情報保護學會論文誌
第19卷 第3號, 2009. 6

코사이클 Hadamard 행렬을 이용한 키 동의 알고리즘*

최 창 희,[1†‡] 김 정 수,[2] 이 문 호[1]
[1]전북대학교, [2]한국사이버대학교

Key Agreement Algorithms Based on Co-cyclic Hadamard Matrices*

Chang-hui Choe,[1†‡] Jeong-Su Kim,[2] Moon Ho Lee[1]
[1]Chonbuk National University, [2]Korea Cyber University

요 약

본 논문에서는 코사이클 재킷 행렬 기반 키 동의 알고리즘의 문제점을 제시하고, 이를 해결할 수 있는 방안으로 코사이클 Hadamard 행렬 기반 키 동의 알고리즘을 제안한다. 제안하는 알고리즘은 기존 방식보다 성능 또한 우수하며, 코사이클 Hadamard 행렬의 생성은 매우 간단하다. 또한 키 동의 과정에 소요되는 시간은 키 동의 과정에서 사용하는 행렬의 크기와 키의 길이에 비례하므로 지수 연산이 필요한 다른 알고리즘에 비하여 계산 속도가 빨라 특히 계산 성능이 낮은 단말 사이의 통신에 유용할 것이다.

ABSTRACT

In this paper, we analyze key agreement algorithms based on co-cyclic Jacket matrices, and propose key agreement algorithms based on co-cyclic Hadamard matrices to fix the problem. The performance of our proposal is better than conventional one's and the construction of the matrices is very simple. Also time complexity of our proposal is proportional to the factor that determinees the size of the matrix, and the length of the key. So our proposal is fast and will be useful for the communcations of two or three users, especially for those have low computing power.

Keywords: key agreement, Hadamard matrix, co-cycle

I. 서 론

두 명의 사용자 A와 B가 공통키를 갖고자 할 때, A가 비밀키를 생성하여 이를 B의 공개키로 암호화하여 B에게 보내면 B가 이를 받아 복호함으로써 A와 B가 동일한 키를 공유할 수 있다. 이 경우, 공통키는 한 사용자(여기에서는 A)에 의해서만 생성되며, 계산양이 많은 공개키 암호 방식을 사용하여야 한다.

Diffie-Hellman 알고리즘 이후, 여러 가지 키 동의 알고리즘이 제안되었다. [1,2]에서는 코사이클 재킷 행렬의 한 형태를 이용한 몇 가지 키 동의 알고리즘이 제안되었다. 본 논문에서는 이를 수정하여 코사이클 재킷 행렬에 포함되는 Sylvester Hadamard 행렬을 이용한 키 동의 알고리즘을 제안하고, 이를 통하여 기존 알고리즘의 문제점을 해결한다. 또한 제안하는 Sylvester Hadamard 행렬 기반 알고리즘의 성능이 [1,2]의 재킷 행렬 기반 알고리즘보다 우수함을 확인한다.

접수일(2008년 11월 3일), 수정일(2009년 3월 16일), 게재확정일(2009년 5월 25일)
* 본 논문은 중소기업청이 주관하는 산학 공동기술개발지원 사업의 연구결과임.
† 주저자, nblue95@chonbuk.ac.kr
‡ 교신저자, nblue95@chonbuk.ac.kr

II. 코사이클 Hadamard 행렬

Sylvester Hadamard 행렬은 다음과 같이 정의될 수 있다.

$$[H]_{2^n}=[H]_{2^{n-1}}\otimes[H]_2,\ n\geq 2,$$
$$[H]_2=\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}. \quad (1)$$

여기에서 ⊗는 Kronecker 곱셈 연산이며 $m\times n$ 행렬 A에 대하여 다음과 같이 정의된다.

$$A\otimes B=\begin{bmatrix}a_{11}B & \cdots & a_{1n}B\\ \vdots & \ddots & \vdots\\ a_{m1}B & \cdots & a_{mn}B\end{bmatrix}. \quad (2)$$

