論文2002-39TE-2-8

음성암호시스템 설계에 관한 연구

(A Study on the design of voice cryptograph system)

崔 太 燮 *, 安 寅 秀 *, 林 承 河 **, 司 空 石 鎭 *

(Tae-Sup Choi, In-Soo Ahn, Seung-Ha Lim, and Sug-Chin Sakong)

요 약

본 논문에서는 음성 통화에서의 안전한 전송과 수신을 위하여 SEED 알고리즘을 이용한 음성 암호 시스템 설계를 하였다. 음성영역의 신호는 CODEC에 의해 디지털 신호로 변환된다. 그리고 개선된 SEED 알고리즘을 적용한 DSP는 이 신호를 암호화한다. CODEC은 암호화된 신호를 아날로그 음성신호로 변환한다. 이 음성 신호는 중간에 도청이나 감청을 한다고 하더라도 암호화되어있기 때문에 안전하게 전송할 수 있다. 수신자는 수신된 음성신호를 복호화 SEED 알고리즘을 이용하여 송신자의 원음성을 들을 수 있다. 본 논문에서는 16라운드인 SEED 알고리즘의 라운드 수를 32라운드로 설계하여 truncated differential 확률을 $2^{-143.1}$에서 $2^{-286.6}$이상으로 개선하였다.

Abstract

In this paper, we studied the voice cryptograph system designed by the SEED algorithm for the safe transmission and receipt on the voice communication. Voice band signal converts to digital signal by the CODEC and DSP that applied the improved SEED algorithm encrypt the digital signal. The CODEC convert Encryption signal into analog voice signal. This voice signal is transmitted safely because of encryption signal even if someone wiretap. Receiver can hear the source voice, because the encryption signal decrypted using the SEED algorithm. In this paper, We designed the 32 round key instead of 16 round key in the SEED algorithm so that we improve the truncated differential probability from $2^{-143.1}$ to $2^{-286.6}$.

Ⅰ. 서 론

자신의 비밀정보를 보호하는 방법과 상대방의 비밀정보를 가로채어 자신에게 유리한 정보를 얻고자 하는 노력은 인류의 역사이래 매우 중요한 개념으로 인식되어왔다. 이러한 정보의 보호와 탈취는 주로 군대, 외교기관, 또는 특수 목적을 가진 곳에서 이용되어 왔지만 현재에는 개인이나 회사에서도 많은 관심을 가지고 있다. 많은 연구 노력의 결과, 미국에서는 1970년대 중반에 민간분야에서 사용될 암호시스템의 표준을 마련하였다. DES(Data Encryption standard)라고 명명된 이 표준 암호의 제정 이후에 대학, 연구소 그리고 정부를 중심으로 암호에 대한 관심이 높아지면서 암호관련 연구에 박차를 가하게 되었다[1]. 한국에서는 정보보호센터를 중심으로 암호 알고리즘에 대한 연구가 활발하게 진행되고있고, 1999년에는 128비트 블록 암호 알고리즘인 SEED를 한국표준으로 선정하였다[2-4]. 본 논문에서

* 正會員, 國民大學校 電子情報通信工學部
(School of Electrical Engineering, Kookmin University)
** 正會員, 富川大學 電子科
(Dept. of Electronics Engineering, Bucheon College)
接受日字:2002年1月8日, 수정완료일:2002年4月18日

는 현대 사회에서 비중이 매우 큰 통신분야, 특히 PCS와 셀룰러폰을 포함한 유·무선 음성통신의 암호화에 관하여 연구하였다. 그 중에서도 음성 통화에서의 안전한 전송과 수신을 위하여 SEED 알고리즘을 이용한 음성 암호 시스템을 설계하였다.

음성영역의 신호(300~3.6kHz)를 CODEC을 통해 PCM 디지털 신호로 변환되고, 이 신호는 DSP의 수신단자로 들어가게 된다. 프로세서에서는 개선된 SEED 알고리즘을 사용하여 암호화시킨다. 암호화된 디지털 신호는 프로세서의 송신부를 통해 CODEC의 D/A 컨버터부로 연결된다. CODEC을 통해 출력된 음성 신호로 전송하면 중간에 도청이나 감청을 한다고 하더라도 암호화되어있기 때문에 안전하게 전송할 수 있다. 수신자는 음성신호를 가지고 복호화 SEED 알고리즘을 사용하여 원래의 음성 신호를 받을 수 있다. 본 논문에서는 16라운드인 SEED 알고리즘의 라운드 수를 32라운드로 설계한 음성암호 시스템을 구성하고, truncated differential 확률을 $2^{-143.1}$에서 $2^{-286.6}$이상으로 개선하였다[5-8].

