Data Loss Prevention for Confidential Web Contents and Security Evaluation with BAN Logic

Yasuhiro Kirihata, Yoshiki Sameshima, Takashi Onoyama, and Norihisa Komoda

***Abstract*—**Since the enforcement of the Private Information Protection Law of Japan, protection of confidential information is one of the significant issues in enterprises and organizations. However, many incidents of confidential information leakage occur and this becomes a serious issue in the industrial society. There is no effective countermeasure to prevent it so far. In this paper, we propose a web content protection system to realize the protection of confidential web contents. The system provides special viewer application to view the encrypted content data and realize the prohibition of copying and taking snapshots for the displayed confidential data. Adopting the dynamical encryption methodology by the intermediate encryption proxy, it is possible to protect the web contents generated dynamically by web applications. Applying our approach to the conventional web system, system administrators can manage the distribution of the confidential information and prevent them from being leaked out from the office. We describe the system architecture and implementation details. We also evaluate the security of the system implementation and the internal authentication protocol with BAN logic.

***Keywords*—**BAN logic, Content access control, Data loss prevention, Protocol verification, Web security.

I. Introduction

CORPORATE governance and compliance are essential keywords for business continuity in the contemporary enterprises and organizations. One of the important points to realize them is the protection and management of confidential information such as customer information, sales data, accounting information, and designs of new products. Especially in Japan, this has been put a great emphasis on since the enforcement of the Private Information Protection Law in 2005. However, many incidents of confidential information leakage occur and it becomes a serious problem in the society.

One of the important protection targets is the web system which has a web application to store and manage the confidential data as simple web pages or data stored in databases. In order to protect the web system from the outside attacker, a lot of techniques and products have already been developed. For example, setting up a firewall between Intranet and Internet, the network administrator restricts and audits the network traffic and keeps attackers out of the Intranet. To protect from the intrusion attack against the web server, the network administrator can detect the attack with the Intrusion Detection System (IDS) [1][2][12] or construct the web site on the Secure OS such as Security Enhanced Linux [3] and OpenSolaris with Solaris Trusted Extensions [4]. In addition, innovative studies are performed to protect the system from not only these direct attacks, but also indirect attacks, for example Denial of Service Attacks (DoS) and Distributed Denial of Service Attacks (DDoS) [5].

However it becomes a serious problem that malicious people disclose confidential information in companies and organizations. Most of these incidents are caused by the internal staff of organizations that can access the confidential information regularly. These crimes are increasing gradually as the propagation of the Internet and it is difficult to prevent them by the conventional system. Enforcing appropriate security policies for system operations to users is only one of the efficient ways to prevent these crimes so far, but it is difficult to enforce security policies completely.

In this paper, we propose the Web Content Protection (WCP) system in which user can only read but cannot leak out confidential information stored on the web server. The WCP system comprises four major components; Viewer, Encryption Proxy, Authentication Server, and Access Control Directory. Viewer is a special application to view the confidential web pages. While Viewer is displaying secret pages, any user cannot copy the displayed data and take screen snapshots. It is also impossible to print out or save the data in the local disks. Therefore, any user cannot leak the secret data distributed to the clients outside the office. Encryption Proxy, which is a proxy server set up between client and web server, encrypts transferred data of the confidential information on demand. Adopting this encryption method, the WCP system supports the confidential information generated dynamically by web applications. Furthermore, the administrator can dynamically change the access rights to the distributed web pages in the clients by modifying the security configuration in the Access Control Directory.

The rest of this paper is structured as follows: In section 2, we

Manuscript received May 30, 2011.

Y. Kirihata is with Hitachi Solutions Ltd., Tokyo, 140-0002 Japan (e-mail: yasuhiro.kirihata.yk@hitachi-solutions.com).

Y. Sameshima is with Hitachi Solutions Ltd., Tokyo, 140-0002 Japan (e-mail: yoshiki.sameshima.vf@hitachi-solutions.com).

T. Onoyama is with Hitachi Solutions Ltd., Tokyo, 140-0002 Japan (e-mail: takashi.onoyama.js@hitachi-solutions.com).

N. Komoda is with Osaka University, Osaka, 565-0871 Japan (e-mail: komoda@ist.osaka-u.ac.jp).

discuss security issues on conventional web systems that deal with confidential information. In section 3, we describe the system requirements and give the overall design of the WCP system. Section 4 details the implementation of the system. In section 5, we evaluate the security of the WCP system from perspectives of the protocol design and implementation. In section 6, we describe related works, and section 7 concludes.

II. Security Issues on the Conventional System

Before discussing the WCP system, we consider the security issues of the conventional web system that stores confidential information. There are several threats to overcome for the protection of the confidential information on the web server. The possible attacks to the conventional web system on the LAN are spoofing, eavesdropping, sneaking, intrusion, screen capturing and cracking implementation flaws of web applications. Fig. 1 shows threats of the conventional web systems.

