情報保護學會論文誌
第17卷 第3號, 2007. 6

변형 Self-Shrinking 생성기에 대한 Guess-and-Determine 공격

이동훈,† 박제홍,‡ 한재우, 박상우
국가보안기술연구소

Guess-and-Determine Attack on the Variant of Self-Shrinking Generator

Dong Hoon Lee,† Je Hong Park,‡ Jaewoo Han, Sangwoo Park
National Security Research Institute

요 약

본 논문에서는 Self-Shrinking 생성기의 변형 구조 SSG-XOR에 대해 guess-and-determine 공격을 적용하여 실제 안전성을 분석하고 이에 대한 시뮬레이션 결과를 소개한다. SSG-XOR은 Self-Shrinking 생성기에 비해 더 좋은 암호학적 성질을 가지는 것으로 소개되었으나 본 논문의 분석 결과, guess-and-determine 공격 관점에서 Self-Shrinking 생성기에 비해 안전성이 많이 떨어지는 것을 확인할 수 있다.

ABSTRACT

In this paper, we analyse the security of the variant of Self-Shrinking generator proposed by Chang et al. against a guess-and-determine attack. This variant, which we call SSG-XOR is claimed to have better cryptographic properties than the Self-Shrinking generator in a practical setting. But we show that SSG-XOR is weaker than the Self-Shrinking generator from the viewpoint of guess-and-determine attack.

Keywords : *Self-Shrinking Generator, Stream Cipher, Guess-and-Determine Attack, Security Analysis*

Ⅰ. 서 론

Self-Shrinking 생성기(Self-Shrinking Generator, 이하 SSG)는 Meier와 Staffelbach가 제안한 논리로 Shrinking 생성기에서 사용하는 두개의 LFSR(Linear Feedback Shift Register)의 동작을 한개의 LFSR로 구현하고자 하는 발상을 기반으로 설계되었다[1]. SSG의 구조는, 우선 LFSR 2회 동작으로 얻어진 비트쌍 (a_{2i}, a_{2i+1})에서 $a_{2i} = 1$이면 a_{2i+1}을 키수열로 출력하고, 아니면 키수열 출력 없이 다시 LFSR을 2회 동작한다.

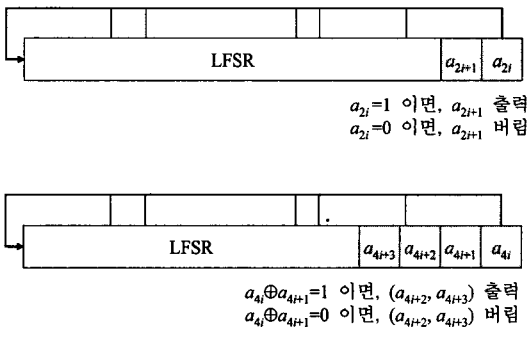
SSG의 설계자들은 기본적인 전수조사 공격과 엔트로피 공격에 대한 안전성을 제시하였는데, 특히 L을 LFSR의 길이라 할 때 SSG에 대한 엔트로피 공격의 시간복잡도가 $O(2^{0.75L})$임을 밝혔다[1]. 그러나 최근까지 제안된 여러 공격방법[2,3,4,5]들에 의해 SSG의 공격에 필요한 시간복잡도는 계속적으로 낮아지고 있다. 간략하게 살펴보면, 논문 [2]에서는 $O(2^{0.695L})$, 그리고 논문 [3]에서는 BDD(Binary Decision Diagram)-공격에 의해 $O(2^{0.656L})$로 점차 개선되었다. 이 두 공격은 많은 메모리 요구량에 의해 비현실적인 것으로 인식되었으나, 최근 논문 [4]에서는 BDD-공격과 거의 같은 시간복잡도를 가지면서 $O(L^2)$의 메모리를 사용하는 공격 방법이 제안되었다. 물론 스트림암호에 대한 가장 일반적인 공

접수일: 2007년 1월 26일; 채택일: 2007년 3월 23일
† 주저자: dlee@etri.re.kr
‡ 교신저자: jhpark@etri.re.kr

격방법인 TMTO(Time-Memory Trade-Off) 공격[6]의 경우 시간복잡도가 $O(2^{0.5L})$로 가장 낮지만, $O(2^{0.75L})$의 사전계산과 함께 $O(2^{0.25L})$-비트 키수열과 $O(2^{0.5L})$의 메모리가 필요하기 때문에 현실적인 공격이라고 볼 수는 없다. TMTO 공격의 시간복잡도에 가장 근접한 공격으로 최근 Zhang과 Feng에 의해 제안된 guess-and-determine 공격[5]은 $L \geq 100$일 때 $O(2^{0.161L})$-비트 키수열과 $O(L^2)$ 메모리를 사용하여 시간복잡도 $O(2^{0.556L})$로 SSG를 공격할 수 있고, $L<100$인 경우, 필요한 키수열이 $O(2^{0.194L})$-비트, 시간복잡도가 $O(2^{0.571L})$로 약간 증가한다. 이 공격은 TMTO 공격보다 시간복잡도가 약간 더 크지만, 사전계산이 필요없고 메모리 사용량이 적은 점을 감안할 때 현실적으로 SSG에 적용 가능한 가장 효율적인 방법으로 볼 수 있다.

