情報保護學會論文誌
第17卷 第2號, 2007. 4

스트림 암호 MICKEY의 TMD-Tradeoff와 내부 상태 엔트로피의 손실에 관한 분석

김 우 환,[1†] 홍 진[2‡]
[1]국가보안기술연구소, [2]서울대학교

Analysis on TMD-Tradeoff and State Entropy Loss of Stream Cipher MICKEY

Woo-Hwan Kim,[1†] Jin Hong[2‡]
[1]National Security Research Institute(NSRI), [2]Seoul National University(SNU)

요 약

본 논문에서는 스트림 암호 MICKEY의 두 가지 취약점에 대해서 논한다. 첫째, time-memory-data tradeoff 공격이 가능함을 보인다. 둘째, 상태 갱신 함수 (state update function)를 반복해서 적용할수록 내부 상태 (internal state)의 엔트로피가 감소하므로 다르게 시작된 키 스트림이 마침내 같아질 수 있다.

ABSTRACT

We give two weaknesses of a recently proposed streamcipher MICKEY. We show time-memory-data tradeoff is applicable. We also show that the state update function reduces entropy of the internal state as it is iterated, resulting in keystreams that start out differently but become merged together towards the end.

Keywords : *stream cipher MICKEY, TMD tradeoff, Entropy loss*

Ⅰ. 서 론

Babbage와 Dodd에 의해 eSTREAM에 제안된 스트림 암호 MICKEY [4] 는 low-end 하드웨어에 적합하도록 설계되었다. MICKEY는 80비트 안전성을 목표로 하고 있으며 Symmetric Key Encryption Workshop (SKEW, Århus, Denmark, May, 2005)에 채택되어 발표된 스트림 암호 중의 하나이다.

MICKEY의 내부 상태 (internal state)는 두 종류의 80 비트 FSR (Feedback Shift Register)으로 이루어져 있으며 두 레지스터는 서로를 불규칙적으로 클로킹하도록 설계되었다. 두 레지스터의 출력을 XOR 하여 키 수열이 생성되므로 단지 두 개의 레지스터만으로 하드웨어 구현이 가능하다. 일반적으로 80비트의 안전성을 가지기 위해서 내부 상태의 크기는 적어도 160비트 이상이 되어야 하므로 MICKEY는 하드웨어 비용 관점에서 최적에 가깝다고 할 수 있다.

본 논문에서는 MICKEY의 두 가지 안전성 취약점에 대하여 논의한다. 첫째, 온라인 공격 복잡도가 전수조사보다 낮은 time-memory-data tradeoff (TMD-tradeoff) 공격이 가능하다. 일반적으로 내부 상태의 크기가 키 크기의 두 배가 되면 TMD-tradeoff 공격에 대해 안전한 것으로 믿어지고 있으나 MICKEY에 대해서 BSW 샘

접수일: 2006년 8월 9일; 채택일: 2006년 12월 12일
† 주저자, whkim5@etri.re.kr
‡ 교신저자, jinhong@snu.ac.kr

플링을 이용하면 검색 공간을 줄일 수 있고 줄어든 검색 공간에 대해 효율적으로 TMD-tradeoff 공격을 적용할 수 있다.

둘째, MICKEY의 상태 갱신 함수 (state update function)가 일대일 함수가 아니며 상태가 갱신될수록 내부상태의 엔트로피가 줄어든다는 것을 보인다. MICKEY의 설계자들은 키 수열이 반복되지 않도록 2^{40} 이하 길이의 키 수열로 제한하여 사용하도록 제안하였으나 이 경우에도 다르게 출발한 키 수열이 마침내 같아지는 현상을 피할 수 없다.

먼저 MICKEY 알고리즘에 대해서 간략히 기술하고 나서 앞에서 언급한 취약점에 대해서 논의하기로 한다.

Ⅱ. 스트림 암호 MICKEY

본 절에서는 MICKEY에 대해서 간략히 소개하고 앞으로의 논의에 사용될 기호에 대해서 설명한다. 키와 초기 벡터로부터 내부 상태를 초기화하는 과정은 논의의 대상이 아니므로 여기에서 다루지 않기로 한다.

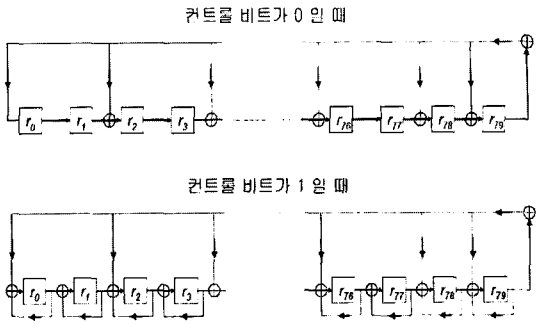
MICKEY의 내부 상태는 두 개의 80비트 레지스터 R과 S로 이루어져 있다. 레지스터 R과 S의 셀을 각각 r_i와 s_i ($0 \leq i \leq 79$)로 나타내기로 한다.

2.1 컨트롤 비트

내부 상태를 갱신하기 위해서 먼저 레지스터의 값으로부터 컨트롤 비트를 계산한다. 레지스터 R의 컨트롤 비트는 $s_{27} \oplus r_{53}$이고 레지스터 S의 컨트롤 비트는 $s_{53} \oplus r_{26}$이다.

