情報保護學會論文誌
第14券 第6號, 2004. 12

k-역행렬을 이용한 메시지 인증 기법*

이 희 정† 김 태 권‡
강남대학교

Message Authentication Code based on k-invertible Matrices

Hee Jung Lee† Tae Gwon Kim‡
Kangnam University

요 약

메시지 인증 코드(MAC)란 메시지의 무결성을 입증하기 위해서나 사용자 인증 등에 사용되는 것으로 2003년 Crypto에서 Cary와 Venkatesan이 새로운 기법을 소개하였다. 비밀키 들을 이용하여 암호화된 값을 결정하고 행렬식이 ±1인 공개된 행렬들을 이용하여 메시지 인증코드를 생성하는 방식이다. 여기서 공개된 행렬들은 k-invertible(k-역행렬)이라는 특성을 갖게 되는데 이러한 k가 충돌이 일어나는 확률에 영향을 주게 된다. k를 작게 하는 행렬들을 선택하는 것이 중요한데 Cary 등은 임의의 행렬들을 소개하고 그것들이 k-역행렬이 되는 이유를 보여 주고 있다. 본 논문에서는 공개키로 사용되는 k-역행렬 들을 어떻게 선택하여야 하는 지를 살펴본다. 효율성을 높이기 위해서 행렬들의 성분들은 -1, 0, 1로만 제한한다. 특정한 성질을 갖는 22개의 행렬들 중에서 4개의 행렬을 선택할 때의 충분조건을 알아보고 이들의 k값도 살펴본다. 또한, Cary등이 제안한 것보다는 효율성과 안정성이 향상된 k=5인 행렬들을 소개한다.

ABSTRACT

MAC is used for data origin authentication or message integrity protection. In Crypto'03 Cary and Venkatesan introduced new MAC based on unimodular matrix groups. It is to encrypt messages using private keys and to encrypt them again using public keys which are matrices whose determinants are ±1. These matrices have property called k-invertible. This k effects on the collision probability of this new MAC. The smaller k is, the less collisions occur. Cary shows 6-invertible matrices, and 10-invertible matrices whose components are only 1, 0, -1. In this paper we figure out sufficient conditions about choosing 4 matrices among special 22 matrices. Also, we introduce 5-invertible matrices whose components are 1, 0, -1. Those have better efficiency and security.

Keywords : *Message authentication, k-invertible matrix, Public key*

Ⅰ. 서 론

MAC은 메시지의 무결성이나 메시지의 출처 확인을 위해서 많이 사용되고 있다. 2003년 Crypto에서 Cary와 Venkatesan[1]은 행렬식이 ±1인 행렬들을 이용하여 새로운 메시지 인증코드를 제안하였다. 기존의 방식이 비밀키를 사용하여 해쉬 함수를 선택하고 이를 메시지 인증 코드 생성에 사용하였다면 새로운 방식은 비밀키를 이용하여 메시지를 암호화하고 암호화한 내용을 공개키를 사용하여 다시 암호화하는 것이다. 여기서 비밀키를 사용한 평문과 암호문은 일대일 대응이 되

접수일 : 2004년 9월 20일 ; 채택일 : 2004년 12월 5일
* 본 논문은 2004년도 강남대학교 교내 지원비로 연구되었음.
† 주저자 : hjlee@kangnam.ac.kr
‡ 교신저자 : ktg@kangnam.ac.kr

지 않으며 공개키를 이용한 메시지 인증코드도 충돌이 일어날 수 있다. Cary 등은 새로운 MAC 생성 방식이 최근에 사용되고 있는 UMAC[2]과 같은 빠른 MAC알고리즘들과 비교해서 효율성이나 안전성 측면에서 견줄 만하거나 더 낫다고 주장하고 있다.

새로 제안된 메시지 인증 기법은 공개키로 2×2 행렬들을 사용하고 있는데 이들은 행렬식이 ±1로써 k-역행렬이라는 특성을 갖고 있다.

