通信情報保護學會論文誌
第 8 卷 第 4 號. 1998. 12

해쉬 알고리듬 개발 및 디지털 이동통신을 위한 인증 시스템에의 응용

이국희*, 이상곤**, 정원영***, 김태근***, 문상재*

Development of Hash Algorithm and Its Application to the Authentication System for Digital Mobile Communication

Kook-Heui Lee*, Sang-Gon Lee**, Won-Young Jeong***,
Tae-Guen Kim***, Sang-Jae Moon*

요 약

이동통신에서의 사용자 인증 서비스는 통화도용 방지와 신뢰성 있는 과금을 위한 중요한 보호 서비스이다. 본 논문에서는해쉬함수를 개발하고 이를 IS-95A 인증 시스템의 인증 알고리듬과 인증 키 생성 알고리듬을 적용하였다. 특히 인증 알고리듬을 활용하여 인증 키 생성알고리듬을 oracle 해쉬함수의 형태로 구현함으로써 인증 시스템의 높은 안전성과 간결성을 동시에 성취하였다. 그리고 통계적 분석 기법을 사용하여 개발된 알고리듬의 출력 특성을 분석한다.

Abstract

The user authentication service can be used to prevent telecommunications piracy and to demand reliable payment from subscriber. In this paper, we developed a hash function and applied it to the authentication algorithm and the authentication key generation algorithm of IS-95A authentication system. By implementing the authentication key generation algorithm with the from of oracle hash function, we accomplished the high security and simplicity of the authentication simultaneously. We analyze the randomness properties of developed algorithms using statistical analysis.

key word : 인증 알고리듬, 해쉬함수, Oracle hash, 통계적 분석

이 연구는 한국통신의 98년도 정보통신 기초연구비 지원에 의한 결과의 일부임.
* 경북대학교 전자전기공학부
** 동서대학교 전자기계공학부
*** 한국통신 가입자망 연구소

Ⅰ. 서 론

이동통신서비스 이용자 수의 증가와 더불어 전파를 통신매체로 이용하는 이동통신의 특성 때문에 불법적인 사용이나 도청 또는 추적을 통한 불법적인 행위와 같은 각종 통신 범죄행위 등도 증가하고 있다. 특히 통화도용은 이동통신서비스 사업자에게는 요금징수와 관련한 피해를 주며, 가입자에게는 요금체계에 대한 불신감을 주어 이동통신 발달에 큰 장해요인이 된다. 이러한 통화도용을 방지하기 위하여 인증 서비스가 필요하다[1]. 이동통신에서의 인증 서비스란 기지국과 이동국간의 합법적인 통신을 위하여 서로 공유한 비밀 데이터가 일치하는지를 확인하는 것을 의미한다. IS-95A[2], GSM[3] 그리고 PACS[4] 등과 같은 이동통신 표준안은 기본적으로 인증 서비스를 제공하고 있다.

지난 몇 년 동안에 걸쳐서 IS-41 북미 이동전화 시스템에 사용자 데이터 보호용 스트림 암호 시스템인 ORYX[24] 와 제어채널 보호용 블록암호 시스템인 CMEA[25], 그리고 사용자 인증을 위한 CAVE[26] 알고리듬이 공격을 당하였다. 그리고 이러한 알고리듬들을 대치할 새로운 알고리듬의 개발작업이 추진중이다[27].

본 논문에서는 디지틀 이동통신의 인증 알고리듬에 적용할 목적으로 해쉬함수를 개발하고 이를 이용하여 국내 개인휴대통신의 표준안인 IS-95A의 인증 시스템을 위한 인증 알고리듬과 인증키 생성 알고리듬을 개발하였다. 인증 키 생성 알고리듬은 인증 알고리듬을 활용하여 oracle 해쉬함수[5,6]의 형태로 구성함으로써 시스템의 높은 안전성과 간결성을 동시에 성취하였다. 그리고 통계적 해석 기법을 사용하여 개발된 알고리듬의 출력특성을 분석하였다. 본 논문에서 제안한 알고리듬은 입력과 출력의 변화를 통하여 GSM과 PACS와 같은 다른 이동통신 표준안에도 적용될 수 있다.

Ⅱ. IS-95A의 인증 메카니즘

IS-95A, GSM 그리고 PACS는 모두 challenge-response 형의 인증 시스템을 채택하고 있다[7]. 즉, 기지국이 임의의 challenge 데이터를 주면 이동국은 자신의 비밀키로 인증 알고리듬을 수행하여 response, 즉, 인증서명 데이터, 를 전송한다. 그리고 기지국은 이동국과 동일한 방법으로 인증 서명 데이터를 생성하여 그 값이 일치하면 이동국을 합법적인 사용자로 간주한다.

IS-95A의 경우 challenge에 대한 response를 계산하는 알고리듬을 Auth_Signature 절차라 명한다. Auth_Signature 절차는 그림 1과 같이 152비트 데이터를 입력으로 하여 18비트의 response를 생성한다.

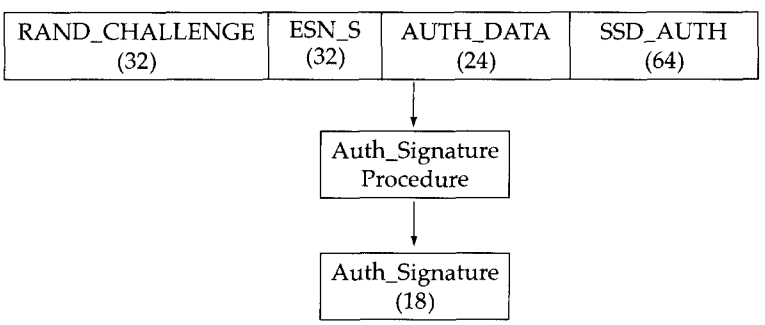


그림 1. Auth_Signature의 계산
Fig 1. Computation of Auth_Signature.

인증 알고리듬의 키이(key)로 사용되는 SSD는 SSD_Generation 절차를 사용하여 갱신된다. SSD_Generation 절차는 이동국 특정 정보, 랜덤 데이터, 이동국 A-key를 입력으로 가진다. A-key는 64비트이며 이동국의 영구 보안 및 식별 메모리에 저장된다. A-key는 단지 이동국과 그에 관련된 HLR/인증 센터만 알 수 있다. 이동국이 SSD 갱신 메시지를 수신하면, 그림 2과 같이 SSD_Genaration 절차의 입력 파라미터를 설정하여 SSD_Generation 절차를 수행한다. 이동국은 SSD_A_NEW와 SSD_B_NEW 값을 SSD_Generation 절차의 출력 값으로 설정한다.

