이산대수 기반 Diffie-Hellman형 표준 키 분배 프로토콜의 안전성 분석에 관한 연구

김 경 진[†]·김 성 덕[†]·심 경 아[††]·원 동 호[†††]

요 약

인터넷과 같은 첨단의 정보 전송 시스템이 발달함에 따라 네트워크 상에서 전송되는 메시지에 대한 기밀성을 제공하기 위해서 암호 시스템의 사용이 증가하고 있으며 그 중요성은 더욱 강조되고 있다. 안전한 암호 시스템을 구현하는 데 있어서 키 분배 프로토콜은 가장 필수적인 요소이며, 지금까지 여러 키 분배 프로토콜들이 표준으로 제안되었으나 이에 대한 엄밀한 안전성 증명은 아직까지 부족한 실정이다. 따라서 본 논문에서는 ANSI X9.42의 Diffie-Hellman형 표준 키 분배 프로토콜들의 특징을 자세히 분석하고 이를 기반으로 여러 능동적 공격자 모델에 대한 프로토콜의 안전성을 증명하고자 한다.

Security Analysis of Diffie-Hellman based Standard Key Agreement Protocols

Kyung-Jin Kim[†]·Sung-Duk Kim[†]·Kyung-Ah Shim[††]·Dong-Ho Won[†††]

ABSTRACT

According to the wide-spread of information transmission system over network, the use of cryptographic system to provide the integrity of transmitted message over network is increasing and the importance of that is emphasized. Because the security of the cryptographic system totally relies on the key, key management is a essential part of cryptographic system. A number of key agreement protocols have been proposed so far, but their rigorous security analysis is still open. In this paper, we analyze the features of Diffie-Hellman based standard key agreement protocols and provide the security analysis of those protocols against several kinds of active attacks.

키워드 : X9.42, 귀착(Reduction), Active Attack, Active Impersonation, Key-Compromised Impersonation, Forward Secrecy, Known-Key Attack

1. 서 론

비밀 통신을 하고자 하는 두 사용자간에 공통의 비밀키를 공유하는 키 분배 프로토콜은 암호 시스템의 가장 중요한 요소이다. 1976년 Diffie-Hellman[1]이 처음으로 공개키 분배 시스템을 제안한 이후로 현재까지 효율적이고 안전한 여러 키 분배 방식들이 제안되고 있으며, 현재 IEEE P1363[2], PKCS #3[3], ANSI X9.42[4], ANSI X9.63[5] 등이 키 분배 표준으로 사용되고 있다.

지금까지 발표된 키 분배 프로토콜의 안전성에 대한 연구 결과는 대부분이 경험적인 안전성 분석 방법으로, 기존에 제안된 여러 공격 방법들을 토대로 키 분배 프로토콜의 안전성을 분석하는 것이 대부분이었다. 경험적 안전성이란 어떤 키 분배 프로토콜이 특정 공격법에 대해서만 안전함을 증명하는 방식으로 묵시적 키인증(implicit key authentication), 명시적 키인증(explicit key authentication), known-key 안전성, forward secrecy, 위장(impersonation)에 대한 안전성, unknown-key share 공격에 대한 안전성 등을 포함한다. 그러나 이러한 경험적 측면의 안전성 분석은 지금까지 제안된 공격 방법들에 대해서만 유효한 것으로 이후에 기존의 공격 방법보다 강력한 새로운 공격 방법이 제안될 경우 그 안전성을 보장받을 수 없다는 문제점이 있다.

따라서 본 논문에서는 각각의 키 분배 프로토콜들에 대하여 그 안전성을 다항식 시간 안에 해결할 수 없는 수학적으로 어려운 문제들로 귀착시킴으로써 키 분배 프로토콜 자체의 안전성을 증명하고자 한다[6, 11]. 이러한 안전성 증명 방

※ 본 연구는 한국정보보호진흥원 위탁과제(2001-S-092)의 지원에 의해 수행하였습니다.
† 준 회 원 : 성균관대학교 정보통신공학부 대학원
†† 정 회 원 : 한국정보보호진흥원(KISA) 암호 기술팀 선임연구원
††† 종신회원 : 성균관대학교 정보통신공학부 교수
논문접수 : 2002년 4월 18일, 심사완료 : 2002년 8월 19일

식은 새로운 공격 방법이 제안될 경우 그 안전성을 보장받을 수 없는 경험적 측면의 안전성 분석과는 달리, 프로토콜의 해당 기반 논리가 깨지지 않는 한 그 안전성을 보장받을 수 있으며, 키 분배 프로토콜과 여러 암호학적 기반 논리와의 상관 관계 연구에 기반이 된다.

현재 표준으로 사용되고 있는 이산대수 기반의 키 분배 프로토콜로는 IEEE P1363에서 제안한 이산대수 기반의 Diffie-Hellman형 키 분배 프로토콜[2], 이산대수 기반의 D-H형 키 분배 프로토콜인 PKCS #3[3], 그리고 ANSI에서 제안한 이산대수 기반의 D-H형 키 분배 프로토콜인 ANSI X9.42[4] 등이 있다. 그런데 PKCS #3는 X9.42의 dhStatic 프로토콜과 그 형태가 동일하고, IEEE P1363의 이산대수 기반 Diffie-Hellman 프로토콜들도 역시, DL/KAS-DH1 프로토콜은 사용되는 키 쌍의 종류(고정/일회용)에 따라서 X9.42의 dhStatic/dhEphem 프로토콜과 그 형태가 동일하며, DL/KAS-DH2 프로토콜은 X9.42의 dhHybrid1 프로토콜과 그 형태가 동일하다.

따라서 본 논문에서는 ANSI X9.42에서 정의하고 있는 프로토콜들 중, MQV(Menezes-Qu-Vanstone)형 키 분배 프로토콜을 제외한 6개의 이산대수 기반의 Diffie-Hellman형 키 분배 프로토콜들을 중심으로 그 특징을 분석하고, 귀착을 이용하여 몇몇 능동적 공격자 모델에 대한 ANSI X9.42의 이산 대수 기반 키 분배 프로토콜들의 안전성을 증명한다[6, 8, 9, 11].

2. 관련 연구

2.1 용 어

본 논문에서는 키 분배 프로토콜의 특징 분석을 위해서 다음과 같은 용어를 사용한다[7].

- **개체 인증(Entity authentication)** : 사용자가 키 분배 프로토콜에 참여하는 상대방이 누구인지를 확인할 수 있을 때 키 분배 프로토콜이 개체 인증을 제공한다고 한다.
- **키 확인(Key confirmation)** : 사용자가 키 분배 프로토콜에 참여한 상대방이 실제로 설정된 비밀 세션키를 알고 있음을 확신할 때 키 분배 프로토콜이 키 확인을 제공한다고 한다.
- **묵시적 키 인증(Implicit key authentication)** : 키의 소유 여부는 알려져 있지 않다고 하더라도 키 분배 프로토콜에 참여한 상대방만이 세션키를 계산할 수 있음을 확신할 수 있을 때 묵시적 키인증을 제공한다고 한다.
- **Key freshness** : 매 세션마다 설정된 세션키가 변경되는 경우에 해당 키 분배 프로토콜은 Key freshness를 제공한다고 한다.

2.2 귀착(Reductions)

본 논문에서는 프로토콜의 안전성을 증명하기 위하여 함수들 사이의 귀착을 사용한다[8, 9].

2.2.1 변환 알고리즘(Transformation algorithm)

어떤 결정 문제들에 대한 함수 F와 G에 대해 F를 푸는 알고리즘은 존재하지 않으나 G를 푸는 알고리즘은 존재하는 경우에, 함수 G를 푸는 알고리즘이 y에 대해 "예"라고 답하면 함수 F에 대한 답도 "예"가 되도록, 함수 G의 모든 사례에 대해 함수 G의 사례 y를 만들어내는 알고리즘을 변환 알고리즘이라고 한다.

