通信情報保護學會誌
第 10 券 第 2 號, 2000. 6

타원곡선 알고리즘 표준화 동향

이 필 중*

요 약

본 고에서는 타원곡선 알고리즘의 표준화 동향을 국내외 표준들을 바탕으로 살펴보았다. 먼저, ISO/IEC JTC1/SC27/WG2 Information technology - Security techniques - Cryptographic techniques based on elliptic curves 문서를 바탕으로 국제표준에 대해서 자세히 살펴보았으며, IEEE P1363, ANSI X9.62/X9.63에 대해서 간략히 살펴보았다. 또한 타원곡선 알고리즘과 관련된 국내 표준화 활동에 대해서도 살펴보았다.

Ⅰ. 서 론

1976년 Diffie와 Hellman[1]에 의해서 공개키 암호시스템이 제안된 이후로 그 응용범위는 전자서명 및 전자상거래까지 확대되기에 이르렀다. 공개키 암호시스템은 크게 두 가지로 분류될 수 있다.

첫 번째로 소인수분해의 어려움(IFP, Integer Factorization Problem)에 기반한 RSA, RW(Rabin- Williams) 등이며 두 번째로 이산대수문제(DLP, Discrete Logarithm Problem)의 어려움에 기반한 ElGamal, DSA, Schnorr, Diffie-Hellman등이 있다. 기존의 DLP에 기반한 프로토콜들은 타원곡선의 이산대수문제(ECDLP, Elliptic Curve DLP)가 풀기 어렵다는데 기초하여 타원곡선에서 정의된 형태로 변형될 수 있다.

이러한 타원곡선 암호시스템은(ECC, Elliptic Curve Cryptosystem) 1985년에 Koblitz[2]와 Miller[3]에 의해서 독립적으로 제안되었다. ECC의 장점으로는 현재까지 subexponential time의 공격방법이 알려져 있지 않다는 데에 있다. 반면 현재 가장 많이 사용되고 있는 IFP나 DLP에 근거한 공개키암호시스템은 subexponential time의 공격방법이 알려져 있다. 따라서 같은 암호학적 안전도를 유지하면서도 ECC는 다른 시스템에 비해 사용되는 키 길이도 현저히 작고(약 1/6정도) 연산도 효율적이다.

따라서 저장 용량 및 연산력(computing power)에 제한이 있는 smart card나 무선통신단말기에 유용하게 사용될 수 있다. 또한 ECC는 기존의 공개키 암호시스템이 사용된 곳 모두에 적용 가능하다.

그 동안 일반적인 공개키 암호 시스템에 관련된 국제 표준들은 많이 진행된 데 비해서 타원곡선과 관련해서는 그 진행이 아직 미비하다 할 수 있다. 최근까지 진행되고 있는 국내외 표준 문서들은 다음과 같다.

- ISO/IEC JTC1/SC27 15946 - 1 Information technology - Security techniques - Cryptographic techniques based on elliptic curves - Part1 : General
- ISO/IEC JTC1/SC27 15946 - 2 Information technology - Security techniques - Cryptographic techniques based on elliptic curves - Part2 : Digital signatures
- ISO/IEC JTC1/SC27 15946 - 3 Information technology - Security techniques - Cryptographic techniques based on elliptic curves - Part3 : Key establishment
- IEEE P1363 Standard Specifications

* 포항공과대학교 전자전기공학과(pjl@postech.ac.kr)

for Public Key Cryptography
- ANSI X9.62 The Elliptic Curve Digital Signature Algorithm(EC-DSA)
- ANSI X9.63 The Elliptic Curve Key Agreement and Transport Protocols
- 부가형 전자서명 방식 표준(안) - 제3부: 타원곡선을 이용한 인증서 기반 전자서명 알고리즘

본 고에서는 이상과 같은 국내외 표준 문서들 중 ISO/IEC JTC1/SC27/WG2 Information technology - Security techniques - Cryptographic techniques based on elliptic curves를 중심으로 타원곡선 알고리즘의 표준화 동향을 자세히 살펴본 후, 다른 표준 문서들에 대해서도 간략히 알아보도록 하겠다.

II. ISO/IEC JTC1/SC27

국제표준화기구인 ISO(International Organization for Standardization)와 IEC (International Electro-technical Commission)는 정보기술의 전세계적인 표준화를 위하여 첫 번째의 Joint Technical Committee를 구성하였다(ISO/IEC JTC1). JTC1 에는 30여 개의 Sub-Committee(SC)가 있으며, 그 중 SC27은 보안기술(security technique)을 다루고 있다.

본 장에서는 ISO/IEC JTC1/SC27 표준 문서들 중에서 타원곡선 알고리즘과 관련된 문서인 ISO/IEC 15946 Information technology - Security techniques - cryptographic based on elliptic curves에 대해서 살펴보겠다.

1. 표준화 과정

ISO/IEC JTC1/SC27 표준 문서들 중에서 타원곡선 알고리즘과 관련하여 초기에 작성된 문서는 WG2/N414이다. WG2/N414는 1998년 1월에 작성된 문서로서 타원곡선에 대한 기본적인 내용들을 정리하고 있다. 그 구성을 살펴보면 아래와 같다.

