

Flexible Cryptographic Component Design for Secure Web Applications

Tae Ho Kim *, Jong Jin Kim **, Chang Hoon Kim *, Chun Pyo Hong *

* Department of Computer and Communication, Daegu University
15 Naeri, Jinryang, Kyungsan, Kyungbuk, 712-714, South Korea
{thkim, chkim}@dsp.daegu.ac.kr, cphong@daegu.ac.kr
Tel : +82-53-850-6574, Fax : +82-53-850-6629

** Terrawave ,105, Suntec-city, 513-15, Sangdaewon-dong,
Jungwon-gu, Sungnam, Gyeonggi, 462-725, South Korea
jjkim001@naver.com
Tel : +82-2-563-2525, Fax : +82-31-777-1138

***Abstract** - Although Internet serves many contents and services, it has serious problems of security: the invasion of privacy, hacking and etc. To prevent these problems, two implementations have been presented: Hardware and Software implementations of cryptographic algorithms. Hardware implementations of cryptographic algorithms provide much faster than software implementations. However, Software implementations are much flexible and low-cost. Many software-approaches have been presented. But, they are not suitable for internet usage. This is because they provide a single cryptographic algorithm and have a large executable code. In addition, they have not an automatic version management function. In order to overcome these problems, this paper presents a new crypto-ActiveX module for secure internet applications. The presented crypto-ActiveX module is designed by using Microsoft (MS) Active Template Library (ATL) and supports Component Object Model (COM) interface. Therefore our crypto-ActiveX module has a small executable code size, a fast COM code, and automatic version management function. Furthermore, since the proposed crypto-ActiveX module supports COM interface, any other software can easily use our crypto-ActiveX functions easily. Therefore, the proposed crypto-ActiveX module is suitable for secure internet applications.*

Keywords: Cryptography, Internet Security, ActiveX, ATL, COM

1 Introduction

Although Internet serves many contents and services, it has serious problems of security: the invasion of privacy, hacking and etc. Generally, designing and implementing of cryptographic systems for Internet are very difficult and it requires many fundamental principles of cryptography [1]. Many software implementations of cryptographic schemes have been presented. But they are not suitable for Internet usage. This is because they provide a single cryptographic algorithm and have a large executable code [2-7]. In addition, they have not an automatic version management function. In order to overcome these problems, this paper presents a new crypto-ActiveX module that requires least knowledge of cryptography and it includes secret-key, public-key and cryptographic hash functions for secure Internet

applications. The presented crypto-ActiveX module is designed by using ATL and supports COM interface. Therefore our crypto-ActiveX module has a small executable code size, a fast COM code, and an automatic version management functions. Furthermore, any other software can easily use our crypto-ActiveX functions, since the proposed crypto-ActiveX module supports COM interface. The remainder of this paper is organized as follows. In section 2, we give a brief overview of cryptographic systems and MS ActiveX related technologies. Implementation issues in the development of cryptographic systems are discussed in section 3. In section 4, we design crypto-ActiveX module that supports COM interface, and describe implementation results. Finally, concluding remarks are given in section 5.

2 Background

2.1 Overview of Cryptographic Systems

Cryptography is a branch of mathematics based on the transformation of data. Cryptography deals with the transformation of ordinary text (also called plaintext) into coded form (also called cipher-text) by encryption and the transformation of cipher-text into plaintext by decryption. Three basic cryptographic functions are required to implement crypto-ActiveX module, i.e., secret-key, public-key cryptographic algorithm and hash function. We summarize the above cryptographic functions as follows:

- Public and private key cryptography

In secret-key (also called symmetric-key), the same key is used for both encryption and decryption. That is, all parties participating in the communication share a single key. In public-key (also called asymmetric-key), there are two keys: a public key and a private key. The public key used for encryption is different from the private key used for decryption. The two keys are mathematically related, but the private key cannot be determined from the public key. The secret-key has an advantage of a high-speed messages en/decryption while the public-key has a feature of simple key-management. These two schemes provide confidentiality and authentication for insecure communication channels.

- Hash function

Hash function like a checksum compresses the bits of a message to a fixed-size hash value in a way that distributes the possible messages evenly among the possible hash values. A cryptographic hash function does this in a way that makes it extremely difficult to come up with a message that would hash to a previously computed hash value. It provides data integrity.

2.2 Overview of MS ActiveX Related Technoledges

- COM

COM is a technology that allows objects to interact across process and machine boundaries

as easily as within a single process. COM enables this by specifying that the only way to manipulate the data associated with an object is through an *interface* on the object. In other words, COM is a standard protocol for connecting objects together.

- ATL

ATL is the Active Template Library, a set of template-based C++ classes with which you can easily create small, fast Component Object Model (COM) objects. A template is somewhat like a macro. As with a macro, invoking a template causes it to expand (with appropriate parameter substitution) to code you have written. However, a template goes further than this to allow the creation of new classes based on types that you pass as parameters. These new classes implement type-safe ways of performing the operation expressed in your template code.

