# Esmaeili-Gulliver 암호시스템의 개선

이용우*, 김영식**, 노종선*

*서울대학교 전기정보공학부 뉴미디어통신공동연구소
**조선대학교 정보통신공학과

*yongwool@ccl.snu.ac.kr, **iamyskim@Chosun.ac.kr, *jsno@snu.ac.kr

# An Improvement of the Esmaeili-Gulliver Cryptosystem

Yongwoo Lee*, Young-Sik Kim**, Jong-Seon No*

*INMC, Department of ECE, Seoul National Univ.
**Department of ICE, Chosun Univ.

요 약

본 논문에서는 [2]에서 제안한 Esmaeili-Gulliver 암호시스템의 개선 방안을 제안한다. Esmaeili 와 Gulliver 가 제안한 것과 같은 파라미터를 사용하면 암호문에서 키와 평문에 대한 정보가 드러나는 것을 보이고, 적절한 파라미터를 제안한다. 또한, 암호문에 난수를 덧붙이기(padding) 하여, 암호문 길이를 이용한 공격을 무효화한다. 마지막으로, 채널 오류를 이용하는 대신에 별도의 오류를 추가하는 방법을 통해 보안성을 높인다.

## Ⅰ. 서 론

최근, Esmaeili 와 Gulliver 에 의해 새로운 부호 기반 대칭 키 암호 시스템이 제안되었다[2]. 이 암호 시스템에서는 선형 오류 정정 부호를 이용하여 평문을 부호화하고, 동기화된 난수 생성기를 이용, 정정할 수 없는 난수를 더한다. 또, 다른 난수를 이용하여 오류가 포함된 부호 워드(codeword)에 비트 삭제와 삽입을 한다. 복호화를 할 때는 이 동기화 된 난수 생성기를 이용해서, 삽입되거나 삭제된 비트를 복원하고 더해진 오류를 제거한 뒤, 오류 정정을 한다. 일반적인 부호 기반 암호와는 달리, 이 암호 시스템은 디코딩 문제(Decoding Problem)에 기반을 둔 암호가 아니다. 따라서 암호화에 사용하는 부호는 공개된다. 이 암호 시스템의 키는 동기화 된 두 개의 난수 생성기이다.

본 논문에서는 Esmaeili-Gulliver 암호 시스템에서 공격의 여지가 있는 약점들을 지적하고, 이를 보완하려는 방법을 제안한다. 안전한 파라미터들을 제안하고, 오류를 생성하는 방법을 변화시키며, 암호문에 난수를 덧붙이기 하여 보안성을 높인다.

## Ⅱ. Esmaeili-Gulliver 암호 시스템

### A. 암호화

Alice 와 Bob 은 통신하기에 앞서, $(n,k)$ 선형 부호(linear code)와 정수 $\beta$ 를 정한다. 이 암호시스템에서는, 일반적인 부호 기반 암호와 달리 오류 정정 부호는 공개된다. 그리고 두 개의 동기화 된 난수 생성기가 키로 사용된다. 각 난수 생성기는 $(n-k)$비트와 $\beta(1+\lfloor\log_2\frac{n}{\beta}\rfloor)$ 비트의 난수를 생성한다. 암호화 과정은 다음과 같다.

1. 평문을 $\boldsymbol{m}$이라고 하면, 이를 부호화해서 부호 워드 $\boldsymbol{c}$를 생성한다.
2. $\boldsymbol{c_e} = \boldsymbol{c} + \boldsymbol{s}(\mathbf{H^{-1}})^T$ 를 구한다. 여기에서 $\boldsymbol{s}$ 는 $(n-k)$ 비트의 난수이다. $\mathbf{H^{-1}}$ 은 선형 부호의 패리티 검사 행렬(parity check matrix)의 우-역 행렬(right inverse)이다. 패리티 행렬은 비가역 행렬이기 때문에, Alice 와 Bob 은 사전에 약속된 방법으로 $\mathbf{H^{-1}}$ 을 구한다. 이때 $\boldsymbol{s}(\mathbf{H^{-1}})^T$ 는 부호의 오류 정정 한계를 넘는 것이어야 한다. 만일 아니라면 $\boldsymbol{s}$를 버리고 다음 난수를 사용한다.
3. 마지막으로, $\boldsymbol{c_e}$ 과 $\beta(1+\lfloor\log_2\frac{n}{\beta}\rfloor)$ 비트의 난수를 각각 $\beta$ 개의 서브 블록으로 나눈다. 각 서브 블록들은 순서대로 대응된다. 난수의 서브 블록의 첫 번째 비트가 대응되는 $\boldsymbol{c_e}$ 의 서브 블록에 비트를 삽입/삭제하는 것을 결정하게 되는데, 0 이면 비트 삭제, 1 이면 비트 삽입이다. 삽입/삭제하는 위치는, 나머지 $\lfloor\log_2\frac{n}{\beta}\rfloor$ 개 비트가 가지는 값과 같다. 삽입/삭제 과정을 거친 암호문 $\boldsymbol{c'}$을 오류가 있는 채널을 통해 Bob 에게 전달한다.