- Finite field F(p), $F(2^m)$
- Definition of elliptic curve over F(p) or $F(2^m)$
- Operation on elliptic curves
- Key agreement mechanism
- Key transport mechanism
- Digital signature
- Entity authentication

이 문서에서는 ECC에 사용되는 유한체를 F(p), $F(2^m)$만으로 지정하고 있으며, key agreement를 위해서는 Diffie-Hellman 방식의 시스템을 소개하고 있다. 또한 key transport mechanism을 위해서는 ElGamal방식의 시스템을 소개하고 있고, 전자 서명을 위해서는 EC-DSA를 규정하고 있다.

1998년 4월 Kista에서 있었던 ISO/IEC 회의에서는 한국의 comment가 받아들여져 유한체에 $F(p^m)$을 추가할 것과 X9.62와 P1363에서는 포함하고 있는 point compression의 내용을 선택사항으로 추가할 것, 그리고 WG2/N414의 내용을 좀 더 확충하고 세 개의 part로 나누는 것 등을 골자로 하는 합의가 있었다. 그 세 부분은 다음과 같다.

- Part 1 : General
- Part 2 : Digital Signatures
- Part 3 : Key Establishment

Part 1에서는 세 가지 유한체 F(p), $F(2^m)$, $F(p^m)$에 대해서 상술하고, 각각의 유한체 상에서 정의되는 타원곡선에 대한 내용을 담도록 하였다. Part 2에서는 WG2/N414에 포함되었던 EC-DSA의 내용과 아울러 KCDSA(Korean Certificate-based Digital Signature Algorithm)의 타원곡선 버전인 EC-KCDSA도 포함될 수 있는 가능성이 보였다. 이러한 여지는 유용한 메커니즘이라면 계약을 두지 않고 표준에 포함시킬 수 있다는 기본 방향에 대한 합의가 Kista에서 이루어졌기 때문이다. 마지막으로 Part 3에서는 N414에 포함되었던 Key agreement와 Key transport mechanism의 내용이 포함되도록 하였다.

이들 세 부분은 1998년 6월 각기 SC27문서의 WD(Working Draft)인 N2034, N2056, N2057로 구분되어 독립된 문서가 되었다. N2034는 Part 1에 해당하는 부분으로 유한체에 대한 설명과 특정 유한체 상에서 정의되는 타원곡선들에 대해서 설명하고 있다. 전체적인 내용을 살피기 위해

목차를 보면 다음과 같다.

- Finite field F(p), F(2^m), F(p^m)
- Elliptic curve definition
- Operation on elliptic curve
- Elliptic curve domain parameters
- Elliptic curve key pair

한국의 제안이었던 유한체 F(p^m)의 내용 추가를 위해 한국에서 문서의 해당부분을 준비하였다.

N2056은 ISO/IEC의 타원곡선 표준문서에서 part 2에 해당하는 부분으로 전자서명에 대한 내용을 담고 있다. 목차를 살펴보면 다음과 같다.

- General model for digital signature
  - Parameter generation process
  - Signature generation process
  - Signature verification process
- EC-ElGamal signature algorithm
- EC-DSA
- EC-KCDSA

N2056은 기본적인 타원곡선 서명 시스템에 대해서 설명을 하고 있으며, 실제적인 시스템으로서 EC-ElGamal, EC-DSA, EC-KCDSA 등을 담고 있다. 또한, 부기에서는 전자 서명의 구현 예제 및 언급된 전자 서명들에 대한 비교가 포함되도록 하였다.

N2056에 국내 표준 서명 KCDSA의 타원곡선 버전인 EC-KCDSA가 포함된 것을 비롯해서 구현 예제, 알고리즘들의 비교 등에서 우리 나라의 주도적인 역할이 있었다.

그 이후 1999년 1월에는 N2134, N2056, N2057이 각각 WD인 N2155, N2157, N2159로 개정되었는데, 우리 나라에서는 Part1 - General인 N2155에 대해서는 본문의 정의 및 기술, 타원곡선의 수학적 개요에 대한 지적을 하였다. 또한 Part2 - Digital Signatures인 N2157에서는 EC-KCDSA에 대한 기술 문서를 제공하였으며, EC-KCDSA와 EC-DSA의 비교 설명을 부기에 첨가하도록 지속적으로 노력하였고, 16비트 p와 m = 11을 사용한 타원곡선에 대한 구현 예제를 제공하였다. Part3 - Key Establishment인 N2159에 대해서는 본문의 정의, 표기, 기법 등에 대한 지적을 함으로써 표준화 활동에 적극적으로 참여하였다. 1999년 4월 Madrid 회의의 결과 중 재미있는 것은 독일에서 제안한 EC-ElGamal은 제대로 된 이름이 아니라는 지적에 그 알고리즘의 이름을 한국의 예를 따라 EC-GDSA로 바꾸자고 독일이 제안한 점이다. 그리고 WD에서 CD(Committee Draft) 수준으로 격상시키기로 했다.