Ⅱ. SEED 알고리즘

1. 개요

암호알고리즘은 암·복호화에 사용되는 키의 특성에 따라 암·복호화 키가 같은 대칭키 암호알고리즘과 암·복호화 키가 서로 다른 공개키 암호알고리즘으로 크게 구분할 수 있으며, 대칭키 암호알고리즘은 데이터 처리 형식에 따라 스트림 암호알고리즘과 블록 암호알고리즘으로 나눌 수 있다. SEED는 대칭키 암호알고리즘으로, 블록 단위로 메시지를 처리하는 블록 암호알고리즘이다. 대칭키 블록 암호알고리즘은 비밀성을 제공하는 암호시스템의 중요 요소이다. n비트 블록 암호알고리즘이란 고정된 n비트 평문을 같은 길이의 n비트 암호문으로 바꾸는 함수이다(n비트 : 블록 크기). 이러한 변형 과정에 암·복호키가 작용하여 암호화와 복호화를 수행한다. SEED의 라운드 수는 16라운드이고 라운드 수의 기준은 불법적인 공격에 안전하다고 생각되는 확률이다. 하지만 인터넷과 통신이 발달로 인해 예전에는 안전하다고 하는 라운드가 지금은 안전하다고 장담할 수 없는 추세이다. 그래서 본 논문에서는 SEED 알고리즘의 라운드 수를 32로 하여 더욱더 불법적인 공격에 강하게 하였다. 128비트의 평문 블록단위당 128비트 키로부터 생성된 32개의 64비트 라운드 키를 입력으로 사용하여 총 32라운드를 거쳐 128비트 암호문 블록을 출력한다. 그림 1은 본 논문의 SEED 알고리즘의 전체구조를 도식화한 것이다[9].

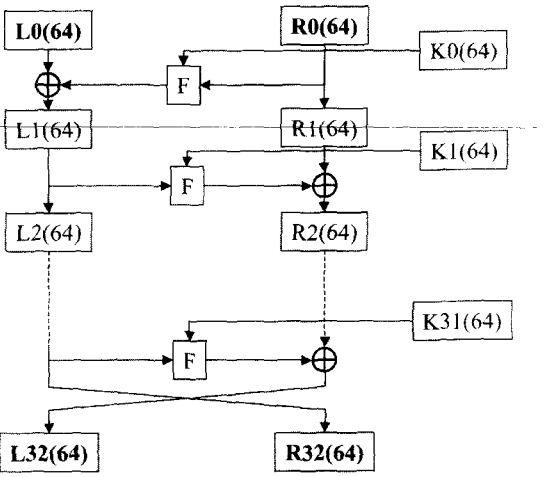

그림 1. 개선된 SEED의 전체 구성도
Fig. 1. Structure diagram of improved SEED algorithm.

128비트 입력 평문블록을 2개의 64비트 블록 $(L_0(64), R_0(64))$으로 나누어, 32개의 64비트 라운드 키를 이용하여 32라운드를 수행한 후, 최종 128비트 암호문 블록 $(L_{16}(64), R_{16}(64))$을 출력한다.

2. F 함수

본 논문에서 사용한 SEED 알고리즘의 전체 구조는 Feistel 구조로 이루어져 있다. Feistel 구조는 각각 t비트인 L_0, R_0 블록으로 이루어진 $2t$비트 평문 블록 (L_0, R_0)이 r 라운드($r \geq 1$)를 거쳐 암호문 (L_r, R_r)으로 변환되는 반복 구조이다. 반복 구조는 평문 블록이 여러 라운드를 거쳐 암호화되는 과정이다. 라운드 함수 $i\ (1 \leq i \leq r)$는 암호키 K로부터 유도된 각 서브키 K_i (또는, 라운드 키라 불림)을 중요 입력으로 하여 $L_i = R_{i-1},\ R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$ 통해 $(L_{i-1}, R_{i-1}) \xrightarrow{K_i} (L_i, R_i)$로 바꾸어 주는 함수이다.

DES, FEAL, LOKI, MISTY, Blowfish, CAST, Twofish 등도 여기에 속한다. 전체 알고리즘의 라운드 수는 요구되는 보안도와 수행 효율성의 상호 절충적 관계에 의해 결정된다. Feistel 구조에서 3라운드 이상이면 짝수 라운드로 구성된다. 이 구조는 라운드 함수

에 관계없이 역변환이 가능하고(즉, 암·복호화 과정이 같음), 두 번의 수행으로 블록간의 완전한 확산(diffusion)이 이루어지며, 알고리즘의 수행속도가 빠르고, H/W 및 S/W로 구현이 용이하고, 그리고 아직 구조상의 문제점이 발견되고 있지 않다는 장점을 지니고 있다.

Feistel 구조를 갖는 블록 암호알고리즘은 F 함수의 특성에 따라 구분될 수 있다. SEED의 F 함수는 수정된 64비트 Feistel 형태로 구성된다. F 함수는 각 32비트 블록 2개(C, D)를 입력으로 받아, 32비트 블록 2개(C′, D′)를 출력한다. 즉, 암호화 과정에서 64비트 블록(C, D)과 64비트 라운드 키 $K_i=(K_{i,0};K_{i,1})$를 F 함수의 입력으로 처리하여 64비트 블록(C′, D′)을 출력한다.

$$C' = G[G[G\{(C\oplus Ki0)\oplus(D\oplus Ki1)\}\boxplus(C\oplus Ki0)]\boxplus G\{(C\oplus Ki0)\oplus(D\oplus Ki1)\}]\boxplus G[G[G\{(C\oplus Ki0)\oplus(D\oplus Ki1)\}\boxplus(C\oplus Ki0)]$$

$$D' = G[G[G\{(C\oplus Ki0)\oplus(D\oplus Ki1)\}\boxplus(C\oplus Ki0)]\boxplus G\{(C\oplus Ki0)\oplus(D\oplus Ki1)\}]$$

그림 2는 F 함수의 구조도이다. 그림에서 i는 라운드수, a ⊕ b 는 XOR 이고, a ⊞ b = (a + b) mod 2^{32}이다.

