情報保護學會論文誌
第18卷 第4號, 2008. 8

축소된 20-라운드 SMS4에 대한 차분 공격*

김 태 현[1†], 김 종 성[1‡], 성 재 철[2], 홍 석 희[1]
[1]고려대학교 정보경영공학전문대학원, [2]서울시립대 수학과

Differential Cryptanalysis of a 20-Round Reduced SMS4 Block Cipher*

Taehyun Kim[1†], Jongsung Kim[1‡], Jaechul Sung[2], Seokhie Hong[1]
[1]Center for Information Security Technologies (CIST), Korea University,
[2]Department of Mathematics, University of Seoul

요 약

128-비트 블록 암호 SMS4는 중국 무선 네트워크 표준인 WAPI에 사용되고 있는 암호 알고리즘으로써 128-비트 비밀키를 사용한다. 본 논문에서는 전체 32 라운드 중 16-라운드 차분 특성을 이용한 20-라운드 SMS4에 대한 차분 공격을 소개한다. 본 논문에서 소개하는 차분 공격은 2^{126}개의 선택 평문과 $2^{105.85}$의 20-라운드 SMS4 암호화 연산을 요구하며, 기 제안된 SMS4에 대한 분석 결과 중 가장 많은 라운드에 대한 분석 결과이다.

ABSTRACT

The 128-bit block cipher SMS4 which is used in WAPI, the Chinese WALN national standard, uses a 128-bit user key with the number of 32 rounds. In this paper, we present a differential attack on the 20-round SMS4 using 16-round differential characteristic. This attack requires 2^{126} chosen plaintexts with $2^{105.85}$ 20-round SMS4 decryptions. This result is better than any previously known cryptanalytic results on SMS4 in terms of the numbers of attacked rounds.

Keywords : Block Cipher, SMS4, Differential Cryptanalysis

Ⅰ. 서 론

무선 네트워크에 대한 보안 메커니즘으로 설계된 WAPI(WLAN Authentication and Privacy Infrast ructure)는 IEEE 802.11i[4]의 대체 목적으로 중국 표준 협회 SAC에 의해 국제 표준 기구 ISO에 제출되었다. WAPI와 IEEE 802.11i는 모두 ISO/IEC 8802-11 WLAN에 대한 보안 메커니즘 표준으로 제안되었으며 두 스킴은 데이터를 암호화하기 위해 IEEE 802.11i가 AES[5]를 사용하는 반면 WAPI는 블록 암호 SMS4[9]를 사용한다. AES와는 다르게 SMS4는 당시 SMS4의 정확한 알고리즘 구조가 공개되지 않았기 때문에 체계적이고 논리적인 분석에 의한 안전성을 보장받지 못하였다. 부분적으로 이러한 이유가 반영되어 2006년 3월에 IEEE 802.11i가 국제 표준으로 채택되었다. 그렇지만 현재 WAPI는 중국 무선 네트워크의 표준으로서 사용되고 있으며 SONY와 같은 많은 국제적 기업에서 자사의 생산된제품들에 WAPI를 지원하고 있다.

SMS4는 2006년 1월에 정확한 알고리즘이 공개되었다. SMS4는 128-비트 비밀키를 가지는 128-비트 크기

접수일 : 2008년 1월 15일; 채택일 : 2008년 5월 6일
* 본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음 (IITA-2008-(C1090-0801-0025))
† 주저자, kimth714@cist.korea.ac.kr
‡ 교신저자, joshep@cist.korea.ac.kr

〔표 1〕 SMS4 및 AES의 분석 결과 비교

블록 암호	공격 유형	라운드	데이터 복잡도	메모리 복잡도	시간 복잡도
AES-128 (10 라운드)	불능 차분 공격 [6]	6	$2^{91.5}$	2^{63}	2^{122}
	부메랑 공격 [2]	6	2^{71}	2^{33}	2^{71}
AES-192 (12 라운드)	불능 차분 공격 [10]	7	2^{92}	2^{153}	2^{186}
	포화 공격 [3]	8	2^{128}-2	2^{64}	2^{18}
AES-256 (14 라운드)	불능 차분 공격 [10]	7	$2^{92.5}$	2^{153}	$2^{250.5}$
	포화 공격 [3]	8	2^{128}-2	2^{104}	2^{204}
SMS4 (32 라운드)	포화 공격 [7]	13	2^{16}	2^{20}	2^{114}
	렉탱글 공격 [8]	14	$2^{121.82}$	$2^{125.82}$	$2^{116.66}$
	불능 차분 공격 [8]	16	2^{105}	2^{109}	2^{107}
	차분 공격 [본 논문]	20	2^{126}	2^{73}	$2^{105.85}$