이 알고리즘은 함수 F의 모든 사례를 함수 G의 사례로 매핑하는 함수로, 함수 G를 푸는 알고리즘을 가지고 함수 F를 푸는 알고리즘을 만들어 내며 $y = h(x)$라고 표시한다.

2.2.2 귀착의 종류와 정의

결정 문제들에 대한 함수 F와 G에 대해 $F(x) = G(h(x))$를 만족하는 다항식 시간에 계산 가능한 변환 함수 h가 존재하면, F는 G에 다항식 시간 many-one 귀착 가능하다고 하며 $F \le_m^p G$라고 표시한다[9].

그리고 다항식 시간 튜링 머신에 G에 대한 adaptive한 질의(query)를 주어 함수 F를 계산할 수 있다면, F는 G에 다항식 시간 튜링 귀착 가능하다고 하며 $F \le_T^{(e)p} G$라고 표시한다[9]. 이때, 모든 $x \in \{0,1\}^*$와 무한 비트 수열 r에 대해 $(t_M(x, r))^e \le |x|$를 만족하는 $e > 0$가 존재하면 M은 예상되는 다항식 시간이라고 한다[9, 12].

튜링 귀착 가능성은 결정 문제의 귀착을 비결정 문제로 일반화시킨 것이다. 따라서 이 정의에 의하면 함수 G를 푸는 다항식 시간 알고리즘이 존재해야할 필요는 없으며, 단지 그 알고리즘이 존재하는 경우 함수 F도 다항식 시간에 풀 수 있다는 것을 정의하고 있다. 여기서 튜링 머신이 G에 대해 "adaptive한 질의"라는 것은 튜링 머신에 질의를 던져 질문을 할 때, 이전 질의의 결과가 다음 질의의 작성 시에 고려될 수 있다는 것으로 튜링 머신에 순차적인(serial) 질의를 통하여 보다 강력한 응답을 얻을 수 있는 것을 의미한다.

또한, 다항식 시간 튜링 머신에 G에 대한 non-adaptive한 질의를 주어 함수 F를 계산할 수 있다면, F는 G에 다항식 시간 truth-table 귀착 가능하다고 하며 $F \le_{tt}^p G$라고 표시한다. 특히, 최대한 k번의 질의를 이용하여 귀착 가능한 경우에 $F \le_{k-tt}^p G$라고 표시한다[9].

이 귀착 방식은 다항식 시간 튜링 귀착의 특수한 방식으로 단지 다항식 능력을 갖는 오라클에 non-adaptive한 질의를 통해 출력값을 계산한다는 차이점이 있다. 여기에서 "non-adaptive한 질의"의 의미는 질의를 한번에 테이블로 오라클에게 전송하여 응답을 얻는 것을 의미하는 것으로 다음 질의의 작성시 이전 질의의 결과가 고려되지 않으며, 이

것을 병렬적인(parallel) 질의라고 한다. 따라서 튜링 귀착보다 약한 정도의 귀착 방식이라 할 수 있다.

이러한 귀착 개념들 사이의 상관관계는 다음과 같다[9].

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

2.3 암호학적 기반 문제

2.3.1 암호학적 기반 문제의 정의

키 분배 프로토콜의 안전성을 분석하기 위해 필요한 암호학적 기반 문제에 대한 정의는 다음과 같다.

먼저 이산대수 문제란 유한체 GF(p) 상에서 위수(order)가 p-1인 순환 그룹(cyclic group) G, 그룹 G의 원시원소 g와 G의 원소 β가 주어졌을 때, $g^x \equiv \beta$를 만족하는 x ($0 \leq x < p$)를 계산하는 문제로 이때, x를 원시원소 g에 대한 β의 이산대수라 한다.

Diffie-Hellman 문제는 키 분배 프로토콜의 기반 문제로 가장 많이 사용되고 있는 암호 기반 논리이며, 유한체 GF(p) 상에서 위수(order)가 p-1인 순환 그룹(cyclic group) G, 그룹 G의 원시원소 g, G의 임의의 원소 a, b에 대해 g^a, g^b가 주어졌을 때, g^{ab}를 구하는 문제이다.

이산대수 문제를 푸는 오라클 DL( )과 유한체 상에서의 Diffie-Hellman 문제를 푸는 오라클 DH( )에 대한 정의는 각각 [정의 1], [정의 2]와 같다.

[정의 1] DL(p,g,y)은 큰 소수 p, Z_p의 원시원소 g, $y \in Z_p$를 입력으로 하여 $0 \leq x < p$이고 $y \equiv g^x \bmod p$를 만족하는 x를 구하는 함수이다.

[정의 2] DH(p,g,A,B)는 큰 소수 p, Z_p의 원시원소 g, $A \in Z_p$, $B \in Z_p$를 입력으로 하여 $C \equiv g^{ab} \bmod p$를 만족하는 C를 출력하는 함수이다. 단, $A \equiv g^a \bmod p$이고 $B \equiv g^b \bmod p$이다.

2.3.2 암호학적 기반 문제들 사이의 상관관계

Diffie-Hellman 문제는 그 안전성이 이산대수 문제에 기반하므로 $DH \leq_{1-tt}^p DL$이다. 즉, Diffie-Hellman 문제는 기반 논리인 이산대수 문제가 깨지지 않는 한 그 안전성을 보장받을 수 있다. 자세한 증명은 [정리 1]과 같다.

[정리 1] $DH \leq_{1-tt}^p DL$

(증명)

$$DH(p, g, A, B) \equiv A^{DL(p, g, B)}$$ □

3. ANSI X9.42 이산대수 기반 키 분배 프로토콜

3.1 기호 정의

각각의 키 분배 프로토콜의 분석에 앞서 ANSI X9.42에서 사용하는 기호와 그 뜻에 대하여 정의하고, 이후 프로토콜에서는 아래의 기호에 대한 정의는 생략한다[4].

[ANSI X9.42 기호 정의]

- p : 유한체 GF(p)를 정의하는 큰 소수
- q : $q \mid p-1$인 소수
- j : p-1의 cofactor, $p-1 = jq$
- g : GF(p)상에서 원시원소
- U : 키 분배 프로토콜의 시행자(initiator)
- V : 키 분배 프로토콜의 응답자(recipient)
- $x_U, x_V / r_U, r_V$: 사용자 U/V의 고정/일회용 비밀키
- $y_U, y_V / t_U, t_V$: 사용자 U/V의 고정/일회용 공개키

3.2 키 분배 과정

ANSI X9.42에는 총 8개의 프로토콜을 정의하고 있는데, 6개의 Diffie-Hellman형 키 분배 프로토콜과 2개의 MQV형 키 분배 프로토콜이 있다[4].

dhStatic 프로토콜은 사용자 U와 V 모두 고정 도메인 파라미터 (p_s, q_s, g_s)만을 사용하며, 고정된 키 쌍을 사용하여 세션키를 설정하는 방식으로 ANSI X9.42에서 정의하는 프로토콜 중 가장 기본이 되는 방식이다. dhEphem 프로토콜은 사용자 U와 V 모두 일회용 도메인 파라미터 (p_e, q_e, g_e)를 사용하며, 일회용 키 쌍을 이용하는 것을 제외하고는 세션키의 설정 방식은 dhStatic 프로토콜과 동일하다.

dhOneFlow 프로토콜은 사용자 U와 V 모두 공통의 고정된 도메인 파라미터 (p_s, q_s, g_s)를 사용하지만, 사용자 U는 일회용 키 쌍을, 사용자 V는 고정된 키 쌍을 이용하여 세션키를 설정하는 것이 다르다. 즉, 사용자 V는 인증된 공개키를 변경하지 않고 여러 번 키 분배 프로토콜을 수행하며, 사용자 U는 프로토콜을 수행할 때마다 서로 다른 일회용 키 쌍을사용하여 키 분배 프로토콜을 수행하게 된다. dhHybrid 1 프로토콜은 앞서 설명한 dhStatic와 dhEphem 프로토콜을 접목시킨 프로토콜로써, 사용자 U와 V가 고정된 도메인 파라미터 (p_s, q_s, g_s)에서 각각 고정된 키와 일회용 키 쌍을 생성하여 세션키를 설정하는 방식이며, dhHybrid 2 프로토콜은 dhHybrid1 프로토콜의 변수 설정을 약간 변형한 프로토콜로써, dhHybrid1 프로토콜과는 달리 사용자 U와 V가 고정된 도메인 파라미터 (p_s, q_s, g_s)와 일회용 도메인 파라미터 (p_e, q_e, g_e)를 모두 사용하여 2개의 공개키를 만들어 프로토콜을 수행한다.