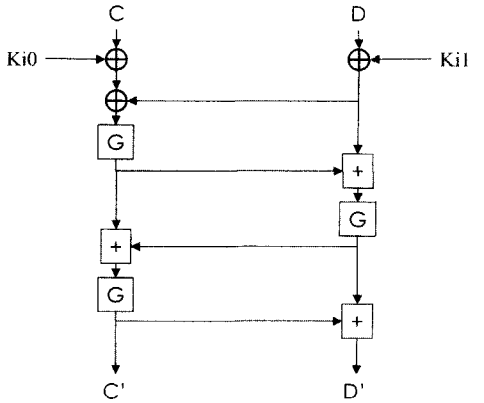

그림 2. F함수 구조도
Fig. 2. Structure diagram of F function.

3. G 함수

G 함수는 매우 좋은 특성을 갖는 두 개의 8비트 S-box를 이용하여 입력의 각 바이트를 비선형 변환 후, 그 결과 32비트를 4비트 왼쪽 회전 이동한 후 출력한다. 전체 G 함수는 다음과 같다.

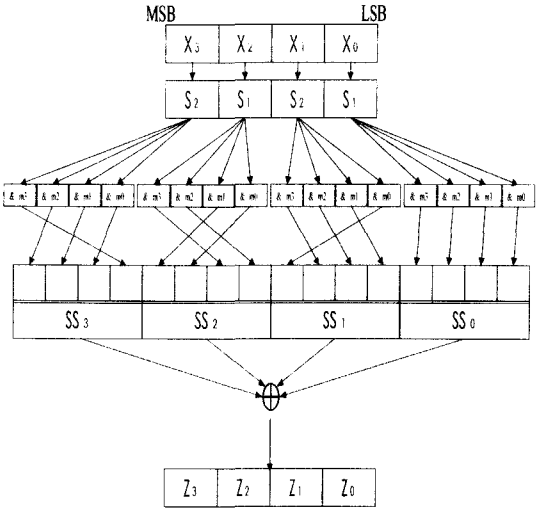

그림 3. G함수의 구조도
Fig. 3. structure diagram of G function.

그림 3은 G함수의 구조도이다.

위의 G 함수는 구현의 효율성을 위해 4개의 확장된 4바이트 SS-box들(4K bytes)의 exclusive-or로 구현할 수 있다. 이를 위해 다음과 같은 4개의 SS-box들을 저장해야 한다.

$$\begin{aligned}
SS_3 &= S_2(X_3)\&m_2 \,\|\, S_2(X_3)\&m_1 \,\|\, S_2(X_3)\&m_0 \,\|\, S_2(X_3)\&m_3,\\
SS_2 &= S_1(X_2)\&m_1 \,\|\, S_1(X_2)\&m_0 \,\|\, S_1(X_2)\&m_3 \,\|\, S_1(X_2)\&m_2,\\
SS_1 &= S_2(X_1)\&m_0 \,\|\, S_2(X_1)\&m_3 \,\|\, S_2(X_1)\&m_2 \,\|\, S_2(X_1)\&m_1,\\
SS_0 &= S_1(X_0)\&m_3 \,\|\, S_1(X_0)\&m_2 \,\|\, S_1(X_0)\&m_1 \,\|\, S_1(X_0)\&m_0,
\end{aligned}$$

(여기서, || 는 concatenation).

이 확장 SS-box들을 이용하면 G 함수는 다음과 같이 구현될 수 있다.

$$Z = SS_3(X_3) \oplus SS_2(X_2) \oplus SS_1(X_1) \oplus SS_0(X_0)$$

$$Y_3 = S_2(X_3),\quad Y_2 = S_1(X_2),\quad Y_1 = S_2(X_1),\quad Y_0 = S_1(X_0),$$

$$\begin{aligned}
Z_3 &= (Y_0\&m_3) \oplus (Y_1\&m_0) \oplus (Y_2\&m_1) \oplus (Y_3\&m_2)\\
Z_2 &= (Y_0\&m_2) \oplus (Y_1\&m_3) \oplus (Y_2\&m_0) \oplus (Y_3\&m_1)\\
Z_1 &= (Y_0\&m_1) \oplus (Y_1\&m_2) \oplus (Y_2\&m_3) \oplus (Y_3\&m_0)\\
Z_0 &= (Y_0\&m_0) \oplus (Y_1\&m_1) \oplus (Y_2\&m_2) \oplus (Y_3\&m_3)
\end{aligned}$$

$$(m_0 = 0\text{xfc},\ m_1 = 0\text{xf3},\ m_2 = 0\text{xcf},\ m_3 = 0\text{x3f})$$

본 논문에서 사용한 알고리즘(SEED)의 S-Box는 8비트 입력(즉, 0~255)을 받아 8비트 출력(즉, 0~255)을 내는 함수이다.

4. 라운드 키

본 논문에서 사용한 SEED의 라운드 키 생성과정은 128비트 암호키를 64비트씩 좌우로 나누어 이들을 교대로 8비트씩 좌/우로 회전 이동한 후, 결과의 4워드들

표 1. 라운드키 생성과정에 사용된 상수
Table 1. Constant generated Round Key.

