情報保護學會論文誌
第15卷 第4號, 2005. 8

다중 독립 PKG 환경에서 인증된 신원기반 키 동의 프로토콜*

이 훈 정,[1†] 김 현 숙,[1] 김 상 진,[2] 오 희 국[1‡]
[1]한양대학교, [2]한국기술교육대학교

Authenticated Identity-based Key Agreement Protocols in a Multiple Independent PKG Environment*

Hoonjung Lee,[1†] Hyunsook Kim,[1] Sangjin Kim,[2] Heekuck Oh[1‡]
[1]Hanyang University, [2]Korea University of Technology and Education

요 약

현재까지 제안된 대부분의 신원기반 키 동의 프로토콜은 단일 PKG(Private Key Generator) 환경을 고려하고 있다. 2002년에 Chen과 Kudla는 처음으로 다중 PKG 환경을 고려한 신원기반 2자간 키 동의 프로토콜을 제안하였지만 이들은 PKG의 마스터키를 제외한 다른 시스템 파라미터는 공유한다고 가정하고 있다. 그러나 PKG의 마스터키 이외에 다른 파라미터마저도 공유하지 않는 것이 보다 현실적인 가정이다. 이 논문에서는 시스템 파라미터를 공유하지 않는 각 PKG들로부터 개인키를 발급받은 두 사용자들 간의 2자간 키 동의 프로토콜과 이를 확장한 두 가지 버전의 3자간 키 동의 프로토콜을 제안한다. 제안된 2자간 키 동의 프로토콜은 다중 PKG 환경에서 최소의 pairing 연산량을 요구하며, 제안된 3자간 키 동의 프로토콜은 다중 PKG 환경에서 기존의 3자간 키 동의 프로토콜들에 비해 효율적이다. 또한 제안된 키 동의 프로토콜들은 키 동의 프로토콜의 보안 요구사항을 만족한다.

ABSTRACT

To date, most identity-based key agreement protocols are based on a single PKG (Private Key Generator) environment. In 2002, Chen and Kudla proposed an identity-based key agreement protocol for a multiple PKG environment, where each PKG shares identical system parameters but possesses distinct master key. However, it is more realistic to assume that each PKG uses different system parameters including the PKG's master key. In this paper, we propose a new two party key agreement protocol between users belonging to different PKGs that do not share system parameters. We also extend this protocol to two types of tripartite key agreement protocols. We show that our two party protocol requires minimal amount of pairing computation for a multiple PKG environment and our tripartite protocol is more efficient than existing protocols. We also show that the proposed key agreement protocols satisfy every security requirements of key agreement protocol.

Keywords : *ID-based cryptosystem, bilinear map, key agreement protocol, multiple independent PKG*

접수일 : 2005년 2월 11일 ; 채택일 : 2005년 8월 12일
* 이 논문은 2004년도 한국학술진흥재단의 지원에 의하여 연구되었음 (KRF-2004-041-D00683)
† 주저자, leehj@cse.hanyang.ac.kr
‡ 교신저자, hkoh@cse.hanyang.ac.kr

Ⅰ. 서 론

키 확립 프로토콜은 비밀통신을 하고자 하는 개체들 간에 공유할 비밀키를 안전하게 생성하는 암호학적 도구이다. 이러한 비밀키는 보통 안전한 통신채널을 구축하기 위한 세션키로 사용된다. 키 확립 프로토콜은 크게 키 전송(key transport) 프로토콜과 키 동의(key agreement) 프로토콜로 나누어진다. 키 전송 프로토콜은 키 확립에 참여하는 개체들 중 하나가 통신에 사용할 비밀키를 생성해 다른 개체들에게 전달하지만 키 동의 프로토콜은 키 확립에 참여하는 모든 개체들이 통신에 사용할 비밀키 생성에 동등하게 기여한다. 키 교환 프로토콜에서는 특정 개체가 비밀키를 생성하므로 다른 참여자들은 이 개체를 신뢰해야 한다. 따라서 보통 신뢰할 수 있는 제 3의 개체를 사용한다. 이렇게 제 3의 개체를 사용하는 방식은 본질적으로 다음과 같은 문제점을 지니고 있다. 첫째, 키를 생성하고 분배하는 개체는 좋은 공격대상이 되며, 이 개체에 대한 공격이 성공되면 시스템 전체의 안전성을 전혀 보장할 수 없다. 둘째, 키 생성 개체는 프로토콜 수행마다 항상 온라인으로 참여해야 하므로 병목현상이 발생할 수 있다. 하지만 키 동의 프로토콜에서는 키 확립에 참여하는 모든 개체들이 모두 키 생성에 참여하므로 위에서 언급한 키 교환 프로토콜이 가지는 문제들을 지니고 있지 않다. 이 논문에서는 후자인 키 동의 프로토콜에 대해서 다룬다.

1976년에 Diffie와 Hellman은 비대칭 암호기법을 이용한 키 동의 프로토콜을 처음으로 제안하였다.[1] 하지만 Diffie-Hellman 키 동의 프로토콜은 인증기능이 없어 중간자 공격(man-in-the-middle-attack)이 가능하다는 약점을 가지고 있었다. 인증기능이 있는 키 동의 프로토콜은 인증된 키 동의(AK, Authenticated Key agreement) 프로토콜이라 하며, 현재까지 이러한 AK에 대한 연구들이 많이 진행되었다.[2,3]

1984년, Shamir는 이메일 주소나 주민등록번호처럼 사용자의 잘 알려진 신원정보로부터 그 사용자의 공개키를 유도해 내는 신원기반 공개키 암호시스템(identity-based public key cryptosystem)의 개념과 이를 이용한 서명기법을 처음으로 소개하였다.[4] 신원기반 공개키 암호시스템에서 사용자의 공개키에 대응되는 개인키는 PKG(Private Key Generator)라 불리는 신뢰기관에 의해 발행된다. 그 결과, PKG는 자신으로부터 개인키를 발급받은 모든 사용자들의 암호문을 해독하거나 서명을 위조할 수 있는 능력을 가진다. 이런 PKG의 키 발급문제는 앞으로 신원기반 공개키 암호시스템의 대중화를 위해 해결해야할 문제이며 현재 활발히 연구 중인 분야이다.

Shamir의 제안 이후 신원기반 공개키 암호시스템은 유한체에서의 이산대수문제와 타원곡선상의 이산대수문제의 어려움을 기반으로 하는 기존 공개키 암호시스템을 기반으로 연구가 진행되었다.[5-7] 2001년, Boneh와 Franklin은 타원곡선상의 Weil pairing을 이용한 신원기반 암호기법을 제안하였다.[8] 이는 최초의 실용 가능한 신원기반 암호기법으로, 이후 신원기반 공개키 암호시스템에 관한 대부분의 연구들은 이들의 기법을 따르고 있다.[9]

Pairing을 이용한 신원기반 2자간 키 동의 기법은 2001년 Smart에 의해 처음으로 제안되었다.[10] 하지만 Smart의 프로토콜은 신원기반 키 동의 프로토콜의 보안 요구사항 중 PKG 전방향 안전성(forward secrecy)를 만족하지 못하였다. 2002년, *Chen*과 *Kudla*는 *pairing*을 이용한 세 가지 신원기반 2자간 키 동의 프로토콜을 제안하였다.[11] 첫째, PKG 전방향 안전성이 만족되도록 Smart의 프로토콜을 확장한 프로토콜을 제안하였고, 둘째, Smart의 프로토콜보다 연산량이 적은 새로운 프로토콜을 제안하였으며, 셋째, 개인키를 발급하는 PKG가 다수 존재하는 다중 PKG 환경에서의 2자간 키 동의 프로토콜을 제안하였다. 세 번째 프로토콜은 다중 PKG 환경을 고려한 최초의 신원기반 2자간 키 동의 프로토콜이다. 이들은 안전성 증명 부분을 보완해 2004년에 다시 발표하였다.[12] 2004년 McCullagh와 Barreto는 다중 PKG 환경에서의 2자간 키 동의 프로토콜을 제안하였는데 기존의 Smart나 Chen과 Kudla의 프로토콜이 Boneh와 Franklin의 키 생성 방법을 따른 반면, 이들은 다른 키 생성 방법을 이용하며 Weil pairing 대신 Tate pairing을 사용한다.[13] 이 논문에서 제안하는 프로토콜은 Smart나 Chen과 Kudla의 프로토콜과 마찬가지로 Boneh와 Franklin의 키 생성 방법을 따른다.

신원기반 3자간 키 동의 프로토콜에 관한 연구는 2000년에 Joux가 제안한 pairing을 이용한 3자간 키 동의 프로토콜에 의해 시작된다.[14] 이전까지의 키 동의 프로토콜은 2자간 키 동의 프로토콜과 2자

이상의 다자간 키 동의 프로토콜로 분류되었으나 Joux는 pairing의 특성을 이용한 3자간 키 동의라는 새로운 키 동의 프로토콜을 처음으로 제안하였다. 3자간 키 동의의 경우 전자상거래처럼 소비자, 판매자 외에 분쟁 발생 시, 이 분쟁해결을 위한 판결자가 필요한 경우 유용하게 사용될 수 있다. 2003년, Al-Riyami 등은 Joux의 프로토콜은 인증기능이 없음을 지적하고 Joux의 프로토콜의 단점을 보완한 pairing 기반의 인증된 3자간 키 동의 프로토콜을 제안하였다.[15] 최초의 신원기반 3자간 키 동의 프로토콜은 2002년 Zhang 등에 의해 제안되었다.[16] 2003년, Nalla와 Reddy는 Zhang 등의 프로토콜보다 효율적인 신원기반 3자간 키 동의 프로토콜을 제안하였다.[17] 그러나 Nalla와 Reddy의 프로토콜은 후에 Chen에 의해 도청된 메시지를 통해 세션키 계산이 가능한 허점이 있다는 것이 밝혀졌다.[18] 2003년, Shim은 Nalla와 Reddy의 프로토콜은 Chen이 지적한 도청공격 외에 중간자 공격 또한 가능하다고 지적하면서 중간자 공격을 방어할 수 있으면서 Zhang 등의 프로토콜보다 연산량을 줄인 신원기반 3자간 키 동의 프로토콜을 제안하였다.[19] 2005년, 박영호와 이경현은 Shim과 유사한 새로운 신원기반 3자간 키 동의 프로토콜을 제안하였다. 이 논문은 Shim과 달리 한번에 복수 개의 키를 합의할 수 있다는 장점이 있다.[20]

신원기반 공개키 암호시스템에서 사용자의 개인키는 PKG가 발급한다. 즉, PKG는 모든 사용자의 개인키를 복구할 수 있는 강력한 권한을 가지고 있다. 따라서 PKG는 모두가 믿을 수 있는 중립적인 신뢰기관이어야 한다. 한 기관에 속한 사용자들이 같은 PKG로부터 개인키를 발급받는다는 것은 현실적인 가정이지만 서로 이해관계가 상충되는 기관에 속한 사용자들이 같은 PKG로부터 개인키를 발급받는다는 것은 현실적이지 못하다. 더 나아가 서로 다른 나라의 기관에 속한 사용자들이 같은 PKG로부터 개인키를 발급받는다는 것은 더욱 현실적이지 못하다. 이런 관점에서 보았을 때, 서로 다른 PKG가 다수 존재하는 다중 PKG 환경을 고려한 Chen과 Kudla의 프로토콜과 McCullagh와 Barreto의 프로토콜은 이전 프로토콜들에 비해 현실적이라 할 수 있다. 하지만 이들 논문에서 가정한 다중 PKG 환경은 PKG의 마스터키만 다르고, 다른 파라미터들은 모두 같은 것을 사용한다. 물론 McCullagh와 Barreto가 지적한 것처럼 표준화된 유한체상의 타원곡선 군을 모든 PKG가 사용한다고 가정할 수 있지만 이런 가정은 여전히 매우 제한적이다. 다중 PKG 환경에서 모든 PKG들이 인정할 수 있는 파라미터를 설정하여 공유하는 것이 어려우며, 안전성 측면에서도 각 PKG들이 마스터키뿐만 아니라 모든 파라미터를 다르게 사용하는 것이 바람직하다. 따라서 이 논문에서는 Chen과 Kudla, McCullagh와 Barreto가 제안한 다중 PKG 환경보다 더욱 독립적인 즉, PKG마다 PKG의 마스터키뿐만 아니라 그 외의 모든 시스템 파라미터를 독립적으로 선택하여 사용하는 완전한 다중 PKG 환경을 고려한 신원기반 키 동의 프로토콜을 제안한다.

