타원곡선 상에서 Diffie-Hellman 문제에 기반한 키 합의 프로토콜

송보연, 김광조
한국정보통신대학원
e-mail : bysong@icu.ac.kr, kkj@icu.ac.kr

Key Agreement Protocol based on Diffie-Hellman Problem over Elliptic Curve

Boyeon Song, Kwangjo Kim
Information and Communications University

요 약

본 논문은 타원곡선 상에서 Diffie-Hellman 문제를 기반으로 하는 세 가지 키 합의 프로토콜을 제안한다. 먼저, MTI/A0, Unified Model, MQV 와 같은 기존의 키 합의 프로토콜들보다 안전하고 효율적인 인증된 키 합의 프로토콜을 제안하고, 다음으로 메시지 인증 부호를 추가해서 3 회의 메시지로 이루어지는 키 공유 확인이 가능한 인증된 키 합의 프로토콜을 설계한다. 마지막으로 메시지 수는 2 회이면서 일방향으로 키 확인이 가능한 인증된 키 합의 프로토콜을 제안한다.

1. 서론

공개된 통신로에서 암호 통신을 하기 위하여 둘 이상의 실체는 암· 복호에 사용될 키를 사전에 공유하는 키 확립 과정을 수행하게 된다. 키 확립 프로토콜은 한 쪽 실체가 키를 생성하여 다른 실체에게 안전하게 전달하는 키 전송 프로토콜과, 양쪽 실체가 상호 작용해서 주고 받은 정보로 키를 생성하는 키 합의 프로토콜로 크게 나누어 생각할 수 있다[2,4,11].

본 고는 타원곡선 상에서 Diffie-Hellman 문제의 어려움에 기반[6]하는 두 실체로 구성된 키 합의 프로토콜을 논의한다. 본 고의 구성은 2 장에서는 키 합의 프로토콜의 요구사항을 기술하고, 3 장에서는 MTI/A0, Unified Model 과 MQV 프로토콜에 대해 개략적으로 설명하며, 4 장에서는 새로운 키 합의 프로토콜들을 제안한다. 5 장에서는 기존의 프로토콜과 제안한 프로토콜을 비교 분석한다. 마지막으로 6 장에서는 본 연구에 대해 결론을 맺고 향후과제를 제시한다.

2. 키 합의 프로토콜의 요구사항

키 합의 프로토콜에서 만족해야 할 안전성과 성능에 대한 요구 사항을 살펴보면 다음과 같다[3,9,11].

2.1. 안전성 요구사항

1) 기본적 안전성 요구사항

A 와 B 는 합법적인 두 실체라고 본다.

- 함축적 키 인증성(IKA, Implicit Key Authentication): 실체 A 가 B 이외에 어느 누구도 공유키를 생성할 수 없다는 확신을 가질 경우 A 에게 B 에 대한 IKA 를 제공한다고 말한다. A 와 B 가 상호 IKA 를 제공하는 프로토콜을 AK(Authenticated Key agreement) 프로토콜이라고 한다.
- 명시적 키 인증성(EKA, Explicit Key Authentication): 실체 B 가 공유키를 계산하여 소유하는 것을 A 가 확신할 수 있을 경우 A 에게 B 에 대한 EKA 를 제공한다고 말한다. A 와 B 가 상호 EKA 를 제공하는 프로토콜을 AKC(Authenticated Key agreement with key Confirmation) 프로토콜이라고 한다.

2) 추가적 안전성 요구사항

- 알려진 키에 대한 안전성(Known-Key security): 이전의 다른 세션키(키 합의 프로토콜의 각 세션에서 생성하는 유일한 비밀키)들이 공격자에게 노출되었을 경우에도 프로토콜의 안전성이 보장됨을 말한다.
- 전향적 보안성(Forward Secrecy): 하나 또는 둘 이상의 실체들의 장기적인 개인키가 노출되었을 경우에도 이전에 합의한 세션키의 안전성이 제공된다.
- 키 위장(Key-Compromise Impersonation)에 대한 안전성: 실체 A 의 장기적인 개인키가 공격자에게 노출되었을 경우 공격자가 A 에게 다른 실체 B 인 것처럼 위장할 수 없는 안전성을 말한다.
- 미지의 키 공유(Unknown Key-Share)에 대한 안전성: 실체 A 는 실체 B 와 키를 공유하고 있다고 믿고 있지만, B 는 공격자 E 와 키를 공유하고 있다고 믿게 하는 공격에 대해 안전한 경우를 말한다[5].