마지막으로 dhHybridOneFlow 프로토콜은 dhHybrid1 프로토콜을 변형한 방식으로, 사용자 U는 고정된 도메인 파라미터 (p_s, q_s, g_s)를 이용하여 고정된 키 쌍과 일회용 키 쌍을 모두 생성하고 사용자 V는 고정된 키 쌍만을 생성한다.

지금까지 설명한 각각의 키 분배 프로토콜들의 세션키 설정에 필요한 고정 데이터, 일회용 데이터, 사용자들의 세션키 설정 과정 및 설정된 세션키는 <표 1>과 같다.

〈표 1〉 ANSI X9.42의 이산대수 기반 키 분배 프로토콜

프로토콜	사용자	전송정보		세션키 설정과정	세션키
		고정(y)	일회용(t)		
dhStatic	U	$g_s^{x_U}$	-	$y_V^{x_U}$	$g_s^{x_U x_V} \bmod p_s$
	V	$g_s^{x_V}$	-	$y_U^{x_V}$	
dhEphem	U	-	$g_e^{r_U}$	$t_V^{r_U}$	$g_e^{r_U r_V} \bmod p_e$
	V	-	$g_e^{r_V}$	$t_U^{r_V}$	
dhOneFlow	U	-	$g_s^{r_U}$	$y_V^{r_U}$	$g_s^{r_U x_V} \bmod p_s$
	V	-	-	$t_U^{x_V}$	
dhHybrid 1	U	$g_s^{x_U}$	$g_s^{r_U}$	$y_V^{x_U} \| t_V^{r_U}$	$g_s^{x_U x_V} \bmod p_s \| g_s^{r_U r_V} \bmod p_s$
	V	$g_s^{x_V}$	$g_s^{r_V}$	$y_U^{x_V} \| t_U^{r_V}$	
dhHybrid 2	U	$g_s^{x_U}$	$g_e^{r_U}$	$y_V^{x_U} \| t_V^{r_U}$	$g_s^{x_U x_V} \bmod p_s \| g_e^{r_U r_V} \bmod p_e$
	V	$g_s^{x_V}$	$g_e^{r_V}$	$y_U^{x_V} \| t_U^{r_V}$	
dhHybridOneFlow	U	$g_s^{x_U}$	$g_s^{r_U}$	$y_V^{x_U} \| y_V^{r_U}$	$g_s^{x_U x_V} \bmod p_s \| g_s^{r_U x_V} \bmod p_s$
	V	-	-	$y_U^{x_V} \| t_U^{x_V}$	

3.3 키 분배 프로토콜의 특징 분석

본 논문에서는 위에서 살펴본 ANSI X9.42의 각 프로토콜들에 대하여, 사용자 U와 V가 공통의 세션키를 설정하기 위하여 필요한 통신 회수, 개체 인증, 키 확인, 묵시적 키인증과 key freshness를 중심으로 프로토콜의 특징을 분석하였다[7]. <표 2>는 ANSI X9.42 키 분배 프로토콜의 특징을 분석한 결과를 정리한 것이다.

〈표 2〉 ANSI X9.42의 이산대수 기반 키 분배 프로토콜의 특징

프로토콜	통신회수	개체인증	키 확인	묵시적 키인증	Key freshness
dhStatic	2	-	-	양방향	-
dhEphem	2	-	-	-	양방향
dhOneFlow	1	-	-	일방향	일방향
dhHybrid 1	2	-	-	양방향	양방향
dhHybrid 2	2	-	-	양방향	양방향
dhHybridOneFlow	1	-	-	양방향	일방향

4. 안전성 분석

4.1 공격자 모델

암호 프로토콜에 대한 공격자는 크게 수동적 공격자(passive attacker)와 능동적 공격자(active attacker)로 나눌 수 있다. 수동적 공격자란 프로토콜의 참가자와 실제로 통신에 참여하지 않고 두 참가자 사이의 통신 내용을 도청(eavesdropping)함으로써 공격을 수행하는 공격자를 말하며, 능동적 공격자란 단순히 참가자들의 통신 내용을 도청하는 것뿐만 아니라 전송되는 메시지를 위·변조하거나 새로운 메시지를 삽입하는 것과 같이 실제 통신에 참여하여 공격을 수행하는 보다 강력한 공격자를 말한다.

본 논문에서 대상으로 하는 공격자 모델은 다음과 같다[6, 11].

- **Active Impersonation(AI) 공격** : 공격자가 프로토콜에 참여하여 자신을 임의의 다른 사용자(B)로 위장하여 정당한 사용자 A와 키 분배를 수행하는 경우에 ***active impersonation***이 가능하다고 한다.
- **Forward Secrecy(FS)** : 사용자 A와(또는) B의 비밀키가 노출되더라도, 공격자가 과거에 두 사용자 사이에 설정된 세션키를 계산할 수 없는 경우에 ***forward secrecy***를 만족한다고 한다.
  - ***Half Forward Secrecy*** : 한 사용자의 비밀키가 노출된 경우에만 세션키가 안전
  - ***Full Forward Secrecy*** : 두 사용자의 비밀키가 모두 노출된 경우에도 세션키가 안전
- **Key-Compromise Impersonation(KCI) 공격** : 사용자 A의 비밀키가 노출되었을 경우, 공격자 C가 누구에게나 사용자 A로 위장할 수 있고, 사용자 A에게 임의의 사용자 B로 위장할 수 있을 때 ***key-compromise impersonation***이 가능하다고 한다. 또한, 공격자 C가 누구에게나 사용자 A로 위장하는 것은 가능하지만, 사용자 A에게는 임의의 다른 사용자로 위장할 수는 없는 경우에, 키 분배 프로토콜은 ***key-compromise impersonation resilience*** 특성을 갖는다고 한다.
- **Known Key Security(KKS)** : 두 사용자 A와 B사이에 설정된 과거의 세션키가 노출되더라도 현재의 세션키의 안전성에는 아무런 영향을 미치지 않는 경우에 ***Known Key Security***를 만족한다고 하며 이것에 대한 공격은 다음과 같이 두 가지로 나눌 수 있다.
  - ***Known Key Passive(KKP) 공격*** : 과거의 세션키와 전송 정보, 그리고 현재 세션의 전송 정보를 이용하여 현재의 세션키를 획득하려는 공격 방법

- ***Known Key Impersonation(KKI) 공격*** : 세션에 직접 참여하여 과거의 세션키와 전송 정보, 그리고 현재 세션의 전송 정보를 이용하여 사용자 A에게 사용자 B로 위장하여 세션키를 설정하려는 공격 방법

4.2 안전성 분석 결과

4.2.1 수동적 공격자에 대한 안전성

X9.62에서 제안하고 있는 키 분배 프로토콜들의 안전성은 기본적으로 DH 문제에 기반하므로 수동적 공격자가 사용자들의 공개 정보와 전송 정보만을 이용하여 세션키를 구하는 어려움은 Z_p상에서 DH 문제를 푸는 어려움과 동일하다.

