Secure HIC System Based on WWW

Tae-Gap Kim*, Jun-Hyuk Choi**, Byung-Do Go**, Jae-Cheoul Ryou*
* Chungnam National University
** Korea Electronics and Telecommunications Research Institute

Abstract

World Wide Web(WWW) is a total solution for multi-media data transmission on Internet. Because of its characteristics like ease of use, support for multi-media data and good graphic user interface, WWW has extended to cover all applications. The HIC system is a information sharing system on WWW. It's main function is to provide information sharing mechanism among its users by file uploading and downloading. Since the transmitted data is a private or secrete information, it is important to protect the information from unauthorized users. The secure HIC system uses conventional cryptographic methods to protect the transmitted data. It supports the encryption and authentication of data. For encryption of data IDEA(International Data Encryption Algorithm) is used and for authentication mechanism MD5 hash function is used. Since secure HIC system is used by its user group, conventional crypto system is efficient in managing its user's secure interactions. However, conventional crypto system can brings about some critical problems on sharing of same key and data transmission between client and server. For example the risk of key exposure and the difficulty of key sharing mechanism. To solve these problems, the secure HIC system provides secure key sharing mechanism and management policies. As the commercial use of WWW increases, security mechanisms like the secure HIC will become widely popular.

1. Introduction

Internet is the biggest inter-communication network in the world. It has several millions of hosts because of its useful characteristics for example, openness, extensibility, and commercial use, etc. Hence it has become the focus of attention world. The introduction of GUI(Graphic User Interface) based web browsers such as netscape and mosaic have extended the notion of Internet to World Wide Web(WWW). WWW supports various multi-media data including normal data, voice, sound, image, video data, etc and provides easy access mechanism to its users. Because of such capabilities, WWW is considered as a total solution for almost all types of applications.

As the use of Internet increases, its users demand ever faster response and integrated service. Practically, the transmissions of multi-media data or real time interactions require considerable transmission capacity and the users want to be served by one integrated channel for various services. The development of high speed transmission media like fiber optic resulted in the introduction of B-ISDN(Broadband Integrated Service Digital Network).

Since a B-ISDN provides several Mbps transmission capability, it can cover the transmissions of most kinds of data. This fact coincides with the request for multi-media data and B-ISDN will be widely used in future. In Korea, such a trend exists and the government hastened to construct B-ISDN. It is called Han-BISDN(Highly Advanced National Broad band Integrated Service Digital Network).

Through the evolution of inter communication technologies like WWW or B-ISDN, the interactions and information sharing among its users and groups become an important issue. To support interactions and information sharing in specific user groups, it is required to collect and store the information generated in that group and to publish or provide that information only to its authorized members. For this management task, HIC(Han B-ISDN information Center) is developed. But the lack of security in WWW makes it impossible to protect such an information sharing mechanism. It provides a background for the design of a secure HIC.

In this paper, we propose a mechanism for information sharing and management such as file transmission from client to server, and vice versa. After analyzing security problems in WWW, we propose a solution for it. It is called secure HIC system. The secure HIC system is based on conventional cryptography and key management mechanisms. In section 2, we describe what is HIC system and the general security mechanisms in WWW. We describe the overall structure of a secure HIC system including key management mechanisms in section 3. Section 4 describes detail implementation features of the secure HIC system. Finally we make conclusions in section 5.

2. What is a secure HIC system?

The data transmission on WWW represents a typical client and server model. The client requests data and the server responds to that request. Figure 1 describes data transmission on WWW.

