情報保護學會論文誌
第12券 第2號, 2002. 4

SFLASH 안전성에 대한 분석

정 배 은*, 류 희 수*

On the security of SFLASH

Bae Eun Jung*, Heuisu Ryu*

요 약

SFLASH는 현재 유럽에서 진행 중인 NESSIE(New European Schemes for Signatures, Integrity, and Encryption) 프로젝트의 전자서명 후보가운데 하나로 제안된 서명 스킴이다. 비밀키의 요소 가운데 아핀 부분의 상수 행렬에 대한 공격은 [1]에서 논의된 바 있다. 본 논문에서는 SFLASH의 안전성이 비밀키의 요소가운데 서명 생성시 마지막 단계에 적용되는 아핀 사상의 선형 부분에 전적으로 의존함을 보인다. 이는 공격자가 비밀키의 요소 중 서명 생성시 마지막 단계에 적용되는 아핀 사상을 알면 다른 비밀 요소의 정보 없이도 임의의 메시지에 대한 위조 공격이 가능함을 보임으로써 증명한다. 또한, 공개키와 이미 알고있는 비밀 요소를 이용하여 다른 선형 부분의 부분 정보를 알아 낼 수 있음을 보이며, 이 결과를 바탕으로 키 길이의 효율성에 대하여 논의한다.

ABSTRACT

SFLASH, one of the asymmetric signature schemes in NESSIE project, was suggested and accepted in the first phase. In the latest, results about attacking the affine parts of SFLASH was published.[1] In this paper, we prove that an attacker knowing one linear part and two affine parts can easily forge signatures for arbitrary messages without information of the other linear part and the secret string. It follows that the security of SFLASH depends only on the linear part, which is used in the last step when a signature is being generated. Also, we show that an attacker can obtain partial information of the linear part by the forging method using known public key and secret elements and we discuss the length of secret key.

keyword : *전자서명, SFLASH*

Ⅰ. 서 론

J. Patarin, N. Courtois, L.Goubin은 MQ problem(MQ : Multivariate Quadratic equation)에 의존하는 전자 서명 스킴인 FLASH[2]를 개발하고, 이를 변형한 SFLASH와 함께 이들을 NESSIE 프로젝트에 제안하였었다. 1차 평가 결과, SFLASH가 후보로 남게되었는데, 이는 안전성 검증에 의한 결정이 아니라, 유사한 두 스킴 가운데, 키 길이 효율성 관점에서 선택된 것이었다. 1차 평가 당시까지만 하더라도 두 스킴에 대한 알려진 공격 알고리즘이 존재하지 않았다. 그러나, 최근 W. Geiselmann, R. Steinwandt, T. Beth는 SFLASH의 비밀키들의 원소들이 {0, 1}로 이루어진 점을 이용하여, 비밀키의 아핀 사상 가운데, 상수행렬에 대한 정보가 공개키로부터 유추되어 2^{11}개의 집합으로 축소됨을 보였다.[1,4] 이는 SFLASH가 완전히 안전하지 않음을 보여주는 것은 아니다. 하지만, 소인수분해 문제나 이산대수문제에 근거한 서명 스킴처럼 안전성에 대한 연구가 많이 되어 있지 못한 점을 감안하

* 한국전자통신연구원 정보보호연구본부{(bejung, hsryu)@etri.re.kr}

면, SFLASH가 취약할 수 있다는 의심을 갖게하는 요인이 될 수 있다.

SFLASH의 비밀키는 두 개의 일대일 아핀 사상과 하나의 80-bit 스트링으로 구성된다. [1]에서도 지적되었듯이 80-bit 스트링은 서명 검증과정에서 전혀 사용되지 않으므로, 이에 대한 정보 없이도 위조 서명 생성이 가능할 수 있다. 실제로 아핀 사상에 대한 정확한 정보만 갖고 있는 공격자는 임의의 80-bit 스트링을 사용하여 임의의 메시지에 대한 위조가 가능하다.