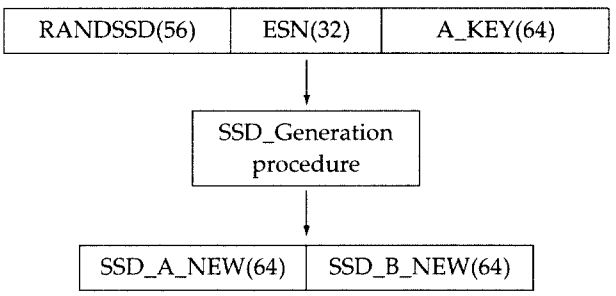


그림 2. 공유 비밀 데이터(SSD)의 계산
Fig. 2. Computation of Shared Secret Data (SSD).

본 논문에서의 연구내용은 그림1의 Auth_Signature 절차와 그림 2의 SSD_Generation 절차의 알고리듬을 개발하는 것이다.

Ⅲ. 디지틀 이동통신에서의 인증 시스템을 위한 해쉬함수의 개발

1. 제안된 해쉬 함수의 기술

1.1 기본적인 기호 및 용어

해쉬 함수를 기술하는데 사용할 용어와 기호는 다음과 같이 정한다. 워드(word)는 해쉬함수가 처리하는 기본 데이터 단위로서 32 비트이다. 길이가 같은 두 개의 이진 데이터 열 S_1, S_2의 비트별(bit-wise) 모듈러(modular) 2 곱셈과 덧셈은 $S_1 \cdot S_2$ 그리고 $S_1 \oplus S_2$로 나타낸다. $\cdot$은 $\oplus$보다 계산상 우선 순위가 앞선다. $S_1 = W_{1,n-1}W_{1,n-2}\cdots W_{1,0}$, $S_2 = W_{2,n-1}W_{2,n-2}\cdots W_{2,0}$라 하자. 여기서 $W_{i,j}$는 32비트 워드이다. S_1과 S_2의 워드별(word-wise) 모듈러 2^{32} 정수 덧셈을 $S_1 \boxplus S_2$로 표시하면 $S_1 \boxplus S_2 = (W_{1,n-1} + W_{2,n-1} \bmod 2^{32})(W_{1,n-2} + W_{2,n-2} \bmod 2^{32})\cdots(W_{1,0} + W_{2,0} \bmod 2^{32})$이다. $\overrightarrow{ROT}(X, s)$는 한 개의 단어 X를 s번 오른쪽으로 순환이동(rotate) 시키는 동작이다. 그리고 두 함수 f와 g의 합성을 f$\circ$g로 표시하는데 함수 g가 먼저 계산된다.

일반적인 해쉬함수의 동작절차를 설명하려면 전처리과정, 입력데이터 로딩, 초기화, 단계연산(step operation)과정 그리고 해쉬 데이터 출력의 과정 등을 설명하여야 하지만, 본 논문에서 개발하는 해쉬함수는 IS-95A에 적용될 것이므로, 전처리 과정과 입력데이터 로딩 절차는 해쉬함수를 인증 시스템에 적용 시 고려되어야 한다. 따라서 본 절에서는 이들 두 부분을 제외한 나머지 부분만 설명한다.

개발된 해쉬함수는 256 비트(8 워드)가 입력된다. 라운드 수는 4 라운드이며, 각 라운드에서는 8개의 워드을 이용하여 8회의 단계연산을 수행한다. 1 라운드와 2 라운드에서는 8개의 연쇄변수가 사용되고, 3라운드와 4라운드에서는 6개의 연쇄변수가 사용된다. 4 라운드의 마지막 단계연산 후

의 연쇄변수와 초기 상수 값을 더하여 해쉬값이 출력이다.

1.2 초기화

8 워드 초기화 상수 시퀀스인 식 (1)의 E_0를 표 1과 같이 초기화한다.

$$E_0 = (E_{0,7}, E_{0,6}, \cdots, E_{0,0})$$

표 1. 상수값들
Teble 1. Contants

파라미터	상수값
$E_{0,7}$, $E_{0,6}$, ⋯, $E_{0,0}$	926cfbe5, f7b46bce, 835fd1a0, 2c1e9f23, 45cbfa73, eb64749a, 96215eda, bac2f6e1
$K_{2,7}$, $K_{2,6}$, ⋯, $K_{2,0}$	5190cfef, a784d904, 38b4da56, 62e7160f, 9cf4f3c7, bf715880, 8aed2a6a, b7e15162
$K_{3,7}$, $K_{3,6}$, ⋯, $K_{3,0}$	90cfd47d, 57f59584, 4f7c7b57, bb1185eb, da06c80a, 8c31d763, f4bf8d8d, 324c7738
$K_{4,7}$, $K_{4,6}$, ⋯, $K_{4,0}$	cfbfa1c8, 8a9a276b, 839a2ddf, 613c31c3, fd5f24d6, d55c4d79, 158d9554, 7c19bb42

1.3 단계연산과정

4개의 각 라운드에서는 다음과 같이 단계연산이 이루어진다.

[1 라운드]

① 우선 $T_{0,i} = E_{0,i}$, $0 \le i \le 7$ 라 두자.

② i=0 에서 7까지 다음의 과정을 반복한다.

$P = \mathrm{F1} \cdot \phi_1(T_{i,7}, T_{i,6}, T_{i,5}, T_{i,4}, T_{i,3}, T_{i,2}, T_{i,1})$;

$R = \overrightarrow{ROT}(P, 13) \boxplus T_{i,7} \boxplus W_i$;

$T_{i+1,7} = T_{i,6}$; $T_{i+1,6} = T_{i,5}$; $T_{i+1,5} = T_{i,4}$; $T_{i+1,4} = T_{i,3}$;

$T_{i+1,3} = \overrightarrow{ROT}(T_{i,2}, 11)$; $T_{i+1,2} = T_{i,1}$; $T_{i+1,1} = T_{i,0}$;

$T_{i+1,0} = R$.

③ $E_1 = (E_{1,7}\ E_{1,6}\ E_{1,5}\ E_{1,4}\ E_{1,3}\ E_{1,2}\ E_{1,1}\ E_{1,0})$

$= (T_{8,7}\ T_{8,6}\ T_{8,5}\ T_{8,4}\ T_{8,3}\ T_{8,2}\ T_{8,1}\ T_{8,0})$

$T_{i,j}$는 각 라운드에서 j회 단계연산 후의 i번째 단어를 의미한다. P값은 i번째 단계연산의 입력 ($T_{i,7}$, $T_{i,6}$, $T_{i,5}$, $T_{i,4}$, $T_{i,3}$, $T_{i,2}$, $T_{i,1}$)이 ϕ_1에 따라 치환되고 난 후에 부울함수 F1()에 의하여 계산된다. 이와 같은 치환은 다른 라운드에서 사용되는 부울함수들 간에 상호 output uncorrelation을 제공하기 위한 것으로 표 2와 같이 좌표축변환을 행한다.

표 2. 좌표축 변환
Teble 2. Permutations on coordination

치환	x6	x5	x4	x3	x2	x1	x0
ϕ_1	x2	x6	x1	x4	x5	x3	x0
ϕ_2	x5	x3	x2	x0	x1	x6	x4
ϕ_3			x4	x0	x1	x2	x3
ϕ_4			x0	x2	x3	x4	x1

[2 라운드]

① 우선 $T_{0,i} = E_{1,i}$, $0 \le i \le 7$ 라 두자.