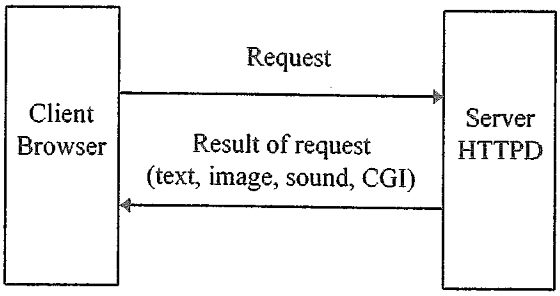


Figure 1. General data transmission on WWW

As described in Figure 1, a general data transmission mechanism on WWW is very simple. A client browser requests data from the server, normally using GET or POST method, and the server's HTTPD services the client's request. It finds the target document from its data base and sends the result to the client. If the client's request is to execute a program, which is called CGI(Common Gateway Interface), the server executes the target program and returns the program's output. In case of normal WWW data transmission mechanism, the client is only a receiver of data sent from the server. But in the HIC system, a client can also send data to the server. It is a main feature of the HIC system. Such a feature can be used in various applications. Figure 2 describes such a use of HIC system.

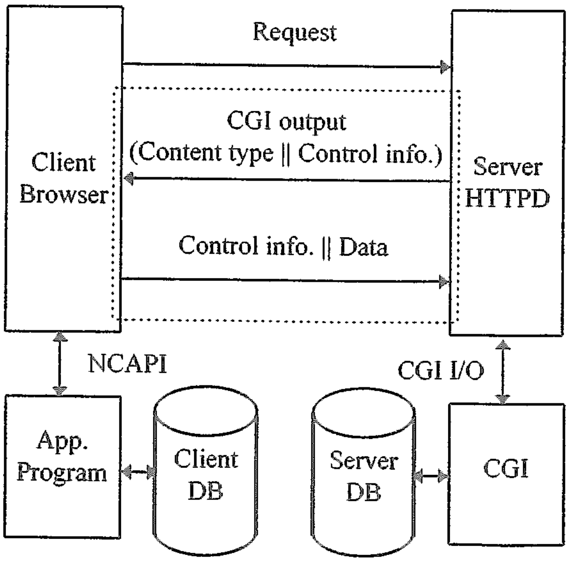


Figure 2. Data transmission in HIC system

To send data, the client uses an application program which is invoked by the CGI's output. When the client browser receives the special content type from the server, it invokes an appropriate client application according to the content type. The client application program reads necessary data from the client's data base and delivers it to the browser again. The control info. is used to manage the data transmission. It contains a information about the reading or writing status of each part. The detail description is given in section 3. If the HIC system is used without security features in its applications, it may pose serious security problems. So it is better to consider adding security mechanisms to the HIC system. In the rest of this section, we will describe the general problems in WWW and security mechanisms that can be applied to an HIC system.

Thanks to its simple structure and being a part of the UNIX kernel, TCP/IP forms the basis of Internet[1]. Since TCP/IP does not consider security problems, there is no policy to prevent unauthorized access in the network. The HTTP(Hyper Text Transfer Protocol), being a base protocol of WWW, also depends on TCP/IP. As such there is also no security mechanism in the HTTP. This means that it cannot be used if there is a need for security. For this reason, EIT(Enterprise Integration Technologies) proposed a security enhanced version of HTTP(S-HTTP)[2] and Netscape company also proposed a security enhanced protocol, SSL(Secure Socket Layer)[3]. Since S-HTTP is a high level security enhanced mechanism, it can be used easily with other security applications. On the other hand, SSL is a more general and low level security mechanism, defined for the network layer.

Secure HIC system is designed to ensure secure document transmission using WWW protocols like S-HTTP and SSL. Its main mechanism is based on conventional cryptography. We can consider the use of a public key system which is a more general mechanism in a network environment, but because the use of HIC is restricted to an authorized user group, it can be efficiently managed only using conventional cryptography. Basically, it is desirable to have same key between HIC server and client and to manage their key sharing procedures. Using secure HIC system, client can push(get) his document to(from) HIC server's data base in secure channel.

To support general WWW security issues, secure HIC system has to satisfy following requirements.

1) *Data Confidentiality*

No one, except client and HIC server, must be able to read data. Such a feature can be easily implemented using Conventional crypto system with key sharing mechanisms.

