능동적 공격자 환경에서의 자체인증 공개키에 기반한 키 분배 프로토콜의 안전성 분석

정회원 양 형 규*

The Security analysis of Self-certified public key based Key agreement protocols against Active Attacks

HyungKyu Yang* *Regular Members*

요 약

Girault는 자체 인증 공개키(self-certified public key)의 개념과 함께 이를 사용한 키 분배 프로토콜을 제안하였고, 후에 Rueppel과 Oorschot는 이를 변형한 프로토콜들을 제안하였다 자체인증 공개키에 기반한 키 분배 프로토콜은 사용자가 자신의 비밀키를 직접 선택하므로 개인식별 정보에 기반한 방식의 문제점으로 지적되었던 신뢰센터가 임의의 사용자로 위장할 수 있는 문제를 해결할 수 있고, 또한 메모리와 계산량을 감소시킬 수 있다는 장점이 있다 그러나, 키 분배 프로토콜의 안전성에 대한 구체적인 증명은 아직까지 미흡한 실정이다 본 논문에서는 지금까지 제안된 자체인증 공개키에 기반한 키분배 프로토콜에 대한 능동적 공격자 환경에서의 구체적인 안전성 분석을 수행하고자 한다. 본 논문에서 고려하는 공격은 active impersonation 공격, key-compromise impersonation 공격, forward secrecy, known key security이며, 안전성 증명에는 수학적 귀착 이론을 이용한다

Key Words : Self-certified public key, Key agreement, Active impersonnation, Reducibility

ABSTRACT

Girault proposed a key agreement protocol based on his new idea of self-certified public key Later Rueppel and Oorschot showed variants of the Girault scheme All of these key agreement protocols inherit positive features of self-certified public key so that they can provide higher security and smaller communication overhead than key agreement protocols not based on self-certified public key Even with such novel features, rigorous security analysis of these protocols has not been made clear yet. In this paper, we give rigorous security analysis of key agreement protocols based on self-certified public key We use reduction among functions for security analysis and consider several kinds of active attacker models such as active impersonation attack, key-compromise impersonation attack, forward secrecy and known key security.

Ⅰ. 서 론

네트워크상에서 전송되는 디지털 정보들은 실제 생활의 문서에 비해 무단 절취, 위·변조 또는 정보의 파기 등과 같은 위협이 훨씬 증가하게 된다. 따라서 이러한 안전성과 관련된 문제를 해결하기 위해 암호 시스템의 사용이 점차 증가하고 있으며, 암호 시스템의 사용에 있어 안전하고 효율적인 키의 분배는 가장 필수적인 구성 요소이다.

인증서에 기반한 공개키 암호 시스템을 이용하는 키 분배 프로토콜은 1976년 Diffie와 Hellman에 의해 처음으로 제안되었으며[2], 그 후에 Girault는 별도의 공개키 인증서를 필요로 하지 않는 자체 인증 공개키의 개념과 이를 이용한 키 분배 프로토콜을 제안하였

* 강남대학교 컴퓨터미디어공학부(hkyang@kangnam ac kr)
논문번호 #KICS2004-10-242, 접수일자 2004년 10월 24일

다[3].

자체인증 공개키에 기반한 방식은 공개키 자체가 인증서의 역할을 하므로 별도의 인증서를 저장하거나 검증할 필요가 없으며, 사용자가 자신의 비밀키를 직접 선택하므로 ID에 기반한 방식에 비해 높은 신뢰수준을 보장한다는 장점이 있다.

Girault는 자체인증 공개키의 개념과 함께 이를 이용하는 키 분배 프로토콜을 처음으로 제안하였으며, 그 후에 Rueppel과 Oorscurt는 이를 변형한 두개의 키 분배 프로토콜을 제안하였다[12]. 그러나 이 프로토콜들에 대한 자세한 안전성 분석은 수행되지 않았다.

Girault의 키 분배 프로토콜은 Z_n 상에서의 Diffie-Hellman 문제와 RSA 서명방식을 이용하므로[11], 프로토콜의 안전성은 암호학적 기반 문제와 연관성이 있다. 본 논문에서는 Girault가 제안한 키 분배 방식과 이를 변형한 방식들에 대해 수학적 귀착이론을 이용하여 자세한 안전성 분석을 수행한다

[5]에서는 귀착이론을 이용하여 Okamoto- Tanaka 키 분배 프로토콜에 대한 구체적인 안전성 분석을 수행하였으며[9], active impersonation 공격에 대해, 공격자가 상대방보다 먼저 메시지를 보내는 경우에는 RSA 문제를 푸는 어려움과 동치이고, 상대방이 먼저 메시지를 보내는 경우에는 Diffie- Hellman 문제를 푸는 어려움과 동치임을 증명하였다[1][14][7].

본 논문에서 고려할 능동적 공격자 모델은 active impersonation 공격자, forward secrecy에 대한 공격자, key compromise impersonation 공격자, known key security에 대한 공격자이다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 키 분배 프로토콜에서 사용하는 기본적인 용어들의 정의와 자체인증 공개키를 이용하는 키 분배 프로토콜의 동작 과정을 설명한다. 그리고 3장에서는 본 논문에서 사용하는 안전성 개념의 정의와 공격자 모델에 대해 설명하고, 4장에서는 [5][6]에서 사용한 수학적 귀착이론을 이용한 증명 방법을 이용하여 능동적 공격자 환경에서의 각각의 프로토콜의 안전성을 증명하고, 마지막으로 5장에서 결론을 맺는다.