이하 K-KS, FS, K-CI, UK-S 로 표시한다.

2.2 성능 요구사항

- 메시지 교환 횟수 최소화.
- 전송되는 비트 수 최소화.
- 산술적 계산량 최소화.
- 온라인 계산량 감소를 위해 사전계산 가능성

3. 기존의 AK 프로토콜

이 장에서는 AK 프로토콜로서 Diffie-Hellman 문제에 안전성을 둔 MTI/A0, Unified Model, MQV 프로토콜을 비교하기 위하여 타원곡선 상의 프로토콜로 확장하여 간단히 기술한다. 먼저, 프로토콜에 포함되는 양 실체가 공통적으로 사용하는 타원곡선 매개변수와 각 실체의 키 쌍에 대해 설명한다.

도메인 매개변수는 표수(characteristic)가 p 인 유한체 F_q 상에서 정의된 타원곡선 E 와 위수가 n 인 기저점 $P \in E(F_q)$로 구성된다[9]. 실체의 개인키는 [1, n-1]에서 임으로 선택한 d 이고, 공개키는 타원곡선의 점 Q=dP 이며, 키 쌍은 (Q,d)가 된다.

기저점 P 에 대하여 aP 와 bP 가 주어졌을 때 abP 를 구하는 것을 타원곡선 상의 Diffie-Hellman 문제라 하며, 통상 유한체 상에서 정의된 Diffie-Hellman 문제보다 어렵다고 알려져 있어 키 비트 당 보다 많은 안전성을 보장하는 장점이 있다.

실체 A 는 (W_A, w_A)로 표시하는 장기키 쌍과 프로토콜의 각 실행에서 생성되는 임시키 쌍 (R_A, r_A)을 가지게 된다[9]. 이하 장기 공개키는 공개키 증명서에 의해 교환된다고 가정한다. $Cert_A$는 A 의 공개키 증명서를 나타낸다. UK-S 공격을 피하기 위해 CA 는 A 가 장기 공개키에 대응하는 장기 비밀키를 소유함을 검증할 수 있다고 가정한다.

실체들이 제시하는 공개키 Q 를 사용하기 전에 Q 가 점 P 에 의해 생성된 부분군의 유한한 점이라는 것을 검증해야 한다. 이 과정을 공개키 검증[3]이라고 하고 공개키 $Q=(x_Q, y_Q)$ 에 대해 다음과 같은 검증을 할 수 있다[9]: $Q \neq O$; $x_Q, y_Q \in F_q$; Q 는 방정식 E 의 점; $nQ = O$ (O: 무한원점). 계산적으로 비용이 많이 드는 스칼라 곱셈을 줄이기 위해 4 번째 단계를 생략한 과정을 축약된(embedded) 공개키 검증이라고 한다.

이 장에서는 검증 과정을 생략하고 설명한다.

3.1 MTI/A0

이 프로토콜은 Matsumoto, Takashima, Imai[14]가 IKA 를 제공하도록 1986 년에 제안한 프로토콜이다.

[프로토콜 1]

① A 는 $r_A \in_R [1, n-1]$를 선택하고, $R_A = r_A P$를 계산해서, B 에게 R_A, $Cert_A$를 보낸다.

② B 는 $r_B \in_R [1, n-1]$를 선택하고, $R_B = r_B P$를 계산해서, A 에게 R_B, $Cert_B$를 보낸다.

③ A 는 $K = w_A R_B + r_A W_B$를 계산한다.

④ B 는 $K = w_B R_A + r_B W_A$를 계산한다.

⑤ 공유하는 비밀정보는 점 K 이다.

A 와 B 는 비밀정보 $K = (r_A w_B + r_B w_A)P$를 공유하고, FS 를 제공하지 않는 약점이 있다.

3.2 Unified Model

Ankney, Johnson, Matyas[1]가 1995 년에 제안했고 ANSI X9.42[12], ANSI X9.63[13], IEEE P1363[14]에 표준안으로 제출되어 있는 AK 프로토콜이다.

