情報保護學會論文誌
第14券 第2號, 2004. 4

정수의 분해를 이용한 ElGamal형 서명기법의 안전성 분석

이 익 권[a)†], 김 동 렬[b)‡]
인하대학교[a)], 한국정보보호진흥원[b)]

Security Analysis of ElGamal-Type Signature Scheme Using Integer Decomposition

Ikkwon Yie[a)], Dongryeol Kim[b)]
Inha University[a)], KISA[b)]

요 약

위수가 q인 생성원 g를 사용하는 ElGamal형 서명기법에서는 서명되어질 각 문서마다 message nonce를 구간 $(0, q-1)$에서 랜덤하게 선택해야 한다는 것은 잘 알려진 사실이다. H. Kuwakado와 H. Tanaka는 사용된 message nonce가 각각 $0 < k_1, k_2 \leq O(\sqrt{q})$인 서명 두 개가 주어졌을 때, 서명자의 비밀키를 다항식 시간으로 계산해내는 알고리즘을 제안하였다[2]. 최근 R. Gallant, R. Lambert, S. Vanstone등은 정수를 적절히 분해하여 타원곡선암호의 효율성을 개선하는 방법을 제안하였다[3]. 이 논문에서는 타원곡선암호의 고속연산에서 사용되었던 정수의 분해기법을 Kuwakado 등의 알고리즘에 적용하여 message nonce가 $|k_1|, |k_2| \leq O(\sqrt{q})$인 경우에도 적용할 수 있도록 확장하고, 알고리즘의 효율성 및 공격의 완성도를 개선하였다.

ABSTRACT

For an ElGamal-type signature scheme using a generator g of order q, it has been well-known that the message nonce should be chosen randomly in the interval $(0, q-1)$ for each message to be signed. In [2], H. Kuwakado and H. Tanaka proposed a polynomial time algorithm that gives the private key of the signer if two signatures with message nonces $0 < k_1, k_2 \leq O(\sqrt{q})$ are available. Recently, R. Gallant, R. Lambert, and S. Vanstone suggested a method to improve the efficiency of elliptic curve crytosystem using integer decomposition. In this paper, by applying the integer decomposition method to the algorithm proposed by Kuwakado and Tanaka, we extend the algorithm to work in the case when $|k_1|, |k_2| \leq O(\sqrt{q})$ and improve the efficiency and completeness of the algorithm.

Keywords: *Digital signature, Message nonce, Extended euclidean algorithm.*

접수일: 2003년 11월 5일; 채택일: 2004년 3월 8일
* 이 연구는 한국과학재단 논문연구과제(R05-2002-0001 360-0)의 지원으로 수행하였습니다.
† 주저자, ikyie@math.inha.ac.kr
‡ 교신저자, drkim@kisa.or.kr

Ⅰ. 서 론

T. ElGamal이 Crypto'84 학회에서 이산대수에 기반한 서명기법을 제안한 이후[1], 그의 기법으로부터 많은 변종들이 개발되고 또 표준서명기법으로 제정되었다.

서명기법은 키 생성 알고리즘, 서명 생성 알고리즘, 서명 검증 알고리즘의 세 개의 알고리즘으로 구성되는데, 대부분의 ElGamal 기법의 변종들은 서명 생성 알고리즘에 변형을 주어 만들어진다. 일반적으로 ElGamal형 서명기법의 서명 생성 알고리즘에는 다음과 같은 모양의 방정식이 관련되어 있다[5].

$$u \equiv xv + kw \pmod{q} \tag{1}$$

여기서 u, v, w는 생성된 서명의 일부로서 서명이 행해지면 공개되는 값들이고, x는 서명자의 비밀키, k는 서명 시 주어진 문서에 사용된 message nonce로서 어떤 종류의 공격자에게도 비밀로 지켜져야 할 값이다. 좀 더 구체적으로, 다음은 몇 가지 ElGamal형 서명기법에서 서명과 관련된 방정식들을 모은 것이다.

표 1. ElGamal 형식 서명기법의 서명방정식

서명기법	서명	서명방정식
ElGamal	$(M,(r,s))$	$H(M) \equiv xr + ks \pmod{q}$
DSA	$(M,(r,s))$	$H(M) \equiv x(-r) + ks \pmod{q}$
KCDSA	$(M,(r,s))$	$-(r \oplus (H(Y \parallel M) \bmod q)) \equiv xs - k \pmod{q}$
Schnorr	$(M,(r,s))$	$s \equiv xe + ks \pmod{q}$, 여기서 $e = H(M \parallel (g^k \bmod p))$

여기서 M은 서명되어질 문서의 내용이고 H는 해쉬함수이다. 또, KCDSA에서의 Y는 서명자의 공개키이다.

