논문 2006-43TC-2-4

안전한 통신채널 확보를 위한 혼합형 인증알고리즘에 관한 연구

(A Study on Hybrid Authentication Algorithm for Security Channel Retention)

이 선 근*, 김 환 용*

(Lee Seon-Keun and Kim Hwan-Yong)

요 약

대칭형 암호알고리즘은 고속의 데이터 처리 및 구현상의 용이성 등의 장점에도 불구하고 인증기능의 부재로 인하여 네트워크 환경에는 부적합하다. 그러므로 대칭형 암호알고리즘이 인증기능을 수행할 수 있도록 MAC와 MDC를 이용한 혼합형 인증알고리즘을 제안하였다.

제안된 알고리즘은 대칭형 암호알고리즘과 별개로 동작하는 것이 아니라 암호알고리즘 자체에 인증기능을 수행할 수 있도록 함으로서 보안이 확보된 통신채널의 범위를 넓힐 수 있도록 하였다.

Abstract

Symmetric cryptographic algorithm is incongruent in network environment by absence of authentication in spite of advantage of easy etc.. on data processing and implementation of high speed.

Therefore, proposed merging style authentication algorithm that use MAC and MDC so that symmetric cryptographic algorithm can achieve authentication.

Proposed algorithm made security can wide of secure communication channel as to achieve authentication to cryptographic algorithm itself not that act independently of symmetric cryptographic algorithm.

Keywords : Authentication, Security channel, Symmetric cryptographic algorithm, merging style

I. 서 론

NIST는 DES의 64 비트에 대한 불안전한 통신체계를 없애고 보다 안전한 통신채널을 확보하기 위하여 2000년에 AES(Advanced Encryption Standard)로 Rijndael 암호알고리즘을 표준안으로 채택하였다. 그러나 네트워크 환경에서 대칭형 암호알고리즘의 효용성은 매우 미약하다. 그러므로 대칭형 암호알고리즘은 네트워크 환경에서 사용될 경우 인증 채널을 동시에 확보해야 한다는 단점을 가지고 암호화를 수행한다.

대칭형 암호알고리즘을 사용할 때 인증채널의 확보에 사용되는 알고리즘들은 해쉬함수 또는 MAC (Message Authentication Code)등이 있다.

일반적으로 인증과정은 해쉬함수 단독으로 수행하지 않으며 인증알고리즘과 암호알고리즘들이 쌍을 이뤄 동작하게 된다. 이때 일반 해쉬함수를 사용해야 할 경우 암호문과 해쉬함수 모두를 관리해야 하는 단점이 있다.[3]

* 정회원, 원광대학교 전기전자및정보공학부
(Department of Electrical Electronic and Information Engineering, Won-kwang University)
※ 본 연구는 정보통신부에서 지원하는 기초기술연구 지원사업으로 수행.
접수일자 : 2005년5월31일, 수정완료일 : 2006년2월15일

그러므로 본 논문에서는 별도의 관리 없이 인증기능을 수행하며 안전한 채널을 확보할 수 있는 혼합형 인증알고리즘(Hybrid Authentication for Block cryptographic Algorithm : HABA)을 제안하였다.

제안된 HABA는 단독으로 사용되던 MAC, MDC (Manipulation Detection Code)를 해쉬함수와 혼합하여 대칭형 블록암호알고리즘에 적용할 수 있도록 만든 인증알고리즘이다. 그러므로 HABA는 보다 안전한 통신 채널의 확보에 따른 인증 및 암호화 과정을 보장받는 기회를 제공할 것이다.

Ⅱ. 기존 인증알고리즘

대칭형 암호알고리즘과 병행하여 사용되던 기존 인증알고리즘은 해쉬함수와 MAC 또는 MDC이다. 기존 암호알고리즘은 MAC와 MDC를 병행하여 사용하지 않는 것이 일반적이다.