Ⅱ. 관련 연구

1. 기본 용어

키 분배 프로토콜이란 안전하지 않은 통신로를 이용하여 두 사용자간에 안전하고 효율적인 비밀 세션키를 공유하기 위한 메커니즘으로써 키 분배 프로토콜에 관한 많은 연구가 진행되어 현재 다양한 프로토콜들이 제안되어 있다 이러한 키 분배 프로토콜은 크게 두 가지로 나눌 수 있다. 먼저, 사용자 A와 사용자 B가 공유하고자 하는 비밀 세션키를 어느 누구도 미리 결정하지 않고 두 사용자간의 합의에 의해 공유하는 키 동의(key agreement) 방식과 사용자 A가 세션키를 공유하고자 하는 사용자 B에게 비밀 세션키를 일방적으로 선택하여 안전하게 전송하는 키 전송(key transport)방식이 있다

또한, 키 분배 방식은 세션키 설정을 위해 사용하는 암호 방식에 따라, 관용 암호 시스템을 이용한 방식과 공개키 암호 시스템을 이용하는 방식으로 나눌 수 있다 관용 암호 시스템을 이용하기 위해서는 사전에 안전한 통신로를 이용하여 두 사용자간에 미리 비밀키를 공유해야하므로 개방된 통신로를 이용하는 컴퓨터 네트워크에 적용하는 데에는 어려움이 많다. 따라서, 공개키 암호 시스템을 이용한 키 분배 방식이 많이 사용되고 있다

본 논문에서 사용하는 키 분배 프로토콜에 관련된 용어의 정의는 다음과 같다[12].

- 개체 인증(Entity authentication) 키 분배 프로토콜에 참여하고 있는 상대방의 신원을 확인하는 것
- 키 확인 (key confirmation) : 키 분배 프로토콜에 참여한 합법적인 사용자가 자신이 의도한 상대방과 실제로 공통의 비밀 세션키를 공유하였음을 확인하는 것
- 묵시적 키 인증(implicit key authentication) : 실제 키의 소유 여부는 알려져 있지 않더라도 키 분배 프로토콜에 참여한 상대방만이 세션키를 계산할 수 있음을 보장하는 것
- Key freshness 세션마다 설정된 키가 바뀌는 것

2. Girault의 키 분배 프로토콜

자체인증 공개키란 공개키 암호 시스템에서 사용자의 공개키를 관리하는 방식으로, 인증서에 기반한 방식과 개인식별 정보에 기반한 방식의 중간 개념이라 할 수 있다. 이 방식은 사용자의 공개키가 곧 사용자의 인증서 역할을 하는 것으로, 별도의 인증서를 필요로 하지 않기 때문에 메모리 측면에서 효율적이며 사용자가 자신의 비밀키를 직접 선택하기 때문에 개인식별 정보에 기반한 방식이 갖는 문제점을 해결할 수 있다는 장점이 있다

Girault는 자체인증 공개키에 기반한 키 분배 프로토콜을 처음으로 제안하였으며[3], Rueppel과 Oorscurt

는 이를 변형한 두개의 프로토콜을 제안하였다[12]

이들 프로토콜은 사용자의 자체인증 공개키를 생성하는 과정과 사용자 A, B 사이의 세션키를 설정하는 키 분배 과정으로 구성되며, 세션키를 설정하기 위해 필요한 통신 회수에 따라 0-pass, 1-pass, 2-pass 프로토콜로 나눌 수 있다.

[자체인증 공개키 생성 과정]

신뢰센터(TA)은 두 개의 큰 소수 p, q를 선택하고 n, g, e를 생성한다. 여기서, n = p•q이고, g는 Z_n 상에서의 최대의 위수를 갖는 원소이며 $e \in Z^*_{\phi(n)}$ 이다.

① 사용자 A는 비밀키 s_A를 선택하고, $g^{-s_A} \bmod n$을 계산하여 TA에게 전송한다.

② TA는 사용자 A의 자체인증 공개키 P_A를 다음과 같이 계산하여 사용자 A에게 전송한다.

$$P_A \equiv (g^{-s_A} - ID_A)^d \bmod n$$

③ 사용자 A는 다음 식을 이용하여 P_A가 올바르게 생성되었는지 확인한다

$$P_A^e + ID_A \equiv g^{-s_A} \bmod n$$

(1) Girault의 0-Pass 프로토콜 (G0)

이 프로토콜은 사전에 공개정보가 유용하다는 가정하에 키 토큰의 교환 없이 키 동의를 수행하며, 사용자 A와 B는 다음과 같이 세션키를 생성한다

$$C \equiv (P_B^e + ID_B)^{s_A} \equiv (P_A^e + ID_A)^{s_B} \equiv g^{-s_A s_B} \bmod n$$

Girault의 0-pass 프로토콜은 사용자간의 통신 과정 없이 상대방의 공개키만을 이용하여 사용자 A와 사용자 B가 비밀 세션키를 공유한다 공개키 $P_A \equiv (g^{-s_A} - ID_A)^d \bmod n$와 $P_B \equiv (g^{-s_B} - ID_B)^d \bmod n$가 각각의 개인식별 정보를 포함함으로써 인증서의 역할을 하고 있음을 알 수 있다 이 프로토콜은 개체 인증과 키 확인 기능을 제공하지는 않으나, 고정된 비밀키를 이용하므로 묵시적 키 인증 기능을 제공한다 그러나 한번 공유된 세션키가 장기간 사용되므로 key freshness는 제공하지 못한다.

(2) Girault의 1-Pass 프로토콜 (G1)

사용자 A와 B가 세션키를 설정하는 과정은 다음과 같다

① 사용자 A는 랜덤수 r_A를 선택하고, $k_A \equiv g^{r_A} \bmod n$을 계산 사용자 B에게 전송한다.

② 사용자 A는 다음과 같이 세션키를 생성한다.

$$C \equiv (P_B^e + ID_B)^{-r_A} \equiv g^{r_A s_B} \bmod n$$

③ 사용자 B는 다음과 같이 세션키를 생성한다.

$$C \equiv (k_A)^{s_B} \equiv g^{r_A s_B} \bmod n$$

Girault의 1-Pass 프로토콜은 1번의 통신 과정을 통해서 사용자 A와 사용자 B가 비밀 세션키를 공유한다. 이 프로토콜은 개체 인증과 키 확인 기능을 제공하지는 않으나, 사용자 B만이 사용자 A와 같은 세션키를 계산할 수 있으므로 일방향의 묵시적 키 인증을 제공한다 또한 사용자 A만이 키 토큰에 자신이 생성한 랜덤수를 사용함으로써 일방향 key freshness를 제공한다

(3) Girault의 2-Pass 프로토콜 (G2)

사용자 A와 B가 세션키를 설정하는 과정은 다음과 같다.