지금부터 앞으로 쓰여지는 문자 q는 고정된 소수를 의미한다.

ElGamal형 서명기법의 운용의 기본적인 요구사항 중 하나는 한 번 사용되었던 message nonce를 다시 사용하는 일이 없어야 한다는 것이다. 뿐만 아니라 message nonce k는 $1 \le k \le q-1$의 범위에서 랜덤하게 선택되어야 한다. H. Kuwakado와 H. Tanaka는 그들의 논문 [2]에서 만일 사용된 message nonce k_1, k_2가 $0 < k_1, k_2 \le O(\sqrt{q})$의 범위에 있는 것으로 알려진 두 개의 서명이 주어진다면 다항식 시간으로 서명자의 비밀키를 계산해내는 알고리즘을 발표하였다. 우리는 편의상 그들의 알고리즘을 KT알고리즘이라 부르기로 한다. 그들은 KT알고리즘의 시간복잡도가 $O((\log p)^3)$ 비트 연산이라고 주장하였다.

KT알고리즘이 주장된 것과 같은 계산 복잡도를 가진다는 것을 보이기 위해서는 알고리즘 수행 도중에 생성되는 벡터들 γ_1, γ_2의 크기가 특정한 경계 안에 들어온다는 것을 알아야 한다. 그러나 Kuwakado와 Tanaka는 이 경계에 대한 구체적인 언급은 없이 컴퓨터 시뮬레이션을 통해 얻어진 벡터들의 평균값만을 제시하고 있다.

이 논문에서 우리는 KT알고리즘의 중요한 한 단계를 개선하고자 한다. 최근에 Gallant 등이 타원곡선에서의 스칼라 배 연산을 가속시키기 위해 정수를 작은 계수를 가지는 형태로 분해하는 방법을 소개하였다[3]. 그들은 유클리드 알고리즘을 이용하여 두 개의 일차 독립인 벡터를 찾아냈는데, 그 중 첫째 벡터에 대해서는 그 길이가 항상 $\sqrt{q}$ 이하가 되도록 조절할 수 있었다. 우리는 Gallant 등의 방법을 이용하여 KT알고리즘에 소용될 적당한 벡터 (γ_1, γ_2)를 찾아내었다. 또한 이 과정에서 message nonce에 대한 조건을 완화시켜 $|k_1|, |k_2| \le O(\sqrt{q})$인 경우에도 적용할 수 있도록 하였다. 그리고 KT알고리즘에서는 가능한 최소의 크기를 갖는 벡터를 찾고자 하는데, 우리의 개선된 알고리즘으로 찾은 벡터의 크기가 평균적으로 KT알고리즘이 찾은 벡터의 크기의 60% 정도이며 (전체 알고리즘의 복잡도에 대한 영향은 미미하지만) 이 부분의 실행시간은 개선된 알고리즘이 KT알고리즘의 대략 1/40인 것을 실험적으로 확인하였다.

이 논문의 구성은 다음과 같다. 2절에서는 KT알고리즘을 간단히 소개한다. 3절에서는 Gallant 등이 사용한대로의 유클리드 알고리즘을 소개하고, 그들의 방법을 응용하여 KT알고리즘에 소용되는 짧은 벡터 (γ_1, γ_2)를 구체적인 범위 안에서 찾는 방법을 이야기한다. 그리고 4절에서는 이렇게 개선된 KT알고리즘의 효율성을 엄정하게 분석하였다. 마지막으로 5절에서는 본 논문의 결말을 지었다.

이 절을 마무리하기 전에 P. Nguyen과 I.