정의: 나열된 2×2 행렬들 $A_1,\cdots,A_t$에서 모든 $i<j$에 대해서 Δ를 $\Delta = det(A_i\cdots A_{j-1}-I)$라 할 때 만약 Δ가 0이 아니고 임의의 $k'\le k$에 대해서 $2^{k'}\mid\Delta$를 만족하면 행렬 $A_1,\cdots,A_t$를 k-*invertible*(k-역행렬)이라고 한다.

공개키 행렬들이 갖는 k값은 메시지 인증 기법의 충돌 확률에 영향을 준다. 구체적으로 l비트 크기의 서로 다른 두 메시지가 충돌할 확률은 $2^{-2l+4+k}$ 이다.[1] 따라서 k값을 작게 할 필요가 있다. Cary 등은 특정한 3개의 행렬을 반복하여 50개의 공개된 행렬을 사용하였을 경우 k가 6이 되는 것을 보였고 효율성을 높이기 위하여 원소들을 0,1,-1만을 사용하였을 경우 15%정도 효율성은 증가되었으나 안전성은 떨어졌다. 그러나 어떠한 이유에서 그러한 행렬들을 선택했는지 또는 그러한 행렬들의 k값이 최소인지에 대해서는 언급이 없었다. 따라서 본 논문에서는 공개키를 선정하는 방법에 대해서 알아보려고 한다. 이를 위해서 구현을 다소 단순화하고 효율성도 높이기 위해서 행렬의 원소들을 1, 0, -1로 제한한다. 이와 같은 조건에서 가장 작은 k는 얼마인지 그러한 값을 갖기 위해서는 어떤 행렬들을 선택하여야 하는지에 대해서 알아보려고 한다. 첫 번째로, 주어진 행렬들로부터 Δ가 0이 되지 않을 조건을 언급하려고 한다. 이는 k-역행렬이 되기 위한 필요조건이다. 두 번째로, 필요조건을 만족하는 22개의 행렬들 중에서 행렬의 구성원들이 모두 같은 부호를 갖는 행렬 6개를 선정하여 임의의 4개의 행렬을 선택하여 반복했을 때 선택된 행렬들의 나열된 위치에 따라 k-역행렬이 되기도 하고 그렇지 않기도 한 것을 발견한다. 먼저 M_{ij}를 $\prod_{s=i}^{j-1}A_s$라고 하자. k-역행렬이 되기 위한 충분조건을 살펴보면 선택된 4개의 행렬들이 반복되어질 때 네 개의 행렬들 중에서 연속된 두 행렬 또는 세 행렬, 네 행렬이 $det(M_{ij})-tr(M_{ij})\neq -1$이 되도록 나열하면 충분함을 보인다. 그러나 이를 이론적으로 증명하지는 못하고 구현에 의해서 보여준다. 세 번째로, 필요조건을 만족하는 22개의 행렬들 중에서 임의의 4개 행렬들을 선택하여 반복적으로 나열할 경우 연속된 12개 행렬이내에서 $det(M_{ij})-tr(M_{ij})\neq -1$이 되면 공개키로 적합함을 알 수 있다. 그러나 이러한 현상도 구현에 의해서 보여 진다. 네 번째로, 필요조건을 만족하는 행렬들의 k값은 어떻게 나타나는 지에 대해서 조사해 본다. 4개의 행렬들을 선택하여 반복할 경우는 k를 7까지 찾아낼 수 있고 7개의 행렬들을 선택하여 50개까지 반복하여 공개키를 사용할 경우 k를 5까지 줄일 수 있다.

2장에서는 [1]에서 사용한 행렬들과 그들의 결과를 살펴보고 0,1,-1들로 이루어진 행렬들 중에서 모두 같은 부호를 갖는 원소들로 이루어진 행렬 6개 중에서 4개를 공개키로 선택할 경우 충분조건을 알아본다. 또한 22개의 행렬들 중에서 4개를 선택할 경우 공개키가 되는 조건을 알아본다. 3장에서 7개의 행렬들을 선택하여 반복한 경우 k가 5인 행렬들을 소개한다.