- OLE and AtiveX

Object Linking and Embedding (OLE) is a feature of Microsoft Windows which enables one application to load and send messages to any application registered with the operating system as an OLE class or server. An ActiveX control is essentially a simple OLE object that supports the *IUnknown* interface. It usually supports many more interfaces in order to offer functionality, but all additional interfaces can be viewed as optional and, as such, a container should not rely on any additional interfaces being supported. By not specifying additional interfaces that a control must support, a control can efficiently target a particular area of functionality without having to support particular interfaces to qualify as a control. As always with OLE, whether in a control or a container, it should never be assumed that an interface is available, and standard return-checking conventions should always be followed. It is important for a control or container to degrade gracefully and offer alternative functionality if a required interface is not available.

3 Implementation Issues

3.1 Hardware vs. Software Solutions

The trade-offs among security, cost, simplicity, efficiency, and ease of implementation need to be evaluated. Cryptography can be implemented in hardware, software - each has its related costs and benefits. Historically, software has been less expensive and slower than hardware. In addition, software is easier to modify or bypass than equivalent hardware products. The advantages of software solutions are in flexibility and portability, ease of use, and ease of upgrade. Web applications change very fast. So, cryptographic module must be easily modifiable. Therefore, we implement cryptographic module in software.

3.2 Public vs. Secret Key Cryptography

The primary advantage of public-key cryptography is increased security and convenience: private keys never need to be transmitted or revealed to anyone. In a secret-key system, the secret keys must be transmitted (either manually or through a communication channel). There may be a chance that an unauthorized individual can access the secret keys during their transmission. The primary advantage of secret key cryptography is speed. There are popular secret-key encryption methods that are significantly faster than any currently available public-key encryption method. Alternatively, public-key cryptography can be used with secret-key cryptography to get the best of both worlds: the security advantages of public-key systems and the speed advantages of secret-key systems. The public-key system can be used to encrypt a secret key that is used to encrypt the bulk of a file or message. In addition, since web applications are various, we do not know sending Message is simple characters or large file. Therefore, we must implement both public and private cryptography system.

3.3 MFC vs. ATL

The MFC (Microsoft Foundation Class Library) is an "application framework" for programming in MS Windows. The MFC

provides much of the code necessary for managing windows, menus, and dialog boxes, and so on. The MFC framework is a powerful approach that lets you build upon the work of expert programmers for Windows. However, the MFC based development is not suitable for Internet applications. Because the MFC contains many built-in functions, executable codes generated by the MFC are very large. On the other hand, the ATL provides a compact and fast COM code, since it does not use all of MFC's functions. Therefore, if we want a compact code size, the ATL is a better choice rather than the MFC.

4 Implementation of Crypto-ActiveX Module

4.1 Crypto-ActiveX Module Design

We design crypto-ActiveX module for secure Internet applications with consideration of issues in the section 3. Overall architecture of crypto-ActiveX module is described in figure 1. Each module is designed by using the ATL technology. As shown in figure 1, Crypto-ActiveX module is the main function block. It processes data or files and provides external interface as ICryptoCOM. It consists of seven sub-function blocks, i.e., DES, AES, and SEED for symmetric key en/decryption, RSA for public key en/decryption, and SHA-1 and SHA-2 for hash function. When specified cryptographic function is requested, the Crypto-ActiveX calls the function of other modules for en/decryption and hashing data or file. The CryptoActiveX module has connections of other modules while the other modules have not any connection. If we need to modify one or two algorithms or functions, we only modify the corresponding module. In addition, we have only to download the modified module to update.

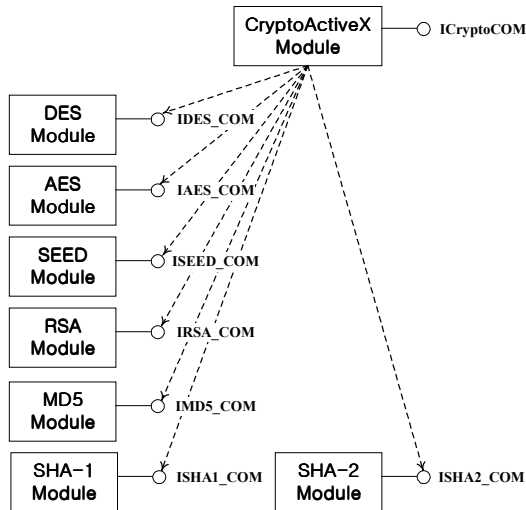


Figure 1. Overall architecture of crypto-ActiveX

4.2 COM Interfaces Design

The crypto-ActiveX module described in figure 1 has 3 types of interfaces. Each module has only one of interface types. Interface types are shown in figure 2.