최근 Chang 등이 제안한 XOR을 이용한 변형 Self-Shrinking 생성기(Self-Shrinking Generator with XOR, 이하 SSG-XOR)[7]는 SSG와 비슷한 방식으로 동작하지만, 두비트의 키수열을 한번에 출력하기 위해 LFSR을 4회 연속으로 동작하여 얻어진 4비트를 다룬다. 각각의 4비트 LFSR 출력 (a_{4i}, a_{4i+1}, a_{4i+2}, a_{4i+3})에서, SSG-XOR은 $a_{4i} \oplus a_{4i+1} = 1$이면 a_{4i+2}와 a_{4i+3}을 키수열로 출력하고, 아니면 키수열 출력 없이 다시 LFSR을 4회 동작한다. 다음 [그림 1]은 SSG와 SSG-XOR의 동작방식의 차이점을 보여준다.

SSG-XOR의 제안자들은 SSG에 대한 각종 공격방법들을 SSG-XOR에 적용한 결과를 토대로 SSG-XOR이 엔트로피 공격[1]이나 BDD-공격[2]과 같이 짧은 키수열을 사용하는 공격에 대해 SSG보다 안전함을 주장하였다. 그러나 본 논문에서는 SSG-XOR에 guess-and- determine 공격을 적용하여 SSG-XOR의 안전성이 SSG에 비해 크게 떨어짐을 보인다. 본 논문의 guess-and-determine 공격은 $O(2^{0.111L})$-비트 키수열과 메모리 $O(L^2)$를 사용하여 시간복잡도 $O(2^{0.384L})$로 SSG-XOR의 초기상태값(initial state)을 찾을 수 있다. 참고로 guess-and-determine 공격을 포함하여 본 논문에서 비교 대상으로 거론한 SSG에 대한 공격 방법들은 모두 LFSR을 정의하는 특성다항식은 공개되어 있다는 가정하에 비밀정보인 LFSR의 초기상태값 또는 동등한 상태값(equivalent state)을 찾는것을 목표로 한다. 이때 동등한 상태값은 주어진 특성다항식에 의해 설정된 초기상태값과 동일한 키수열을 출력하는 상태값을 의미한다.



[그림 1] Self-Shrinking 생성기와 SSG-XOR

본 논문의 구성은 다음과 같다. 먼저 Ⅱ절에서는 LFSR의 기본적인 성질을 살펴보고, SSG에 대한 guess-and-determine 공격에 대해 정리한다. 이어 Ⅲ절에서는 guess-and-determine 공격에 대한 SSG-XOR의 안전성을 분석하고 실제 시뮬레이션 결과를 소개한다. 마지막 Ⅳ절에서는 본 논문의 내용을 요약 정리한다.

Ⅱ. Guess-and-Determine 공격

본 절에서는 SSG에 대한 guess-and-determine 공격[5] 방식을 설명한다. 이를 위해 우선 기본적인 LFSR의 성질에 대해 간략하게 설명한다.

2.1. Guess-and-Determine 공격의 기반이 되는 LFSR의 성질

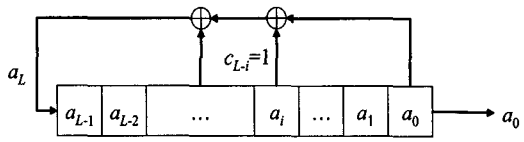
유한체 GF(2)에서 정의된 차수가 L인 다항식

$$f(x) = 1 + c_1x + c_2x^2 + \cdots + c_{L-1}x^{L-1} + x^L \in GF(2)[x]$$

가 LFSR을 정의한다는 것은, LFSR 출력비트 a_{L+i}가 다음과 같이 표현됨을 의미한다.

$$a_{L+i} = c_1a_{L+i-1} + c_2a_{L+i-2} + \cdots + c_{L-1}a_{i+1} + a_i. \quad (1)$$

위와 같이 LFSR을 정의하는 다항식 $f(x)$를 일반적으로 궤환다항식 또는 특성다항식 이라고 한다. 내부상태값 (a_i, a_{i+1}, $\cdots$, a_{L+i-1})을 가지고 있는 LFSR은 식 (1)을 통해 새로운 비트 a_{L+i}를 생성하고 내부상태값을 (a_{i+1}, a_{i+2}, $\cdots$, a_{L+i})로 갱신하면서 동시에 비트 a_i를 출력한다. 그러므로 최초의 내부상태값 (a_0, a_1, $\cdots$, a_{L-1})과 식 (1)에 의해 LFSR 출력수열이 결정된다. 아래 [그림 2]는



〔그림 2〕 LFSR의 동작방식

초기상태값이 $(a_0, a_1, \cdots, a_{L-1})$일때, LFSR의 동작방식을 보여준다.