① 사용자 A는 랜덤수 r_A를 선택하고, $k_A \equiv g^{-r_A} \bmod n$을 계산 사용자 B에게 전송한다

② 사용자 B는 랜덤수 r_B를 선택하고, $k_B \equiv g^{-r_B} \bmod n$을 계산 사용자 A에게 전송한다.

③ 사용자 A는 다음과 같이 세션키를 생성한다

$$C \equiv (k_B)^{s_A} (P_B^e + ID_B)^{r_A} \equiv (g^{-r_B})^{s_A} (P_B^e + ID_B)^{r_A} \equiv g^{-s_B r_A - s_A r_B} \bmod n$$

④ 사용자 B는 다음과 같이 세션키를 생성한다

$$C \equiv (k_A)^{s_B} (P_A^e + ID_A)^{r_B} \equiv (g^{-r_A})^{s_B} (P_A^e + ID_A)^{r_B} \equiv g^{-s_B r_A - s_A r_B} \bmod n$$

Girault의 2-Pass 프로토콜은 2번의 통신 과정을 통해서 사용자 A와 사용자 B가 비밀 세션키를 공유한다. 이 프로토콜은 개체 인증과 키 확인 기능을 제공하지는 않으나, 고정된 비밀키를 이용하므로 양방향 묵시적 키 인증 기능을 제공한다 또한 매 세션마다 키 토큰이 바뀌므로 양방향 key freshness를 제공한다

Ⅲ. 안전성 개념 및 공격자 모델

1 안전성 개념

본 논문에서 고려할 능동적 공격자 모델은 active impersonation 공격자, forward secrecy에 대한 공격자, key compromise impersonation 공격자, known key security에 대한 공격자이며 각각에 대한 정의는 다음과 같다

[정의 3 1] Active Impersonation(AI) attack
공격자가 자신을 임의의 다른 사용자로 위장하여 프로토콜에 참여하고, 정당한 사용자 A와 키 분배를 성공적으로 수행하는 경우에 active impersonation이 가능하다고 한다

[정의 3.2] Forward Secrecy(FS)
사용자 A와 B의 비밀키가 노출되더라도, 공격자가 두 사용자 사이에 설정된 과거 세션키를 계산할 수 없는 경우에 forward secrecy를 만족한다고 한다 forward secrecy는 노출되는 사용자의 키에 따라 다음과 같이 나눌 수 있다

- Half Forward Secrecy . 한 사용자의 비밀키가 노출된 경우에만 세션키가 안전
- Full Forward Secrecy . 두 사용자의 비밀키가 모두 노출된 경우에도 세션키 안전

[정의 3 3] Key-Compromise Impersonation(KCI) attack
사용자 A의 비밀키가 노출되었을 때, 공격자가 누구에게나 사용자 A로 위장할 수 있고 사용자 A에게 임의의 사용자 B로 위장할 수 있을 때 key-compromise impersonation 가능하다고 한다 그러나 공격자가 누구에게나 사용자 A로 위장할 수 있지만 사용자 A에게 임의의 다른 사용자로 위장할 수는 없는 경우에는 키 분배 프로토콜이 key-compromise impersonation resilience 특성을 갖는다고 한다.

[정의 3 4] Known Key Security(KKS)
두 사용자 A, B 사이의 과거 세션키가 노출되더라도 현재 세션키의 안전성에는 아무런 영향을 미치지 않는 경우에 KKS를 만족한다고 한다. KKS에 대한 공격은 다음과 같이 두 가지로 나눌 수 있다

- KKP(Known Key Passive) 공격 과거의 세션키와 전송 정보 및 현재 세션의 전송 정보를 이용하여 현재의 세션키를 획득하려는 공격 방법
- KKI(Known Key Impersonation) 공격 · 세션에 직접 참여 과거의 세션키와 전송 정보 그리고 현재 세션의 전송 정보를 이용하여 사용자 U에게 사용자 V로 위장하여 세션키를 설정하려는 공격 방법

2 공격자 모델

암호 프로토콜의 안전성을 증명하기 위해서는 대상이 되는 공격자 모델 크게 수동적 공격자(passive attacker)와 능동적 공격자(active attacker)로 나눌 수 있으며, 각각의 특징은 다음과 같다

- 수동적 공격자 : 프로토콜의 참가자와 실제로 통신에 참여하지 않고 두 참가자 사이의 통신 내용을 도청함으로써 공격을 수행하는 공격자
- 능동적 공격자 : 단순히 참가자들의 통신 내용을 도청하는 것뿐만 아니라 전송되는 메시지를 위·변조하거나 새로운 메시지를 삽입하거나 하는 등 실제 통신에 참여하는 보다 강력한 공격자

본 논문에서 고려하는 Girault의 키 분배 프로토콜에 대한 공격자의 동작 과정은 다음과 같다.

(1) 0-pass 프로토콜에 대한 active impersonation 공격자의 동작 과정

공격자 E는 프로토콜에 참가하여 자신을 정당한 사용자 A로 위장하고 사용자 B와 세션키를 설정하기 위해 다음과 같이 동작한다.

① 공격자 E는 k_A를 사용자 B에게 전송한다.
② 공격자 E는 k_A와 공개 정보를 이용하여 세션키를 계산한다

(2) 2-pass 프로토콜에 대한 active impersonation /key compromise/known key impersonation 공격자의 동작 과정

공격자 E는 프로토콜에 참가하여 자신을 정당한 사용자 B로 위장하고 사용자 A와 세션키를 설정하기 위해 다음과 같이 동작한다

① 공격자 E는 k_B를 사용자 A에게 전송하고 A로부터 k_A를 수신한다
② 공격자 E는 k_A k_E와 공개 정보/A의 비밀키/이전의 세션키를 이용하여 해당 세션키를 계산한다.