2.2 내부 상태의 갱신

레지스터 R은 선형으로 갱신된다. 즉 R은 LFSR (Linear Feedback Shift Register)이다. R의 클로킹에서의 탭 위치는 {0,2,4,6,7,8,9,13,14,16,17,20,22,24,26,27,28,34,35,37,39,41,43,49,51,52,54,56,62,67,69,71,73,76,78,79}이다. R의 컨트롤 비트가 0이면 보통의 LFSR과 같이 클로킹이 이루어지며 R의 컨트롤 비트가 1이면 $J = 2^{40} - 23$번 클로킹이 된다. R의 특성 다항식이 $x^J + x + 1$을 나누도록 설계하여 J번 클로킹을 한 번에 할 수 있도록 하였다. R의 갱신 과정은 [그림 1]과 같다. 한편, 레지스터 S는 고정된 시퀀스를 이용하여 비선형으로 갱신된다. 사용되는 시퀀스는 COMP0, COMP1, FB0, FB1이며 컨트롤 비트가 0이면 COMP0, COMP1, FB0가 사용되고 컨트롤 비트가 1이면 COMP0, COMP1, FB1이 사용된다.



[그림 1] 레지스터 R의 클로킹

레지스터 R과 레지스터 S의 컨트롤 비트를 가정하면 이전 레지스터 값을 계산할 수 있다. 가능한 컨트롤 비트 쌍은 네 가지이므로 특정 내부 상태의 이전 상태는 최대 네 가지가 있을 수 있다. 컨트롤 비트를 가정하여 이전 내부 상태를 구한 후 다시 계산한 컨트롤 비트가 가정한 컨트롤 비트와 일치하면 유효한 이전 내부 상태가 된다.

2.3 출력

레지스터 R과 레지스터 S의 피드백 비트를 각각 r_0, s_0라 두었을 때, 출력 키 스트림 비트 z는 $z = r_0 \oplus s_0$가 된다. 내부 상태가 1회 갱신될 때마다 1비트씩 출력한다.

Ⅲ. BSW 샘플링을 이용한 TMD tradeoff

본 절에서는 MICKEY의 내부 상태에 대하여 온라인 공격 복잡도가 전수 조사보다 낮은 TMD-tradeoff 공격이 가능함을 보인다.

3.1 BS-tradeoff

스트림 암호에서 주어진 키 수열로부터 내부 상태를 구하는 문제는 다음의 문제로 일반화 될 수 있다.

□ 일방향 함수 $f: X \rightarrow Y$와 집합 $D = \{y_i\} \subset Y$가 주어졌을 때, $f(x_i) = y_i$를 만족하는 x_i를 적어도 하나 찾아라.

이를 스트림 암호에 적용하면 X는 내부 상태 전체의 집합[1], Y는 내부 상태를 결정하기에 충분히 긴 길이[2]의 부분 키 수열의 집합, f는 내부 상태로부터 부분 키 수열을 얻는 함수가 된다. 만약 어떤 부분 키 수열로 내부 상태를 복구할 수 있다면 이후에 출력될 키 수열을 모두 얻을 수 있다.

Time-memory-(data) tradeoff (TMD- tradeoff) 공격은 Hellman [9] 에 의해 블록암호를 공격하기 위해 처음 제안되었으며 이후 스트림암호에 대한 TMTO 등 많은 연구결과들이 발표되었다. 본 절에서는 Biryukov와 Shamir의 방법 [5] (BS-tradeoff)을 MICKEY에 적용한다. 설명에 사용될 기호는 다음과 같다.

- N : 검색 공간의 크기
- P : 사전 계산량 (pre-computation time).
- M : 사전 계산 결과를 저장하는데 필요한 저장 공간의 크기.
- T : 온라인 공격 시간.
- D : 실제 공격 시 필요한 데이터 크기.

BS-tradeoff에서는 다음의 tradeoff 곡선을 만족하는 (T, M, D)를 이용한다.

$$TM^2D^2 = N^2,\ 1 \le D^2 \le T \tag{1}$$

이를 이용한 공격은 사전 계산 단계와 온라인 계산 단계로 이루어진다. 공격자는 먼저 사전 계산을 통하여 저장공간의 크기가 M인 테이블을 구성하며 이를 위해 필요한 사전 계산량은 $P=N/D$이다. 온라인 공격 단계에서는 크기 D의 데이터가 주어졌을 때, 공격 시간 T 안에 높은 확률로 적어도 하나의 역원을 구할 수 있다. 관계식 (1)을 만족하는 전형적인 값으로 $T=M=N^{1/2}, D=N^{1/4}$를 들 수 있으며, 이 때 필요한 사전 계산량은 $P=N^{3/4}$이다. 일반적으로 (T, M, D) 중 가장 큰 값을 이 공격의 공격 복잡도로 간주하며 사전 계산량은 공격 복잡도에서 제외한다. 생일 역설에 기반한 TMD-tradeoff는 스트림 암호의 내부 상태 크기를 안전도의 두 배 이상이 되도록 설계하는 이유 중의 하나이며 MICKEY의 경우에도 이런 경향을 따라 80비트 안전도를 얻기 위해 160비트 크기의 내부 상태를 사용한다.

1) X를 (키, 초기벡터)의 집합으로 두기도 한다. 내부 상태의 집합과 비교하여 크기가 작은 집합을 택한다.

2) 일반적으로 내부 상태 공간의 크기가 2^n일 때, n 비트 길이의 키 수열을 이용한다.

