웹과 윈도우 환경에서 암호화와 복호화 설계

정태일, 장현희, 박성순
안양대학교 컴퓨터공학과

Encryption and Decryption Design in Web and Window Environment

Taeil Jung, Hyunhee Jang and Sung-Soon Park
Dept. of Computer Engineering, Anyang Univ.

요 약

웹 페이지와 윈도우 어플리케이션 사이의 데이터 전송하는 과정에서 데이터가 다른 사람에 의해 유출 되었을 경우 데이터의 보안을 위해 암호 알고리즘을 사용한다. 자료의 기밀이 유지되어야 할 경우 다른 사람이 알 수 없는 형태로 변형하고, 사용할 때는 원문으로 변형하는 방법이 암호 알고리즘이다. 본 논문에서는 암호 알고리즘과 웹과 윈도우 환경에서 전송하고자 하는 데이터에 대해 암호화 및 복호화하는 방법을 제안한다.

1. 서론

암호화란 누구나 알아 볼 수 있는 평문에 대해 허용된 사람 이외에는 알아볼 수 없도록 암호문으로 바꾸어 주는 것을 말하고, 복호화는 허용된 사람에게 평문을 보여주기 위해 암호문을 역 변환 해주는 것을 말한다.

암호 알고리즘은 특정 내용을 정해진 사람만이 알 수 있도록 하기 위하여 개발 되기 시작했다. 최초 암호 알고리즘은 전쟁 중, 아군에게만 명령을 전달하기 위해 사용 되었다. 문헌에 기초하면 고대 로마제국의 시저는 이미 독자적인 암호 알고리즘을 만들어 사용한 것으로 알려져 있다. 그 후, 암호 알고리즘은 많은 발전을 거듭 하였고, 현재는 DES, RSA와 같이 보다 발전되고 안정된 암호 알고리즘으로 발전되었다.

암호 알고리즘은 암호화 알고리즘과 복호화 알고리즘으로 구성된다. 암호화 알고리즘은 암호화 키를 사용하고, 복호화 알고리즘은 복호화 키를 사용한다. 암호 알고리즘은 일종의 함수로써 키와 데이터를 입력받아서 암호화 및 복호화를 한다.

암호 알고리즘은 크게 비밀키 암호 알고리즘과 공개키 암호 알고리즘으로 구분할 수 있다. 이들 중, 암호화 키와 복호화 키가 같은 것을 비밀키 암호 알고리즘이라 하며, 암호화 키와 복호화 키가 서로 다른 것을 공개키 암호 알고리즘이라 한다[1].

2. 관련 연구

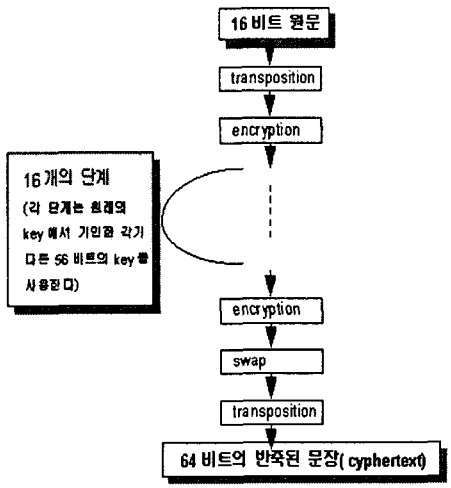
2.1 암호화 방식

암호화 방식중 하나는 대칭형 암호화 방식으로 자료를 암호화하는 키와 암호화된 자료를 복호화하

는 키가 동일한 암호화 방식이다. 이 방식의 장점은 암호화와 복호화가 빠르고, 여러 가지 다양한 암호화 기법이 개발되어 있다는 점이다. 단점은 복수의 사용자가 동일한 자료를 사용할 때 키의 공유문제가 발생한다는 점이다[2].