한편, W. Geiselmann, R. Steinwandt, T. Beth 들의 결과로부터 공격자가 두 개의 아핀 사상 가운데 상수행렬에 대한 정보를 갖고 있다고 가정하여도 무방하다. 우리는 이러한 상황에서, SFLASH의 안전성이 s의 선형사상 부분인 S_L에 전적으로 의존함을 알 수 있었다. S_L의 정보를 이용하여 T_L의 일부 정보를 계산할 수 있으며, 이 정보만으로도 임의의 메시지에 대한 유효한 서명을 쉽게 생성할 수 있음을 증명한다. 또한, 비밀 80-bit 스트링 뿐 아니라 비밀키 중 t의 일부가 비밀키로써 의미가 없음을 지적하고 이로부터 비밀키 길이의 효율성에 대하여 논의한다.

우리는 먼저 Ⅱ장에서 SFLASH의 구조와 서명 생성 및 검증 과정을 설명한다. 다음 Ⅲ장에서는 하나의 S_L에 대한 정보로부터 공격자가 서명을 생성하는 방법을 설명하고, 유효한 서명이 됨을 증명한다. 마지막으로, 얻은 결과를 바탕으로 SFLASH의 비밀키 길이의 효율성 및 안전성에 대하여 논의한다.

Ⅱ. SFLASH

2.1 SFLASH의 구조

본 절에서는 SFLASH에서 사용되는 두 개의 유한체 및 사용자의 비밀키와 공개키, 그리고 이에 대한 표기 등을 기술한다. 그 표기 방식은 NEESIE에 제안된 문서(참고문헌[3])를 따르기로 한다.

Notation :

- $K := F_2[X]/(X^7+X+1)$, 한체
- $\pi : \{0,1\}^7 \to K$

  $(b_0,\dots,b_6) \mapsto \sum_{i=0}^{6} b_i X^i \mod (X^7+X+1)$
- $K^* := \pi(\{0,1\}\times\{0\}^6)$
- $L := K[X]/f(X)$, 유한체

  $f(X) = X^{37}+X^{12}+X^{10}+X^2+1$
- $\varphi : K^{37} \to L$

  $(b_0,\dots,b_{36}) \mapsto \sum_{i=0}^{36} b_i X^i \ (\mathrm{mod}\ f(X))$
- $F : L \to L\ \alpha \mapsto \alpha^{128^{11}+1}$
- 이진 비트 스트링 $\lambda=(\lambda_0,\dots,\lambda_m)$과 $0\le r\le s$에 대하여 $[\lambda]_{r\to s}$은 $(\lambda_r,\dots,\lambda_s)$를 나타내기로 한다.
- 기호 $\|$ 은 두 비트 스트링의 연접을 나타낸다.

비밀키 : (Δ, s, t)

- Δ:80-bit 비밀 스트링
- $s=(S_L, S_C)$: K^{37}에서 정의되는 일대일 아핀사상으로써 37×37 행렬 $S_L \in K^{37\times37}$과 행벡터 $S_C \in K^{37}$로 구성된다
- $t=(T_L, T_C)$: K^{37}에서 정의되는 일대일 아핀 사상으로써 37×37 행렬 $T_L \in K^{37\times37}$과 행벡터 $T_C \in K^{37}$로 구성된다.

공개키 : $(P_i,\ i=0,\dots,25)$

공개키는 주어진 s, t와 F에 의해 결정되는 식으로써, K^{37}에서 정의되는 26개의 이차 다항식으로 주어지는데 다음과 같은 성질을 갖는다.

$$G : K^{37} \to K^{26}$$
$$G(X) = [t(\varphi^{-1}(F(\varphi(s(X)))))]_{0\to 181}$$

실제로 공개키는 다음과 같이 표현된다.