II. {1,0,-1}만을 이용한 k-역행렬 분석

새로 제안된 메시지 인증코드 생성 기법은 대략 다음과 같다. 먼저 메시지 X를 임의의 개수를 갖는 블록으로 나눈다. 이때 한 블록은 l비트 크기의 워드 t개를 갖는다고 하자. 마지막 블록에서 필요시에 패딩을 한다. 한 블록에 대한 메시지 인증코드는 비밀키로 암호화한 후에 공개된 행렬들을 이용하여 다시 암호화한다. 여기서 비밀키로 암호화한다는 것은 l비트의 비밀키와 l비트의 메시지를 곱하여 2차원의 벡터로 만드는 것이다. 이러한 벡터를 공개된 행렬들과 합하여 확장된 행렬들을 만들고 이들의 곱으로부터 메시지 인증코드를 찾아내는 것을 공개키 암호화 과정이라 한다. 한 블록에 대한 메시지 인증 코드를 만들고 그것

들이 다음 블록의 초기값이 되어 연속적으로 마지막 블록까지 반복되는데 이러한 과정을 서로 다른 두개의 비밀키와 공개키 체계를 이용하여 따로 구한 후 두개의 코드를 보낸다. 이것이 새로 제안된 메시지 인증 코드(MAC) 기법이다.

Cary 등은 아래의 세 행렬들 을 반복 사용하여

$$A_1=\begin{pmatrix}-1 & 1\\ 1 & -2\end{pmatrix},\ A_2=\begin{pmatrix}2 & 1\\ 1 & 1\end{pmatrix},\ A_3=\begin{pmatrix}1 & 3\\ 1 & 2\end{pmatrix}$$

A_{19}은 4-역행렬이고 A_{50}은 6-역행렬임을 보였다.[1] 이때 A_t란 t개의 행렬들이 공개키로 사용된다는 뜻이다. 또한 Cary 등은 효율성을 높이기 위해서 원소들이 1, -1, 0으로 구성된 아래의 네 행렬들을 사용하였는데 이들을 순서대로 t개 나열한 경우에는 대략적으로 $\log_{1.5}t$-invertible 행렬이 되는 것을 보였다.

$$B_1=\begin{pmatrix}1 & 1\\ 0 & -1\end{pmatrix},\ B_2=\begin{pmatrix}0 & 1\\ 1 & 1\end{pmatrix},\ B_3=\begin{pmatrix}-1 & 0\\ 1 & -1\end{pmatrix},\ B_4=\begin{pmatrix}1 & 1\\ 1 & 0\end{pmatrix}$$

따라서 A_i들의 경우와 달리 1, -1, 0 만을 사용한 행렬들은 효율성은 높아지지만 안전성은 다소 떨어진다.

같은 행렬들이더라도 행렬을 어떻게 나열하여 반복하는 지에 따라 공개키로서 사용가능 여부가 달라진다. 어떻게 해서 이러한 행렬들을 선택하게 되었으며 이들 주어진 행렬들의 k가 가장 작은 값인 지에 대한 언급이 없었다. 따라서 0, 1, -1만을 사용하여 행렬식이 ±1인 모든 행렬들을 구하고 이들을 어떻게 선택하여 어떻게 반복했을 때 k가 가장 작아지는 지를 살펴보려고 한다. 0, 1, -1만을 사용하여 행렬식이 +1인 행렬들은 모두 40개가 되었다. 이들 중에서 [1]이 선택한 행렬들은 연속되어 곱하여진 행렬들의 trace의 절대값이 2보다 크거나 같았다.

정리 1 : (필요조건) $A_1, A_2, \cdots, A_t$을 순서대로 나열된 행렬식이 ±1이고 원소들이 {0,1,-1}인 2×2 행렬들이라 할 때, 임의의 i, j에 대해서, $M_{ij}=\prod_i^{j-1}A_s$ ($1\le i<j\le t$) 이라면, $det(M_{ij}-I)\neq 0$ 이기 위해서는 $det(M_{ij})-tr(M_{ij})\neq -1$ 이어야 한다.

