Introduction to Cryptography

By Mohan Atreya (matreya@rsasecurity.com)

Summary

This article is the first in a series of articles, which plans to give the reader a bottoms-up introduction to the basics of e-security. The goal of this article is to introduce the reader to the basics of cryptography. Special emphasis will be given to the differences, advantages, and disadvantages of the various methods used in cryptography, without delving too deeply into the mathematical foundations of cryptography. For more information on cryptography, please see RSA Security Labs to reference "Frequently Asked Questions About Today's Cryptography" at http://www.rsasecurity.com/rsalabs/faq/index.html

What is Cryptography?

The word "cryptography" is derived from Greek and when literally translated, means "secret writing." Before the advent of digital communications, cryptography was used primarily by the military for the purposes of espionage. With the advances in modern communication, technology has enabled businesses and individuals to transport information at a very low cost via public networks such as the Internet. This development comes at the cost of potentially exposing the data transmitted over such a medium. Therefore, it becomes imperative for businesses to make sure that sensitive data is transferred from one point to another in an airtight, secure manner over public networks. Cryptography can help us achieve this goal by making messages unintelligible to all but the intended recipient.

Encryption refers to the transformation of data in "plaintext" form into a form called "ciphertext," which renders it almost impossible to read without the knowledge of a "key," which can be used to reverse this transformation. The recovery of plaintext from the ciphertext requires the key, and this recovery process is known as decryption. This key is meant to be secret information and the privacy of the ciphertext depends on the cryptographic strength of the key.

Types of Cryptography

There are two types of cryptographic algorithms: Secret Key Cryptography and Public Key Cryptography.

Secret Key Cryptography:

- This crypto-system uses the same key for both encryption and decryption (this is also referred to as "symmetric" cryptography).
- Both the sender and the receiver need to have the same key in order to communicate successfully.
- Examples: DES, 3-DES, RC4, RC5, etc.
- Advantages:
  - Very fast relative to public key cryptography;
  - Considered secure, provided the key is relatively strong;

    - The ciphertext is compact (that is, encryption does not add much excess "baggage" to the ciphertext);
    - Widely used and very popular.
- <u>Disadvantages</u>:
    - The administration of the keys can become extremely complicated;
    - A large number of keys is needed to communicate securely with a large group of people;
    - Non-repudiation is not possible (see sidebar for a detailed discussion on non-repudiation);
    - The key is subject to interception by hackers.

<u>Public Key Cryptography</u>

- This crypto-system uses one key for encryption and another key for decryption (also known as "asymmetric" cryptography);
- Each user has two keys – one **public** key, which is revealed to all users, and one **private** key, which remains a secret. The private key and the public key are mathematically linked;
- Encryption is performed with the public key and decryption is performed with the private key;
- Examples: RSA, Elliptic Curve Cryptography (ECC).
- <u>Advantages</u>:
    - Considered very secure;
    - No form of secret sharing is required, thus reducing key administration to a minimum;
    - Supports non-repudiation;
    - The number of keys managed by each user is much less compared to secret key cryptography.
- <u>Disadvantages:</u>
    - Much slower compared to secret key cryptography;
    - The ciphertext is much larger than the plaintext, relative to secret key cryptography.

<u>Basics of E-Security</u>

Today's cryptography is more than just encryption and decryption. Cryptography is also closely tied to security. Public key cryptography, for instance, is heavily used for digital authentication purposes—i.e., assuring that communication is from a particular party. In today's digital world, authentication has become as important as privacy of data. In many cases, it is meaningless to encrypt data if the other party cannot be authenticated. Thus, strong authentication is becoming a necessity.

The following table highlights and defines the basic requirements for e-security.

Levels of security	**What is it**	**How to achieve**
Authentication/Verification	Establishes and verifies that the communicating parties are who they say they are.	**Public or secret key cryptography**
Confidentiality	Ensures the protection of sensitive and private data	**Cryptography**
Data Integrity	Ensures that the data has not been altered or manipulated.	**Message Digests**
Non-repudiation	Ensures that information cannot be disowned.	**Message Digest + Digital Signatures**

Achieving Authentication and Confidentiality

As mentioned earlier, a good way to achieve confidentiality and authentication is through the use of cryptography.

All cryptographic algorithms are good for establishing secure and confidential communications. Each is based on solving "hard" mathematical problems. The RSA public key cryptosystem, however, is also good for achieving authentication.

