakthrough in that field would cause considerable problems for RSA's security. Currently, RSA is vulnerable to an attack by factoring the 'modulus' part of the public key, even when keys are properly chosen, for keys shorter than perhaps 700 bits. Most authorities suggest that 1024 bit keys will be secure for some time, barring a fundamental breakthrough in factoring practice, but others favor even longer keys.

At least two other asymmetric algorithms were invented after the GCHQ work, but before the RSA publication. These were the Ralph Merkle puzzle cryptographic system and the Diffie-Hellman system. Well after RSA's publication, Taher Elgamal invented the Elgamal discrete log cryptosystem which relies on the difficulty of inverting logs in a finite field. It is used in the Secure Sockets Layer SSL and

Transport Layer Security TLS , its successor, are cryptographic protocols which provide secure communications on the Internet. A relatively new addition to the class of asymmetric key algorithms is elliptic curve cryptography. Elliptic curve cryptography (ECC is an approach to public-key cryptography based on the mathematics of elliptic curves. Proponents claim that ECC can be faster and use smaller keys than older methods — such as RSA — while providing an equivalent level of. While it is more complex computationally, many believe it to represent a more difficult mathematical problem than either the factorisation or discrete logarithm problems[5].

5. RSA algorithm

RSA it is an asymmetric algorithm and plays a key role in public key cryptography. It is widely used in electronic commerce protocols. The algorithm was described in 1977 by Ron Rivest, Adi Shamir and Len Adleman who were all at MIT at the time; the letters RSA are the initials of their surnames.

Clifford Cocks, a British mathematician working for GCHQ, described an equivalent system in an internal document in 1973. His discovery, however, was not revealed until 1997 due to its top-secret classification [2].

The security of the RSA system relies on the difficulty of factoring very large numbers; were such factorization to be quick, cryptanalysis of RSA messages would be quick as well. New fast algorithms in this field could render the RSA algorithm insecure. A working quantum computerMolecule of alanine used in NMR implementation of error correction. Qubits are implemented by spin states of carbon atoms. A quantum computer is any device for computation that makes direct use of distinctively quantum mechanical phenomena, such as superp implementing Shor's algorithmShor's algorithm is a quantum algorithm for factoring a number N in O((log N 3) time and O(log N space, named after Peter Shor. Many public key cryptosystems, such as RSA, will become obsolete if Shor's algorithm is ever implemented in a practical quantum could render RSA insecure through fast factorization. However, this is generally considered not a problem in the short term. At the moment, just as for all ciphers, inadequately long RSA keys are vulnerable to a brute force search approach. The likely effect of an improvement in factoring technique will be to increase the size of adequately long RSA keys. As of 2004, there is no known method of attack which is feasible against the basic algorithm, and sufficiently long. In cryptography, the key size (alternatively key length is a measure of the number of possible keys which can be used in a cipher. Because modern cryptography uses binary keys, the length is usually specified in bits. The length of a key is critical in de RSA keys make brute force attacks infeasible -- that is, effectively impossible [7]. The algorithm was patented by MIT in 1983. Suppose a user Alice wishes to allow Bob to send her a private message over an insecure transmission medium. She takes the following steps to generate a **public key** and a **private key**:

1. Choose two large prime numbers. In mathematics, a prime number or prime for short, is a natural number whose only distinct positive divisors are 1 and itself; otherwise it is called a composite number. Hence a prime number has exactly two divisors. The number 1 is neither prime nor com $p \neq q$ randomly and independently of each other. Compute $N = p\,q$.
2. Choose an integer $1 < e < N$ which is coprime to $(p\text{-}1)(q\text{-}1)$.
3. Compute d such that $d\,e \equiv 1 \pmod{(p\text{-}1)(q\text{-}1)}$.

- (Steps 2 and 3 can be performed with the extended Euclidean algorithm; see modular arithmetic.)
- (Step 3, rewritten, can also be found by finding integer x which causes $d = (x(p\text{-}1)(q\text{-}1) + 1)/e$ to be an integer, then using the value of d (mod $(p\text{-}1)(q\text{-}1)$).)

N and e are the public key, and N and d are the private key. Note that only d is a secret as N is known to the public. Alice transmits the public key to Bob, and keeps the private key secret. p and q are also very sensitive since they are the factors of N, and allow computation of d given e. They are sometimes securely deleted, and sometimes kept secret along with d in order to speed up decryption and signing using the Chinese Remainder Theorem[6].

6. Conclusions

Cryptography is an interdisciplinary subject, drawing from several fields. Before the time of computers, it was closely related to linguistics. Nowadays the emphasis has shifted, and cryptography makes extensive use of technical areas of mathematics, especially those areas collectively known as discrete mathematics. This includes topics from number theory, information theory, computational complexity, statistics and combinatorics. The security of all practical encryption schemes remains unproven, both for symmetric and asymmetric schemes. For symmetric ciphers, confidence gained in an algorithm is usually anecdotal — e.g. no successful attack has been reported on an algorithm for several years despite intensive analysis. Such a cipher might also have provable security against a limited class of attacks. For asymmetric schemes, it is common to rely on the difficulty of the associated mathematical problem, but this, too, is not provably secure. Surprisingly, it is proven that cryptography has only one secure cipher: the one-time pad. However, it requires keys (at least) as long as the plaintext, so it was almost always too cumbersome to use.

References

[1] Douglas Stinson, "Cryptography: Theory and Practice", CRC Press, 1995
[2] W. Alexi, B. Chor, O. Goldreich and C. P. Schnorr. RSA and Rabin functions: certain parts are as hard as the whole. SIAM Jounal on Computing, 17 (1988), 194-209.
[3] H. Beker and F. Piper. Cipher Systems, The Protection of Communications. John Wiley and Sons, 1982.
[4] G. Brassard. Modern Cryptology - A Tutorial. Lecture Notes in Computer Science, vol. 325, Springer-Verlag, 1988.
[5] F. Chabaud. On the security of some cryptosystems based on error-correcting codes. Lecture Notes in Computer Science, to appear. (Advances in Cryptology - EUROCRYPT '94.)
[6] D. Coppersmith (Ed.) Advances in Cryptology - CRYPTO '95 Proceedings. Lecture Notes in Computer Science, vol. 963, Springer-Verlag, 1995.
[7] W. Diffie and M. E. Hellman. New directions in cryptography. IEEE Transactions on Information Theory, 22 (1976), 644-654.
[8] N. Koblitz. A Course in Number Theory and Cryptography (Second Edition). Springer-Verlag, 1994.