3.2 검색 공간의 축소

BSW 샘플링 [6] 을 이용하면 검색 공간을 줄일 수 있다. 다음의 성질을 만족하는 집합 X', Y'과 함수 h를 찾으려 한다.

- $X' \subset X, Y' \subset Y, f(X') = Y'$.
- Y'의 원소들은 Y'에 속하지 않는 원소들과 쉽게 구분된다.
- 효율적으로 계산할 수 있으며 일대일에 가까운 함수 $h: Y' \to X'$가 존재한다.

위 성질을 만족하는 X', Y', h가 존재할 때, f를 X'에 제한한 함수 $f': X' \to Y'$를 생각하자. $N' = |X'| = |Y'| = rN$이라 두고 주어진 D개의 데이터로부터 Y'에 속하는 $D' = rD$개의 데이터를 얻을 수 있다고 가정하자. 이를 이용한 tradeoff 곡선은

$$TM^2D'^2 = N'^2,\ P = N'/D',\ 1 \le D'^2 \le T \tag{2}$$

가 되며 이를 다시 쓰면

$$TM^2D^2 = N^2,\ P = N/D, D' = rD,\ 1 \le D'^2 \le T \tag{3}$$

이 된다. 위 곡선 상의 전형적인 점으로 $T=M=D=N^{2/5}$, $T=M=D=N^{2/5}$, $D'=N^{1/5}$, $P=N^{3/5}$을 들 수 있다. 이 값을 검색공간을 축소하기 전과 비교해 보면 online 공격 시간 (T), 메모리 (M), 사전 계산량 (P)은 줄어들었으며 필요한 데이터의 양 (D)은 늘어났다. 이제, BSW 샘플링을 이용한 TMD-tradeoff를 MICKEY에 적용한다.

3.3 MICKEY 키 수열의 샘플링

27개의 0으로 시작하는 키 수열의 집합을 Y'라 두면 Y'의 원소는 Y'에 속하지 않는 원소들과 쉽게 구분된다. 이제 Y'의 f에 대한 역이미지를 X'이라 했을 때, 집합 X'와 함수 $h: Y' \to X'$에 대해 살펴보자. 이해를 돕기 위해 h를 계산하는 과정을 각 단계별로 기술하였다. R의 클로킹 방식은 [그림 1]을 참고하기 바란다.

(1) Y'의 원소 y의 0으로 이루어진 초기 27비트를 무시하면 y를 133비트 키 수열로 간주할 수 있다.
(2) y의 초기 80비트로 레지스터 S를 채워 넣는다.
(3) 나머지 53비트로 레지스터 R의 $r_1,...,r_{53}$을 채워 넣는다. R의 채워지지 않은 부분, 즉 $x_0, x_{54},...,x_{79}$를 미지수로 두고 키 수열의 초기 27비트가 0이

되도록 미지수를 아래와 같이 결정한다.

(4) 첫 번째 출력값이 $r_0 \oplus s_0 = 0$이며 s_0가 이미 결정되어 있으므로 $r_0 = s_0$을 얻는다.

(5) S와 R의 컨트롤 비트 $s_{53} \oplus r_{26}$과 $s_{27} \oplus r_{53}$을 계산한다.

(6) S를 컨트롤 비트에 따라 클로킹한다.

(7) R을 컨트롤 비트에 따라 클로킹하고 각 셀을 변수 $x_{54},\ldots,x_{79}$의 일차 결합으로 나타낸다. 특히, 새로운 r_0의 값은 R의 컨트롤 비트가 0이면 x_{79}가 되고 컨트롤 비트가 1이면 $x_{79} \oplus x_0$가 된다. 키 수열의 두 번째 비트가 0이 되도록 x_{79}을 결정하면 x_{79}를 상수로 둘 수 있다.

(8) 피드백 비트가 결정되면 갱신된 $r_0,\ldots,r_{53}$이 정해지며 나머지 값들은 $x_{54},\ldots,x_{78}$의 일차 결합으로 표현된다. 예를 들어 r_{79}는 컨트롤 비트에 따라 $x_{78} \oplus x_{79}$ 또는 x_{78}이 되며 컨트롤 비트는 이미 알려져 있다.

(9) 컨트롤 비트를 계산하고 이에 따라 S와 R을 클로킹 한다.

(10) 새로 계산된 s_0를 이용하여 출력 키 수열이 0이 되도록 피드백 비트를 결정한다.

(11) 위의 방식으로 미지수를 결정해 나간다. 한 번 클로킹 할 때 마다 하나의 미지수가 결정되므로 전체를 결정하기 위해 26번의 클로킹이 필요하다.

이와 같은 샘플링은 특정 형태 (초기 키 수열이 27개의 0으로 시작)의 키 수열을 가지는 내부상태를 쉽게 찾을 수 있다는 사실에 기인한다.

3.4 샘플링을 이용한 TMD-Tradeoff

본 소절에서는 축소된 검색 공간 X'에 대한 TMD-tradeoff의 공격 복잡도에 대해 살펴본다. 2^{60} 크기의 샘플링 이전 데이터가 주어졌다고 하자. 예를 들어 각각의 길이가 $2^{40}+159$ 비트인 2^{20}개의 키 수열이 주어지면 160 비트 길이의 키 수열을 2^{60}개 얻을 수 있다. 키 수열들은 2^{-27}의 확률로 초기 27 비트가 0이 되므로 샘플링을 거치고 나면 약 2^{33}개의 키 수열이 남는다. 예를 들어 $T=2^{66}$, $M=2^{67}$, $D=2^{33}$이고$N=2^{133}$이면 TMD-tradeoff 관계식 (1)을 만족한다. 이 때, 사전 계산량은 $P=2^{93.5}$이다. 사전 계산량이 키의 전수조사 공격량 보다 크기 때문에 위 공격은 MICKEY에 대한 유효한 공격으로 간주되지 않을 수 있지만 샘플링이 가능하다는 점 자체를 약점으로 볼 수 있으며 이로 인하여 다른 공격이 가능할 지도 모른다.