증명: $det(M_{ij}-I)=det(M_{ij})+1-tr(M_{ij})$ 따라서 $det(M_{ij}-I)\neq 0$이기 위하여 $det(M_{ij})-tr(M_{ij})\neq -1$이어야 한다. □

행렬식이 ±1인 40개의 행렬들 중에서 한 개의 행렬 자체가 위의 필요조건을 만족하는 행렬은 ± 단위행렬을 포함하여 24개로 압축되었다. [1]에서 선택한 4개의 행렬 B_i들도 이들 24개중에 포함되어 있다.

1. 같은 부호를 갖는 원소들로 이루어진 6개 행렬들 중에서 4개를 선택하는 경우

〈원소들이 0, 1, -1로 되어 있으며 행렬식이 ±1이고 각 행렬 자체가 $detA-trA\neq -1$을 만족하는 행렬들〉

$$\begin{pmatrix}-1 & -1\\ -1 & 0\end{pmatrix}\quad\begin{pmatrix}-1 & -1\\ 0 & -1\end{pmatrix}\quad\begin{pmatrix}-1 & -1\\ 1 & 0\end{pmatrix}$$
$$\begin{pmatrix}-1 & 0\\ -1 & -1\end{pmatrix}\quad\begin{pmatrix}-1 & 0\\ 1 & -1\end{pmatrix}\quad\begin{pmatrix}-1 & 1\\ -1 & 0\end{pmatrix}$$
$$\begin{pmatrix}-1 & 1\\ 0 & -1\end{pmatrix}\ \begin{pmatrix}-1 & 1\\ 1 & 0\end{pmatrix}\ \begin{pmatrix}0 & -1\\ -1 & -1\end{pmatrix}\ \begin{pmatrix}0 & -1\\ -1 & 1\end{pmatrix}$$
$$\begin{pmatrix}0 & -1\\ 1 & -1\end{pmatrix}\ \begin{pmatrix}0 & -1\\ 1 & 0\end{pmatrix}\ \begin{pmatrix}0 & -1\\ 1 & 1\end{pmatrix}\ \begin{pmatrix}0 & 1\\ -1 & -1\end{pmatrix}$$
$$\begin{pmatrix}0 & 1\\ -1 & 0\end{pmatrix}\ \begin{pmatrix}0 & 1\\ -1 & 1\end{pmatrix}\ \begin{pmatrix}0 & 1\\ 1 & -1\end{pmatrix}\ \begin{pmatrix}0 & 1\\ 1 & 1\end{pmatrix}$$
$$\begin{pmatrix}1 & -1\\ -1 & 0\end{pmatrix}\ \begin{pmatrix}1 & -1\\ 1 & 0\end{pmatrix}\ \begin{pmatrix}1 & 1\\ -1 & 0\end{pmatrix}\ \begin{pmatrix}1 & 1\\ 1 & 0\end{pmatrix}$$

어떠한 행렬들을 공개키로 선택가능한 지를 알아보기 위해서 문제를 먼저 간단히 축소해서 알아보자. 위의 행렬들 중에서 같은 부호를 갖는 원소들로 이루어진 행렬 6개를 선택하여 이들 중에서 4개의 행렬을 선택하여 반복할 경우 어떻게 나열하여야 공개키로 가능하며 동시에 k값은 얼마인지를 알아보려고 한다.

다음 행렬들을 편의상 순서대로 각각 0,1,2,3,4,5 라고 부르자.

$\begin{pmatrix}-1 & 0\\ -1 & -1\end{pmatrix}$, $\begin{pmatrix}-1 & -1\\ 0 & -1\end{pmatrix}$, $\begin{pmatrix}-1 & -1\\ -1 & 0\end{pmatrix}$, $\begin{pmatrix}0 & 1\\ 1 & 1\end{pmatrix}$, $\begin{pmatrix}0 & -1\\ -1 & -1\end{pmatrix}$, $\begin{pmatrix}1 & 1\\ 1 & 0\end{pmatrix}$.

