Journal of The Korea Institute of Information Security & Cryptology(JKIISC) ISSN 1598-3986(Print)
VOL.23, NO.3, June 2013 ISSN 2288-2715(Online)
http://dx.doi.org/10.13089/JKIISC.2013.23.3.351

Pointcheval-Zimmer 다중 인증 요소 기반 인증된 키 교환 프로토콜의 안전성 연구*

변 진 욱†‡
평택대학교 정보통신학과

On the Security of Pointcheval-Zimmer Multi-Factor Authenticated Key Exchange Protocol*

Jin Wook Byun†‡
Pyeongtaek University, Department of Information and Communication

요 약

Pointcheval과 Zimmer는 2008년에, 사용자의 비밀 키, 패스워드, 생체정보를 모두 이용한 다중 인증 요소 기반 키 교환 프로토콜을 제안하였다. 하지만, Hao와 Clarke는, 공격자가 한 개의 인증요소 (패스워드) 를 알고 있다고 가정 했을 때 이를 이용해서 나머지 두 개의 인증요소인 사용자의 비밀 키와 생체정보를 모두 알 수 있는 안전성 결함이 있음을 보였다. 하지만, Hao와 Clarke은 프로토콜 설계의 구조적인 결함 및 복잡성을 이유로 그에 대한 해결책을 제안하지 못하고 남겨 두었다. 본 논문에서는 Hao와 Clarke이 제시한 공격을 효율적으로 방어할 수 있는 대응방안을 제시하고 안전성을 논의한다.

ABSTRACT

In 2008, Pointcheval and Zimmer have presented multi-factor authenticated key exchange protocol with client's secret key, password, biometrics. However, it has been found to be flawed by Hao and Clarke if an attacker has single authentication factor (password), then the attacker can deduce other authentication factors. Interestingly, its countermeasure has not been presented due to the difficulty of design and structural problem. In this paper, an efficient countermeasure is briefly presented and its security is discussed as well.

Keywords: multi-factor authenticated key exchange, authentication, biometrics, password

I. 서 론

패스워드는 암기의 편리함과 구현의 용이함으로 인해 사용자 인증에 가장 보편적으로 사용되는 인증 요소이다. 인증과 키 교환을 동시에 제공하기 위하여 패스워드 기반 키 교환 연구도, 그룹, 다른 영역 환경 등에서 많은 연구가 진행되었다 [1,2,3,4,8,9]. 하지만, 다중 요소 기반 인증된 키 교환 프로토콜은 세 개의 인증 요소 (패스워드, 비밀 키, 생체 정보) 중 두 가지 이상을 결합하여 인증과 기 교환을 동시에 유도하는 프로토콜이다. 두 가지 이상의 인증요소를 적용하여 프로토콜을 설계한다는 측면에서, 다중 인증 요소 기반 키 교환 프로토콜은, 현재 IT 기술의 핵심 트렌드 중에 하나인 융합 보안의 핵심 가치와도 연결 되므로 그 연구가 중요함은 자명하다.

접수일(2013년 1월 8일), 수정일(2013년 3월 19일), 게재확정일(2013년 4월 3일)
* 이 논문은 2012학년도 평택대학교 학술연구비의 지원에 의하여 연구되었음.
† 주저자, jwbyun@ptu.ac.kr
‡ 교신저자, jwbyun@ptu.ac.kr(Corresponding author)

다중 요소 기반 인증된 키 교환 기법은 현재까지 모두 대칭적인 기반 기술로 설계되어져왔다. 즉, PKI 도움 (서명 구조) 없이, 패스워드, 비밀 키, 생체정보 중 두 가지 이상을 결합하여 인증 및 키 교환이 이루어진다. 크게 두 가지의 연구 흐름이 있다. 첫째는 패스워드와 비밀 키를 병합해서 키 교환을 유도하는 방법이고, 두 번째는 세 가지 인증요소를 모두 적용하는 방법이다. 패스워드와 비밀 키를 병합한 기법은, 1981년도에 Lamport가 처음으로 스마트카드를 이용해서 사용자를 인증할 수 있는 서비스를 제안한 이후로 활발히 연구가 진행되고 있다 [5]. 하지만, 세 가지 인증 요소를 적용한 키 교환 프로토콜은 안전성 증가라는 장점을 지닙에도 불구하고 아직 연구가 활발히 진행되지 않고 있다.