② i=0에서 7까지 다음의 과정을 반복한다.

$P = \mathrm{F}_2 \cdot \phi_2(T_{i,7}, T_{i,6}, T_{i,5}, T_{i,4}, T_{i,3}, T_{i,2}, T_{i,1})$

$R = \overrightarrow{ROT}(P, 13) \boxplus T_{i,7} \boxplus W_{ord2(i)} \boxplus K_{2,i}$;

$T_{i+1,7} = T_{i,6}$; $T_{i+1,6} = T_{i,5}$; $T_{i+1,5} = T_{i,4}$; $T_{i+1,4} = T_{i,3}$;

$T_{i+1,3} = \overrightarrow{ROT}(T_{i,2}, 11)$; $T_{i+1,2} = T_{i,1}$; $T_{i+1,1} = T_{i,0}$;

$T_{i+1,0} = R$.

③ $E_1 = (E_{2,7}\ E_{2,6}\ E_{2,5}\ E_{2,4}\ E_{2,3}\ E_{2,2}\ E_{2,1}\ E_{2,0})$

$= (T_{8,7}\ T_{8,6}\ T_{8,5}\ T_{8,4}\ T_{8,3}\ T_{8,2}\ T_{8,1}\ T_{8,0})$

1라운드에서와 마찬가지로 i번째 라운드의 입력 ($T_{i,7}$, $T_{i,6}$, $T_{i,5}$, $T_{i,4}$, $T_{i,3}$, $T_{i,2}$, $T_{i,1}$)는 ϕ_2에 따라 치환되고 난 후에 부울함수 F_2()에 의하여 P값이 계산된다. ϕ_2은 좌표축 치환으로서 표 2에 규정되어 있다. R 값의 계산에서 $W_{ord2(i)}$는 각 단계연산에서

사용되는 입력 데이터인데 ord2(i)는 2라운드에서 사용되는 워드의 처리순서를 의미한다. 2, 3, 4 라운드에서 처리되는 입력 데이터의 순열을 표 3에 나타내었다. 각 단계연산에서 사용되는 상수 $K_{2,i}$는 16진수로 표 1에 나타내었다.

표 3. 입력단어의 처리순서
Teble 3. word processing order

단계순서 / 라운드	0	1	2	3	4	5	6	7
1	0	1	2	3	4	5	6	7
2	3	6	0	4	1	7	2	5
3	1	5	6	2	0	3	7	4
4	6	4	2	1	3	5	0	7

[3 라운드]

① 우선 $T_{0,i} = E_{3,i}$, $0 \le i \le 5$라 두자.

② i=0에서 7까지 다음의 과정을 반복한다.

$P = F_3 \cdot \phi_3 (T_{i,5}, T_{i,4}, T_{i,3}, T_{i,2}, T_{i,1})$

$R = \overrightarrow{ROT}\ (P, 13) \boxplus T_{i,0} \boxplus T_{i,5} \boxplus W_{ord3(i)} \boxplus K_{3,i}$;

$T_{i+1,5} = T_{i,4}$; $T_{i+1,4} = T_{i,3}$; $T_{i+1,3} = \overrightarrow{ROT}\ (T_{i,2}, 11)$;

$T_{i+1,2} = T_{i,1}$; $T_{i+1,1} = T_{i,0}$; $T_{i+1,0} = R$.

③ $E3 = (E_{3,5}\ E_{3,4}\ E_{3,3}\ E_{3,2}\ E_{3,1}\ E_{3,0})$
$= (T_{8,5}\ T_{8,4}\ T_{8,3}\ T_{8,2}\ T_{8,1}\ T_{8,0})$

3 라운드에서는 5 변수 부울함수 $F_3()$를 사용함으로 희생되는 안전도를 보상해 주기 위하여 R을 구하는 모듈러 2^{32} 정수 덧셈 항에 $T_{i,0}$ 연쇄변수를 하나 더 첨가하였다. P값의 계산 순서는 2 라운드와 동일하여 ϕ_3는 표 2에 규정되어있다. ord3(i)는 3 라운드의 각 단계연산에서 사용되는 워드의 처리순서를 의미하는 것으로 표 3에 나타내었다. 각 단계연산에서 사용되는 상수 $K_{3,i}$는 16진수로 표 1에 나타내었다.

[4 라운드]

① 우선 $T_{0,i} = E_{3,i}$, $0 \le i \le 7$라 두자.

② i=0에서 7까지 다음의 과정을 반복한다.

$P = F_4 \cdot \phi_4 (T_{i,5}, T_{i,4}, T_{i,3}, T_{i,2}, T_{i,1})$

$R = \overrightarrow{ROT}\ (P, 13) \boxplus T_{i,0} \boxplus T_{i,5} \boxplus W_{ord4(i)} \boxplus K_{4,i}$;

$T_{i+1,7} = T_{i,6}$; $T_{i+1,6} = T_{i,5}$; $T_{i+1,5} = T_{i,4}$; $T_{i+1,4} = T_{i,3}$;

$T_{i+1,3} = \overrightarrow{ROT}\ (T_{i,2}, 11)$; $T_{i+1,2} = T_{i,1}$; $T_{i+1,1} = T_{i,0}$;

$T_{i+1,0} = R$.

③ $E_4 = (E_{4,7}\ E_{4,6}\ E_{4,5}\ E_{4,4}\ E_{4,3}\ E_{4,2}\ E_{4,1}\ E_{4,0})$
$= (T_{8,5}\ T_{8,4}\ T_{8,3}\ T_{8,2}\ T_{8,1}\ T_{8,0})$

4라운드에서도 3 라운드에서와 마찬가지로 R을 구하는 모듈러 2^{32} 정수 덧셈 항에 $T_{i,0}$ 연쇄변수를 하나 더 첨가하였다. P값의 계산 순서는 2 라운드와 동일하여ϕ_4는 표 2에 규정되어있다. ord4(i)는 4 라운드에서 사용되는 워드의 처리순서를 의미하는 것으로 표 3에 나타내었다. 각 단계 연산에서 사용되는 상수 $K_{4,i}$는 16진수로 표 1에 나타내었다.