이들 중에서 4개의 행렬을 순서를 고려하여 선택하였더니 모두 90개의 경우가 나왔다. 이들을 사용하여 t가 50일 때까지 반복하였더니 k-역행렬의 필요조건인 $det(M_{ij}-I)\neq 0$을 만족하는 행렬들의 경우는 모두 26개였다. 여기서 순환하는 행렬들은 한 개로 본다. 예를 들어, 0123=1230=2301=3012. 이때 k값은 7,8,9,10이 나왔다. 그러면 64개의 행렬들의 순서쌍은 왜 $det(M_{ij}-I)=0$이 되었는지를 살펴보아야 한다. 이들은 선택한 4개의 행렬들 중에서 연속된 2개, 또는, 3개 또는, 4개의 행렬들의 곱이 행렬식과 트레이스(tr)값과의 차가 -1이 되는 것을 알아내었다. 다시 말하면 선택된 4개의 행렬들만을 살펴봄으로써 공개키 여부를 판단할 수 있다. 구체적으로 살펴보면 위의 행렬들 중에서 3과 5나 2와 4가 연속적으로 나열되거나 세 개의 행렬들 032(320, 203), 450,(504,045), 123,(231,312), 415,(154, 541)가 연속적으로 선택되거나 또는 0315(=3150=1503=5031)나 0412가 선택된 경우는 모두 $det(M_{ij}-I)=0$가 되었다. 따라서 다음과 같은 결과를 알아 낼 수 있다.

정리 2: 0, 1, -1만을 사용한, 행렬식이 ±1인 행렬들 중에서 같은 부호를 갖는 원소들로 이루어진 행렬 6개를 선택하여 그들 중에서 임의의 4개를 선정하여 공개키를 생성한다고 하자. 이때 선택한 4개의 행렬들이 연속적으로 나열되었을 때 j=3,4,5에 대해서 $det(M_{ij})-tr(M_{ij})\neq -1$이면 k-역행렬이 된다.

증명: 위에 언급되었듯이 구현을 통하여 알 수 있다. □

정리 2를 만족하는 공개키는 모두 26개가 생성되었데 이들 중에서 k값은 아래 표 1과 같다.

표 1. 공개된 행렬들에 따른 k값

k	선택된 행렬들과 나열순서(t=50)
7	0234, 0254, 0523, 0543, 1325, 1345, 1432, 1452
8	0215, 0314, 0413, 0512
9	0125, 0152, 0341, 0431, 2345, 2543
10	0134, 0143, 0213, 0214, 0251, 0513, 0514, 0521

2. 필요조건을 만족하는 22개 행렬들 중에서 임의의 4개를 선택하는 경우

22개의 행렬들 중에서 임의의 4개를 선택할 경우는 정리2에서 제안한 조건을 충족시킴에도 불구하고 공개키로 가능하지 않는 경우들이 발생하였다. 총 43890개중에서 258개의 공개키가 생성되었는데 이때 k는 최저 7이 나왔다. 공개키로 사용할 수 없는 경우들을 살펴보니 다음과 같은 결론을 얻을 수 있었다.

정리 3: 필요조건을 만족하는 22개 행렬들 중에서 임의의 행렬 4개를 선택하여 공개키로 사용하기 위해서는 j=13까지 $det(M_{ij})-tr(M_{ij})\neq -1$이면 k-역행렬이 된다.

증명: 구현에 의해서 알 수 있다. □

Ⅲ. k=5 인 행렬들

Cary 등이 제시한 가장 작은 k값은 6이었다. 이때 원소들에 1,2,3 등을 포함시킨 행렬들을 사용하였고 3개의 행렬을 50개까지 반복하여 사용하였다. 위의 장에서는 1,0,-1만을 사용하여 k-역행렬이 되기 위한 필요조건을 만족하는 행렬들 중에서 같은 부호를 갖는 원소들로 이루어진 6개의 행렬들을 선택하고 그들 중에서 4개를 선택할 경우 가능한 모든 공개키들을 살펴보았다. 그때 가장 작은 k값은 7임을 알 수 있었다. 7보다 작은 k값을 찾기 위해서 22개의 행렬들 중에서 서로 다른 행렬들 5개, 6개, 7개, 8개를 선택하여 반복했을 때 조사해보았다. 5개를 선택하여 반복하였을 때는 가장 작은 k는 7이였고 6개를 선택하여 반복했을 경우는 6, 그리고 7개를 선택하여