(3) Forward secrecy에 대한 공격자의 동작 과정

사용자 A와(또는) B의 비밀키를 아는 공격자는 현재 세션의 전송 정보들과 사전에 획득한 정보를 이용하여 해당 세션키를 계산한다

Ⅳ. 안전성 분석 결과

1 수학적 귀착

본 논문에서는 능동적 공격자에 대한 키 분배 프로토콜의 안전성을 증명하기 위해 함수들 사이에 수학적 귀착이론을 사용한다[5][13][15] 본 논문에서 사용하는 귀착(reducibility)의 종류는 다항식 시간 many-one 귀착, 다항식 시간 튜링 귀착, 다항식 시간 truth-table 귀착이며 각각에 대한 자세한 정의는 다음과 같다 .

[정의 4.1] 다항식 시간 many-one 귀착 . 만약 함수 F, G에 대해 F(x) = G(h(x))를 만족하는 다항식 시간에 계산 가능한 변환 함수 h가 존재하면, 함수 F는 G에 다항식 시간 many-one 귀착 가능(reducibility)하다고 하며 $F \leq_m^p G$ 라고 표시한다 그리고 역이 성립할 경우에는 다항식 시간 many-one 동치라 하고 $F \equiv_m^p G$라 표시한다

[정의 4 2] 다항식 시간 튜링 귀착 · 다항식 시간 튜링 머신 G에 대한 adaptive한 query를 주어 F를 계산할 수 있다면 F는 G에 다항식 시간 튜링 귀착 가능하다고 하며 $F \leq_T^{(e)p} G$라고 표시한다 이때, 모든 x ∈ {0, 1}*과 무한 비트 수열 r에 대해 $(t_{M(x,r)})^e \leq |x|$를 만족하는 e > 0가 존재하면 M은 expected 다항식 시간이라고 한다.

[정의 4 3] 다항식 시간 truth-table 귀착 다항식 시간 튜링 머신에 G에 대한 non-adaptive한 query를 주어 F를 계산할 수 있다면 F는 G에 다항식 시간 truth-table 귀착 가능하다고 하며 $F \leq_{tt}^p G$라고 표시한다. 특히, 최대한 k번의 query를 이용하여 귀착 가능한 경우에 $\leq_{k-tt}^p$라 표시한다

다항식 시간 truth-table 귀착은 다항식 시간 튜링 귀착의 특수한 경우라 할 수 있다 단지 다항식 능력을 갖는 오라클에 non-adaptive한 query를 통해 결과를 계산한다는 차이점이 있다. 여기에서 "non-adaptive한 query"의 의미는 query를 한번에 table로 오라클로 보내고 오라클로부터 응답을 얻음을 의미한다. 즉, 두 번째 query의 작성은 첫 번째 query의 결과가 고려되지 않는다는 것을 의미하고, 이것을 병렬적인 query라고 한다. 따라서 튜링 귀착보다 약한 정도의 귀착 방식이라 할 수 있다
앞에서 설명한 다항식 시간 귀착 개념사이에는 다음과 같은 상관관계가 성립한다

$$F \leq_m^p G \Rightarrow F \leq_{k-tt}^p G \Rightarrow F \leq_{tt}^p G \Rightarrow F \leq_T^p G$$

2 Active Impersonation(AI) 공격에 대한 안전성

G0 프로토콜은 세션키 설정에 사용자 사이에 별도의 통신이 필요하지 않으므로 공격자가 세션에 참여하여 정당한 사용자로 위장하려는 공격 자체가 불가능하다 그러나 G1 프로토콜은 일방향 묵시적 키 인증만을 제공하므로 프로토콜에 참여하는 사용자 B는 상대방의 신원을 확인할 수 있는 방법을 제공하지 않는다. 따라서, G1 프로토콜은 AI 공격에 대해 안전하지 않다 반면에, G2 프로토콜에서는 세션키를 생성하기 위해 사용자 A와 B의 비밀키를 모두 사용하므로 AI 공격에 대해 안전하다. 따라서, 본 절에서는 G2 프로토콜에 대한 AI 공격의 어려움이 Diffie-Hellman 문제를 푸는 것과 동치임을 증명한다.
먼저, Diffie-Hellman 문제와 G2 프로토콜에 대한 AI 공격자 함수에 대한 정의는 [정의 4.4]-[정의 4.6]과 같다.

[정의 4.4] DH(n, g, A, B)는 큰 소수 p, Z_n^*의 원시원소 g, $A \in Z_n^*$, $B \in Z_n^*$를 입력으로 하여 $C \equiv g^{ab} \bmod n$를 만족하는 $C \in Z_n^*$를 출력하는 함수이다 단, $A \equiv g^a \bmod n$, $B \equiv g^b \bmod n$이다.

[정의 4.5] $G2_{AI}$(n ,e ,g, ID_A, ID_B, P_A, P_B, k_A)

는 $n \in N_{>1}$, $e \in Z^*_{\phi(n)}$, $g \in Z^*_n$, ID_A, $ID_B \in Z_n$, P_A, $P_B \in Z_n$, $k_A \in Z^*_n$를 입력으로 하여 $C \equiv (k_B)^{s_A}(P_B^e + ID_B)^{r_A}$를 만족하는 $(C, k_B) \in Z_n \times Z_n$를 출력하는 함수이다. 단, 이러한 (C, k_B)가 존재하면 $P_B^e + ID_B \equiv g^{-s_B} \bmod n$에 대해 $P_A^e + ID_A \equiv g^{-s_A} \bmod n$, $\exists s_A \in Z_n$, $k_A \equiv g^{-r_A} \bmod n$을 만족한다.

[정의 4.5] $G2^*_{AI}$(n, e, g, ID_A, ID_B, P_A, P_B, k_A)는 $e \notin Z^*_{\phi(n)}$인 경우에 $\perp$를 출력하는 것을 제외하고는 $G2_{AI}$와 동일한 함수이다.