Shparlinski의 최근 논문[6]에 대해 언급할 필요가 있겠다. Nguyen과 Shparlinski는 이 논문에서, 적당한 수의 서명에서 사용된 nonce들에 대해서 몇 개의 연속한 비트들이 알려지면, 이 정보로부터 서명자의 비밀키를 계산해내는 다항식 시간 알고리즘을 선보였다. 좀 더 정확하게 말하자면, 그들의 알고리즘은 $\log q$ 에 선형으로 변하는 개수의 서명에서 각각의 nonce에 대해 대략 $(\log q)^{1/2}$개의 비트가 알려지면 다항식 시간에 실행된다. 하지만 그들은 오히려 부지수의 실행시간을 감수하면서라도 알아야하는 nonce의 비트 수를 줄이는데 더욱 초점을 맞추고 있다. 따라서 그들은 nonce들에 대해 단지 2~3 비트만이 알려졌다는 가정 아래 여러 가지 실험을 진행하였다. 한편, KT알고리즘이나 우리의 결과는 반대쪽으로의 극단으로, nonce들의 절반에 해당하는 비트들을 알아야하는 반면에 단지 두 개의 서명에 대해서만 이런 정보를 알면 충분하다. 그러므로 이 논문의 결과는 ElGamal 형식의 서명기법에서는 message nonce들이 '아주 random'하게 선택되어져야 하는 또 다른 이유를 제공하는 것이라 하겠다.

Ⅱ. KT알고리즘

이 절에서는 Kuwakado와 Tanaka가 [2]에서 제안한 KT알고리즘을 살펴보도록 하겠다.

우선 문서 m_1, m_2에 각각 message nonce k_1, k_2를 사용하여 생성된 두 개의 서명이 주어졌다고 가정하자. 그러면 식 (1)로부터 다음의 연립방정식을 얻게 된다.

$$u_2 \equiv xv_1 + k_1 w_1 \pmod q$$
$$u_2 \equiv xv_2 + k_2 w_2 \pmod q$$

여기서 x를 소거하면 미지수 k_1, k_2에 대한 부정방정식

$$u_1 v_2 - u_2 v_1 \equiv k_1(w_1 v_2) + k_2(-v_1 w_2) \pmod q \quad (2)$$

을 얻게 된다. KT알고리즘은 이 부정방정식을 풀기 위한 것으로 다음의 세 단계로 요약될 수 있다.

1 단계: 아래의 조건을 만족하는 정수들의 쌍 (벡터라 부르자.) $(\gamma_1, \gamma_2) \in \mathbb{Z}^2$를 찾는다.

① 두 합동식 $\gamma_1 \equiv w(w_1 v_2) \pmod q$와 $\gamma_2 \equiv w(-v_1 w_2) \pmod q$를 동시에 만족하는 정수 w가 존재한다.

② $0 \le \gamma_1, \gamma_2 \le O(\sqrt{q})$.

③ γ_1과 γ_2는 서로 소이다.

2 단계: 확장된 유클리드 알고리즘을 이용하여 $K_1\gamma_1 + K_2\gamma_2 = 1$을 만족하는 정수 K_1, K_2를 구하고, $\gamma_3 \equiv w(u_1 v_2 - u_2 v_1) \pmod q$를 계산한다.

3 단계: $g^{k_1} \equiv g^{k'} \pmod p$를 만족하는 정수 k'을 집합 $S = \{(lq + \gamma_3)K_1 + \gamma_2 t \mid l, t \in \mathbb{Z}\}$에서 전수조사를 통해 찾는다.

위의 3 단계에서 k'을 위와 같은 집합 S에서 찾는 이유는 다음과 같다. 부정방정식 (2)의 양변에 w을 곱하면 $\gamma_3 \equiv k_1\gamma_1 + k_2\gamma_2 \pmod q$이고, 따라서 적당한 정수 l에 대하여

$$k_1\gamma_1 = lq + \gamma_3 - k_2\gamma_2 \quad (3)$$

가 된다. 이제 양변에 K_1을 곱하여 항들을 정리하면

$$k_1 = (lq + \gamma_3)K_1 + \gamma_2(k_1K_2 - k_2K_1)$$

를 얻는다. 이러한 계산의 목표 자체가 서명에 사용된 nonce, 예를 들어 k_1, 하나를 구하기 위한 것이므로, 위 식에서 $k_1K_2 - k_2K_1$을 t라 놓은 후 이 t의 정확한 값을 찾아가도 되는 것이다.

위에서 k'을 찾을 때의 조건 $g^{k_1} \equiv g^{k'} \pmod p$은 서명기법에 따라 달라지는데, 여기서는 편의상 ElGamal의 본래 서명기법에서의 조건만을 고려하였다. 일단 k'이 하나만 찾아지면, 서명방정식 $u_1 \equiv xv_1 + k'w_1 \pmod q$으로부터 서명자의 비밀키가 $x \equiv (u_1 - k'w_1)v_1^{-1} \pmod q$로 계산되어진다.