기호 ||는 연접을 의미한다.

[프로토콜 2]

① A 는 $r_A \in_R [1, n-1]$를 선택하고, $R_A = r_A P$를 계산해서, B 에게 R_A, $Cert_A$를 보낸다.

② B 는 $r_B \in_R [1, n-1]$를 선택하고, $R_B = r_B P$를 계산해서, A 에게 R_B, $Cert_B$를 보낸다.

③ A 는 $Z_s = w_A W_B$, $Z_e = r_A R_B$를 계산한다.

④ B 는 $Z_s = w_B W_A$, $Z_e = r_B R_A$를 계산한다.

⑤ 공유하는 비밀정보는 점 $K = kdf(Z_s \| Z_e)$ 이다.

(kdf:키유도함수로 해쉬함수나 대칭키암호기법 이용)

A 와 B 는 $K = kdf(w_A w_B P \| r_A r_B P)$를 공유하고 K-CI 안전성을 제공하지 않는다.

3.3 MQV

1998 년 Menezes, Qu, Vanstone 등[9]에 의해 제안되었으며 ANSI X9.42, ANSI X9.63, IEEE P1363 에 표준안으로 제출되어 있다.

여기에서 사용되는 기호 $\overline{Q}$는 다음과 같이 정의된다. Q 의 x 좌표를 x 라고 했을 때 $\overline{x}$는 x 의 이진 표시(binary representation)로부터 얻어지는 정수라고 하자. 그 때 $\overline{Q} = \overline{x} \bmod 2^{\lceil f/2 \rceil} + 2^{\lceil f/2 \rceil}$이고, 이때 f 는 n 의 비트 길이이다. $\overline{Q} \bmod n \neq 0$ 임을 주의하자[9].

[프로토콜 3]

① A 는 $r_A \in_R [1, n-1]$를 선택하고, $R_A = r_A P$를 계산해서, B 에게 R_A, $Cert_A$를 보낸다.

② B 는 $r_B \in_R [1, n-1]$를 선택하고, $R_B = r_B P$를 계산해서, A 에게 R_B, $Cert_B$를 보낸다.

③ A 는 $s_A = r_A + \overline{R_A} w_A \bmod n$를 계산하고 $K = s_A(R_B + \overline{R_B} W_B)$를 계산한다.

④ B 는 $s_B = r_B + \overline{R_B} w_B \bmod n$ 를 계산하고 $K = s_B(R_A + \overline{R_A} W_A)$를 계산한다.

⑤ 공유하는 비밀정보는 점 K 이다.

A 와 B 는 비밀정보로 $K = s_A s_B P$, 즉 $K = (r_A r_B + r_A w_B \overline{R_B} + r_B w_A \overline{R_A} + w_A w_B \overline{R_A R_B})P$를 공유한다.

이 프로토콜은 UK-S 공격이 가능하다는 것을 Kaliski 가 최근에 제시했다[8].

4. 새로운 키 합의 프로토콜

이 장은 새로운 AK 프로토콜과 AKC 프로토콜, 그리고 일방향 AKC 프로토콜을 제안한다.

4.1 AK 프로토콜

[프로토콜 4]

① A 는 $r_A \in_R [1, n-1]$를 선택하고, $R_A = r_A P$를 계산해서, B 에게 R_A, $Cert_A$를 보낸다.

② B 는 $r_B \in_R [1, n-1]$를 선택하고, $R_B = r_B P$를 계산해서, A 에게 R_B, $Cert_B$를 보낸다.

③ A 는 R_B에 대해 축약된 공개키 검증을 수행한다. 타당하지 않으면 프로토콜을 실패로 종료하고, 그렇지 않으면 $K = r_A W_B + (w_A + r_A) R_B$를 계산한다. $K = O$이면 프로토콜을 종료한다.

④ B 는 R_A에 대해 축약된 공개키 검증을 수행한다. 타당하지 않으면 프로토콜을 실패로 종료하고, 그렇지 않으면 $K = r_B W_A + (w_B + r_B) R_A$를 계산한다. $K = O$이면 프로토콜을 종료한다.

⑤ 공유하는 비밀정보는 점 K 이다.

$K = O$인지를 검사함으로 K 가 유한한 점임을 보증한다[9]. A 와 B 는 비밀정보 $K = (r_A w_B + r_B w_A + r_A r_B) P$를 공유하게 된다.