4.2.2 능동적 공격자에 대한 안전성

본 논문에서는 앞에서 분석한 키 분배 프로토콜들에 대하여, AI, FS, KCI, KKS 공격자 환경에서의 안전성을 증명하였다. 각각의 프로토콜의 안전성 증명에 사용하게 될 Diffie-Hellman 문제에 대한 정의와 공격자 함수에 대한 정의는 <표 3>과 같다.

각각의 함수는 주어진 입력값을 가지고 C 또는 (C, t_V)를 출력한다. 단, 그러한 출력값 C 또는 (C, t_V)가 존재하면, 각 함수의 입력값들은 주어진 조건(R)을 만족한다[6, 11]. 예를

〈표 3〉 공격자 함수 정의

함 수(F)	입 력(I)	출 력 (C, t_V)	조 건(R)
$dhEphem_{FS}$	$p_e, g_e, t_U, t_V, x_U, x_V$	$g_e^{r_U r_V} \bmod p_e$	$t_U \equiv g_e^{r_U} \bmod p_e,\ t_V \equiv g_e^{r_V} \bmod p_e$
$dhEphem_{KKP}$	$p_e, g_e, t_U, t_V, t_U', t_V', C'$	$g_e^{r_U r_V} \bmod p_e$	$t_U \equiv g_e^{r_U} \bmod p_e,\ t_V \equiv g_e^{r_V} \bmod p_e,$ $t_U' \equiv g_e^{r_U} \bmod p_e,\ t_V' \equiv g_e^{r_V} \bmod p_e,$ $C' \equiv g_e^{r_U' r_V'} \bmod p_e$
$dhOneFlow_{HFS}$	p_s, g_s, t_U, y_V, x_U	$g_s^{r_U x_V} \bmod p_s$	$t_U \equiv g_s^{r_U} \bmod p_s,\ \ y_V \equiv g_s^{x_V} \bmod p_s$
$dhOneFlow_{KKP}$	$p_s, g_s, t_U, y_V, t_U, C'$	$g_s^{r_U x_V} \bmod p_s$	$t_U \equiv g_s^{r_U} \bmod p_s,\ y_V \equiv g_s^{x_V} \bmod p_s,$ $t_U' \equiv g_s^{r_U'} \bmod p_s,\ C' \equiv g_s^{r_U' x_V} \bmod p_s$
$dhHybrid1_{AI}$	p_s, g_s, y_U, y_V, t_U	$g_s^{x_U x_V} \| g_s^{r_U r_V} \bmod p_s,\ g_s^{r_V} \bmod p_s$	$y_U \equiv g_s^{x_U} \bmod p_s,\ y_V \equiv g_s^{x_V} \bmod p_s,$ $t_U \equiv g_s^{r_U} \bmod p_s$
$dhHybrid1_{FS}$	$p_s, g_s, y_U, y_V, t_U, t_V, x_U, x_V$	$g_s^{x_U x_V} \| g_s^{r_U r_V} \bmod p_s$	$y_U \equiv g_s^{x_U} \bmod p_s,\ y_V \equiv g_s^{x_V} \bmod p_s,$ $t_U \equiv g_s^{r_U} \bmod p_s,\ t_V \equiv g_s^{r_V} \bmod p_s$
$dhHybrid1_{KKP}$	$p_s, g_s, y_U, y_V, t_U, t_V, t_U', t_V', C'$	$g_s^{x_U x_V} \| g_s^{r_U r_V} \bmod p_s$	$y_U \equiv g_s^{x_U} \bmod p_s,\ y_V \equiv g_s^{x_V} \bmod p_s,$ $t_U \equiv g_s^{r_U} \bmod p_s,\ t_V \equiv g_s^{r_V} \bmod p_s,$ $t_U' \equiv g_s^{r_U'} \bmod p_s,\ t_V' \equiv g_s^{r_V'} \bmod p_s,$ $C' \equiv g_s^{x_U x_V} \| g_s^{r_U' r_V'} \bmod p_s$
$dhHybrid1_{KKI}$	$p_s, g_s, y_U, y_V, t_U, t_U', t_V', C'$	$g_s^{x_U x_V} \| g_s^{r_U r_V} \bmod p_s,\ g_s^{r_V} \bmod p_s$	$y_U \equiv g_s^{x_U} \bmod p_s,\ y_V \equiv g_s^{x_V} \bmod p_s,$ $t_U \equiv g_s^{r_U} \bmod p_s,\ t_U' \equiv g_s^{r_U'} \bmod p_s,$ $t_V' \equiv g_s^{r_V'} \bmod p_s,\ C' \equiv g_s^{x_U x_V} \| g_s^{r_U' r_V'} \bmod p_s$
$dhHybrid2_{AI}$	$p_s, g_s, p_e, g_e, y_U, y_V, t_U$	$g_s^{x_U x_V} \bmod p_s \| g_s^{r_U r_V} \bmod p_e,$ $g_e^{r_V} \bmod p_e$	$y_U \equiv g_s^{x_U} \bmod p_s,\ y_V \equiv g_s^{x_V} \bmod p_s,$ $t_U \equiv g_e^{r_U} \bmod p_e$
$dhHybrid2_{FS}$	$p_s, g_s, p_e, g_e, y_U, y_V, t_U, t_V, x_U, x_V$	$g_s^{x_U x_V} \bmod p_s \| g_s^{r_U r_V} \bmod p_e$	$y_U \equiv g_s^{x_U} \bmod p_s,\ y_V \equiv g_s^{x_V} \bmod p_s,$ $t_U \equiv g_e^{r_U} \bmod p_e,\ t_V \equiv g_e^{r_V} \bmod p_e$
$dhHybrid2_{KKP}$	$p_s, g_s, p_e, g_e, y_U, y_V, t_U, t_V, t_U', t_V', C'$	$g_s^{x_U x_V} \bmod p_s \| g_s^{r_U r_V} \bmod p_e$	$y_U \equiv g_s^{x_U} \bmod p_s,\ y_V \equiv g_s^{x_V} \bmod p_s,$ $t_U \equiv g_e^{r_U} \bmod p_e,\ t_V \equiv g_e^{r_V} \bmod p_e,$ $t_U' \equiv g_e^{r_U'} \bmod p_e,\ t_V' \equiv g_e^{r_V'} \bmod p_e$
$dhHybrid2_{KKI}$	$p_s, g_s, p_e, g_e, y_U, y_V, t_U, t_U', t_V', C'$	$g_s^{x_U x_V} \bmod p_s \| g_s^{r_U r_V} \bmod p_e,$ $g_e^{r_V} \bmod p_e$	$y_U \equiv g_s^{x_U} \bmod p_s,\ y_V \equiv g_s^{x_V} \bmod p_s,$ $t_U \equiv g_e^{r_U} \bmod p_e,\ t_U' \equiv g_e^{r_U'} \bmod p_e,$ $t_V' \equiv g_e^{r_V'} \bmod p_e$
$dhHybridOneFlow_{AI}$	p_s, g_s, y_U, y_V, t_U	$g_s^{x_U x_V} \| g_s^{r_U x_V} \bmod p_s$	$y_U \equiv g_s^{x_U} \bmod p_s,\ y_V \equiv g_s^{x_V} \bmod p_s,$ $t_U \equiv g_s^{r_U} \bmod p_s$
$dhHybridOneFlow_{HFS}$	$p_s, g_s, y_U, y_V, t_U, x_U$	$g_s^{x_U x_V} \| g_s^{r_U x_V} \bmod p_s$	$y_U \equiv g_s^{x_U} \bmod p_s,\ y_V \equiv g_s^{x_V} \bmod p_s,$ $t_U \equiv g_s^{r_U} \bmod p_s$
$dhHybridOneFlow_{KKP}$	$p_s, g_s, y_U, y_V, t_U, t_U', C'$	$g_s^{x_U x_V} \| g_s^{r_U x_V} \bmod p_s$	$y_U \equiv g_s^{x_U} \bmod p_s,\ y_V \equiv g_s^{x_V} \bmod p_s,$ $t_U \equiv g_s^{r_U} \bmod p_s,\ t_U' \equiv g_s^{r_U'} \bmod p_s,$ $C' \equiv g_s^{x_U x_V} \| g_s^{r_U' x_V} \bmod p_s$