$$Y_i = P_i(X_0,\dots,X_{36}),\ i=0,\dots,25.$$

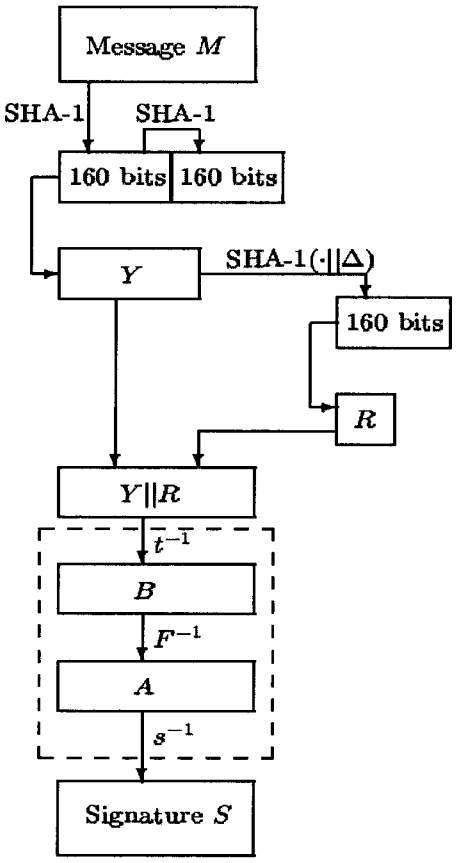
위에서 P_i는 모든 총 변수에 대하여 2차 이하이며 각 계수는 K의 원소로 표현된다.[3]

2.2 SFLASH의 서명 생성 과정과 검증 과정

이 절에서는 SFLASH의 서명 생성 및 검증 방식에 대하여 간단히 살펴본다.

2.2.1 서명 생성 과정

서명 생성 과정은 아래 그림과 같다. [그림1]에서 빗금으로 그려진 박스가 개인키의 비밀 요소 가운데 두 아핀 사상인 t와 s의 역사상 t^{-1}와 s^{-1}사용되는 부분이다. F^{-1}은 공개된 함수이다.



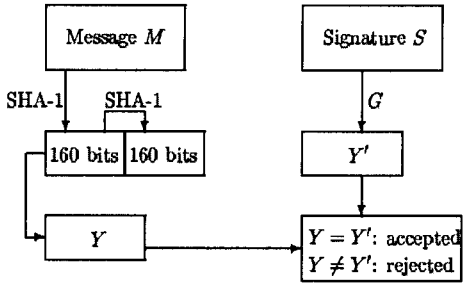
〔그림 1〕 서명 생성 알고리즘

위 그림을 간략히 설명하면 다음과 같다. 서명자는 주어진 메시지 M에 대하여 다음과 같이 서명을 생성한다.

(1) $M_1 = SHA-1(M)$, $M_2 = SHA-1(M_1)$
(2) $V=[M_1]_{0\to159} \| [M_2]_{0\to21}$
(3) $W= [SHA-1(V \| \Delta)]_{0\to76}$
(4) $Y = (\pi([V]_{0\to6}),\ldots,\pi([V]_{175\to181})) \in K^{26}$
(5) $R = (\pi([W]_{0\to6}),\ldots,\pi([W]_{70\to76})) \in K^{11}$
(6) $B = \varphi(t^{-1}(Y \| R)) \in L$ 계산
(7) $A = F^{-1}(B)$ 계산
(8) $X = (X_0,\ldots,X_{36}) = s^{-1}(\varphi^{-1}(A)) \in K^{37}$
(9) 서명 값 $S = \pi^{-1}(X_0) \| \ldots \| \pi^{-1}(X_{36})$을 계산

2.2.2 검증 과정

생성된 서명을 검증하는 방법은 아래 그림과 같다. 검증자는 메시지 M으로부터 Y를, 서명 값 S로부터 Y'을 계산하여 비교함으로써 검증을 수행한다.



〔그림 2〕 서명검증 알고리즘

위 그림을 간략히 설명하면 다음과 같다. 검증자는 서명 값 (M, S)에 대하여 다음의 순서로 검증한다.