의 블록 암호이다. 현재까지 SMS4 블록 암호에 대한 가장 효과적인 암호 분석 결과는 전체 32-라운드 중 16-라운드 SMS4에 대한 불능 차분 공격이다[8]. 본 논문에서는 20-라운드 SMS4에 대한 차분 공격을 소개한다. Biham등에 의해 제안된 차분 공격[1]은 선형 공격과 함께 현재까지 블록 암호에 대한 가장 강력한 분석 방법 중 하나이다. 차분 공격은 평문과 암호문 사이에 확률 p를 가지는 차분 특성을 이용한다. 공격자는 차분 특성의 입력 차분을 만족하는 선택 평문쌍에 대해 비밀키 K로 암호화된 암호문 쌍을 얻고 처음 또는 마지막 몇 라운드 키를 추측하여 평문쌍 또는 암호문쌍이 원래의 차분 특성으로부터 유도된 출력 차분과 일치하는지 확인한다. 본 논문에서 소개하는 차분 공격은 16-라운드 차분 특성을 이용하여 처음 1-라운드, 마지막 3-라운드 키 복구공격(4R-Attack)을 수행한 후 최종적으로 128-비트 비밀키를 복구한다. 본 논문의 분석과 기존 SMS4에 대한 분석 결과 및 세계 표준 블록 암호 AES의 기존 분석 결과들에 대한 비교는 [표 1]과 같다.

본 논문의 구성은 다음과 같다. 2절에서는 축소된 SMS4에 대한 차분 공격을 소개하기에 앞서 본 논문에서 사용하는 표기법을 정리하고 SMS4에 대한 간략한 알고리즘을 설명할 것이다. 3절에서는 SMS4에 대한 16-라운드 차분 특성을 구성하고 4절에서, 16-라운드 차분 특성을 이용한 20-라운드 SMS4의 차분 공격을 소개한다. 마지막으로 5절은 본 논문의 결론으로 구성된다.

Ⅱ. 표기법 및 SMS4

본 절에서는 본 논문에 전반적으로 사용되는 표기법을 정리하고, SMS4 블록 암호를 간략히 소개한다.

2.1 표기법

본 논문의 표기법은 다음과 같다. 단 워드의 비트 위치는 가장 오른쪽 최하위 비트부터 0으로 시작하며, 왼쪽으로 갈수록 커진다.

- $\oplus$: 비트별 배타적 논리합.
- $\lll i$: i 비트만큼 왼쪽으로 순환이동.
- ? : 알 수 없는 값.
- $[X]_{[j]}$ $(j=0,1,2,3)$: 32-비트 워드 X의 j번째 최하위 바이트.

2.2 SMS4 블록 암호 소개

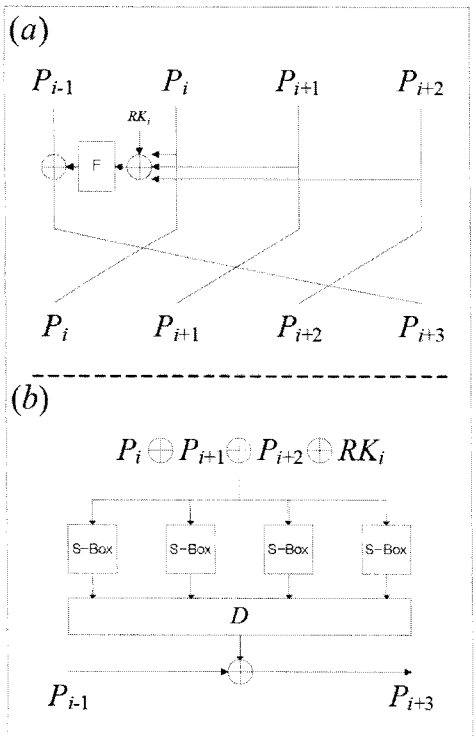
SMS4는 전체 32-라운드 불균형 Feistal network 구조이다. 블록 크기와 비밀키 크기는 128-비트로써 각각 평문과 키 128-비트를 입력받아 128-비트의 암호문을 출력한다. 평문을 4개의 32-비트 워드 $P=(P_0,P_1,P_2,P_3)$로 표기하고 $i(=1,2,...,32)$번째 라운드 암호화후의 중간 값을 X^i로 표기한다. SMS4의 전체 32-라운드 암호화 과정은 다음과 같다.

1. 128-비트 평문 $P=(P_0,P_1,P_2,P_3)$을 입력한다.
2. $i(=1,...,32)$번째 라운드 암호화 과정은 다음과 같다.
   - $P_{i+3}=P_{i-1}\oplus D(S(P_i\oplus P_{i+1}\oplus P_{i+2}\oplus RK_i))$,
   - $X^i=(P_i,P_{i+1},P_{i+2},P_{i+3})$,
3. 암호문 $X^{32}=(P_{32},P_{33},P_{34},P_{35})$을 출력한다.