1999년 6월에는 N2301, N2303, N2305로 문서들이 개정되었는데 N2303에서는 EC-KCDSA의 서명 생성 과정을 수정하였다. 서명의 첫 번째 부분인 r을 계산할 때에 난수인 정수값 k와 생성자인 타원곡선의 점 G에 대해서 kG = (x_1,x_2)을 계산하여 r = h(x_1||y_1)으로 계산하였던 것을 r = h(x_1)으로 계산하도록 수정하였다. 이것은 x좌표계만을 이용하여 rG의 x좌표값을 효율적으로 계산할 수 있는 방법(Montgomery Approach[6])이 있기 때문이다. 또, N2303에서는 N2157의 부기에 있던 E(F(p^m))의 예제를 32비트 p와 m = 5인 예제로 갱신하였다. 또한 E(F(p)), E(F(2m))의 예제를 추가하였고, EC-KCDSA가 random oracle model에서 secure하기 때문에 EC-DSA에 비해 provable security를 갖는다는 것을 내용으로 하는 contribution paper를 제출하여 N2157의 부기에서 EC-KCDSA와 EC-DSA를 비교한 내용에 대한 근거를 보였다.

1999년 12월에는 N2441, N2443, N2445의 문서가 새로 배포되었으며 Part1 - General인 N2441과 Part2 - Digital Signatures인 N2443은 FCD(Final Committee Draft)가 되었다.

다음 장에서는 최근 문서인 N2441, N2443, N2445를 바탕으로 타원곡선 알고리즘의 국제 표준화 내용을 자세히 살펴보도록 하겠다.

2. Part1-General

Information Technology - Security Techniques - Cryptographic Techniques based on Elliptic Curves : Part1 - General 문서에서는 타원곡선 알고리즘의 기반이 되는 유한체와 이러한 유한체에서 정의되는 타원곡선 및 타원곡선 도메인 변수들의 생성과 검증, 키의 생성과 검증 등에 대해서 기술하고 있다.

전체적인 내용을 살피기 위해 목차를 보면 다음과 같다.

- Finite fields
- Elliptic curves over F(p), $F(2^m)$, $F(p^m)$
- Elliptic curve domain parameters and their validation
- Elliptic curve key pair generation and public key validation
- Background Information on elliptic curves

2.1 Finite fields

유한체 F(p), $F(2^m)$, $F(p^m)$를 표준으로 정하고 있으며, 각각에 대해서 addition과 multiplication 방법을 기술하고 있다.

2.2 Elliptic curves over F(p), $F(2^m)$, $F(p^m)$

유한체 F(p), $F(2^m)$, $F(p^m)$에서 정의되는 타원곡선을 각각 정의하고 있으며, 타원곡선상에서의 group law에 대한 설명과 negative point의 정의 및 integer multiplication과 ECDLP에 대해서 정의하고 있다. 또한, 타원곡선 상의 점을 integer로 변환하는 방식에 대해서도 기술하고 있다.

2.3 Elliptic curve Domain Parameters and their Validation

타원곡선의 도메인 변수들에 대해서 정의하고 있으며 이들의 검증방법에 대해서 기술하고 있다. F(p), $F(p^m)$에서 정의된 타원곡선의 경우(F(p)는 m = 1인 $F(p^m)$의 특수한 경우라고 생각하면), 도메인 변수들은 유한체의 크기 pm, 타원곡선이 임의로 생성되었다면 SEED 값, 타원곡선 E : $y^2 = x^3 + ax + b$를 정의하는 유한체상의 값 a, b, 타원곡선 E에서 소수인 위수를 가지는 점 $G(x_G, y_G)$, G의 위수 n, 그리고 cofactor h = #E(F(p^m))/n 이다.

$F(2^m)$에서 정의된 타원곡선의 경우에도 F(p), $F(p^m)$에서 정의된 타원곡선과 같으나, 이 경우에는 타원곡선이 E : $y^2 + xy = x^3 + ax^2 + b$가 된다.

2.4 Elliptic curve Key Pair Generation and Public Key Validation

주어진 타원곡선 도메인 변수들에 대해서 비밀키와 공개키를 생성하는 방법에 대해서 기술하고 있다.

3. Part2-Digital signatures

Information Technology - Security Techniques - Cryptographic Techniques based on Elliptic Curves : Part2 - Digital Signatures에서는 EC-GDSA, EC-DSA EC-KCDSA 에 대해서 기술하고 있다. 각각의 서명 알고리즘에 대해서 살펴보면 다음과 같다.

3.1 EC-GDSA

1) 도메인 변수와 사용자 변수

- n의 비트 길이는 해쉬 함수 h의 출력 비트 길이보다 길어야 한다.
- 서명자 A의 비밀키와 공개키 dA, PA는 Part1에서 정의된 과정과 일치되게 생성되어야 한다.

2) 서명 생성 과정

서명 과정에 필요한 입력은 다음과 같다.

- domain parameters
- 서명자의 비밀키 d_A
- 메시지 M

서명 생성 과정의 출력은 $(r,s) \in F(n)^* \times F(n)^*$이며, 이 값이 메시지 M에 대한 A의 서명이 된다. 서명 과정을 기술하면 다음과 같다.

1. 해쉬 코드 e = h(M)을 계산한다.
2. {1, . . ., n-1}에서 임의의 값 k를 선택한다.
3. 타원곡선 상의 점 (x_1, y_1) = kG를 계산한다.
4. r = π(kG) mod n을 계산한다.
5. s = $(kr - e)d_A$ mod n을 계산한다.

만일 s = 0이거나 r = 0이면 2번째 단계에서 새로운 k값을 선택해서 서명 생성 과정을 되풀이한다.

3) 서명 검증 과정

서명 검증 과정에 필요한 입력은 다음과 같다.