1.4 데이터의 출력

아래 식 (2)와 같이 4 라운드의 마지막 단계연산 후의 연쇄변수 $E_{4,i}$와 초기 상수 값 $E_{0,i}$을 더하여 해쉬함수의 출력으로 한다.

$$E_{4,i} = E_{4,i} \boxplus E_{0,i}, \quad i=0, \cdots, 5 \qquad (2)$$

2. 제안된 해쉬 함수의 설계 기준

2.1 이동통신의 인증 시스템에 적합한 알고리듬으로서 해쉬함수의 선정

IS-95A와 관련된 미국의 특허[8]에서는 인증 알고리듬과 인증 키 생성 알고리듬에 "Jumbel"이라는 해쉬함수를 사용하였으며, 다른 형태의 해쉬함수도 사용할 수 있다고 언급하였다. 본 논문에서는 해쉬함수를 사용하여 인증 알고리듬을 구현하였다. 해쉬함수는 부울함수를 사용하는 형태와 블록암호화 알고리듬을 사용하는 형태 그리고 모듈라 연산을 사용하는 형태로 구분된다[9]. 이동통신시스템에서는 고속처리가 가능한 Bool 함수를 사용한 해쉬함수를 사용하는 것이 유리하다[8]. 부울함수를 사용한 해쉬함수로는 MD5[10], SHA-1[11], RIPEMD-160[12],HAVAL[13] 등이 있다. 본 논문에서는 이들 함수에 대한 공격[14, 15]과 반복연산을 분석

하여 인증 알고리듬의 설계에 반영하였다.

2.2 부울함수

이동통신에서 실시간으로 처리되어야 하는 알고리듬인 것을 고려할 때 안전성과 신속성을 동시에 만족시키기 위하여 7변수 부울함수 2개와 5변수 부울함수 2개를 사용하였다. 그리고 해쉬함수에서 사용되는 부울함수는 다음과 같은 안전성 기준[13,16]을 만족하는 것을 선정하여 사용하였다.

(1) (0-1 balance) : 부울함수의 출력은 0-1 balance 이어야 한다.

(2) (nonlinearity) : 비 선형적(high nonlinear) 이어야 한다.

(3) (SAC) : strictly avalanche criterion 를 만족해야 한다.

(4) (pairwise linearly nonequivalent) : 입력좌표를 선형 변환하여 해쉬함수에 사용되는 함수 집합내의 다른 함수로 변환되지 않아야 한다.

(5) (mutual output uncorrelate) : 두 부울함수의 출력은 상호 비 연관적이어야 한다.

선정된 부울함수 각각은 안전성 기준 1, 2, 3을 만족하고, 부울함수들 사이에는 안전성 기준 4와 5를 만족한다. 특히 안전성 기준 5는 표 2와 같이 좌표축 선형 변환을 해줌으로서 만족된다. 사용된 7변수와 5변수 부울함수의 비선형도는 각각 56과 12이다. 7변수 부울함수의 이론상 최대 비선형도는 58이고, 5변수 부울함수의 최대 비선형도는 12이다.

2.3 순환이동 및 이동량

RIPEMD-160에서는 a 레지스터, 입력메시지, 상수값 그리고 부울함수 출력을 모듈라 연산한 결과와 c 레지스터 등 두 곳에 좌측 순환이동을 적용함으로써 differential attack[14]에 대비하였다. 그리고 SHA-1 는 a와 b 레지스터 두 곳에 순환이동이 있으며, HAVAL은 부울함수의 출력과 h 레지스터에 순환이동을 적용함으로써 differential attack에 대비하였다. 위의 예들을 분석해 볼 때, 순환이동을 적용하는 레지스터의 위치는 적절히 선택하고 두 곳 정도에 순환이동을 적용함으로써 differential attack을 피할 수 있음을 알 수 있다. 본 알고리듬에서는 부울함수의 출력과 c 레지스터(Ⅲ장 1절의 T_{12})에 순환이동을 적용하였다.

RIPEMD-160과 MD5는 단계마다 다른 순환 이동량을 사용하며, HAVAL과 SHA-1은 동일한 순환 이동량을 사용하였다. RIPEMD-160에서는 순환 이동량을 5~15의 값으로 하였으며, MD5에서는 처리 데이터 단위(32비트)에 대하여 상대소수가 되도록 하였다. 이때 주의할 점은 각 순환이동 레지스터는 어떤 특정한 패턴을 갖지 않도록 하여야 한다는 것이다. 예를 들어, 전체 순환 이동량은 32의 배수가 되지 않아야 하며, 순환상수는 4의 배수가 되지 않도록 한다. 본 알고리듬에서는 부울함수의 출력에 적용되는 순환 이동량은 13, c 레지스터의 경우는 11로 하였다.

2.4 입력 메시지의 처리순서

MD4와 RIPEMD에서는 각 단계연산에서 사용되는 메시지 입력 순서에 약점이 발견되어 공격을 당하였다[14,15]. RIPEMD을 개선한 RIPEMD-160에서는 round 1-2에서 가까이 있었던 두 단어는 2-3라운드에서는 멀리 떨어져 있도록 하였다. 반대로 round 1-2에서 멀리 떨어져 있었던 두 단어는 2-3라운드에서는 가까이 있도록 하였다. 본 알고리듬에서도 이러한 면을 고려하여 아래 표 3과 같이 입력 메시지를 배열하였다. 256비트의 입력 데이터는 8개의 32비트 워드로 나뉘어져 순서에 따라 각 라운드의 단계연산에 적용된다.

2.5 프로세스의 구조와 알고리듬의 관계

대부분의 최신 프로세스들은 superscalar 구조로 설계되어 pipeline processing이 가능하므로 해쉬함수의 단계연산을 이러한 구조에 맞게 설계하면 소프트웨어 구현이 매우 효율적이다[18, 19]. 제안된 알고리듬의 경우 매 단계연산에서 갱신되

는 R 값은 연이은 단계연산의 부울함수 입력으로 사용되지 않기 때문에 다음 단계연산에서 사용될 부울함수의 값을 미리 계산해 놓을 수 pipeline processing이 가능하다.

Ⅳ. 해쉬함수의 IS-95A에의 적용

1. 인증 알고리듬(Auth__Signature procedure)에의 적용

1.1 알고리듬의 구조

그림 3은 Ⅲ장에서 제안한 해쉬 알고듬을 IS-95A 메카니즘에 적용한 AUTH_SIGNATURE의 구조이다.

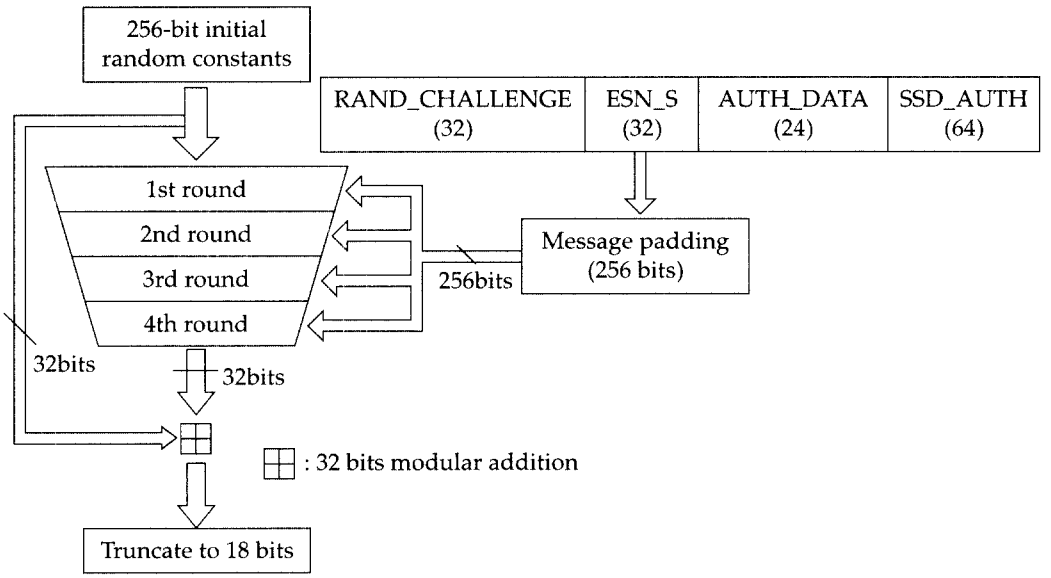


그림 3. AUTH__SIGNATURE 알고리듬의 구조
Fig 3. The structure of AUTH_signature algorithm.