또한 식 (1)은 다음과 같이 표현할 수 있다[3].

$$[H]_{(g,h)}=(-1)^{<g,h>},\ g,h\in\{0,1,\cdots,2^n-1\}. \quad (3)$$

여기에서 $<g,h>=g_0h_0+g_1h_1+\cdots+g_{n-1}h_{n-1}$은 $g=(g_{n-1},g_{n-2},\cdots,g_0)$와 $h=(h_{n-1},h_{n-2},\cdots,h_0)$의 2진 내적이며, Sylvester Hadamard 행렬의 원소의 위치를 나타내는 행과 열의 인덱스가 된다. 따라서 이 행렬의 전체를 미리 계산 또는 저장하여 두지 않고 식 (3)을 이용하여 특정 원소를 바로 계산하여 얻을 수 있다. 단, 본 논문에서는 행렬의 행 또는 열 인덱스를 1이 아닌 0부터 시작한다. 즉, Sylvester Hadamard 행렬의 두 번째 행 세 번째 열에 위치한 원소는 $[H]_{(1,2)}=(-1)^{<1,2>}=1$이다. 또한 식 (3)은 밑이 -1이므로 사실상 지수 연산을 할 필요 없이 $(g_0h_0,\cdots,g_{n-1}h_{n-1})$에 1(또는 0)이 몇 개인지만 세면 그 값을 바로 알 수 있다.

위수가 v인 유한군 G와 위수가 w인 유한 가환군 C에 대하여 다음과 같은 코사이클 방정식을 만족하는 사상 $\varphi: G\times G\rightarrow C$를 코사이클이라고 한다[3].

$$\forall g,h,k\in G,\ \varphi(g,\ h)\varphi(g\circ h,\ k)=\varphi(g,\ h\circ k)\varphi(h,\ k),$$
$$\forall g,h\in G,\ \varphi(g,\ 0)=\varphi(0,\ h)=1. \quad (4)$$

이 코사이클은 행과 열의 인덱스를 G의 원소로 나타내어 $(g,\ h)$ 위치의 원소가 $\varphi(g,\ h)$인 $v\times v$ 정방행렬 M_φ로 표현할 수 있다. 이러한 행렬 M_φ를 코사이클 행렬이라고 한다.

만약 $\varphi(g,\ h)=\varphi(h,\ g),\forall g,h\in G$이면, M_φ는 대칭 행렬이며, 이 때 식 (4)로부터 다음 식이 성립한다.

$$\varphi(g,\ h)\varphi(g\circ h,\ k)=\varphi(h,\ k)\varphi(h\circ k,\ g)$$
$$=\varphi(k,\ g)\varphi(k\circ g,\ h). \quad (5)$$

한편 [1,2]의 코사이클 재킷 행렬은 다음과 같이 표현할 수 있다.

$$[J]_{(g,h)}=(-1)^{<g,h>}w^{f(g)f(h)},$$
$$f(x)=\begin{cases}0, & \text{if } x_{n-1}=x_{n-2}\\ 1, & \text{otherwise.}\end{cases} \quad (6)$$

$g\circ h$는 $g\circ h=(g_{n-1}\oplus h_{n-1},g_{n-2}\oplus h_{n-2},\cdots,g_0\oplus h_0)$과 같은 비트 단위 XOR 연산으로 계산되는 2진 표현으로 정의되며, 식 (6)에서 $w=1$로 놓으면 식 (3)과 동일하다. 즉, 식 (3)은 식 (6)의 한 형태라고 볼 수 있다. 식 (6)은 코사이클 행렬이므로[1,2] 식 (3) 또한 코사이클 행렬이다.

III. Hadamard 행렬 기반 키 동의 알고리즘

이 장에서는 앞에서 제시한 Sylvester Hadamard 행렬을 이용한 2, 3자간 키 동의 알고리즘을 제안한다. 각 사용자는 모두 TA(Trusted Authority)와 안전한 통신을 할 수 있으며, 이 TA에 의하여 인증된다. Sylvester Hadamard 행렬의 크기는 $2^n\times 2^n$이고, g, h, k는 N비트의 수(단, $N=mn$)로 각각 $(g_0,g_1,\cdots,g_{m-1})$과 같이 m개의 n비트 수로 나눌 수 있으며 최종적으로 m비트의 세션키를 공유하게 된다.