특성다항식은 LFSR의 주기와 연관성을 가지고 있다. 특성다항식 $f(x)$가 $x^r - 1$을 나누는 최소의 r을 $f(x)$의 order라 하고 ord($f(x)$)라 표기하면, LFSR의 출력수열의 주기는 일반적으로 ord($f(x)$)의 약수가 된다. 만일 $f(x)$가 기약이면 출력수열의 주기는 정확하게 ord($f(x)$)이며 $2^L - 1$의 약수이다. 또한 특성다항식이 원시일 경우, LFSR의 주기는 $2^L - 1$로 최대가 된다. 본 논문의 나머지에서는 원시 특성다항식 $f(x)$에 의해 정의된 LFSR만을 고려한다.

한편 다항식 $f(x)$에 대한 reciprocal $f^*(x)$은 다음과 같이 정의한다.

$$f(x) = 1 + c_1x + c_2x_2 + \cdots + c_{L-1}x^{L-1} + x_L$$
$$f^*(x) = x_L + c_1x_{L-1} + c_2x_{L-2} + \cdots + c_{L-1}x + 1$$
$$= x^Lf(1/x).$$

그러면 다음과 같은 연관성에 의해, LFSR의 출력 a_i는 $x^i \bmod f^*(x)$와 대응시킬 수 있음을 쉽게 알 수 있다.

$$a_L = c_1a_{L-1} + c_2a_{L-2} + \cdots + c_{L-1}a_1 + a_0$$
$$x^L = c_1x^{L-1} + c_2x^{L-2} + \cdots + c_{L-1}x + 1 \bmod f^*(x). \quad (2)$$

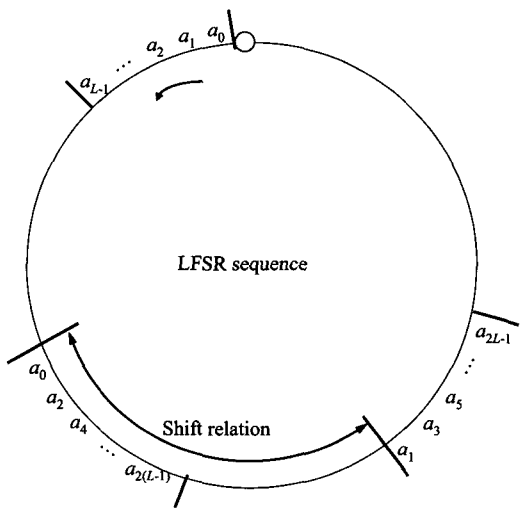
식 (2)의 제곱은 표수(characteristic) 2인 유한체 상에서 제곱 연산의 성질에 의해 다음과 같이 표현된다.

$$x^{2L} = c_1x^{2(L-1)} + c_2x^{2(L-2)} + \cdots + c_{L-1}x^2 + 1 \bmod f^*(x)$$

그러면, a_i는 $x^i \bmod f^*(x)$와 대응된다는 사실로부터 다음을 알 수 있다.

$$a_{2L} = c_1a_{2(L-1)} + c_2a_{2(L-2)} + \cdots + c_{L-1}a_2 + a_0.$$

즉, 수열 b = $\{a_{2i}\}$는 원래의 수열 a = $\{a_i\}$와 같은 특성다항식을 공유하는 수열임을 알 수 있다. 또한 식 (3)에



〔그림 3〕 LFSR 출력수열 a를 기준으로 수열 b와 c사이의 관계 표현

x를 곱한 식으로부터 다음을 알 수 있다.

$$a_{2L+1} = c_1a_{2(L-1)+1} + c_2a_{2(L-2)+1} + \cdots + c_{L-1}a_3 + a_1.$$

그러므로 c = $\{a_{2i+1}\}$ 또한 수열 a와 b를 생성하는 동일한 LFSR에 의해 생성되는 수열로 그 초기상태값만 $(a_1, a_3, \cdots, a_{2L-1})$로 다르다는 것을 알 수 있다. 다음 〔그림 3〕은 LFSR 출력수열 a를 기준으로 수열 b, c사이의 관계를 보여준다.

식 (2)를 제곱뿐만 아니라, 4제곱이나 일반적인 2^k제곱을 하는 경우에도 위에서 설명한 것과 같은 논리에 의하여 각각의 수열 $\{a_{4i}\}$나 $\{a_{2^k i}\}$가 역시 초기상태값만 다를 뿐, 원래의 수열 a와 같은 LFSR로 생성됨을 알 수 있다. 다음 정리 1은 위 〔그림 3〕에서 보이는 수열 b와 c사이의 정확한 간격을 제시한다.