- 도메인 변수들
- A의 공개키 P_A

- 받은 메시지 M′
- 메시지 M에 대한 받은 서명, r′, s′

A의 메시지 M′에 대한 서명을 검증하기 위해 B는 다음과 같은 검증 과정을 거친다.

1. 0 < r′ < n인지 0 < s′ < n인지 확인한다. 만일 r′와 s′가 이 범위 내에 있지 않으면 서명은 잘못된 것으로 간주한다.
2. 해쉬 코드 e′ = h(M′)을 계산한다.
3. $w=(r')^{-1} \bmod n$을 계산한다.
4. $u_1 = e'w \bmod n$과 $u_2 = s'w \bmod n$을 계산한다.
5. 타원곡선 상의 점 $(x_1, y_1) = u_1G + u_2P_A$를 계산한다.
6. $v = \pi((x_1, y_1))$를 계산한다.

만일 r′ = v이면 서명은 올바른 것으로 받아들여지며 그렇지 않은 경우 서명은 잘못된 것으로 간주한다.

3.2 EC-DSA

1) 도메인 변수와 사용자 변수
   - n의 비트 길이는 해쉬 함수 h의 출력 비트 길이보다 길어야 한다.
   - 서명자 A의 비밀키와 공개키 d_A, P_A는 Part1에서 정의된 과정과 일치되게 생성되어야 한다.

2) 서명 생성 과정
서명 과정에 필요한 입력은 다음과 같다.

- 도메인 변수들
- 서명자의 비밀키 d_A
- 메시지 M

서명 생성 과정의 출력은 $(r, s) \in F(n)^* \times F(n)^*$이며, 이 값이 메시지 M에 대한 A의 서명이 된다. 서명 과정을 기술하면 다음과 같다.

1. 해쉬 코드 e = h(M)을 계산한다.
2. {1, . . . , n-1}에서 임의의 값 k를 선택한다.
3. 타원곡선 상의 점 $(x_1, y_1) = kG$를 계산한다.
4. $r = \pi(kG) \bmod n$을 계산한다.
5. F(n)에서 k^{-1}를 계산한다.
6. $s = (d_Ar + e)k^{-1} \bmod n$을 계산한다.

만일 s = 0이거나 r = 0이면 2번째 단계에서 새로운 k값을 선택해서 서명 생성 과정을 되풀이한다.

3) 서명 검증 과정
서명 검증 과정에 필요한 입력은 다음과 같다.

- 도메인 변수들
- A의 공개키 P_A
- 받은 메시지 M′
- 메시지 M에 대한 받은 서명, r′, s′

A의 메시지 M′에 대한 서명을 검증하기 위해 B는 다음과 같은 검증 과정을 거친다.

1. 0 < r′ < n인지 0 < s′ < n인지 확인한다. 만일 r′와 s′가 이 범위 내에 있지 않으면 서명은 잘못된 것으로 간주된다.
2. 해쉬 코드 e′ = h(M′)을 계산한다.
3. $w = (s')^{-1} \bmod n$을 계산한다.
4. $u_1 = e'w \bmod n$과 $u_2 = r'w \bmod n$을 계산한다.
5. 타원곡선 상의 점 $(x_1, y_1) = u_1G + u_2P_A$를 계산한다.
6. $v = \pi((x_1, y_1))$를 계산한다.

만일 r′ = v이면 서명은 올바른 것으로 받아들여지며 그렇지 않은 경우 서명은 잘못된 것으로 간주한다.

3.3 EC-KCDSA

1) 도메인 변수와 사용자 변수
   - n의 비트 길이는 해쉬 함수 h의 출력 비트 길이보다 길어야 한다.
   - 서명자 A의 비밀키와 공개키 d_A, P_A는 Part1에서 정의된 과정과 일치되게 생성되어야 한다.
   - 서명자 A는 Cert_Data의 해쉬코드인 z_A를 가지고 있다. Cert_Data는 A의 certification data로서 최소한 A의 distinguished identifier, 공개키 P_A와 domain parameter들을 포함해야 한다.

2) 서명 생성 과정
서명 과정에 필요한 입력은 다음과 같다.

- 도메인 변수들
- 서명자의 비밀키 d_A
- 서명자의 Cert_Data의 해쉬코드인 z_A
- 메시지 M

서명 생성 과정의 출력은 (r, s)이며 메시지 M에 대한 A의 서명이 된다. 서명 과정을 기술하면 다음과 같다.

1. {1, . . . , n-1}에서 임의의 값 k를 선택한다.
2. 타원곡선 상의 점 $(x_1, y_1) = kG$를 계산한다.
3. $r = h(x_1)$을 계산한다.
4. $H = h(z_A||M)$을 계산한다.
5. $e = r$ XOR H를 계산한다. 만일 $e \geq n$이면 $e = e - n$을 한다.
6. $s = d_A(k - e) \bmod n$을 계산한다.

만일 s=0이면 새로운 k값을 선택해서 서명 생성 과정을 되풀이한다.

3) 서명 검증 과정
서명 검증 과정에 필요한 입력은 다음과 같다.

- 도메인 변수들
- A의 공개키 P_A
- A의 Cert_Data의 해쉬코드인 zA
- 받은 메시지 M′
- 메시지 M에 대한 받은 서명, r′, s′

A의 메시지 M′에 대한 서명을 검증하기 위해 B는 다음과 같은 검증 과정을 거친다.