대표적인 방식 중 하나는 DES(Data Encryption Standard)방식이 있다. 이것은 미국 상무성의 국립표준국(NBS)에서 미국 표준 암호 알고리즘으로 채택한 64 비트 블럭의 입력 및 출력을 가지는 64비트 블럭 암호이다. DES에서 메시지의 입력단위와 비밀키는 모두 64비트로 각 바이트는 1비트의 패리티 비트를 갖는다. 따라서 암호화에 사용되는 실제 비밀키의 길이는 56비트가 되고, 이는 암호화 및 복호화에 사용된다. 그리고, 나머지 8비트는 키 블럭의 parity check 용으로 사용된다[6].

이 과정은 여러 가지 모드에서 실행될 수 있으며, 16번의 연산이 수반된다. 대부분의 연산은 XOR와 비트의 순서바꿈으로 이루어진다. DES에 의한 암호화의 전체적인 구현 절차는 [그림1]과 같다**[2][6].**



[그림 1] DES 전개도(개요)

암호화 방식중 또 다른 하나는 비대칭형 암호화 방식으로 대칭형 암호화 방식이 암호화 정보를 네트워크 상의 상대편에게 보낼 때 키까지 보내야 하지만 그 키를 보호할 길이 없다는 문제 의식에서 개발되기 시작했다.

대표적인 방식중 RSA(Rivest, Shamir, Adleman) 방식이 있다. RSA 암호화 방식은 매우 큰 정수의 소인수 분해가 어렵다는 가정 하에서 설계된 비대칭적 암호화 시스템이다[3].

3. 구성 및 설계

3.1 암호화 문제점과 방안

웹 환경에서 암호화되는 데이터를 윈도우 환경에서 복호화 하기 위해서는 몇 가지 문제점을 해결해야 한다.

첫째, '%$#@'와 같은 특수문자를 포함하는 경우이다. 이러한 특수 문자로 이루어진 암호화된 데이터와 Tab, Shift등에 대한 아스키 코드 값이 전달되어 갈 경우 윈도우 환경에서는 넘어온 문자에 대한 아스키 코드 값을 정확히 알아 낼 수 없다.

예를 들어, 웹 환경(PHP)에서 아스키 코드 값 28~32는 모두 공백을 표시하지만 공백에 대한 아스키 코드는 32하나뿐이다. 28~32의 아스키 코드 중 하나가 윈도우 환경으로 전달됐을 경우에 그 문자에 대한 아스키 코드 값은 32을 가지게 된다. 그렇게 될 경우, 윈도우 환경에서 복호화 할 때 원하지 않는 데이터로 복호화 되게 된다. 이러한 문제점에 대한 대처 방안으로 생각한 것이 28~32가 웹브라우저상에 표시되는 공백 문자의 형태가 아닌 그 문자에 대한 아스키 코드 값으로 암호화 함으로써 암호화된 데이터가 윈도우 환경으로 넘어 제대로 전송 될수 있도록 암호 알고리즘을 제안하였다.

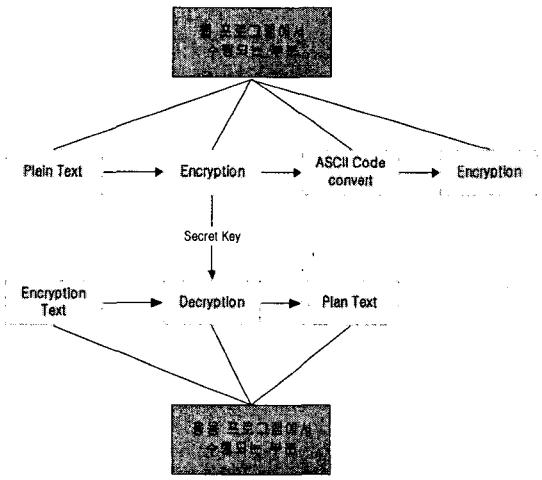
둘째, 암호화된 데이터가 공백 이외의 문자열을 가질 경우 에도 윈도우 환경으로 잘못된 데이터가 전달되는 경우가 생긴다. 이것은 웹 환경에서 윈도우 환경으로 데이타를 보낼 때 PHP형태 "<? echo 문자열; ?>" 형태로 전달되기 때문에 큰 따옴표 같이 PHP 구문으로 전달되는 문자나 개행문자('\n')등의 문자열이 암호화된 데이터로 인식되는 것이 아니라 그 문자에 대한 실행 코드로써 인식된다. 즉, 암호화 될 문자열이 '\n'과 같은 문자열을 포함할 경우 그 문자열은 '\n'을 escape문자로 인식하고, 처리되며 escape