Kuwakado와 Tanaka는 그들의 논문 [3]에서 만약 $0 < k_1, k_2 \le O(\sqrt{q})$라면 그들의 알고리즘의 복잡도가 $O((\log q)^3)$ 비트 연산이라고 주장하였

다. 이 KT알고리즘의 복잡도는 3 단계에서 전수조사를 할 때 시도되어질 l, t의 쌍의 개수에 의해 결정되는데, 이 숫자는 1 단계에서 찾은 벡터 (γ_1, γ_2)의 크기와 직접적으로 관계가 있다. 따라서 이 알고리즘의 복잡도가 주장된 것과 같다는 것을 확인하기 위해서는 벡터 (γ_1, γ_2)를 찾을 때 그 크기가 특정한 범위 안에 있도록 찾는 것이 중요하다. 그러나 Kuwakado와 Tanaka는 이 범위를 구체적으로 주는 대신, 그들의 컴퓨터 시뮬레이션에서 조사된 벡터들의 평균 크기만을 제공하고 있고, 따라서 그들의 복잡도에 대한 분석은 엄밀한 것으로 볼 수 없다.

Ⅲ. KT알고리즘에 대한 개선

앞 절에서 우리는 1 단계의 벡터 (γ_1, γ_2)를 구체적으로 어떻게 찾을 것인가에 대해서는 언급하지 않았다. Kuwakado와 Tanaka는 이 벡터를 찾기 위해 연분수를 사용했으며, 앞에서 언급한 바와 같이 이 벡터의 크기에 대한 구체적인 정보를 주지는 못하였다. 이 절에서 우리는 Gallant 등이 [3]에서 보여준 유클리드 알고리즘을 이용하여 짧은 벡터를 찾아가는 방법을 적절히 변형하여 KT알고리즘의 1 단계를 개선하고, 더불어 여기서 찾게 되는 벡터의 크기의 구체적인 범위를 제시할 것이다.

우선 Gallant 등의 방법을 간단히 살펴보자. 소수 q와 그보다 작은 양의 정수 λ가 있다 하자. q와 λ의 최대공약수를 계산하기 위해 확장된 유클리드 알고리즘을 실행시키면 모든 항들이 식 $s_i q + t_i \lambda = r_i$을 만족하는 수열들 s_i, t_i, r_i가 생성된다. 이 수열들의 초기 항들은 $s_0 = 1, t_0 = 0, r_0 = q, s_1 = 0, t_1 = 1, r_1 = \lambda$이고, 또 이 수열들은 다음 식들을 만족한다.

$$
\begin{aligned}
&r_i > r_{i+1} \geq 0, \ i \geq 0,\\
&|s_i| < |s_{i+1}|, \ i \geq 1,\\
&|t_i| < |t_{i+1}|, \ i \geq 0,\\
&r_{i-1}|t_i| + r_i|t_{i-1}| = q, \ i \geq 1
\end{aligned}
$$

수열 r_i는 감소수열이고 언젠가는 q와 λ의 최대공약수인 1이 그 항으로 나타나게 되므로, $r_{m+1} \leq \sqrt{q} < r_m$을 만족하는 유일한 정수 m이 존재한다. 또, 위의 관계식 $r_m|t_{m+1}| + r_{m+1}|t_m| = q$ 로부터 $|t_{m+1}| < \sqrt{q}$ 임을 알 수 있다. 그러므로 확장된 유클리드 알고리즘을 적용하여 식 $s_{m+1}q + t_{m+1}\lambda = r_{m+1}$을 만족하고 $0 < r_{m+1}$, $|t_{m+1}| < \sqrt{q}$ 인 두 정수 r_{m+1}, t_{m+1}을 항상 구할 수 있다. 지금까지의 논의를 정리하면 다음 정리를 얻는다.

정리 1. [3] 소수 q와 그보다 작은 양의 정수 λ가 있을 때, $0 < r, |t| < \sqrt{q}$, $r - t\lambda \equiv 0 \pmod q$ 인 정수 r, t가 존재한다.

위의 정리를 이용하면 다음과 같은 개선된 KT알고리즘의 1 단계를 얻을 수 있다.