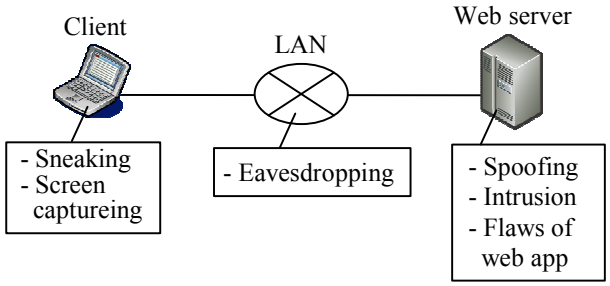

Fig. 1 Threats on the conventional web systems

(1) Spoofing

Attackers might spoof the web server as authorized user to steal the secret pages. To spoof the web server, attackers usually crack password. For instance, they sniff the network (online password cracking) or analyze a local password file (offline password cracking) and steal the IDs and passwords of the authorized users to access the web server. Generally, one-time password or robust password management are the efficient defensive measures against these attacks.

(2) Eavesdropping

Sniffers are the tools with which network administrators observe the network communication for troubleshooting. There are several implementations widely available through the Internet such as TCPDump, EthDump, and Packetman. However, using sniffers, attackers can intercept the transferred data on the LAN. They can analyze the transferred data and steal the confidential information or ID and password of the authorized user to spoof the web server.

(3) Sneaking

Usually, authorized users can access secret pages on the web server and save them in the client machine as necessary. If they bring out the saved secret pages, the confidential information is leaked out. The sneaking is one of the major reasons for the incidents of information leakage. From the viewpoint of the information leakage, it is essential to defend the web system from sneaking confidential data.

(4) Screen capturing

Capture tool is a tool to copy whole or a part of screen data as an image data. It is available on the Internet as a freeware or shareware. After downloading secret pages and viewing the secret information with browser application, the authorized users might take the capture image of the screen with capture tools and sneaking it as an image data.

(5) Intrusion

The goal of most system attacks is intrusion, that is, attackers get the 'root' privilege of the target machine so that they could operate the victim machine freely. To succeed in the intrusion, attackers might carry out buffer over flow attacks or crack the 'root' user's password. If the web server was invaded, confidential information on the web page might be stolen easily.

(6) Attack flaws of web application

If implementation of the web application has security flaws, attackers can make use of them to obtain the secret data illegally. The major security risks on the vulnerable web applications related to the confidential information leakage are as follows [13][14][15][16][17][18].

(a) Cross Site Scripting (XSS)

If there are security flaws for XSS on the web site, attackers can steal the cookie data of the victim users. Firstly, the attacker sets up a web site which has malicious scripts. If victim users access the vulnerable web site after accessing the malicious web site, the malicious embedded code works to send the cookie data to the attacker illegally. If the cookie with the session information is stolen, attacker could achieve the session hijack to access the confidential web site.

(b) Cross Site Request Forgeries (CSRF)

If there are security flaws for CSRF, attackers can set a malicious code into the vulnerable web site for access users to do some unaware operation to the other web site such as publishing confidential information the victim user can access. This is also an attack to reveal the confidential data on the web site.

(c) Injection Flaws

If there are flaws for injection attacks such as SQL injection and JavaScript injection, attackers can possible to get the confidential data in the backend database illegally. There are following attacking methods for injection flaws on the web application.

- SQL Injection
- XML Injection
- JavaScript Injection
- OS command Injection

(d) Directory Traversal

If the security configuration of ACLs for directories on the web server has mistakes, attackers can execute the directory traversal attack to access the confidential pages in the non-shared directories through the other shared directories. Actually attacker uses combination of path "../" to realize the unauthorized access on the web server. One of the protection

methods are checking the input data format and exception handling for "../".

As for the eavesdropping, spoofing and intrusion, there are effective countermeasures to defend the system. For example, installing the IDS into the web server, we can detect the intrusion attacks. To construct the web site on the Secure OS, remote user cannot get the 'root' privilege and it becomes difficult for remote attacker to intrude the web server. Against eavesdropping, encrypting transferred data is an effective method. It is possible to avoid spoofing attacks by applying one time password system or changing password periodically. In terms of the security flaws of the implementation of web applications, there are countermeasures such as sanitizing, HTTP request filtering, input URI check, and cookie encryption to defend the internal web contents.

However, it is difficult to avoid sneaking and screen capturing the secret information by authorized users. In the conventional web system, authorized user can copy or print out the displayed confidential information by using ordinary functions of web browser. The goal of the WCP system is to defend the web system from sneaking and screen capturing.

III. System Architecture

In this section, we propose the WCP system to solve the issues described in the previous section. At first, we discuss the requirements of the WCP system in consideration of the vulnerability of the conventional web system as discussed above. After that, we propose the system architecture to achieve those requirements. Finally, we explain the dynamics of the data flow in detail.

A. System Requirements

Before illustrating the architecture of the WCP system, we describe the requirements of the system.

(1) Compensate for vulnerabilities of the conventional web system

As we discussed in the previous section, we need to defend the web system from several vulnerabilities. To prohibit the information leakage, the WCP system has to cover them completely.

(2) Manage the access control for the distributed confidential information

After distributing the secret data, the system administrator might want to manage the access control for the distributed secret data in accordance with the change of access rights for some users. For instance, if a user who had the access right to a secret page and saved the secret page data to the local client machine, the administrator of the WCP system needs to change the access rights of the saved secret data when the user moves to other department or its role is changed.

(3) Support to secret pages generated dynamically

In many companies, the confidential information, such as customer information and personnel information, is stored in databases. When user queries the database system to search the confidential information, web application such as CGI and Java Servlet dynamically generates the result page. Accordingly, the WCP system needs to support the secret pages generated dynamically.