1)안전성 분석: 안전성이 B-R 과 같은 분산 계산 모델[2,4]에서 형식적인 방법으로 증명된 것은 아니지만 경험적 논증으로 상호 IKA 를 제공한다고 볼 수 있다.

- 알려진 키 공격에 대한 안전성 (K-KS): 공격자가 다른 세션키들을 알고 있어도 임시 비밀키 r_A, r_B를 모르면 $r_A w_B P$, $r_B w_A P$, $r_A r_B P$를 알기 어려우므로 프로토콜의 안전성에 영향을 줄 수 없다.
- 전향적 보안성 (FS): 장기 비밀키 w_A 또는 w_B가 노출되고 R_A, R_B를 알게 되더라도 임시 비밀키 r_A, r_B를 모르면 Diffie-Hellman 문제에 의해 $r_A P, r_B P$로부터 $r_A r_B P$를 계산하기 어렵다.(그러나 FS 는 모든 세션키에 대해 EKA 가 제공될 때 확실하게 보증된다[3].)
- 키 위장에 대한 안전성 (K-CI): A 의 장기 비밀키 w_A 가 노출되었을 경우 $w_A, r_B, r_A P, r_B P, w_A P, w_B P$ 을 알고 있더라도 w_B를 알지 못하면, 공격자는 B 처럼 $r_A w_B P$을 계산할 수 없는 Diffie-Hellman 문제에 봉착하게 된다. 즉, 공격자는 B 처럼 위장할 수 없다.
- 미지의 키 공유에 대한 안전성 (UK-S): 앞에서 가정한 대로 CA 는 A 가 장기 공개키에 대응하는 장기 비밀키를 소유하고 있음을 검증할 수 있다고 했으므로 UK-S 공격을 막을 수 있다

2)성능 분석: 각 실체가 계산해야 할 스칼라 곱셈은 3 회이다. 실체 A 의 경우는 $r_A P$, $r_A W_B$와 $(w_A + r_A) R_B$를 계산한다. 만약 실체가 미리 알 수 있는 자신의 장기키와 임시키, 상대방 실체의 장기 공개키를 포함하는 값을 사전에 계산한다면 온라인 상에서 실체 당 계산해야 할 스칼라 곱셈은 1 회로 줄게 된다. 즉, A 의 경우 $(w_A + r_A) R_B$만 계산하면 된다.

4.2 AKC 프로토콜

[프로토콜 4]는 메시지 인증 부호(MAC)를 부가함으로 AKC 프로토콜로 확장할 수 있다. 여기에서 사용하는 H_1, H_2는 서로 독립적인 해쉬함수이다.

[프로토콜 5]

① A 는 r_A를 선택하고, B 에게 R_A, $Cert_A$ 를 보낸다.

② (1) B 는 R_A에 대해 축약된 공개키 검증을 한다.
(2) r_B를 선택하고 R_B를 계산한다.
(3) $K = r_B W_A + (w_B + r_B) R_A$를 계산한다.
(4) 점 K 의 x 좌표 z 를 사용해서 $k = H_1(z)$ 와 $k' = H_2(z)$를 계산한다.
(5) $MAC_{k'}(2, B, A, R_B, R_A)$를 계산하고 이 값을 R_B, $_B$와 함께 A 에게 보낸다.

③ (1) A 는 R_B에 대해 축약된 공개키 검증을 한다.
(2) $K = r_A W_B + (w_A + r_A) R_B$를 계산한다.
(3) A 는 점 K 의 x 좌표 z 를 사용해서 $k = H_1(z)$ 와 $k' = H_2(z)$를 계산한다.
(4) $MAC_{k'}(2, B, A, R_B, R_A)$를 계산하고 B 가 보낸 값과 같은지 검증한다.
(5) $MAC_{k'}(3, A, B, R_A, R_B)$를 계산하고 이 값을 B 에게 보낸다.

④ B 는 $MAC_{k'}(3, A, B, R_A, R_B)$를 계산하고 A 가 보낸 값과 같은지 검증한다.

⑤ 세션키는 k 이다.

1)안전성 분석: 키 확인 기능을 제공함으로 상호 EKA 를 만족한다. 즉, IKA, EKA 와 K-KS, FS, K-CI, UK-S 안전성를 모두 제공한다.