라 운 드 상 수	
KC_0 = 0x9e3779b9	KC_{16} = 0x79b99c37
KC_1 = 0x3c6ef373	KC_{17} = 0xf3733c6e
KC_2 = 0x78dde6e6	KC_{18} = 0xe6e678dd
KC_3 = 0xf1bbcdcc	KC_{19} = 0xcdccf1bb
KC_4 = 0xe3779b99	KC_{20} = 0x9b99e377
KC_5 = 0xc6ef3733	KC_{21} = 0x3733c6ef
KC_6 = 0x8dde6e67	KC_{22} = 0x6e678dde
KC_7 = 0x1bbcdccf	KC_{23} = 0xdccf1bbc
KC_8 = 0x3779b99e	KC_{24} = 0xb99e3779
KC_9 = 0x6ef3733c	KC_{25} = 0x733c6ef3
KC_{10} = 0xdde6e678	KC_{26} = 0xe678dde6
KC_{11} = 0xbbcdccf1	KC_{27} = 0xccf1bbcd
KC_{12} = 0x779b99e3	KC_{28} = 0x99e3779b
KC_{13} = 0xef3733c6	KC_{29} = 0x33c6ef37
KC_{14} = 0xde6e678d	KC_{30} = 0x678dde6e
KC_{15} = 0xbcdccf1b	KC_{31} = 0xcf1bbcdc

그림 4는 라운드키 생성과정의 구조도이다.

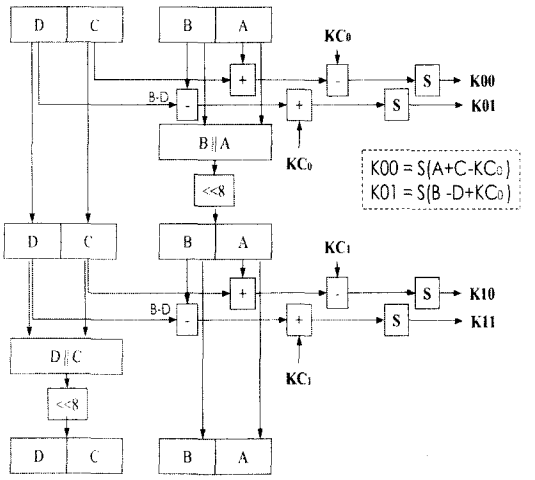

그림 4. 라운드키 생성과정 구조도
Fig. 4. Structure diagram of round key generator.

에 대한 간단한 산술연산과 G 함수를 적용하여 라운드 키를 생성한다. 라운드 키 생성과정은 효율성을 위하여, 암호화나 복호화시 암호키로부터 필요한 라운드 키를 간단히 계산할 수 있도록 설계한다. 각 라운드에 사용되는 라운드 키는 다음과 같은 방식으로 생성한다. 먼저 128비트 입력키를 32비트씩 4개의 조각으로 쪼갠 후 (A, B, C, D), 1라운드 키 $K_{1,0}$ = G(A+C-KC_0), $K_{1,1}$ = G(B-D+KC_0) (단, KC0 : 1 라운드 상수)을 생성한다. 다음에 레지스터 A, B를 우측으로 8비트 회전 이동시킨다. 2라운드 키 $K_{2,0}$ = G(A+C-KC_1) ; $K_{2,1}$ = G(B-D+KC_1) (단, KC_1 : 2 라운드 상수)을 생성한다. 다음에 레지스터 C, D를 좌측으로 8비트 회전 이동시킨다. 3라운드 키 $K_{3,0}$ = G(A+C-KC_2), $K_{3,1}$ = G(B-D+KC_2)(단, KC_2 : 3 라운드 상수)를 생성한다. 4라운드 키는 2라운드 키와 같은 방법으로 키를 생성하고 계속해서 16라운드 키를 생성할 때까지 반복한다.

다음은 SEED에서 생성하는 각 라운드 i에 사용되는 라운드 키 K_i=($K_{i,0}$; $K_{i,1}$) 생성 프로그램이다.

```
for( i=1; i<=32; i++) {
    Ki,0 ← G(A+C-KCi) ;
    Ki,1 ← G(B-D+KCi) ;
    if( i%2==1 ) A|| B ← (A||B)>>8 ;
    else C|| D ← (C||D)<<8 ;
}
```

표1은 라운드키 생성과정에 사용된 상수 값을 나타낸다.

5. truncated differential

G함수가 F함수에 충분한 diffusion을 준다면 안전성 측면에서 좋은 구조라고 할 수 있다. 그래서 G 함수에서 S-box를 통과한 결과들은 3라운드의 변형 Feistel network를 통과하면서 서로 permutation 되지만, S-box의 확률적 특성을 이용하면 각 워드의 상위 바이트의 차이가 다른 바이트에 영향을 주지 않는 truncated differential들이 높은 확률로 발생할 수 있다. F함수 구조 변경을 통해 안전도를 높이는 방법은 G함수에서 S-box 출력들을 적절한 치환(permutation)을 통해 충분히 섞어 주는 것이다.

F 함수의 3라운드 truncated differential을 구하기 위해 다음과 같은 3라운드 iterative truncated differential을 정한다.

$$(A, B, 0, 0) \rightarrow (C, D, 0, 0)$$

여기서 A, B, C, D는 모두 상위 두 비트만이 nonzero

인 32비트 워드이다. 각각의 가능한 A, B에 대해 3라운드 후 상위 두 비트만 nonzero인 모든 가능한 C, D로 가는 characteristic들에 대한 확률들을 구한다. 그 결과 3라운드에서는 $P_{3R} = \sum_{i=0}^{3} P_{3i} = 2^{-28.4}$라는 결과를 얻는다. 16라운드에서의 확률은 $2^{-143.1}$이고 본 논문에서는 32라운드를 적용한 SEED를 사용하여 $2^{-286.6}$로 증가시켜 더욱 더 알고리즘 공격을 어렵게 하였다.