### B. 복호화

Bob 도 Alice 와 같은 난수를 생성할 수 있다. 따라서 삽입된 비트를 삭제하고, $\boldsymbol{s}(\mathbf{H^{-1}})^T$를 계산하여 제거한 뒤 오류 정정을 하게 되면 평문을 구할 수 있다.

## Ⅲ. Esmaeili-Gulliver 암호 시스템 의 개선

*A.* $n$과 $\beta$값의 선택

비트를 삭제/삽입하는 과정을 보면, 암호문 서브 블록의 길이는 $n/\beta$인데 반해, $\lfloor \log_2 \frac{n}{\beta} \rfloor$개의 비트를 통해서 표현할 수 있는 수는 0부터 $2^{\lfloor \log_2 \frac{n}{\beta} \rfloor} - 1$까지이다. 따라서, 각 암호문 서브 블록의 맨 오른쪽 $\frac{n}{\beta} - 2^{\lfloor \log_2 \frac{n}{\beta} \rfloor}$개의 비트는 반드시 삽입이나 삭제가 되지 않는다. [2]에서 제안한 $n = 900, \beta = 20$ 을 적용할 경우, 각 서브 블록마다 13 개의 비트가 삽입 혹은 삭제가 되지 않는 것으로, 이는 총 암호문의 28%에 달한다.

삽입/삭제가 되지 않는 부분을 이용하여 공격할 수 있으므로, 삽입/삭제로부터 자유로운 비트가 생기게 하면 안 된다. 따라서 $\frac{n}{\beta}$는 2 의 제곱수가 되도록 $n$과 $\beta$를 선택하여야 한다. 다만, 최대 $\beta$개의 비트 삭제가 일어날 수 있으므로 $\beta$ 는 부호의 최소 거리(minimum distance)보다 작아야 한다. 예를 들어, [1]과 같이 (1024,524,50)부호를 사용할 경우 $\beta$는 32 가 적절하다.

*B.* $\mathrm{H}^{-1}$을 이용하지 않는 오류 생성

[2]에서 제안된 방법대로 $\mathbf{H}^{-1}$을 구하게 되면, $\mathbf{H}^{-1}$의 행 중 $k$ 개는 반드시 영벡터가 된다. 즉, $\boldsymbol{s}(\mathbf{H}^{-1})^T$ 는 정해진 위치에 최대 $(n-k)$개의 1 이 있고 나머지는 0 인 벡터이다. 이 때문에, 공격자는 추가된 오류의 위치를 알게 된다. 따라서 공격자는 높은 확률로 평문을 알아낼 수 있다. 일반적인 erasure decoding 을 할 때와 같이, 위치를 알고 있는 오류들에 대해 모두 0 을 대입하여 복호화해 보고, 모두 1 을 대입하여 복호화하는 방법을 암호문에 적용할 수 있다. 이때 해당 위치의 값들이 0 혹은 1 로 적절히 치우쳐 있다면 높은 확률로 복호화 된다. 예를 들어 [2]에서 제안된 것처럼 $n = 900, k = 810, \beta = 20$을 사용한다면, 채널 오류가 없다고 가정할 시에 복호화가 가능할 확률은 $1 - 1.5 \times 10^{-15}$ 로, 거의 1 에 가깝다. 이는 오류를 더해주고 비트를 삭제하거나 추가하는 과정이 암호 시스템에 큰 영향을 주지 못하는 것으로, [2]의 오류로 생각된다. 제안된 방법대로 $\mathbf{H}^{-1}$를 생성해서는 안 되며, 길이가 $n$ 인 난수 생성기를 사용하거나, 암호화 해시 함수 등을 사용해서 이러한 $\boldsymbol{s}(\mathbf{H}^{-1})^T$의 특성을 이용한 공격을 방지해야 한다.