Ⅳ. State entropy 손실

MICKEY의 내부 상태 갱신은 다음과 같다. 먼저 내부 상태로부터 두 비트의 컨트롤 비트를 계산하며 컨트롤 비트에 따라 내부 상태가 갱신된다. 이와 같이, 동일한 정보가 이중으로 사용되면 계산 결과는 기존의 정보를 모두 담고 있지 못할 가능성이 있다. MICKEY의 상태 갱신 함수는 일대일 함수가 아니며 상태 갱신이 거듭될 수록 내부 상태의 엔트로피가 줄어들게 된다. 본 절에서는 실험을 통해 실제로 엔트로피가 줄어드는 정도를 추정하고 이에 대한 근거를 제시한다.

4.1 엔트로피

주어진 집합 $X=\{x_i\}_{i\in I}$에 대해 각각의 x_i가 p_i의 확률로 나타날 때, 집합 X의 엔트로피는 다음과 같이 정의된다.

$$H(X) = -\sum_{i\in I} p_i \log_2 p_i$$

예를 들어 집합 Y가 N개의 원소를 가지고 있고 각 원소가 나타날 확률이 같다면 $H(Y)=\log_2 N$이 된다. 따라서 크기가 2^n이며 균등한 확률분포를 가지는 집합의 엔트로피는 n이 된다. 크기가 N이고 균등분포를 가지는 집합에서 정의된 함수 φ에 대해 다음의 기호를 정의하자. 여기서 $\mathrm{Im}(\varphi)$는 φ의 치역을 의미한다.

$$\mathrm{EL}(\varphi) = \log_2 N - H(\mathrm{Im}(\varphi)),$$
$$\overline{\mathrm{EL}}(\varphi) = \log_2 N - \log_2 |\mathrm{Im}(\varphi)|.$$

EL은 균등한 분포를 가지는 집합에 함수 φ를 적용했을 때 생기는 엔트로피의 손실량을 나타낸다. $\overline{\mathrm{EL}}$은 치역의 크기만을 이용해 구한 값으로 EL의 추정값으로 간주될 수 있으며 치역의 각 원소가 같은 확률로 나타난다면 EL과 $\overline{\mathrm{EL}}$은 같아진다. 여기서 임의의 함수 φ에 대해 항상 $\mathrm{EL}(\varphi) \geq \overline{\mathrm{EL}}(\varphi)$가 성립한다.

4.2 상태 갱신 함수와 랜덤 함수의 비교

크기가 N인 집합에 정의된 랜덤 함수를 φ_N이라 두자. 랜덤 함수는 다음과 같은 특성을 가진다는 것이 잘

알려져 있다[7,10].

보조정리 1. N이 증가함에 따라 φ_N의 치역 크기의 기대값은 $(1-1/e)N$에 점차 가까워진다.

보조정리 2. N이 증가함에 따라 $\overline{\mathrm{EL}}$의 기대값은 $-\log_2(1-1/e)\sim 0.6617$에 점차 가까워진다.

따라서 랜덤 함수를 스트림 암호의 상태 갱신 함수로 사용한다면 내부 상태를 한 번 갱신했을 때 0.66 비트 이상의 엔트로피 손실을 기대할 수 있다. 앞으로의 논의에서는 MICKEY의 상태 갱신 함수가 엔트로피 손실 관점에서 랜덤함수와 비슷한 특성을 가짐을 보인다.

MICKEY의 상태 갱신 함수의 성질을 살펴보기 위해 다음의 과정을 생각하자.

(1) 레지스터 R과 S를 랜덤하게 설정한다.

(2) 컨트롤 비트를 가정하고 레지스터 R과 S를 거꾸로 클로킹 하여 이전 내부 상태를 구한다. R의 컨트롤 비트를 0과 1이라 가정했을 때 이전의 레지스터를 각각 (R_0,R_1)이라 두자. 마찬가지로 레지스터 S에 대해서도 (S_0,S_1)을 계산한다.

(3) 네 가지의 가능한 (R_i,S_j)에 대하여 컨트롤 비트 $s_{27}\oplus r_{53}, s_{53}\oplus r_{26}$을 계산한 후 (i,j)와 비교하여 일치하는 쌍의 갯수를 구한다.

위 과정은 특정 내부 상태로 갱신될 수 있는 이전 상태의 개수를 구하는 과정이다. 이전 상태의 갯수는 0개에서 4개까지 가능하다. 위 과정을 2^{20}번 반복하여 얻은 결과를 〔표 1〕에 정리하였다. 이 결과에 의하면 임의로 선택한 2^{20}개의 상태 중 307,988개의 상태는 이전 상태를 가지지 않고 740,588개의 상태는 이전 상태를 가진다. 따라서 내부 상태의 약 70.63%가 상태 갱신 함수 f의 치역에 속한다는 것을 알 수 있다. 이러한 현상이 내부 상태 전체에 대해 성립한다고 가정하면 $\overline{\mathrm{EL}}(f)\sim -\log_2 0.7063\sim 0.5017$을 얻을 수 있다. 이 값을 보조정리 2로부터 얻을 수 있는 0.6617과 비교하면 f가 엔트로피 손실의 관점에서 랜덤 함수와 치환 함수 사이에 위치함을 알 수 있다.