B. Architecture of the WCP System

Under the above requirements, we propose the WCP system. Fig. 2 illustrates the system architecture of the WCP system. The system comprises web server, Encryption Proxy, Access Control Directory, Authentication Server and client. Viewer is installed in the client, which is an application to display secret web pages in this system. In addition, we assume that Viewer is a tamper-resistant application [19][20][21]. Thereby, it is difficult to analyze and modify the binary code of Viewer. Web server, Encryption Proxy, Access Control Directory and Authentication Server are separated physically so that anyone cannot access them directly and they identify mutually. Furthermore, we assume that the communication in the separated network segment is secure and no one can access each component without being authorized. The web application on the web server deals with confidential information in a database or replies secret pages to clients. Confidential data is stored in the web server with plain text. Encryption Proxy is placed between client and web server to prevent the client from accessing to the web server directly via the network. We explain each component in detail.

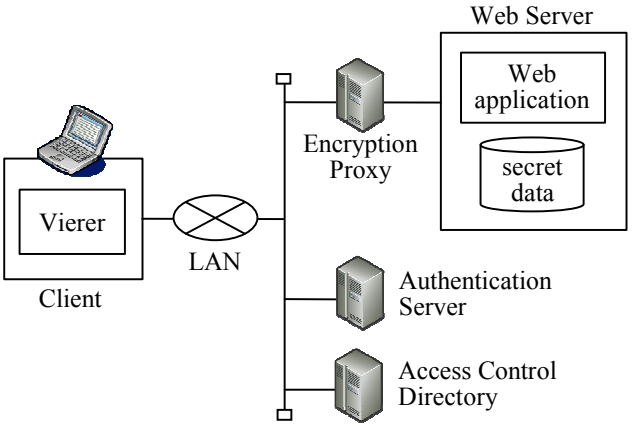

Fig. 2 system architecture of the WCP system

(1) Viewer (Vw)

Viewer downloads an encrypted secret page and negotiates Authentication Server to get a content key, which is a secret key for encryption. When Viewer negotiates Authentication Server, it requires user to enter ID and password. After the negotiation, Viewer decrypts the downloaded secret page with the content key and displays it. While Viewer running, the user can neither take screen capture nor copy the displayed confidential information. Viewer provides saving and printing functions for the encrypted secret pages according to ACLs of the contents.

(2) Encryption Proxy (EP)

This component analyzes the HTTP request sent from the client and queries Access Control Directory whether it is for a

secret page or not. If the HTTP request is for the secret page, it downloads the secret page from the web server and encrypts it with the corresponding content key and sends it to the client.

(3) Access Control Directory (ACD)

This is a database in which system administrator registers the following tables; user table and secret data table. The user table records user account and password. The secret data table records data descriptor, secret page URL, content keys to encipher and decipher the secret pages, and access control list. The user table is used by authentication of access users. To check the access permission to each secret page, the system uses the secret data table.

(4) Authentication Server (AS)

When the user opens a secret page with Viewer, it requests the content key for the secret page to Authentication Server. Authentication Server queries access control list and checks permission of the corresponding secret page to Access Control Directory with ID, password and data descriptor, which identifies the secret page. If the authentication process succeeds, it sends back the corresponding content key and ACL to Viewer. This server has roles of user authentication and content key management.

C. Dynamics of the System

We explain the dynamics of the WCP system. On the WCP system, we can divide system dynamics into two phases; download phase and open data phase.

(1)Download phase

Fig. 3 shows the data flow among the components of the WCP system on the download phase. This phase occurs when Viewer sends a request for a secret page stored in the web server to Encryption Proxy.

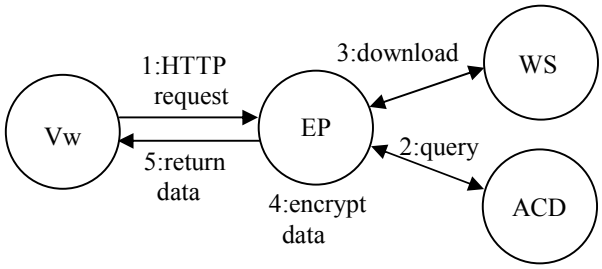

Fig. 3 download phase of the system

In the first step, Viewer sends HTTP request for a secret page to Encryption Proxy. Encryption Proxy analyzes the HTTP request and queries Access Control Directory if the request is for the secret page or not. If the request is for secret page, Encryption Proxy downloads it from the web server and encrypts it with the content key stored in the Access Control Directory. After that, Encryption Proxy adds data descriptor to the encrypted data and sends it to Viewer.

On this phase, user cannot see the secret web pages because the secret web pages are only encrypted and stored in the local client. Hence, Viewer needs to obtain the content key to decrypt and display the downloaded secret page.

(2)Open data phase

This phase is occurred when user opens secret web pages encrypted and stored in the local client. Fig. 4 illustrates the data flow among Viewer, Authentication Server and Access Control Directory on the open data phase.

At first, Viewer sends ID, password, and data descriptor to Authentication Sever. After receiving those data, Authentication Server queries Access Control Directory in terms of ID, password, and data descriptor whether the user may access the secret page or not. If user is allowed to see the data, Authentication Server sends content key to Viewer. Viewer decrypts secret page data with the received key.