정리 1. 길이가 L인 LFSR에서 출력된 최대 길이를 가지는 이진수열 a = $(a_0, a_1, \cdots)$가 있을 때, 두 수열 b = $\{b_i\ (= a_{2i})\}$와 c = $\{c_i\ (= a_{2i+1})\}$의 간격은 2^{L-1}이다. 즉, 각 $i \geq 0$에 대해, $b_i = c_{i+2^{L-1}}$이 성립한다.

증명: 다음의 식으로부터 자명하다.

$$a_{2i+1} = a_{2i+1+(2^L-1)} = a_{2(i+2^{L-1})}$$

2.2. Self-Shrinking 생성기에 대한 Guess-and-Determine 공격

정리 1에 의해, a_{2j+1}은 $(a_{2j}, a_{2(j+1)}, \cdots, a_{2(j+L-1)})$가 초기상태값인 수열 $b = \{b_i (= a_{2(i+j)})\}$의 2^{L-1}번째 원소임을 알 수 있다. 또한 LFSR 출력과 reciprocal $f^*(x)$와의 관계를 이용하면 2^{L-1}번째 원소는 $x^{2^{L-1}} \bmod f^*(x)$에 대응된다는 사실은 이미 앞에서 설명하였다.

따라서 a_{2j+1}을 다음과 같은 방법으로 구할 수 있다. $h(x) = x^{2^{L-1}} \bmod f^*(x)$는 (거듭제곱을 $L-1$번 반복함으로써) 쉽게 계산이 가능한 식으로 다음과 같이 표현할 수 있다.

$$h(x) = h_{L-1}x^{L-1} + h_{L-2}x^{L-2} + \cdots + h_1x + h_0. \quad (3)$$

그러면 $(b_0, b_1, \cdots, b_{L-1})$을 초기상태값으로 하는 LFSR의 출력에서 2^{L-1}번째 원소는 다음과 같다.

$$b_{2^{L-1}} = h_{L-1}b_{L-1} + \cdots + h_1b_1 + h_0b_0.$$

여기에서 $b_i = a_{2(i+j)}$이므로, $b_{2^{L-1}}$이 바로 a_{2j+1}이다. 즉 a_{2j+1}은 $(a_{2j}, a_{2(j+1)}, \cdots, a_{2(j+L-1)})$를 이용한 선형식으로 표현 가능함을 알 수 있다. 이러한 LFSR의 성질을 이용하여, SSG에 대한 guess-and-determine 공격은 $(a_0, a_2, \cdots, a_{2(L-1)})$을 찾은 후 단순한 선형대수를 통해 초기상태값 $(a_0, a_1, \cdots, a_{L-1})$를 찾는다.

실제 SSG에 대한 guess-and-determine 공격은 수열 $\{a_{2i}\}$의 초기상태값 $(a_0, a_2, \cdots, a_{2(L-1)})$의 처음 l 비트 guess = $(a_0, a_2, \cdots, a_{2(l-1)})$를 추정한 후 나머지를 결정한다. SSG의 구조에 의해 guess와 키수열 $z = \{z_i\}$로부터 다음의 관계가 성립함을 알 수 있다.

$$a_{2j} = 1 \Rightarrow a_{2j+1} = z_{\sum_{i=0}^{j-1} a_{2i}}. \quad (0 \le j \le l-1)$$

이때, a_{2j+1}는 앞에서 설명한 바와 같이 $(a_{2j}, a_{2(j+1)}, \cdots, a_{2(j+L-1)})$를 이용한 선형식으로 표현 가능하고, 또한 $(a_{2j}, a_{2(j+1)}, \cdots, a_{2(j+L-1)})$는 초기상태값이 $(a_0, a_2, \cdots, a_{2(L-1)})$인 LFSR의 출력이므로, a_{2j+1}는 $(a_0, a_2, \cdots, a_{2(L-1)})$의 선형식으로 표현 가능하다. 여기서 $a_0, a_2, \cdots, a_{2(l-1)}$은 공격자가 처음에 guess로서 추측한 값이므로, 나머지 $L-l$ 개의 변수를 가지는 선형방정식을 guess의 Hamming weight H_w(guess)개 만큼 얻을 수 있다. 그러나 효율적인 공격을 위해서 실제로는 모든 l 비트의 guess를 고려하는 것이 아니라, 공격 파라미터 α ($0.5 \le \alpha \le 1$)를 정한 후 H_w(guess)$\ge \lfloor \alpha l \rfloor$ 을 만족하는 guess만 선택해서 다룬다. 이는 충분한 Hamming weight를 가지는 guess를 다루어야 선형방정식을 충분히 얻어서 초기상태값을 정확히 구할 수 있기 때문이다. 이러한 guess를 모아둔 집합을 $G = \{\text{guess} \mid H_w(\text{guess}) \ge \lfloor \alpha l \rfloor\}$, 그리고 $a_0 = 1\}$라 하고, G의 크기 $|G| = 2^{\beta l}$이라고 하자. 그러면 집합 G의 원소를 guess로 선택하여 얻어진 αl개의 선형방정식은 거의 선형 독립이다[5]. 따라서, $O(\alpha l) = L-l$ 이면 나머지 초기상태값을 결정할 수 있고, 그러면 공격자가 추측해야 할 guess의 길이 l 은 $O(L/(\alpha+1))$임을 알 수 있다.