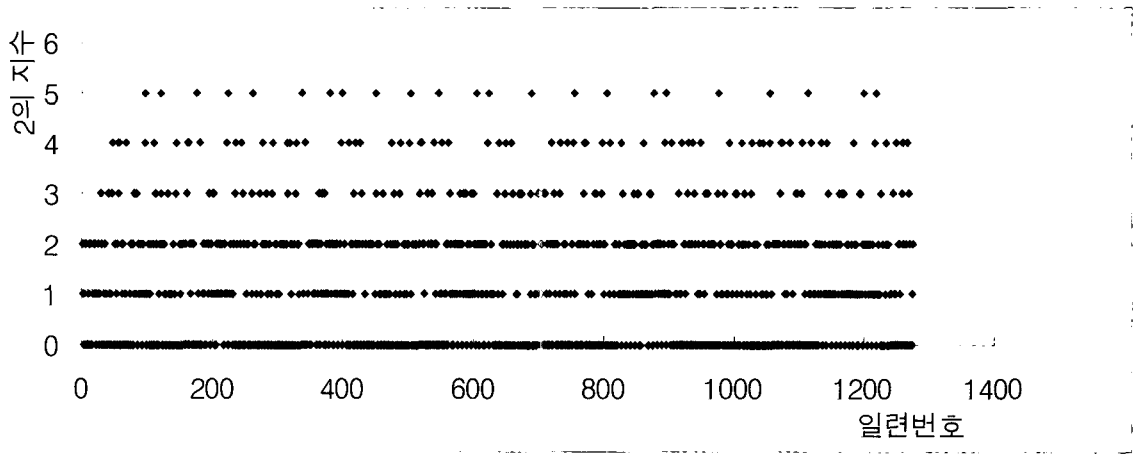

그림 1. 위의 7개 행렬들의 k값

* y축은 $2^k \mid det(\prod_i^j A_s - I)$를 만족하는 k를 나타내고 x축은 연속되는 행렬들의 시작과 끝이 i, j인 일련번호를 뜻한다.

반복해서는 5가 됨을 알 수 있었다. 8개를 선택하였을 때는 다시 6이 가장 작은 k값이었다. 서로 다른 행렬들을 많이 선택하여 반복할수록 k를 찾는데 걸리는 시간이 길었다. 7개를 선택할 경우 1024종류의 공개키가 생성되었는데 이중에서 k가 5가 되는 공개키는 모두 8개였다. 그 중에서 다음 공개키의 경우를 소개하려고 한다.

정리 4: 다음 행렬들을 선택하여 순서대로 50개의 행렬들을 반복하여 공개키로 사용할 경우 이들은 5-역행렬이 된다.

$$\begin{pmatrix}-1 & 0\\ 1 & -1\end{pmatrix}\begin{pmatrix}-1 & -1\\ 1 & 0\end{pmatrix}\begin{pmatrix}0 & -1\\ 1 & -1\end{pmatrix}\begin{pmatrix}0 & 1\\ -1 & -1\end{pmatrix}\begin{pmatrix}-1 & 1\\ -1 & 0\end{pmatrix}$$
$$\begin{pmatrix}-1 & 1\\ 0 & -1\end{pmatrix}\begin{pmatrix}1 & -1\\ -1 & 0\end{pmatrix}$$

증명: 직접 구현에 의해서 알 수 있다. 그림 1에서 k의 생성과정을 볼 수 있다. □

7개의 행렬들을 선택하여 공개키로 사용할 경우 충분조건이 무엇인지에 관한 조사는 아직 이루어지지 않았다. 4개를 선택하는 경우와 마찬가지로 선택된 7개 행렬들의 인접한 행렬 곱들이 $det(M_{ij} - I) \neq 0$이라 할지라도 공개키가 되지 못하는 경우도 있었다.