2) *User Authentication*

Server or Client must confirm each other. In other words, server must distinguish between authorized client's request and unauthorized client's request. Similarly, client must confirm if the received acknowledgment is sent from the authorized server.

3) *Message Authentication*

Client or server must confirm if the received data is not modified by unauthorized user in their transmission time. This requirement can be implemented using the message digest mechanism.

The satisfaction of these requirements must be followed by the system security and access control of data. It is no use preventing unauthorized access in communication network level if the server system is not secure against unauthorized user's access.

3. Design of secure HIC system

Figure 3 describes the overall structure of a secure HIC system. The client's web browser (specially netscape) sends a request to HIC server's HTTPD. According to the server's acknowledgment, which is the type of specific application, the web browser invokes the client application program and communicates with it. For the interaction between the browser and the application program, NCAPI(Netscape Client Application Programming Interface) is used. To establish a secure communication channel, the client program encrypts all sending data. At the same time a hash value, which is a result of message digest, is generated. When all data to be sent is generated, the application program passes it to the browser using NCAPI and then the browser sends the passed data to the server using HTTP. Both GET and POST methods can be used for this but POST method is preferable because of its security advantages. For encryption of data, IDEA(International Data Encryption Algorithm)[4] module is used. MD5(Message Digest Algorithm)[4] is used to generate the hash value for message digest. The server's HTTPD receives the data sent. According to the client's request, the HTTPD invokes a corresponding CGI(Common Gateway Interface)[5] program which is generally used to serve various client's requests in WWW. The invoked CGI program reads the encrypted data and hash value through standard input channel(in the case of POST) or environment variables(in the case of GET). Since CGI program has IDEA and MD5 module, the encrypted data can be decrypted, and the server can confirm user and message authentication using hash value. The same method can be applied when the server sends data to the client. But this time, the server encrypts the data and the client decrypts it. In Figure 3, we can see that the secure HIC system makes use of original HTTP, not modified, but for security enhancement cryptography method and application program are used. The client application program and CGI program are used to help web browser and HTTPD.

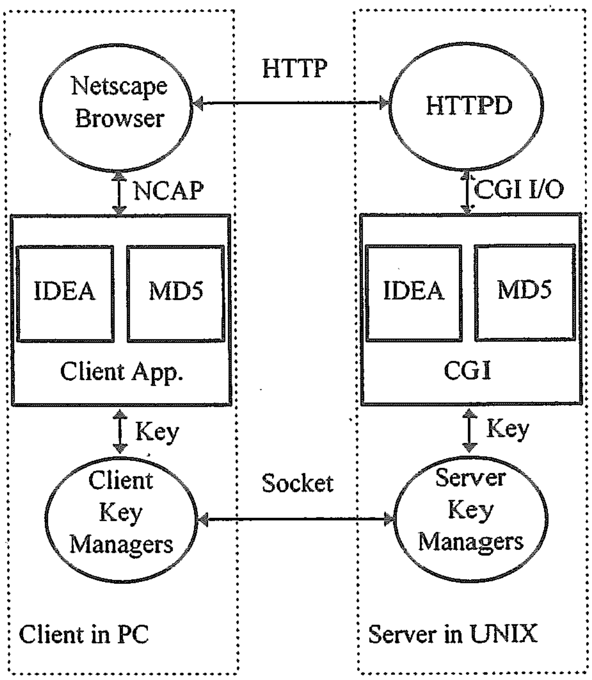


Figure 3. Overview of Secure HIC system

The client's key manager is used to manipulate the keys which are used in the HIC system. At this point we must emphasis one important aspect of the system: key sharing between client and server. It is a premise of conventional cryptography. In the secure HIC system, the key managers accomplish the sharing procedures. Their main functions are sharing of same key and updating of their key database. It is therefore natural that key management mechanisms be independent of the HTTPD and the browser. So they are implemented using socket interface. The detail description of key managers and key server is explained later in this section.