Pointcheval과 Zimmer는, 2008년에, 사용자의 비밀 키, 패스워드, 생체정보, 세 가지 인증 요소를 이용한 증명 가능한 다중 인증 요소 기반 키 교환 프로토콜을 처음 제안하였다 [7]. 하지만, 이 프로토콜은 Hao와 Clarke에 의해 구조적 약점이 존재함이 밝혀졌다 [6]. 즉, 공격자가 한 개의 인증요소 (패스워드)를 알고 있다고 가정 했을 때 이를 이용해서 나머지 두 개의 인증요소인 사용자의 비밀 키와 생체정보를 모두 알 수 있는 공격이 가능하였다. 하지만, Hao와 Clarke은 프로토콜 설계의 구조적인 결함 및 복잡성을 이유로 해결책을 제안하지 못하고 남겨 두었다.

본 논문에서는 Hao와 Clarke이 제시한 공격을 효율적으로 방어할 수 있는 대응방안을 연구한다. 다시 말해, 참고문헌 [7]의 프로토콜을 전반적으로 수정하지 않고 프로토콜의 첫 번째, 두 번째 전달 메시지의 구성만을 변경함으로써 궁극적으로 공격에 대한 해결책이 됨을 보인다. 대응방안의 핵심은 서버의 공개키 개인키를 두어 상호 키 요소를 만들 때 적용시키는 것이다. 또한 제안된 방법이 Hao와 Clarke이 제시한 공격에 안전함을 보인다.

II. Pointcheval과 Zimmer의 프로토콜

본 장에서는 Pointcheval과 Zimmer가 제안했던 프로토콜을 기술한다. 우선, 프로토콜에 사용되는 표기를 정리하면 다음과 같다.