Ⅲ. CODEC과 DSP

1. CODEC의 개요

CODEC이란 음성이나 비디오 등의 아날로그 신호를 펄스 부호 변조(PCM)를 사용하여 전송에 적합한 디지털 비트 스트림으로 변환하고, 역으로 수신 측에서 디지털 신호를 아날로그 신호로 변환하는 기기 또는 장치를 말한다. 본 논문에서는 TI사의 TLV320AC36CN을 사용한다[10].

1.1 VBAP 궤환 회로

그림 5는 본 논문에서 사용한 VBAP(Voice Band Audio Processor) CODEC의 동작을 시험하기 위한 회로도이다.

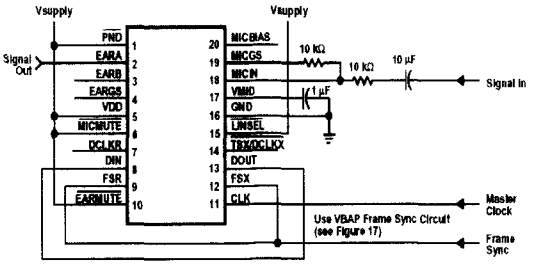

그림 5. VBAP PCM 루프백 실험 블록도
Fig. 5. VBAP PCM Loopback Test Configuration Diagram.

그림 5에서 마스터 클럭과 프레임 동기 신호(Frame Sync Circuit)는 디지털 시스템 프로세서에서 생성되는 클럭을 사용한다. 루프백 실험에서 VBAP는 송신부와 수신부는 동기되어 사용된다. MICIN에 입력되는 아날로그 신호는 직렬 데이터로 부호화된다. 루프백 실험을 위해 DOUT을 DIN으로 연결하고, VBAP에 의해서 디지털에서 아날로그로 다시 부호화된다. 그림 6은 입력 2㎑ 아날로그 신호와 출력 신호를 비교한 것이다.

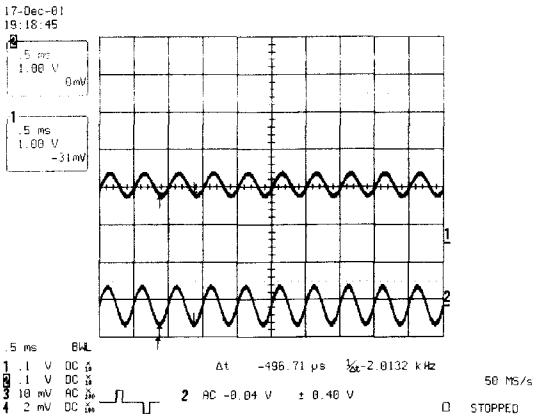

그림 6. 입력(아래)과 출력 신호 파형(위) (2㎑)
Fig. 6. Input signal (down) and output signal wave(up) (2㎑).

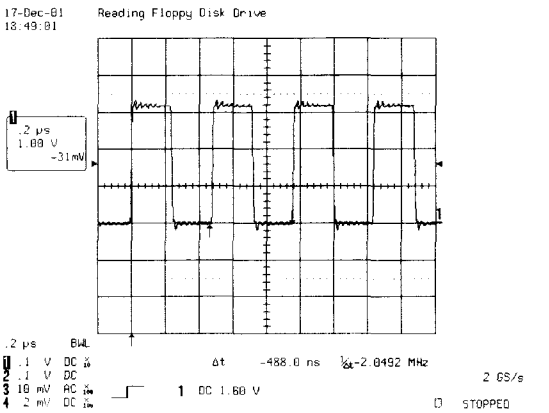

그림 7. 마스터 클럭 신호(2.05 ㎒)
Fig. 7. Master clock signal(2.05 ㎒).

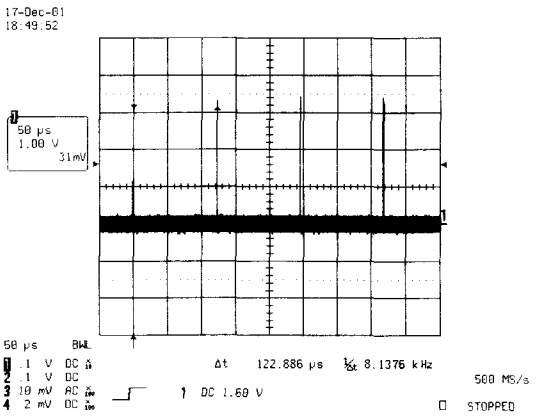

그림 8. 프레임 동기 신호(8.14 ㎑)
Fig. 8. Frame Sync signal(8.14 ㎑).

마스터 클럭 2.05 ㎒과 프레임 동기 신호는 8.14 ㎑

은 DSP에서 설계한 클럭을 사용한다. 그림 7은 DSP에서 출력되는 마스터클럭 파형이고 그림 8은 프레임 동기 신호이다.

그림9는 고정 데이터 모드에서의 수신 타이밍 도를 나타낸다. 마스터 클럭이 하이가 되고, 수신 프레임 동기 신호의 한 클럭이 하이가 되면, 다음 폴링 에지에서부터 8비트 데이터를 CODEC의 DIN에서 수신하게 된다.