3.1 2자간 키 동의 알고리즘

두 명의 사용자 A와 B가 공개 채널을 통하여 안전한 통신을 하기 위한 공통의 세션키를 갖고자 한다. 또한 두 사용자 중 어느 쪽도 세션키 생성에 필요한 비밀 정보를 독점하지 않으며, 따라서 어느 쪽도 모든 비밀 정보를 알 수 없도록 하고자 한다.

TA와 A, B는 사전에 n을 공유한다. 즉, 동일한 크기의 행렬을 사용할 것을 약속한다.

① A와 TA, B는 각각 임의의 수 g, h, k를 생성한다.

② A와 B는 TA에게 각각 g, k를 안전하게 전송한다.

③ TA는 A에게 $h\circ k$와 S_A를, B에게 $g\circ h$와 S_B를 각각 안전하게 전송한다. S_A와 S_B는 다음과 같고, 여기에서 $\varphi(a,\ b)$는 공유하는 Sylvester Hadamard 행렬의 a번째 행 b번째 열 위치의 원소로 ± 1이 된다. 그리고 ‖는 단순 연결 연산자이다. 즉 0‖1‖1‖0는 (0,1,1,0)이다.

$$
\begin{aligned}
S_A &= S_{A_0}\|S_{A_1}\|\cdots\|S_{A_{m-1}} \\
&= \varphi(g_0,\ h_0)\|\varphi(g_1,\ h_1)\|\cdots\|\varphi(g_{m-1},\ h_{m-1}), \\
S_B &= S_{B_0}\|S_{B_1}\|\cdots\|S_{B_{m-1}} \\
&= \varphi(h_0,\ k_0)\|\varphi(h_1,\ k_1)\|\cdots\|\varphi(h_{m-1},\ k_{m-1}).
\end{aligned} \tag{7}
$$

④ A와 B는 다음과 같은 P_A와 P_B를 동시에 교환한다.

$$
\begin{aligned}
P_A &= P_{A_0}\|P_{A_1}\|\cdots\|P_{A_{m-1}} \\
&= \varphi(g_0,\ h_0\circ k_0)\|\varphi(g_1,\ h_1\circ k_1)\|\cdots\| \\
&\quad \varphi(g_{m-1},\ h_{m-1}\circ k_{m-1}), \\
P_B &= P_{B_0}\|P_{B_1}\|\cdots\|P_{B_{m-1}} \\
&= \varphi(g_0\circ h_0,\ k_0)\|\varphi(g_1\circ h_1,\ k_1)\|\cdots\| \\
&\quad \varphi(g_{m-1}\circ h_{m-1},\ k_{m-1}).
\end{aligned} \tag{8}
$$

⑤ A와 B는 다음과 같이 각자의 키 K_A와 K_B를 계산한다.

$$
\begin{aligned}
K_A &= K_{A_0}\|K_{A_1}\|\cdots\|K_{A_{m-1}} \\
&= (S_{A_0}\times P_{B_0})\|(S_{A_1}\times P_{B_1})\|\cdots\|(S_{A_{m-1}}\times P_{B_{m-1}}), \\
K_B &= K_{B_0}\|K_{B_1}\|\cdots\|K_{B_{m-1}} \\
&= (S_{B_0}\times P_{A_0})\|(S_{B_1}\times P_{A_1})\|\cdots\|(S_{B_{m-1}}\times P_{A_{m-1}}).
\end{aligned} \tag{9}
$$

⑥ 식 (4)로부터 A와 B가 계산한 키는 다음과 같이 동일함을 알 수 있다.

$$
\begin{aligned}
&\forall t \in \{0,1,\cdots,m-1\}, \\
&K_{A_t} = S_{A_t}\times P_{B_t} \\
&\quad = \varphi(g_t,\ h_t)\varphi(g_t\circ h_t,\ k_t) \\
&\quad = \varphi(g_t,\ h_t\circ k_t)\varphi(h_t,\ k_t) = S_{B_t}\times P_{A_t} = K_{B_t}.
\end{aligned} \tag{10}
$$

⑦ 이제 A와 B는 공통의 세션키 $K=K_A=K_B$를 이용하여 안전한 통신을 할 수 있다.