2.1 표기

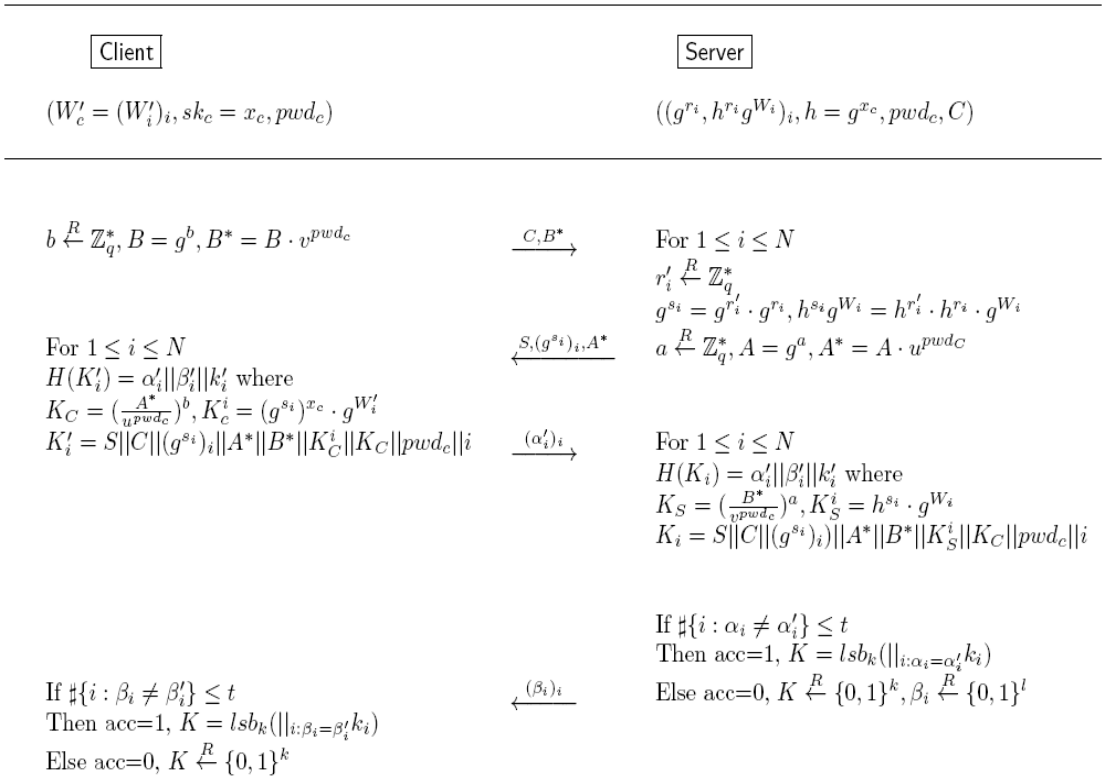
사용자는 $t_c=(W_c', sk_c=x_c, pwd_c)$를 보유하고 있으며, W_c'은 생체정보, sk_c는 비밀 키, pwd_c는 사용자의 패스워드이다. 서버는 사용자의 pwd_c를 보유하고 있고, 생체정보는 Elgamal 공개키 암호로 보호되어 저장된다. 생체정보 암호화는 등록단계에서 이루어지며 이러한 생체정보 탬플릿을 $W_c=(W_i)_i$이라 표기한다. 단, W_i는 W_c의 i번째 비트를 의미하고, N은 생체정보의 총 비트수이다. $(W_i)_i$ 표기는 W_i가 연속적으로 총 N비트 이하로 구성됨을 나타낸다. 예를 들어, $N=5$라면, $W_C=10110$이 될 수 있으며, 이를 위치별 비트 값을 표기하기 위해 $(W_i)_i$로 정의하였다. 본 논문에서는 공통의 파라미터 (u,v,p,g,q)를 가정한다. g는 Z_p^*내에서 위수가 q인 원소이고 서브그룹 G를 형성한다. 이후, mod p에 대한 표기는 생략한다. u,v는 서브 그룹 G의 랜덤 원소들이다. H는 랜덤 오라클로 간주한다. Elgamal로 암호화되어진 생체정보는 $(g^{r_i}, h^{r_i}\cdot g^{W_i})_i$이다. r_i는 비트별로 생성되는 랜덤 값이며, $h=g^{x_c}$는 사용자의 공개키이다. lsb(·) 함수는 참고문헌 [7]에 정확히 정의되지 않았다. 그 약자를 least significant bit이라고 유추했을 때, 키 원소를 입력으로 받아 최종 키를 유도하는 함수라 간주한다. 참고로, lsb(·) 함수의 정의는 본 논문에서 다루어지는 공격과 관계가 없다.

2.2 프로토콜 설명

- **[1 단계]** 사용자는 먼저 랜덤하게 원소 b를 선택하여 $B=g^b$를 계산하고 $B^*=B\cdot v^{pwd_c}$를 만들어 자신의 ID와 함께 서버에게 보낸다.
- **[2 단계]** 메시지를 받은 후, 암호화된 생체정보 $(g^{r_i}, h^{r_i}\cdot g^{W_i})$에서 g^{r_i}를 가져온다. 그리고, 서버는 랜덤 원소 r_i'을 선택해서 $g^{r_i}\cdot g^{r_i'}$를 만든다. s_i를 r_i+r_i'라 했을 때, 암호화된 생체정보 $h^{r_i}\cdot g^{W_i}$로부터 $h^{r_i'}$를 곱하여, $h^{r_i'}\cdot h^{r_i}\cdot g^{W_i}=h^{s_i}g^{W_i}$를 계산한다. 그리고 랜덤하게 원소 a를 선택하여 $A=g^a$를 계산하고 $A^*=A\cdot u^{pwd_c}$를 만들어 ID, $(g^{s_i})_i$, A^*를 모두 사용자에게 전달한다. $(g^{s_i})_i$는 $1\le i\le N$에 대한 i번째 비트 값들을 의미한다.
- **[3 단계]** 사용자는 서버로부터 받은 $(g^{s_i})_i$를 이용해서 K_i'를 다음과 같이 계산한다.