정리 2. 소수 q와 두 정수 $0 < x, y < q$ 가 있다. 그러면 다음 세 조건을 만족하는 정수 w, γ_1, γ_2 가 존재한다.

① $\begin{cases} wx \equiv \gamma_1 \pmod q \\ wy \equiv \gamma_2 \pmod q \end{cases}$;

② $0 < \gamma_1, |\gamma_2| \leq \sqrt{q}$;

③ γ_1과 γ_2는 서로 소이다.

증명: 법 q에 대한 x, y의 역원을 각각 a_1, a_2라고 하자. 즉, $a_1 x \equiv a_2 y \equiv 1 \pmod q$ 이다. 여기서 $\lambda \equiv -a_2 a_1^{-1} \pmod q$라 두고, 정리 1을 적용하면 $0 < \gamma_1, |\gamma_2| < \sqrt{q}$ 이고 $\gamma_1 + \gamma_2\lambda \equiv 0 \pmod q$ 인 두 정수 γ_1, γ_2를 얻게 된다. 이제 λ에 관한 두 합동식으로부터 $a_1\gamma_1 \equiv a_2\gamma_2 \pmod q$ 임을 알 수 있는데, 이 값을 w라 두면 정리의 조건 ①이 성립하게 된다.

만약 γ_1과 γ_2가 서로 소가 아니라면, 이들의 최대공약수 d로 두 수를 나누어 $\gamma_1' = \gamma_1/d$, $\gamma_2' = \gamma_2/d$라 하고 γ_1', γ_2'로 γ_1, γ_2를 대신하게 하면, 정리의 조건을 만족하는 오히려 더 작은 수들을 찾은 것이 된다. (물론 이 때 $w' \equiv a_1\gamma_1' \equiv a_2\gamma_2' \pmod q$이 w를 대신하게 된다.)

Ⅳ. 계산 복잡도

이제 앞 절의 결과로 KT알고리즘 1 단계의 벡터

(γ_1, γ_2)를 그 크기에 관하여 정확한 범위 안에서 찾을 수 있게 되었고, 따라서 KT알고리즘의 복잡도를 정확하게 분석할 수 있게 되었다.

구체적인 분석에 앞서 먼저 사용될 기호들을 정하기로 하자. 먼저 p는 소수이고 g는 소체 GF(p)의 원소로 그 곱셈에 대한 위수가 q이다. 이미 앞에서 q가 소수인 것은 가정하였다. 이제 g에 의해 생성된 부분군 위에 ElGamal의 서명기법을 운용한다고 가정하자. 이런 상황에서 누군가가 message nonce k_1, k_2를 사용하여 생성한 두 개의 서명이 주어진 것으로부터 문제가 시작된다.

2 절의 시작부분에서와 같이 서명 방정식과 두 개의 서명으로부터 부정방정식 (2)를 유도해내고, 여기에 정리 2를 적용하여 KT알고리즘 1 단계의 벡터 (γ_1, γ_2)를 계산해낸다. 그리고 2 단계에서 K_1, K_2와 $\gamma_3 \equiv k_1\gamma_1 + k_2\gamma_2 \pmod q$를 계산한다. 그러면 적당한 정수 l에 대하여 $k_1 = (lq + \gamma_3)K_1 + \gamma_2(k_1K_2 - k_2K_1)$이 되고, 따라서 우리는 집합 $S = \{(lq + \gamma_3)K_1 + \gamma_2 t \mid l, t \in Z\}$ 안에서 전수조사를 통하여 k_1을 찾을 수 있게 된다.

이 전수조사의 실현 가능성을 위해 k_1, k_2와 (γ_1, γ_2)가 $\sqrt{q}$의 적당한 상수 배 범위에 있다는 전제 아래에서 모든 논의가 시작되었다. 논문 [2]에서는 이와 더불어 이들 숫자들이 모두 양수인 것을 가정하고 있다. 그러나 아래의 논의에서 드러나는 바와 같이 이들 중 몇 개의 숫자를 자신의 q-여수 (즉, 어떤 수 X 대신 $q - X$)로 대체하더라도 모든 이야기가 완전히 똑같이 진행될 수 있음을 알 수 있다. 따라서 이 모든 경우를 포함하기 위해 의미를 확대 해석하여 "$|k_1|, |k_2|$ 와 $|\gamma_1|, |\gamma_2|$가 $O(\sqrt{q})$의 범위에 있다"고 말할 것이다.