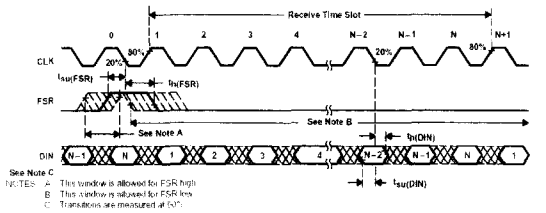

그림 9. 고정 데이터 모드에서의 수신 타이밍도
Fig. 9. Receive Side Timing Diagram in Fixed-Data Rate Mode.

그림 10은 본 논문의 음성암호시스템에서의 프레임 동기 신호와 수신 데이터의 파형이다. 프레임 동기 신호가 하이가 되고 나서 8비트가 수신되는 것을 알 수 있다.

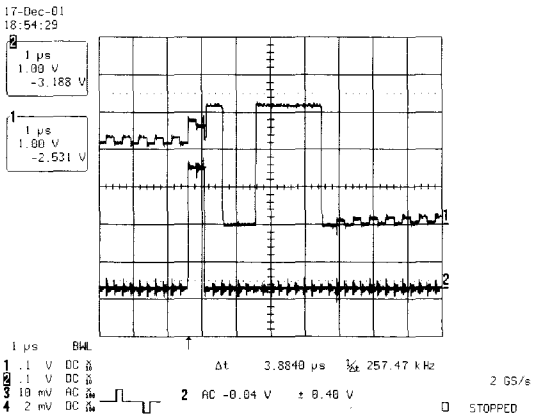

그림 10. 수신된 디지털 신호(위)와 프레임 동기신호 (아래)
Fig. 10. Receive digital signal(up) and Frame Sync signal(down).

그림 11은 고정 데이터 모드에서의 송신타이밍도를 나타낸다. 마스터 클럭이 하이가 되는 시점에서 송신 프레임 동기 신호가 한 클럭 하이가 되면, 다음 폴링 에지에서 CODEC의 DOUT에서 8비트의 PCM 신호를

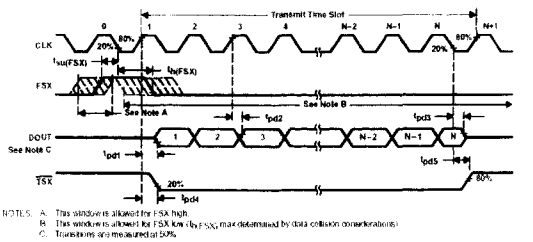

그림 11. 고정 데이터 모드에서의 송신 타이밍도
Fig. 11. Transmit Side Timing Diagram in Fixed-Data Rate Mode.

송신한다.

그림 12는 본 논문의 음성암호시스템에서의 프레임 동기 신호와 송신 데이터의 파형이다. 프레임 동기 신호가 하이가 되고 나서 8비트가 송신되는 것을 알 수 있다.

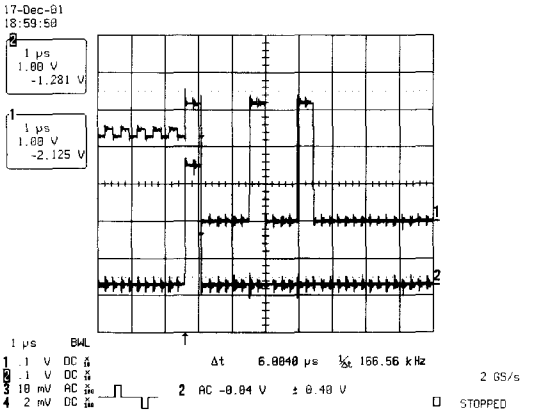

그림 12. 송신된 디지털 신호(위)와 프레임 동기신호 (아래)
Fig. 12. Transmit digital signal(up) and Frame Sync signal(down).

2. Digital Signal Processor(DSP)

본 논문에서는 CODEC과의 인터페이스를 가능하게 하는 전이중다채널 버퍼(multichannel buffered serial ports : McBSP)에 직렬 포트가 포함되어 있다. McBSP는 분리된 전송 수신 채널로 구성되어 있고 독립적으로 작동한다[11]. 송신기상에서, 송신 프레임 동기화와 클로킹은 BFSX 와 BCLKX 핀에 각각 지정되어 있다. DMA는 DXR(data transmit register)에 쓴 것에 대한 데이터의 전송을 초기화한다. DXR에 쓰여진 데이터는 XSR(transmit shift register) 를 통해 BDX 핀상으로 출력된다. 현재 워드의 전송이 처리되는 동안 다음 워

드를 로드시킨다. 수신단 상에서, 수신 프레임 동기화와 클로킹은 BFSR 와 BCLKR 핀에 각각 지정되어 있다. DMA의 DRR(data receive register)로부터 수신된 데이터를 읽고, BDR 핀에 전송된 데이터는 RSR(receive shift register)로 이동된후, RBR(receive buffer register)에 저장된다. CPU와 DMA는 McBSP로 부터 데이터를 송수신하고, McBSP 인터럽트, 이벤트 신호, 상태 플랙에 기초한 전송을 동기화 한다. DMA는 McBSP와 메모리 사이에서 CPU로 부터의 간섭없이 데이터 이동을 다룰 수 있다. 표준 직렬 포트 기능 외에 McBSP 는 프로그램 클럭과 프레임 동기화 신호를 생성한다. 프로그램 기능으로 on-chip 압축 하드웨어는 μ-law와 A-law에서 데이터의 압축과 신장을 가능하게 한다. 압축이 사용되고 있을 때, 전송 데이터는 정해진 압축규칙에 따라 부호화되고, 수신된 데이터는 2's 컴플리먼트 포맷으로 복호화된다.