해쉬함수를 사용하여 안전한 보안채널을 확보하기 위해서는 해쉬함수에 대한 안전성을 고려해야 한다. n비트의 해쉬함수 값을 갖는 키가 없는 해쉬함수의 경우, 다음 두 가지 조건을 만족할 때 이상적인 해쉬함수에 대한 안전성을 갖는다.[1][2][4]

i) 각각의 역상과 두 번째 역상을 생성하는데 2^n번의 연산이 필요

ii) 충돌을 생성하는데 $2^{n/2}$번의 연산이 필요

이상과 같은 조건을 만족할 때 t비트 키와 고정된 입력에 대해 임의로 선택된 키가 옳은 MAC 키가 될 확률은 $t>n$일때 2^{-t}이다. 그러므로 n비트 MAC 알고리즘에 대하여 주어진 입력에 대한 MAC 값을 올바르게 추측하거나 주어진 MAC 값에 대한 역상의 안전성을 보장받으며 추측할 확률은 2^{-n}이 된다.

해쉬함수가 n비트의 해쉬 함수값을 출력한다고 할 경우, 계산능력을 2^{96}의 연산이 가능하다고 할 경우 일방향 해쉬함수는 전수 조사 공격에 대항하기 위해서는 n의 값이 96보다 커야하며 충돌저항 해쉬함수는 birthday 공격으로부터 안전하기 위해서는 n의 값이 192보다 커야 한다.[5][6]

메시지 인증코드는 64비트의 키가 사용된다고 가정할 경우 n의 값은 64보다 커야 한다.

(n,k,m) 블록암호는 k비트의 키를 이용하여 n비트 크기의 평문을 m비트의 암호문으로 1:1 변환시키는 역변환 가능한 함수를 의미한다. 이때 E를 블록암호함수라 하면 $E_k(n)$라고 표현할 수 있다. 1:1 변환이므로 이와 같은 경우는 $n=m$이다.

블록암호에 적용하기 위한 해쉬함수는 암복호 길이가 평문 n비트, 암호문 n비트와 같이 1:1 변환일 경우의 단일 길이의 해쉬값을 생성하는 것과 평문 n비트에 대한 암호문 $2n$비트를 생성하는 것과 같이 이중길이의 해쉬값을 생성하는 해쉬함수로 나눌 수 있다.

h가 압축함수 f를 갖는 블록암호로부터 생성된 해쉬함수일 경우, h의 해쉬 비율(Hr : Hashing rate)은 $1/r$로 정의되며 표 1과 같은 Hr을 갖는다. 이때 압축함수 f는 각각의 연속적인 n비트 블록 메시지를 r번의 암호화 과정이 필요하게 된다.[7][8][9]

전용 해쉬함수란 블록암호 또는 모듈러 연산과 같은 기존의 암호시스템 성분들을 다시 이용하지 않고 해쉬만을 목적으로 최적화된 수행을 하도록 디자인된 해쉬함수이다.

이러한 전용 해쉬함수로서는 MD4(Message Digest 4)가 32비트 CPU의 S/W 구현에 적합하도록 설계된 전용 해쉬함수로서 많은 사용을 하였으나 안전성의 문제로 인하여 지금은 MD5를 기준으로 전용 해쉬함수가 사용되고 있다. MD4에 기반을 둔 SHA-1은 미국 NIST에서 제안한 전용 해쉬함수이며 MD4와 비교하여 160비트를 사용하며 4 라운드를 사용한다는 것이 다른 점이다.[11]

데이터 무결성은 인증된 기초자료들을 이용하여 생성, 전송, 저장되는 동안에 인증되지 않은 방법으로 변형되지 않았다는 것을 보장하는 것을 의미한다. 그러므로 데이터 무결성의 판별기준은 제 3자가 임의의 비트를 임의의 길이만큼 추가하거나 삭제 또는 별개의 항목을 추가했는지를 구별하는 것이다.[10]

표 1. n비트를 갖는 블록암호에서의 Hr
Table 1. Hr with n bits in block cryptographic.