3.2 3자간 키 동의 알고리즘

앞 절의 상황에 세 번째 사용자 C가 추가된다.

TA와 A, B, C는 사전에 n을 공유한다. 즉, 동일한 크기의 행렬을 사용할 것을 약속한다.

① A, B, C는 임의로 각각 g, h, k를 생성하여, TA에게 안전하게 전송한다.

② TA는 A에게 $h\circ k$와 S_A를, B에게 $k\circ g$와 S_B를, C에게 $g\circ h$와 S_C를 각각 안전하게 전송한다. S_A, S_B, S_C는 다음과 같다.

$$
\begin{aligned}
S_A &= S_{A_0}\|S_{A_1}\|\cdots\|S_{A_{m-1}} \\
&= \varphi(k_0,\ g_0)\|\varphi(k_1,\ g_1)\|\cdots\|\varphi(k_{m-1},\ g_{m-1}), \\
S_B &= S_{B_0}\|S_{B_1}\|\cdots\|S_{B_{m-1}} \\
&= \varphi(g_0,\ h_0)\|\varphi(g_1,\ h_1)\|\cdots\|\varphi(g_{m-1},\ h_{m-1}), \\
S_C &= S_{C_0}\|S_{C_1}\|\cdots\|S_{C_{m-1}} \\
&= \varphi(h_0,\ k_0)\|\varphi(h_1,\ k_1)\|\cdots\|\varphi(h_{m-1},\ k_{m-1}).
\end{aligned} \tag{11}
$$

③ A는 C에게, B는 A에게, C는 B에게 각각 다음과 같은 P_A, P_B, P_C를 동시에 전송한다.

$$
\begin{aligned}
P_A &= P_{A_0}\|P_{A_1}\|\cdots\|P_{A_{m-1}} \\
&= \varphi(h_0\circ k_0,\ g_0)\|\varphi(h_1\circ k_1,\ g_1)\|\cdots\| \\
&\quad \varphi(h_{m-1}\circ k_{m-1},\ g_{m-1}), \\
P_B &= P_{B_0}\|P_{B_1}\|\cdots\|P_{B_{m-1}} \\
&= \varphi(k_0\circ g_0,\ h_0)\|\varphi(k_1\circ g_1,\ h_1)\|\cdots\| \\
&\quad \varphi(k_{m-1}\circ g_{m-1},\ h_{m-1}), \\
P_C &= P_{C_0}\|P_{C_1}\|\cdots\|P_{C_{m-1}} \\
&= \varphi(g_0\circ h_0,\ k_0)\|\varphi(g_1\circ h_1,\ k_1)\|\cdots\| \\
&\quad \varphi(g_{m-1}\circ h_{m-1},\ k_{m-1}).
\end{aligned} \tag{12}
$$

④ A, B, C는 각각 자신의 키 K_A, K_B, K_C를 계산한다.

$$
\begin{aligned}
K_A &= K_{A_0}\|K_{A_1}\|\cdots\|K_{A_{m-1}} \\
&= (S_{A_0}\times P_{B_0})\|(S_{A_1}\times P_{B_1})\|\cdots\|(S_{A_{m-1}}\times P_{B_{m-1}}), \\
K_B &= K_{B_0}\|K_{B_1}\|\cdots\|K_{B_{m-1}} \\
&= (S_{B_0}\times P_{C_0})\|(S_{B_1}\times P_{C_1})\|\cdots\|(S_{B_{m-1}}\times P_{C_{m-1}}), \\
K_C &= K_{C_0}\|K_{C_1}\|\cdots\|K_{C_{m-1}} \\
&= (S_{C_0}\times P_{A_0})\|(S_{C_1}\times P_{A_1})\|\cdots\|(S_{C_{m-1}}\times P_{A_{m-1}}).
\end{aligned} \tag{13}
$$