현재까지 제안된 대부분의 키 동의 프로토콜들은 단일 PKG 환경을 고려하고 있다. 이런 프로토콜을 확장하여 쉽게 다중 PKG 환경을 위한 키 동의 프로토콜을 만들 수 있을 것으로 생각할 수도 있다. 예를 들어 각 PKG 환경에서 기존 프로토콜을 각각 수행한 후에 결과 키들을 하나로 결합하여 사용할 수 있다고 생각할 수 있다. 하지만 각 사용자들은 하나의 PKG로부터만 개인키를 발급받은 상태이므로 이 방법을 사용하는 것은 어렵다. 따라서 다중 PKG 환경을 위한 키 동의 프로토콜을 만들기 위해서는 세션키를 생성할 때 입력으로 사용되는 서로 다른 환경에서 제공되는 각 사용자의 일시적 비밀값을 인증하고 이 값들을 사용하여 세션키를 만드는 방법을 제공해야 한다. 이 논문에서는 최소의 pairing 연산을 사용하는 다중 PKG 환경을 위한 신원기반 2자간 키 동의 프로토콜과 이를 확장한 신원기반 3자간 키 동의 프로토콜을 제안한다. 이는 완전한 다중 PKG 환경을 고려한 최초의 시도이다. 또한 제안된 프로토콜들은 키 동의 프로토콜을 충족해야 하는 모든 보안 요구사항을 만족한다. 이 논문에서는 두 종류의 3자간 키 동의 프로토콜을 제안한다. 하나는 세션키를 생성할 때 기존의 신원기반 3자간 키 동의 방법들처럼 pairing을 이용하며, 다른 하나는 pairing의 특성을 이용하지 않고 세션키를 생성한다. 제안하는 3자간 키 동의 프로토콜 역시 최소의 pairing 연산을 사용한다.

이 논문의 구성은 다음과 같다. 2장에서는, 이 논문의 이해를 돕기 위한 수학적 배경과 키 동의 프로토콜이 만족해야 하는 보안 요구사항 및 기존에 제안되었던 신원기반 키 동의 프로토콜 등에 대해 설명한다. 3장에서는 제안하는 프로토콜을 자세히 기술하며, 4장에서는 제안하는 프로토콜의 안전성과

효율성을 분석한다. 끝으로 5장에서는 결론과 향후 연구방향을 제시한다.

Ⅱ. 연구 배경

2.1 수학적 배경과 관련 암호학적 문제

Bilinear pairing은 원래 Weil pairing 연산을 이용한 MOV (Menezes-Okamoto-Vanstone) 공격[21]이나 Tate pairing을 이용한 FR(Frey-Ruck) 공격[22]처럼 타원곡선상의 이산대수 문제를 유한체상의 이산대수 문제로 축소시켜 그 어려움을 줄여 타원곡선 암호시스템을 공격하는 도구로 사용되었다.

Pairing 연산을 이용하면 타원곡선상의 DDHP (Decision Diffie-Hellman Problem)를 쉽게 풀 수 있기에 이런 공격들이 가능했다. 그러나 2000년에 Joux는 bilinear pairing이 공격 도구가 아닌 정보보호를 위한 암호학적 도구로 사용될 수 있음을 보였다. Joux는 그의 논문에서 Weil pairing을 이용한 간단한 3자간 Diffie-Hellman 키 동의 기법을 보여주었다.[14] 키 동의 이외에도 bilinear pairing을 이용한 여러 암호학적 기법들이 활발히 연구되고 있다.[23] 현재 연구되고 있는 대부분의 신원기반 공개키 암호시스템들도 이러한 pairing을 이용해 구현되고 있다.

이 논문에서는 앞으로 다음과 같은 표기법을 사용한다. 1) q는 매우 큰 소수를 의미한다. 2) $\mathbb{G}_1$과 $\mathbb{G}_2$는 모두 위수가 q인 군으로서, $\mathbb{G}_1$은 타원곡선위의 덧셈군이고 $\mathbb{G}_2$는 유한체위의 곱셈군이다. 3) P, Q, R은 $\mathbb{G}_1$의 임의의 원소들이다. 4) a, b, c는 $\mathbb{Z}_q^*$의 임의의 원소들이다.

정의 1 (Admissible bilinear map). 다음과 같은 조건들을 만족하는 함수 $\hat{e}: \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$를 admissible bilinear map이라 한다.

- **Bilinear**: 임의의 $P, Q, R \in \mathbb{G}_1$에 대해 다음이 성립해야 한다.
  - $\hat{e}(P+Q, R) = \hat{e}(P, R) \cdot \hat{e}(Q, R)$
  - $\hat{e}(P, Q+R) = \hat{e}(P, Q) \cdot \hat{e}(P, R)$
- **Non-Degenerate**: $\mathbb{G}_1$의 모든 쌍 P, Q에 대해 $\hat{e}(P, Q)$는 $\mathbb{G}_2$의 항등원이 아니어야 한다.
- **Computable**: 임의의 $P, Q \in \mathbb{G}_1$에 대하여 $\hat{e}(P, Q)$를 계산할 수 있는 효율적인 알고리즘이 존재해야 한다.

Bilinear 특성에 의해 다음 특성을 추가적으로 유도할 수 있다.

$$\hat{e}(aP, bQ) = \hat{e}(P, bQ)^a = \hat{e}(aP, Q)^b = \hat{e}(P, Q)^{ab} = \hat{e}(abP, Q) = \hat{e}(P, abQ)$$

정의 2 (DLP (Discrete Logarithm Problem) in $\mathbb{G}_1$). $\mathbb{G}_1$의 원소 P와 aP가 주어졌을 때, $a(\in \mathbb{Z}_q)$를 계산하는 문제를 말한다.

정의 3 (CDHP (Computational Diffie-Hellman Problem) in $\mathbb{G}_1$). $\mathbb{G}_1$의 원소 P, aP, bP가 주어졌을 때, $abP(\in \mathbb{G}_1)$를 계산하는 문제를 말한다.

정의 4 (BDHP (Bilinear Diffie-Hellman Problem) in $\mathbb{G}_1$ and $\mathbb{G}_2$). $\mathbb{G}_1$의 원소 P, aP, bP, cP 가 주어졌을 때, $\hat{e}(P, P)^{abc} (\in \mathbb{G}_2)$를 계산하는 문제를 말한다.

현재까지 DLP, CDHP, BDHP를 다항시간 내에 계산하는 것은 계산적으로 어렵다고 알려져 있다.[8] 이 논문에서 제안하는 프로토콜의 안전성은 위의 문제들을 다항시간 내에 계산하는 것이 어렵다는 가정에 기반하고 있다.

2.2 키 동의 프로토콜의 보안 요구사항

다음은 키 동의 프로토콜이 만족해야 하는 보안 요구사항들이다. 이 중에서는 신원기반 공개키 암호시스템을 사용할 경우에만 적용되는 특성들도 있다.

- 알려진 키 안전성(known-key security): 프로토콜이 진행될 때마다 유일하고 독립적인 세션키가 생성되어야 한다. 공격자가 이전에 생성된 세션키를 알아냈다 하더라도 그것을 이용하여 그 이전에 생성되었던 세션키나 앞으로 생성될 세션키를 알아내는 것이 계산적으로 어려워야 한다.

- 전방향 안전성(Forward secrecy): 공격자가 세션키 생성에 참여한 사용자들의 개인키를 하나 또는 그 이상 알아냈다 하더라도 이전에 생성된 세션 키를 알아내는 것이 계산적으로 어려워야 한다. 신원기반 키 동의 프로토콜에서는 다음과 같이 보다 세분화할 수 있다.
  - 완전한 전방향 안전성(Perfect forward secrecy): 세션키 생성에 참여하는 모든 사용자들의 개인키가 노출된다 하더라도 이전 세션키를 얻는 것이 계산적으로 어려워야 한다.
  - PKG 전방향 안전성(PKG forward secrecy): 개인키를 발급하는 PKG의 마스터키가 노출된다 하더라도 이전 세션키를 얻는 것이 계산적으로 어려워야 한다.
- 키노출 저항성(key-compromise resilience): 공격자가 사용자 A의 개인키를 알아내어도 다른 사용자로 위장하여 A와 프로토콜을 성공적으로 수행하는 것이 계산적으로 어려워야 한다.
- 미지의 키공유 저항성(unknown key-share resilience): 참여자가 생각하고 있는 사용자가 아닌 다른 사용자와 키를 동의하도록 참여자를 속이는 것이 계산적으로 어려워야 한다.
- 키 제어(key control): 참여자 중 어느 누구도 공유되는 세션키가 사전에 계산되거나 선택되어진 값이 되도록 만드는 것은 계산적으로 어려워야 한다.

PKG 전방향 안전성은 완전한 전방향 안전성보다 강력한 개념으로 신원기반 공개키 암호시스템의 특성 때문에 생긴 보안 요구사항이다. 신원기반 공개키 암호시스템에서 PKG의 마스터키 노출은 그 PKG로부터 개인키를 발급받은 모든 사용자의 개인키의 노출을 의미한다. 즉, PKG 전방향 안전성을 만족하면 완전한 전방향 안전성은 당연히 만족된다. 하지만 그 역은 항상 참은 아니다.