3.1 Key definitions

In HIC system there are five different keys. Each of them is used for encryption or decryption of data and key. The detail description is as follows:

1. ***Base key(Kb)***

It is a 64 bit key for secure communication between client and server. It must be shared by both client and server. Base key is not used directly for communication due to its secrecy but used for only making and sharing of a temporary session key which is a real key for encrypted communication. The server must have all base keys of its clients in a HIC key file. The HIC key file's format is "*ID:EncryptedKey:*". The manager of HIC system must distribute each client's base key using floppy disk to decrease chances of exposure. The procedure for distribution and confirmation of base key is described later in this section. The base key is encrypted with the client key in client key file, but it is encrypted with server key in the HIC key file. IDEA module is used for their encryption and ID is a format of e-mail addresses. Figure 4 describes the mechanism to generate a base key. The reason that such complex mechanism is used to generate a client's base key is to increase the difficulty of guessing. In Figure 5, the key and initial value is a kind of random number. This mechanism can be applied with other random number generations.

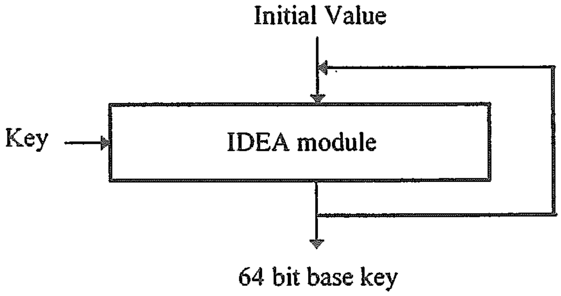


Figure 4. Generation of client's base key

2. *Server key(Kv)*

The server key is a 128 bit key for encryption of client's base key. It is stored in the HIC key file in encrypted form using UNIX password mechanism with root ID. The format of server key is *"root:EncryptedKey:"*. The *EncryptedKey* is of 208 bit length. Since the server key is 128 bit, it is required to apply the password mechanism, which is for 64 bit key, twice. The UNIX password mechanism reads 64 bit input and generates 104 bit encrypted output which consists of 16 bit salt (a kind of key) and 88 bit output[6]. When someone tries to retrieve client base key in HIC key file, he must know the server key. For example, when the HIC system manager wants to update the HIC key files contents, he must pass the server key checking routine. As the UNIX password mechanism is based on 1-way hash function, *crypt*, it can't be decrypted with key(specially the salt). The 128 bit server key size is chosen because of two reasons: one reason is that compared to 64 bit DES(Data Encryption Standard) key it is more secure and the other reason is convenience of conversion to IDEA key.

3. *Message key(Km)*

The message key is used to generate session key which is real key of secure communication. Message key consists of a 64 bit random number and is combined with base key to make a session key. Since the message key is a temporary key it does not need to be stored.

4. *Session key(Ks)*

The session key is a real key used for encryption of sending data and decryption of receiving data. It must be shared between both client and server. When a client wants to send a file to its server or get a file from the server, a session key is generated. For generation of a session key, a base key and a message key are needed. The procedure of making session key is as follows :

$$Ks = Kb1..16 \,||\, Km1..16 \,||\, ... \,||\, Kb49..64 \,||\, Km49..64$$

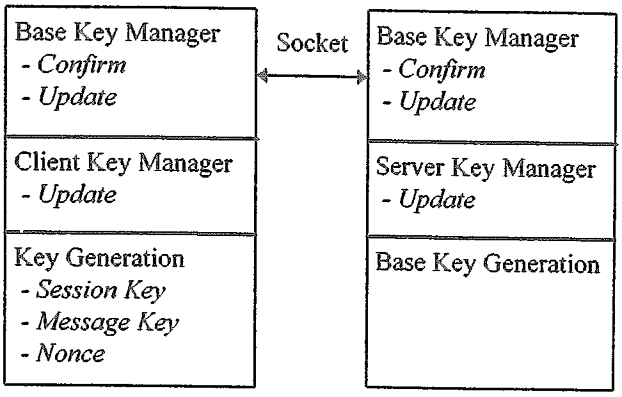
Each part(normally 16 bit) of Kb and Km is appended in turns until all of 64 bit base key and message key is appended. Since a session key has 128 bit(64 bit Kb + 64 bit Km) key size it can be used as an IDEA encryption, decryption key. As a session key must be shared by both client and server, it must be sent to the server in a secure channel, made using encryption and message digest. The encrypted session key can be successfully decrypted, because the server can retrieve all client's base key using its server key. After the client and the server know the session key, the encrypted data transmission is started. The detail description of this is described in section 4.