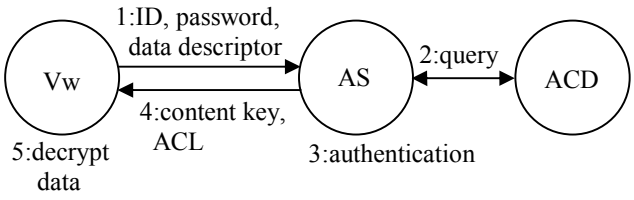

Fig. 4 open phase of the system

The most important process of the system dynamics is the two-way authentication between Viewer and Authentication Server. Actually, the result of user authentication is transferred to Viewer precisely on the basis of the two-way authentication of those two components. Fig. 5 shows the arranged protocol to complete two-way authentication between Viewer and Authentication Server.

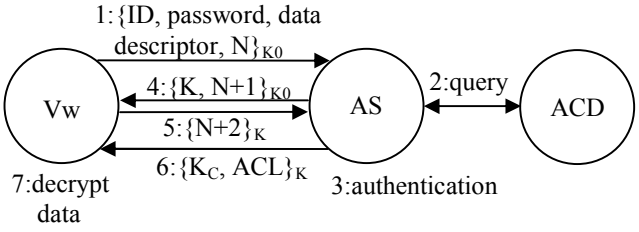

Fig. 5 open data phase of the arranged protocol

K_0 is a shared key between Viewer and Authentication Server initially, and each Viewer has different K_0. K is a session key generated by Authentication Server and K_C is the key with which Viewer decrypts the secret page data. N is a nonce generated by Viewer. At first, Viewer generates a nonce N, encrypts ID, password, data descriptor, and N with K_0, and sends them to Authentication Server as open request of the secret page. Receiving the request, Authentication Sever decrypts requests with K_0 and queries Access Control Directory whether the user can access the secret page or not. If authentication process succeeds, Authentication Server generates a session key K, encrypts (K, N+1) with K_0, and sends it to Viewer. Viewer decrypts received data, encrypts N+2 with

K and sends it to Authentication Server. Authentication Server decrypts the received data and check the data equals to N+2. Finally, Authentication Server encrypts K_C with K and sends it to Viewer.

Applying this protocol, secret page publisher can control the access to the distributed secret web pages. This is because users have to be authenticated and given access permission by Authentication Server whenever they open the secret web page with Viewer.

IV. Implementation

The prototype of the WCP system has been constructed; Viewer, Encryption Proxy, and Access Control Directory. We have implemented the prototype of Viewer on the Windows and Encryption Proxy on FreeBSD. In this section, we illustrate the implementation of the WCP system on those platforms.

A. Design Concept

As for the WCP system, we consider the following design concepts of implementation in addition to the requirements of its architecture.

(1)Seamless GUI of Viewer to web browser

When the user accesses to secret pages with web browser, Viewer is automatically started and displays the secret page into the web browser. The user does not aware of the existence of Viewer as if it is the plug-in module of the web browser. The one can see the secret page like ordinary pages. Furthermore, the one can move to another hyper-link page on Viewer as well as web browser.

(2)Distribution and version management of Viewers

Operating the WCP system, the administrator has to distribute Viewers to each client machine and manage their version. Therefore, distribution and version control mechanism are required.

(3)Prohibition of screen capture

While Viewer is running, the user may not take screen capture with the Print Screen key or capture tools. It is relatively easy to prohibit the Print Screen key. However, prohibition of the screen capture is not simple. This is because its difficulty attributes to the difficulty to identify the capture tools or capturing process. To realize this, we implemented prohibition mechanism of the GDI function calls for screen capturing.

(4)Reutilization of existing components

The Internet Explorer consists of several components. WebBrowser control is one of those components which is a HTML parsing and rendering engine in the Internet Explorer. Reusing this component in Viewer, it is easy to maintain the HTML parsing and rendering function of Viewer.

B. Viewer Control

To satisfy the above requirements, we implemented Viewer as ActiveX control. Fig. 6 illustrates the architecture of Viewer control. Viewer control comprises some components; Container Component, Event Handler, HTML Control, Image Control, and Application-Specific Control.

(1)Container Component

This is the component that manipulates each control such as WebBrowser Control, Image Control, and Application-Specific Control. This component gives the page view and prohibits taking hardcopy with Print Screen key and capture tools, and copying data displayed on each control. In addition, this component downloads encrypted secret page data from Encryption Proxy and decrypts it with the content key.

(2)Event Handler

On the Windows platform, user can copy data displayed on an application view by selecting copy menu in the context menu or mouse operation. Furthermore, user can take a screen capture by pushing the Print Screen key. This component is invoked when a mouse drag event or a context menu event occur. In this handler, Viewer cancels those events to prohibit copy of displayed secret information by drag and drop operation or context menu selection.

(3)WebBrowser Control

This component is the main parsing and rendering engine of the Internet Explorer. Container Component uses this component to parse and render the HTML page after decrypting with content key. If a user traces a hyperlink on the secret web page, Container Component catches the mouse click event on the hyperlink and makes the Internet Explorer send HTTP request to Encryption Proxy.

(4)Image Control

This component decodes and renders image data, for example GIF, JPEG, PNG, and BMP. If a secret image data is linked in the HTML page, this control is embedded and renders the image data.