2)성능 분석: MAC 을 추가로 계산해야 하지만 MAC 은 효과적으로 계산될 수 있기 때문에 계산량에 큰 영향을 주지 않는다. 메시지 수는 3 회로 증가한다.

4.3 일방향 AKC 프로토콜

여기서는 2 회의 메시지로 상호 실체 인증과 여러 안전성 요구사항을 제공하는 프로토콜을 설계한다.

[프로토콜 6]

① A 는 r_A를 선택하고, R_A와 서명값 $S_A(R_A, A)$를 계산해서, B 에게 R_A, S_A, $Cert_A$를 보낸다.

② (1) B 는 R_A에 대해 축약된 공개키 검증을 한다.
(2) B 는 A 의 공개키를 이용하여 S_A를 검증한다. 만약 검사가 실패하면, 프로토콜을 종료한다.
(3) r_B를 선택하고 R_B를 계산한다.
(4) $K = r_B W_A + (w_B + r_B) R_A$를 계산한다.
(5) B 는 점 K 의 x 좌표 z 를 사용해서 $k = H_1(z)$ 와 $k' = H_2(z)$를 계산한다.
(6) $MAC_{k'}(B, A, R_B, R_A)$를 계산하고 이 값을 R_B, $Cert_B$와 함께 A 에게 보낸다

③ (1) A 는 R_B에 대해 축약된 공개키 검증을 한다.
(2) $K = r_A W_B + (w_A + r_A) R_B$를 계산한다.

(3) A 는 점 K 의 x 좌표 z 를 사용해서
$k = H_1(z)$ 와 $k' = H_2(z)$ 를 계산한다.
(4) $MAC_{k'}(B, A, R_B, R_A)$ 를 계산하고
B 가 보낸 값과 같은지 검증한다.

④ 세션키는 k 이다.

1)안전성 분석: 상호 IKA 를 제공하고 프로토콜 개시자에 대하여 EKA 를 제공한다. 추가적 안전성 요구사항을 모두 만족하고 상호 실체 인증을 제공한다.

2)성능 분석: 서명을 추가적으로 계산해야 하지만 부하가 가장 많은 메시지 수를 AKC 프로토콜보다 줄일 수 있다. 임시 공개키와 서명값은 사전 계산이 가능하므로 온라인 상에서는 서명 검증 계산량만 증가한다.

5. 키 합의 프로토콜의 비교

이 장에서는 지금까지 제시한 키 합의 프로토콜을 비교한다.

<표 1>은 두 실체가 정직하고 항상 프로토콜을 바르게 실행한다고 간주할 때 AK, AKC 프로토콜들이 제공한다고 보는 안전성 요구사항을 나타내고 있다.

<표 1> 키 합의 프로토콜이 제공하는 안전성 요구사항

	IKA	EKA	K-KS	FS	K-CI	UK-S
프로토콜 1	√√	×	√√	×	√√	√√
AKC KEA	√√	√√	√√	×	√√	√√
프로토콜 2	√√	×	√?	√?	×	√√
AKC U.M.	√√	√√	√√	√√	×	√√
프로토콜 3	√√	×	√√	√?	√√	×
AKC MQV	√√	√√	√√	√√	√√	√√
프로토콜 4	√√	×	√√	√?	√√	√√
프로토콜 5	√√	√√	√√	√√	√√	√√
프로토콜 6	√√	√I	√√	√√	√√	√√

√√: 실체가 프로토콜을 개시하는지에 관계없이 확실성 제공.
√I: 프로토콜의 개시자에게 확실성 제공.
√?: 특별한 조건에 따라 확실성 제공[3].
(FS 의 경우 모든 세션키에 대해 EKA 가 가능하면 확실성 보증)
×: 이 프로토콜에서 확실성이 제공되지 않음.

<표 2>는 키 합의 프로토콜을 계산할 때 요구되는 스칼라 곱셈의 횟수를 사전 계산하기 전(장기 비밀키와 임시 공개키 값을 포함하는 양에 대해)과 사전 계산 후로 나누어 비교한 것이다.