〔표 1〕 이전 내부 상태 개수에 따른 2^{20}개의 내부 상태의 분포

이전상태 개수	0	1	2	3	4	합계
내부상태	307988	452017	279418	0	9153	2^{20}

〔표 2〕 $f\circ f$에 대한 이전 내부 상태 개수에 따른 2^{20}개의 내부 상태의 분포

이전상태 개수	0	1	2	3	4	5
내부상태	$2^{18.81}$	$2^{17.95}$	$2^{17.92}$	$2^{15.82}$	$2^{14.65}$	$2^{10.35}$
이전상태 개수	…	10	11	12	13~16	합계
내부상태	…	$2^{3.32}$	0	$2^{4.00}$	0	2^{20}

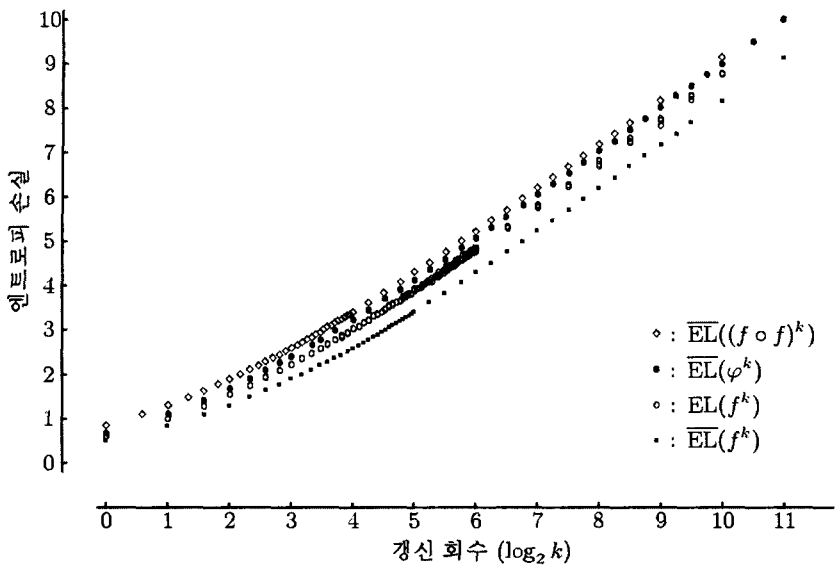
이제 f를 두 번 적용한 경우에 대해 살펴보자. 임의로 선택한 상태의 이전 상태가 존재하면 다시 그 이전 상태를 구한다. $f\circ f$를 적용했을 때 치역의 크기는 588,990이며 2^{20}개의 내부 상태 중 약 56.17%가 이에 해당한다. 앞에서와 같이 $\overline{\mathrm{EL}}$을 계산하면 $\overline{\mathrm{EL}}(f\circ f)\sim 0.8321$을 얻을 수 있으며 엔트로피를 보존하는 관점에서 $f\circ f$는 랜덤 함수보다 좋지 않은 특성을 보였다. $f\circ f$에 대한 실험 결과는 〔표 2〕에 정리하였다.

본 절에서는 상태 갱신함수 f를 2^{40}번 적용했을 때 엔트로피의 손실량을 계산하는 것이 목표이다. $f\circ f$를 2^{39}번 적용하는 것은 f를 2^{40}번 적용하는 것에 해당한다. 위의 실험 결과에 따르면 f는 랜덤 함수보다 엔트로피 손실이 적었지만 $f\circ f$는 랜덤 함수보다 더 많은 엔트로피 손실이 있었다. 따라서 함수 φ의 엔트로피 손실을 이용하여 f의 엔트로피 손실을 추정하는 것이 가능할 것으로 보인다.

4.3 갱신 함수와 랜덤 함수의 반복 적용

f를 적용했을 때의 치역, $\mathrm{Im}(f)$의 엔트로피를 계산해 보자.

〔표 1〕에 의하면 약 $307988/2^{20}\cdot 2^{160}\sim 2^{158.23}$개의 내부 상태가 f를 적용했을 때 나타나지 않으므로 $2^{158.23}$개의 내부 상태가 나타날 확률은 0이다. 하나의 이전 상태를 가지는 내부 상태의 갯수는 $452017/2^{20}\cdot 2^{160}\sim 2^{158.79}$개이며 각각이 나타날 확률은 $1/2^{160}$이다. 두 개의 이전 상태를 가지는 내부 상태의 개수는 $279418/2^{20}\cdot 2^{160}\sim 2^{158.09}$개이며 각각이 나타날 확률은 $2/2^{160}$이다. 네 개의 이전 상태를 가지는 내부 상태의 갯수는 $9153/2^{20}\cdot 2^{160}\sim 2^{153.16}$개이며 각각이 나타날 확률은 $4/2^{160}$이다. 이 확률들을 모두 합하면



〔그림 2〕 내부상태 엔트로피의 손실

$$0\cdot 2^{158.23}+1/2^{160}\cdot 2^{158.79}+2/2^{160}\cdot 2^{158.09}$$
$$+3/2^{160}\cdot 0+4/2^{160}\cdot 2^{153.16}\sim 0.99894$$

이 되어 1이 되지 않지만 1과 충분히 가까우므로 오차를 무시하기로 한다. 만약 2^{20}개 보다 많은 샘플을 이용하면 확률의 합이 1과 더 가까워질 것을 기대할 수 있다.