들어, $dhEphem_{FS}$ 함수의 경우에는 $(p_e, g_e, t_U, t_V, x_U, x_V)$를 입력으로 하여 $C \equiv g_e^{r_U r_V} \bmod p_e$를 만족하는 C를 출력한다. 단, $t_U \equiv g_e^{r_U} \bmod p_e$, $t_V \equiv g_e^{r_V} \bmod p_e$이다.

• dhStatic 프로토콜

dhStatic 프로토콜은 사용자의 고정된 비밀키만으로 세션키를 생성하기 때문에 AI 공격이 불가능하며 사용자의 비밀키가 노출된 경우에 공격자는 사용자에게 임의의 다른 사용자로 위장하는 것이 가능하다. 따라서 dhstatic 프로토콜은 KCI 공격이 가능하므로 key compromise impersonation resilience 특성을 갖지 못한다. 또한 사용자의 고정된 비밀키가 노출되면, 공격자는 그것을 사용하여 세션키를 계산할 수 있기 때문에 forward secrecy를 제공하지 않으며 이전 세션의 전송 정보와 세션키가 노출되면 KKP 공격자는 현재의 세션키를 쉽게 구할 수 있다. 또한 이전 세션의 전송 정보와 세션키를 획득한 KKI 공격자가 이를 이용하여 다른 사용자로 위장하려는 경우에도 획득한 정보를 이용하여 쉽게 다른 사용자로 위장하는 것이 가능하다. 따라서 dhstatic 프로토콜은 KKP와 KKI 공격에 대하여 안전하지 못하다.

• dhEphem 프로토콜

dhEphem 프로토콜은 두 사용자 사이에 세션키를 설정하기 위해서 매 세션마다 다르게 선택한 일회용 비밀키-공개키 쌍만을 사용하기 때문에 상대방에 대한 어떠한 인증도 제공하지 않는다. 그러므로 공격자가 임의의 다른 사용자로 위장하는 것이 가능하며 AI 공격에 대하여 안전하지 않다.

사용자의 비밀키가 노출되더라도 세션키가 매 세션마다 다르게 생성되기 때문에 공격자에게는 아무런 도움이 되지 않는다. 즉, 두 사용자의 비밀키가 모두 노출되더라도 공격자가 현재의 세션키를 구하는 어려움은 수동적 공격자와 동일하므로 full forward secrecy를 제공한다. 그러나 사용자에 대한 인증을 제공하지 못하므로 공격자가 다른 사용자로 위장하는 것이 가능하기 때문에 key compromise impersonation resilience 특성은 갖지 못한다.

또한 공격자가 과거의 세션키와 전송 정보를 획득하더라도 매 세션마다 그 값들이 변하기 때문에 KKP 공격자가 현재의 세션키를 계산하는 데에는 아무런 도움이 되지 않는다. 즉, 과거의 세션키와 전송 정보가 있는 경우와 그렇지 않은 경우에 공격자의 능력은 동일하므로 KKP 공격에 대하여 안전하다. 그러나 이전 세션의 전송 정보와 세션키를 획득한 KKI 공격자가 이를 이용하여 다른 사용자로 위장하려는 경우에는 dhEphem 프로토콜은 사용자에 대한 인증을 제공하지 못하기 때문에 공격자는 획득한 정보를 이용하여 쉽게 다른 사용자로 위장하는 것이 가능하다. 따라서 dhEphem 프로토콜은 KKI 공격자에 대하여는 안전하지 못하다.

FS와 KKP에 대한 dhEphem의 자세한 안전성 증명은 [정리 2], [정리 3]과 같다.

[정리 2] $dhEphem_{FS} \equiv m^p$ DH

(증명)

① $dhEphem_{FS} \leq_m^p$ DH

$dhEphem_{FS}(p_e, g_e, t_U, t_V, x_U, x_V)$

$\equiv DH(p_e, g_e, t_U, t_V)$

② $DH \leq_m^p dhEphem_{FS}$

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

$\equiv dhEphem_{FS}(p, g, A, B, x_U, x_V)$ □

[정리 3] $dhEphem_{KKP} \equiv_m^p$ DH

(증명)

① $dhEphem_{KKP} \leq_m^p$ DH

$dhEphem_{KKP}(p_e, g_e, t_U, t_V, t_U', t_V', C')$

$\equiv DH(p_e, g_e, t_U, t_V)$

② $DH \leq_m^p dhEphem_{KKP}$

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

$\equiv dhEphem_{KKP}(p, g, A, B, A', B', C')$ □

• dhOneFlow 프로토콜

dhOneFlow 프로토콜은 일방향 프로토콜이기 때문에 사용자 U에 대한 인증을 제공하지 못한다. 따라서 공격자가 사용자 V에게 사용자 U로 위장하는 것이 가능하기 때문에 AI 공격에 대해 안전하지 않으며, 사용자 V의 비밀키가 노출되는 경우에 공격자는 사용자 V에게 임의의 다른 사용자로 위장할 수 있으므로 key-compromise impersonation resilience 특성을 갖지 못한다.

dhOneFlow은 일방향 프로토콜로 사용자 V의 비밀키가 노출된 경우에 forward secrecy를 만족하지 못하지만, 사용자 U의 비밀키는 세션키 생성에 포함되지 않으므로 사용자 U의 비밀키가 노출된 경우에는 세션키가 안전하므로 half forward secrecy를 제공한다.

또한 dhOneFolw 프로토콜은 세션키를 생성하는데 사용자 U가 선택한 랜덤 수가 포함되므로 과거의 세션키와 전송 정보를 획득하더라도 사용자가 서로 다른 난수를 이용하는 경우에 현재의 세션키를 계산하는데 도움이 되지 않기 때문에 과거의 세션키와 전송 정보가 있는 경우와 그렇지 않은 경우에 공격자의 능력은 동일하다. 그러나 이전 세션의 전송 정보와 세션키를 획득한 KKI 공격자가 이를 이용하여 다른 사용자로 위장하려는 경우에 사용자 V에게 사용자 U로 위장할 수 있기 때문에 KKI 공격에 대해서는 안전하지 못하다.

HFS와 KKP에 대한 dhOneFlow의 자세한 안전성 증명은 [정리 4], [정리 5]와 같다.