<<Interface Type>> ICryptoCOM	<<Interface Type>> ICIPHER_COM	<<Interface Type>> IHASH_COM
Encrypt() Decrypt() Hash() Verify()	Encrypt() Decrypt()	Hash() Verify()

Figure 2. Three types of COM Interfaces

First, ICryptoCOM interface type has four methods. Only CryptoActiveX module has this interface. This interface has four methods to call methods of other Interface type for encryption, decryption, hashing and verifying. Second, ICIPHER_COM interface type is used cryptographic algorithms (e.g. IDES_COM, IAES_COM, IRSA_COM and etc.) and has two methods for en/decryption. Lastly, IHASH_COM interface type is used cryptographic hashing and verifying.

4.3 Implementation Results

We implemented crypto-ActiveX control using the ATL. Figure 3 shows user-interface of the crypto-ActiveX.

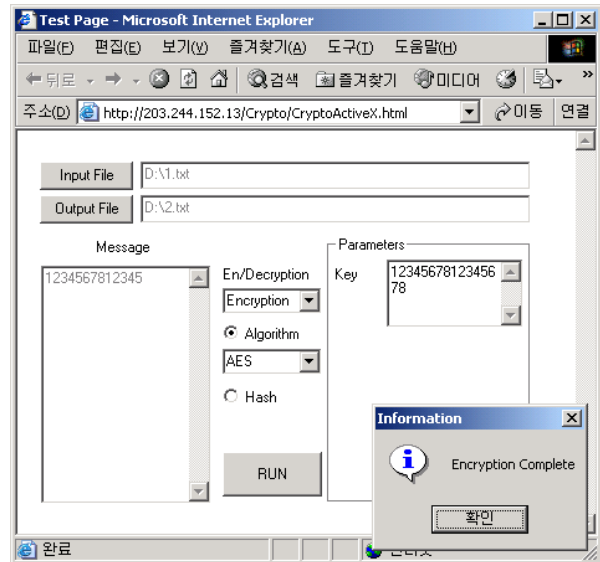


Figure 3. User-interface

As described in figure 3, each input file and output file button selects input or output file name. At bottom-left, the message edit-box is used to show the contents of input files, and en/decryption combo-box is used to select whether encrypt or decrypt files. The algorithm combo-box, which contains DES, SEED, and RSA as cryptographic algorithm, is used for selecting a specific cryptographic algorithm to perform. The hash radio-button is used to perform hash functions. Although the combo box window for hash functions selection is not shown in figure3, if the hash radio button is selected, the algorithm combo-box disappears and the combo-box for hash algorithms is shown. The hash combo-box has five algorithms, i.e., MD5, SHA-1, SHA-1, SHA-2(256), SHA2-(384), SHA-2(512). Finally, we use the key edit-box for input key.

Table 1. File size when the ATL is used

File Name	File Size
MFC42.DLL	1,015,859 Bytes
MFC42.DLL (CAB format)	461,003 Bytes
Our module+ATL.DLL (CAB format)	268,972 Bytes
Our module (CAB format)	231,819 Bytes

We summarize implementation results in table 1 from the viewpoint of file size. From

table 1, we can see that the file size using the ATL is significantly smaller than that of using the MFC..

5 Conclusions

In this paper, we have presented the new crypto-ActiveX module for secure Internet applications. The presented crypto-ActiveX module is designed by using the ATL and supports COM interface. As mentioned in the section 4, the file size is significantly small when the ATL is used. In what follows, our crypto-ActiveX module has a small executable code size, a fast COM code, and automatic version management function. In addition, since the proposed crypto-ActiveX module supports COM interface, any other software can use our crypto-ActiveX functions easily. Therefore, the proposed crypto-ActiveX module is suitable for secure Internet applications.

References

- [1] William Stallings, "Cryptography and Network Security Principles and Practices Third Edition", Prentice Hall, 2003.
- [2] Eli Biham, "A Fast New DES Implementation in Software", LNCS, Vol. 1267, pp. 260-272. Springer-Verlag, 1997.
- [3] M. Brown, D. Hankerson, J. Lopez and A.J. Menezes. "Software implementation of NIST elliptic curves over prime fields.", CT-RSA 2001, LNCS, Vol. 2020, pp. 250-265, Springer-Verlag, 2001.
- [4] D. Hankerson, J. Lopez and A. Menezes, "Software Implementation of Elliptic Curve Cryptography over Binary Fields.", CHES 2000, LNCS, Vol. 1965, pp. 1-24, Springer-Verlag, 2000.
- [5] S. Halevi and H. Krawczyk. "MMH: Software message authentication in the Gbit/second rates." Proceedings of the 4th Workshop on Fast Software Encryption, LNCS, Vol. 1267, pp. 172-189, Springer-Verlag, 1997.
- [6] G. Tsudik, "Message Authentication with One-Way Hash Functions.", INFOCOM '92, Vol. 3, pp. 2055 - 2059, IEEE, 1992.

[7] Brent R. Waters and Edward W. Felten and Amit Sahai, "Receiver Anonymity Via Incomparable Public Keys.", Proceedings of the 10th ACM Conference on Computer and Communication Security (CCS-03), pp. 112-121, ACM, 2003.