(5)Application-Specific Control

This is the ActiveX control that renders application-specific format data, for instance, Acrobat Control that renders PDF format file. Viewer can support any file format that has the specific rendering control.

Implemented as ActiveX control, the Viewer can be embedded into the web browser. If a component of frame page is a secret page, the Viewer displays only a part of frame page as a secret page on the web browser.

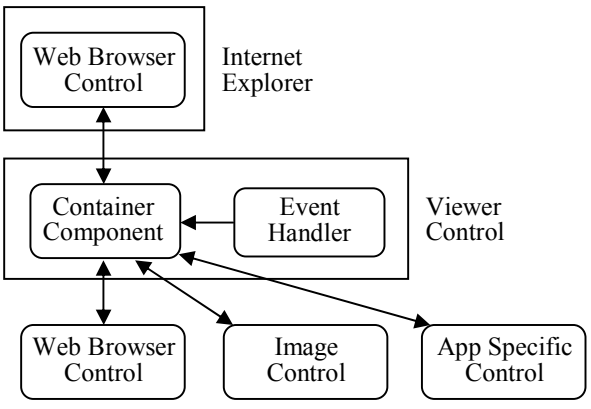

Fig. 6 Viewer control architecture

C. Capture Prohibition

Generally, users can take screen snapshot with the Print Screen key or capture tools. If the malicious user takes screen capture when confidential information is displayed, the

confidential information is copied as image data. It is necessary to prohibit capturing screen image while Viewer is running.

To prohibit screen capture, Viewer implements prohibition of the Print Screen key function and the GDI function calls by which capture tools take hardcopies. The details are as follows.
(1)Prohibition of the Print Screen key

On Windows platform, keyboard event can be monitored by windows message hooking mechanism. Actually, the monitoring and canceling the process of the keyboard message event of the Print Screen key, it is possible to make it invalid.
(2)Methods prohibition by system call hooking

It is possible to intercept and replace Windows NT system services. When a system call is invoked by a user-mode component, a system trap occurs and control is transferred to a software interrupt handler. This handler indexes into a system service table with a system call number to find the address of the NT function that will handle the request. To intercept the system call invocation, you replace entries of this table. As for the GDI function, this table is defined in WIN32K.sys. Functions concerned with screen capturing are GDI functions. Accordingly, when the Viewer is activated, it replaces the system service table in WIN32K to intercept the target GDI functions to invalidate the invocation by capture tools while secret pages are displayed.

D. Encryption Proxy

The prototype of Encryption Proxy authenticates users and checks permission of secret pages. When the Viewer issues a HTTP request to the Encryption Proxy, it analyzes the header field of HTTP request whether it is for a secret page or not. If the request is for secret page, the Encryption Proxy returns a dummy page to active the Viewer Control installed in the client. Web browser receives the dummy page and activates Viewer Control. After that, the Viewer Control is embedded into the dummy page and sends the request of secret page to the Encryption Proxy. Receiving the request from the Viewer Control, the Encryption Proxy queries Access Control Directory in terms of the user authentication and permission to the secret page. The prototype adopts the basic authentication to authenticate the access user. Secret pages' URLs and user accounts are previously registered in the Access Control Directory and the Encryption Proxy queries it with LDAP. After the authentication process, Encryption Proxy downloads the secret page from Web Server, encrypts and sends it to the Viewer Control. Encryption Proxy plays a part of roles of Authentication Server.

Registered URLs are represented by the normal representation and there are two types of matching pattern; upper correspondence and complete correspondence. Upper correspondence matching is the comparison of the URLs initial pattern. For example, if http://foo.com/secure* is enrolled as a secret page, http://foo.com/secure/foo.html becomes a secret page.

This prototype is the modified Squid proxy, which is one of the most popular and stable proxies. We add the authentication function as plug-in of the Squid. On the arrival of the HTTP request issued by web browser, the Squid analyzes the Proxy-Authentication attribute in the HTTP header field to authenticate the access user and returns a dummy page for web browser to activate the Viewer Control. Fig. 7 shows the architecture of the Encryption Proxy.

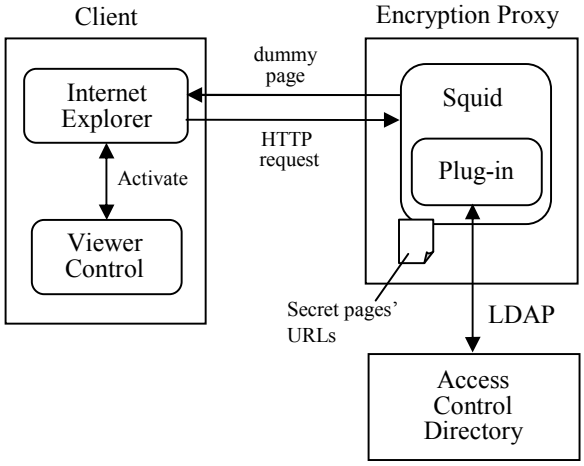

Fig. 7 Architecture of the Encryption Proxy

V. System Evaluation

As we discussed above, we propose the WCP system architecture and describe the implementation method. In this section, we present the outline of BAN logic and discuss the security of the WCP system from the viewpoint of protocol validity and implementation.