5. *Client key(Kc)*

It is important to protect the base key from the unauthorized users. For this reasons of security and convenience of delivery, the HIC manager distributes the client base key using floppy disk. But this is not enough. Usually a client does not keep in mind the needs of security of his own key. So there is another mechanism called client key which is used to encrypt the client's base key. Using the 128 bit client key, the client's base key is encrypted and stored in client key file. So the client must know the his client key to retrieve the base key. There is no reason that the client must know the base key. For checking the validity of client key, the base key's hash value is stored with encrypted base key.

3.2 Key managers

To manage the 5 HIC keys described in 3.1, various key managers are designed. The key managers provide a convenient and efficient access interface to the HIC's five basic keys for the client and the HIC manager. To increase the usability of programs(independent from HTTPD), the socket based design mechanism is considered. The communication between key managers is accomplished in a secure channel.



Client Key Managers Server Key Managers

Figure5. Key managers

Figure 5 describes the overall structure of key managers. The only key manager which is related with the base key uses a socket interface to confirm or update the base key between client and server. Other modules update the key file or generate the keys described in section 3.1. The rest of this section describes the key managers.

1. ***Base Key Generation***

The base key generator is normally used by HIC manager. It generates a client's base key and stores it in the HIC key file in encrypted form. Before it generates client's base key, it requests the server key to check the authority of the user. If the authorization check is passed, the user can generate a base key, and it is encrypted with entered server key. Like a generation of client's base key, the server key can be changed using this module, but another mechanism is applied. IDEA module is used to encrypt the base key.

2. ***Base key confirm***

After a base key is delivered to the client, the client may want to test his base key. For such use, base key confirm module is designed. Figure 6 describes the base key confirm procedure.

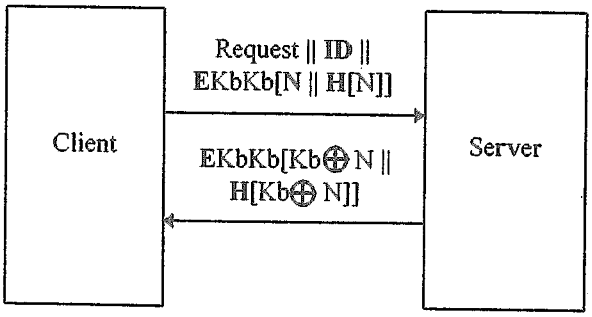


Figure 6. Base key confirm

In Figure 6, KbKb means the repeat of a base key. Such repeat of base key is to make a 128 bit IDEA key using 64 bit key size. N is a nonce which is a kind of random number and used only once. To make this value, the mechanism described in Figure 5 can also be used.

3. ***Base Key Update***

Because the base key is a source of the secure communication between client and server, its secure maintenance is a critical problem. It is desirable to update base key regularly. Figure 7 describes a base key updating procedure.