여기서 RK_i는 i 번째 라운드에 대한 32-비트 라운드 키이다. SMS4의 라운드 함수에는 두 가지 함수가 사용된다. 확산의 성질을 주는 선형 변환 함수 $D(x)=x\oplus$

〔표 2〕 SMS4의 S 박스 테이블 ($s(0x01)=0x90$)

	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf
0x0	*d6*	90	*e9*	*fe*	*cc*	*e1*	3*d*	*b7*	16	*b6*	14	*c2*	28	*fb*	2*c*	05
0x1	2*b*	67	9*a*	76	2*a*	*be*	04	*c3*	*aa*	44	13	26	49	86	06	99
0x2	9*c*	42	50	*f4*	91	*ef*	98	7*a*	33	54	0*b*	43	*ed*	*cf*	*ac*	62
0x3	*e4*	*b3*	1*c*	*a9*	*c9*	08	*e8*	95	80	*df*	94	*fa*	75	8*f*	3*f*	*a6*
0x4	47	07	*a7*	*fc*	*f3*	73	17	*ba*	83	59	3*c*	19	*e6*	85	4*f*	*a8*
0x5	68	6*b*	81	*b2*	71	64	*da*	8*b*	*f8*	*eb*	0*f*	4*b*	70	56	9*d*	35
0x6	1*e*	24	0*e*	5*e*	63	58	*d1*	*a2*	25	22	7*c*	3*b*	01	21	78	87
0x7	*d4*	00	46	57	9*f*	*d3*	27	52	4*c*	36	02	*e7*	*a0*	*c4*	*c8*	9*e*
0x8	*ea*	*bf*	8*a*	*d2*	40	*c7*	38	*b5*	*a3*	*f7*	*f2*	*ce*	*f9*	61	15	*a1*
0x9	*e0*	*ae*	5*d*	*a4*	9*b*	34	1*a*	55	*ad*	93	32	30	*f5*	8*c*	*b1*	*e3*
0xa	1*d*	*f6*	*e2*	2*e*	82	66	*ca*	60	*c0*	29	23	*ab*	0*d*	53	4*e*	6*f*
0xb	*d5*	*db*	37	45	*de*	*fd*	8*e*	2*f*	03	*ff*	6*a*	72	6*d*	6*c*	5*b*	51
0xc	8*d*	1*b*	*af*	92	*bb*	*dd*	*bc*	7*f*	11	*d9*	5*c*	41	1*f*	10	5*a*	*d8*
0xd	8*a*	*c1*	31	88	*a5*	*cd*	7*b*	*bd*	2*d*	74	*d0*	12	*b8*	*e5*	*b4*	*b0*
0xe	89	69	97	4*a*	0*c*	96	77	7*e*	65	*b9*	*f1*	09	*c5*	6*e*	*c6*	84
0xf	18	*f0*	7*d*	*ec*	3*a*	*dc*	4*d*	20	79	*ee*	5*f*	3*e*	*d7*	*cb*	39	48



〔그림 1〕 (a) 라운드 함수, (b) F 함수

$(x \lll 2) \oplus (x \lll 10) \oplus (x \lll 18) \oplus (x \lll 24)$와 S 박스를 사용하는 비선형 함수 $S(x)$가 사용된다. SMS4의 S 박스는 동일한 네 개의 8×8 S 박스를 사용한다. 초기에 SMS4에 사용되는 S 박스의 설계원리가 공개되지 않았지만 Fen등에 의해 AES의 S 박스를 기반으로 설계됨이 밝혀졌다[7]. 네 개의 입력 바이트에 대해 함수 $S(x)$는 다음과 같이 정의된다([표 2] 참조 [7]).

입력: $X=(x_0,x_1,x_2,x_3)\in(Z_2^8)^4$,

출력: $S(X)=(s(x_0),s(x_1),s(x_2),s(x_3))$,

여기서 $s(x)$는 SMS4의 8×8 S 박스이다. 본 논문에서는 합성 함수 $D \circ S$를 비선형 함수 F로 표기한다. SMS4의 복호화 과정은 사용되는 라운드 키가 암호화 과정과 반대로 사용되는 것을 제외하고 암호화 과정과 동일하다. [그림 1]은 SMS4의 라운드 함수와 라운드 함수의 F 함수를 나타낸다.

SMS4의 키스케줄 함수는 128-비트 비밀키 MK를 이용한다. 입력받은 비밀키 MK를 네 개의 32-비트 워드 (MK_0,MK_1,MK_2,MK_3)로 표기하고 j 번째 라운드 키를 RK_j로 표기한다($j=1,2,...,32$). SMS4의 키스케줄 함수는 SMS4의 라운드 함수와 거의 동일하다. 차이점은 라운드 함수의 선형 함수 대신에 다음의 선형 함수 $D'(x)=x\oplus(x\lll 13)\oplus(x\lll 23)$이 사용된다. 비밀키는 system parameter라 불리는 고정된 상수와 XOR 연산되어 SMS4의 키스케줄 함수의 입력값으로 사용된다.

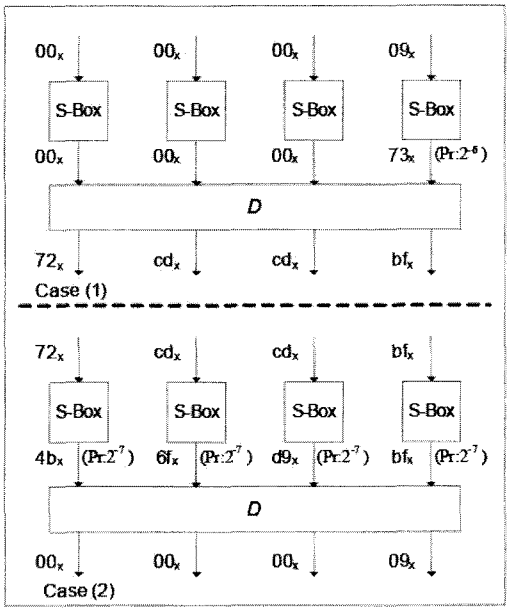
SMS4의 라운드 키가 생성되는 과정은 다음과 같다.