1. $0 < s' < n$인지 그리고 r′의 비트 길이가 h()의 비트 길이보다 작거나 같은지 확인한다. 만일 r′과 s′가 이 범위 내에 있지 않으면 서명은 잘못 된 것으로 간주된다.
2. 해쉬 코드 $H' = h(z_A||M)$을 계산한다.
3. $e' = r'$ XOR H′를 계산한다. 만일 $e' \geq n$이면 $e' = e' - n$을 한다.
4. $(x_1, y_1) = s'P_A + e'G$를 계산한다.
5. $v = h(x_1)$을 계산한다.

만일 $r' = v$이면 서명은 올바른 것으로 받아들여지며 그렇지 않은 경우 서명은 거절된다.

표 1. Comparison of the number of operations

Process	Operation	EC-DSA	EC-GDSA	EC-KCDSA
Signature Generation	h()	1	1	2
	$\pi()$	1	1	0
	$k^{-1} \bmod n$	1	0	0
	multiplication in Z_n	2	2	1
	addition in Z_n	1	1	1
	scalar multiplication of a curve point	1	1	1
Signature Verification	h()	1	1	2
	$\pi()$	1	1	0
	$s^{-1} \bmod n$	1	1	0
	multiplication in Z_n	2	2	0
	scalar multiplication of a curve point	2	2	2
	addition of curve points	1	1	1

표 2. Comparison of description and operations

	EC-DSA	EC-GDSA	EC-KCDSA
Security Parameters	n, h()		
Condition of n	$n \geq 2^{\lfloor h() \rfloor}$	$n \geq 2^{\lfloor h() \rfloor}$	$n \geq 2^{\lfloor h() \rfloor - 1}$
Private key	$d_A \in \{1,\ldots,n-1\}$		
Public key computation	$P_A = d_A G$	$P_A = (d_A^{-1} \bmod n)G$	$P_A = (d_A^{-1} \bmod n)G$
Signature Generation	$k \in \{1,\ldots,n-1\}$ $r = \pi(kG) \bmod n$ $s = (d_A r + h(M)k^{-1}) \bmod n$	$k \in \{1,\ldots,n-1\}$ $r = \pi(kG) \bmod n$ $s = (kr - h(M))d_A \bmod n$	$k \in \{1,\ldots,n-1\}$ $r = h(x_1)$ $s = d_A(k-r)$ XOR $h(z_A\|\|M) \bmod n$
Signature size	$0 < r < n,\ 0 < s < n$	$0 < r < n,\ 0 < s < n$	$0 < s < n,$ $len_r \leq lenh()$
Signature verification	$u_1 = s'^{-1}h(M') \bmod n$ $u_2 = s'^{-1}r' \bmod n$ $\pi(u_1G + u_2P_A) \bmod n = r'?$	$u_1 = r'^{-1}h(M') \bmod n$ $u_2 = r'^{-1}s' \bmod n$ $\pi(u_1G + u_2P_A) \bmod n = r'?$	$e' = r'$ XOR $h(z_A\|\|M') \bmod n$ $h(s'P_A + e'G) = r'?$

부록에서는 EC-DSA와 EC-GDSA, EC-KCDSA 알고리즘을 다음의 표1과 표2와 같이 비교해 놓고 있고 여러 가지 다른 비교도 하고 있다. 특히 EC-KCDSA는 다른 것들과는 달리 안전성에 대한 증명이 가능함을 설명하고 있다.

4. Part3-Key Establishment

Information Technology - Security Techniques - Cryptographic Techniques based on Elliptic Curves : Part3 - Key Establishment 문서에서는 타원곡선을 이용한 session key 공유 알고리즘들을 key agreement mechanism과 key transport mechanism으로 나누어 기술하고 있다.

Cofactor multiplication을 사용할 경우와 그렇지 않을 경우의 호환성을 위해서, h와 ι을 parameter로 사용한다. Cofactor multiplication을 사용하고 사용하지 않는 경우와의 호환성이 필요없는 경우 h = #E/n, ι = 1로 하고 cofactor multiplication을 사용하고 사용하지 않는 경우와 호환성이 필요한 경우에는 h = #E/n, l = h^{-1} mod n으로 한다. 만일 cofactor multiplication을 사용하지 않는 경우에는 h = 1, ι = 1로 한다.

4.1 Key agreement mechanisms

1) Non-interactive key agreement of Diffie-Hellman type (KANIDH)

(1) Mechanism

- A는 자신의 비밀 키 dA와 B의 공개 키 PB를 사용하여 다음과 같이 session key를 생성한다.

$$K_{AB} = (d_A \cdot l)(h \cdot P_B)$$

- B는 자신의 비밀 키 dB와 A의 공개 키 PA를 사용하여 다음과 같이 session key를 생성한다.

$$K_{AB} = (d_B \cdot l)(h \cdot P_A)$$

(2) Properties

- Number of passes : 0
- mutual implicit key authentication을 제공한다.

2) Key agreement of ElGamal type (KAEG)

(1) Mechanism

- A는 {1, . . . , n-1}에서 임의의 r을 선택하여 다음과 같은 key token을 만들어 B에게 전달한다.

$$KT_{A1} = rG$$

- A는 다음과 같은 session key를 생성한다.

$$K_{AB} = (r \cdot l)(h \cdot P_B)$$

- B는 KTA1이 타원곡선상의 점인지 확인하고 자신의 비밀키를 사용하여 다음과 같이 session key를 생성한다.

$$K_{AB} = (d_B \cdot l)(h \cdot KT_{A1})$$

(2) Properties

- Number of passes : 1
- B로부터 A로 implicit key authentication을 제공한다.
- A에 대해서 forward secrecy를 제공한다.