공격자가 키수열 $z = \{z_i\}$의 처음 N 비트를 알고 있다고 가정할 때, 공격 시나리오는 다음과 같다.

1. 다항식 $h(x) = x^{2^{L-1}} \bmod f^*(x)$를 계산 (식 (3))하고 $j = 0$으로 초기화
2. 집합 G의 원소 guess를 선택하고 초기상태값 찾기
   2.1. 선형방정식을 $L-l$ 개 만드는데, 키수열을 z_j 부터 사용함
   2.2. 선형 연립방정식의 해를 구해 초기상태값의 후보로 함
   2.3 SSG의 결과값과 실제 키수열 $z_j, z_{j+1}, \cdots$이 일치하는 지 확인. 맞으면 초기상태값으로 반환하고 공격 종료, 틀리면 j 를 1 증가시키고 단계 2.1부터 반복
3. 단계 2부터 반복

즉, 집합 G의 각 원소에 대해 키수열의 시작부분을 하나씩 증가시키면서 만들어지는 선형 연립방정식의 해를 찾아서 키수열과의 비교를 통해 실제 사용된 초기상태값인지 확인하는 것으로, 결국 각각의 선택된 guess에 대해 약 $N-L$번의 시도가 가능하다. 이때, 공격이 한 번 이상 성공할 확률이 1/2이 되도록 하려면 $|G| \times (N-L) \ge 2^{l-1}$ 을 만족해야 한다.

정리 2. SSG에 대한 guess-and-determine 공격은 $O(2^{(1-\beta)L/(1+\alpha)})$-비트의 키수열과 메모리 $O(L^2)$를 사용하여 시간복잡도 $O(L^3 \cdot 2^{(L/(1+\alpha))})$로 LFSR의 초기상태값을 찾을 수 있다. 이때 L은 SSG에서 사용하는 LFSR의 길이, $0.5 \le \alpha$

≤1이고, β는 G의 크기를 나타내는 파라미터로서 α에 의해 결정된다.

정리 2의 자세한 분석에 대해서는 논문 [5]를 참고하기 바란다. 다음 절에서는 SSG-XOR에 대한 guess-and-determine 공격의 계산복잡도를 분석한다.

Ⅲ. SSG-XOR에 대한 안전성 분석

앞 절에서 다룬 SSG에 대한 guess-and-determine 공격을 이제 SSG-XOR에 적용해 보자. 우선 SSG-XOR에서 사용하는 LFSR을 정의하는 원시 특성다항식을 $f(x) = 1 + c_1x + c_2x_2 + \cdots + c_{L-1}x^{L-1} + x^L$라 하자. 앞선 정리 1과 마찬가지로 SSG-XOR에 대해서 다음의 관계가 성립한다.

정리 3. 길이가 L인 LFSR에 의해 생성된 최대 길이의 이진수열을 a = $(a_0, a_1, \cdots)$라 하자. 그러면 다음 수열 $s^{(0)} = \{a_{4j}\}$, $s^{(1)} = \{a_{4j+1}\}$, $s^{(2)} = \{a_{4j+2}\}$, $s^{(3)} = \{a_{4j+3}\}$들은 a와 같은 특성다항식을 공유하는 LFSR의 출력으로 볼 수 있으며, 각 $i = 0, 1, 2$에 대해 $s^{(i)}$와 $s^{(i+1)}$의 간격은 2^{L-2}이다.

증명: 앞서 언급한 바와 같이, 각 수열 $s^{(i)} = \{s_j^{(i)}\}$는 a와 같은 특성다항식을 공유한다. 그러므로 각 i에 대해 $s^{(i)}$와 $s^{(i+1)}$은 일정한 간격을 두고 떨어져 있다. 이때 두 수열 사이의 간격은 다음 식으로부터 알 수 있다.

$$s^{(i)}_{j+2^{L-2}} = a_{4(j+2^{L-2})+i} = a_{4j+2^L+i}$$
$$= a_{4j+(i+1)+(2^L-1)} = s_j^{(i+1)}.$$

각 $i = 1, 2, 3$에 대해 다항식 $h_i(x) = x^{(i \cdot 2^{L-2})} \bmod f^*(x)$를 다음과 같이 표시하자.