1. $(K_0, K_1, K_2, K_3) =$
$(MK_0 \oplus C_0, MK_1 \oplus C_1, MK_2 \oplus C_2, MK_3 \oplus C_3)$,
여기서 $C_0 = 0xa3b1bac6$, $C_1 = 0x56aa3350$,
$C_2 = 0x677d9197$, $C_3 = 0xb27022dc$와 같다.

2. j 번째 라운드 키 $RK_j = K_{j+3} = K_{j-1} \oplus D'(S(K_j, K_{j+1}, K_{j+2} \oplus CK_j))$를 출력한다. 여기서 CK_j는 라운드 상수로서 $CK_j = ((28 \cdot j), (28 \cdot j+7), (28 \cdot j+14), (28 \cdot j+21))$로 계산된다. 라운드 상수는 네 개의 바이트로 구성되어 있으며 각각 의 연산은 Z_{256}에서 수행된다.

Ⅲ. SMS4의 16-라운드 차분 특성 구성

본 절에서는 SMS4의 16-라운드 차분 특성을 소개한다. SMS4의 경우, 비선형 함수인 S 박스만이 혼돈의 성질을 주는 과정으로 이 과정을 제외하면 모두 확산의 성질을 주는 과정이다. 따라서 XOR 차분 관점으로 살펴본다면 S 박스를 제외한 모든 과정에 대하여 입력 차분에 대한 출력 차분의 모양을 확률 1로 알 수 있다. SMS4의 S 박스는 임의의 0이 아닌 입력 차분에 대하여 127개의 출력 차분이 존재한다. 127개의 출력 차분 중 하나의 값은 확률 2^{-6}을 가지며 나머지 126개의 출력 차분의 확률값은 2^{-7}로 나타난다. 이 사실은 간단한 컴퓨터 프로그램을 통해 쉽게 증명할 수 있다. 본 차분 특성에서는 F 함수에 대한 두 개의 차분 특성을 이용한다. F 함수에 대한 첫 번째 차분 특성은 입력 차분 $\alpha = 00000009_x$와 출력 차분 $\beta = 72cdcdbf_x$이다. 이 경우, F 함수에 능동 S 박스는 한 개로써 본 차분 특성을 만족할 확률은 2^{-6}이다. 두 번째 차분 특성은 입력 차분 $\beta = 72cdcdbf_x$와 출력 차분 $\alpha = 00000009_x$이다. 이때의 능동 S 박스는 네 개로써 확률은 $2^{-28}(=(2^{-7})^4)$로 나타난다([그림 2] 참조). F 함수에 대한 위의 세 가지 차분 특성에는 세 개의 차분값 α, β, $\alpha \oplus \beta$이 존재한다. 따라서 평문을 구성하는 네 개의 각 32-비트 워드에 대하여 앞의 세 개의 차분값과 0을 포함한 네 개의 차분값으로 평문쌍을 만들 수 있다. 간단한 컴퓨터 프로그램을 이용하여 모든 가능한 $256(=4^4)$가지의 평문의 입력 차분에 대한 출력 차분과 확률을 조사했다. 본 논문에서 소개하는 차분 공격은 그 중 확률 2^{-128}을 넘지 않으면서 가장 많은 라운드를 나타내는 차분 특성 중 한 개의 차분 특성을 이용한다. 공격에 사용되는 차분 특성의 정확한 차분값들과 확률은 [표 3]과 같다.