3) Key agreement of Diffie-Hellman type with 2 key pairs(KADH2KP)

(1) Mechanism

- A는 {1,...,n-1}에서 임의의 r_A을 선택하여 다음과 같은 key token을 만들어 B에게 전달한다.

$$KT_{A1} = r_A G$$

- B는 {1,...,n-1}에서 임의의 r_B을 선택하여 다음과 같은 key token을 만들어 A에게 전달한다.

$$KT_{B1} = r_B G$$

- A는 KT_{B1}이 타원곡선상의 점인지 확인하고 자신의 비밀키를 사용하여 다음과 같이 session key를 생성한다.

$$K_{AB} = (d_A \cdot l)(h \cdot KT_{B1}) || (r_A \cdot l)(h \cdot P_B)$$

- B는 KT_{A1}이 타원곡선상의 점인지 확인하고 자신의 비밀키를 사용하여 다음과 같이 session key를 생성한다.

$$K_{AB} = (d_B \cdot l)(h \cdot KT_{A1}) || (r_B \cdot l)(h \cdot P_A)$$

(2) Properties

- Number of passes : 2

- A와 B 각각에 대해서 forward secrecy를 제공한다.
- mutual implicit key authentication을 제공한다.

4) Key agreement of Diffie-Hellman type with 2 signatures and key confirmation (KADH2SKC)

(1) Mechanism

- A는 {1,...,n-1}에서 임의의 r_A을 선택하여 다음과 같은 key token을 만들어 B에게 전달한다.

$$KT_{A1}=r_AG$$

- B는 KT_{A1}이 타원곡선상의 점인지 확인하고 {1,...,n-1}에서 임의의 r_B을 선택하여 다음과 같은 session key를 생성한다.

$$K_{AB}=(r_B \cdot l)(h \cdot KT_{A1})$$

그리고, 다음과 같은 key token을 만들어 A에게 보낸다.

$$KT_{B1}=S_B(DB_1)||f\_K_{AB}(DB_1),$$

여기서 $DB_1=r_BG||KT_{A1}||A\{||Text1\}$ 이다.

- A는 r_BG가 타원곡선 상의 점인지 확인하고 다음과 같이 session key를 생성한다.

$$K_{AB}=(r_A \cdot l)(h \cdot r_BG)$$

그리고 K_{AB}를 이용하여 $f_K_{AB}(DB_1)$를 확인하고 다음과 같은 key token을 만들어 B에게 전달한다.

$$KT_{A2}=S_A(DB_2)||f\_K_{AB}(DB_2)$$

여기서 $DB_2=r_AG||r_BG||B\{||Text2\}$ 이다.

- B는 K_{AB}를 사용하여 $f_K_{AB}(DB_2)$를 확인한다.

(2) Properties

- Number of passes : 3
- mutual forward secrecy를 제공한다.
- mutual explicit key authentication과 mutual entity authentication을 제공한다.

4.2 Key agreement mechanisms not included in ISO/IEC 11770-3

1) Key agreement of MQV type with 1 pass (KAMQV1P)

(1) Mechanism

- A는 {1,...,n-1}에서 임의의 r을 선택하여 다음과 같은 key token을 만들어 B에게 전달한다.

$$KT_{A1}=rG$$

- A는 다음과 같이 session key를 생성한다.

$$K_{AB}=((r+\pi(KT_{A1})d_A) \cdot l)(h \cdot (P_B+\pi(P_B)P_B))$$

- B는 KT_{A1}이 타원곡선상의 점인지 확인하고 자신의 비밀키를 사용하여 다음과 같이 session key를 생성한다.

$$K_{AB}=((d_B+\pi(P_B)d_B) \cdot l)(h \cdot (KT_{A1}+\pi(KT_{A1})P_A))$$

(2) Properties

- Number of passes : 1
- mutual implicit key authentication을 제공한다.
- A에 대해서 forward secrecy를 제공한다.

2) Key agreement of MQV type with 2 passes (KAMQV2P)

(1) Mechanism

- A는 {1, . . . , n-1}에서 임의의 r_A을 선택하여 다음과 같은 key token을 만들어 B에게 전달한다.

$$KT_{A1} = r_AG$$

- B는 {1,...,n-1}에서 임의의 r_B을 선택하여 다음과 같은 key token을 만들어 A에게 전달한다.

$$KT_{B1} = r_BG$$

- A는 KT_{B1}이 타원곡선상의 점인지 확인하고 다음과 같이 session key를 생성한다.

$$K_{AB}=((r_A+\pi(KT_{A1})d_A) \cdot l)(h \cdot (KT_{B1}+\pi(KT_{B1})P_B))$$

- B는 KT_{A1}이 타원곡선상의 점인지 확인하고 다음과 같이 session key를 생성한다.

$$K_{AB}=((r_B+\pi(KT_{B1})d_B) \cdot l)(h \cdot (KT_{A1}+\pi(KT_{A1})P_A))$$

(2) Properties

- Number of passes : 2
- mutual implicit key authentication을

제공한다.
· mutual forward secrecy를 제공한다.