해쉬함수	(n,k,m)	해쉬비율(Hr)	해쉬출력값
Matyas-	(n,k,n)	1	$H_i=E_{g(H_{i-1})}(x_i)\oplus x_i$
Davies-Meyer	(n,k,n)	k/n	$H_i=E_{x_i}(H_{i-1})\oplus H_{i-1}$
Miyaguchi-Preneel	(n,k,n)	1	$H_i=E_{g(H_{i-1})}(x_i)\oplus x_i\oplus H_{i-1}$
MDC-2(DES)	(n,k,m)	1/2	$H_i=2E_{g(H_{i-1})}(x_i)\oplus x_i$
MDC-4(DES)	(n,k,m)	1/4	$H_i=4E_{g(H_{i-1})}(x_i)\oplus x_i$

데이터 무결성과 유사한 데이터 고유 인증은 과거에 생성되어 기입된 근거로 데이터 인증에 확신을 주는 부분을 나타낸다. 메시지 인증은 데이터 인증과 비슷하게 사용되는 단어이며 메시지의 근거에 대한 데이터 인증을 포함한다. 데이터 인증은 일반적으로 메시지 인증 코드, 전자서명, 암호문에 추가된 비밀 인증자의 정보를 포함한다.

처리인증은 메시지 인증에 부가적으로 데이터의 유일성과 생성된 시간보장을 포함한 것이다. 유일성과 시간보장 방법으로는 메시지 인증에 시간 변수를 부가적으로 포함시키는 것이다. 암호화과정만을 이용해서는 데이터 무결성을 보장할 수 없다. 만약 임의의 메시지가 A의 키에 의해 복호화 되었을때 복호화된 메시지가 의미있는 것이라면 복호화된 메시지는 A로부터 생성되었다는 가정하에 발생할 수 있는 보편적 오해는 암호화가 데이터 고유 인증과 데이터 무결성을 제공한다는 것이다. 직관적으로 한 공격자가 메시지들을 조작하기 위해서는 비밀키를 알아야만 한다. 그러나 경우에 따라 공격자가 평문 메시지를 선택할 수 있고 어떤 경우에는 평문을 선택할 수 없음에도 불구하고 평문을 효과적으로 조작할 수 있다.[12]

III. 제안된 HABA 암호알고리즘

일반적으로 대칭형 암호알고리즘은 인증기능이 없다. 그러므로 unsecurity channel에서의 대칭형 암호알고리즘의 사용은 네트워크 환경에서 매우 한정되어 사용될 수 밖에 없다. 비대칭형 암호알고리즘은 네트워크 환경에서 키 관리 및 분배, 인증기능이 강화되었다는 점에서 많이 사용된다. 그러나 처리시간이 너무 길고 구현상의 어려움으로 인한 제약으로 현실적으로는 많이 사용되고 있지 않다는 단점이 있다. 그러므로 HABA 암호알고리즘은 대칭형 기반 암호알고리즘이지만 인증기능을 포함함으로서 네트워크 환경에 적합하도록 하였다.

기존 대칭형 암호알고리즘은 인증기능이 없기 때문에 인증기능이 필요한 네트워크 환경에서는 해쉬함수 또는 MAC와 더불어 사용된다. 그러나 일반 해쉬함수를 사용해야 할 경우 암호문과 해쉬함수 모두를 관리해야하는 단점이 있다.

HABA 암호알고리즘은 해쉬함수와 MAC, MDC를 혼합하여 사용함으로서 기존 인증기능을 강화하였다.

대칭형 암호알고리즘을 사용하면서 인증기능과 무결성 기능을 강화시키기 위하여 해쉬함수의 예측 가능한

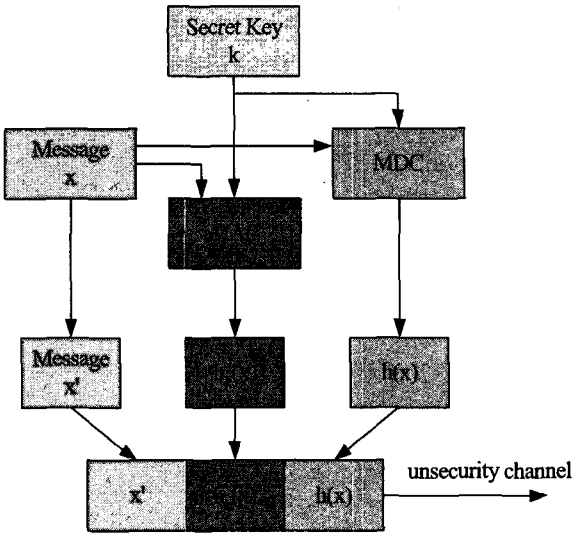

그림 1. MAC/MDC 혼합형 인증 구조
Fig. 1. MAC/MDC hybrid authentication structure.