4.1. 집합 S 안에서의 k'에 대한 전수조사

논의를 위하여 먼저 l은 상수라 가정하고 $f(t) = |(lq + \gamma_3)K_1 + \gamma_2 t|$라 하자. 우선 $l = 0$이라 놓고, 이 경우에 적절한 t의 값을 찾기 위해 f의 값이 최소가 되게 하는 정수 $t = t_0$ ($-\frac{lq+\gamma_3}{\gamma_2}$를 정수로 반올림한 값)에서부터 시작하자. 이제 $k'_0 = (lq + \gamma_3)K_1 + \gamma_2 t_0$가 S의 원소들 중 그 절대값이 가장 작은 것임은 자명하다.

만약 $g^{k_1} \equiv g^{k'_0} \pmod p$라면, 원하는 $k' = k'_0$을 찾은 것이다. 만약 아니라면, 정수 i에 대해 $k'_i = (lq + \gamma_3)K_1 + \gamma_2(t_0 + i)$라 둔다. 이제 $|k'_i|$가 사용된 message nonce의 예상된 범위에 있는 동안 $i = \pm1, \pm2, \pm3, \cdots$로 키워가며 합동식 $g^{k_1} \equiv g^{k'_i} \pmod p$이 성립하는지 확인한다. 만일 $|k'_i|$가 예상된 범위를 벗어날 때까지 이 합동식을 만족하는 것을 찾지 못했다면, 그때는 차례로 $l = \pm1, \pm2, \pm3, \cdots$이라 가정하고 앞의 과정을 반복한다.

이제 서명에 사용된 nonce들이 $O(\sqrt{q})$의 범위에 있다고 가정하자. 또한, 정리 1에 의해 $|\gamma_1|, |\gamma_2| \le \sqrt{q}$이다. KT알고리즘의 계산 복잡도에는 다음의 세 가지 구성요소가 있다.

- 벡터 (γ_1, γ_2)를 계산할 때의 계산 복잡도
- 집합 S에서 전수조사를 할 때 시도해야 할 정수 쌍 l, t의 개수
- 고정된 정수 쌍 l, t가 원하는 k'을 주는지 확인할 때의 계산 복잡도

첫째 요소인 (γ_1, γ_2)의 계산은 확장된 유클리드 알고리즘을 사용하며, 그 복잡도는 $O((\log_2 p)^2)$인 것이 알려져 있다. 셋째 요소의 계산은 $g^{k_1} \equiv g^{k'_i} \pmod p$이 성립하는지 확인하게 위한 거듭제곱연산인데, 그 복잡도는 $O((\log_2 p)^3)$이다. 따라서 남은 것은 둘째 요소인 시도해야 할 정수 쌍 l, t의 개수를 추정하는 것뿐이다.

4.2. 정수 l의 범위

식 (3) $k_1\gamma_1 = lq + \gamma_3 - k_2\gamma_2$과 조건 $|\gamma_1|, |\gamma_2| \le \sqrt{q}$으로부터

$$|lq + \gamma_3| = |k_1\gamma_1 + k_2\gamma_2| \le |k_1||\gamma_1| + |k_2||\gamma_2| \le O(q)$$

임을 알 수 있다. 그러므로 l의 선택의 범위는 p 또는 q의 크기와는 상관이 없고, 최초에 k_1, k_2의 존재 범위에 대한 가정에 관련된 상수의 2배 이하가 된다.

4.3. 정수 t의 범위

먼저 l은 고정된 것으로 가정한다. 4.1 소절에서와 같이 t_0는 $-\frac{lq+\gamma_3}{\gamma_2}$ 를 정수로 반올림한 값이라 하고 $i=0, +1, \pm 2, \pm 3, \cdots$에 대해서 $k'_i = (lq+\gamma_3)K_1 + \gamma_2(t_0+i)$라 하자. 그러면 t_0의 선택에 의하여 $|k'_0|$는 γ_2로 무언가를 나눌 때의 나머지가 되므로 $|k'_0| < |\gamma_2|$가 성립한다. 이로부터