4.3 Key transport mechanisms

1) Key transport of ElGamal type (KTEG)

(1) Mechanism
· A는 {1,...,n-1}에서 임의의 r을 선택하여 rG를 계산하고 다음과 같이 K를 암호화한다.

$BE=(A||K)\ XOR\ kdf(\pi((r\cdot l)(h\cdot P_B)),$ parameters)

그리고 A는 다음과 같은 K'을 만든다.

$K'=kdf(\pi((r\cdot l)(h\cdot P_B)),MACparameter)$

그리고 다음과 같은 key token을 만들어 B에게 전달한다.

$KT_{A1}=BE||rG||MAC(K',BE)$

· B는 rG가 타원곡선 상의 점인지 확인한다. 그리고 d_B와 rG를 사용해서 $(d_B\cdot l)(h\cdot rG)$를 계산한다. 그 다음 $\pi((d_B\cdot l)(h\cdot rG))$를 계산해서 다음을 얻음으로 K를 알게 된다.

$A||K=BE\ XOR\ kdf(\pi((d_B\cdot l)(h\cdot rG)),$ parameters)

마지막으로 B는 다음의 K''을 계산해서 MAC(K'', BE)가 MAC(K', BE)와 같은지 확인한다.

$K''=kdf(\pi((dB\cdot l)(h\cdot rG)),$ MACparameter)

(2) Properties
· Number of passes : 1
· B로부터 A로 implicit key authentication을 제공한다.
· A에 대해서 forward secrecy를 제공한다.

2) Key transport of ElGamal type with originator signature (KTEGOS)

(1) Mechanism
· A는 {1,...,n-1}에서 임의의 r을 선택하여 rG를 계산하고 다음과 같이 A||K를 계산한다.

$BE=(A||K)\ XOR\ kdf(\pi((r\cdot l)(h\cdot P_B)),$ parameters)

· A는 다음과 같은 key token을 생성한다.

$KT_{A1} = B||TVP||rG||BE$

그 다음 A는 다음과 같이 서명을 해서 B에게 전달한다.

$S_A(KT_{A1})$

· B는 A의 서명을 확인하고 rG가 타원곡선 상의 점인지 확인한다. 그리고 다음과 같이 K를 구하게 된다.

$A||K=BE\ XOR\ kdf(\pi((d_B\cdot l)(h\cdot rG)),$ parameters)

(2) Properties
· Number of passes : 1
· mutual implicit key authentication을 제공한다.
· A에 대해서 forward secrecy를 제공한다.

Ⅲ. IEEE P1363

P1363은 타원곡선과 관련된 알고리즘뿐만 아니라 일반적인 공개키 암호 시스템 전반의 내용을 포괄적으로 다루고 있다. 이 표준에서는 DLP, ECDLP, IFP를 바탕으로 하는 여러 가지 primitive들과 이들을 이용한 key agreement schemes, signature schemes, encryption schemes을 표준으로 정하고 있다.

P1363의 대체적인 내용은 아래의 목차로 정리할 수 있다.

· Finite field
· Elliptic curves and points
· Primitives based on the discrete logarithm problem
· Primitives on the elliptic curve discrete logarithm Problem
· Primitives on the integer factorization problem
· Key agreement schemes
· Signature schemes
· Encryption schemes
· Message encoding methods
· Key derivation functions

ECDLP를 바탕으로 하는 primitive의 경우, 유

한체로는 F(p)와 $F(2^m)$만을 고려하고 있으며, Diffie-Hellman방식과 MQV(Menezes-Qu-Vanstone)방식의 key agreement scheme과 Nyberg-Rueppel방식과 DSA를 signature scheme의 표준으로 정하고 있다.

P1363은 1994년부터 꾸준히 갱신되고 있으며 최근 문서는 1999년 11월 12일에 표준으로 확정되었다(Draft version 13[5]).

Ⅳ. ANSI X9.62/X9.63

ANSI X9의 문서들은 미국 금융산업에 적용될 수 있는 여러 가지 보안 기술들을 표준 문서로 정리하고 있는데, X9.62는 미국 디지털 서명 표준인 DSA[4]의 타원곡선 버전인 EC-DSA를 구현할 수 있도록 실제적이고 다양한 내용들을 담고 있으며 유한체로는 F(p)와 $F(2^m)$만을 고려해 두고 있다. 그 주된 내용의 목차를 간략히 살펴보면 다음과 같다.

- Finite field arithmetic over F(p) or $F(2^m)$
- Elliptic curve over F(p) or $F(2^m)$
- EC-DSA
- Annex

Annex에서는 number-theoretic algorithm을 비롯해서 mathematical background와 numerical example을 담고 있어서 EC-DSA를 구현하는 지침서의 역할을 할 수 있도록 구성되어 있다.

또한 X9.63에서는 session key를 생성하기 위한 타원곡선을 이용한 key agreement scheme들과 key transport scheme들에 대하여 설명하고 있다.