(1) $M_1 = SHA-1(M)$, $M_2 = SHA-1(M_1)$
(2) $V = [M_1]_{0\to159} \| [M_2]_{0\to21}$
(3) $Y = (\pi([V]_{0\to6}),\ldots,\pi([V]_{175\to181})) \in K^{26}$
(4) $Y' = G((\pi([S]_{0\to6}),\ldots,\pi([S]_{175\to181}))$

Y와 Y'이 일치하면 서명검증이 성공한 것이고 일치하지 않으면 서명이 유효하지 않다고 판단한다.

Ⅲ. SFLASH 안전성

W. Geiselmann, R. Steinwandt, T. Beth는 SFLASH의 비밀 키들의 원소들이 {0, 1}로 이루어진 점을 이용하여, 비밀 키의 아핀 사상 가운데, 아핀 부분에 대한 정보가 공개키로부터 유추될 수 있음을 증명하였는데 앞 2.1절에 기술된 비밀키 가운데, $S_L^{-1} \cdot S_C$와 T_C가 2^{74} 집합에서 2^{11}의 후보 집합으로 축소 될 수 있다는 사실을 보인 바 있다.[1] 따라서, 이 두 정보는 공격자에게 노출되어 있다고 가정해도 무방하다. 따라서, 이 논문에서는 T_C와 $S_L^{-1} \cdot S_C$가 알려져 있다고 가정하기로 한다. 한편, 비밀키 요소 가운데, S_L과 T_L은 모두 37×37 크기의 행렬로써, 생성방법은 모두 독립적이다. 따라서, 안전성이 S_L과 T_L에 같은 비중으로 의존할 것으로 기대하게 된다. 그러나, 이제 우리는 SFLASH의 안전성이 비밀 요소 가운데, S_L에 전적으로 의존함을 보인다. 이는 비밀키 관리에 있어서, S_L에 대한 주의가 각별해야함을 시사한다.

3.1 위조 서명 생성

이 절에서는 공격자가 T_L에 대한 정보 없이도

공개키 G와 s, T_C를 이용하여 유효한 서명을 생성할 수 있음을 보인다.

[보조정리 1]

$H(X)$: $K^{37} \rightarrow K^{26}$ 인 함수로 다음과 같이 정의하자.

$$H(X) = [G(s^{-1}(\varphi^{-1}(F^{-1}(\varphi(X)))))] - [T_C]_{0 \rightarrow 181}$$

이 때,

(1) $H(X)$는 선형사상이 된다.

(2) 실제로, $H(X)$는 $[T_L(X)]_{0 \rightarrow 181}$과 동일하다.

[증명]

$t(\varphi^{-1}(F(\varphi(s(X)))))$
$= T_L(\varphi^{-1}(F(\varphi(s(X))))) + T_C$ 에서
$\varphi^{-1}(F(\varphi(s(X)))) = Y$라 하자.
그러면, $X = s^{-1}(\varphi(F^{-1}(\varphi(Y))))$.

한편, G의 정의로부터

$$[G(s^{-1}(\varphi^{-1}(F^{-1}(\varphi(X)))))] - [T_C]_{0 \rightarrow 181} = [T_L(X)]_{0 \rightarrow 181}$$

이 성립한다.

따라서, 보조정리 (1),(2)가 성립한다. ■

이제 e_i를 K^{37}의 원소로써 i번째만 1이고 나머지는 0인 단위 벡터라 하자. $H(X)$이 선형사상이므로 e_i, $1 \le i \le 37$에 대한 $H(e_i)$에 대한 이미지로 $H(X)$를 26×37 행렬로 표현할 수 있다.[5]

$H(e_i) = (a_{1,i}, \ldots, a_{26,i})$로 나타내기로 하자. 이러한 벡터들을 이용하여 37×37 행렬 T'_L을 다음과 같이 구성한다.

$$T'_L = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,35} & a_{1,36} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,36} & a_{2,37} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{26,1} & a_{26,2} & \cdots & a_{26,36} & a_{26,37} \\ b_{1,1} & b_{1,2} & \cdots & b_{1,36} & b_{1,37} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ b_{11,1} & b_{11,2} & \cdots & b_{11,36} & b_{11,37} \end{pmatrix} \in K^{37} \times K^{37}$$