개발된 해쉬함수는 인증 알고리듬은 256비트의 입력 데이터가 필요한 반면, IS-95A 인증 메카니즘에서는 그림 1과 같이 152비트의 데이터가 입력되므로 104비트의 메워넣기(padding) 데이터를 만들어야한다. 그림 4는 152비트의 데이터를 사용하여 256비트의 입력데이터를 만드는 방법을 도식화한 것이다. W(2)에서 AUTH_D는 24비트이므로 32비트로 만들기 위하여 입력 데이터의 길이 152(0X98)를 8비트로 표현하여 padding 데이터로 설정하였다.

192비트의 해쉬함수 출력 가운데 $E_{4,0}$의 최하위 18비트를 AUTH_SIGNATURE 알고리듬의 출력으로 취한다.

RAND_C (32)	ESN_S (32)	AUTH_D (24)	0x98 (8)	SSD_A[0] (32)	SSD_A[1] (32)	PAD[0] (32)	PAD[1] (32)	PAD[2] (32)
W(0)	W(1)	W(2)		W(3)	W(4)	W(5)	W(6)	W(7)

$$PAD[0] = RAND \oplus ESN\_S \oplus AUTH\_D$$
$$PAD[1] = PAD[0] \oplus SSD\_A[0]$$
$$PAD[2] = PAD[1] \oplus SSD\_A[1]$$
$$w(2) = AUTH\_D \,\|\, 0x98, \quad \| : \text{bit concatenation}$$

그림 4. AUTH_SIGNATURE 알고리듬을 위한 데이터 메워넣기
Fig. 4. Data padding for AUTH_SIGNATURE algorithm.

1.2 알고리듬의 안전성 분석

인증 알고리듬의 출력이 가져야할 성질에는 랜덤성, SAC 그리고 입출력간의 비선형성이 있다[17]. SAC란 알고리듬의 입력중 한 비트가 변할 경우 출력의 각 비트가 변할 확률이 1/2이 됨을 의미하며, 입출력간의 비선형성이란 입력과 출력의 특정 비트들 간의 선형관계가 성립하지 않음을 의미한다. 개발된 인증 알고리듬이 이러한 특성을 가지는지 검사하는 방법으로 통계적 분석방법을 사용한다. 출력의 랜덤성을 검사하기 위하여 equidistribution test, hamming weight 그리고 runs test를 사용한다. 입출력간의 비선형성 검사를 위하여 linear dependency test를 사용한다. equidistribution test을 통하여 출력 데이터 패턴이 균일한 분포를 가지는지 검사하고, hamming weight test를 통하여 출력의 각 비트에서 1이 발생할 확률이 1/2인지 검사한다. 그리고 runs test를 통하여 출력의 각 비트 위치에서 이전 비트에 대하여 변화가 발생할 확률이 1/2인지 검사한다.

통계적 분석(statistical analysis)이란 임의의 랜덤 프로세스로부터 구한 일련의 관측치(observations)를 가지고 있으며, 이러한 관측치가 특정한 성질을 가지는가 검사하는 것이다. 여기서 관측치가 가지기를 바라는 성질을 영가정(null hypothesis)이라 하며, 반대의 경우를 선택가정(alternate hypothesis)이라 한다. 통계적 분석을 위하여 관측치와 영가정간의 차이에 해당하는 통계치를 계산한다. 통계치의 계산방식은 다를 수 있으나 통계치가 작을수록 관측치가 영 가정에 가까워짐을 의미한다. 통계치의 확률분포가 P(x)라 주어질 때, P(x>a)를 임의의 통계치 a에 대한 유의수준(significance level)이라 부른다. 통계치로부터 계산된 유의 수준과 임계치를 비교하여 영 가정의 채택여부를 가린다. 일반적으로 사용되는 임계치는 0.05이다. 임계치가 0.05란 말은 영가정이 참이지만 거짓이라는 잘못된 결론을 내릴 확률이 0.05라는 의미가 된다. 그러므로 통계치가 임계치보다 클 경우에는 영가정을 채택한다. 본 논문에서는 chi squared goodness-of-fit test기법을[20] 사용하여 통계치의 계산과 영가정의 채택여부를 판별한다.

각각의 평가 항목에 대하여 동일한 실험을 다른 데이터를 사용하여 20번 반복하였다. Fisher-Pearson test를[21] 통하여 20개의 유의수준으로부터 하나의 유의수준을 얻고 이 값을 임계치인 0.05와 비교한다. 표 4는 각 항목의 통계적 분석 결과를 나타낸다. 표 4의 첫째 칸의 입력특성은 인증 알고리듬의 입력 데이터에서 1이 차지하는 비율이 20%, 50%, 80% 임을 의미한다. 결과를 분석해 볼 때 설계된 인증 알고리듬의 출력은 통계적으로 양호한 특성을 가짐을 알 수 있다. linear dependency 검사를 위한 행렬은 행 감소(row reduction)를 수행한 결과 full rank를 가지며, 이는 알고리듬의 입력과 출력간에 선형성이 존재하지 않음을 의미한다.

표 4. 인증 알고리듬에 대한 유의수준
Teble 4. Data padding for AUTH_SIGNATURE algorithm.

분석항목 / 입력특성	Eguidistribution	Hamming weight	Runs	SAC
20%	0.071466	0.066234	0.839401	0.670356
50%	0.143139	0.413732	0.156449	0.199088
80%	0.522117	0.546972	0.175676	0.347470

2. 인증 키 생성 알고리듬(SSD_Generation)에의 적용

인증 키 생성 알고리듬의 간결성을 도모하기 위하여 인증 알고리듬과 동일한 루틴을 사용하도록 한다. 본 절에서는 인증 키 생성 알고리듬의 근본적 구조를 제공하는 oracle 해쉬함수의 특성과 그 구현 방안[5, 6]에 간략히 관하여 서술한다.