$$(|i|-1)|\gamma_2| \leq |k'_i| \leq (|i|+1)|\gamma_2|$$

임을 알 수 있다. 사용된 nonce가 $|k_1| \leq O(\sqrt{q})$의 범위에 있으므로, 그의 추정 값인 $|k'_i|$도 같은 범위에 있는 것들만 고려하면 된다. 여기서 만약 $|\gamma_2|$의 크기가 $\sqrt{q}$와 거의 같다면 (즉, 1에 가까운 상수 배라면), t_0+i에서의 $|i|$의 선택도 앞 소절의 l의 선택과 마찬가지로 상수의 범위에서 선택하게 된다. Gallant 등의 방법은 대부분의 경우 $|\gamma_2|$의 크기가 $\sqrt{q}$와 아주 가까운 γ_2를 산출해 낸다. 하지만 아주 드문 경우에 $|\gamma_2|$가 너무 작게 나타나는 경우도 있는데, 이 때는 같은 조건의 다른 서명을 이용하여 알고리즘을 적용하게 된다.

따라서 집합 S에서의 전수조사에서 시도해야 할 정수 쌍 l, t의 개수는 상수의 범위에 있다고 말할 수 있다. 그러므로 종합적인 KT알고리즘의 계산 복잡도는 논문 [2]에서 주장된 대로 $O((\log_2 p)^3)$이다.

4.4. 예제

이번 소절에서는 우리의 개선된 KT알고리즘이 수행되는 두 개의 예제를 제시한다. 우선 가상으로 1024비트 소수 p와 160비트 소수 q를 사용하는 ELGamal 서명기법을 구성한다. 다음은 16진법으로 표시한 이 기법의 주요 변수들이다.

$p =$ FFFFFFFF FFFFFFFF 942AC239 A910E7D1
5CF9919D 4C7BC767 3B141098 471B42F4
14BF37AC 1821CD90 0B73E10F 8390353F
1DA775D1 F133805D 22C8C51F 50B9E674
3AD8AC62 6DED728F 8858C076 B7AFB8E9
8FBD1A4C 02D4BA2A AAD9E3FD 52B119E2
BD217BF6 6A4336F9 5DF71428 AAD1B8F6
C3E9B091 15D7E4F3 FFFFFFFF FFFFFFFF

$q =$ 8510730C 80AB5EFA 91485CEF 7AA554FF
E9EC9EC5

$g =$ A0973E51 EB5D1A3C 1473045C 9F1D0A23
1D47458B 165F02EC 81338045 354CE8D4
EF7384E1 E9C8C003 EB847DC8 56CF104B
54782806 387CD6D7 CBA89F3A C0EAD0D9
D1E71E6D 9AC75E62 0E2AB737 00D8E51E
48A91663 EA5A9FFE BC2B0B05 56108BA1
E0C7154B 389E6338 1F8F39A2 513E8DB5
D3B79A7C C2A9E132 F850B443 75BE1271

$x =$ 16592123 3CC19E3C E8C1B984 98EDC945
E6387387

첫 번째 예제에서는 $0 < k_1, k_2 \leq 1000\sqrt{q}$ 인 nonce k_1, k_2를 사용하여 서명을 생성한다.

$k_1 =$ 017977F6 6C8F7B38 33544E3A

$k_2 =$ 01D58679 9D671AD4 2966C813

서명을 생성할 때 메시지를 미리 정하는 대신, 편의상 w_1, w_2의 값을 임의로 추출한 값으로 미리 정하여 사용하였다. ElGamal 서명기법에서는 $v_i = g^{k_i} \bmod p$, $i=1, 2$ 이므로 u_1, u_2는 서명방정식 (1)에 의해 결정된다.

$w_1 =$ 059DB842 5768C3E8 89E6D214 E65225E7
04D39522

$w_2 =$ 2D825EF2 8E5A0C7E DA64FF1D 1F29678F
DB572331

그 후에 우리의 알고리즘을 적용하여

$w =$ 007EC67B 8A691597 080ED136 B69262D4
1A96A276

$\gamma_1 =$ 99E9 F07D140B 2C4FD7E5

$\gamma_2 =$ - 415D 9434BE97 D078F9AD

를 얻는다. 이 경우에는 γ_1의 값이 $|\gamma_2|$보다 두 배 가까이 크므로 γ_1, γ_2의 역할을 서로 바꾸어 집

합 $S=\{-(lq+\gamma_3)K_2+\gamma_1 t \mid l, t\in \mathbb{Z}\}$에서 k_2를 찾는 편이 계산량이 적게 된다. 이 전수조사에서 $l=205, t=t_0+780$ 일 때 k_2를 찾았다.