위 행렬에서 $(a_{i,j})_{1 \le i \le 26, 1 \le j \le 37}$은 $H(e_i)$의 결과에서 얻는 값들이고, $(b_{i,j})_{1 \le i \le 11, 1 \le j \le 37}$은 T'_L이 invertibe 하도록 선택하는 임의의 값들이다. T_L이 invertible 하므로 보조정리 1.(2)로부터 $\{H(e_i), 1 \le i \le 37\}$이 서로 독립임을 알 수 있고, 따라서, invertible한 T'_L을 구성할 수 있다.

Note : 사용자 비밀키 $T_L = (c_{i,j}), 1 \le i, j \le 37$에서 보조정리 1.(2)로부터 $1 \le i \le 26, 1 \le j \le 37$에 대하여 $c_{i,j} = a_{i,j}$, 이 성립함을 알 수 있다. 즉, 위 방법은 T_L의 상위 26개의 행 벡터에 대한 정보를 얻는 방법을 설명한다.

[정리 2]

SFLASH의 안전성은 전적으로 S_L에 의존한다.

[증명]

S_L이 노출되었을 때, 먼저, 공격자는 공개키 G를 이용하여 임의의 메시지에 대한 유효한 서명을 생성할 수 있음을 보인다. 위에 기술한 방법으로 $T'_L \in K^{37} \times K^{37}$ 행렬을 구성한다. 메시지 M에 대하여 다음을 계산한다.

(1) $M_1 = SHA-1(M)$, $M_2 = SHA-1(M_1)$

(2) $V = [M_1]_{0 \rightarrow 159} \| [M_2]_{0 \rightarrow 21}$

(3) 임의의 77비트 스트링 R을 선택하여 $V' = V \| R$을 구성한다.

(4) $Z = (\pi([V']_{0 \rightarrow 6}), \ldots, \pi([V']_{252 \rightarrow 258}) \in K^{37}$

(5) $Z' = T'^{-1}_L (Z - T_C) \in K^{37}$

(6) $X = (X_0, \ldots, X_{36})$
$= s^{-1}(\phi^{-1}(F^{-1}(\phi(Z')))) \in K^{37}$

(7) $S = \pi^{-1}(X_0) \| \ldots \| \pi^{-1}(X_{36})$

[주장] 위에서 얻은 S는 유효한 서명이 된다.

[주장의 증명]

메시지 M에 대하여 검증 알고리즘에 대입하여 보자. 먼저 검증과정에서 $Y \in K^{26}$을 계산한다.

한편,
$Y' = G((\pi([S]_{0 \rightarrow 6}), \ldots, \pi([S]_{175 \rightarrow 181})))$에서
$Y' = G((\pi([S]_{0 \rightarrow 6}), \ldots, \pi([S]_{175 \rightarrow 181})))$
$= G(X)$
$= [t(\varphi^{-1}(F(\varphi(s(X)))))]_{0 \rightarrow 181}$
$= [t(Z')]_{0 \rightarrow 181}$

$$= [T'_L(Z') + T_C]_{0\to 181}$$
$$= [Z]_{0\to 181}$$
$$= (\pi(V_{0\to 6}), \ldots, \pi(V_{175\to 181}))$$
$$= Y$$ ■

따라서, s, T_C 및 공개키 G를 이용하여 임의의 메시지에 대한 유효한 서명 생성이 가능하다. 따라서 SFLASH의 안전성은 S_L에 의존한다.

한편, T_L 정보를 갖고 있는 공격자가 S_L에 대한 정보 없이 유효한 서명을 생성할 수 있다면, SFLASH의 안전성이 오직 S_L에만 의존한다고 할 수 없다.