A. BAN Logic

Michael Burrows, Martin Abadi and Roger Needham proposed the BAN logic in 1990 [10][11]. This is the semantics in order to analyze an authentication protocol running among multiple entities and verify its validity. The principle of verification is that validity of the authentication protocol is derived from assumptions as for information and beliefs possessed with each principal and some logical postulates. Representative logical postulates of the BAN logic are described as follows:
(1)Message Meaning Rule

This rule is relevant to the interpretation of messages. In other words, it implies derivation of the beliefs with respect to the origin of messages.

Let P and Q be principals and K be a shared key between P and Q. If P believes that K is the shared key between P and Q, and P sees a message $\{X\}_K$, we can conclude that P believes X is sent from Q. In formal terms, it is described as follows:

$$\frac{P \rhd \{X\}_K,\ P \models P \overset{K}{\leftrightarrow} Q}{P \models Q \mid\sim X} \qquad (1)$$

(2)Nonce Verification Rule

Assume that P believes Q once said X and X is fresh, we can conclude P believes that Q believes X. In formal terms, it is described as follows:

$$\frac{P \models \#(X),\ P \models Q \mid\sim X}{P \models Q \models X} \quad (2)$$

This rule implies that a principal can deduce the belief of another principal for a message from its freshness and the fact of its utterance.

(3)Jurisdiction Rule

Let P believe that Q has an authority for X and Q believes X. We can conclude P believes X. In formal terms, it is described as follows:

$$\frac{P \models Q \models X,\ P \models Q \Rightarrow X}{P \models X} \quad (3)$$

In other words, if there is an authority for a message and the authority believes it, the belief is transferred to other principals. This rule is applied in the case when a principal generates a valid key and other principals use it in a system.

Using the postulates, we can prove that principals believe that they can communicate with certain keys.

There are other logical postulates in addition to above rules, but we do not use them in this paper. However, some simple rules such as the ability to deduce $P \models \#(X)$ from $P \models \#(X, Y)$ is used to prove protocol security.

B. Analysis of Protocol

As we described in the previous section, the system dynamics can be divided into two phases; download phase and open data phase. In this section, we discuss the validity of the system dynamics on the viewpoint of the BAN logic.

In the system dynamics of the WCP system, which is the essential part for authentication? Unless an attacker cracked Encryption Proxy or modified the Access Control Directory, downloaded secret pages are automatically encrypted and no one can see the secret information until the finish of the download phase. It is clear that there is no weakness of the protocol in the download phase because of its simplicity. Therefore, user cannot see the secret pages before being authenticated by Authentication Server in the open data phase. To discuss the vulnerability of protocol in the WCP system, we only need to consider the open data phase.

Fig. 8 shows the idealized protocol of the authentication protocol in the open data phase.

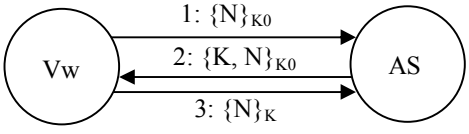

Fig. 8 idealized open data phase in the BAN logic

In this phase, the assumptions are that (1) the Viewer and the Authentication Server share the secret key K_0, (2) the Viewer believes the Authentication Server generates the valid session key K, and (3) the Viewer and the Authentication Sever believe their own nonce and key to be fresh respectively.

$$Vw \models Vw \overset{K_0}{\leftrightarrow} AS \quad (4)$$

$$AS \models Vw \overset{K_0}{\leftrightarrow} AS \quad (5)$$

$$AS \models Vw \overset{K}{\leftrightarrow} AS \quad (6)$$

$$Vw \models AS \Rightarrow Vw \overset{K}{\leftrightarrow} AS \quad (7)$$

$$Vw \models \#(N) \quad (8)$$

$$AS \models \#(Vw \overset{K}{\leftrightarrow} AS) \quad (9)$$

We proceed with the analysis of the idealized protocol. From the first message and Message Meaning Rule, we can deduce $AS \models Vw \mid\sim N$. The second message and Message Meaning Rule give the following formula:

$$Vw \models AS \mid\sim (K, N) \quad (10)$$

The Viewer knows the nonce N is fresh because it generates N recently. In addition, using simple deduction as to the belief operator, we obtain as follows:

$$Vw \models AS \models Vw \overset{K}{\leftrightarrow} AS \quad (11)$$

Using this formula, assumption, and the Jurisdiction Rule, we obtain $Vw \models Vw \overset{K}{\leftrightarrow} AS$. The third message actually means that $\{N, Vw \overset{K}{\leftrightarrow} AS\}_K$. Hence the following formula is derived from this statement, assumption, and Message Meaning Rule:

$$AS \models Vw \mid\sim Vw \overset{K}{\leftrightarrow} AS \quad (12)$$

In the same way, we obtain the following formula from the assumption formula (9) and the Nonce Verification Rule.

$$AS \models Vw \models Vw \overset{K}{\leftrightarrow} AS \quad (13)$$

Above all, we can conclude the following four statements.

$$Vw \models Vw \overset{K}{\leftrightarrow} AS \quad (14)$$

$$AS \models Vw \overset{K}{\leftrightarrow} AS \quad (15)$$

$$Vw \models AS \models Vw \overset{K}{\leftrightarrow} AS \quad (16)$$

$$AS \models Vw \models Vw \overset{K}{\leftrightarrow} AS \quad (17)$$

These four statements are the goal of the verification of this protocol's validity in the BAN logic. Hence, we can conclude that the designed protocol in open data phase provides secure authentication mechanism.