Ⅳ. 시스템 구성과 실험 결과

1. 시스템 구성

다음 그림13은 음성 암호 시스템의 전체 구성도이다.

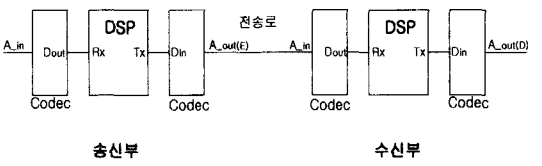

그림 13. 시스템 구성도
Fig. 13. System structure diagram.

송신자의 음성 아날로그 신호는 CODEC의 A_IN핀으로 들어가서 PCM 디지털 신호로 변환하여 D_{OUT}핀으로 전송한다. D_{OUT}핀을 DSP의 RX핀과 연결되어 메모리로 저장된다. DSP는 이 신호를 개선된 SEED 알고리즘을 사용하여 암호화하여 TX핀으로 내보낸다. TX핀은 CODEC의 D_{IN}핀과 연결되어 다시 음성대역의 신호로 변환된다. 이 변환된 신호는 암호화되어 있어 잡음처럼 들리게된다. 암호화된 신호는 전송로를 통해 전송되며, 중간에 타인이 불법적으로 데이터를 알려고 하여도 알 수 없게된다. 수신자는 CODEC의 A_IN으로 암호화된 음성 아날로그 신호를 받아서 DSP를 통해 복호화 하여 원래의 송신자의 음성을 수신하게된다.

2. DSP 메모리

다음 그림14는 DSP 메모리에서의 데이터 처리의 과정이다.

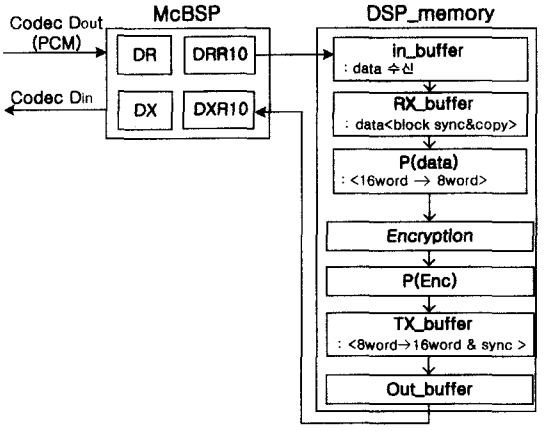

그림 14. DSP 메모리의 처리과정 블록도
Fig. 14. DSP memory processing block diagram.

CODEC에서 들어온 신호는 McBSP의 DR에서 수신하고 하위레지스터인 DRR10에 저장되어 8.14khz마다 들어온 8비트의 신호를 DSP의 메모리로 전송한다. DSP의 in_buffer에서는 일단 데이터를 저장하고, RX_buffer로 전송한다. RX_buffer에서는 16워드마다 저장된 데이터를 P메모리 영역으로 전송한다. P 영역에서는 16워드의 데이터를 8워드(128비트)로 변환한다. DSP의 CPU는 이 데이터를 암호하여 다시 P 영역으로 전송한다. 전송된 데이터는 TX_buffer에서 다시 16워드

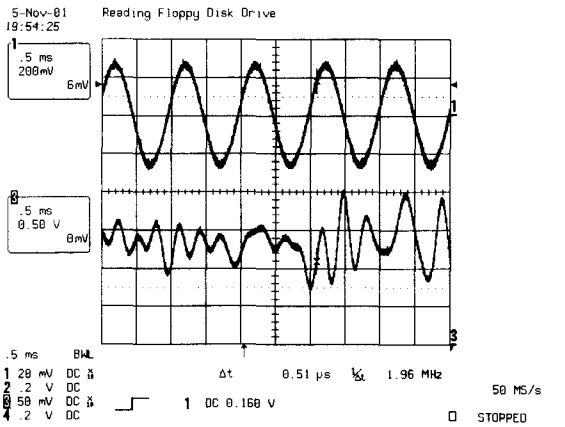

그림 15. 1kHz 입력 사인파(위)와 암호화된 파형(아래)
Fig. 15. 1kHz input sine wave(up) and encrypted wave(down).

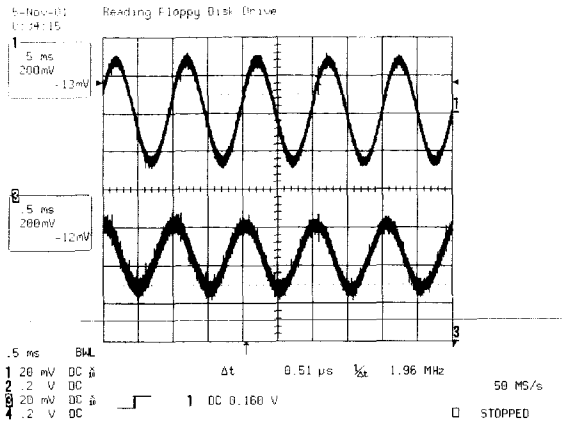

그림 16. 1㎑ 입력 사인파와 복호화된 신호 파형
Fig. 16. 1㎑ input sine wave(up) and decrypted signal wave(down).