2.3 Pairing을 이용한 신원기반 키 동의 프로토콜

이 절에서는 다음과 같은 표기법을 사용하여 프로토콜들을 설명한다. 1) P는 위수가 소수 q인 덧셈군 $\mathbb{G}_1$의 임의의 생성자이다. 2) a, b, s는 $\mathbb{Z}_q^*$의 원소이다. 3) g는 위수가 소수 q인 곱셈군 $\mathbb{G}_2$의 생성자이다. 4) PKG의 마스터키는 s이고 공개키는 $P_{pub} = sP$이다. 5) ID_A는 참여자 A의 신원정보를 의미한다. 6) $Q_A = H_1(Q_A)$는 참여자 A의 공개키이다. 여기서 $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1$는 임의의 길이의 문자를 $\mathbb{G}_1$의 원소로 사상해주는 암호학적 해쉬함수이다. 7) $S_A = sQ_A$는 참여자 A의 개인키로 PKG에 의해 발급된다. 8) $H_2 : \mathbb{G}_2 \rightarrow \{0,1\}^k$는 $\mathbb{G}_2$의 원소를 k길이의 문자로 사상해주는 암호학적 해쉬함수이다. 여기서, k는 프로토콜에서 사용되는 부분 세션키의 길이를 의미한다.

Diffie-Hellman 방식의 키 동의 프로토콜을 타원곡선 기반으로 바꿀 경우 aP, bP를 서로 교환하여 abP 형태의 세션키를 사용할 수 있다. 하지만 이렇게 할 경우에는 중간자 공격이 가능하다. 이 공격을 방어하기 위한 가장 단순한 방법은 교환되는 aP, bP 값을 서명하는 것이다. Pairing을 이용한 신원기반 공개키 암호시스템을 사용하면 기본적으로 서명을 확인할 때 두 개의 pairing 연산이 요구된다. 중간자 공격을 방어하기 위한 또 다른 방법은 Smart가 제안한 것처럼 aP, bP 값은 서명하지 않고 세션키를 생성하는 방법을 제한하여 오직 적법한 참여자만 세션키를 계산할 수 있도록 하여 방어할 수 있다.

Msg 1. $A \rightarrow B$: $T_A = aP$

Msg 2. $B \rightarrow A$: $T_B = bP$

그림 1. Smart의 신원기반 2자간 키 동의 프로토콜

그림 1은 Smart가 제안한 신원기반 키 동의 프로토콜이다. 이 프로토콜에서 두 참여자 A와 B는 메시지 1과 2를 교환한 후, A는 임시키 $K_{AB} = \hat{e}(aQ_B, P_{Pub})\hat{e}(S_A, T_B)$를 계산하고, B는 임시키 $K_{BA} = \hat{e}(bQ_A, P_{Pub})\hat{e}(S_B, T_A)$를 계산한다. 두 참여자는 이 두 키를 공유하게 되고 각각 키 유도함수 H_2를 이용해 세션키 $SK = H_2(K_{AB}) = H_2(K_{BA})$를 계산하게 된다. 이 프로토콜은 서명을 이용하는 방법과 마찬가지로 각 사용자마다 두 개의 pairing 연산이 요구되지만 교환하는 메시지의 크기 면에서 우수하다. 하지만 PKG는 $K_{AB} = \hat{e}(S_B, T_A)\hat{e}(S_A, T_B)$를 통해

Msg 1. $A \rightarrow B$: $W_A = aQ_A$, $T_A = aP$
Msg 2. $B \rightarrow A$: $W_B = bQ_B$, $T_B = bP$

그림 2. Chen과 Kudla의 신원기반 2자간 키 동의 프로토콜 1

항상 세션키를 계산할 수 있으므로 PKG 전방향 안전성을 만족하지 못한다는 단점을 가지고 있다.

Chen과 Kudla는 각 사용자마다 하나의 pairing 연산만을 요구하면서 PKG 전방향 안전성을 만족하는 그림 2와 같은 신원기반 키 동의 프로토콜을 제안하였다. 이 프로토콜은 각 사용자마다 하나의 pairing 연산만을 요구하면서 PKG 전방향 안전성을 만족한다. 이 프로토콜에서 두 참여자 A와 B는 메시지 1과 2를 교환 후, A는 임시키 $K_{AB} = \hat{e}(S_A, W_B + aQ_B)$를 계산하고, B는 $K_{BA} = \hat{e}(S_B, W_A + bQ_A)$를 계산한다. 그 다음 두 참여자는 키 유도함수 H_2'를 이용해 세션키 $SK = H_2'(K_{AB}, abP) = H_2'(K_{BA}, abP)$를 계산하게 된다. 여기서 $H_2' : \mathbb{G}_2 \times \mathbb{G}_1 \rightarrow \{0,1\}^l$는 $\mathbb{G}_2$의 원소와 $\mathbb{G}_1$의 원소를 입력값으로 받아 세션키의 크기인 l길이의 비트열로 사상해주는 암호학적 해쉬함수이다. 여기서 $\mathbb{G}_1$의 원소로 입력받는 abP는 참여자 A와 B 이외에는 계산할 수 없다. 즉 PKG는 $K_{AB} = \hat{e}(Q_A, W_B)^s \cdot \hat{e}(W_A, Q_B)^s$를 통해 K_{AB}를 계산할 수 있지만 abP를 계산할 수 없으므로 이 프로토콜은 PKG 전방향 안전성을 만족한다.

Chen과 Kudla는 그림 3과 같은 마스터키를 제외한 동일한 시스템 파라미터를 공유하는 두 개의 서로 다른 PKG로부터 개인키를 발급받은 사용자간에 키 동의 프로토콜을 제안하였다. 이 프로토콜에서 참여자 A는 PKG_1로부터 개인키를 발급 받은 사용자이며, 참여자 B는 PKG_2로부터 개인키를 발급받은 사용자이다. 이 프로토콜에서 사용자 A에게 개인키를 발급해주는 PKG_1의 마스터키는 $s_1 (\in \mathbb{Z}_q^*)$, 공개키는 $P_{pub}^1 = s_1 P$, 참여자 A의 개인키는 $S_A = s_1 Q_A$가 된다. 이와 마찬가지로 사용자 B에게 개인키를 발급해주는 PKG_2의 개인키는 $s_2 (\in \mathbb{Z}_q^*)$, 공개키는 $P_{pub}^2 = s_2 P$, 참여자 B의 개인키는 $S_B = s_2 Q_B$가 된다. 나머지 표기법은 단일 PKG 환경에서의 2자간 키 동의 프로토콜과 같다. 이 프로토콜에서 참여자 A와 B는 메시지 1, 2를 교환한 후, A는 임시키 $K_{AB} = \hat{e}(S_A, T_B) \cdot \hat{e}(Q_B, as_2 P)$를 B는 임시키 $K_{BA} = \hat{e}(S_B, T_A) \cdot \hat{e}(Q_A, bs_1 P)$를 계산한 후 두 참여자는 키 유도함수 H_2'를 이용해 세션키 $SK = H_2'(K_{AB}, abP) = H_2'(K_{BA}, abP)$를 계산한다. 이 프로토콜에서 두 PKG PKG_1과 PKG_2는 시스템 파라미터로 동일한 $\mathbb{G}_1$, $\mathbb{G}_2$, P를 사용한다. 만약 두 PKG가 서로 완전히 다른 시스템 파라미터를 사용할 경우에는 두 PKG가 사용하는 군이 다르기 때문에 위와 같은 pairing 연산을 사용하는 것이 어렵다.

3자간 키 동의의 경우 유한체위의 곱셈순환군을 이용할 경우에는 3자간 세션키로 g^{abc} 형태의 키를 공유하기가 어렵다. 하지만 pairing을 이용할 경우에는 pairing 연산의 특성을 이용해 3자간 세션키로 g^{abc} 형태인 $\hat{e}(P,P)^{abc}$를 세션키로 공유하여 사용할 수 있다. 따라서 3자간 키 동의에서는 aP, bP, cP에 서명하여 교환함으로써 단일 PKG 환경에서는 신원기반 3자간 키 동의 프로토콜을 쉽게 만들 수 있다. 이 경우에는 각 사용자마다 두 개의 값에 대한 서명을 확인하기 위해 네 번 그리고 최종 세션키를 생성하기 위해 한 번, 총 다섯 번의 pairing 연산이 요구된다.

Shim은 그림 4와 같은 신원기반 3자간 키 동의 프로토콜을 제안하였다. 이 프로토콜에서 참여자 A, B, C는 메시지 1, 2, 3을 교환한 후, A는 $\hat{e}(P, V_B + V_C)$와 $\hat{e}(P_{pub}, H(U_B)Q_B + H(U_C)Q_C + U_B + U_C)$가 동일한 값인지 검증한 후, 세션키

Msg 1. $A \rightarrow B$: $T_A = aP$
Msg 2. $B \rightarrow A$: $T_B = bP$

그림 3. Chen과 Kudla의 신원기반 2자간 키 동의 프로토콜 2

Msg 1. $A \rightarrow B, C$: $U_A = aP$,
$V_A = H(U_A)S_A + aP_{Pub}$
Msg 2. $B \rightarrow C, A$: $U_B = bP$,
$V_B = H(U_B)S_B + bP_{Pub}$
Msg 3. $C \rightarrow A, B$: $U_C = cP$,
$V_C = H(U_C)S_C + cP_{Pub}$

그림 4. Shim의 신원기반 3자간 키 동의 프로토콜

$SK=\hat{e}(U_B,U_C)^a=\hat{e}(P,P)^{abc}$를 계산한다. B도 A와 같은 방법으로 $\hat{e}(P,V_C+V_A)$와 $\hat{e}(P_{pub},H(U_C)Q_C+H(U_A)Q_A+U_C+U_A)$가 같은지 검증한 후, 세션키 $SK=\hat{e}(U_C,U_A)^b=\hat{e}(P,P)^{abc}$를 계산한다. C 역시 마찬가지로 $\hat{e}(P,V_A+V_B)$와 $\hat{e}(P_{pub},H(U_A)Q_A+H(U_B)Q_B+U_A+U_B)$가 동일한지 검증한 후, 세션키 $SK=\hat{e}(U_A,U_B)^c=\hat{e}(P,P)^{abc}$를 계산한다.

앞서 설명한 바와 같이 서명을 이용하는 3자간 키 동의 프로토콜의 경우에는 기본적으로 각 참여자는 자신 이외의 두 참여자의 값을 인증하기 위해 참여자당 2번의 pairing연산, 총 4번의 pairing 연산과 세션키 생성을 위한 1번의 pairing 연산 즉, 총 5번의 pairing 연산이 필요하다. 그러나 Shim이 제안한 기법은 키 동의에 참여하는 모든 참여자가 동일한 $\mathbb{G}_1$과 P를 사용하므로 2번의 pairng 연산을 통해 다른 두 참여자의 값을 동시에 인증할 수 있다.

만약 각 사용자의 PKG가 사용하는 시스템 파라미터가 완전히 다른 경우에는 하나의 식을 통해 두 개의 값을 동시에 인증하는 것은 어렵다. 또 만약 가능하다고 하더라도 세 참여자가 속해있는 PKG 중 어떤 PKG의 $\mathbb{G}_1$과 P를 사용할 것인지 결정해야 한다. 그런데 참여자들은 자신의 PKG가 선택한 군이 아닌 다른 군을 사용하는 것에 쉽게 동의하기는 어려울 것이다.