〔그림 2〕 F 함수 차분 특성

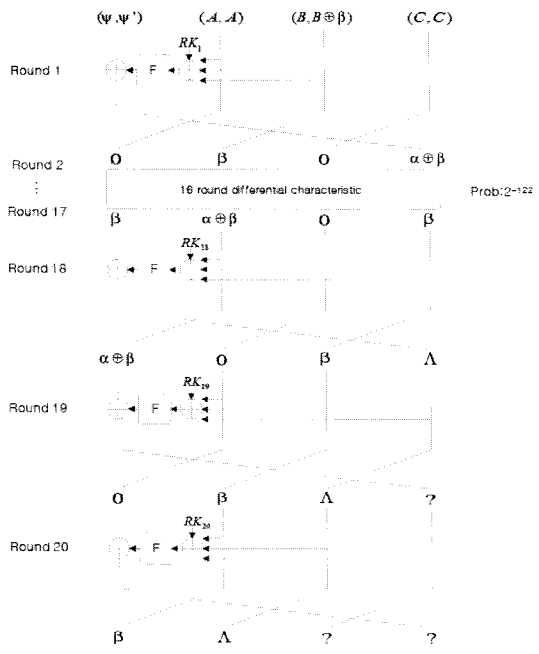
〔표 3〕 SMS4에 대한 16 라운드 차분 특성

라운드	차분값	누적 확률	case
i	$(0, \beta, 0, \alpha \oplus \beta)$	1	-
$i+1$	$(\beta, 0, \alpha \oplus \beta, \beta)$	2^{-6}	(1)
$i+2$	$(0, \alpha \oplus \beta, \beta, 0)$	2^{-12}	(1)
$i+3$	$(\alpha \oplus \beta, \beta, 0, \beta)$	2^{-18}	(1)
$i+4$	$(\beta, 0, \beta, \alpha \oplus \beta)$	2^{-18}	(1)
$i+5$	$(0, \beta, \alpha \oplus \beta, 0)$	2^{-24}	(1)
$i+6$	$(\beta, \alpha \oplus \beta, 0, \beta)$	2^{-30}	(1)
$i+7$	$(\alpha \oplus \beta, 0, \beta, 0)$	2^{-36}	(1)
$i+8$	$(0, \beta, 0, \beta)$	2^{-64}	(2)
$i+9$	$(\beta, 0, \beta, 0)$	2^{-64}	-
$i+10$	$(0, \beta, 0, \alpha \oplus \beta)$	2^{-92}	(2)
$i+11$	$(\beta, 0, \alpha \oplus \beta, \beta)$	2^{-98}	(1)
$i+12$	$(0, \alpha \oplus \beta, \beta, 0)$	2^{-104}	(1)
$i+13$	$(\alpha \oplus \beta, \beta, 0, \beta)$	2^{-110}	(1)
$i+14$	$(\beta, 0, \beta, \alpha \oplus \beta)$	2^{-110}	-
$i+15$	$(0, \beta, \alpha \oplus \beta, 0)$	2^{-116}	(1)
$i+6$	$(\beta, \alpha \oplus \beta, 0, \beta)$	2^{-122}	(1)

Ⅳ. SMS4에 대한 차분 공격

본 절에서는 20-라운드 SMS4의 차분 공격을 소개한다. 앞 절에서 소개한 확률 2^{-122}의 16-라운드 차분 특성을 2 라운드에서 17 라운드까지 적용한다. 본 차분 공격은 먼저 확률 1로 1-라운드 차분 특성을 16-라운드 차분 특성에 구조체를 이용하여 덧붙인 후, 마지막 3 라운드(18~20 라운드) 키 복구 공격을 수행한다. 따라서 우선 확률 1로 16-라운드 차분 특성을 17-라운드 차분 특성으로 확장하는 방법을 살펴보고, 이렇게 만들어진 17-라운드 차분 특성을 통한 20-라운드 공격을 소개한다. 본 20-라운드 차분 공격에 대한 모델은 [그림 3]과 같다. 확률 1로 16-라운드 차분 특성을 17-라운드로 확장하는 방법은 첫 라운드 후의 출력 차분값이 원래의 16-라운드 차분 특성의 입력 차분값 $(0, \beta, 0, \alpha\oplus\beta)$이 되도록 하는 평문쌍을 만드는 것이다. 이런 평문쌍을 만들기 위해서 다음과 같은 구조체(Structure)를 구성한다. 고정된 각각의 32-비트 워드 A, B, C에 대하여 다음의 2^{32}개의 평문쌍을 구성할 수 있다.

$$P=(F(A\oplus B\oplus C\oplus RK_1)\oplus t_i,\ A,\ B,\ C),$$
$$P'=(F(A\oplus B\oplus C\oplus RK_1\oplus\beta)\oplus t_i\oplus\alpha\oplus\beta,\ A,\ B\oplus\beta,\ C)$$
단 $t_i=0,1,2,\ldots,2^{32}-1$



〔그림 3〕 SMS4에 대한 차분 공격 모델

고정된 A, B, C, α, β에 대해 1 라운드 키 RK_1의 값을 추측하여 2^{32}개의 (P, P')평문쌍을 구성한다. 만약 RK_1을 올바로 추측하였다면 (P, P')의 1 라운드 후의 차분값이 $(0, \beta, 0, \alpha\oplus\beta)$으로 앞 절의 16-라운드 차분 특성의 입력값과 동일하게 된다. 한 개의 구조체당 2^{32}개의 평문쌍이 있으므로 고정된 32-비트 워드 A, B, C의 값을 변화시켜 2^{93}개의 구조체를 준비한다. 따라서 앞 절에서 소개한 차분 특성을 만족하는 2^{125}개의 평문쌍을 선택할 수 있다. 선택된 평문쌍에 대하여 비밀키 K로 암호화된 20-라운드 SMS4의 암호문쌍 $C=(C_0, C_1, C_2, C_3)$, $C=(C_0', C_1', C_2', C_3')$을 요구한다.