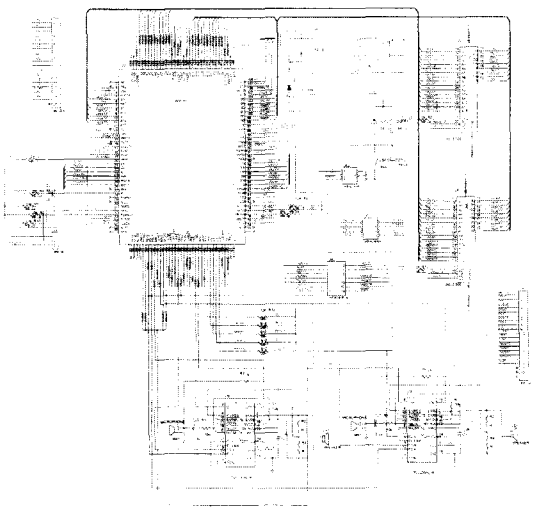

그림 17. 시스템 회로도
Fig. 17. System circuit.

로 변화되고 Out_buffer로 보내진다. 전송을 위해서 이 신호는 McBSP의 DXR10으로 들어가고, DX를 통해 CODEC의 D_{IN}핀으로 전송된다. 이 신호는 암호화되어 안전하게 전송로를 통해 수신자에게 전달된다. 수신자는 이 신호를 다시 복호화 하여 원신호 음성으로 듣는다.

다음 그림15는 1㎑ 사인파를 입력으로 하고 DSP를 통해 암호화된 파형을 보여준다. SEED 알고리즘을 통해 암호화되어 있기 때문에 파형이 찌그러진 것처럼 보인다.

그림16은 입력 1㎑ 사인파와 수신기의 DSP를 통해 복호화된 파형을 보여준다. 그림에서 보듯이 입력의 신호와 같은 파형임을 알 수 있다.

그림 17은 전체 음성 암호화 시스템의 회로도이다.

V. 결 론

본 논문에서는 개선된 SEED 알고리즘을 사용하여 음성영역에서의 안전한 통화를 할 수 있는 시스템을 설계하였다. 음성 시스템에서 0.3~3.4㎑의 아날로그 음성 신호를 코덱을 사용하여 디지털 신호로 변환하고, 이 데이터를 DSP 메모리에서 개선된 SEED 알고리즘을 사용하여 암호화시킨다. 암호화된 아날로그 음성 신호로 전송하기 때문에 중간에 도청이나 감청을 한다고 하더라도 암호화되어있서 알아들을 수 없는 이상한 음성으로 들리게 된다. 수신자는 수신된 신호를 반대 과정을 거쳐 복호화 SEED 알고리즘을 사용하여 원송신자의 음성 신호를 듣게된다. 또한 본 시스템에서는 SEED의 16라운드에서 더 나아가 32라운드로 라운드수를 늘림으로써 truncated differential확률을 $2^{-143.1}$에서 $2^{-286.6}$이상으로 개선하여 더욱 더 불법공격이 어렵게 하였다.

참 고 문 헌

[1] 박창섭, "암호 이론과 보안", 대영사, pp.41-57, 1999

[2] "128비트 블록 암호 표준 SEED", 한국정보보호센터, pp.14, 1998

[3] "128비트 블록 암호알로리즘(SEED) 개발 및 분석 보고서", 한국정보보호센터, pp.1-21, 1998

[4] 송문빈, 고명관, 정연모, "SEED 암호화 알고리즘의 하드웨어 구현", 한국정보처리학회, 2000년도 추계학술발표논문집 제7권 제2호, pp. 1453-1456, 2000

[5] Eli Biham and Adi Shamir, "Differential cryptanalysis of DES-like cryptosystems", Journal of cryptology, vol.4, no.1, pp.3-72, 1991.

[6] Eli Biham and Adi Shamir, "Differential cryptanalysis of the full 16 round DES", In Ernest F. Brickell, editor, Advance in Cryptology-Ccrypto'92, vol.740 of Leture Notes in Computer Science, pp.487-496, Springer-Verlag, Berlin, 1993.

[7] Mitsuri Matsui, "Linear cryptanalysis method for DES cipher", In Tor Helleseth, editor, Advances in Crytology-Eurocrypt'93, vol.765 of Lecture Notes in Computer Science, pp.386-397, Springer-Verlag, Berlin, 1994.

[8] Mitsuri Matsui, "New structure of block ciphers with provable security against defferential and linear cryptanalysis", In Deiter Gollman, editor, Fast Software Encryption, Third International Workshop, vol.1039 of Lecture Notes in Computer Science, pp.205-218, Springer-Verlag, Berlin, 1996.

[9] 이민섭, "현대암호학", 교우사, pp.145-152, 1999

[10] Thomsa P.Barnwell III, "Speech Coding A Computer Laboratory Textbook", John Willy & Sons Inc., 1996.

[11] Texas Instruments Inc., TMS320C54x Evaluation Module, 1995.

저 자 소 개

崔 太 燮(正會員) 第37卷 SD編 第6號 參照

安 寅 秀(正會員) 第37卷 SD編 第6號 參照

林 承 河(正會員) 第36卷 T編 第3號 參照
현재 : 부천대학 전자과 교수

司空石鎭(正會員) 第37卷 SD編 第6號 參照
현재 : 국민대학교 전자정보통신공학부 교수