C. Security Evaluation of Implementation

As we discussed in section 2, using existing technologies such as mutual authentication, encryption of the transmitted data, Secure OS, and fixing flaws of the web application implementation, security on the network and web server can be guaranteed. Therefore, we intensively discuss the security of client-side implementation in this section.

(1) Spoof by cracking ActiveX Control

If the attacker created the wrapping ActiveX control over the WebBrowser control and registered it as the WebBrowser control, he could intercept the method and data flow with the wrapper control. Hence, the Viewer Control certificates the WebBrowser control when it creates its instance. There is a general method to check integrity of a program code, that is, the control creator signs the signature to the code itself and Viewer certifies the validity of the code by its signature in order to verify the validity of the ActiveX control. However, it is not practical to sign the digital signature to all controls used by Viewer Control because the existing controls are already installed in the client and it is not practical to sign them when Viewer Control is installed. It is enough to check whether each control is not modified and correct by calculating the hash data of each control before instantiating.

(2) Cancel the prohibition mechanism of GDI function calls

As we mentioned above, the WCP system prohibits the screen capture with capture tools by prohibiting GDI function calls related to screen capturing. However, if the attacker could overwrite the replaced entries of the system service table for hooking of GDI system calls, it is possible to avoid the hooking routine and cancel the prohibition of the GDI function calls for screen capture. One of the countermeasures for this is to check the replaced entries in the system service table periodically to monitor the validity of the hook mechanism for screen capture cancellation.

(3) Analyzing the secret key in Viewer

If attackers get the secret key K_0 embedded in Viewer, they can obtain the session key K and content key K_C by observing the negotiation between Viewer and Authentication Server. Finally they can decrypt any secret pages with K_C. Hence, protection of K_0 is essential in the WCP system. The attackers might modify or analyze the binary code of Viewer to use or obtain K_0 in the Viewer. In addition, they might collect a lot of messages sent to the Authentication Server and attempt to guess K_0. To prevent these attacks, it is necessary that the Viewer is tamper-resistant and encryption with K_0 is strong enough to resist against chosen-plaintext attack, differential attack, and linear attack.

(4) Spoof clients as server-side entities

On the open data phase, Viewer and Authentication Server authenticate mutually before distribution of content key. However, Viewer and Encryption Proxy do not authenticate each other. Accordingly, the cracking proxy or web site might spoof Viewer. This attack does not reveal the confidential information, but it disturbs the service of providing confidential information in the intranet. It is necessary for the Viewer to authenticate the server-side entities such as the Encryption Proxy and web server.

VI. Related Works

Digital Rights Management (DRM) technology [22] allows digital content owners to distribute valuable digital content with its copyright not infringed by malicious people and set the accessible duration and conditions. In this technology framework, player or viewer applications download the decryption key every time from the key management server operated by the content provider when they open the encrypted content files. When the contents are expired, the key management server stops distribution of the corresponding key to clients. Thus, users cannot play or view the contents anymore after the expiration. Content provider can take a flexible access policy for the contents after the distribution. The DRM technology will be applied to the industries of music and e-books. However, it does not support to the web contents dynamically generated by web applications.

IBM research proposes a system for web content protection [8]. It verifies the browser code with the digital signature and prevents users from performing several actions such as printing, saving, and so on. It provides transparent DRM functions to an application. However, it distinguishes the protected contents with the specific protocol such as rmfile (for local content) and rmhttp (for remote content). The Internet Explorer invokes the Trusted Control Handler with these method names. This implies that it is necessary to change the existing web pages for this system. In addition, usage right information is stored in the client and there is no mechanism to change the usage right dynamically. Hence, it is impossible to change the access rights of contents dynamically for roles and users.

VII. Conclusion

The confidential information leakage becomes one of the most serious security problems in enterprises and organizations. In this paper, we proposed the WCP system that realizes the protection of confidential information on the web server. Our approach prohibits copy of the displayed text by the mouse operation and taking hardcopy of the displayed image when the confidential data is on the browser window. In addition, the on-demand encryption on the proxy realizes the protection of the confidential web pages generated dynamically by web applications such as CGI and Java Servlet, and it is the distinctive feature of our solution. According to the security evaluation of internal protocol with BAN logic, we verified that it provides secure authentication mechanism. We also discussed the security in terms of the system implementation and clarified countermeasures against the possible vulnerabilities. The WCP system realizes the secure web distribution framework.

References

[1] A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras, and B. Stiller, "An Overview of IP Flow-based Intrusion Detection," *IEEE Communications Surveys & Tutorials*, Vol.12, No.3, 2010, pp. 1-14.

[2] H. Debar and J. Viinikka, "Intrusion detection: Introduction to intrusion detection and security information management," *Foundations of Security Analysis and Design III*, 2005, pp. 207-236.

[3] Security-Enhanced Linux, http://www.nsa.gov/research/selinux/

[4] Trusted Extensions, http://hub.opensolaris.org/bin/view/Community+Group+security/tx

[5] J. Yuan and K. Mills, "Monitoring the Macroscopic Effect of DDoS Flooding Attacks," *IEEE Transactions on Dependable and Secure Computing*, Vol.2, No.4, 2005, pp. 1-12.