앞 절의 차분 특성이 만족할 확률은 2^{-122}이므로 전체 2^{125}개의 평문쌍 중에서 본 차분 특성을 만족하는 옳은 평문쌍(right pair)의 기대값은 8개로 계산할 수 있다. 전체 암호문쌍 중에서 옳은 평문쌍을 찾기 위해 20-라운드 SMS4 암호문쌍의 두 개의 32-비트 워드 차분값을 확인함으로서 다음과 같은 여과 과정(Filtering)을 수행할 수 있다([그림 3] 참조). 17 라운드 후의 암호문쌍의 출력 차분값은 $(\beta, \alpha\oplus\beta, 0, \beta)$로서 18 라운드의 F 함수의 32-비트 워드 입력 차분값은 $\alpha=00000009_x$이다. 18 라운드 키와 17 라운드후의 정확한 중간값을 알 수 없기 때문에 F 함수의 정확한 출력 차분값은 알 수 없지만 SMS4의 S 박스의 성질에 의해서 α에 대한 정확히 127개의 출력 차분값을 알 수 있다. 이 127개의 차분값들의 집합을 Θ로 표기하자. 그러면 옳은 평문쌍에 대해서 20-라운드 SMS4 암호문쌍의 첫 번째 32-비트 워드의 차분값은 β가 되고 두 번째 32-비트 워드의 차분값은 집합 $\Lambda=\{\lambda\oplus\beta|\lambda\in\Theta\}$에 속하게 된다. 임의의 두 32-비트 워드에 대하여 첫 번째 워드의 차분값이 β일 확률은 2^{-32}이며, 두 번째 워드의 차분값이 집합 Λ에 속할 확률은 $2^{-25}(\approx 127\cdot 2^{-32})$이므로 전체 암호문쌍 중 여과되어 남는 쌍의 개수는 $2^{68}(=2^{125}\cdot 2^{-32}\cdot 2^{-25})$이다. 이 여과된 암호문쌍들을 테이블에 저장한다. SMS4의 마지막 3R-Attack은 다음과 같다.

1) 20 라운드 키의 최하위 1 바이트 $RK_{20,0}$를 추측한다. 저장된 각각의 20-라운드 SMS4의 암호문쌍에 대해 20 라운드 F 함수의 첫 번째 S 박스 출력 차분값을 계산한다. 즉, 다음의 두 식을 계산한다.

$$\gamma=s([C_0\oplus C_1\oplus C_2]_{[0]}\oplus RK_{20,0}) \oplus s([C_0'\oplus C_1'\oplus C_2']_{[0]}\oplus RK_{20,0}),$$

$$\delta=[D^{-1}(C_3\oplus C_3')]_{[0]}$$

$(RK_1, RK_{20,0})$에 대해 올바른 키를 추측했을 때 계산된 암호문쌍이 옳은 암호문쌍이라면 γ의 값은 δ의 값과 일치하게 된다. 따라서 두 값이 일치하지 않는다면 계산된 암호문쌍은 버린다. 옳은 암호문쌍이 아니라면 두 값이 같을 확률은 키 값에 상관없이 랜덤한 확률 2^{-8}을 따르게 된다. 이 과정에서 $2^{60}(=2^{68}\cdot 2^{-8})$개의 암호문쌍이 남게 된다.

2) 20 라운드 키의 최하위 두 번째 바이트 $RK_{20,1}$를 추측한다. 여과된 각각의 20-라운드 SMS4의 암호문쌍에 대해 20 라운드 F 함수의 두 번째 S 박스 출력 차분값을 계산한다. 즉, 다음의 두 식을 계산한다.

$$\gamma=s([C_0\oplus C_1\oplus C_2]_{[1]}\oplus RK_{20,1}) \oplus s([C_0'\oplus C_1'\oplus C_2']_{[1]}\oplus RK_{20,1})$$

$$\delta=[D^{-1}(C_3\oplus C_3')]_{[1]}$$

앞 단계와 마찬가지로, $(RK_1, RK_{20,0}, RK_{20,1})$에 대해 올바른 키를 추측했을 때 계산된 암호문쌍이 옳은 암호문쌍이라면 γ의 값은 δ의 값과 일치하게 된다. 따라서 두 값이 일치하지 않는다면 계산된 암호문쌍은 버린다. 옳은 암호문쌍이 아니라면 두 값이 같을 확률은 키 값에 상관없이 랜덤한 확률 2^{-8}을 따르게 된다. 이 과정에서 $2^{52}(=2^{60}\cdot 2^{-8})$개의 암호문쌍이 남게 된다.

3) 남아있는 20 번째 라운드 두 개의 바이트 키와 19 번째 라운드 각 바이트 키에 대해 단계 1, 2의 과정을 동일하게 적용한다. 마찬가지로 S 박스의 8-비트 출력 차분값을 비교하여 부분적으로 복호화된 암호문쌍을 여과할 수 있다. 이 단계까지 대략 $2^4(=2^{52}\cdot(2^{-8})^6)$개의 암호문쌍이 남게 된다.

4) 18 라운드 키의 최하위 1 바이트 $RK_{18,0}$를 추측한다. 마찬가지로 S 박스의 출력 차분값을 비교하여 암호문쌍을 여과한다. 이 단계에서는 이미 공격의 시작 과정에서 18 라운드 F 함수의 출력 차분값들에 대한 여과과정을 수행했기 때문에 대략 $2^{-7}\left(\approx\frac{1}{127}\right)$의 확률로 암호문쌍이 여과된다. 따라서 살아남는 암호문쌍의 개수는 $2^{-3}(=2^4\cdot 2^{-7})$로 기대할 수 있다. 반면 18 라운드까지 올바른 키가 추측됐을 때 차분 특성을 따르는 올바른 평문쌍의 개수는 8개로 기대할 수 있다. 즉, 마지막 단계까지 남아 있는 암호문쌍의 개수가 6개 이상인 키를 올바른 $RK_1, RK_{20}, RK_{19}, RK_{18,0}$에 대한 후보키로 저장한다.