3자간의 키 동의의 경우에는 2자간 키 동의 프로토콜에서 사용한 방법처럼 교환하는 값에 서명하는 방법이 아닌 세션키 생성을 통해 다른 참여자를 인증하는 하는 방법을 사용하는 것은 어렵다. 먼저 2자간을 그대로 확장하여 3자간 프로토콜에 적용하는 방법을 생각할 수 있다. 즉, A와 B, A와 C, B와 C간에 2자간 키 동의 프로토콜을 수행한 다음에 결과 키들을 결합하여 사용하는 것이다. 하지만 이 경우 A는 B와 C간에 동의된 키를 계산할 수 없다는 문제점이 있다. 만약 계산할 수 있다면 2자간 키 동의 프로토콜 자체가 안전하지 않다는 것을 말한다. 이 문제를 극복하기 위해 1라운드에서는 2자간 키 동의 프로토콜을 통해 키 동의 프로토콜을 수행한 다음에 B가 A와 B간에 동의된 키를 이용하여 B와 C간에 동의된 키를 암호화하여 전달할 수 있다. Chen과 Kudla의 2자간 키 동의 프로토콜을 사용하였다면 이 방식의 키 동의 프로토콜은 각 개체마다 총 2번의 pairing 연산만을 이용하여 키 동의를 할 수 있다. 이렇게 두 라운드를 통해 세션키를 생성하지 않고 pairing 연산을 활용하여 한 라운드를 통해 세션키를 생성하는 것은 증명하고 있지는 못하지만 가능하지 않을 것으로 생각되며, 이렇게 한 라운드를 통해 세션키를 생성할 수 있다고 하더라도 앞서 언급한 방법에 비해 pairing의 수가 많이 요구될 것으로 생각된다.

Ⅲ. 제안하는 프로토콜

제안하는 프로토콜은 서로 다른 시스템 파라미터를 사용하는 PKG가 다수 존재하는 다중 PKG 환경에서 각 PKG들로부터 개인키를 발급받은 사용자들 간의 키 동의 프로토콜이다. 이 장에서는 먼저 다중 PKG 환경에서의 키 동의를 위한 시스템 설정과 키 동의에 대해 설명하고, 그 다음 다중 PKG 환경을 위한 2자간 키 동의 프로토콜과 이를 확장한 3자간 키 동의 프로토콜을 제안한다. 제안하는 3자간 키 동의 프로토콜의 경우 기존의 프로토콜들처럼 pairing을 이용한 방법과 pairing을 이용하지 않는 방법, 두 가지 방법을 제안한다.

3.1 시스템 설정

제안하는 키 동의 프로토콜은 기본적으로 Boneh와 Franklin의 키 생성 방법을 따른다. 그러나 제안하는 시스템은 총 n개의 PKG가 존재한다. 각 PKG들은 모든 시스템 파라미터를 독립적으로 선택하여 사용하므로 각 PKG를 구별하기 위해 PKG_i $(0\le i\le n)$로 표기한다. 제안하는 시스템은 다음과 같은 시스템 파라미터를 가지며 이후부터는 다음에서 정의한 표기법을 사용한다.

- PKG_i가 선택하는 기본적인 시스템 파라미터: $\langle \mathbb{G}_1^{(i)}, \mathbb{G}_2^{(i)}, P^{(i)}, \hat{e}^{(i)} \rangle$. $\mathbb{G}_1^{(i)}$은 위수 $q^{(i)}$인 타

Msg 1. $A\rightarrow B$:

$T_{AB}^{(2)}=a^{(2)}P^{(2)}$, $T_{AB}^{(1)}=a^{(1)}P^{(1)}$

Msg 2. $B\rightarrow A$:

$T_{BA}^{(1)}=b^{(1)}P^{(1)}$, $T_{BA}^{(2)}=b^{(2)}P^{(2)}$

그림 5. 2PAK-MPE 프로토콜

원곡선위의 덧셈군이고, $\mathbb{G}_2^{(i)}$는 위수 $q^{(i)}$인 유한체위의 곱셈군이다. $P^{(i)}$는 $\mathbb{G}_1^{(i)}$의 임의의 생성자이다. $\hat{e}^{(i)}$는 $\mathbb{G}_1^{(i)}$과 $\mathbb{G}_2^{(i)}$사이의 admissible bilinear map이다.

- PKG_i가 선택하는 암호학적 해쉬함수: $H_1^{(i)}$: $\{0,1\}^* \rightarrow \mathbb{G}_1^{(i)}$는 임의의 길이의 문자를 $\mathbb{G}_1^{(i)}$의 원소로 사상해주는 암호학적 해쉬함수이고, $H_2^{(i)}: \mathbb{G}_2^{(i)} \times \mathbb{G}_1^{(i)} \rightarrow \{0,1\}^k$는 $\mathbb{G}_2^{(i)}$의 원소와 $\mathbb{G}_1^{(i)}$의 원소를 입력값으로 받아 k길이의 문자로 사상해주는 암호학적 해쉬함수이다. 여기서 k는 프로토콜에서 사용되는 부분 세션키의 길이를 의미한다.
- PKG_i의 공개키, 마스터키 쌍: $\langle P_{pub}^{(i)}, s^{(i)} \rangle$. $s^{(i)} (\in \mathbb{Z}_{q^{(i)}}^*)$는 PKG_i의 마스터키로 자신의 공개키 및 모든 사용자의 개인키 발급에 사용된다. $P_{pub}^{(i)} = s^{(i)} P^{(i)}$는 PKG_i의 공개키이다.
- PKG_i의 전체 시스템 파라미터: $\langle q^{(i)}, \mathbb{G}_1^{(i)}, \mathbb{G}_2^{(i)}, P^{(i)}, P_{pub}^{(i)}, H_1^{(i)}, H_2^{(i)}, \hat{e}^{(i)} \rangle$.

3.2 다중 PKG 환경을 위한 2자간 키 동의 프로토콜 (The 2PAK-MPE Protocol)

이 논문에서 제안하는 다중 PKG 환경을 위한 2자간 키 동의 프로토콜을 2PAK-MPE (2-Party Authenticated Key agreement protocol for Multiple PKG Environment) 프로토콜이라 한다. 2PAK-MPE 프로토콜에 참여하는 사용자 A와 B는 각각 PKG_1과 PKG_2로부터 개인키를 발급받은 참여자이며, 각 참여자들은 프로토콜을 수행하기 전에 상대방의 ID와 PKG 환경에 대한 정보를 알고 있다고 가정한다. A와 B의 공개키와 개인키 쌍은 다음과 같다.

A의 공개키/개인키 쌍:
$(Q_A^{(1)} = H_1^{(1)}(ID_A), S_A^{(1)} = s^{(1)} Q_A^{(1)})$
B의 공개키/개인키 쌍:
$(Q_B^{(2)} = H_1^{(2)}(ID_B), S_B^{(2)} = s^{(2)} Q_B^{(2)})$

여기에서 ID_A와 ID_B는 A와 B의 잘 알려진 신원정보이다. 각 참여자들은 프로토콜을 시작하기 전에 각 PKG 환경을 위한 임시 개인키를 선택한다. 예를 들어 A는 $a^{(1)} \in \mathbb{Z}_{q^{(1)}}^*$과 $a^{(2)} \in \mathbb{Z}_{q^{(2)}}^*$를 선택한다. 이들 키는 프로토콜을 수행할 때마다 매번 다른 것을 선택한다. 참여자 A와 B는 메시지 1과 2를 교환한 후에 다음과 같은 부분 세션키를 계산한다.

A가 계산하는 부분 세션키:
$K_{AB}^{(1)} = \hat{e}^{(1)}(S_A^{(1)}, T_{BA}^{(1)})$, $K_{AB}^{(2)} = \hat{e}^{(2)}(Q_B^{(2)}, a^{(2)} P_{pub}^{(2)})$
B가 계산하는 부분 세션키:
$K_{BA}^{(1)} = \hat{e}^{(1)}(Q_A^{(1)}, b^{(1)} P_{pub}^{(1)})$, $K_{BA}^{(2)} = \hat{e}^{(2)}(S_B^{(2)}, T_{AB}^{(2)})$

두 참여자는 각자 계산한 부분 세션키들을 키 유도함수 $H_2^{(1)}$과 $H_2^{(2)}$를 이용해 세션키 생성에 필요한 값을 얻는다. 그런 후 일반적인 암호학적 해쉬함수 $H: \{0,1\}^{2k} \rightarrow \{0,1\}^l$를 이용해 최종 세션키를 생성한다. 이때 l은 최종 세션키의 길이가 된다.

A가 계산하는 최종 세션키:
$SK_{AB} = H(H_2^{(1)}(K_{AB}^{(1)}, a^{(1)} T_{BA}^{(1)}) \| H_2^{(2)}(K_{AB}^{(2)}, a^{(2)} T_{BA}^{(2)}))$
B가 계산하는 최종 세션키:
$SK_{BA} = H(H_2^{(1)}(K_{BA}^{(1)}, b^{(1)} T_{AB}^{(1)}) \| H_2^{(2)}(K_{BA}^{(2)}, b^{(2)} T_{AB}^{(2)}))$

여기서 '||'는 비트 간 결합을 의미한다. 최종 세션키를 생성할 때 abP를 사용하는 것은 Chen과 Kudla가 제안한 방법으로서, PKG 전방향 안전성을 보장하기 위한 수단이다. 제안하는 2PAK-MPE 프로토콜은 대칭적인(role symmetric) 키 동의 프로토콜이다. 즉, 모든 참여자는 동일한 연산들을 수행해 동일한 세션키를 생성하게 된다.

2AK-MPE 프로토콜에서 A와 B가 각각 계산하는 부분 세션키 $K_{AB}^{(1)}$와 $K_{BA}^{(1)}$ 그리고 $K_{AB}^{(2)}$와 $K_{BA}^{(2)}$가 같음을 다음과 같이 보임으로써 A가 계산한 최종 세션키 SK_{AB}와 B가 계산하는 최종 세션키 SK_{BA}가 같음을 보일 수 있다.

$$
\begin{aligned}
K_{AB}^{(1)} &= \hat{e}^{(1)}(S_A^{(1)}, T_{BA}^{(1)}) \\
&= \hat{e}^{(1)}(s^{(1)} Q_A^{(1)}, b^{(1)} P^{(1)}) \\
&= \hat{e}^{(1)}(Q_A^{(1)}, P^{(1)})^{s^{(1)} b^{(1)}} \\
&= \hat{e}^{(1)}(Q_A^{(1)}, b^{(1)} s^{(1)} P^{(1)}) \\
&= \hat{e}^{(1)}(Q_A^{(1)}, b^{(1)} P_{pub}^{(1)}) = K_{BA}^{(1)}
\end{aligned}
$$

$$
\begin{aligned}
K_{AB}^{(2)} &= \hat{e}^{(2)}(Q_B^{(2)}, a^{(2)}P_{pub}^{(2)}) \\
&= \hat{e}^{(2)}(Q_B^{(2)}, a^{(2)}s^{(2)}P^{(2)}) \\
&= \hat{e}^{(2)}(Q_B^{(2)}, P^{(2)})^{a^{(2)}s^{(2)}} \\
&= \hat{e}^{(2)}(s^{(2)}Q_B^{(2)}, a^{(2)}P^{(2)}) \\
&= \hat{e}^{(2)}(S_B^{(2)}, T_{AB}^{(2)}) = K_{BA}^{(2)}
\end{aligned}
$$

3.3 다중 PKG 환경을 위한 3자간 키 동의 프로토콜 (The 3PAK-MPE Protocol)

3PAK-MPE 프로토콜은 2PAK-MPE 프로토콜을 확장한 것으로 총 2라운드로 구성된다. 첫 번째 라운드 에서 각 사용자는 다른 사용자들과 2PAK-MPE 프로토콜을 수행해 공유키를 생성한다. 두 번째 라운드에서는 첫 번째 라운드에서 생성한 공유키를 이용해 구축된 안전하고 인증된 통신채널을 통해 사용자를 인증하고 세션키 생성에 사용되는 값들을 교환하여 최종 세션키를 생성하게 된다. 이 논문에서는 두 종류의 3PAK-MPE 프로토콜을 제안하는데, 하나는 최종 세션키를 생성할 때 pairing을 이용하며, 다른 하나는 최종 세션키 생성할 때 pairing을 이용하지 않는다. 앞으로 이들을 각각 3PAK-MPE-p와 3PAK-MPE-r 프로토콜이라 한다.