2.1 Oracle 해쉬함수

Oracle 해쉬함수는 동일한 입력에 대하여 해쉬함수를 수행할 때마다 서로 다른 해쉬값을 낸다. 이러한 랜덤화는 해쉬함수 H(x)가 x에 관한 모든 부분정보(partial information)를 숨기는 요구조건을 제공한다. 그리고 이러한 확률적 성질을 가짐에도 불구하고 주어진 해쉬값에 대하여 어떤 입력으로부터 생성되었는지 검증할 수 있는 능력을 갖는다.

Oracle hashing의 정의는 아래 3개의 요구조건으로 구성된다.

- **완전성(Completeness)** : c가 x에 H를 적용하여 생성되었다고 할 때, 검증 알고리듬(verification algorithm) V는 x와 c의 쌍을 받아들일 것이다.
- **정확성(Correctness)** : c가 x에 대하여 해쉬함수 H를 적용하여 생성되지 않았다고 할 때, x와 c 쌍을 받아들이도록 V를 속일 수 없다. 다시 말하면, c를 x와 y의 합법적 해쉬값으로 받아들일 수 있는 두 개의 다른 입력 x , y 와 c를 찾는 것이 불가능하다.
- **비밀성(oracle security)** : c=H(x)인 c를 가지더라도 x에 관한 아무런 정보를 얻을 수 없다. 게다가 c=H(x)되는 x를 찾기 위해 전 도메인을 완전탐색 할 수도 없다.

해쉬함수의 입력에 랜덤성을 부과하여 아래와 같이 3개의 간단한 oracle hashing을 구현할 수 있다[5].

형 태 1) H(x, r) = r, h(r, x)

h는 해쉬함수이고 r은 길이가 β인 랜덤열이다. MD5의 경우는 β=128, SHA의 경우에는 β=160이 적절하다. 검증과 완전성은 간단하다. 즉 주어진 r에 대하여 x의 해쉬값을 계산하여 이전에 계산된 것과의 일치여부만 확인하면 되는 것이다. 정확성은 해쉬함수 h의 충돌 저항성을 따른다. 비밀성의 요구조건은 h에 관한 아래의 정의를 따른다.

정의 (τ, δ)-비밀성[5, 22]

만약 시간 τ내에 수행되는 어떤 adversary A와 distinguishier D에 대하여 아래의 조건이 성립하면, 해쉬함수 h는 H(x, r) = r, h(r, x)와 {0, 1}* 상의 어떤 Δ분포에 대하여 (τ, δ)-비밀성이다.

$$| \text{Prob}(D(x, A(r, h(r, x)))=1) - \text{Prob}(D(x, A(r,h(r,y)))=1) | \leq \delta \qquad (3)$$

여기서 {0, 1}*는 길이가 유한한 이진열을 나타낸다. 그리고 x와 y는 각각 Δ와 $r \in_r \{0,1\}^\beta$로부터 독립적으로 추출한 것이다.

형 태 2) H(x, r) = r, h(r, h(x))

완전성과 정확성은 형태 1)의 구조와 같다. 안전성도 유사하게 정의할 수 있다. 형태 2)가 형태 1)보다 안전하다. 왜냐하면 형태 2)가 공격당하면 형태 1)도 공격당하나 역은 성립하지 않기 때문이다.

형 태 3) H(x, r) = r, h(r_1, h(r_2, x))

이 구조는 HMAC(Hash based MAC)[23]에 기초를 하여 설계하였다. r_1 = r $\oplus$ opad, r_2 = r $\oplus$ ipad 그리고 ipad 와 opad는 고정된 상수 값이다. 완전성과 정확성은 위의 경우와 같다. 형태 3)은 훨씬 더 안전하다. 왜냐하면 형태 3)이 공격당하면 형태 1)과 형태 2)는 공격당하나 역은 성립되지 않기 때문이다.

2.2 Oracle 해쉬함수를 이용한 인증 키 생성 알고리듬의 구조

개발된 인증 키 생성 알고리듬은 2절에서 제시한 구조 중에서 h(r, h(x)) 형의 oracle hash 함수 구조를 취한다. 인증 키 생성 메카니즘에서는 oracle hash 알고리듬에서 H가 선택하는 랜덤변수 r 값

에 대응되는 값은 없다. 그러나 152비트의 입력 중 인증 키 갱신과정에서 인증 키 생성 함수가 수행될 때마다 바뀌는 RANDSSD(56비트)가 있으며, 이를 랜덤변수 r 값 대용으로 사용한다.

개발된 인증 키 생성 알고리듬의 전체적인 구조는 그림 5에 나타나 있으며 자세한 과정은 다음과 같다. 256비트 데이터가 해쉬함수에 입력되어 4라운드의 단계연산을 거쳐 192비트의 데이터가 출력된다. 이 해쉬함수를 수식으로 표현하여 h(w)라 하자. 여기서 w는 256비트의 입력 데이터이다.

과정 1) 그림 6과 같이 152비트 입력 데이타를 256비트의 1차 해쉬함수의 입력, W_1, 으로 만든다.

과정 2) 과정 1)의 W_1을 입력으로 한 1차 해쉬함수의 출력 h_out은 다음과 같다.

$h\_out = h(W_1)$

$= \{h\_out[0], h\_out[1], h\_out[2], h\_out[3], h\_out[4], h\_out[5]\}$

과정 3) 1차 해쉬값으로 나온 192비트와 randssd 56비트를 이용하여 그림 7과 같이 2차 해쉬함수의 입력 데이터 W_2를 구성한다.

과정 4) 2차 해쉬함수의 출력은 다음과 같이 계산된다.

$h\_out = h(W_2)$

$= \{h\_out[0], h\_out[1], h\_out[2], h\_out[3], h\_out[4], h\_out[5]\}$

과정 5) h_out의 192비트로부터 인증키 SSD_A와 SSD_B를 다음과 같이 취한다.