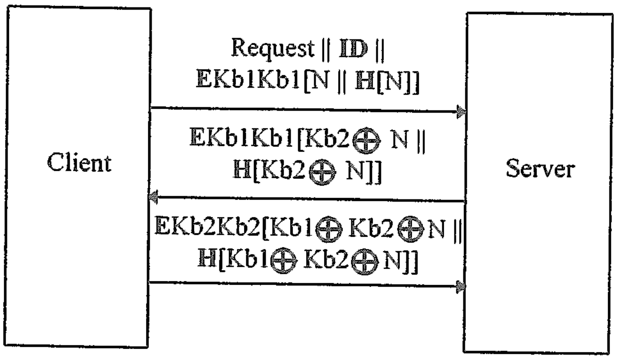


Figure 7. Base key update

The client sends a request with ID, encrypted nonce(N) using current base key(Kb1) and the hash value. The compound of Kb1 and nonce gives key server an identification of valid client's request. If the received message can be successfully decrypted, the server makes a new base key(Kb2) and with received nonce encrypts it again. Then the client can extract the Kb2 using Kb1 and nonce. If the procedure ends here, the base key consistency between client and server may be broken when there is an error in transmission media or unauthorized update of messages. To prevent such inconsistency, client sends an acknowledgment which is encrypted using new base key. If the received acknowledgment is correct then the key updating is complete, otherwise the key server marks that client's base key is inconsistent. If the client base key is marked inconsistent, all requests from that client are rejected and the base key must be reassigned to the client by the HIC manager.

The client key update module updates the client key which is used to encrypt the base key stored in the client key file. The base key is stored in encrypted form with client ID of server's. The server key update module updates the root's server key. Initially, it makes the encrypted server key using the UNIX password mechanism. Later, it modifies all the list of client's base keys with a newly entered server key. However in this case, all applications related with HIC key file must be stopped for the consistency of the base keys and the contents of the HIC key file must be regenerated. This requires serious consideration. The message key and nonce generation modules have similar mechanism to the base key generation which generates a 64 bit random number.

4. Implementation

Secure HIC system is a typical client and server model. The client part is implemented on PC(Personal Computer) and the server part is implemented on a UNIX system. We use C++ language which is a representative general purpose object oriented language. The HIC server part is based on OSF Motif environment to support GUI and gcc compiler is used to make executables. The server key manager can serve maximum 10 concurrent requests and for general usage, socket interface is supported. In the case of client program, it interacts with PC netscape browser. It is necessary to distinguish normal reply and special reply from the server to invoke the helper applications in netscape, so special content types are used. With this type, client netscape browser invokes the client application programs. For interactions between browser and application program, NCAPI is used. Like a server part, the client application is based on GUI, typically MS windows environment.

Now, we will describe the detailed design specifications of secure HIC system. It is divided two parts: session key sharing and encrypted data transmission. As described in section 3, the base key is not used in encrypted data transmission, instead temporary session key is used. Figure 8 describes session key sharing mechanism. When a client wants to send or receive the data, he sends a request to the server and the server's HTTPD invokes a appropriate CGI program. The CGI program generates the reply containing a special content type. With this special content type the client browser can invoke a client application program. This mechanism depends on netscape's helper application facility.

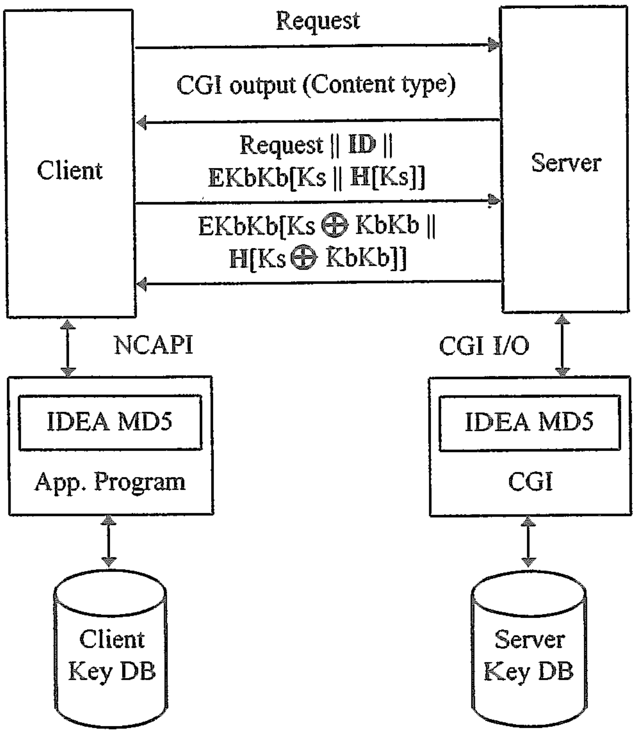

Request
CGI output (Content type)
Request || ID || EKbKb[Ks || H[Ks]]
EKbKb[Ks ⊕ KbKb || H[Ks ⊕ KbKb]]
Client
Server
NCAPI
CGI I/O
IDEA MD5
App. Program
IDEA MD5
CGI
Client Key DB
Server Key DB