3PAK-MPE-p와 3PAK-MPE-r은 두 번째 라운드에서 전송하는 메시지와 세션키를 생성하는 방법은 다르지만 첫 번째 라운드는 동일하게 진행되며 두 프로토콜 모두 대칭적인 프로토콜이다. 3PAK-MPE 프로토콜은 다중 PKG 환경이라는 제한된 상황 때문에 앞서 제안한 2PAK-MPE 프로토콜을 그대로 적용할 수 없다. 즉 2PAK-MPE 프로토콜을 3자간 키 동의에 그대로 적용할 경우, 참여자 A와 B간의 키 동의 부분, A와 C간의 키 동의 부분, B와 C간의 키 동의 부분 이렇게 3부분의 키 동의 부분을 만든 후, 이 3부분을 결합해야 하는데 이때 A는 B와 C간의 키 동의 부분까지 알아야 하는 문제가 생긴다. B와 C간의 키 동의 부분을 생성하기 위해서는 B 또는 C의 개인키를 알아야 하기 때문에 A는 이 부분을 계산 할 수 없다(2.3 Pairing을 이용한 신원기반 키 동의 프로토콜 참조). 따라서 3PAK-MPE 프로토콜은 2PAK-MPE 프로토콜을 이용하여 각 사용자간에 안전하고 인증된 채널을 만든 후에 이 채널로 세션키 생성에 필요한 정보를 교환하는 방식을 사용한다.

3.3.1 3PAK-MPE 프로토콜의 첫 번째 라운드

3PAK-MPE 프로토콜은 PKG_3으로부터 개인키를 발급받은 사용자 C가 추가되었을 뿐 그 외에는 2PAK-MPE 프로토콜과 동일하다. 표기법은 2PAK-MPE 프로토콜에서 정의한 표기법을 그대로 따른다.

3PAK-MPE에서는 세 개의 다른 환경을 사용하므로 각 사용자는 각 환경을 위한 세 개의 임시 개인키를 프로토콜을 수행하기 전에 선택한다. 3PAK-MPE 프로토콜의 첫 번째 라운드는 그림 6과 같다. 첫 번째 라운드를 통해 각 사용자들은 다른 사용자와 2PAK-MPE 프로토콜을 수행하여 각 사용자간에 안전하고 인증된 채널을 구축한다. 각 참여자가 공유하게 되는 부분 공유키는 표 1과 같다.

표 1에서 생성된 부분 공유키를 이용해 표 2와 같은 공유키를 생성한다. 표 3에서 생성된 공유키는 2PAK-MPE 프로토콜을 이용한 것으로 (K_{AB}, K_{BA}), (K_{AC}, K_{CA}), (K_{BC}, K_{CB})가 각각 동일한 값이 된다. 각 개체들은 이 공유키를 이용해 안전하고 인증된 통신채널을 구축하게 된다.

3.3.2 Pairing을 이용하는 3PAK-MPE 프로토콜의 두 번째 라운드

Pairing을 사용하는 3PAK-MPE-p 프로토콜의 2번째 라운드는 사용자간 인증을 하는 단계이다. 메시지 4, 5, 6을 주고받은 후, 각 개체는 자신의 공유키로 그 값들을 해독하여 키 동의에 참여하는 다른 개체들로부터 수신한 값들을 확인한다. 수신한 값들이 올바른 값임을 확인한 후 세션키를 계산한

Msg 1. $A \rightarrow B, C$: $T_{AB}^{(2)} = a^{(2)}P^{(2)}$, $T_{AC}^{(3)} = a^{(3)}P^{(3)}$, $W_A^{(1)} = a^{(1)}P^{(1)}$

Msg 2. $B \rightarrow C, A$: $T_{BA}^{(1)} = b^{(1)}P^{(1)}$, $T_{BC}^{(3)} = b^{(3)}P^{(3)}$, $W_B^{(2)} = b^{(2)}P^{(2)}$

Msg 3. $C \rightarrow A, B$: $T_{CA}^{(1)} = c^{(1)}P^{(1)}$, $T_{CB}^{(2)} = c^{(2)}P^{(2)}$, $W_C^{(3)} = c^{(3)}P^{(3)}$

그림 6. 3PAK-MPE 프로토콜의 첫 번째 라운드

표 1. 3PAK-MPE 프로토콜의 첫 번째 라운드에서 각 사용자가 생성하는 부분 공유키

참여자	부분 공유키 생성 대상	부분 공유키
A	B	$K_{AB}^{(1)}=e^{(1)}(S_A^{(1)}, T_{BA}^{(1)})$, $K_{AB}^{(2)}=e^{(2)}(Q_B^{(2)}, a^{(2)}P_{pub}^{(2)})$
	C	$K_{AC}^{(1)}=e^{(1)}(S_A^{(1)}, T_{CA}^{(1)})$, $K_{AC}^{(3)}=e^{(3)}(Q_C^{(3)}, a^{(3)}P_{pub}^{(3)})$
B	A	$K_{BA}^{(2)}=e^{(2)}(S_B^{(2)}, T_{AB}^{(2)})$, $K_{BA}^{(1)}=e^{(1)}(Q_A^{(1)}, b^{(1)}P_{pub}^{(1)})$
	C	$K_{BC}^{(2)}=e^{(2)}(S_B^{(2)}, T_{CB}^{(2)})$, $K_{BC}^{(3)}=e^{(3)}(Q_C^{(3)}, b^{(3)}P_{pub}^{(3)})$
C	A	$K_{CA}^{(3)}=e^{(3)}(S_C^{(3)}, T_{AC}^{(3)})$, $K_{CA}^{(1)}=e^{(1)}(Q_A^{(1)}, c^{(1)}P_{pub}^{(1)})$
	B	$K_{CB}^{(3)}=e^{(3)}(S_C^{(3)}, T_{BC}^{(3)})$, $K_{CB}^{(2)}=e^{(2)}(Q_B^{(2)}, c^{(2)}P_{pub}^{(2)})$

표 2. 3PAK-MPE 프로토콜의 첫 번째 라운드에서 각 사용자가 생성하는 공유키

참여자	공유키 생성 대상	공유키
A	B	$K_{AB}=H(H_2^{(1)}(K_{AB}^{(1)}, a^{(1)}T_{BA}^{(1)}) \| H_2^{(2)}(K_{AB}^{(2)}, a^{(2)}W_B^{(2)}))$
	C	$K_{AC}=H(H_2^{(1)}(K_{AC}^{(1)}, a^{(1)}T_{CA}^{(1)}) \| H_2^{(3)}(K_{AC}^{(3)}, a^{(3)}W_C^{(3)}))$
B	A	$K_{BA}=H(H_2^{(2)}(K_{BA}^{(2)}, b^{(2)}T_{AB}^{(2)}) \| H_2^{(1)}(K_{BA}^{(1)}, b^{(1)}W_A^{(1)}))$
	C	$K_{BC}=H(H_2^{(2)}(K_{BC}^{(2)}, b^{(2)}T_{CB}^{(2)}) \| H_2^{(3)}(K_{BA}^{(3)}, b^{(3)}W_C^{(3)}))$
C	A	$K_{CA}=H(H_2^{(3)}(K_{CA}^{(3)}, c^{(3)}T_{AC}^{(3)}) \| H_2^{(1)}(K_{CA}^{(1)}, c^{(1)}W_A^{(1)}))$
	B	$K_{CB}=H(H_2^{(3)}(K_{CB}^{(3)}, c^{(3)}T_{BC}^{(3)}) \| H_2^{(2)}(K_{CA}^{(2)}, c^{(2)}W_B^{(2)}))$

Msg 4. $A \rightarrow B, C$: $\{H(T_{AB}^{(2)} \| T_{AC}^{(3)} \| W_A^{(1)} \| Q_B^{(2)})\}_{K_{AB}}$, $\{H(T_{AB}^{(2)} \| T_{AC}^{(3)} \| W_A^{(1)} \| Q_C^{(3)})\}_{K_{AC}}$

Msg 5. $B \rightarrow C, A$: $\{H(T_{BA}^{(1)} \| T_{BC}^{(3)} \| W_B^{(2)} \| Q_C^{(3)})\}_{K_{BC}}$, $\{H(T_{BA}^{(1)} \| T_{BC}^{(3)} \| W_B^{(2)} \| Q_A^{(1)})\}_{K_{BA}}$

Msg 6. $C \rightarrow A, B$: $\{H(T_{CA}^{(1)} \| T_{CB}^{(2)} \| W_C^{(3)} \| Q_A^{(1)})\}_{K_{CA}}$, $\{H(T_{CA}^{(1)} \| T_{CB}^{(2)} \| W_C^{(3)} \| Q_B^{(2)})\}_{K_{CB}}$

그림 7. 3PAK-MPE-p 프로토콜의 두 번째 라운드

다. 1라운드에서 올바른 공유키를 생성한 참여자들만 메시지 4, 5, 6을 생성하고 해독할 수 있으므로 사용자를 인증하는 효과를 얻을 수 있다. 세션키는 키 동의에 참여하는 참여자들의 개인키를 발급한 모든 PKG의 환경에서 각각 생성한 후 그 값을 하나로 합치는 방법으로 생성한다.

메시지 4, 5, 6를 수신한 사용자들은 메시지를 해독한 값과 자신이 1 라운드에서 받은 값들과 자신의 공개키를 해쉬한 값을 비교하여 동일한 경우 다음과 같은 부분 세션키와 최종 세션키를 계산한다. 각 개체는 3개의 부분 세션키를 결합하여 다음과 같은 최종 세션키를 계산하게 된다.

$$SK=H(H_2^{(1)}(K_{ABC}^{(1)}) \| H_2^{(2)}(K_{ABC}^{(2)}) \| H_2^{(3)}(K_{ABC}^{(3)}))$$

이 방법은 각 사용자 간 인증에 4번, 최종 세션키 생성에 3번, 총 7번의 pairng 연산을 요구한다. 각 사용자는 최종 세션키 생성을 위해 3번의 pairing 연산을 하게 되는데 실제로 서로 다른 세 개의 군 중 하나의 군에서만 한번의 pairing 연산을 하더라도 키의 안전성에는 차이가 없다. 하지만 참여자들이 이렇게 키를 생성하는 것에 동의하지 않을 수 있으므로 이렇게 각 군에서 중복하여 pairing 연산을 하는 것이다.