$SSD\_A = \{h\_out[0], h\_out[1]\}$

$SSD\_B = \{h\_out[2], h\_out[3]\}$

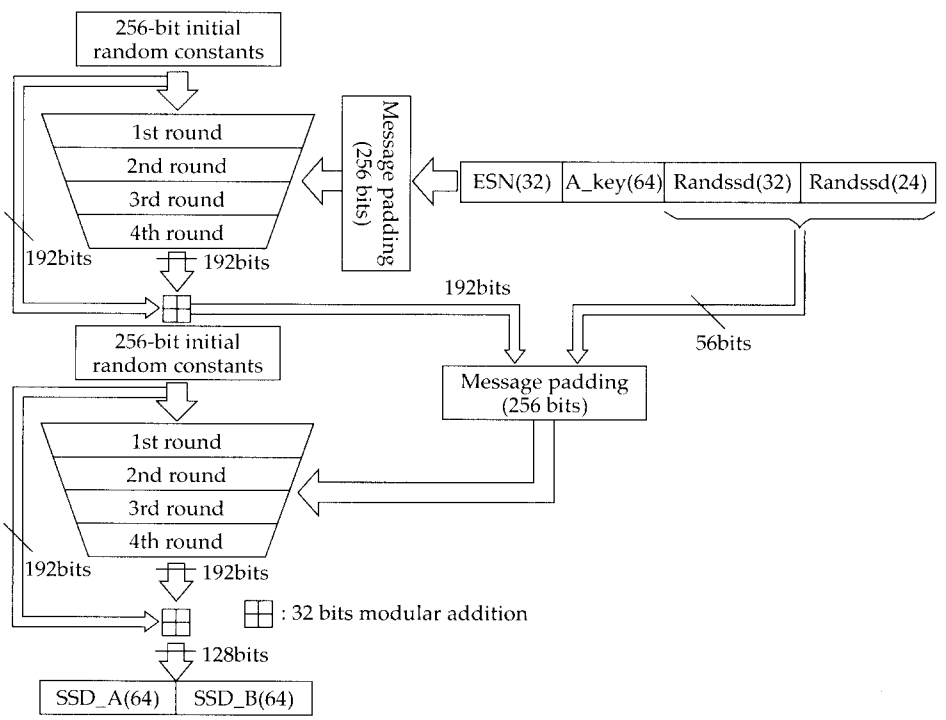


그림 5. 인증 키 생성 알고리듬의 블럭선도

Fig. 5. Block diagram of authentication key generation algorithm

ESN '(32)	A_key[0] (32)	A_key[1] (32)	Randssd[0] (32)	Randssd[1] (24)	0x98 (8)	PAD[0] (32)	PAD[1] (32)	PAD[2] (32)
$W_1(0)$	$W_1(1)$	$W_1(2)$	$W_1(3)$	$W_1(4)$		$W_1(5)$	$W_1(6)$	$W_1(7)$

PAD[0] = ESN ⊕ A_key[0] ⊕ Randssd[0]
PAD[1] = PAD[0] ⊕ A_key[1]
PAD[2] = PAD[1] ⊕ (Randssd[0] ‖ 0x98), ‖ : bit concatenation

그림 6. 1차 해쉬를 위한 데이터 메워넣기
Fig. 6. Data padding for the 1st hashing.

h_Out[0] (32)	h_Out[1] (32)	h_Out[2] (32)	h_Out[3] (32)	h_Out[4] (32)	Randssd[0] (32)	Randssd[1] (24)	0x98 (8)	PAD[0]
$W_2(0)$	$W_2(1)$	$W_2(2)$	$W_2(3)$	$W_2(4)$	$W_2(5)$	$W_2(6)$		$W_2(7)$

PAD[0] = h_out[5] ⊕ (Randssd[1] ‖ 0x98), ‖ : bit concatenation

그림 7. 2차 해쉬를 위한 입력 데이터 배치
Fig. 7. Inpuit data setting for the 2nd hashing.

2.3 인증 키 생성 알고리듬의 안전성 검토

인증 알고리듬에서의 분석방법과 동일하게 equidistribution test, hamming weight test, runs test를 수행하였으며, 그 결과를 표 5에 나타내었다. 모든 test 들의 유의수준이 임계치 보다 크므로 인증 키 생성 알고리듬의 출력특성도 양호함을 알 수 있다.

표 5. 인증 키 생성 알고리듬에 대한 유의수준
Teble 5. Significance level for the authentication key generation algorithm.

분석항목 / 입력특성	Equidistribution	Hamming weight	Runs
20%	0.286993	0.991408	0.928421
50%	0.358905	0.799294	0.989545
80%	0.587574	0.246302	0.913677

3. 속도 비교

본 논문에서 개발된 해쉬함수를 IS-95A에 적용한 경우의 연산 속도 비교를 위하여 비교 대상으로 IS-41의 인증 알고리듬과 인증 키 생성 알고리듬에 사용된 CAVE 알고리듬[28]을 선정하였다. CAVE 알고리듬은 IS-95A의 인증 알고리듬과 인증 키 생성 알고리듬에 같이 사용된다. 입력 데이터를 고정하고 각 알고리듬을 120,000번 반복 수행하는데 소요된 시간을 측정하여 표 6에 나타내었다. 그 결과 본 논문에서 제안한 해쉬함수를 이동통신을 위한 인증 시스템에 적용할 경우 기존의 CAVE 알고리듬 보다 수행속도가 향상됨을 알 수 있다. 알고리듬 수행속도 비교는 SUN Ultra II (200MHz)에서 이루어졌다.

표 6. 제안된 알고리듬과 CATV 알고리듬의 수행속도 비교

Teble 6. Significance level for the authenticationkey generation algorithm.

	CATV	개발된 알고리듬
Auth_Signature 절차	22.8673	0.450
SSD_Generation 절차	23.2839	0.922

V. 결 론

본 논문에서는 부울함수 기반의 해쉬함수를 개발하고 이를 이용하여 IS-95A 인증 시스템에 적용 가능한 인증 알고리듬을 개발하였다. 개발된 알고리듬은 4라운드의 해쉬함수이며 파이프라인 처리가 가능한 구조이다. 1라운드와 2라운드에는 7변수 부울함수를 사용하였으며, 3라운드와 4라운드에는 5변수 부울함수를 사용하였다. 또한 개발된 인증 알고리듬을 oracle 해쉬함수의 형태로 사용하여 IS-95A의 인증 키 생성 알고리듬을 개발하였다. 그리고 통계적 분석 방법을 사용하여 개발된 알고리듬의 출력 특성을 분석하였다. 출력 특성 분석에 사용된 검사로는 equidistribution test, hamming weight test, runs test, SAC test 그리고 linear dependency test가 있다. 통계적 분석을 적용한 결과 본 논문에서 제안한 인증 알고리듬과 인증 키 생성 알고리듬은 통계적으로 양호한 특성을 나타내었다. 개발된 인증 알고리듬과 인증 키 생성 알고리듬을 C언어로 구현하고, SUN Ultra Ⅱ에서 구현하여 수행속도를 측정한 결과 120,000번의 반복 수행에 각각 0.450초와 0.922초 정도가 소요되었다.

본 논문에서 제안한 알고리듬은 입력 데이터의 형태와 출력 데이터의 길이를 변경하면 GSM과 PACS와 같은 다른 이동통신 인증 시스템에도 동일하게 적용될 수 있다.

[참고문헌]

[1] ISO/IEC 7498-2, Information Processing-OSI Basic Reference Model - Part 2 : Security Architecture, 1989.

[2] TIA/EIA IS-95A, Mobile-Base Station Compatibility Standard for Dual-Mode Wideband Spread spectrum Cellular System, Interim Standard 95, July 1993.

[3] ETSI, European Digital Cellular Telecommunication System(phase 2) - Security Related Network Functions, July 1993.

[4] JTC, Personal Communications Services PACS Air Interface Specification, Jan. 1995.

[5] Ran Canetti, "Toward realizing Random Oracles : Hash Functions that Hides All Partial Information", Crypto'97, Spring-Verlag ,LNCS 1294, pp.455-469, 1997.

[6] M. Bellare, J. Kilian and P. Rogaway. "Random oracles are practical : a paradigm for designing efficient protocols," 1st ACM Conference on Computer and Communications Security, pp.62-73, 1993.