다음 예제에서는 $0<q-k_1, k_2\le 1000\sqrt{q}$ 인 nonce k_1, k_2를 사용하여 서명을 생성한다.

$k_1=$ 8510730C 80AB5EFA 8FF0DF54 22BA46EF E2F75685

$k_2=$ 01058679 9D671AD4 2966C813

편의상 w_1, w_2의 값은 앞의 예제와 같게 잡았다.

이제 $\lambda\equiv -\frac{w_1 v_2}{w_2 v_1} \pmod q$ 라 두고 우리의 알고리즘을 적용하면,

$w=$ 725F9C36 625CAB46 CB8EEDA1 5852B370 6894B657

$\gamma_1=$ 36DB F4182E39 BC74A2BE

$\gamma_2=$ A0BE 20602C85 6A64B767

를 얻는다. 이 경우에는 $q-k_1$을 집합 $S=\{(lq+\gamma_3)K_1+\gamma_2 t \mid l, t\in \mathbb{Z}\}$에서 찾게 되며, $l=708, t=t_0+548$ 일 때 $q-k_1=(lq+\gamma_3)K_1+\gamma_2 t$ 를 찾았다.

V. 결 론

이 논문에서 우리는 Gallant 등의 방법에서와 같이 확장된 유클리드 알고리즘을 적용하여 KT알고리즘에 소용되는 짧은 벡터를 보다 구체적이고 효율적으로 찾는 방법을 제시하였다. 구체적으로 우리의 개선된 알고리즘으로 찾은 짧은 벡터의 경우, KT알고리즘의 경우와 비교해서 크기는 60% 정도이고 이 벡터를 찾는데 걸린 시간은 대략 1/40 정도가 되었다. 또 KT알고리즘을 적용하기 위한 nonce k에 대한 조건을 $|k|\le O(\sqrt{q})$ 로 약화시켰다. 더불어 이렇게 개선된 KT알고리즘의 계산복잡도가 정확하게 $O((\log_2 p)^3)$임을 보였고, 여기에 포함된 상수인수를 비교적 정확하게 분석하였다.

참고문헌

[1] T. ElGamal, "A Public Key Cryptosystem and a signature scheme based on discrete logarithms", Advances of *Cryptology-CRYPTO '84, LNCS* 196, pp. 10-18, 1985.

[2] H. Kuwakado, H. Tanaka, "On the Security of the ElGamal-Type Signature Scheme with small parameters", *IEICE Trans. Fundamentals*, Vol. E82-A. No. 1, pp. 93-97, Jan. 1999.

[3] R. Gallant, R. Lambert, S. Vanstone, "Faster Point Multip- lication on Elliptic Curves with Efficient Endomorphism", *Advances in Cryptology-Crypto'2001*, pp. 190-201, 2001.

[4] C. H. Lim, P. J. Lee, "A Study on the proposed Korean digital signature algorithm", *Advances in Cryptology-Asiacrypt'1998*, LNCS 1514, pp. 175-186, 1998.

[5] A. Menezes, P. Ooschot, S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1997.

[6] P. Q. Nguyen, I. E. Shparlinski, "The Insecurity of the Digital Signature Algorithm with Partially Known Nonces", *J. Cryptology*, Vol. 15, pp. 151-176, 2002.

[7] C. P. Schnorr, "Efficient Identification and signatures for smart cards", Advances in *Cryptology-Crypto'1989, LNCS* 435, pp. 239-252, 1990.

[8] U.S. Department of Commerce/ N.I.S.T., *Digital Signature Standard*, FIPS Pub. 186, 1994.

〈著者紹介〉

이 익 권 (Ikkwon Yie) 정회원
1985년 2월: 서울대학교 수학과 이학사
1987년 2월: 서울대학교 수학과 이학석사
1995년 5월: 미국 Purdue University Ph.D.
1995년 9월~현재: 인하대학교 수학과 부교수

김 동 렬 (Dongryeol Kim)
1990년 2월: 서울대학교 수학과 이학사
1992년 2월: 서울대학교 수학과 이학석사
1997년 8월: 서울대학교 수학과 이학박사
2001년 9월~현재: 한국정보보호진흥원 선임연구원