*C.* 덧붙이기를 이용한 암호문 길이 조절

암호화 과정에서 암호문에 비트를 삽입하거나 삭제하는 관계로, 암호문의 길이는 최대 $n+\beta$에서 최소 $n-\beta$로 일정하지 않다. 이를 이용한 공격이 가능하다. 왜냐하면, 삭제된 비트의 수를 $a$, 제거된 비트의 수를 $d$, 암호문의 길이를 $|\boldsymbol{c}'|$이라 하면, 이들 사이에 $a + d = \beta$, $d - a = |\boldsymbol{c}'| - n$이라는 관계가 성립하기 때문이다. 이 관계를 통해 간단하게 난수의 특정 비트들에 대한 정보를 알 수 있다. 이렇게 정보를 누출하는 것은 바람직하지 않다. 따라서 암호문의 끝에 난수를 덧붙이기 하여 같은 길이로 만들어야 한다. 이때 덧붙이기에 사용하는 난수는 동기화할 필요가 없으므로 키 크기에 영향을 주지 않는다.

*D.* 채널 오류 대신 고의적인 오류 추가하기

Esmaeili 와 Gulliver 는 암호문을 별도의 오류 정정 부호를 사용하지 않고 오류가 있는 채널을 통해 송신하는 방법을 제안했다. 이러한 방법은 시스템 전체의 효율을 떨어트릴 확률이 있다. 왜냐하면, 고의로 비트를 삭제하여 오류 정정 부호가 정정할 수 있는 오류의 수를 줄이기 때문이다. 이로 인해서 복호화가 불가능하게 되거나 다른 값으로 복호화될 수도 있다. 만일 이러한 염려가 없을 정도로 안정적인 채널이라면 높은 확률로 매우 적은 채널 오류가 생성될 수 있다. 이 경우 *C* 절에서 언급한 것과 같은 공격이 가능해질 수 있다.

위와 같은 이유로 채널 오류를 이용하는 방법은 개선되어야 한다. 채널 오류를 이용하는 대신에, $\boldsymbol{s}(\mathbf{H}^{-1})^T$ 와는 별도로 정정이 가능한 오류를 추가하고, 암호문은 별도의 오류 정정 부호를 통해 송신하는 방법을 생각할 수 있다. 이렇게 되면 시스템의 안정성도 높아질 뿐 아니라 채널 오류가 발생하는 상황을 이용한 공격에 대해 저항성을 갖게 된다. 이때 추가하는 오류는 정정이 가능한 값이기 때문에 Alice 와 Bob 사이에 공유될 필요가 없다. 고로 키 크기에 영향을 주지 않는다.

## Ⅳ. 결론

Esmaeili-Gulliver 암호시스템은 작은 키 크기를 가지며, 암호화와 복호화도 선형 오류 정정부호의 부호화 복호화 정도로 가벼운 암호 시스템이다. 그러나 비트를 삽입/삭제하는 과정에서 의도하지 않게 정보가 누출될 수 있다. 본 논문에서는 암호문에 난수를 덧붙이기 하는 기법, 그리고 적절한 $n$과 $\beta$를 선택하는 방법을 통해 이러한 정보의 누출을 줄이는 방안을 제안한다. 또, 고의로 더하는 오류가 정해진 위치에만 발생하는 문제가 있어 이 문제를 고칠 수 있도록 더 큰 크기의 난수 생성기나 암호화 해시 함수 등을 이용하는 방안을 제시한다. 암호 시스템이 채널 오류를 사용 할 때는 암호문의 안정성이 낮아지고, 확률적으로 적은 수의 오류가 발생하는 것을 이용한 공격이 가능하다. 따라서 채널 오류를 이용하는 대신에 고의적인 오류를 추가하고 별도의 오류 정정부호를 통해 송신하는 방법을 제안한다.

## ACKNOWLEDGMENT

이 논문은 0000 년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (R-20160229-002941, IoT 및 클라우드 컴퓨팅을 위한 경량 포스트 양자 암호 시스템 연구)

## 참 고 문 헌

[1] R. J. McEliece, “A public-key cryptosystem based on algebraic coding theory,” DSN Progress Report, Jet Propulsion Laboratory, Pasadena, CA, pp. 114-116, Feb. 1978.

[2] M. Esmaeili and T. A. Gulliver, “A secure code based cryptosystem via random insertions, deletions, and errors,” IEEE Common. Lett., vol. 20, no. 5, pp. 870-873, May 2016.