⑤ 식 (5)로부터 A, B, C가 계산한 키는 다음과 같이 동일함을 알 수 있다.

$$
\begin{aligned}
&\forall t \in \{0,1,\cdots,m-1\}, \\
&K_{A_t} = S_{A_t}\times P_{B_t} \\
&\quad = \varphi(k_t,\ g_t)\varphi(k_t\circ g_t,\ h_t) \\
&\quad = \varphi(g_t,\ h_t)\varphi(g_t\circ h_t,\ k_t) = S_{B_t}\times P_{C_t} = K_{B_t} \\
&\quad = \varphi(h_t,\ k_t)\varphi(h_t\circ k_t,\ g_t) = S_{C_t}\times P_{A_t} = K_{C_t}.
\end{aligned} \tag{14}
$$

⑥ 이제 A, B, C는 공통의 세션키 $K=K_A=K_B=K_C$를 이용하여 안전한 통신을 할 수 있다.

IV. 재킷 행렬 기반 알고리즘과의 비교 및 안전성 분석

2장에서 언급한 바와 같이, [1,2]에서는 코사이클 재킷 행렬을 이용하면서 $g\circ h$를 비트 단위 XOR 연

산으로 정의하였다. 이에 따라, 제안하는 2자간 키 동의 알고리즘에서 A는 g와 h를 이미 알고 있고 B로부터 $h \circ k$를 얻음으로써 $k = h \circ (h \circ k)$를 알 수 있으며 이는 B나 3자간 알고리즘의 경우에도 마찬가지이다. 이는 키 동의의 의미를 잃게 하는 결과를 낳는다. 본 논문에서는 이러한 문제를 해결하기 위하여 TA를 도입하였다. 2자간 알고리즘의 경우, h를 A(또는 B)와 공유하지 않음으로써 A(또는 B)가 k(또는 g)를 알 수 없도록 하였다. 3자간 알고리즘 역시, A(또는 B, C)에게 h와 k(또는 g와 h, h와 k)에 대한 정보를 제공하지 않기 때문에 이를 알 수가 없다. 부수적으로 [1,2]에서 언급되지 않았던 인증 문제도 이 TA가 각 사용자를 인증하여 주는 것으로 해결할 수 있다.

제안하는 2자간 알고리즘에서 A가 자신이 알고 있는 정보(g_t, $h_t \circ k_t$, $\varphi(g_t, h_t)$, $\varphi(g_t \circ h_t, k_t)$)를 이용하여 나머지 비밀 정보($h_t$ 또는 k_t)를 알아내려고 한다고 하자. 식 (1)에서 보면 Sylvester Hadamard 행렬은 첫 번째 행과 첫 번째 열을 제외하고는 각 행 또는 열의 원소로 1과 -1이 각각 절반씩 존재하므로, $\varphi(g_t, h_t)$로부터 가능한 h_t의 범위를 절반으로 줄일 수 있다. 이 때 $h_t \circ k_t$를 이용하여 각 h_t에 해당하는 k_t 값이 정해진다. 이제 $\varphi(g_t \circ h_t, k_t)$로부터 가능한 h_t의 범위를 다시 절반으로 줄일 수 있다. 따라서 A가 이와 같은 방법을 통하여 나머지 비밀 정보를 알 수 없도록 하기 위해서는 $n \geq 3$, 즉 8×8 이상의 Sylvester Hadamard 행렬을 적용하여야 한다. 이는 3자간 알고리즘에서도 마찬가지이다. 반면 재킷 행렬을 사용할 경우 행렬의 원소로 ± 1, $\pm w$의 4가지가 존재하므로 같은 방법으로 보면 $n \leq 4$, 즉 16×16 이하의 크기로는 안전성을 보장할 수 없다. 결과적으로 m비트의 세션키를 공유하기 위해서 재킷 행렬을 사용할 때에는 최소 $5m$비트의 g, h, k가 필요한 반면, Sylvester Hadamard 행렬을 사용하면 최소 $3m$비트로 가능하다. 또한 제안하는 알고리즘에서는 [1,2]와는 달리 w가 존재하지 않아 이와 관련된 계산이 필요하지 않으므로 상대적으로 처리 속도가 그만큼 향상될 것이다.