5) RK_1를 이용하여 남아있는 각 평문쌍에 대하여 1 라운드 암호화후의 중간값을 계산하고 이 값을 $u=(u_0,u_1,u_2,u_3)$, $u'=(u'_0,u'_1,u'_2,u'_3)$로 표기한다. $RK_{2,0}$을 추측하고 다음의 식을 계산한다.

$$s([u_0\oplus u_1\oplus u_2]_{[0]}\oplus RK_{2,0})\oplus s([u'_0\oplus u'_1\oplus u'_2]_{[0]})\oplus RK_{2,0})=D^{-1}(\beta)_{[0]}$$

위의 방정식을 만족하는 6개 이상의 평문쌍이 존재하면, 추측된 키값들을 RK_1, RK_{20}, RK_{19}, $RK_{18,0}$, $RK_{2,0}$에 대한 올바른 키후보로서 출력한다.

6) 출력된 RK_1, RK_{20}, RK_{19}, $RK_{18,0}$, $RK_{2,0}$값에 대해서 비밀키의 남아있는 16 비트를 전수 조사하여 복구한다.

본 차분 공격의 데이터 복잡도는 2^{126}선택 평문이며, 테이블에 2^{68}개의 선택-평문 암호문쌍을 저장해야 하므로 메모리 복잡도는 $2^{73}(=2^{68}\cdot 16\cdot 2)$바이트를 요구한다. 또한 본 차분 공격의 시간 복잡도는 다음과 같이 계산할 수 있다. 단계 1에서의 시간 복잡도는 추측된 40 비트의 키$(RK_1, RK_{20,0})$에 대해, 2^{68}개의 암호문쌍을 한 개의 S 박스를 통하여 부분적으로 암호화하므로 $2^{102.68}\left(\approx 2^{68}\cdot 2^{40}\cdot 2\cdot\frac{1}{20}\cdot\frac{1}{4}\right)$ 20-라운드 SMS4의 암호화 연산이다. 단계 2에서의 시간 복잡도는 추측된 48 비트의 키$(RK_1, RK_{20,0}, RK_{20,1})$에 대해, 2^{60}개의 암호문쌍을 한 개의 S 박스를 통하여 부분적으로 암호화하므로 $2^{102.68}\left(\approx 2^{60}\cdot 2^{40+8}\cdot 2\cdot\frac{1}{20}\cdot\frac{1}{4}\right)$ 20-라운드 SMS4의 암호화 연산이다. 매 단계마다 여과되어 버리는 암호문쌍의 개수와 추가적으로 추측해야 되는 키 비트들의 수는 동일하므로 18 라운드의 첫 번째 바이트 키를 추측할 때까지 시간 복잡도는 $2^{102.68}$ 20-라운드 SMS4 암호화 연산으로 동일하다. 그러므로 단계 4까지의 시간 복잡도는 대략 $2^{105.85}(\approx 9\cdot 2^{102.68})$ 20-라운드 SMS4 암호화 연산이다. 단계 5에서의 시간 복잡도를 계산하기 위해서는 단계 4까지 남아있는 부분키의 개수를 계산해야 된

다. 포아송 분포 $X \sim Poi(\lambda = 2^{-3})$, $\Pr_X[X > 6] \approx 2^{-33.5}$에 의해서, 단계 4까지 살아남는 키의 개수는 대략 $2^{70.5}(=2^{104} \cdot 2^{-33.5})$로 계산할 수 있다. 그러므로 단계 5에서의 시간 복잡도는 대략 $2^{75.99}$ $(\approx 2^{70.5} \cdot 2^8 \cdot 7 \cdot 2 \cdot \frac{1}{20} \cdot \frac{1}{4})$이다. 단계 5에서 잘못 추측된 키에 대해서 6개 이상의 평문쌍이 식을 만족할 확률은 $2^{-42}(=(2^{-6})^7)$이기 때문에, $2^{36.5}(=2^{70.5} \cdot 2^8 \cdot 2^{-42})$개의 부분키들이 올바른 키의 후보로서 살아남는다. SMS4의 키스케쥴은 암호화 알고리즘과 동일한 구조를 갖기 때문에 비밀키의 남아있는 비트들은 전수 조사를 통하여 복구할 수 있다. 연속적인 72 비트 부분키 RK_{20}, RK_{19}, $RK_{18,0}$는 최대 $2^{36.5}$개의 후보를 갖고 있으므로, 단계 6에서의 시간 복잡도는 $2^{92.5}(=2^{36.5} \cdot 2^{128-72})$보다 적은 암호화 연산이 요구된다. 따라서 본 차분 공격의 전체 시간 복잡도는 대략 $2^{105.85}(\approx 9 \cdot 2^{102.68})$ 암호화 연산이다. 올바른 키에 대하여, 본 차분 공격의 성공률은 포아송 분포 $X \sim Poi(\lambda = 2^3)$에 의해서 $\Pr(X > 6) = 0.7$을 갖는다.