$G2$ 프로토콜에 대한 AI 공격의 어려움은 Diffie-Hellman 문제와 동치이며, 자세한 증명 과정은 [정리 4.1]과 같다

[정리 4.1] $G2^*_{AI} \equiv^p_T DH$

(증명)

① $G2^*_{AI} \le^p_{1-tt} DH$

먼저, 랜덤 수 $r_B \in_R Z_n$를 선택하고,

$G2^*_{AI}(n, e, g, ID_A, ID_B, P_A, P_B, k_A)$

$= (DH(n, g, P_B^e + ID_B, k_A^{-1}) \cdot (P_A^e + ID_A)^{r_B}, g^{-r_B})$

$= (g^{-s_B r_A - s_A r_B} \bmod n, g^{-r_B} \bmod n)$

② $DH \le^p_T G2^*_{AI}$

DH()의 입력이 (n, g, A, B)라 할 때, 홀수 $e \in Z^*_n \setminus \{1\}$와 s_A, P_A, $P_B \in Z_n$를 랜덤하게 선택한 후, $x \notin \perp$ 일때 까지 다음을 계산한다.

$G2^*_{AI}(n, e, g, g^{-s_A} - P_A^e, B^{-1} - P_B^e, P_{A,P_B}, A) = (x, y)$

출력된 결과는 어떤 $k_B \in Z^*_n$에 대해 (x, y) = $(k_B^{s_A} B^a, k_B)$을 만족하며, 다음을 계산할 수 있다.

$$DH(n, g, A, B) = x / y^{s_A} \equiv g^{ab} \bmod n$$

$Z^*_{\phi(n)}$에서 위의 식을 만족하는 e가 선택될 확률은 $\rho \ge 2\ln 2 / \ln(2n)$이며[4][6], 위의 알고리즘은 |n|에 대한 다항식 시간 반복 후에 정당한 값을 출력할 수 있게 된다.

3 Key-Compromise Impersonation (KCI) 공격에 대한 안전성

G0는 사용자 A의 비밀키가 노출된 경우에 공격자는 사용자 A에게 임의의 다른 사용자로 위장할 수 있고, G1은 일방향 프로토콜로 사용자 A에 대한 인증을 제공하지 못하므로 en 프로토콜은 Key compromise impersonation resilience 특성을 갖지 못한다

G2 프로토콜에 대한 KCI 공격자의 함수에 대한 정의는 [정의 4.6]-[정의 4.7]과 같다

[정의 4.6] $G2_{KCI}$(n, e, g, ID_A, ID_B, P_A, P_B, k_A, s_A)는 $n \in N_{>1}$, $e \in Z^*_{\phi(n)}$, $g \in Z^*_n$, ID_A, $ID_B \in Z_n$, P_A, $P_B \in Z_n$, $k_A \in Z^*_n$, $s_A \in Z_n$을 입력으로 하여 $C \equiv (k_B)^{s_A}(P_B^e + ID_B)^{r_A} \bmod n$을 만족하는 (C, k_B) $\in Z^*_n \times Z^*_n$를 출력하는 함수이다 단, 이러한 (C, k_B)가 존재하면 $P_B^e + ID_B \equiv g^{-s_B} \bmod n$에 대해 $\exists s_A \in Z_n$, $P_A^e + ID_A \equiv g^{-s_A} \bmod n$, $k_A \equiv g^{-r_A} \bmod n$, 을 만족한다.

[정의 4.7] $G2^*_{KCI}$(n, e, g, ID_A, ID_B, P_A, P_B, k_A, s_A)는 $e \notin Z^*_{\phi(n)}$인 경우에 $\perp$를 출력하는 것을 제외하고는 $G2_{KCI}$와 동일한 함수이다

G2 프로토콜에 대한 KCI 공격의 어려움은 DH 문제와 동치이며, 자세한 증명 과정은 [정리 4.2]와 같다

[정리 4.2] $G2^*_{KCI} \equiv^{ep}_T DH$

(증명)

① $G2^*_{KCI} \le^p_{1-tt} DH$

먼저 $r_B \in_R Z_n$를 랜덤하게 선택하고,

$G2^*_{KCI}(n, e, g, ID_A, ID_B, P_A, P_B, k_A, s_A)$

$= (DH(n, e, (k_A)^{-1}, P_B^e + ID_B) \cdot g^{-r_B s_1}, g^{-r_B})$

$= (g^{-r_A s_B - r_B s_A} \bmod n, g^{-r_B} \bmod n)$

② $DH \leq_T^{ep} G2^*_{KCI}$

Z_n에서 r_B, s_A, e, P_A, P_B를 랜덤하게 선택한 후, 다음을 계산한다

$G2^*_{KCI}(n, e, g, ID_A, B-P_B^e, P_A, P_B, A^{-1}, s_A)$

$= (g^{-r_B s_A} g^{ab}, g^{-r_B}) = (x, y)$ until $x \neq \perp$

$DH(n, e, A, B) = x / y^{s_A} = g^{ab} \bmod n$

Girault의 키 분배 프로토콜에서, 센터의 키 쌍은 RSA 형태지만 사용자의 자체인증 공개키로부터 비밀키를 구하는 것은 이산대수 문제에 기반한다. 그리고 세션키를 계산하는 데 두 사용자가 각각 생성한 랜덤 수가 포함되므로 공격자가 사용자 A의 비밀키를 알더라도 사용자 A가 생성한 랜덤 수를 모르므로 사용자 B로 위장 할 수 없다. 즉, 사용자 A의 비밀키가 주어지더라도 사용자 A에게 다른 임의의 사용자로 위장하는 것은 Diffie-Hellman 문제의 어려움과 동치이다.

4 Forward Secrecy(FS)에 대한 안전성

G0는 세션키가 항상 같은 형태이므로 forward secrecy를 제공하지 않고, G1은 일방향 프로토콜로 사용자 B의 비밀키가 노출된 경우에 forward secrecy를 만족하지 못한다. 그러나 사용자 A의 비밀키는 세션키 생성에 포함되지 않으므로 사용자 A의 비밀키가 노출된 경우에는 세션키는 안전하므로 Half Forward Secrecy를 제공한다. G1에 대한 Half forward secrecy 공격자 함수에 대한 정의는 [정의 4.8]과 같고 자세한 증명 과정은 [정리 4 3]과 같다.