위에서 계산한 값들을 이용하여 엔트로피의 정의에 따라 계산하면 다음을 얻는다.

$$\begin{aligned}&H(\mathrm{Im}(f))\\&=-(2^{158.79}\frac{1}{2^{160}}\log_2\frac{1}{2^{160}}+2^{158.09}\frac{2}{2^{160}}\log_2\frac{2}{2^{160}}\\&\quad+0\frac{3}{2^{160}}log_2\frac{3}{2^{160}}+2^{153.16}\frac{4}{2^{160}}log_2\frac{4}{2^{160}})\\&\sim 160-0.6028.\end{aligned}$$

따라서 $\mathrm{EL}(f)=0.6028$를 얻을 수 있다. 예상했던 것처럼 이 값은 앞에서 구한 $\overline{\mathrm{EL}}(f)=0.5017$ 보다 큰 값이며 같은 과정을 $f\circ f$에 대해 적용하면 $\mathrm{EL}(f\circ f)=0.9913$을 얻는다.

f를 k번 합성한 함수를 f^k라 나타내자. 위 방법을 이용하여 여러 개의 k 값에 대해 $\mathrm{EL}(f^k)$를 계산하였으며 계산 결과는 〔그림 2〕의 ○로 표현되었다. 그림에서 x축은 k의 로그값이며 y축은 엔트로피 손실 비트 수를 의미한다. k 값이 커짐에 따라 실험값들 사이의 차이가 커졌으며 정확도를 높이기 위해 k 값이 클 때에는 샘플의 수를 증가시켜 실험하였다.

〔그림 2〕에 $\overline{\mathrm{EL}}(f^k),\overline{\mathrm{EL}}(\varphi),\overline{\mathrm{EL}}(f\circ f)$의 그래프를 함께 나타내었다. $\overline{\mathrm{EL}}(f^k)$와 $\mathrm{EL}(f^k)$의 계산에 동일한 실험 데이터가 사용되었으며 $\overline{\mathrm{EL}}(\varphi)$의 그래프는 잘 알려진 다음의 보조 정리를 이용하여 얻었다[7,10].

보조정리 3. 크기가 N인 집합에 정의된 랜덤 함수를 k번 적용했을 때 치역 크기의 기대값은 N이 커짐에 따라 $(1-\tau_k)N$에 가까워진다. 여기서 τ_k는 점화식 $\tau_0=0,\tau_{k+1}=\exp(\tau_k-1)$에 의해 귀납적으로 정의된다.

집합의 크기 N이 고정되어 있을 때 함수 φ를 계속 적용하면 마침내 시작한 값이 반복되어 사이클을 이루며 사이클 길이의 기대값은 $O(\sqrt{N})$이 된다는 것이 알려져 있다. 따라서 k가 N과 비슷해지면 치역의 크기가 더 이상 줄어들지 않는다. 하지만 여기에서는 $N=2^{160},1\le k\le 2^{40}$이므로 k가 $\sqrt{N}$에 비해 충분히 작다. 따라서 큰 무리없이 치역의 크기가 $(1-\tau_k)N$이 될 것이라고 가정할 수 있다. φ를 k 번 반복 적용했을 때 치역의 크기가 $(1-\tau_k)N$이 된다고 가정하면 $\overline{\mathrm{EL}}(\varphi)=\log_2 N-\log_2(1-\tau_k)N=-\log_2(1-\tau_k)$가 된다. $\overline{\mathrm{EL}}(\varphi^k)$는 위 그림에서 기호 •로 나타내었다.

〔표 3〕 랜덤 함수를 반복 적용했을 때 엔트로피 손실의 추정값

k	1	2	2^2	2^3	2^4
$-\log_2(1-\tau_k)$	0.66	1.09	1.68	2.40	3.23
k	...	2^9	2^{10}	2^{10}	2^{10}
$-\log_2(1-\tau_k)$	...	8.01	9.01	10.01	11.00

〔그림 1〕에서 랜덤 함수의 엔트로피 손실을 나타내는 $\mathrm{EL}(\varphi^k)$의 그래프는 f에 대한 그래프와 $f\circ f$에 대한 그래프 사이에 위치하고 있으며 갱신 회수가 증가하더라도 이 현상이 유지될 것을 기대할 수 있다. 또한, 상태 갱신 함수 f의 엔트로피 손실을 의미하는 $\mathrm{EL}(f^k)$의 값이 $\overline{\mathrm{EL}}(\varphi^k)$와 매우 비슷하다는 것을 알 수 있다.

$\sqrt{N}$에 비해 충분히 작은 k에 대해 위 내용이 성립함을 기대할 수 있으므로 $-\log_2(1-\tau_k)$를 계산하여 $\mathrm{EL}(f^k)$의 값을 추정하자.

추측 1. $k\to\infty$ 이면 $-\log_2(1-\tau_k)\to\log_2 k-1$이 성립한다.

위 추측을 증명할 수 없었으나 〔표 3〕의 계산 결과는 이를 강하게 뒷받침한다.

지금까지의 결과는 다음과 같이 요약할 수 있다.

(1) 엔트로피 보존의 관점에서 상태 갱신 함수 f는 랜덤 함수와 비슷한 특성을 가진다.