The RSA algorithm is based on two large prime numbers multiplied together to produce a public and a private key. Trying to solve this equation without knowledge of the "key" comes only at great mathematical and computational expense. Because the strength of RSA lies in the fact that it is difficult to factor[1], only by producing the correct key, does the "hard problem" become much easier to solve. Therefore, the foundation of public key cryptography is the public/private key pair exchange, where we can safely assume that when the public key is used to encrypt and a user's private key pair is used to decrypt, the message become decipherable only to the intended receiver.

Key sizes vary. In lay terms, the key can be considered as a number or a set of numbers. For example, the Data Encryption Standard (DES), a secret key cryptographic algorithm, uses a single number, which is in reality a 64-bit key (56-bits effective) and the EI Gamal cipher comprises three numbers.

[1] The RSA algorithm works as follows: take two large primes, p and q, and compute their product $n=pq$; where n is called the modulus. Choose a number, e, less than n and relatively prime to $(p-1)(q-1)$, which means e and $(p-1)(q-1)$ have no common factors except 1. Find another number d such that $(ed-1)$ is divisible by $(p-1)(q-1)$. The values e and d are called public and private exponents, respectively. The public key is the pair (n,e); the private key is (n,d). The factors p and q may be destroyed or kept with the private key.

It is difficult to obtain the private key d from the public key (n, e). However, if one could factor n into p and q, then one could obtain the private key d. Thus the security of the RSA system is based on the assumption that factoring is difficult.

Does Key Size Matter?

Often, people associate the size of the key to the amount of security being applied.

While cryptography does make e-commerce possible by protecting electronic information from prying eyes, the effectiveness of this protection partly depends on the cryptographic size of the key[2]. If the algorithm is inherently strong, then it can be assumed that the larger the key size for the ciphers, the harder it is for a hacker to perform an attack (i.e., a brute-force attack) on the ciphertext.

But applying the right level of security through key size is also somewhat dependent on the value of the data being transferred. The higher the value, the less risks the user is willing to take, the higher the level of security they will want to apply. Often, this means larger keys but larger keys leads to lower levels of performance. Simply put, it takes longer to communicate and in an e-commerce situation, larger keys can severely degrade server performance. There are, therefore, trade-offs, which are traditionally made between the level of security and other factors, like performance.[3] These are issues, which must be addressed in the successful implementation of a cryptosystem or even a good security policy.

Achieving Data Integrity

While symmetric and asymmetric ciphers are great for establishing confidentiality and some forms of user authentication, they cannot guarantee data integrity. We must use alternate mechanisms to ensure data integrity through an added layer of security. Message digests have been designed specifically to solve this problem. Mathematically, message digest algorithms are one-way functions. Calculation of the digest is infinitely easier compared to the process of retrieving the message from the digest. Message digest algorithms generally possess the following characteristics:

- They take in a variable length input and generate a fixed length output called the “hash” or a message digest (for example, the MD5 algorithm generates a 128-bit digest and the SHA-1 algorithm generates a 160-bit digest);
- It is not computationally feasible to calculate the message based on the digest;
- It is not computationally feasible to find two messages which will generate the same digest (this feature is also called “collision resistance”).

Thereby, message digests achieve data integrity by applying complex math to data to ensure that this data has not been tampered with on route to its final destination. Again, data integrity is needed only in situations where the value of the data being transferred is high enough to warrant the added layer of security versus the risks of exposing the digital data.

[2] There are many other factors to consider such as Pseudo Random Number Generators, which will be covered in the next installment of this series.

[3] RSA Security now offers unprecedented levels of performance for public key cryptosystems. Working with partners like Intel, Compaq, HP and Sun, RSA is able to focus on performance and security implementation issues which keep developers one step ahead in the development stage.

Implementation Issues

From the above discussion, it is quite evident that just one cryptosystems will not solve every problem. More than ever, the way security is implemented using cryptography against users' security requirements is the issue. Most of the systems in use today employ a hybrid system, for example making use of public key cryptography to securely transport the secret key as well as a message digest to ensure data integrity. This makes good security sense. Think about password-based authentication. At one time this was an acceptable level of security, now it is viewed as insecure because passwords are easily subject to interception and replay attacks by hackers. Now it is commonly accepted that two-factor authentication is much more acceptable taking authentication to a much higher level based on something you ***know*** (your PIN) as well as something you ***have*** (like a the SecurID token).