[정의 4 8] $G1_{HFS}$(n ,e ,g, ID_B, P_B, k_A, s_A)는 $n \in N_{>1}$, $e \in Z^*_{\phi(n)}$, $g \in Z^*_n$, $ID_B \in Z_n$, $P_B \in Z_n$, $k_A \in Z_n$, $s_A \in Z_n$를 입력으로 하여 $C = (k_A)^{s_B} = (P_B^e + ID_B)^{-r_A} \bmod n$을 만족하는 $C \in Z_n$을 출력하는 함수이다. 단, 이러한 C가 존재하 면 $P_B = (g^{-s_B} - ID_B)^d \bmod n$, $r_A \in Z_n$, $k_A = g^{r_A} \bmod n$, $e \cdot d \equiv 1 \bmod \phi(n)$이다

[정리 4 3] $G1_{HFS} \equiv_m^p DH$

(증명)

① $G1_{HFS} \leq_m^p DH$

$G1_{HFS}(n, e, g, ID_B, P_B, k_A, s_A)$

$= DH(n, g, (P_B^e + ID_B)^{-1}, k_A)$

$= g^{s_B r_A} \bmod n$

② $DH \leq_m^p G1_{HFS}$

$x \in_R Z_n$에 대해,

$DH(n, g, A, B)$

$= G1_{HFS}(n, e, g, B - P_B^e, P_B, A^{-1}, x)$

$\equiv g^{ab} \bmod n$

G2는 두 사용자의 비밀키 s_A, s_B가 모두 노출된 경우에, 누구든지 전송 정보 k_A, k_B를 이용하여 다음과 같이 세션키를 계산할 수 있다

$C = (k_A)^{s_B} (k_B)^{s_A} = g^{-s_B r_A - s_A r_B} \bmod n$

따라서, G2는 forward secrecy를 만족하지 못한다 그러나 사용자 A, B 중 한 명의 비밀키가 노출된 경우에는 세션키가 안전하므로 역시 Half Forward Secrecy를 제공한다.

G2에 대한 Half forward secrecy 공격자 함수에 대한 정의는 [정의 4.9]-[정의 4.10]과 같고 자세한 증명 과정은 [정리 4.4]와 같다.

[정의 4.9] $G2_{HFS}$(n ,e ,g, ID_A, ID_B, P_A, P_B, k_A, k_B, s_A)는 $n \in N_{>1}$, $e \in Z^*_{\phi(n)}$, $g \in Z^*_n$, ID_A, $ID_B \in Z_n$, $P_A, P_B, \in Z_n$, $k_A, k_B, \in Z^*_n$, $s_A \in Z_n$를 입력으로 하여 $C = (k_B)^{s_A} (P_B^e + ID_B)^{r_A} = (k_A)^{s_B} (P_A^e + ID_A)^{r_B} \equiv g^{-s_B r_A - s_A r_B} \bmod n$을 만족하는 $C \in Z_n$을 출력하는 함수이다. 단, 이러한 C가 존재하면 $P_A^e + ID_A \equiv g^{-s_A} \bmod n$, $k_A \equiv g^{-r_A} \bmod n$, $P_B^e + ID_B \equiv g^{-s_B} \bmod n$, $k_B \equiv g^{-r_B} \bmod n$을 만족한다.

[정의 4.10] $G2^*_{HFS}$(n ,e ,g, ID_A, ID_B, P_A, P_B, k_A, k_B, s_A)는 $e \notin Z^*_{\phi(n)}$인 경우에 ⊥를 출력하는

것을 제외하고는 $G2_{HFS}$와 동일한 함수이다.

[정리 4 4] $G2^{*}_{HFS} \equiv^{ep}_{T} DH$

(증명)

① $G2^{*}_{HFS} \leq^{p}_{1-tt} DH$

$G2_{HFS}(n, e, g, ID_A, ID_B, P_A, P_B, k_A, k_B, s_A)$

$= DH(n, g, P_B^e + ID_B, k_A^{-1}) \cdot (k_B)^{s_A}$

$= g^{-r_A s_B} \cdot (g^{-r_B})^{s_A} \bmod n$

$\equiv g^{-r_A s_B - r_B s_A} \bmod n$

② $DH \leq^{ep}_{T} G2^{*}_{HFS}$

DH()의 입력이 (n, g, A, B)라 할 때, 홀수 $e \in Z_n^{*} \setminus \{1\}$와 $P_A, P_B \in Z_n$를 랜덤하게 선택한 후, $x \notin \perp$ 일때 까지 다음을 계산한다

$G2^{*}_{HFS}(n, e, g, g^{-e} - P_A^e, B^{-1} - P_B^e, P_A, P_B, A, g, e) = x$

따라서, 다음을 계산할 수 있다.

$DH(n, g, A, B) = x / g^e = g^{ab} \bmod n$

[정리 4 1]과 같이 위의 알고리즘은 |n|에 대한 다항식 시간 반복 후 정당한 값을 출력할 수 있다

5 Known-Key Security(KKS)에 대한 안전성

G0는 세션키가 항상 동일한 값을 가지므로 known-key passive(KKP), known-key impersonation(KKI)에 대해 안전하지 않고 G1은 과거의 세션키과 전송 정보가 노출되는 경우에 공격자가 이를 이용하여 사용자 A로 위장할 수 있으므로 KKI에 대해 안전하지 않다.

그러나 G1 프로토콜은 세션키를 생성하는데 사용자 A가 선택한 랜덤 수가 포함되므로 사용자가 서로 다른 랜덤 수를 사용하는 경우, 과거의 세션키와 전송 정보를 획득하더라도 현재의 세션키를 계산하는데 도움이 되지 않는다. 따라서, G1 프로토콜은 과거의 전송 정보와 세션키를 이용하여 현재의 세션키를 알아내려는 KKP 공격에 대해서는 안전하다

G2 프로토콜은 세션키를 생성하는데 사용자 A와 B가 선택한 랜덤 수가 포함되므로 과거의 세션키와 전송 정보를 획득하더라고 사용자들이 매 세션마다 서로 다른 난수를 사용하는 경우에, 현재의 세션키를 구하는데 아무런 도움이 되지 않는다 즉, 과거의 전송 정보와 세션키를 이용하여 현재의 세션키를 구하려는 KKP 공격자의 어려움은 아무런 정보도 주어지지 않은 수동적 공격자의 어려움과 동일하다