[7] 이국희, 류정수, 하재철, 문상재, "디지털 이동통신 인증 서비스의 기술동향," Telecommunication Review, vol.6, no.2, 1996

[8] J. A. Reeds Ⅲ, P. A. Treventi, Service Provision Authentication Protocol, U.S.A Patent No. 5153919, Oct. 6, 1992.

[9] C.J. Mitchell, K. Piper and P. Will, Contemporary Cryptology : The Science of Information Integrity, G.J.Simmons, editor, IEEE Press, pp.325-378, 1991.

[10] R. Rivest, "The MD5 message digest algorithm," Requests for Comments (RFC) 1321, 1992

[11] U.S. Department of Commerce(NIST), Secure Hash Standard, FIPS PUB 180-1, 1995. April 17.

[12] H. Dobbertin, A. Bosselaers and B. Preneel, RIPEMD-160 : A strengthened version of RIPEMD, Fast Software Encryption, LNCS1039, Springer-Verlg, 1996.

[13] Y. Zhang, J. Pieprzyk, and J. Seberry, "HAVAL - A One-way Hashing Algorithm with Variable Length of Output," Auscrypt'92 Abstract, 1992.

[14] B. den Boea and A. Bosselaers, "Collision for the compression function of MD5," Advances in Cryptology-Eurocrypt '93, LNCS 773, Spring-Verlag, pp.293-304, 1994.

[15] H. Dobbertin, " RIPEMD with two-round compression function is not collision free," Journal of Cryptology.

[16] S. Bakhtiary, R. Safavi-Naini, J. Pieprzky, "Keyed Hash Function," Cryptography : Policy and Algorithm, Springer-Verlag, July 1995.

[17] 이국희, 정명준, 이상곤, 문상재, 정원영, 김태근," 이동통신을 위한 인증 알고리듬 개발," WCT '97 워크샵자료집, pp.171-175, 1997. 12. 16.

[18] A. Bosselaers, R. Govaerts, and J. Vandewalls, " Fast hashing on the Pentium," Advances in Cryptology-Crypt '96, LNCS 1109, Spring-Verlag, pp.298-312., 996.

[19] A. Bosselaers, R. Govaerts, and J. Vandewalls, " SHA : a design for parallel architecture," proceeding of Eurocrypt '97, LNCS 1233, Spring-Verlag, pp.348-362, 1997.

[20] Paul G. Hoel, Introduction to Mathematical Statics, 4th ed., John Wiley & Sons, 1971.

[21] H. M. Gustafson, E. P. Dawson, and J. Dj. Goli , "Randomness Measures Related to Subset Occurrence," Cryptography: Policy and Algorithms, Brisbane, Queensland, Australia, pp. 132-143, July 1995.

[22] S. Bakhtiari, R. Safavi-Naini, and J. Pieprzyk, "Cryptographic Hash Functions: A Survey," Tech. rep. 95-09, Department of Computer Science, University of Wollongong, July 1995.

[23] S. Bakhtiari, R. Safavi-Naini, and J. Pieprzyk, "Practical and Secure Message Authentication," in Series of Annual Workshop on Selected Areas in Cryptography(SAC '95), (Ottawa, Canada), pp. 55-68, May 1995.

[24] D. Wagner, L. Simpson, E. Dawson, J. Kelsey, W. Millan, and B. Shneier, Cryptanalysis of ORYX. In wokrshop on Selected Areas in Cryptology 1998, Peproceedings, To appear in Springer-Verlag LNCS, pp. 403-317, 1998.

[25] D. Wagner, B. Schneier, and J. Kelsey, Cryptanalysis of the Cellular Message Encryption Algorithm. In Advances in Cryptology - Crypto '97 Proceedings, LNCS 1294, pp.526-537, Springer-Verlag, 1997.

[26] W. Millan, Cryptanalysis of the Aleged CAVE Algorithm, in the 1st International Conference on Information Security and Cryptology, Proceedings, pp.107-119, 998.

[27] Telecommuniactions Industry Association, New security algorithms : Call for development, 1998, can be found at http://scitech.ctia.org/Security/index.html

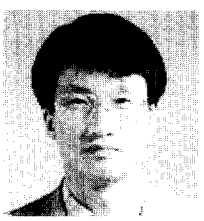
[28] TIA IS-54 Appendix A. Dual Mode Cellular System: Authentication, Message Encryption, Voice Privacy Mask Generation, Shared Secret Data generation, A-Key Verification, and Test Data, February 1992. Rev B. This document may be found at http://www.replay.com/mirror/cave/.

□ 著者紹介

이 국 희

1993년 2월 경북대학교 전자공학과 (학사)
1995년 2월 경북대학교 전자공학과(공학석사)
1005년 3월 ~ 현재 경북대학교 대학원 전자공학과 박사과정

※ 주관심 분야 : 암호이론, 이동통신

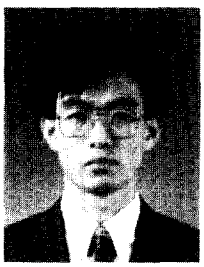


이 상 곤(종신회원)

1986년 2월 경북대학교 전자공학과(공학사)
1988년 2월 경북대학교 대학원 전자공학과(정보통신공학, 공학석사)
1993년 2월 경북대학교 대학원 전자공학과(정보통신공학, 공학박사)
1991년 3월 ~ 1997년 2월 창신대학 전자통신과 조교수
1997년 3월 ~ 동서대학교 정보통신 공학부 전임강사

※ 주관심 분야 : 암호학, 부호기술, JAVA Technology

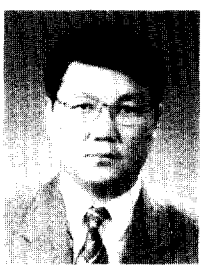
□ 著者紹介



정 원 영

1990년 강원대학교 전자공학과 졸업
1995년 강원대학교 대학원 전자공학(석사)
1995년 ~ 현재 한국통신 가입자망연구소 전임연구원

※ 주관심 분야 : 암호이론, 무선통신보안



김 태 근

1981년 한국항공대학교 전자공학과 졸업
1983년 한국과학기술원 전자공학(석사)
1991년 영국 ESSEX 대학교 전자공학(박사)
1983년 ~ 현재 한국통신 가입자망연구소 책임연구원

※ 주관심 분야 : 무선통신시스템, 무선통신보안

□ 著者紹介

문 상 재

1972년 2월 서울대학교 공과대학 공업교육과(전자공학, 공학사)
1974년 2월 서울대학교 대학원 전자공학과(통신공학, 공학석사)
1984년 6월 미국 UCLA(통신공학, 공학박사)
1984년 6월 ~ 1985년 6월 UCLA Postdoctor 근무
1984년 6월 ~ 1985년 6월 미국 OMNET 컨설턴트
1994년 ~ 현재 경북대학교 공과대학 전기전자공학부 교수

※ 주관심 분야 : 정보보호, 디지털 통신, 정보통신망