3.3.3 Pairing을 이용하지 않는 3PAK-MPE 프로토콜의 두 번째 라운드

Pairing을 이용하지 않는 3PAK-MPE-r 프로토콜의 두 번째 라운드는 3PAK-MPE 프로토콜의

표 3. 3PAK-MPE-p에서 각 참여자들이 생성하는 부분 세션키

참여자	PKG 환경	부분 세션키
A	PKG_1	$K_{ABC}^{(1)}=\hat{e}^{(1)}(b^{(1)}P^{(1)},c^{(1)}P^{(1)})^{a^{(1)}}=\hat{e}^{(1)}$
	PKG_2	$K_{ABC}^{(2)}=\hat{e}^{(2)}(b^{(2)}P^{(2)},c^{(2)}P^{(2)})^{a^{(2)}}=\hat{e}^{(2)}(P^{(2)},P^{(2)})^{a^{(2)}b^{(2)}c^{(2)}}$
	PKG_3	$K_{ABC}^{(3)}=\hat{e}^{(3)}(b^{(3)}P^{(3)},c^{(3)}P^{(3)})^{a^{(3)}}=\hat{e}^{(3)}(P^{(3)},P^{(3)})^{a^{(3)}b^{(3)}c^{(3)}}$
B	PKG_1	$K_{ABC}^{(1)}=\hat{e}^{(1)}(a^{(1)}P^{(1)},c^{(1)}P^{(1)})^{b^{(1)}}=\hat{e}^{(1)}(P^{(1)},P^{(1)})^{a^{(1)}b^{(1)}c^{(1)}}$
	PKG_2	$K_{ABC}^{(2)}=\hat{e}^{(2)}(a^{(2)}P^{(2)},c^{(2)}P^{(2)})^{b^{(2)}}=\hat{e}^{(2)}(P^{(2)},P^{(2)})^{a^{(2)}b^{(2)}c^{(2)}}$
	PKG_3	$K_{ABC}^{(3)}=\hat{e}^{(3)}(a^{(3)}P^{(3)},c^{(3)}P^{(3)})^{b^{(3)}}=\hat{e}^{(3)}(P^{(3)},P^{(3)})^{a^{(3)}b^{(3)}c^{(3)}}$
C	PKG_1	$K_{ABC}^{(1)}=\hat{e}^{(1)}(a^{(1)}P^{(1)},b^{(1)}P^{(1)})^{c^{(1)}}=\hat{e}^{(1)}(P^{(1)},P^{(1)})^{a^{(1)}b^{(1)}c^{(1)}}$
	PKG_2	$K_{ABC}^{(2)}=\hat{e}^{(2)}(a^{(2)}P^{(2)},b^{(2)}P^{(2)})^{c^{(2)}}=\hat{e}^{(2)}(P^{(2)},P^{(2)})^{a^{(2)}b^{(2)}c^{(2)}}$
	PKG_3	$K_{ABC}^{(3)}=\hat{e}^{(3)}(a^{(3)}P^{(3)},b^{(3)}P^{(3)})^{c^{(3)}}=\hat{e}^{(3)}(P^{(3)},P^{(3)})^{a^{(3)}b^{(3)}c^{(3)}}$

Msg 4. $A\rightarrow B, C$: $\{R_A\}_{K_{AB}}, \{R_A\}_{K_{AC}}$
Msg 5. $B\rightarrow C, A$: $\{R_B\}_{K_{BA}}, \{R_B\}_{K_{BC}}$
Msg 6. $C\rightarrow A, B$: $\{R_C\}_{K_{CA}}, \{R_C\}_{K_{CB}}$

그림 8. 3PAK-MPE-r 프로토콜의 두 번째 라운드

첫 번째 라운드에서 생성된 공유키를 이용해 세션키 생성에 필요한 값들을 교환한다. 3PAK-MPE-r 프로토콜의 두 번째 라운드는 그림 8과 같다.

메시지 4, 5, 6을 받은 각 사용자들은 자신의 생성한 공유키로 이 메시지들을 해독해 R_A, R_B, R_C를 이용해 다음과 같은 최종 세션키를 계산한다. 여기서 R_A, R_B, R_C는 특정 군에 속하는 원소가 아니라 임의의 비트값을 이용한다.

$$SK=H(R_A\|R_B\|R_C)$$

따라서, 추가적인 pairing 연산없이 사용자마다 네 번의 pairing 연산만을 수행하여 키 동의를 할 수 있다.

Ⅳ. 분 석

이 장에서는 제안된 프로토콜이 가지는 효율성과 안전성을 분석한다. 안전성은 제안하는 프로토콜이 키 동의 프로토콜의 보안 요구 사항을 만족하는가에 대해 휴리스틱하게 논의하며, 효율성은 기존 프로토콜들과 제안하는 프로토콜의 pairing 연산의 수를 비교하여 분석한다.

4.1 안전성 분석

제안하는 프로토콜의 안전성 분석은 2PAK-MPE 프로토콜에 대해서만 논의한다. 3PAK-MPE-p 프로토콜과 3PAK-MPE-r 프로토콜의 첫 번째 라운드에서는 2PAK-MPE 프로토콜을 사용하므로 이 두 프로토콜들의 안전성은 2PAK-MPE 프로토콜의 안전성에 기반을 두고 있다. 또한 3PAK-MPE-p 프로토콜의 경우 2PAK-MPE 프로토콜을 공격하여 두 번째 라운드에서 교환되는 값들을 얻을 수 있다고 하더라도 각 사용자들의 임시 개인키 $a^{(i)}, b^{(i)}, c^{(i)}$를 계산하는 것이 계산적으로 어렵다면 BDHP 문제에 의해 세션키를 계산하는 것 역시 계산적으로 어렵다. 하지만 3PAK-MPE-r 프로토콜의 경우에는 두 번째 라운드에서 교환되는 값을 얻을 수 있으면 공격자가 세션키를 계산할 수 있다. 결론적으로 2PAK-MPE 프로토콜이 안전하다면 3PAK-MPE 프로토콜 역시 안전하다고 할 수 있다.

보조정리 1. 2PAK-MPE 프로토콜은 정당한 참여자만이 부분 세션키를 계산 할 수 있다.

증명. 프로토콜에 참여하는 A와 B가 계산하게 되는 부분 세션키와 최종 세션키는 다음과 같으며, A가 계산하는 부분 세션키:

$K_{AB}^{(1)}=\hat{e}^{(1)}(Q_A^{(1)},P^{(1)})^{s^{(1)}b^{(1)}}$, $K_{AB}^{(2)}=\hat{e}^{(2)}(Q_B^{(2)},P^{(2)})^{a^{(2)}s^{(2)}}$

A가 계산하는 최종 세션키:

$SK_{AB}=H(H_2^{(1)}(K_{AB}^{(1)},a^{(1)}T_{BA}^{(1)})\|H_2^{(2)}(K_{AB}^{(2)},a^{(2)}T_{BA}^{(2)}))$

B가 계산하는 부분 세션키:

$K_{BA}^{(1)}=\hat{e}^{(1)}(Q_A^{(1)},P^{(1)})^{b^{(1)}s^{(1)}}$, $K_{BA}^{(2)}=\hat{e}^{(2)}(Q_B^{(2)},P^{(2)})^{a^{(1)}s^{(1)}}$

B가 계산하는 최종 세션키:

$SK_{BA}=H(H_2^{(1)}(K_{BA}^{(1)},b^{(1)}T_{AB}^{(1)})\|H_2^{(2)}(K_{BA}^{(2)},b^{(2)}T_{AB}^{(2)}))$

공개되어 있거나 통신 메시지를 도청하여 누구나 얻을 수 있는 정보는 다음과 같다.

$Q_A^{(1)}$, $s^{(1)}P^{(1)}$, $a^{(1)}P^{(1)}$, $a^{(2)}P^{(2)}$, $Q_B^{(2)}$, $s^{(2)}P^{(2)}$, $b^{(1)}P^{(1)}$, $b^{(2)}P^{(2)}$

이 프로토콜을 공격하는 공격자는 메시지를 변조하지 않고 공격하는 수동 공격자와 메시지를 변조하여 공격하는 능동 공격자로 나눌 수 있다. 먼저 수동 공격자는 크게 다음과 같은 네 가지 공격자로 나눌 수 있다.

첫째, 공개 정보 외에 다른 정보는 없는 공격자이다. 이 정보들을 이용해서는 부분 세션키를 직접 계산하는 것은 가능하지 않다. 부분 세션키 $K_{AB}^{(1)}$를 계산하기 위해서는 $b^{(1)}P^{(1)}$으로부터 참여자 B의 임시 개인키 $b^{(1)}$를 계산해 $\hat{e}^{(1)}(Q_A^{(1)},P_{pub}^{(1)})^{b^{(1)}}$를 계산하거나 $s^{(1)}P^{(1)}$로부터 $s^{(1)}$을 계산해 $K_{AB}^{(1)}=\hat{e}^{(1)}(Q_A^{(1)},b^{(1)}P^{(1)})^{s^{(1)}}$를 계산해야 한다. 하지만 두 경우 모두 $\mathbb{G}_1$에서 DLP이므로 가능하지 않다.

둘째, 공개 정보 외에 두 PKG의 마스터키를 알고 있는 공격자이다. 이 공격자는 공개 정보 외에 $s^{(1)}$를 추가적으로 알고 있으므로 $b^{(1)}$를 얻을 수 없어도 $K_{AB}^{(1)}=\hat{e}^{(1)}(Q_A^{(1)},b^{(1)}P^{(1)})^{s^{(1)}}$를 통해 부분 세션키 $K_{AB}^{(1)}$를 계산 할 수 있다. 그러나 최종 세션키 $SK_{AB}=H(H_2^{(1)}(K_{AB}^{(1)},a^{(1)}T_{BA}^{(1)})\|H_2^{(2)}(K_{AB}^{(2)},a^{(2)}T_{BA}^{(2)}))$를 계산하기 위해서는 부분 세션키 $K_{AB}^{(1)}$외에 $a^{(1)}T_{BA}^{(1)}=a^{(1)}b^{(1)}P^{(1)}$를 계산할 수 있어야 한다. 주어진 정보로부터 A의 임시 개인키 $a^{(1)}$를 계산하는 문제 역시 $\mathbb{G}_1$에서 DLP이므로 가능하지 않다.