[정리 4] $dhOneFlow_{HFS} \equiv_m^p DH$

(증명)

① $dhOneFlow_{HFS} \leq_m^p DH$

$dhOneFlow_{HFS}(p_s, g_s, t_U, y_V, x_U)$

$\equiv DH(p_s, g_s, t_U, y_V)$

② $DH \leq_m^p dhOneFlow_{HFS}$

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

$\equiv dhOneFlow_{HFS}(p, g, A, B, x_U)$ □

[정리 5] $dhOneFlow_{KKP} \equiv_m^p DH$

(증명)

① $dhOneFlow_{KKP} \leq_m^p DH$

$dhOneFlow_{KKP}(p_s, g_s, t_U, y_V, t_U', C')$

$\equiv DH(p_s, g_s, t_U, y_V)$

② $DH \leq_m^p dhOneFlow_{KKP}$

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

$\equiv dhOneFlow_{KKP}(p, g, A, B, t_U', C')$ □

• dhHybrid1 프로토콜

dhHybrid1 프로토콜은 공개키 생성 단계에 비밀키가 g^{x_U} 형태로 포함되므로 능동적 공격자라 할 지라도 AI 공격에 성공하는 것은 Z_p상에서 Diffie-Hellman 문제를 푸는 어려움과 동치이다. 따라서 dhHybrid1 프로토콜은 AI 공격에 대하여 안전하다. 그러나 사용자 U의 비밀키가 노출된 경우에 공격자가 사용자 U로 위장할 수 있을 뿐만 아니라 사용자 U에게 임의의 다른 사용자로 위장할 수 있으므로 key-compromise impersonation resilience 특성을 갖지 않는다.

dhHybrid1 프로토콜은 사용자 U와 V의 비밀키가 세션키 생성에 포함되기는 하지만, 두 사용자가 선택한 랜덤수가 연접되어 포함되어 있기 때문에 사용자 U와 V의 비밀키가 노출된 경우에도 세션키는 안전하므로 full forward secrecy를 제공하며, 세션키를 생성하는데 사용자 U와 V가 매 세션마다 다르게 선택한 랜덤 수가 포함되므로 공격자가 과거의 세션키와 전송 정보를 획득하더라도 현재의 세션키를 계산하는데 도움이 되지 않는다. 즉, 과거의 세션키와 전송 정보가 있는 경우와 그렇지 않은 경우에 공격자의 능력은 동일하다. 따라서 dhHybrid1 프로토콜은 KKP 공격에 대하여 안전하다.

또한, 공격자가 사용자 V로 위장하여 세션키를 설정하기 위해, 사용자 U로부터 t_U를 받은 후에, 랜덤한 수 $r_V \in_R Z_{p_s}$를 선택하여 t_V를 계산하고 이를 U에게 전송하면 세션키는 $C \equiv g_s^{x_U x_V} \| g_s^{r_U r_V} \bmod p_s$가 된다. 이때, 공격자는 사용자 V의 비밀키 x_V를 알지 못하므로 Diffie-Hellman 문제를 해결하지 못하면 세션키를 계산할 수 없게 된다. 즉, 과거의 세션키와 전송 정보가 있는 경우와 그렇지 않은 경우에 공격자의 능력은 동일하므로 dhHybrid1 프로토콜은 KKI 공격에 대해 안전하다.

AI, FS, KKP, KKI에 대한 dhHybrid1의 자세한 안전성 증명은 [정리 6], [정리 7], [정리 8], [정리 9]와 같다.

[정리 6] $dhHybrid1_{AI} \equiv_{tt}^p DH$

(증명)

① $dhHybrid1_{AI} \leq_{2-tt}^p DH$

$r_V \in_R Z_{p_s}$에 대해,

$dhHybrid1_{AI}(p_s, g_s, y_U, y_V, t_U)$

$\equiv (DH(p_s, g_s, y_U, y_V) \| DH(p_s, g_s, t_U, g^{r_V}), g^{r_V})$

② $DH \leq_m^p dhHybrid1_{AI}$

$r_V \in_R Z_{p_s}$에 대해,

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

$\equiv dhHybrid1_{AI}(p, B, A, 1, A^{r_V^{-1}})$ □

[정리 7] $dhHybrid1_{FS} \equiv_{tt}^p DH$

(증명)

① $dhHybrid1_{FS} \leq_{2-tt}^p DH$

$dhHybrid1_{FS}(p_s, g_s, y_U, y_V, t_U, t_V, x_U, x_V)$

$\equiv DH(p_s, g_s, y_U, y_V) \| DH(p_s, g_s, t_U, t_V)$

② $DH \leq_{1-tt}^p dhHybrid1_{FS}$

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

$\equiv \lceil dhHybrid1_{FS}(p, g, A, B, A, B, x_U, x_V) \rceil^{|C|}$ □

[정리 8] $dhHybrid1_{KKP} \equiv_{tt}^p DH$

(증명)

① $dhHybrid1_{KKP} \leq_{2-tt}^p DH$

$dhHybrid1_{KKP}(p_s, g_s, y_U, y_V, t_U, t_V, t_U', t_V', C')$

$\equiv DH(p_s, g_s, y_U, y_V) \| DH(p_s, g_s, t_U, t_V)$

② $DH \leq_{1-tt}^p dhHybrid1_{KKP}$

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

$\equiv \lceil dhHybrid1_{KKP}(p, g, A, B, A, B, t_U', t_V', C') \rceil^{|C|}$ □

[정리 9] $dhHybrid1_{KKI} \equiv_{tt}^p DH$

(증명)

① $dhHybrid1_{KKI} \leq_{2-tt}^p DH$

$r_V \in_R Z_{p_s}$에 대해,

$dhHybrid1_{KKI}(p_s, g_s, y_U, y_V, t_U, t_U', t_V', C')$

$\equiv (DH(p_s, g_s, y_U, y_V) \| DH(p_s, g_s, t_U, g_s^{r_V}), g_s^{r_V})$

② $DH \leq_{1-tt}^p dhHybrid1_{KKI}$

$r_V \in_R Z_{p_s}$에 대해,

$DH(p, g, A, B)$ □

• dhHybrid2 프로토콜

dhHybrid2 프로토콜의 공개키 생성 단계에 비밀키가 g^{x_U} 형태로 포함되므로 능동적 공격자라 할 지라도 AI 공격을 성공하는 것은 Z_p상에서 Diffie-Hellman 문제를 푸는 어려움과 동치이다. 따라서 dhHybrid2 프로토콜은 AI 공격에 대하여 안전하다. 그러나 사용자의 비밀키가 노출된 경우에 공격자는 사용자에게 임의의 다른 사용자로 위장할 수 있으므로 key-compromise impersonation resilience 특성을 갖지 않는다.

dhHybrid2 프로토콜은 사용자 U와 V의 비밀키가 세션키 생성에 포함되기는 하지만, 두 사용자가 선택한 랜덤수가 연접되어 포함되어 있기 때문에 사용자 U와 V의 비밀키가 모두 노출된 경우에도 세션키는 안전하므로 full forward secrecy를 제공하며, 세션키를 생성하는데 사용자 U와 V가 매 세션마다 다르게 선택한 랜덤 수가 포함되므로 공격자가 과거의 세션키와 전송 정보를 획득하더라도 현재의 세션키를 계산하는데 도움이 되지 않는다. 즉, 과거의 세션키와 전송 정보가 있는 경우와 그렇지 않은 경우에 공격자의 능력은 동일하므로, KKP 공격에 대해 안전하다.

또한, 공격자가 사용자 V로 위장하여 세션키를 설정하기 위해, 사용자 U로부터 t_U를 받은 후에, 랜덤한 수 $r_V \in_R Z_{p_e}$를 선택하여 t_V를 계산하고 이를 U에게 전송하면 세션키는 $C \equiv g_s^{x_U x_V} \bmod p_s \| g_e^{r_U r_V} \bmod p_e$가 된다. 이때, 공격자는 사용자 V의 비밀키 x_V를 알지 못하므로 Diffie-Hellman 문제를 해결하지 못하면 세션키를 계산할 수 없게 된다. 즉, 과거의 세션키와 전송 정보가 있는 경우와 그렇지 않은 경우에 공격자의 능력은 동일하므로 KKI 공격에 대해서도 안전하다.