문자에 대한 아스키 코드 값이 윈도우 환경으로 전달 되게 된다. Tab키에 해당되는 '\t'나 큰 따옴표 등도 웹 환경에서 데이터 형태로 전달되지 않고, 그 문자열을 escape문자로 인식하게 되고, 그 escape문자의 아스키코드 값이 윈도우 환경으로 전달되게 된다.

예를 들어, 암호화된 문자열이 "· · \n· · "와 같다면 문자열 내의 "\n"은 92(\)와 110(n)으로 표기되는 것이 아니라 Enter에 해당하는 13(CR)과 11(LF)로 인식 된다.

그러나, 이러한 문제도 암호화된 데이터를 아스키 코드로 전달하는 방식을 사용하여 올바른 암호문이 웹 환경에서 윈도우 환경으로 넘어갈 수 있도록 했다.

웹과 윈도우 환경에서 암호화 및 복호화를 수행할 때 사용되는 키는 아래 [그림 2]와 같이 똑같은 키를 사용해서 암호 알고리즘을 구현했다..



[그림 2] 암호화와 복호화의 모듈

3.2 암호화

암호화 알고리즘은 암호화 할 데이타를 입력받아 rand함수를 이용하여 임의의 키값을 추출한다. 그런 다음에 배열을 이용해서 짝수번 째의 인수 값은 키로써, 그리고 홀수번 째 인수 값은 키와 암호화 할 문자을 XOR한 아스키 코드 값으로 저장한다. 이와 같은 저장 방식으로 배열에 데이터와 키를 XOR해서 저장한다.

웹 환경에서 윈도우 환경으로 데이터가 전달되는 형태는 문자열로 진행 되어야 한다. 그러므로, 암호화되어 저장된 데이터는 배열이 아닌 문자열로 전달해야 한다. [표 1]의 알고리즘에서 보는 것과 같이 윈도우 환경으로 암호화된 문자열이 전달될 때 키 값은 넘어가지 않는다. 그렇기 때문에 전달되는 문자열에 특수문자가 아닌 A~N의 문자중 임의의 문자를 구분자로써 삽입한다. 복호화 할 경우에는 윈도우 환경으로 전달되어 온 암호화된 데이터에서 키를 추출하여 복호화에 사용한다.

```
        :
$str;                              // 암호화할 데이터
$receive_key = RandPass();  // key 생성함수
$Enc = Encrypt($str, $receive_key);
$Enc_re = "";
for ( $i = 0; $i < count($Enc); $i++) {
        $Enc_re .= $Enc[$i].chr(rand(65, 78));
}
return $Enc_re;
        :
```

[표 1] 암호화 알고리즘

3.3 복호화

다운로드나 업로드를 실행하기 위해서는 사용자 정보(아이디, 패스워드, IP)가 필요하다. 그렇기 때문에, 웹 환경에서 윈도우 환경으로 정보를 전달하여 사용해야 한다. 그러나, 그 정보가 암호화 되지 않고 웹 환경에서 윈도우 환경으로 넘어온다면 사용자에 대한 개인 정보가 아무런 여과없이 다른 사람에게 유출될 수 있다.

이러한 문제를 해결하기 위한 방법이 개인 정보를 암호 알고리즘을 사용해서 윈도우 환경으로 전달하는 방법이다. 웹 환경에서 암호화하여 전달된 데이터를 윈도우 환경에서 복호화 함으로써 업로드 및 다운로드를 실행 할 수 있다.