관계 또는 입출력 상관관계를 나타내지 않도록 하였고 blocked hand-shake 방법을 적용한 MAC와 MDC를 동시에 사용함으로 N:1, 1:N 그리고 N:N 네트워크 환경에 적합하도록 하는 HABA 암호알고리즘을 개발하였다.

그림 1과 같이 HABA 암호알고리즘의 인증기능은 비밀키와 평문 또는 암호문을 이용하여 인증정보를 생산한다.

메시지 x'는 x가 암호문인 경우 평문이고 평문인 경우 암호문을 의미한다. HABA는 무결성을 제공하며 키 관리를 수행해야 하는 MAC 부분과 해쉬함수를 생성하여 해쉬값을 가지게 되는 MDC를 함께 공유하는 구조로 되어 있다.

MDC 계산을 수행하는 방법은 다음과 같다.

입력은 $t \geq 2$, 비트 길이 $l = 64t$인 메시지 x이며 출력은 x에 대한 128 비트의 해쉬값이다.

i) x를 64 비트의 배수를 기준으로

$x = x_1, x_2, x_3, \cdots, x_t$로 정리한다.

ii) 64 비트의 초기값 IV, $\widetilde{IV}$를 다음과 같이 선택한다.

$$IV = 0\text{X}5252525252525252$$

$$\widetilde{IV} = 0\ \text{X}2525252525252525$$

iii) C_i^L과 C_i^R을 각각 C_i의 왼쪽과 오른쪽 32 비트 값이라고 하면 그림 2와 같은 구조에서 식 (1)과 같은 계산을 수행하여 $h(x) = H_t \| \widetilde{H_t}$를 출력하게 된다.

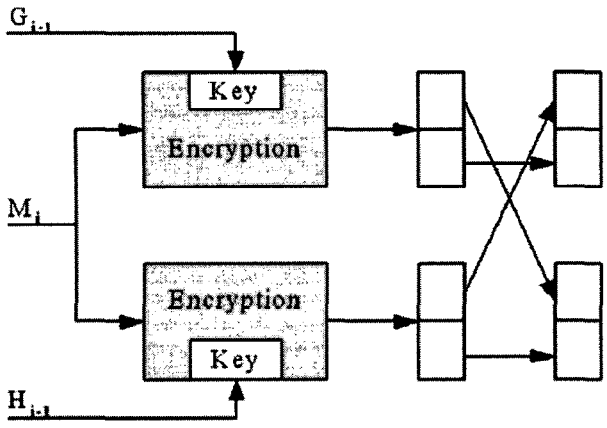

그림 2. MDC 구조
Fig. 2. MDC architecture.

$$H_0 = IV; \qquad \widetilde{H_0} = \widetilde{IV}; \tag{1}$$
$$k_i = g(H_{i-1}); \qquad \tilde{k}_i = \tilde{g}(\widetilde{H_{i-1}});$$
$$C_i = E_k(x_i) \oplus x_i\ ; \qquad \widetilde{C_i} = E_{\tilde{k}}(x_i) \oplus x_i\ ;$$
$$H_i = C_i^L \parallel \widetilde{C_i^R} \qquad \widetilde{H_i} = \widetilde{C_i^L} \parallel \widetilde{C_i^R}$$

MAC 계산은 다음과 같다.

입력값은 비트 길이 $32j$인 데이터 x이며 여기에서 $1 \leq j \leq 10^6$이다. 그리고 MAC 64 비트에 대한 비밀키 $Z = Z[1], Z[2], \cdots Z[8]$이다. 이때 출력값은 x에 대한 32 비트 MAC가 된다.

i) 메시지와 무관한 키 확장한다. 즉 키 Z를 6개의 32 비트로 구성된 X, Y, V, W, S, T 로 확장한다. 이때 X, Y는 초기값이며 V, W는 주 순환값이며 S, T는 메시지에 첨가되는 padding 값이다.

a) Z의 바이트 0x00나 0xff를 다음에 의해 임의의 바이트로 전환한다.