$$h_1(x)=\sum_{i=0}^{L-1} h_{1,i}\, x^i,\ h_2(x)=\sum_{i=0}^{L-1} h_{2,i}\, x^i,\ h_3(x)=\sum_{i=0}^{L-1} h_{3,i}\, x^i. \tag{4}$$

그러면 다음의 관계식을 얻을 수 있다.

$$a_{4i+1} = \sum_{j=0}^{L-1} h_{1,j}\ a_{4(i+j)},$$
$$a_{4i+2} = \sum_{j=0}^{L-1} h_{2,j}\ a_{4(i+j)},\ a_{4i+3} = \sum_{j=0}^{L-1} h_{3,j}\ a_{4(i+j)} \tag{5}$$

따라서, 수열 $\{a_{4i}\}$와 다항식 $h_1(x)$, $h_2(x)$, $h_3(x)$을 이용하면 수열 $\{a_{4i+1}\}$, $\{a_{4i+2}\}$, $\{a_{4i+3}\}$을 구할 수 있다.

공격자에게 노출된 길이가 N인 SSG-XOR의 키수열을 z = $\{z_i\}$라 하자. 먼저 L개의 변수들의 집합을 A = $(a_0, a_4, \cdots, a_{4(L-1)})$라 하면, SSG-XOR을 공격하기 위해서는 이 변수들을 찾는 것으로 충분하다. 하지만 A의 처음 l 비트를 직접 추측하는 방식의 SSG에 대한 공격과는 다르게, SSG-XOR의 경우 $(a_0 \oplus a_1, a_4 \oplus a_5, \cdots, a_{4(l-1)} \oplus a_{4(l-1)+1})$에 대응하는 l 비트 길이의 guess = $(g_0, g_1, \cdots, g_{l-1})$를 추측한다. 그러면 우리는 식 (5)로부터 다음과 같은 L개의 변수로 이루어진 $l + 2H_w$(guess)개의 선형방정식을 얻을 수 있다.

$$a_{4i} \oplus a_{4i+1} = a_{4i} \oplus \sum_{j=0}^{L-1} h_{1,j}\ a_{4(i+j)} = g_i \ (0 \le i < 1)$$

$$a_{4i+2} = \sum_{j=0}^{L-1} h_{2,j}\ a_{4(i+j)} = z_{2(\sum_{j=0}^{i} g_j)-2} \quad (g_i = 1)$$

$$a_{4i+3} = \sum_{j=0}^{L-1} h_{3,j}\ a_{4(i+j)} = z_{2(\sum_{j=0}^{i} g_j)-1} \quad (g_i = 1)$$

이러한 선형방정식들의 공통해를 구할 수 있다면, SSG-XOR에 사용된 LFSR의 초기상태값을 찾을 수 있다. 이를 위해서, 가능한 한 많은 선형다항식을 찾을 필요가 있다. 그러므로 SSG에 대한 공격과 마찬가지로, 모든 가능한 경우를 전수조사 하는 대신에 H_w(guess) $\ge \lfloor \alpha l \rfloor$ 을 만족하는 guess에 한정해서 공격을 수행한다. 이때 α $(0.5 \le \alpha \le 1)$는 실제 공격과정에서 결정할 파라미터이다.

논문 [5]에서는 위의 과정에서 얻어지는 선형방정식들이 거의 선형 독립이라 언급하고 있다. 그러므로 우리는 L개의 변수로부터 $O(1+2\alpha l)$개의 선형 독립인 방정식을 얻게된다. 이러한 관계식으로부터 구할 수 있는 guess의 길이 l 은 $O(L/(1+2\alpha))$이다. SSG-XOR에 대한 공격 시나리오는 다음과 같다.

1. 각각의 $i = 1, 2, 3$에 대해, 먼저 다항식 $h_i(x)$ =

$x^{(i \cdot 2^{L-2})} \bmod f^*(x)$ (식(4))를 계산하고 $j = 0$으로 초기화

2. 주어진 파라미터 α에 대해, $H_w(\text{guess}) \geq \lfloor \alpha l \rfloor$를 만족하는 guess를 추측한 후, 키수열값을 넣을 상수항 부분을 채우지 않은 채 L개의 변수로 이루어진 선형 표현(linear expression)을 구성하여 행렬 U에 저장
   2.1. 각 j $(0 \leq j \leq N-1-(1+2\lfloor \alpha l \rfloor))$에 대해, z_j에서 시작하는 키수열을 사용하여 행렬 U를 구성하는 선형 표현으로부터 상수항을 채운 선형 연립방정식을 구성
   2.2. 선형 연립방정식의 해를 구하여 초기상태값의 후보로 지정
   2.3. 이 후보값을 초기상태값으로 하여 SSG-XOR을 돌려서 나오는 키수열과 원래의 키수열 $\{z_i\}_{i \geq j}$을 비교하여 후보값이 옳은지 검증. 검증이 성공할 경우, 공격을 중지하고 찾은 값을 SSG-XOR의 초기상태값으로 반환. 검증이 실패할 경우, j를 증가시키고 단계 2.1로 이동
3. 만일 선택한 guess에서 초기상태값을 찾을 수 없을 경우, 다른 guess를 임의로 선택하고 단계 2에서부터 다시 공격을 시도