Figure 8. Session key sharing mechanism

After the client application is invoked, the communication between the browser and client application is performed using NCAPI. Initially, the client application program makes a session key using the client key manager and encrypts the session key and its hash value with its base key using IDEA and MD5 modules. After the session key generation is complete the application program sends it to browser using NCAPI. Then the client browser sends the session key to the server, practically it was requested by NCAPI facility. Finally the server receives the encrypted session key and decrypts it. At this point the user and message authentication mechanism is executed. From the Figure 8, we can see that a session key(Ks) is generated by the client every time, because the client always requests a data sending or receiving. To generate a session key, the base key(Kb) and message key are needed as described in section 3.

After the session key exchange is finished, the encrypted data transmission is enabled. The basic mechanism is similar to session key exchange. But in the encrypted data transmission, a session key is used to encrypt the data. Figure 9 describes the encrypted data transmission from the client to the server.

The Ks means session key and D is a data block. The control info. contains the size of read block, file size, sending status and so on. Notation E means the encryption of data and the notation M means a message digest.

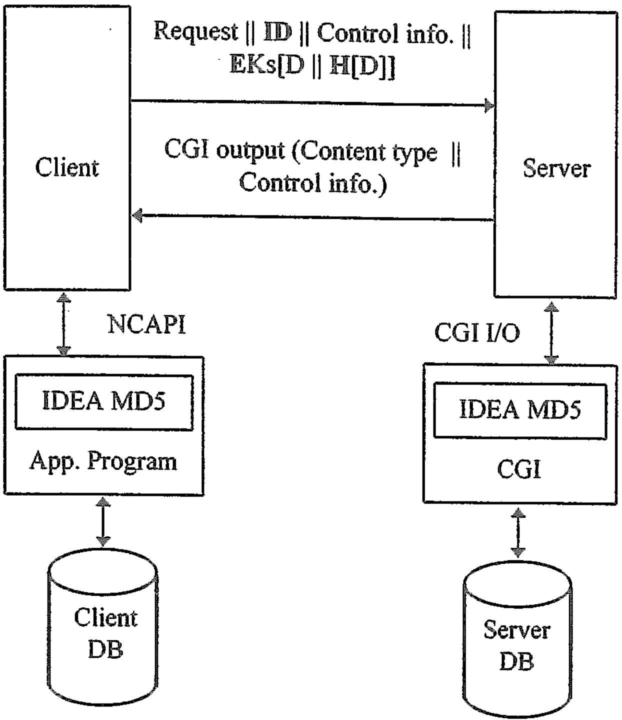


Figure 9. Data sending from client to server

We can see that the case of data sending to the server in the secure HIC system is similar to normal data sending mechanism which is described in section 2. The only difference is the use of encrypted data instead of plain data.