(2) k가 $\sqrt{N}$ 보다 충분히 작을 때 f^k의 엔트로피 손실 $\mathrm{EL}(f^k)$는

$$\overline{\mathrm{EL}}(\varphi^k)\sim-\log_2(1-\tau_k)\sim\log_2 k-1$$

으로 근사시킬 수 있으며 이에 대한 근거를 제시하였다.

(3) 위 내용에 근거하여 f에 의해 내부 상태를 2^{40}번 갱신하면 39 비트 정도의 엔트로피 손실이 발생할 것으로 결론 내린다.

여기서 39 비트는 정확한 수치가 아니라 실험에 근거한 대략적인 근사값이다.

4.4 안전성 분석

앞에서 MICKEY의 상태 갱신 함수는 일대일 함수가 아니며 랜덤 함수와 비슷한 성질을 가짐을 살펴보았다. 이러한 성질을 가진다는 사실만으로도 f는 스트림 암호의 상태 갱신 함수로 적절하지 못하다 할 수 있다. MICKEY의 설계자들은 키와 초기 벡터를 설정한 후 2^{40} 비트 이상 사용하지 않도록 권고하고 있으므로 아래의 분석에서는 2^{40} 비트 이하의 키 수열을 이용한다. 앞에서 살펴본 바와 같이, 내부 상태를 2^{40}번 갱신하면 39 비트의 엔트로피의 손실을 기대할 수 있다. 초기 내부 상태가 160 비트의 엔트로피를 가지므로 2^{40} 비트를 출력하고 나면 내부 상태는 121 비트의 엔트로피를 가지게 된다. 각각의 길이가 $2^{40}-\epsilon$ 비트인 키 수열을 얻었다고 가정하자. 단, 여기서 ε은 2^{40}에 비해 작은 값 (예를 들어 2^{20}) 이다. 각각의 키 수열에 대해 키 수열을 출력하고 난 후의 내부 상태를 생각할 수 있다. 만약 이러한 키 수열을 $2^{60.5}$개 얻을 수 있다면 생일 역설에 의해 마지막 내부 상태가 같아지는 두 개의 키 수열이 높은 확률로 존재한다. 만약 내부 상태가 같아진다면 그 이후에 동일한 키 수열을 출력하게 된다. 다시 말해서, 긴 키 수열이 충분히 많이 주어지면 비록 다르게 시작된 키 수열이라도 종국에는 같아지는 키 수열이 존재한다. 여기에서, 주어진 키 수열이 동일한 키로부터 생성되었든, 다른 키로부터 생성되었든 상관없다. 이러한 분석이 유효한 공격으로 간주되지 않을 수 있다. 하지만 MICKEY를 이용해 긴 길이의 키 수열을 여러 번 사용하면 위험할 수 있음을 보여 준다.

주어진 키 수열을 한 비트씩 옮기는 방법을 이용하면 새로운 키 수열을 추가로 얻을 수 있다. 예를 들어 길이가 2^{40}인 키 수열을 이용하여 길이가 $2^{39}-\epsilon$인 키 수열을 $2^{39}+\epsilon$개 얻을 수 있다. 단, 같은 키 수열로부터 비롯되는 내부 상태 ($2^{39}-\epsilon$ 비트 출력 후의 $2^{39}+\epsilon$개의 내부 상태)들 사이에는 일치하는 쌍을 찾을 수 없다. 2^{40} 길이의 키 수열이 m개 주어졌을 때 2^{39} 비트 출력 후의 내부 상태들을 비교하여 일치하는 쌍이 존재할 확률을 살펴보자. 2^{39} 비트 출력 후 내부 상태의 엔트로피가 N일 때 충돌쌍이 있을 확률은

$$\begin{aligned}&1-(1-\frac{2^{39}}{N})\cdot(1-\frac{2\cdot 2^{39}}{N})\cdots(1-\frac{(m-1)\cdot 2^{39}}{N})\\&=1-\prod_{k=1}^{m-1}(1-\frac{k\cdot 2^{39}}{N})\sim 1-\exp(\sum_{k=1}^{m-1}\frac{k\cdot 2^{39}}{N})\\&\sim 1-\exp(-\frac{m^2\cdot 2^{38}}{N})\end{aligned}$$

이상이 된다[3]. 2^{39} 비트 출력 후 내부 상태의 엔트로피

3) $x\ll 1$일 때 $1-x\sim\exp(-x)$이 성립함을 이용하여 근사시켰다. 실제 확률은 $1-\exp(-2^{38}\cdot m^2/N)$와 $1-\exp(2^{-39}\cdot m^2/N)$ 사이에 존재한다.

를 $N=122$라고 추정할 수 있으므로 $m=2^{42}$이면 높은 확률로 충돌쌍을 찾을 수 있다. 즉, 2^{40} 길이의 키 수열이 2^{42}개 주어지면 높은 확률로 충돌쌍을 찾을 수 있다.

지금까지 살펴본 MICKEY의 엔트로피 손실을 이용한 분석은 MICKEY에 대한 공격으로 간주되지 않을 수 있지만 이러한 성질이 바람직하지 않다는 것은 분명하다. 수집된 데이터를 이용하여 이 분석법을 실제로 적용하는 경우, 전체 키 수열을 저장할 필요는 없으며 일치하는 키 수열을 찾기 위해 마지막 부분을 저장하는 것으로 충분하다. 첫번째 공격 시나리오에서 실제 저장해야 하는 키 수열은 $C\cdot 2^{60.5}\sim 2^{68}$ 비트이다 (단, C는 내부 상태를 구분할 수 있을 정도 길이의 키 수열 비트 수로서 여기서는 160). 어떤 값을 데이터 복잡도 (data complexity)로 간주해야 할지는 명확하지 않다.