[정의 4.11] $G2_{KKP}$(n ,e ,g, ID_A, ID_B, P_A, P_B, k_A, k_B, k_A', k_B', C')는 $n \in N_{>1}$, $e \in Z^{*}_{\phi(n)}$, $g \in Z^{*}_{n}$, ID_A, $ID_B \in Z_n$, P_A, P_B, $\in Z_n$, $k_A, k_B, k_A', k_B' \in Z_n$, $C \in Z_n$를 입력하여 $C \equiv (k_B)^{s_A}(P_B^e + ID_B)^{r_A}$를 만족하는 $C \in Z_n$를 출력하는 함수이다 단, 이러한 C가 존재하면

$P_A^e + ID_A \equiv g^{-s_A} \bmod n$, $k_A = g^{-r_A} \bmod n$,

$P_B^e + ID_B \equiv g^{-s_B} \bmod n$, $k_B = g^{-r_B} \bmod n$,

$k_A' = g^{-r_A} \bmod n$, $k_B = g^{-r_B'} \bmod n$,

$C \equiv (k_B)^{s_A}(P_B^e + ID_B)^{r_A} \equiv (k_A')^{s_B}(P_A^e + ID_A)^{r_B} \bmod n$

이다.

[정의 4 12] $G2^{*}_{KKP}$(n ,e ,g, ID_A, ID_B, P_A, P_B, k_A, k_B, k_A', k_B', C')는 $e \notin Z^{*}_{\phi(n)}$인 경우에 ⊥를 출력하는 것을 제외하고는 $G2_{KKP}$와 동일한 함수이다

[정리 4.4] $G2^{*}_{KKP} \equiv^{ep}_{T} DH$

(증명)

① $G2^{*}_{KKP} \leq^{p}_{2-tt} DH$

$r_B \in_R Z_n$를 선택하고,

$G2_{KKP}(n, e, g, ID_A, ID_B, P_A, P_B, k_A, k_B, k_A', k_B', C')$

$= (DH(n, g, P_A^e + ID_A, k_B^{-1}) \cdot DH(n, g, P_B^e + ID_B, k_A^{-1}))$

$= (g^{-s_A r_B} \cdot g^{-s_B r_A} \bmod n) = g^{-s_A r_B - s_B r_A} \bmod n$

② $DH \leq^{ep}_{T} G2^{*}_{KKP}$

DH()의 입력이 (n, g, A, B)라 할 때, 홀수 $e \in Z_n^{*} \setminus \{1\}$와 s_A, r_B, r_A', r_B', P_A, $P_B \in Z_n$를 랜덤하게 선택한 후, $x \notin \perp$ 일때 까지 다음을 계산한다

$G2^{*}_{KKP}(n, e, g, g^{-s_A} - P_A^e, B^{-1} - P_B^e, P_A, P_B, A, g^{-r_B}, g^{-r_A'}, g^{-r_B'}, g^{-s_A r_B} B^{-r_a'}) = x$

그리고, 다음을 계산할 수 있다.

$DH(n,g,A,B) = x(g^{s_A r_B} \equiv g^{ab} \bmod n$

G2 프로토콜은 세션키를 생성하는데 사용자 A와 B가 선택한 랜덤 수가 포함되므로 과거의 세션키와 전송 정보를 획득하더라도 이를 이용하여 정당한 사용자로 위장하는 것을 불가능하다. 즉, 과거의 세션키와 전송 정보를 이용하여 정당한 사용자로 위장하려는 KKI 공격자의 어려움은 과거 세션에 대한 아무런 정보도 가지고 있지 않은 AI 공격자의 어려움과 동일하다. 자세한 증명 과정은 [정리 4.5]와 같다.

[정의 4.13] $G2_{KKI}$(n ,e ,g, ID_A, ID_B, P_A, P_B, k_A, k_A', k_B', C')는 $n \in N_{>1}$ $g \in Z_n^*$, $e \in Z_{\phi(n)}^*$, ID_A, $ID_B \in Z_n$, P_A, $P_B \in Z_n$, $k_A, k_A', k_B' \in Z_n$, $C' \in Z_n$를 입력으로 하여 $C \equiv (g^{-r_B})^{s_A}(P_B^e + ID_B)^{r_A} \bmod n$이 고 $r_B \in Z_n$에 대해 $k_B = g^{-r_B} \bmod n$을 만족하는 $(C, k_B) \in Z_n \times Z_n$를 출력하는 함수이다. 단, 이러한 (C, k_B)가 존재하면 $P_A^e + ID_A \equiv g^{-s_A} \bmod n$, $k_A = g^{-r_A} \bmod n$, $P_B^e + ID_B \equiv g^{-s_B} \bmod n$, $k_A = g^{-r_A'} \bmod n$, $k_B = g^{-r_B'} \bmod n$, $C' \equiv (k_B')^{s_A}(P_B^e + ID_B)^{r_A'} \equiv (k_A')^{s_B}(P_A^e + ID_A)^{r_B'} \bmod n$이다.

[정의 4.14] $G2^*_{KKI}$(n ,e ,g, ID_A, ID_B, P_A, P_B, k_A, k_A', k_B', C')는 $e \notin Z_{\phi(n)}^*$인 경우 ⊥를 출력하는 것을 제외하고 $G2_{KKI}$와 동일한 함수이다.

[정리 4.5] $G2^*_{KKI} \equiv {}^{ep}_{T}\ DH$

(증명)

① $G2^*_{KKI} \leq {}^{p}_{1-tt}\ DH$

$r_B \in {}_R Z_n$를 선택하고,

$G2^*_{KKI}(n, e, g, ID_A, ID_B, P_A, P_B, k_A, k_A', k_B', C')$

$= ((P_A^e + ID_A)^{r_B} \cdot DH(n, g, P_B^e + ID_B, k_A^{-1}), g^{-r_B})$

$= (g^{-s_A r_B - s_B r_A} \bmod n, g^{-r_B} \bmod n)$

② $DH \leq {}^{p}_{tt}\ G2^*_{KKI}$

DH()의 입력이 (n, g, A, B)라 할 때, 홀수 $e \in Z_n^* \setminus \{1\}$와 s_A, r_A', r_B', P_A, $P_B \in Z_n$를 랜덤하게 선택한 후, $x \notin \perp$일때 까지 다음을 계산한다.