이제, 아래에서 위 방법이 T_L, T_C, S_C을 아는 공격자에게는 적용될 수 없음을 보인다. 먼저, G의 정의로부터

$G(X) + [T_C]_{0\to 181} = [T_L(\varphi^{-1}(F(\varphi(s(X)))))]_{0\to 181}$

임을 알 수 있다. 이 식으로부터 $S_L(X)$ 혹은 $[S_L(X)]_{0\to 181}$로 표현될 수 있는 선형사상을 유도할 수 있는지 알아보자. 우변에서는 $[\ ]_{0\to 181}$을 취하기 전에 T_L^{-1}, F^{-1}을 적용할 수 있지만, 좌변에서는 $G(X)$가 이미 K^{26}의 원소이므로 T_L^{-1}, F^{-1}에 적용될 수 없다. 따라서, 위 식으로부터 $s(X)$와 관련되는 선형함수를 t, G, F 만으로는 찾는 것은 불가능하다.

따라서, 지금까지 알려진 방법으로는 T_L, T_C, S_C를 알아도, S_L에 대한 정확한 정보없이는 위조가 불가능하며, S_L의 정보를 얻는 것 또한 불가능하다. 그러므로, SFLASH의 안전성이 S_L에 전적으로 의존함을 알 수 있다. ■

Remark : 37×37 크기의 행렬 S_L과 T_L에 대한 제안자들이 제시한 생성 방법 가운데 하나가 "Trial and error" 방법으로 다음과 같다.

```
(Trial and error)
generate the matrix T_L by
 for (i=0 to 36)
 for (j=0 to 36)
T_L[i,j]=pi(next_random_bit,0,0,0,0,0,0)
until we obtain an invertible matrix.
```

위에서 pi함수는 랜덤 비트 생성 함수이다. 위에서 (i=0 to 36)을 (i=36 to 0)으로 생성하여도 무관하다.

3.2 SFLASH 안전성

암호 알고리즘의 안전성이란 현실적인 시간 안에 계산 가능한 공격 알고리즘이 있는가에 의존한다. 이 절에서는 3.1절에 기술한 방법을 적용할 때, 소요되는 시간에 대해 살펴보기로 한다. 아핀 사상 s와 T_C를 알고있는 공격자가 유효한 서명을 생성하기 위해서는 37개의 e_i에 대한 $H(e_i) = G(s^{-1}(\varphi^{-1}(F^{-1}(\varphi(e_i))))) - [T_C]_{0\to 181}$ 계산과 이 벡터들을 이용한 invertible한 행렬 T'_L을 구한 후, 임의의 80-bit 스트링(혹은 77-bit 스트링)을 이용하여 s와 $t' = (T'_L, T_C)$에 대한 서명 생성 알고리즘을 수행하면 되는 것이다. $\{H(e_i), 1\le i\le 37\}$를 찾는 계산은 37번의 서명과 검증을 수행하는 것보다 적게 걸린다. 한편 $\{H(e_i), 1\le i\le 37\}$으로부터 invertible한 T'_L을 만들기 위해서는 비밀키 생성 알고리즘에 사용되는 방법가운데 첫 행에서 26번째 행까지는 $\{H(e_i), 1\le i\le 37\}$로 정하고, 27번째 행부터 "Trial and error"방법을 사용하면 된다. 이 때 걸리는 시간은 일반적인 비밀키 생성보다 적게 걸린다. 이러한 t'을 찾으면 이 후부터는 같은 사용자의 위조 서명은 정당한 서명 생성 방식과 동일하게 적용된다. 따라서, 위에서 기술한 방법은 서명 키 생성 및 서명 생성 알고리즘과 검증 알고리즘을 수행하는 시간의 상수배 안에 계산되어 질 수 있다.