Ⅴ. 국내 표준화 활동

전자서명의 국내 표준인 KCDSA와 연결되는 표준으로서 EC-KCDSA를 표준화하기 위하여 1999년 3월 EC-KCDSA의 표준 초안이 작성되었다. 이어 같은 3월에는 EC-KCDSA의 표준화를 위한 1차 회의가 정보보호센터에서 있었다. 이 회의에서는 표준 초안에 대한 설명과 이에 대한 토의가 이루어졌다. Q의 정의와 그 길이와 해쉬 함수의 출력 길이에 대한 토의가 이루어졌으며 메시지 등을 표현하는 데이터 타입에 대한 논의가 이루어졌다. 토의의 결론으로 Q는 160비트 이상의 소수이고, 그 비트 길이와 해쉬 코드의 길이는 160비트 이상이면 되는 것으로 정하였으며 메시지와 유한체의 원소를 표현하는 방법 등은 표준에서 정하지 않기로 하되 구현 예제 등에서는 표준 방법을 명시하는 것으로 결정되었다.

1999년 3월말에는 KIISC/SIS(통신정보보호학회 정보보호표준연구회)에서 2차 발표 및 회의가 이루어졌으며 이를 바탕으로 5월에는 표준안의 수정이 이루어졌다. 수정된 표준안에서는 서명의 생성과 검증 단계에서 도메인 변수 등을 확인하는 단계를 추가하고 $kG = (x_1, y_1)$을 계산하여 $r = h(x_1||y_1)$를 얻는 과정에서 필요한 $x_1||y_1$를 얻는 방법과 $E = (R \text{ XOR } H) \bmod Q$를 계산하는 방법을 명시하였다. 또한 32비트 p와 m=7을 사용하여 타원곡선 $E(F(p^m))$의 예제를 m=5의 예제로 바꾸었으며, $E(F(2^m))$, E(F(p))의 예제를 추가하였다.

10월에 있었던 EC-KCDSA 한국 표준(안) 검토 회의에서는 유한체와 유한체에 따른 타원곡선의 표기를 각각의 표기인 F(p), $F(2^m)$, $F(p^m)$을 사용하는 것이 아니라 F(q)로 정의하고 q의 형태를 여러 가지로 설명하는 방법으로 하고, 이에 따라서 타원곡선의 정의도 변경하는 것으로 토의가 이루어졌다. 또한 EC-KCDSA를 구현하기 위해 필요한 데이터 타입(byte string, bit string, integer, finite field element 등)들 간의 변형에 대해서 명확히 정하자는 데 대한 토의가 이루어졌다.

12월에 있었던 회의에서는 Cert_Data에 대한 정의를 명확히 하여, 특히 $H = h(Z||M)$, $Z = h(\text{Cert_Data})$를 계산함에 이득이 있도록 하자는 의견에 대한 토의가 이루어졌으며, EC-KCDSA의 알고리즘 자체를 수정할 경우에 국제 표준과의 연계성을 어떻게 유지할 것인가에 대한 토의가 이루어졌다.

금년 5월에 정보보호학회 표준화연구회에서 여름중으로 표준안을 완성하여 TTA에 상정하기로 하였다.

Ⅵ. 결 론

본 고에서는 타원곡선 알고리즘의 표준화 동향을 알아보기 위하여 국내외 표준문서들인 ISO/IEC 문

서와 IEEE P1363, ANSI X9.62/9.63 및 국내 표준화 활동에 대해서 살펴보았다.

ISO/IEC/JTC1/SC27/WG2 15946 Information technology - Security techniques - Cryptographic techniques based on elliptic curves 문서를 바탕으로 타원곡선 알고리즘의 국제 표준을 자세히 살펴보았는데, 표준화 초기 과정에서부터 한국의 많은 참여가 있었음을 알 수 있었다. 특히, Part2 : Digital signatures 문서에서는 한국의 표준(안)인 EC-KCDSA가 국제표준에 포함되는 등의 큰 성과가 있었다. 이 15946-2는 2000년 4월 런던회의에서 CD단계에서 DIS(Draft International Standard)단계로 격상하기로 하였고 금년 여름에 투표를 할 예정이다. 따라서 빠르면 2001년 4월, 늦어도 2002년 4월에는 국제표준으로 한국의 EC-KCDSA가 국제표준이 될 것이다.

참 고 문 헌

[1] W.Diffie and M.Hellman, "New direction in cryptography", IEEE Transaction on Information Theory, 22, pp.644-654, 1976

[2] N.Koblitz, "Elliptic curve cryptosystems", Math. comp., 48, pp.203-209, 1987

[3] V.Miller, "Uses of elliptic curves in cryptography," Advances in cryptology - Crypto'85, pp.417-426, 1986

[4] NIST, Digital signature standard, FIPS PUB 186, 1994

[5] IEEE P1363(D13) : Standard Specifications For Public-Key Cryptography, http://grouper.ieee.org/groups/1363/draft.html, Nov., 12, 1999

[6] P.L.Montgomery, "Speeding the Pollar and Elliptic Curve Methods for Factorizations", Math. of Comp., 48, pp.243-264, 1987

〈著者紹介〉

이 필 중 (Pil Joong Lee)

1974년 2월 : 서울대학교 전자공학과 졸업
1977년 2월 : 서울대학교 전자공학과 석사 졸업
1982년 6월 : U.C.L.A. System Science, Engineer
1985년 6월 : U.C.L.A. Electrical Engineering, Ph.D.
1980년 6월~1985년 8월 : Jet Propulsion Laboratory, Senior Engineer
1985년 8월~1990년 2월 : Bell Communications Research, M.T.S.
1990년 2월~현재 : 포항공과대학교 전기전자공학과 교수