$P \leftarrow 0;$

for $i = 1$ *to* 8

$\{P \leftarrow 2P;$

if$(Z[i] = 0x00$ or $0xff)$

$\{P \leftarrow P+1;\ Z[i] \leftarrow Z[i]\ OR\ P;\}$

$\}$

b) J와 K를 각각 Z의 MSB, LSB 4 바이트라고 할 때 다음을 계산한다.

$X \leftarrow J^4(\mathrm{mod}2^{32}-1) \oplus J^4(\mathrm{mod}2^{32}-2)$

$Y \leftarrow [K^5(\mathrm{mod}2^{32}-1) \oplus K^5(\mathrm{mod}2^{32}-2)](1+P)^2(\mathrm{mod}2^{32}-2)$

$V \leftarrow J^6(\mathrm{mod}2^{32}-1) \oplus J^6(\mathrm{mod}2^{32}-2)$

$W \leftarrow K^7(\mathrm{mod}2^{32}-1) \oplus K^7(\mathrm{mod}2^{32}-2)$

$S \leftarrow J^8(\mathrm{mod}2^{32}-1) \oplus J^8(\mathrm{mod}2^{32}-2)$

$T \leftarrow K^9(\mathrm{mod}2^{32}-1) \oplus K^9(\mathrm{mod}2^{32}-2)$

c) 3개의 결과 쌍 (X, Y), (V, W), (S, T)에서 a)에서와 같이 0x00과 0xff를 제거한다.

d) AND-OR 상수를 정의한다.

$A =$ 0x02040801, $B =$ 0x00804021

$C =$ 0x$bfef7fdf$, $D =$ 0x$7dfefbff$

ii) 초기화와 padding을 수행한다.

a) 초기화 : $v \leftarrow V$, $H_1 \leftarrow X$, $H_2 \leftarrow Y$

b) S, T를 x에 padding한다. 이때 $x_1, \cdots, x_t$를 padding된 32 비트 메시지 블록이라고 표기한다.

iii) 블록을 처리한다.

각 32 비트 메시지 블록 x_i를 다음과 같이 처리한다.

$v \leftarrow (v \hookleftarrow 1)$

$U \leftarrow (v \oplus W)$

$t_1 \leftarrow (H_1 \oplus x_i) \times_1 ((H_2 \oplus x_i) + U)\ OR\ A)\ AND\ C)$

$t_2 \leftarrow (H_2 \oplus x_i) \times_2 ((H_1 \oplus x_i) + U)\ OR\ B)\ AND\ D)$

$H_1 \leftarrow t_1, \quad H_2 \leftarrow t_2$

여기에서 x_i는 $\mathrm{mod}2^{32}-i$에서 곱셈연산을, $\oplus$는 $\mathrm{mod}2^{32}$에서의 덧셈을, $\hookleftarrow$은 왼쪽으로 1 비트 rotation을 의미한다.

iv) 마지막으로 MAC의 결과값은

$$H = H_1 \oplus H_2$$

이다.

이상과 같이 MDC, MAC를 구한 후 두 가지 함수에 대한 값을 산출하여 식 (2)와 같이 하나의 frame에 더한다.

$$C = E_k(x \parallel h(x))\ \ \&\ \ E_k(x \parallel h_k(x)) \tag{2}$$

식 (2)에서 $h(x)$를 기준으로 메시지 x를 암호화하여 구해진 MDC 값과 $h_k(x)$를 이용하여 구한 MAC 값을 합하여 하나의 데이터를 만든 것이 암호문 C값이 된다. 이때 MDC와 MAC에 의하여 생성된 값들은 식 (3)과 같이 동일한 IV와 동일한 메시지 x를 사용하여도

항상 상호 독립성을 유지해야 한다.

$$E_k(x \| h(x)) \neq (E_k(x),\ E_k(h(x))) \quad (3)$$

만약 식 (3)의 조건을 무시하거나 MDC와 MAC가 종속관계를 가지게 되면 padding을 수행하였다고 하여도 데이터 내부의 MDC, MAC 값을 구별할 수 있는 기준이 사라져 복호화를 수행할 수 없게 된다.