3.3 SFLASH 비밀키 길이의 효율성

3.1절에서 살펴본 위조 방법에서 비밀키 T_L은 s와 T_C가 알려지면 더 이상 비밀 정보가 될 수 없음을 알 수 있다. 또한 유효한 서명을 생성하는데 있어서, 비밀키 요소 중 행렬 T_L의 27번째 행부터 37번째 행까지의 정보와 80-bit 스트링 Δ는 의미가 없음을 알 수 있다. 따라서, 비밀키 생성시, T_L을 생성할 때, 하위 27번째부터 36번째까지 행 벡터를 e_i 등과 같이 고정시킨 후, "Trial and error"방법을 사용하여 행렬 T_L을 생성하고, 비밀키 T_L의 정보를, 26번째까지의 행 벡터로 제한할 것과 Δ의 사용 대신 서명 생성시 생성 알고리즘 (3)단계에서 W를 임의의 난수 77비트로 사용함으로써 비밀키의 크기를 줄이는 방안을 고려할 것을 제안하는 바이다.

Ⅳ. 결 론

우리는 이상에서 전자서명 SFLASH의 안전성이

S_L에 전적으로 의존함을 살펴보았다. S_L의 정보가 누출되면 공개키와 기타 유추할 수 있는 비밀 요소 T_C, S_C 등을 이용하여 유효한 서명을 생성하는 방법을 제시하였다. 위조 서명 생성에 있어 중요한 단계는 사용자의 알려지지 않은 비밀 정보의 일부 T'_L를 계산하는 과정이었으며 짧은 시간안에 가능함을 설명하였다. 이는 비밀키 t의 일부 정보와 Δ가 정당한 서명 생성에 대한 검증에 아무런 역할을 하지 못함을 의미하므로 비밀키에서 제외할 수도 있음을 논의하고, T_L의 생성에서 하위 11개의 행 벡터를 고정시켜 사용하고, Δ대신 임의의 난수 패딩을 사용함으로써 비밀키를 줄일 것을 제안하였다.

향 후, 프로토콜 공격이나 대수적 성질을 이용하여 S_L에 대한 정보를 유추할 수 있는 알고리즘에 대한 연구가 이루어진다면 이는 SFLASH의 강력한 공격 알고리즘이 될 것이다.

마지막으로, 본 논문에서 얻은 결과가 NESSIE 전자서명 후보인 SFLASH의 안전성에 대해 의문을 제기하는 단서가운데 하나가 될 수 있다고 사료된다.

참 고 문 헌

[1] W. Geiselmann, R. Steinwandt, and T. Beth, Attacking the Affine Parts of SFLASH, in Proceedings of Cryptography and Coding, LNCS 2260, Springer-Verlag, pp. 355~359, 2001.

[2] J. Patarin, N. Courtois, and L. Goubin, FLASH, a fast asymmetric signature scheme for low-cost smartcards. Primitive specification and supporting documentation, Presented at First Open NESSIE Workshop.

[3] J. Patarin, N. Courtois, and L. Goubin, SFLASH, a fast asymmetric signature scheme for low-cost smartcards. Primitive specification and supporting documentation, Presented at First Open NESSIE Workshop.

[4] F. Bao, R. H. Deng, W. Geiselmann, C. Schnorr, R. Steinwandt, and H. Wu, Cryptanalysis of Two Sparse Polynomial Based public Key Cryptosystems, in Proceedings of PKC 2001, K. Kim, ed., LNCS1992, Springer-Verlag, pp. 153~164, 2001.

[5] B. Jacob, Linear Algebra, University of Washington, W.H.Freeman and Company, 1990.

〈著者紹介〉

정 배 은 (Bae Eun Jung) 정회원
1993년 2월 : 서울대학교 수학교육과 학사
1995년 2월 : 서울대학교 수학과 석사
2000년 2월 : 서울대학교 수학과 박사
2000년 4월~현재 : 한국전자통신연구원 선임연구원
〈관심분야〉 암호이론, 이동통신 정보호호, 가환대수

류 희 수 (Heuisu Ryu) 정회원
1990년 2월 : 고려대학교 수학과 학사
1992년 2월 : 고려대학교 수학과 석사
1999년 5월 : Johns Hopkins University 수학과 박사
2000년 7월~현재 : 한국전자통신연구원 선임연구원
〈관심분야〉 정보보호, 타원곡선 암호, 이동통신 보안