이제 위 공격이 성공하기 위해 필요한 키수열의 길이를 결정하자. 공격자가 추측하는 guess는 결국 다음 집합의 원소로 볼 수 있다.

$$G = \{\text{guess} \mid \lfloor \alpha l \rfloor \leq H_w(\text{guess}) \leq l, \text{ 그리고 } g_0 = 1\}$$

그러면 G의 원소의 개수는 다음과 같다.

$$|G| = \sum_{i=\lfloor \alpha l \rfloor - 1}^{l-1} \binom{l-1}{i}.$$

이 때, G의 크기를 $|G| = 2^{\beta l}$이라고 하자. 각각의 l개의 비트로 이루어진 guess에 대해, 공격자는 초기상태값을 찾기 위해 $N-L$번의 공격을 시도한다. 이 과정에서 공격자는 초기상태값을 찾아야 하므로 $|G| \times (N-L) \geq 2^{l-1}$를 만족해야 하고, 이는 SSG-XOR의 공격에 필요한 키수열의 길이 $O(2^{(1-\beta)L/(1+2\alpha)})$을 결정한다. 또한 공격에 필요한 시간복잡도는 최악의 경우 다음과 같다.

$$O(L^3)O(N-L)O(2^{\beta l}) = O(L^3 \cdot 2^{(L/(1+2\alpha))}).$$

여기에서 $O(L^3)$는 크기 L인 선형 연립방정식의 해를 구하는데 필요한 복잡도를 반영한 것이다.

정리 4. SSG-XOR에 대한 guess-and-determine 공격은 $O(2^{(1-\beta)L/(1+2\alpha)})$-비트의 키수열과 메모리 $O(L^2)$를 사용하여 시간복잡도 $O(L^3 \cdot 2^{(L/(1+2\alpha))})$로 초기상태값을 찾을 수 있다. 이때 L은 SSG-XOR에서 사용하는 LFSR의 길이, $0.5 \leq \alpha \leq 1$이고, β는 G의 크기를 나타내는 파라미터로서 α에 의해 결정된다.

SSG와 SSG-XOR 사이의 비교 결과를 〔표 1〕에 제시하였다. 여기에서 시간복잡도에서 다항식 요소 L^3은 무시하였다.

SSG-XOR에 대한 공격방법은 SSG의 경우와 동일하지만, SSG에 비해 l의 크기가 작아졌다. 이는 추정값으로부터 얻을 수 있는 방정식의 개수가 SSG-XOR의 경우 더 많다는 사실에 기인한 것이다. 그러므로 공격에 드는 시간복잡도 또한 SSG-XOR이 훨씬 낮기 때문에 SSG에 비해 공격하기 쉬워진다.

저자들은 본 논문에서 제시한 공격의 실효성을 확인하기 위해 일반 PC에서 C 언어로 작성한 코드를 통해 다양한 실험을 실시하였다. 길이가 30인 LFSR을 사용하는 경우를 예로 들어본다. LFSR의 특성다항식은 아래와 같이 선택하였다.

$$f(x) = x^{30} + x^{27} + x^{23} + x^{22} + x^{17} + x^{16} + x^{14} + x^{11} + x^3 + x + 1$$

〔표 1〕 LFSR의 길이가 100보다 큰 경우 SSG와 SSG-XOR을 공격하기 위해 필요한 시간복잡도, 메모리 양, 그리고 키수열 길이

알고리즘	공격량	(α, β)		
		(0.5, 0.99)	(0.8, 0.71)	(1.0, 0)
SSG	시간복잡도	$O(2^{0.667L})$	$O(2^{0.556L})$	$O(2^{0.5L})$
	메모리	$O(L^2)$	$O(L^2)$	$O(L^2)$
	키수열	$O(2^{0.007L})$	$O(2^{0.161L})$	$O(2^{0.5L})$
SSG-XOR	시간복잡도	$O(2^{0.5L})$	$O(2^{0.384L})$	$O(2^{0.333L})$
	메모리	$O(L^2)$	$O(L^2)$	$O(L^2)$
	키수열	$O(2^{0.005L})$	$O(2^{0.111L})$	$O(2^{0.333L})$

여기에서 $f(x)$는 원시다항식이다. 그러므로 생성되는 수열은 최대 길이를 가진다. 그러면 $h_1(x)$, $h_2(x)$, 그리고 $h_3(x)$는 다음과 같이 계산할 수 있다.