셋째, 한 쪽의 임시 개인키를 알고 있는 공격자이다. 중간 공격자가 여기에 해당한다. B의 임시 개인키를 알고 있는 공격자는 $\hat{e}^{(1)}(Q_A^{(1)},P_{pub}^{(1)})^{b^{(1)}}$를 통해 $K_{AB}^{(1)}$는 계산할 수 있지만 $K_{AB}^{(2)}=\hat{e}^{(2)}(Q_B^{(2)},P^{(2)})^{a^{(2)}s^{(2)}}$를 계산하는 것은 가능하지 않다. 이것을 계산하기 위해서는 $a^{(2)}P^{(2)}$으로부터 $a^{(2)}$를 얻어 $\hat{e}^{(2)}(Q_B^{(2)},P_{pub}^{(2)})^{a^{(2)}}$를 계산하거나 $s^{(2)}P^{(2)}$로부터 $s^{(2)}$를 계산해 $K_{AB}^{(2)}=\hat{e}^{(2)}(Q_B^{(2)},b^{(2)}P^{(2)})^{s^{(2)}}$를 계산해야 한다. 하지만 두 경우 모두 $\mathbb{G}_1$에서 DLP이므로 가능하지 않다.

넷째, A의 개인키를 알고 있지만 B의 개인키는 모르며, A가 선택한 임시 개인키 $a^{(1)}$과 $a^{(2)}$는 모르지만 B가 선택한 임시 개인키 $b^{(1)}$과 $b^{(2)}$를 알고 있는 공격자이다. 이 공격자는 세 번째 공격자가 알고 있는 정보 외에 추가적으로 $s^{(1)}Q_A^{(1)}$를 알고 있는 공격자이다. 세 번째 공격자에 의해 알 수 있듯이 이 정보가 없어도 $K_{AB}^{(1)}$는 계산할 수 있다. 그러나 $s^{(1)}Q_A^{(1)}$는 $K_{AB}^{(2)}$를 계산하는데 전혀 도움이 되지 않으므로 세 번째 공격유형의 공격자와 마찬가지로 $K_{AB}^{(2)}$를 계산하는 것은 가능하지 않다.

능동 공격자는 전송되는 메시지를 변조하여 공격하는 공격자이다. 능동 공격자는 프로토콜에 정의된 메시지 규약을 따르지 않고 공격 성공을 위해 변조된 메시지를 전달하여 공격할 수 있다. 즉, 보통의 참여자들이 원래 전달해야 하는 aP형태의 값 대신에 aP_{pub} 또는 aQ_A 형태의 메시지를 전달하여 공격을 할 수 있다. 이것을 수신한 상대방은 이 값의 형태를 확인할 수 없어 수신한 값을 이용하여 키를 계산할 수밖에 없다. 그러나 참여자가 계산하는 부분 세션키 중 $K_{BA}^{(1)}=\hat{e}^{(1)}(Q_A^{(1)},b^{(1)}P_{pub}^{(1)})$는 상대방으로부터 수신한 값의 사용 없이 오직 자신만이 알고 있는 정보와 공개된 정보만을 이용하여 생성되므로 공격자가 원래와 다른 형태의 값을 전달하더라도 영향을 받지 않는다. 그런데 능동 공격자가 $K_{BA}^{(1)}$를 계산하기 위해서는 $b^{(1)}P^{(1)}$으로부터 $b^{(1)}$을 얻어 $\hat{e}^{(1)}(Q_A^{(1)},P_{pub}^{(1)})^{b^{(1)}}$를 계산하거나 $s^{(1)}P^{(1)}$으로부터 $s^{(1)}$을 얻어 $K_{AB}^{(1)}=\hat{e}^{(1)}(Q_A^{(1)},b^{(1)}P^{(1)})^{s^{(1)}}$를 계산해야 한다. 하지만 이것은 $\mathbb{G}_1$에서 DLP이므로 가능하지 않다. ■

위에서 증명한 보조정리 1을 이용하여 제안하는

프로토콜이 키 동의 프로토콜의 보안 요구사항을 만족함을 보이겠다.

- **중간자 공격**: 중간자 공격은 보조정리 1의 세 번째 형태에 해당한다. 즉, 중간자 M은 기존 메시지 1과 2를 가로챈 후에 A와 B에게 메시지 1′과 메시지 2′를 각각 전달하는 공격이다.

  Msg 1′. $M \rightarrow B$: $T_{MB}^{(2)} = m^{(2)}P^{(2)}, T_{MB}^{(1)} = m^{(1)}P^{(1)}$
  Msg 2′. $M \rightarrow A$: $T_{MA}^{(1)} = m^{(1)}P^{(1)}, T_{MA}^{(2)} = m^{(2)}P^{(2)}$

  하지만 보조정리 1에서 증명한 바와 같이 이렇게 공격을 하더라도 중간자는 이런 메시지를 통해 A와 B가 생성하는 부분 세션키를 생성할 수 없다. 메시지 1′, 2′를 수신한 후 생성하는 부분 세션키는 다음과 같다.

  A가 계산하는 부분 세션키:
  $K_{AM}^{(1)} = \hat{e}^{(1)}(S_A^{(1)}, T_{MB}^{(1)})$, $K_{AM}^{(2)} = \hat{e}^{(2)}(Q_B^{(2)}, a^{(2)}P_{pub}^{(2)})$

  B가 계산하는 부분 세션키:
  $K_{BM}^{(1)} = \hat{e}^{(1)}(Q_A^{(1)}, b^{(1)}P_{pub}^{(1)})$, $K_{BM}^{(2)} = \hat{e}^{(2)}(S_B^{(2)}, T_{MB}^{(2)})$

  공격자는 $K_{AM}^{(1)}$과 $K_{BM}^{(2)}$와 동일한 값을 가지는 $K_{MA}^{(1)} = \hat{e}^{(1)}(Q_A^{(1)}, m^{(1)}P_{pub}^{(1)})$, $K_{MB}^{(2)} = \hat{e}^{(2)}(Q_B^{(2)}, m^{(2)}P_{pub}^{(2)})$를 계산할 수 있지만 $K_{BM}^{(1)}$과 $K_{AM}^{(2)}$와 동일한 값을 가지는 $K_{MB}^{(1)}$, $K_{MA}^{(2)}$를 계산 할 수 없으므로 중간자 공격이 불가능하다.

- **알려진 키 안전성**: 2PAK-MPE 프로토콜은 매 세션마다 새로운 임시 개인키 $a^{(i)}$, $b^{(i)}$를 사용한다. 따라서 매 세션마다 유일하고 독립적인 세션키가 생성된다. 따라서 과거의 세션키가 노출된다 하더라도 다음에 생성될 세션키에는 영향을 미치지 않는다.
- **PKG 전방향 안전성**: 보조정리 1의 두 번째 공격에서 증명한 바와 같이 두 PKG의 마스터키를 알고 있더라도 부분 세션키를 생성할 수 없다. 따라서 이 프로토콜은 PKG 전방향 안전성을 제공한다.
- **키 노출 저항성**: 보조정리 1의 네 번째 공격에서 증명한 바와 같이 A의 개인키를 알고 있더라도 B의 개인키를 모르면 B로 위장하여 A와 키 동의를 할 수 없다.
- **미지의 키공유 저항성**: 보조정리 1에 의해 부분 세션키는 오직 정당한 참여자만이 계산할 수 있으므로 참여자를 생각하고 있는 사용자가 아닌 다른 사용자와 기틀 동의하도록 참여자를 속이는 것은 계산적으로 어렵다.
- **키 제어**: 두 개의 부분 세션키에는 두 참여자의 임시 개인키가 모두 포함되므로 어느 한 사용자가 세션키의 값을 미리 결정할 수 없다.

4.2 효율성 분석

이 절에서는 기존 신원기반 키 동의 프로토콜들과의 pairing 연산의 수를 비교하여 제안하는 프로토콜의 효율성을 분석한다. 이것은 pairing이 기타 다른 연산에 비해 상대적으로 비용이 많이 소요되는 연산이기 때문이다. 2PAK-MPE 프로토콜, 3PAK-MPE 프로토콜을 기존에 제안된 프로토콜들과 각각 비교한 후, 2PAK-MPE 프로토콜이 가지는 pairing 연산량이 다중 PKG 환경에서 최소임을 보일 것이고, 또한 다중 PKG 환경에서 3PAK-MPE 프로토콜의 효율성에 대해 논의한다.

4.2.1 2PAK-MPE 프로토콜의 효율성 분석

현재까지 제안된 신원기반 2자간 키 동의 프로토콜 중 최소한의 pairing 연산을 요구하는 프로토콜은 Chen과 Kudla가 제안한 1번의 pairing 연산을 요구하는 키 동의 프로토콜이다. 그러나 이 프로토콜은 단일 PKG 환경을 위한 프로토콜이다. 제안하는 2PAK-MPE는 2개의 PKG(PKG_1, PKG_2)가 존재하는 다중 PKG 환경이며, 각 PKG마다 사용하는 시스템 파라미터가 완전히 다르다. 따라서 각 PKG들은 서로 다른 군($\mathbb{G}_1^{(1)}$, $\mathbb{G}_1^{(2)}$)에서 각 종 연산을 수행한다. 이 중 한 군만을 사용하여 키 동의를 할 경우에는 최소한 한 번의 pairing 연산이 요구된다. 그러나 각 사용자는 자신의 군에서 동작하는 개인키/공개키 쌍을 가지고 있으므로 Chen과 Kudla 방식을 다중 PKG 환경에 그대로 적용할 수 없을 뿐만 아니라 가능하다고 할 경우에는 상대방이 이를 수용하지 못할 수 있다. 또한 각기 다른 군의 원소 두개를 이용할 수 있는 Weil pairing 기반 bilinear map이 존재하지 않는다. 따라서 각 군마다 pairing 연산을 1번씩 총 2번의 pairing

표 4. 2PAK-MPE 프로토콜의 연산량 비교

프로토콜	pairing 연산 수	메시지 수	메시지 크기	환경
Smart	2/개체	1/개체	1 타원곡선점/메시지	단일 PKG 환경
Chen과 Kudla 1	1/개체	1/개체	2 타원곡선점/메시지	단일 PKG 환경
Chen과 Kudla 2	2/개체	1/개체	1 타원곡선점/메시지	다중 PKG 환경*
McCullagh와 Barreto	1/개체	1/개체	1 타원곡선점/메시지	다중 PKG 환경*
2PAK-MPE	**2/개체**	**1/개체**	**2 타원곡선점/메시지**	**다중 PKG 환경****

Smart 프로토콜을 제외한 모든 프로토콜은 PKG 전방향 안정성을 충족한다.
McCullagh와 Barreto 프로토콜은 Tate pairing을 사용하며, 나머지는 모두 Weil pairing을 사용한다.
*: 시스템 파라미터를 공유하는 다중 PKG 환경 (PKG의 마스터키는 다름)
**: 모든 시스템 파라미터를 공유하지 않는 다중 PKG 환경

연산을 하는 경우가 Weil pairing을 사용할 경우에는 다중 PKG 환경에서 요구되는 최소의 연산이라 할 수 있다. 이 논문에서 제안하는 2PAK-MPE 프로토콜의 경우 2번의 pairing 연산을 요구하므로 최소의 연산을 사용한다고 할 수 있다. 또한 각 군에서 pairing 연산을 하기 위해서는 각 군의 원소를 하나씩 교환해야 한다. 표 4는 기존 프로토콜과 제안된 2PAK-MPE 프로토콜과의 비교를 나타내고 있다. 이 표에서 알 수 있듯이 pairing 연산 수, 메시지 수, 메시지 크기 측면에서 완전한 다중 PKG 환경이라는 것을 고려하였을 때 최적이다.