<표 2> 프로토콜 계산 시 필요한 스칼라 곱셈의 횟수

스칼라 곱셈	사전계산 전	사전계산 후
프로토콜 1 (KEA)	3	1
프로토콜 2 (U.M.)	3	1
프로토콜 3 (MQV)	2.5	1.5
프로토콜 4,5 (제안)	3	1

<표 1>에서 비교한 대로 [프로토콜 4]는 다른 AK 프로토콜보다 많은 안전성을 제공한다. [프로토콜 5]는 앞장에서 기술한 모든 안전성 요구사항을 만족한다. 성능면에서 보면 <표 2>에서 나타난 바와 같이 사전 계산 후 온라인 상에서 필요한 스칼라 곱셈 수가 AKC 인 MQV 보다 적다. [프로토콜 6]은 EKA 를 일방향으로만 제공하지만 이후 암호 통신을 통해 상호 EKA 가 가능하기 때문에 [프로토콜 5]보다 안전성에는 큰 차이가 없고 성능면에서는 효율적이다.

6. 결론

본 고에서는 IKA, EKA 와 K-KS, FS, K-CI, UK-S 에 대한 안전성을 고려하여 많은 안전성 요구사항을 제공하도록 타원곡선 상에서 AK 프로토콜을 제안하였다. 다음으로 MAC 을 추가하여 앞에서 논의한 모든 안전성 요구사항을 만족하는 AKC 프로토콜을 설계하였다. 이는 MTI/A0 와 Unified Model 프로토콜보다 안전성 요구사항을 많이 제공하고 성능면에서는 사전 계산 시 MQV 프로토콜보다 온라인 상 계산량이 적게 든다. 또 다른 하나로 안전성은 AKC 프로토콜과 비슷하게 제공하면서 2 회의 메시지를 가지는 일방향 AKC 프로토콜을 제안하였다.

본 고에서는 제안한 프로토콜들을 비형식적인 방법으로 경험적 논증에 의해 분석해보았지만, B-R 과 같은 분산 환경에서 형식적인 방법으로 증명 가능한 안전성임을 보이는 것이 추후 연구 과제이다.

참고문헌

[1] R. Ankney, D. Johnson and M. Matyas, "The Unified Model", contribution to X9F1, October 1995.

[2] M. Bellare, P. Rogaway, "Entity Authentication and Key Distributions - the Three Party Case", Advances in Cryptology-Crypto '93, LNCS 773, pp232-249, 1994.

[3] S. Blake-Wilson, A. Menezes, "Authenticated Diffie-Hellman Key Agreement Protocols", Proceedings of the 5th Annual Workshop on Selected Areas in Cryptography (SAC '98), *Lecture Notes in* Computer Science, 1556, pp339-361, 1999.

[4] S. Blake-Wilson, C. Johnson, A. Menezes, "Key Agreement Protocols and their Security Analysis", Proceedings of the sixth IMA International Conference on Cryptography and Coding, LNCS 1355, pp30-45, 1997.

[5] S. Blake-Wilson, A. Menezes, "Unknown Key-Share Attacks on the Station-To-Station (STS) Protocol", Technical report CORR 98-42, University of Waterloo, 1998.

[6] W. Diffie, M. E. Hellman, "New Directions in Cryptography", IEEE Transactions on Information Theory, 22, pp644-654, 1976.

[7] D. Johnson, Contribution to ANSI X9F1 working groups, June 1997.

[8] B. Kaliski, Contribution to ANSI X9F1 and IEEE P1363 working groups, June 1998.

[9] L. Law, A. Menezes, M. Qu, J. Solinas and S. Vanstone, "An Efficient Protocol for Authenticated Key Agreement Protocol", Technical report CORR 98-05, University of Waterloo, Canada, March 1998.

[10] T. Matsumoto, Y. Takashima and H. Imai, "On Seeking Smart Public-Key Distribution Systems", The Transactions of the IECE of Japan, E69, pp99-106, 1986.

[11] Menezes, P. van Oorschot, S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1997

[12] ANSI X9.42, *Agreement of Symmetric Algorithm Keys using Diffie-Hellman*, working draft, May 1998.

[13] ANSI X9.63, *Elliptic Curve Key Agreement and Key Transport Protocols*, working draft, July 1998.

[14] IEEE P1363, *Standard Specifications for Public-Key Cryptography*, working draft, July 1998.