V. 결 론

[1,2]에서 제안한 코사이클 재킷 행렬 기반 키 동의 알고리즘은 빠른 속도로 실행될 수 있도록 하였지만 키 동의의 의미를 상실하게 되는 문제가 있다. 본 논문에서는 이러한 문제점을 확인하고 이를 해결할 수 있는 Sylvester Hadamard 행렬 기반 키 동의 알고리즘을 제안하였다. 추가적으로 [1,2]의 코사이클 재킷 행렬을 적용할 때보다 Sylverster Hadamard 행렬을 적용할 경우에 성능 향상 또한 기대할 수 있다는 점도 확인하였다. 제안하는 알고리즘에서 키 동의 과정에 소요되는 시간은 키의 길이 m과 사용되는 행렬의 크기를 결정하는 n의 곱, $N = mn$에 비례하므로 지수 연산이 필요한 다른 기법에 비하여 빠른 속도로 키 동의를 할 수 있다.

제안하는 알고리즘을 이용하여 기존의 공개키 혹은 비밀키 암호 방식을 사용하지 않고 공통의 세션키를 공유할 수 있으며, 또한 지수 연산을 할 필요가 없어 계산 성능이 떨어지는 단말 간의 안전한 통신에 유용할 것이다.

참 고 문 헌

[1] C.H. Choe, G.Y. Hwang, S.H. Kim, H.S. Yoo, and M.H. Lee, "Key agreement protocols based on the center weighted Jacket matrix as a symmetric co-cyclic matrix," MRCS 2006, LNCS 4105, pp. 121-127, 2006.

[2] C.H. Choe, J. Hou, S.J. Choi, S.Y. Kim, and M.H. Lee, "Co-cyclic Jacket matrices for secure communication," Proc. of IWSDA'05, pp. 103-105, Oct. 2005.

[3] K.J. Horadam, Hadamard Matrices and Their Applications, Princeton University Press, pp. 10-11, Nov. 2006.

[4] W. Stallings, Cryptography and Network Security, 4th Ed., Prentice Hall, Nov. 2005.

[5] K.J. Horadam and P. Udaya, "Cocyclic Hadamard codes," IEEE Transactions on Information Theory, vol. 46, no. 4, pp. 1545-1550, July 2000.

[6] M.H. Lee, "The center weighted Hadamard transform," IEEE Transactions on Circuits and Systems, vol. 36, no. 9, pp. 1247-1249, Sep. 1989.

[7] 강명희, 유황빈, "유비쿼터스 컴퓨팅 환경에서의 익명성을 보장하는 사용자 인증 및 키 동의 프로토콜 설계," 정보보호학회논문지, 16(2), pp. 3-12, 2006년 4월.

[8] 최규영, 황정연, 홍도원, 이동훈, "비대칭 컴퓨팅 환경을 위한 ID-기반의 인증된 키 동의 프로토콜," 정보보호학회논문지, 16(1), pp. 23-33, 2006년 2월.

[9] 신성철, 이성운, "동일 서버를 사용하는 두 사용자간 효율적인 패스워드 기반의 키 교환 프로토콜," 정보보호학회논문지, 15(6), pp. 127-133, 2005년 12월.

[10] 박영호, 이경현, "효율성을 개선한 신원기반의 3자간 복수 키 합의 프로토콜," 정보보호학회논문지, 15(3), pp. 77-89, 2005년 6월.

[11] 이성운, 유기영, "간단하고 효율적인 상호 인증 키 동의 프로토콜," 정보보호학회논문지, 13(5), pp. 105-112, 2003년 10월.