AI, FS, KKP, KKI에 대한 dhHybrid2의 자세한 안전성 증명은 [정리 10], [정리 11], [정리 12], [정리 13]과 같다.

[정리 10] $dhHybrid2_{AI} \equiv_{tt}^{p} DH$

(증명)

① $dhHybrid2_{AI} \leq_{2-tt}^{p} DH$

$r_V \in_R Z_{p_e}$에 대해,

$dhHybrid2_{AI}(p_s, g_s, p_e, g_e, y_U, y_V, t_U)$

$\equiv (DH(p_s, g_s, y_U, y_V) \| DH(p_e, g_e, t_U, g_e^{r_V}), g_e^{r_V})$

② $DH \leq_{m}^{p} dhHybrid2_{AI}$

$r_V \in_R Z_{p_e}$에 대해,

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

$\equiv dhHybrid2_{AI}(p, g, p, B, A, B, A^{r_V^{-1}})$ □

[정리 11] $dhHybrid2_{FS} \equiv_{tt}^{p} DH$

(증명)

① $dhHybrid2_{FS} \leq_{2-tt}^{p} DH$

$dhHybrid2_{FS}(p_s, g_s, p_e, g_e, y_U, y_V, t_U, t_V, x_U, x_V)$

$\equiv DH(p_s, g_s, y_U, y_V) \| DH(p_e, g_e, t_U, t_V)$

② $DH \leq_{1-tt}^{p} dhHybrid2_{FS}$

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

$\equiv \lceil dhHybrid2_{FS}(p, g, p, g, A, B, A, B, x_U, x_V) \rceil^{|C|}$ □

[정리 12] $dhHybrid2_{KKP} \equiv_{tt}^{p} DH$

(증명)

① $dhHybrid2_{KKP} \leq_{2-tt}^{p} DH$

$dhHybrid2_{KKP}(p_s, g_s, p_e, g_e, y_U, y_V, t_U, t_V, t_U', t_V', C')$

$\equiv DH(p_s, g_s, y_U, y_V) \| DH(p_e, g_e, t_U, t_V)$

② $DH \leq_{1-tt}^{p} dhHybrid2_{KKP}$

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

$\equiv \lceil dhHybrid2_{KKP}(p, g, p, g, A, B, A, B, t_U', t_V', C') \rceil^{|C|}$

□

[정리 13] $dhHybrid2_{KKI} \equiv_{tt}^{p} DH$

(증명)

① $dhHybrid2_{KKI} \leq_{2-tt}^{p} DH$

$r_V \in_R Z_{p_e}$에 대해,

$dhHybrid2_{KKI}(p_s, g_s, p_e, g_e, y_U, y_V, t_U, t_U', t_V', C')$

$\equiv (DH(p_s, g_s, y_U, y_V) \| DH(p_e, g_e, t_U, g_e^{r_V}), g_e^{r_V})$

② $DH \leq_{1-tt}^{p} dhHybrid2_{KKI}$

$r_V \in_R Z_{p_e}$에 대해,

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

$\equiv \lceil dhHybrid2_{KKI}(p, g, p, B, A, B, A^{r_V^{-1}}, t_U', t_V', C') \rceil^{|C|}$

□

• dhHybridOneFlow 프로토콜

dhHybridOneFlow 프로토콜의 공개키 생성 단계에 비밀키가 g^{x_U} 형태로 포함되므로 능동적 공격자라 할지라도 AI 공격을 성공하는 것은 Z_p상에서 Diffie-Hellman 문제를 푸는 어려움과 동치이다. 즉, dhHybridOneFlow 프로토콜에 대해 AI 공격을 성공하는 것은 Diffie-Hellman 문제에 truth-table 귀착 가능하며 그 역도 성립한다[11].

dhHybridOneFlow 프로토콜은 일방향 프로토콜로 사용자 U에 대한 인증을 제공하지 못하므로 key compromise impersonation resilience 특성을 갖지 못하며, 사용자 V의 비밀키가 노출된 경우에 forward secrecy를 만족하지 못한다. 그러나 사용자 U의 비밀키가 노출된 경우에 공격자는 사용자 V의 비밀키 x_V를 알지 못하므로 Diffie-Hellman 문제를 해결하지 못하면 세션키를 계산할 수 없기 때문에 half forward secrecy를 제공한다.

dhHybridOneFlow 프로토콜은 세션키를 생성하는데 사용자 U가 선택한 랜덤 수가 포함되므로 과거의 세션키와 전송 정보를 획득하더라도 사용자가 서로 다른 난수를 이용하는 경우에 현재의 세션키를 계산하는데 도움이 되지 않는다. 즉, 과거의 세션키와 전송 정보가 있는 경우와 그렇지 않은 경우에 공격자의 능력은 동일하므로 KKP 공격에

〈표 4〉 ANSI X9.42의 이산대수 기반 키 분배 프로토콜의 안전성 분석 결과

Protocol	Passive	AI	KCI	FS	KKP	KKI
dhStatic	$\equiv^{p}_{tt}$ DH	공격 불가능	안전하지 않음	안전하지 않음	안전하지 않음	안전하지 않음
dhEphem	$\equiv^{p}_{tt}$ DH	안전하지 않음	안전하지 않음	$\equiv^{p}_{tt}$ DH	$\equiv^{p}_{tt}$ DH	안전하지 않음
dhOneFlow	$\equiv^{p}_{tt}$ DH	안전하지 않음	안전하지 않음	Half FS $\equiv^{p}_{tt}$ DH	$\equiv^{p}_{tt}$ DH	안전하지 않음
dhHybrid 1	$\equiv^{p}_{tt}$ DH	$\equiv^{p}_{tt}$ DH	안전하지 않음	$\equiv^{p}_{tt}$ DH	$\equiv^{p}_{tt}$ DH	$\equiv^{p}_{tt}$ DH
dhHybrid 2	$\equiv^{p}_{tt}$ DH	$\equiv^{p}_{tt}$ DH	안전하지 않음	$\equiv^{p}_{tt}$ DH	$\equiv^{p}_{tt}$ DH	$\equiv^{p}_{tt}$ DH
dhHybridOneFlow	$\equiv^{p}_{tt}$ DH	$\equiv^{p}_{tt}$ DH	안전하지 않음	Half FS $\equiv^{p}_{tt}$ DH	안전하지 않음	안전하지 않음

대해 안전하다. 그러나 이전 세션의 전송 정보와 세션키를 획득한 KKI 공격자가 이를 이용하여 다른 사용자로 위장하려는 경우에는 공격자가 사용자 V에게 임의의 다른 사용자로 위장할 수 있기 때문에 KKI 공격에 대해서는 안전하지 못하다.

AI, HFS, KKP에 대한 dhHybridOneFlow의 자세한 안전성 증명은 [정리 14], [정리 15], [정리 16]과 같다.