복호화 방법은 아래 [표 2]의 알고리즘과 같이

암호화되어 넘어온 문자열을 가지고 XOR을 이용해서 복호화 한다. 처음에 구분자(A~N)중 하나에 의해 분류된 데이터를 정수 형태로 변환해서 배열에 저장한다. 그 배열에서 짝수번 째 인수에 저장되어 있는 값은 암호화 할 때 생성한 키 값이고 홀수번 째 인수에 있는 값은 그 키와 암호화 될 문자를 XOR한 아스키코드 값이다.

XOR한 문자열은 다시 한번 XOR해주면 원래의 값을 얻을수 있기 때문에, 위의 저장된 배열에서도 짝수번 째 인수에 저장된 값과 홀수번 째 인수에 저장된 값을 XOR해줌으로써 웹 환경에서 XOR을 한 암호화된 데이터를 윈도우 환경에서 우리가 원하는 정보로 바꿀 수 있다.

```
Decrypt(char *txt) {    // txt : 암호화된 문자열
        :
ptr = strtok(txt, "ABCDEFGHIJKLMN");
rec[i] = atoi(ptr);
while(1) {
        ptr = strtok(NULL, "ABCDEFGHIJKLMN");
        if (ptr == NULL) break;
        i++;
        rec[i] = atoi(ptr);
}
for(cnt = 0, j = 0; cnt < i; cnt++) {
        md5 = rec[cnt];
        cnt++;
        tmp[j] = rec[cnt] ^ md5;
        j++;
}
return tmp;
}
```

[표 2] 복호화 알고리즘

4. 구현

본 연구에서 구현한 암호화는 웹 환경에서 PHP를 이용하여 구현하였다. 여기서 사용한 비밀키 암호 알고리즘은 사용자의 정보를 가지고 업로드나 다운로드를 실행 할 때 사용자의 정보를 체크 할 수 있는 알고리즘이다.

암호화된 문자열이 웹에서 윈도우 환경으로 전달되면서 생겨나는 특수 문자 또는 '\t', '\n'과 같은 escape문자에 대한 문제점을 해결하기 위해 문자열을 아스키 코드로 변환해서 전송한다.

복호화는 Window XP 환경에서 Visual C++을 사용하여 구현하였다. 복호화 알고리즘은 웹 환경에서 암호화 된 데이터, 즉, 아스키 코드 값과 구분자를 이용한 암호문에서 구분자에 의해 각각의 암호 문자를 분류해서 정수 형태로 변환한다. 복호화 할 때의 키는 암호문에 들어 있기 때문에 암호문에서 키를 찾아내서 암호문과 XOR를 다시 해줌으로써 업로드나 다운로드 할 때 암호화 되기 전의 사용자 정보(데이터)를 얻어올 수 있는 복호화 방법을 사용하였다.

5. 결론

본 논문에 사용된 암호 알고리즘은 웹 페이지와 윈도우 어플리케이션을 연동해야 하는 경우 아이디, 패스워드, IP 등의 개인 정보가 윈도우 환경으로 전달 되면서 유출 될 수 있는 보안 취약점을 해결하기 위해서 구현한 암호 알고리즘이다. 그러나, 위 알고리즘은 데이터의 길이가 길어 질수록 암호화 되는 문자열도 비례해서 길어지는 문제점이 있다.

향후에는 이러한 데이터의 길이나 아스키 코드에 대한 문제점 없이 암호화 및 복호화가 가능한 암호 알고리즘을 구현할 것이다.

[참고문헌]

[1] "security-Related information" http://www.softforum.co.kr/ , 2003

[2] http://www.jinacampus.com/

[3] Ssangyong Information&Communications Corp. Network Online, *netonline.sicc.co.kr*

[4] www.owl.co.kr저 PHP 웹 솔루션, 2000

[5] 강선명저 Visual C++ 암호화 프로그래밍, 프리렉, 2002.

[6] http://home.ewha.ac.kr/~jhkim/972project/

[7] http://www.phpschool.com/

[8] 박재진, "PHP Programming Bible Vel.4", 2001

[9] 박광덕,"PHP를 이용한 웹서버 프로그래밍", 1999