V. 결 론

본 논문에서는 SMS4에 대한 20-라운드 차분 공격을 소개했다. 차분 공격을 이용하여 2^{126}개의 선택 평문의 데이터 복잡도와 $2^{105.85}$의 시간 복잡도를 가지고 20-라운드 SMS4의 비밀키를 복구하였다. 본 논문의 이론적인 분석 결과는 축소된 SMS4에 대해서 설계자가 제시한 안전성을 만족하지 못함을 의미한다. 또한, 본 논문에서 제시된 차분 특성은 향후 SMS4의 다른 안전성 분석 기법에 적용될 것으로 기대된다.

참고문헌

[1] E. Biham, A. Shamir, "Differential crypt a nalysis of DES-like cryptosystems", *CRYPTO'90*, LNCS 537, pp. 2-21, Springer-Verlag, 1990.

[2] A. Biryukov, "The Boomerang Attack on 5 and 6-Round AES", *Advanced Encryp tion Standard-AES*. LNCS 3373, Sprin ger, Heidelberg, pp. 11-16, 2005.

[3] N. Ferguson, J. Kelsey, S. Lucks, B. Sch neier, M. Stay, D. Wagner, and D. Whiti ng. "Improved cryptanalysis of Rijndael". *FSE'00*, volume 1978 of LNCS, Springer -Verlag, pp. 213-230. 2001.

[4] The Institute of Electrical and Electronics Engineers (IEEE), http://grouper.ieee.org/groups/802/11

[5] National Institute of Standards and Technology, U.S.A., "Advanced Encryption Standard(AES)", FIPS-197, 2001.

[6] J. Cheon,, M. Kim, K. Kim, J. Lee, S. Kang, "Improved Impossible Differential Cryptanalysis of Rijndael and Crypton", *ICISC'01*, LNCS 2288, Springer, Heidelb erg, pp. 39-49, 2002.

[7] F. Liu,, W. Ji,, L. Hu,, J. Ding,, S. Lv,, A. Pyshkin, R.P. Weinmann, "Analysis of the SMS4 block cipher", *ACISP'07*, LNCS 4586, Spriger-Verlag, pp. 85-100, 2007.

[8] J. Lu, "Attacking Reduced-Round Versions of the SMS4 Block Cipher in the Chines WAPI Standard", *ICICS'07*, LNCS 4861, Springer-Verlag, pp. 306-318, 2007.

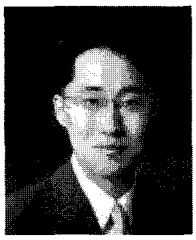
[9] Office of State Commercial Cryptography Administration, P.R. China, The SMS4 Block Cipher (in Chinese). Archive available at http://www.oscca.gov.cn/UpFile/200621016423 19799 0.pdf

[10] R. C. W. Phan. "Impossible differential cryptanalysis of 7-round Advanced Encryption Standard AES", *Information Processing Letters*, vol 91, pp. 33-38, 2004.

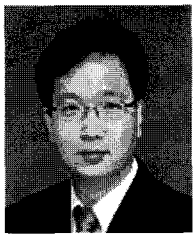
〈著者紹介〉

김 태 현 (Taehyun Kim) 학생회원
2005년 2월 : 고려대학교 수학과 학사
2006년 2월~현재 : 고려대학교 정보경영공학전문대학원 석사과정
<관심분야> 블록암호, 스트림 암호 및 해쉬 함수의 분석 및 설계

김 종 성 (Jongsung Kim) 정회원
2000년 8월 : 고려대학교 수학과 학사
2002년 8월 : 고려대학교 수학과 석사
2006년 11월 : K.U.Leuven, ESAT/SCD-COSIC 박사
2007년 2월 : 고려대학교 정보보호대학원 박사
2007년 3월~2008년 3월 : 고려대학교 정보보호연구원 박사후연구원
2008년 4월~현재 : 고려대학교 정보보호연구원 연구교수
<관심분야> 암호 알고리즘 설계 및 분석



성 재 철 (Jaechul Sung) 종신회원
1997년 8월 : 고려대학교 수학과 학사
1999년 8월 : 고려대학교 수학과 석사
2002년 8월 : 고려대학교 수학과 박사
2002년 8월~2004년 1월 : 한국정보보호진흥원 선임연구원
2004년 2월~현재 : 서울시립대학교 수학과 조교수
<관심분야> 암호 알고리즘 설계 및 분석



홍 석 희 (Seokhie Hong) 종신회원
1995년 2월 : 고려대학교 수학과 학사
1997년 2월 : 고려대학교 수학과 석사
2001년 2월 : 고려대학교 수학과 박사
1999년 8월~2004년 2월 : (주) 시큐리티 테크놀로지스 선임연구원
2004년 4월~2005년 2월 : K.U.Leuven 박사후연구원
2005년 3월~현재 : 고려대학교 정보경영공학전문대학원 조교수
<관심분야> 암호 알고리즘 설계 및 분석, 컴퓨터 포렌식