[정리 14] $\mathrm{dhHybridOneFlow}_{AI} \equiv^{p}_{tt} \mathrm{DH}$

(증명)

① $\mathrm{dhHybridOneFlow}_{AI} \leq^{p}_{2-tt} \mathrm{DH}$

$\mathrm{dhHybridOneFlow}_{AI}(p_s, g_s, y_U, y_V, t_U)$

$\equiv \mathrm{DH}(p_s, g_s, y_U, y_V) \parallel \mathrm{DH}(p_s, g_s, y_V, t_U)$

② $\mathrm{DH} \leq^{p}_{1-tt} \mathrm{dhHybridOneFlow}_{AI}$

$\mathrm{DH}(p, g, A, B)$

$\equiv \lceil \mathrm{dhHybridOneFlow}_{AI}(p, g, A, B, A) \rceil^{|C|}$ □

[정리 15] $\mathrm{dhHybridOneFlow}_{HFS} \equiv^{p}_{tt} \mathrm{DH}$

(증명)

① $\mathrm{dhHybridOneFlow}_{HFS} \leq^{p}_{2-tt} \mathrm{DH}$

$\mathrm{dhHybridOneFlow}_{HFS}(p_s, g_s, y_U, y_V, t_U, x_U)$

$\equiv \mathrm{DH}(p_s, g_s, y_U, y_V) \parallel \mathrm{DH}(p_s, g_s, y_V, t_U)$

② $\mathrm{DH} \leq^{p}_{1-tt} \mathrm{dhHybridOneFlow}_{HFS}$

$\mathrm{DH}(p, g, A, B)$

$\equiv \lceil \mathrm{dhHybridOneFlow}_{HFS}(p, g, A, B, A, x_U) \rceil^{|C|}$ □

[정리 16] $\mathrm{dhHybridOneFlow}_{KKP} \equiv^{p}_{tt} \mathrm{DH}$

(증명)

① $\mathrm{dhHybridOneFlow}_{KKP} \leq^{p}_{2-tt} \mathrm{DH}$

$\mathrm{dhHybidOneFlow}_{KKP}(p_s, g_s, y_U, y_V, t_U, t_U'.C')$

$\equiv \mathrm{DH}(p_s, g_s, y_U, y_V) \parallel \mathrm{DH}(p_s, g_s, y_V, t_U)$

② $\mathrm{DH} \leq^{p}_{1-tt} \mathrm{dhHybridOneFlow}_{KKP}$

$\mathrm{DH}(p, g, A, B)$

$\equiv \lceil \mathrm{dhHybridOneFlow}_{KKP}(p, g, A, B, A, t_U', C') \rceil^{|C|}$ □

〈표 4〉는 지금까지 살펴본 ANSI X9.42 키 분배 프로토콜의 안전성을 분석한 결과를 정리한 것이다.

6. 결 론

키 분배 프로토콜은 안전한 암호 시스템의 사용에 있어 가장 필수적인 요소이다. 네트워크 상에서의 안전한 메시지의 전송을 위한 암호 시스템의 사용이 증가함에 따라 키 분배 프로토콜에 대한 많은 연구가 진행되었으며 최근 들어 키 분배 프로토콜의 표준화 작업이 활발히 진행되고 있다.

본 논문에서는 이산대수 기반의 표준 키 분배 프로토콜인 ANSI X9.42의 Diffie-Helman형 키 분배 프로토콜들의 세션키 설정 과정 및 특징을 분석하고, 능동적 공격자 모델을 AI 공격, FS에 대한 공격, KCI 공격, KKS에 대한 공격으로 나누어 각각에 대한 안전성을 분석하였다. 본 논문에서 안전성 증명에 사용한 방식은 각각의 키 분배 프로토콜들의 안전성을 다항식 시간 안에 해결할 수 없는 수학적으로 어려운 문제들로 귀착시킴으로써 키 분배 프로토콜의 안전성을 증명하였다. 이러한 안전성 증명 방식은 해당하는 프로토콜이 지금까지 제안된 공격 방법들에 대해서만 안전하다는 것을 보장하는 기존의 경험적 측면의 안전성 분석 방법과는 차별되는 것으로, 이후에 기존의 공격 방법보다 강력한 새로운 공격 방법이 제안되는 경우라 할지라도, 프로토콜이 기반으로 하고 있는 해당 기반 논리가 깨지지 않는 한 그 안전성을 보장받을 수 있다.

이것은 키 분배 프로토콜과 여러 암호학적 기반 논리와의 상관 관계 연구에 기반이 되며, 각각의 키 분배 프로토콜들의 특징과 여러 가지 공격들에 대한 안전성을 자세히 분석함으로써, 향후 키 분배 프로토콜을 응용 분야에 적용할 경우, 해당 응용 분야에 가장 적합한 키 분배 방식을 선택하는 기준으로 활용될 것으로 기대된다.

참 고 문 헌

[1] W. Diffie, M. E. Hellman, "New directions in cryptogra-

phy," IEEE Trans. Inform. Theory, IT-22, 6, pp.644-654, 1976.
[2] IEEE P1363/D13, "Standard Specifications for Public Key Cryptography," 1999.
[3] RSA Laboratories Technical Note v1.4, "PKCS #3 : Diffie-Hellman Key Agreement Standard," 1993.
[4] ANSI X9.42, "Agreement of symmetric Key on Using Diffie-Hellman Cryptography," 2001.
[5] ANSI X9.63, "Public Key Cryptography for the financial services industry : key agreement and key transport using elliptic curve cryptography," 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.E82-A, No.1, January, 1999.
[7] R. A Rueppel and P. C vanOorschot, "Modern Key Agreement Techniques," *Computer communications*, Vol.17, No. 7, pp.458-465, 1994.
[8] H. Woll, "Reductions among number theoretic problems," *Information and Computation*, Vol.72, pp.167-179, 1987.
[9] K. Sakurai and H. Shizuya, "Relationships among the computational powers of breaking discrete log cryptosystems," Proc. Eurocrypt'95, LNCS 921, pp.341-355, Springer Verlag, 1995(J. Cryptology, Vol.11, pp.29-43, 1998).
[10] E. Bach, "Discrete Logarithms and factoring," Technical Report UCB/CSD 84/186, University of California, Computer Science Division (EECS), 1984.
[11] S. J. Kim, M. Mambo et al., "On the security of the Okamoto-Tanaka ID-Based Key Exchange scheme against Active attacks," IEICE Trans, pp.231-238, January, 2001.
[12] R. E. Neopolitan, Kumarss Niamipour, "Foundations of Algorithm," pp.409-444, 1999.

김 경 진

e-mail : kjkim@dosan.skku.ac.kr
2001년 성균관대학교 전기전자 및 컴퓨터 공학부(공학사)
2001년~현재 성균관대학교 정보통신공학부 석사 과정
관심분야 : 암호 프로토콜, 공개키 기반구조

김 성 덕

e-mail : sdkim@koscom.co.kr
1994년 성균관대학교 정보공학과(공학사)
1996년 성균관대학교 정보공학과 (공학석사)
1996년~1999년 한국전산원 초고속사업단
1996년~현재 성균관대학교 정보통신공학부 박사 과정
1999년~현재 한국증권전산(주) 전자인증사업팀
관심분야 : 공개키 기반구조, 인증서 검증, 암호 프로토콜

심 경 아

e-mail : kashim@kisa.or.kr
1992년 이화여자대학교 수학과(이학사)
1994년 이화여자대학교 대학원 수학과 (이학석사)
1999년 이화여자대학교 대학원 수학과 (이학박사)
2000년~현재 한국정보보호진흥원(KISA) 암호 기술팀 선임연구원
관심분야 : 암호 프로토콜

원 동 호

e-mail : dhwon@dosan.skku.ac.kr
성균관대학교 전자공학과(공학사, 공학석사, 공학박사)
1978년~1980년 한국전자통신연구소 전임 연구원
1985년~1986년 일본 동경공업대 객원연구원
1992년~1994년 성균관대학교 전자계산소 소장
1995년~1997년 성균관대학교 교학 처장
1996년~1998년 국가정보화 추진위원회 자문위원
1999년~2001년 성균관대학교 전기전자 및 컴퓨터공학부 학부장
1982년~현재 성균관대학교 정보통신공학부 교수
1999년~현재 BK 21 핵심분야(정보시스템보안기술) 팀장
2000년~현재 정통부지정 정보보호인증기술연구센터 센터장
2002년~현재 한국정보보호학회 회장
2002년~현재 대검찰청 컴퓨터 범죄 수사 자문위원
2002년~현재 성균관대학교 연구처장
2002년~현재 감사원 IT 감사 자문위원
관심분야 : 암호 이론, 정보 이론