$$\begin{aligned} h_1(x) &= x^{2^{28}} \bmod f^*(x) \\ &= x_{28} + x_{26} + x_{21} + x_{20} + x_{15} + x_{11} + x_9 + x_8 \\ &\quad + x_7 + x_6 + x_4 + x_2 + 1 \\ h_2(x) &= x^{2^{29}} \bmod f^*(x) \\ &= x_{29} + x_{28} + x_{27} + x_{21} + x_{17} + x_{16} + x_{15} \\ &\quad + x_{14} + x_{13} + x_{12} + x_{11} + x_8 + x_7 + x_4 \\ h_3(x) &= x^{3 \cdot 2^{28}} \bmod f^*(x) \\ &= x_{27} + x_{24} + x_{23} + x_{17} + x_{14} + x_{11} + x_{10} \\ &\quad + x_6 + x_5 + x_3 \end{aligned}$$

임의로 생성한 초기상태값에 대해, 본 논문에서 제안한 공격은 200비트 길이의 키수열로부터 1분 이내에 초기상태값을 찾는 것을 확인하였다.

Ⅳ. 결 론

본 논문에서는 논문 [7]에서 제안한 변형 Self-Shrinking 생성기에 대해 guess-and-determine 공격을 수행하여 Self-Shrinking 생성기에 비해 안전성이 크게 떨어지는 것을 확인하였다.

참고문헌

[1] W. Meier, O. Staffelbach, "The self-shrinking generator," Advances in Cryptology-EUROCRYPT '94, *Lecture Notes in Computer Science*, vol. 950, pp. 205-214, 1994.

[2] E. Zenner, M. Krause, S. Lucks, "Improved cryptanalysis of the self-shrinking generator," Information Security and Privacy-ACISP 2001, *Lecture Notes in Computer Science*, vol. 2119, pp. 21-35, 2001.

[3] M. Krause, "BDD-based cryptanalysis of keystream generator," Advances in Cryptology-EUROCRYPT 2002, *Lecture Notes in Computer Science*, vol. 2332, pp. 222-237, 2002.

[4] M. Hell, T. Johansson, "Two new attacks on the self-shrinking generator," *IEEE Transactions on Information Theory,* vol. 52, no. 8, pp. 3837-3843, 2006.

[5] B. Zhang, D. Feng, "New guess-and-determine attack on the self-shrinking generator," *IEEE Transactions on Information Theory,* vol. 52, no. 8, pp. 3837-3843, 2006.

[6] A. Biryukov, A. Shamir, "Cryptanalytic time/memory/data tradeoffs for stream ciphers," Advances in Cryptology - ASIACRYPT 2000, *Lecture Notes in Computer Science*, vol. 1976, pp. 1-13, 2000.

[7] K.-Y. Chang, J.-S. Kang, M.-K. Lee, H. Lee, D. Hong, "New variant of the self-shrinking generator and its cryptographic properties," Information Security and Cryptology - ICISC 2006, *Lecture Notes in Computer Science*, vol. 4296, pp. 41-50, 2006.

〈著者紹介〉

이 동 훈 (Dong Hoon Lee) 정회원
1994년 2월 : 서울대학교 수학교육과 졸업
1996년 2월 : 한국과학기술원 수학과 석사
2000년 2월 : 한국과학기술원 수학과 박사
2000년 2월~2002년 3월 : (주)퓨쳐시스템 선임연구원
2002년 4월~현재 : 국가보안기술연구소 선임연구원
<관심분야> 응용정수론, 암호론, 인터넷보안

박 제 홍 (Je Hong Park) 정회원
1998년 2월 : 경북대학교 수학과 졸업
2000년 2월 : 한국과학기술원 수학과 석사
2004년 2월 : 한국과학기술원 수학과 박사
2004년 3월~현재 : 국가보안기술연구소 연구원
<관심분야> 암호론, 응용정수론

한 재 우 (Jaewoo Han) 정회원
1991년 2월 : 서강대학교 수학과 졸업
1993년 2월 : 한국과학기술원 수학과 석사
1999년 8월 : 한국과학기술원 수학과 박사
1999년 7월~1999년 12월 : 한국전자통신연구원 선임연구원
2000년 1월~현재 : 국가보안기술연구소 선임연구원
<관심분야> 암호프로토콜, 스트림암호, 매듭이론

박 상 우 (Sangwoo Park) 정회원
1989년 2월 : 고려대학교 수학교육과 졸업
1991년 8월 : 고려대학교 수학과 석사
2003년 2월 : 고려대학교 수학과 박사
1991년 8월~1999년 12월 : 한국전자통신연구원 선임연구원
2000년 1월~현재 : 국가보안기술연구소 책임연구원
<관심분야> 암호론, 정보보호