추가로, 구체적인 공격 시나리오에서는 사전에 키 수열을 생성한 후 공격 대상 온라인 데이터와 비교하는 방법을 생각해 볼 수 있다.

참고 1. $\overline{\mathrm{EL}}$ 대신 EL을 사용한 이유는 다음과 같다. 엔트로피 손실을 분석하기 위해 치역의 크기를 의미하는 $\overline{\mathrm{EL}}$을 고려하는 것이 더 적절하다고 생각할 지도 모른다. $\overline{\mathrm{EL}}$과 EL의 차이점은 치역의 원소가 나타날 확률이 고르지 않다는 데에서 비롯한다. 같은 크기의 집합이 주어졌을 때, 나타날 확률이 고르지 않을수록 충돌쌍을 찾기 쉽다. 따라서 치역의 분포가 분석에 영향을 미치며 $\overline{\mathrm{EL}}(f)$ 보다는 $\mathrm{EL}(f)$를 이용해 분석하는 것이 더 적절하다 할 수 있다.

참고 2. 본 소절의 내용을 통해, 일대일 함수가 아닌 랜덤 함수를 이용하여 스트림 암호의 상태를 갱신하도록 한다면 내부 상태의 크기가 원하는 안전도의 두 배보다 커야 한다는 것을 알 수 있다.

V. 결론

본 논문에서는 eSTREAM에 제안된 스트림 암호 MICKEY의 안전성에 대하여 논의하였다. 지금까지 언급한 MICKEY의 약점은 다음 두 가지로 요약할 수 있다.

(1) MICKEY에 대해 온라인 공격량이 전수조사 공격량 보다 적은 time-memory-data tradeoff 공격이 가능하다.사전 계산량은 전수조사 공격량보다 더 크지만 가까운 미래에 도달 가능할 것으로 여겨진다.

(2) 내부 상태가 갱신될 수록 내부 상태의 엔트로피가 줄어들어 다르게 시작한 키 수열이 나중에 같아질 수 있다.

첫 번째 약점은 내부 상태의 크기를 키우거나 더 복잡한 출력 필터를 사용하면 해결이 가능하다. 하지만 두 방법 모두 효율성을 저해하는 요인이 된다. 두 번째 약점은 MICKEY의 설계에 대한 보다 근본적인 문제점으로 볼 수 있으며 쉽게 고칠 수 없어 보인다.

위 약점들은 치명적인 것으로 간주되지 않을 수 있다. 하지만 MICKEY를 사용하려 한다면 반드시 위의 약점들을 고려해야 한다.

참고문헌

[1] ECRYPT, "ECRYPT yearly report on algorithms and keysizes (2004)". Version 1.1, March, 2005. Available from http://www.ecrypt.eu.org.

[2] ECRYPT, eSTREAM - the ECRYPT Stream Cipher Project. Available from http://www.ecrypt.eu.org/stream/.

[3] S. Babbage, "Improved exhaustive search attacks on stream ciphers". *European Convention on Security and Detection*, IEE Conference publication No. 408, pp. 161-166, IEE, 1995.

[4] S. Babbage and M. Dodd, "The stream cipher MICKEY (version 1)". *ECRYPT Stream Cipher Project Report 2005/015*, 2005.

[5] A. Biryukov and A. Shamir, "Cryptanalytic time/memory/data tradeoffs for stream ciphers". *Asiacrypt 2000*, LNCS 1976, pp. 1-13, Springer-Verlag, 2000.

[6] A. Biryukov, A. Shamir, and D. Wagner, "Real time cryptanalysis of A5/1 on a PC". *FSE 2000*, LNCS 1978, pp. 1-18, Springer-Verlag, 2001.

[7] P. Flajolet and A. Odlyzko, "Random mapping statistics". *Eurocrypt '89*, LNCS 434, pp. 329-354, Springer-Verlag, 1990.

[8] J. Golić, "Cryptanalysis of alleged A5 stream

cipher". *Eurocrypt '97*, LNCS 1233, pp. 239-255, Springer-Verlag, 1997.

[9] M. Hellman, "A cryptanalytic time-memory trade-off." *IEEE Trans. on Infor. Theory*, vol 26, pp. 401-406, 1980.

[10] A. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1997.

〈著者紹介〉

김 우 환 (Woo-Hwan Kim) 정회원

1998년 2월: 서울대학교 수학과 졸업
2000년 2월: 서울대학교 대학원 수학과 석사
2004년 8월: 서울대학교 대학원 수학과 박사
2004년 11월~현재: 국가보안기술연구소 연구원
<관심분야> 비밀키 암호 분석, 공개키 암호 프로토콜

홍 진 (Jin Hong) 종신회원

1994년 2월: 서울대학교 수학과 졸업
1996년 2월: 서울대학교 대학원 수학과 석사
2000년 8월: 서울대학교 대학원 수학과 박사
2000년 9월~2002년 9월: 고등과학원 연구원
2002년 9월~2006년 8월: 국가보안기술연구소 선임연구원
2006년 9월~현재: 서울대학교 수리과학부 조교수
<관심분야> 비밀키 암호 설계 및 분석