$G2^*_{KKI}(n, e, g, g^{-s_A} - P_A^e, B^{-1} - P_B^e, P_A, P_B, A, g^{-r_A'}, g^{-r_B'}, g^{-s_A r_B'} B^{-r_A'}) = (x, y)$

출력 결과는 어떤 $k_B \in Z_n$에 대해 (x, y) $= (k_B^{s_A} g^{ab}, k_B)$을 만족하므로, 다음을 계산할 수 있다.

$DH(n, g, A, B) = x / A^{r_B} \equiv g^{ab} \bmod n$

[표 4-1]은 Girault의 키 분배 프로토콜의 안전성을 분석한 결과를 정리한 것이다.

표 4-1. Girault 키 분배 프로토콜의 안전성 분석 결과

	G0 프로토콜	G1 프로토콜	G2 프로토콜
AI	공격 불가능	안전하지 않음	$G2_{AI} \equiv {}^{ep}_{T} DH$
KCI	안전하지 않음	안전하지 않음	$G2_{KCI} \equiv {}^{ep}_{T} DH$
FS	안전하지 않음	half forward secrecy $G1_{HFS} \equiv {}^{p}_{m} DH$	half forward secrecy $G2_{HFS} \equiv {}^{ep}_{T} DH$
KKP	안전하지 않음	$G1_{KKP} \equiv {}^{p}_{m} DH$	$G2_{KKP} \equiv {}^{ep}_{T} DH$
KKI	안전하지 않음	안전하지 않음	$G2_{KKI} \equiv {}^{ep}_{T} DH$

V. 결 론

키 분배 프로토콜은 안전한 암호 시스템 구현에 있어 가장 필수적인 요소이며, 이에 대한 연구가 꾸준히 진행되고 있다. 최근 들어 국내·외에서는 암호화 기능을 제공하는 정보 보안 제품들이 많이 개발되고 있으며, 각 제품에서 사용되는 키 분배 프로토콜의 안전성에 대한 정확한 증명없이 산발적으로 제안되고 있는 실정이다. 따라서, 본 논문에서는 수학적 귀착이론을 이용하여 기존에 제안된 자체인증 공개키에 기반한 키 분배 프로토콜들의 안전성과 암호학적 기반 문제와의 상관관계를 증명하였다.

자체인증 공개키에 기반한 방식은 공개키 자체가 인증서의 역할을 하므로 별도의 인증서를 저장하거나 검증할 필요가 없으며, 사용자가 자신의 비밀키를 직

접 선택하므로 ID에 기반한 방식에 비해 높은 신뢰수준을 보장한다는 장점이 있어 여러 분야에 활용할 수 있다. 또한, 본 논문의 연구 결과는 기존에 제안된 키 분배 프로토콜의 안전성 뿐만 아니라 향후에 개발될 키 분배 프로토콜의 안전성을 평가하는데 활용될 수 있을 것으로 기대된다.

참 고 문 헌

[1] E. Bach, "Discrete logarithms and factoring," Technical Report UCB/CSD 84/186, *University of California, Computer Science Division* (EECS), 1984

[2] W. Diffie and M. Hellman, "New directions in cryptography", *IEEE Trans. Inf. Theory*, vol. IT-22, no.6, pp.644-654, 1974

[3] M. Girault, "Self-certified public keys," *Advances in Cryptology-Eurocrypt '91*, LNCS 547, Springer-Verlag, Berlin, pp. 490-497, 1991

[4] Y. Gurevich, "Average Case Completeness," *Journal of Computer and System Sciences*, Vol. 42, pp. 346-398, 1991

[5] S.J. Kim, M. Mambo, T. Okamoto, H. Shizuya, M. Tada, D.H. Won, "On the security of the Okamoto-Tanaka ID-Based Key Exchange scheme against Active attacks," *IEICE Trans. Fundamentals*, vol. E84-A, pp.231-238, Jan. 2001

[6] M. Mambo and H. Shizuya, "A note on the complexity of breaking Okamoto-Tanaka ID-based key exchange scheme," *IEICE Trans. Fundamentals*, vol. E 82-A, pp77-80, Jan. 1999.

[7] K.S. McCurley, "A key distribution system equivalent to Factoring," *Journal of Cryptology, vol. 1*, pp.95-105, 1988

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

[9] E. Okamoto and K. Tanaka, "Key distribution system based on identification information," *IEEE J. Sel. Areas Commun.*, vol.7, pp.481-485, 1989.

[10] P. Ribenboim, "The Book of Prime Number Records," Springer-Verlag, 1988

[11] R.L. Rivest, A. Shamir and L. Adleman, "A method for obtaining digital signatures and public key cryptosystems," *Communication ACM*, vol. 21, no. 2, pp.120-126, 1978.

[12] R.A Rueppel and P.C van Oorschot, "Modern key agreement techniques," *Computer Communications, vol.17* pp.458-465, Jul. 1994.

[13] K. Sakurai and H. Shizuya, "Relationships among the computational powers of breaking discrete log cryptosystems," *Advances in Cryptology-Eurocrypt '95 LNCS 921*, pp.341-355, Springer-Verlag, 1995.

[14] Z. Shmuely, "Composite Diffie-Hellman public-key generating systems are hard to break," Technical report no. 356, *Computer science department, Technion-Israel Institute of Technology*, 1985

[15] H. Woll, "Reduction among number theoretic problems," *Information and Computation*, vol. 72, pp. 167-179, 1987

양 형 규(HyungKyu Yang) 정회원

1983년 2월 : 성균관대학교 전자공학과 학사

1985년 2월 : 성균관대학교 전자공학과 석사

1995년 2월 : 성균관대학교 정보공학과 박사

1995년 ~현재 : 강남대학교 컴퓨터미디어공학부 부교수

1985년 ~ 1991년 : 삼성전자 컴퓨터부문 과장

<관심분야> 컴퓨터통신 보안, 정보보호프로토콜