Ⅳ. 결 론

새로운 메시지 인증코드 생성 기법에서 공개키로 사용되어지는 행렬들의 특성을 살펴보았다. 효율성을 높이기 위하여 원소들을 0, 1, -1로 제한하여 행렬식이 ±1인 것들을 찾아보니 40개의 행렬들이 생성되었다. 이들 중에서 공개키가 되기 위한 필요조건을 만족하는 행렬들 22개를 찾아내어($\pm I$ 제외) 7개의 행렬을 선택해서 이들을 50개의 행렬로 반복 나열하였을 때 가장 작은 k값을 찾아 낼 수 있었다. 이때 k는 5이다. 4개의 행렬을 선택할 경우 모든 공개키들을 찾아내었는데 k값은 최저 7이었다. 몇 개의 행렬들을 어떻게 나열하는 것이 k값을 가장 작게 할 것인가를 살펴보았는데 8개까지의 행렬들을 선택하여 k값을 찾아보았다. 같은 부호를 갖는 원소들로 이루어진 6개의 행렬들 중에서 4개의 행렬을 선택할 경우 4개의 행렬들 간에만 $det(M_{ij}) - tr(M_{ij}) \neq -1$을 만족하면 충분하다는 것을 보였다. 22개의 행렬들 중에서 임의의 4개 행렬을 선택하여 사용할 경우는 이들이 연속적으로 3번 반복되기 전까지 $det(M_{ij}) - tr(M_{ij}) \neq -1$이면 공개키로 가능하다는 것을 알아내었다. 어떠한 근거로 이와 같은 일이 일어나는 지는 이론적으로 밝히지 못하였다. 앞으로 행렬들을 어떻게 선정하여 어떻게 나열하

여야 하는지에 관한 충분조건을 찾아내어야 한다. 또한 k값을 가장 작게 하는 공개키는 어떻게 찾아낼 수 있는지에 관하여서도 계속적으로 연구되어져야 할 것이다.

참 고 문 헌

[1] Mattew. Cary, Ramarathnam Venkatesan, "A Message Authentication Code Based on Unimodular Matrix Groups", Crypto 2003, LNCS 2729, pp.500-512, 2003.

[2] J. Black, S .Halevi, H. Krawczyk, T. Krovetz, P. Rogaway, "UMAC : Fast and Secure Message Authentication", Advances in Cryptology-Crypto '99, LNCS, vol. 1666, M. Wiener,ed. Springer-Verlag, pp.216-233, 1999.

[3] D. Bernstein, "Factoring-point arithmetic and message authentication", draft available as http://cr.yp.to/papers/hash127.dvi.

[4] Mihir Bellare, Joe Kilian, Phillip Rogaway, "The security of the cipher block chaining message authentication code", Journal of computer and system science, 61(3), 362-399, 2000.

[5] Mariusz H. Jakubowski, R. Venkatesan, "The chain and sum primitive and its application to MACs and stream ciphers", In Advances in Cryptology-Eurocrypt '98, vol.1403, LNCS, 281-293, Springer-Verlag, 1998.

[6] 홍도원, 신상욱, 강주성, 이옥연, "3GPP MAC 알고리즘 안전성 분석", 정보보호학회논문지, 제11권 제2호, pp.52~59, 2001.

[7] 임채훈, 이필중, "상호 신분인증 및 디지털 서명 기법에 관한 연구", 정보보호학회논문지, 제2권 1호, 1992.

〈著者紹介〉

이 희 정 (Hee Jung Lee) 정회원
1980년 2월 : 이화여자대학교 문리대학 수학과 졸업
1989년 8월 : 펜실베니아 주립대학교(Penn. State Univ.) 이학박사
1994년 3월~현재 : 강남대학교 응용수학 전공 부교수
〈관심분야〉 정수론, 암호학

김 태 권 (Tae Gwon Kim)
1986년 2월 : 서울대학교 자연과학대학 물리학과 졸업
1994년 2월 : 서울대학교 컴퓨터과학 이학박사
1994년 3월~현재 : 강남대학교 컴퓨터공학 전공 부교수
〈관심분야〉 데이타베이스, 정보보안