[6] A. Aijaz, S. R. Mohsin, and M. U. Haque, "IP Trace Back Techniques to Ferret out Denial of Service Attack Source," in *Proc. of the 6th WSEAS International Conference on Information Security and Privacy*, 2007, pp. 135-140.

[7] Jeffrey Richter, *Programming Applications for Microsoft Windows Fourth Edition*, Microsoft Press, 1999.

[8] M. Mourad, J.Munson, T. Nadeem, G. Pacifici, M. Pistoia, and A. Youssef, "WebGuard: A System for Web Content Protection," in *Poster Proc. of the 10th International World Wide Web Conference*, 2001.
[9] M. de Vivo, G. O. de Vivo, and G. Isern, "Internet Security Attacks at the Basic Levels," *ACM SIGOPS Operating Systems Review*, Vol.32, 1998, pp. 4-15.
[10] M. Burrows, M. Abadi, and R. Needham, "A Logic of Authentication," *ACM Transaction on Computer Systems*, Vol.8, No.1, 1990, pp. 18-36.
[11] J. Wen, M. Zhang, and X. Li, "The study of the application of BAN logic in formal analysis of authentication protocols," in *Proc. of the 7th International Conference on Electronic Commerce*, Vol.113, 2005, pp. 744-747.
[12] R. A. Wasniowski, "An Agent-Based Knowledge System for Intrusion Detection," *WSEAS Transactions on Business and Economics*, Issue 2, Vol.2, 2005, pp. 70-74.
[13] A. Barth, C. Jackson, and J. C. Mitchell, "Robust Defenses for Cross-Site Request Forgery," in *Proc. of the 15th ACM Conference on Computer and Communications Security*, 2008
[14] G. Wassermann and Z. Su, "Static Detection of Cross-Site Scripting Vulnerabilities," in *Proc. of the 30t International Conference on Software Engineering*, 2008
[15] S. Mohammadi and F. Koohbor, "Protecting Cookies against Cross-site Scripting Attacks Using Cryptography," in *Proc. of 9th WSEAS International Conference on E-Activities, Information Security and Privacy*, 2010, pp. 22-31.
[16] R. Razvan and M. Maria, "The Security of Web 2.0 and Digital Economy," in *Proc. of 11th WSEAS International Conference on Recent Advances in Mathematics and Computers in Business, Economics*, Biology and Chemistry, 2010, pp. 168-170.
[17] A. Yip, X. Wang, N. Zeldovich, and M. F. Kaashoek, "Improving Application Security with Data Flow Assertion," in *Proc. of the ACM SIGOPS 22nd Symposium on Operating Systems Principles*, 2009.
[18] R. Razvan, "Over the SQL Injection hacking method," in *Proc. of 3rd WSEAS International Conference on Communications and Information Technology*, 2009, pp. 116-118.
[19] D. Aucsmith, "Tamper Resistant Software: An Implementation," in *Proc. of 1st International Information Hiding Workshop (Lecture Notes in Computer Science)*, Vol. 1174, 1997, pp.317-333.
[20] S. T. Chow, Y. Gu, H. J. Johnson, and V. A. Zakharov, "An Approach to the Obfuscation of Control-Flow of Sequential Computer Programs," in *Proc. of International Security Conference*, 2001.
[21] J. R. Nickerson, S. T. Chow, and H. J. Johnson, "Tamper Resistant Software: Extending Trust into a Hostile Environment," in *Proc. of Multimedia and Security Workshop at ACM Multimedia*, 2001, pp. 64-67.
[22] Q. Liu, R. Safavi-Naini, and N. P. Sheppard, "Digital Rights Management for Content Distribution," in *Proc. of Australasian Information Security Workshop Conference on ACSW 2003*, Vol. 21, 2003, pp. 49-58.

Yasuhiro Kirihata received the B.S. and M.S. degrees in Mathematics from Kyoto Univ. and Osaka Univ. in 1997 and 1999, respectively. He also received the M.S. degree in Computer Science from Univ. of Illinois at Chicago in 2004. During 1999-now, he has engaged in the research on security and storage applications in R&D dept., Hitachi Solutions, Ltd.

Yoshiki Sameshima received the B.S. degree from Kyoto University in 1984, the MS degree in Mathematics from Osaka University in 1986, and the Ph.D. degree in computer science and technology from Osaka University in 2008. He is working for Hitachi Solutions since 1986, and stayed at Computer Science Department of University College London from 1992 to 1994. Since 1993, he has been working in research of information security. He is a member of IEICE, IPSJ, JSSM, and, the USENIX Association.

Takashi Onoyama received the B.S. and PhD. degrees in Mathematics and Information Science from Osaka University in 1981 and 2007, respectively. In 2003, he became Dept. Manager of R&D in Hitachi Solutions, Ltd. The focus of his research is concentrated within the topics of planning method for logistics system.

Norihisa Komoda received the B.S., M.S. and PhD degrees in Engineering from Osaka Univ. in 1972, 1974 and 1981, respectively. He joined Hitachi Ltd. in 1974. In 1991 he joined the engineering faculty of Osaka Univ. as associate professor. Since 1992, he is professor of Graduate School of Information Science and Technology of Osaka Univ. He is mainly interested in the information system in the area of manufacturing and distribution industry and knowledge based information processing.