4.2.2 3PAK-MPE 프로토콜의 효율성 분석

3PAK-MPE 프로토콜은 기존에 제안된 다른 신원기반 3자간 키 동의 프로토콜과는 다른 방법을 사용하여 중간자 공격을 방어한다. 기존의 프로토콜들은 서명을 이용해 중간자 공격을 방어하고 있지만 3PAK-MPE 프로토콜은 각 참여자 간에 안전한 채널을 구축하여 값을 교환하여 중간자 공격을 방어한다.

신원기반 공개키 암호시스템을 사용하여 교환되는 값을 서명하여 교환할 경우에는 다른 두 참여자의 서명을 확인하기 위해 각각 두 번, 총 네 번의 pairing 연산이 요구된다. 이 때 한 가지 주목해야 하는 것은 단일 PKG 환경에서는 Zhang 등의 프로토콜처럼 각 값의 서명을 별도의 식을 통해 확인하지 않고 하나의 식에서 확인할 수 있다. 뿐만 아니라 Shim의 프로토콜처럼 bilinear map의 특성을 이용하여 pairing 연산의 수를 줄이는 대신에 타원곡선 군의 덧셈 연산의 수를 늘릴 수 있다. 하지만 다중 PKG 환경에서는 Shim의 프로토콜과 같은 방법을 사용할 수 없으며, 각 서명을 별도로 확인해야 한다. 따라서 다중 PKG 환경에서는 서명을 이용할 경우에는 서명을 확인하기 위해 최소 네 번의 연산이 요구된다. 여기에 $\hat{e}(P,P)^{xyz}$와 같은 형태의 값을 이용하여 세션키를 사용하기 위해서는 추가적으로 하나의 pairing 연산이 요구되며, 3PAK-MPE-p처럼 다른 세 가지 군에서 모두 $\hat{e}(P,P)^{xyz}$와 같은 형태의 값을 만들 경우에는 추가적으로 세 개의 pairing 연산이 요구된다. 이런 추가적인 pairing 연산을 줄이기 위해 3PAK-MPE -r과 같이 세션키를 생성할 때 pairing 연산을 사용하지 않는 방법을 생각할 수 있으나 3PAK- MPE-r과 같은 방법으로 세션키를 생성하기 위해서는 교환되는 값의 비밀성까지 요구되므로 가능하지 않다. 3PAK-MPE-r 프로토콜은 요구되는 pairing 연산 수가 서명을 사용하는 방식보다 적으며, 단일 PKG 환경과 비교하였을 때 하나의 pairing 연산만이 추가로 요구된다. 따라서 3자간의 경우 세션키를 계산하는 방법을 제한하여 중간자 공격을 방어할 수 없다는 것을 고려할 때 3PAK-MPE-r은 매우 효율적이다. 표 5는 Zhang, Shim 등의 프로토콜들과 제안된 3PAK-MPE 프로토콜들과의 비교를 나타내고 있다.

V. 결 론

이 논문에서 다중 PKG 현경을 고려한 신원기반 2자간 키 동의 프로토콜과 이를 확장한 두 가지 버전의 신원기반 3자간 키 동의 프로토콜을 제안하였다. 이는 모든 시스템 파라미터를 독립적으로 사용하는 다중 PKG 환경을 고려한 최초의 시도이다. 또한, 제안된 프로토콜들은 PKG 전방향 안전성을

표 5. 3PAK-MPE 프로토콜의 연산량 비교

프로토콜	pairing 연산 수	방송 메시지의 수	메시지의 크기	중간자 공격 방어방법	환경
Zhamg등의 프로토콜	5/개체	1/개체	2 타원곡선점/메시지	서명	단일 PKG 환경
Shim	3/개체	1/개체	2 타원곡선점/메시지	서명	단일 PKG 환경
3PAK-MPE-p	**7/개체**	**2/개체**	3 타원곡선점/메시지 2 암호문/메시지	**암호화**	**다중 PKG 환경**
3PAK-MPE-r	**4/개체**	**2/개체**	3 타원곡선점/메시지 2 암호문/메시지	**암호화**	**다중 PKG 환경**

포함한 키 동의 프로토콜의 모든 보안 요구사항을 만족한다. 제안하는 키 동의 프로토콜들의 안전성은 타원곡선에서의 DLP, CDHP, BDHP에 기반을 두고 있다. 제안된 2자간 키 동의 프로토콜인 2PAK-MPE는 pairing 연산 수, 메시지 수, 메시지 크기 측면에서 다중 PKG 환경이라는 것을 고려하였을 때 최적이다. 또한 제안된 3자간 키 동의 프로토콜인 3PAK-MPE-r은 pairing 연산 수 측면에서 다중 PKG 환경을 고려하였을 때 역시 최적이다.

참 고 문 헌

[1] W. Diffie and M. Hellman, "New Direction in Cryptography," *In IEEE Transaction of Information Theory*, Volume 22, pp. 664-654, 1976.

[2] A. Menezes, M. Qu, and S. Vanstone, "Some New Key Agreement Protocols Providing Mutual Implicit Authentications," *2nd Workshop on Selected Areas in Cryptography (SAC '95)*, pp. 22-32, 1995.

[3] 이성운, 유기영, "간단하고 효율적인 상호 인증 키 동의 프로토콜", *한국정보보호학회논문지* 제13권 제1호, pp. 105-113, 2003.

[4] A. Shamir, "Identity-based Cryptosystems and Signature Scheme," *CRYPTO '84*, LNCS Volume 196, pp. 47-53, Springer-Verlag, 1984.

[5] G. Günther, "An Identity-based Key exchange Protocol," *EUROCRYPT '89*, LNCS Volume 434, pp. 29-37, Springer-Verlag, 1990.

[6] H. Sakazaki, E. Okamoto, and M. Mambo, "Constructing Identity-based Key Distribution Systems over Elliptic Curves," *IEICE TRANS. Fundamentals*, Volume E81-A, pp. 2138-2143, 1998.

[7] 박영호, 박호상, 정수환, "ECDSA를 적용한 ID 기반의 사용자 인증 및 키 교환 프로토콜", *한국정보보호학회논문지* 제12권 제1호, pp. 3-10, 2002.

[8] D. Boneh and M. Franklin, "Identity-based Encryption from Weil Pairing," *CRYPTO '01*, LNCS Volume 2139, pp. 213-229, Springer-Verlag, 2001.

[9] 김동현, 김상진, 오희국, 구본석, 유권호, "동적 ID 정보가 포함된 신원기반 암호시스템에서 효율적인 키 재발급 모델", *한국정보보호학회논문지* 제15권 제2호, pp. 23-37, 2005.

[10] N. Smart, "An Identity-based Authenticated Key Agreement Protocol Based on Weil Pairing," *In Electronic Letters*, Volume 38, pp.630-632, 2002.

[11] L. Chen and C. Kudla, "Identity-based Authenticated Key Agreement Protocols from Pairings," *In Proceedings of the 16th IEEE Computer Security Foundations Workshop*, pp. 219-233, IEEE Computer Society Press, 2003.

[12] L. Chen and C. Kudla, "Identity-based Authenticated Key Agreement Protocols from Pairings," *Cryptology ePrint Archive*, Reoprt 2002/184, 2004.

[13] N. McCullagh, and P. Barreto, "A New Two-Party Identity-Based Authenticated Key Agreement," *Cryptology ePrint Archive*, Report 2004/122, 2004.

[14] A. Joux, "A One Round Protocol for Tripartite Diffie-Hellman," *ANTS-IV*, LNCS Volume 1838, pp. 385-394, Spinger-Verlag, 2000.

[15] S. S. Al-Riyami and K. G. Patterson, "Tripartite Authenticated Key Agreement Protocols from Pairings," *In IMA Conference on Cryptography and Coding*, LNCS Volume 2898, pp. 332-359, Spinger-Verlag, 2003.

[16] F. Zhang, S. Liu, K. Kim, "ID-Based One Round Authenticated Tripartite Key Agreement Protocols with Pairings," *Crypology ePrint Archive*, Report 2002/122, 2002.

[17] D. Nalla and K. Reddy, "ID-Based tripartite Authenticated Key Agreement Protocols from Pairings," *Cryptology ePrint Archive*, Report 2003/04, 2003.

[18] Z. Chen, "Security Analysis on Nalla-Reddy's ID-Based Tripartite Key Agreement Protocols," *Cryptology ePrint Archive*, Report 2003/103, 2003.

[19] K. Shim, "Cryptanalysis of ID-based Tripartite Authenticated Key Agreement Protocols," *Cryptology ePrint Archive*, Report 2003/115, 2003.

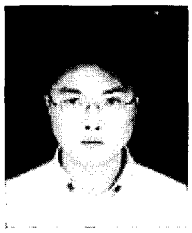
[20] 박영호, 이경현, "효율성을 개선한 신원기반의 3자간 복수 키 합의 프로토콜", *한국정보보호학회논문지* 제15권 제3호, pp. 77-89, 2005.

[21] A. Menezes, T. Okamoto, and S. Vanstone, "Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field," *In IEEE Transaction of Information Theory*, Volume 39, pp. 1639-1646, 1993.

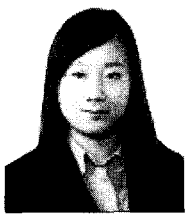
[22] G. Frey and H. Ruck, "A Remark Concerning m-divisibility and The Discrete Logarithm in the Divisor class Group of Curves," *Mathematics of Computation*, Volume 62, pp. 865-874, 1994.

[23] R. Dutta, R. Barua, P. Sarkar, "Pairing-Based Cryptography : A Survey," *Cryptology ePrint Archive*, Report 2004/064, 2004.

〈著者紹介〉



이 훈 정 (Hoonjung Lee) 학생회원
2003년 2월: 단국대학교 전자 · 컴퓨터학부(학사)
2005년 8월: 한양대학교 컴퓨터공학과(석사)
〈관심분야〉 키 관리, 신원기반 암호기법



김 현 숙 (Hyunsook Kim)
2001년 2월: 한양대학교 전자컴퓨터공학부(학사)
2003년 2월: 한양대학교 컴퓨터공학과(석사)
2004년 7월~현재: 충청북도청원교육청
〈관심분야〉 암호기술 응용

김 상 진 (Sangjin Kim) 종신회원
1995년 2월: 한양대학교 전자계산학과(학사)
1997년 2월: 한양대학교 전자계산학과(석사)
2002년 8월: 한양대학교 전자계산학과(박사)
2003년 3월~현재: 한국기술교육대학교 인터넷미디어공학부 조교수
〈관심분야〉 암호기술 응용
URL:http://infosec.kut.ac.kr/sangjin/

오 희 국 (Heekuck Oh) 종신회원
1983년: 한양대학교 전자공학과(학사)
1989년: 아이오와주립대학 전자계산학과(석사)
1992년: 아이오와주립대학 전자계산학과(박사)
1993년~1994년: 한국전자통신연구원 선임연구원
1995년 3월~현재: 한양대학교 컴퓨터공학과 부교수
〈관심분야〉 암호프로토콜, 네트워크 보안
URL:http://infosec.hanyang.ac.kr/~hkoh/