식 (3)과 같이 수행할 수 있기 때문에 HABA 암호알고리즘은 기존의 인증 알고리즘과 구별된다.

Ⅳ. 결 론

기존 대칭형 암호알고리즘은 인증기능이 없기 때문에 네트워크 환경에 사용되기 위하여 일반적으로 해쉬함수와 MAC 또는 MDC를 병행하여 사용하게 된다. 이때 전송채널은 암호화 된 데이터와 인증용 데이터들이 독립적으로 채널을 확보하여 전송하게 된다. 이러한 현상은 점유 대역폭을 증가시키며 이로 인하여 전송률이 저하되는 현상이 발생하게 된다.

그러므로 본 연구에서는 하나의 대역폭만을 사용하여 암호화된 데이터에 인증기능을 포함시키기 위하여 MAC와 MDC를 결합한 HABA 암호알고리즘을 제안하였다.

제안된 HABA 암호알고리즘은 기존에서 사용되던 MAC와 MDC를 하나의 인증용으로 사용하기 위하여 결합한 알고리즘이므로 HABA는 보다 안전한 통신채널의 확보에 따른 인증 및 암호화 과정을 보장하는 기회를 제공할 것이다. 그러므로 향후 네트워크 환경에서 비대칭형 암호알고리즘의 대용으로 대칭형 암호알고리즘을 사용할 경우에 HABA 암호알고리즘은 매우 유용한 알고리즘이라고 생각된다.

참 고 문 헌

[1] R. Anderson, "The classification of hash functions", P. G. Farrell, editor, Codes and Cyphers: Cryptography and Coding Ⅳ, Institute of Mathematics & Its Applications, pp. 83-93, 1995.

[2] I. B. Damgard, "Collision free hash functions and public key signature schemes", Advances in Cryptology-EUROCRYPT'87, LNCS, vol. 304, pp. 203-216, 1988.

[3] B. Preneel, "Cryptographic hash functions", European Transactions on Telecommunications, vol. 5, pp. 431-448, 1994.

[4] R. C. Merkle, "One way hash functions and DES", Advances in Cryptology-CRYPTO'89, LNCS, Vol. 435, pp. 428-446, 1990.

[5] H. Dobbertin, "Cryptanalysis of MD-4", D. Gollmann, editor, Fast Software Encryption, Third International Workshop, LNCS, vol. 1039, pp. 71-82, Springer-Verlag, 1996.

[6] M. J. Bach, "The design of the UNIX operating system", ISBN 0-13-201799-7 or ISBN 0-13-201757-1 (international ed.). Prentice-Hall 1986.

[7] J. J. Bae and T. Suda, "Survey of Traffic Control Schemes and Protocols in ATM Networks", IEEE vol. 79, no. 2. February 1991.

[8] Colella, Callon, Gardner &Rekhter, "Guidelines for OSI NSAP Allocation in the Internet", RFC 1629, IETF May 1994.

[9] ATM Forum, "LAN Emulation Over ATM Version 1.0", af-lane-0021.000, January 1995.

[10] ATM Forum, "ATM User Network Interface (UNI) Specification Version 3.1", ISBN 0-13-393828-X, Prentice Hall, Englewood Cliffs, NJ, June 1995.

[11] M. W. Garrett, "A Service Architecture for ATM: From Applications to Scheduling", IEEE Network May/June 1996.

[12] G. C. Sacket and C. Y. Metz, "ATM and Multiprotocol Networking", ISBN 0-07- 057724-2, McGraw-Hill 1997.

저 자 소 개

이 선 근(정회원)
대한전자공학회 논문지, 제42권 SD편 제10호 참조

김 환 용(정회원)
대한전자공학회 논문지, 제42권 SD편 제10호 참조