So too should a layered security approach now be closer to helping you achieve security goals.
It is important to understand that public-key cryptography was never meant to completely replace secret-key cryptography. Often, it is used to supplement secret key cryptography, thus making symmetric cryptography more secure and helping users to achieve their unique needs.

Conclusion

In this article, we discussed the inner workings of symmetric ciphers, asymmetric ciphers, and the message digest algorithms. We also discussed how symmetric ciphers, asymmetric ciphers and message digests can be used together to enable e-security. The concept of strong authentication was highlighted as the basis of all security. We also discussed non-repudiation in the digital world and how it compares with non-repudiation in the paper-based world (see sidebar).

In the next article, we will discuss how public key cryptography has been used to build some *de facto* standards (PKCS) that ensure interoperability between different systems.

About the Author
Mohan Atreya is a Technical Consultant with RSA Security. He has advanced degrees from National University of Singapore and Nanyang Technological University.

About RSA Security Inc.
RSA Security Inc., The Most Trusted Name in e-Security™, helps organizations build secure, trusted foundations for e-business through its RSA SecurID® two-factor authentication, RSA BSAFE® encryption and RSA Keon® digital certificate management systems. With more than a half billion RSA BSAFE-enabled applications in use worldwide, more than six million RSA SecurID users and almost 20 years of industry experience, RSA Security has the proven leadership and innovative technology to address the changing security needs of e-business and bring trust to the new, online economy. RSA Security can be reached at www.rsasecurity.com.

A Message to Developers
The RSA BSAFE family of toolkits provides you with all the components you need to make your applications safe and secure. As a developer, you can save many months of development and testing, thus allowing you to focus on your application development and roll out your application with confidence. The BSAFE family comprises the following toolkits:

The BSAFE family comprises the following toolkits:

Core Functionality	BSAFE Toolkit Details
Core Cryptographic Toolkits	BSAFE Crypto-C & BSAFE Crypto-J
Public Key Infrastructure (PKI) Toolkits	BSAFE Cert-C & BSAFE Cert-J
Protocol Level Toolkits	BSAFE SSL-C & BSAFE SSL-J (SSL protocol for point-point security) BSAFE S/MIME-C (S/MIME Protocol for secure messaging) BSAFE WTLS-C (Wireless Transport Layer Security for WAP)

SideBar
Non-repudiation in the digital world
Signatures are typically used for non-repudiation purposes in the paper-based world. Can a digital signature be considered equivalent to a traditional signature in the paper-based world?

The answer to this question is not very straightforward. There is a difference in terminologies and the underlying meanings of "non-repudiation" in the paper-based world and the digital world.

Some of the significant differences include the following:

- In the paper-based world, there is a concept of "***witnessing***", which reduces the ability of the "***signer***" to repudiate the signature in the future.
- In the *paper-based world*, if the "signer" disputes the signature, then the responsibility falls on the *"**receiving party**"* to non-repudiate the signature. Whereas, in the *digital world*, due to the concept of "Trusted Third Parties", the responsibility shifts from the *"**receiving party**"* to the *"**sender**"* or in some cases denies the *"**signer**"* the right to repudiate a digital signature.

Paper-based World
The "signer" can repudiate a signature if

- The signature claimed to be the "signer's" is a forgery.
- The signature claimed to be the "signer's" is not a forgery, but was obtained from the "signer" via (a) unconsionable means and/or, (b) fraudulent means and/or, (c) under conditions of undue duress or under unfair bargaining influence.

Digital World
Cryptographic techniques are utilized which usually prevents the "signer" from denying having performed the transaction. This denial of right of being able to repudiate a digital signature is currently hotly debated.

A "***Trusted Third Party***" such as a Certificate Authority only establishes a relationship between the "signer" and his/her "public key" which are part of the certificate. It does not perform any "witnessing" services during the actual process of signing. We also have to take into account the real problem of "**identity thefts**" resulting from "**private key thefts**" or "illegal usage of private keys".

This brings us back to the issue of Strong Authentication. Private Key thefts and illegal usage resulting in e-forgeries can be reduced or stamped out completely if the "***signer***" is strongly authenticated to the private key store. Password based (one-factor) authentication is just not sufficient for this purpose. As a consequence, two-factor authentication to the private key store is strongly recommended.