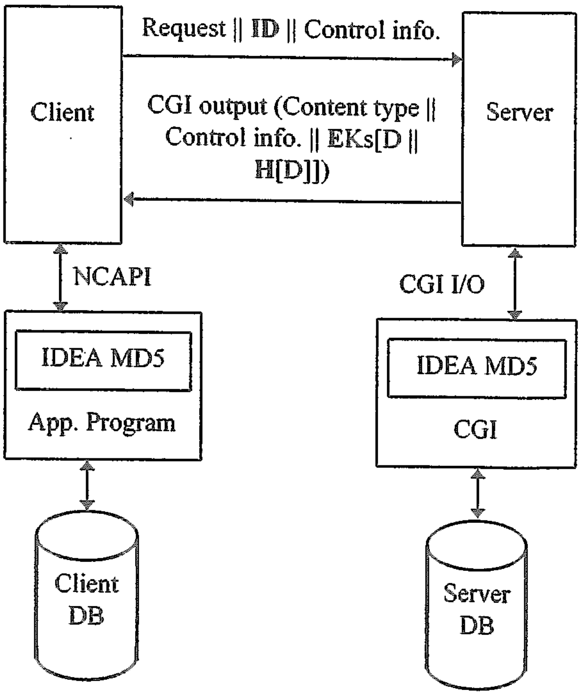

Request || ID || Control info.
CGI output (Content type ||
Control info. || EKs[D ||
H[D]])
Client
Server
NCAPI
CGI I/O
IDEA MD5
App. Program
IDEA MD5
CGI
Client
DB
Server
DB

Figure 10. Data sending from server to client

The mechanism by which the server sends data to the client is also similar to the case of sending data to the server. Therefore, its detail description is omitted. Figure 10 describes this situation.

Besides the encrypted data transmission and the key managers, we implemented a user interface program. For example, the GUI program is required to perform the following functionality. The secure HIC system manager must make a floppy disk that contains the encrypted base key with the client key to assign a base key to client.

5. Conclusion

The HIC system provides information sharing mechanism within its user group. The file uploading and downloading are typical facilities in HIC system. These interactions between client and server are based on WWW.

But WWW has some security problems, simple data transmission mechanism can not be used. The plain text format data can be easily acquired by unauthorized user. To prevent such unauthorized access some cryptographic methods must be applied. Typical cryptographic methods are encryption and authentication mechanisms. To add such additional mechanisms to data transmission on WWW, the client's web browser needs external helper application program and the server's HTTPD needs a CGI program. These helper application and CGI program contain encryption and authentication modules. In our secure HIC system, we adopt IDEA and MD5 algorithms for encryption and authentication. The IDEA and MD5 modules can be used to support general security requirements on WWW. But the encryption mechanism using IDEA module is based on conventional cryptography, client and server must share the same key. In this paper we define 5 HIC keys. These keys are used in key sharing and management mechanisms. The base key is shared between client and server and all data transmissions are started from the manipulation of this base key. For example if a client wants to send a file to server, a session key which is used to encrypt client's data is encrypted using base key and sent to server. Because server knows a client's base key he decrypts the session key and uses this session key to decrypt client's data. The session key is designed to preserve the base key in a secure state and to decrease the chances of exposure of the base key. For the secrecy of base key, we design base key confirming and changing modules as described in section 3. Besides such base key managers we provide various key managers to manage other keys which are used in secure HIC system. Through this security mechanism, the WWW's authorized users can transfer their secret documents with confidence.

Because the secure HIC system uses not-modified HTTP, it can be easily applied to other WWW security applications. The only thing that user must do is to copy external application program to his system. And this application is effective only in HIC system no other overhead or change are made in other WWW communications. This factor can be advantageous.

And to facilitate users, we provides all these programs in a GUI based environment. The first version of HIC system is based on conventional cryptography mechanism, but in later versions we will implement more general mechanisms using public key system[4] or PGP(Pretty Good Privacy)[7] mechanism.

References

1. Richard Stevens, "TCP/IP Illustrated, Volume 1", Addition-Wesley.
2. Allan Schifman and Eric Rescolrla, "Secure HTTP Description", http://www.eit.com/creations/s-http.
3. Kipp E.B. Hickman, "The SSL protocol", http://www.netscape.com/newsref/std/SSL.html.
4. William Stallings, "Network and Internetwork Security", Prentice Hall International Edition.
5. "World Wide Web Journal", O'Reilly & Associates, Inc.
6. "crypt : UNIX system call", Sun OS manual
7. Phil Zimmerman, "PGP:Pretty Good privacy", O'Reilly & Associates, Inc.