$$K_i'=S\|C\|(g^{s_i})_i\|A^*\|B^*\|K_c'\|K_c\|pwd_c\|i$$

$$K_c=\left(\frac{A^*}{v^{pwd_c}}\right)^b,\ K_c'=(g^{s_i})^{x_c}\cdot g^{W_i'}$$



〔그림 1〕 Pointcheval-Zimmer 프로토콜 〔7〕

이 값을 해시시켜서 $H(K_i') = \alpha_i' \| \beta_i' \| k_i'$를 계산하고 $(\alpha_i')_i$값을 서버에게 전달한다. $(\alpha^{s_i})_i$는 $1 \le i \le N$에 대한 i번째 비트 값들을 의미한다.

- **〔4 단계〕** 서버는 모든 N에 대해 사용자로부터 받은 $(\alpha_i')_i$값을 자신이 계산한 동일한 K_i'값을 이용해서 해시시켜, $H(K_i') = \alpha_i' \| \beta_i' \| k_i'$, 유도한 $(\alpha_i)_i$값과 동일한지 체크한다. 만약, 비교 시 비트별로 틀린 비트 수가 t개 이하이면 올바른 사용자라 간주하고 k_i를 lsb(·)유도함수에 대입하여 키 K를 계산한다. 그리고 $(\beta_i')_i$값을 사용자에게 전달한다. 만약, t개 이하가 아니라면, 즉, 올바른 사용자가 아니라면, 랜덤하게 키 K와 $(\beta_i')_i$값을 선택해서 사용자에게 전달한다.
- **〔5 단계〕** 사용자 역시 서버로부터 받은 $(\beta_i')_i$값을 자신의 K_i'값을 이용해서 해시시켜, $H(K_i') = \alpha_i' \| \beta_i' \| k_i'$, 유도한 $(\beta_i)_i$값과 체크해서 t개 이상 틀리지 않으면 올바른 서버라고 간주하고 키 K를 계산한다. 그렇지 않다면, 프로토콜의 인증은 실패하게 된다.

III. Hao와 Clarke이 제시한 공격 분석

Hao와 Clarke는 두 개의 공격이 가능함을 보였다. 첫째는 사용자의 패스워드를 알고 있는 공격자는 그 정보를 이용해 사용자의 생체정보를 유도할 수 있음을 보였고, 둘째는 사용자의 패스워드와 생체정보를 이용해서 사용자의 비밀 값 x_c를 유도할 수 있음을 보였다. 다음은 그 두 가지 공격 방법들이다.

3.1 생체정보 유도 방법

공격자 A가 사용자의 패스워드를 미리 알고 있다고 가정했을 때 A는 서버가 프로토콜 〔2 단계〕에서 보내는 $S, (g^{s_i})_i, A^*$를 자신이 선택한 정보로 위조해서 사용자에게 보낼 수 있다. 즉, 랜덤한 원소 s_i를 임의로 선

- **〔1 단계〕** 공격자는 s_i, a값과 pwd_c값을 알고 있기 때문에, K_c, K_c'값을 계산해서, 1부터 N까지 K_i'값과 그 해시 값을 다음처럼 계산한다.

$$H(K_i') = \alpha_i'' \| \beta_i' \| k_i'$$
$$K_i' = S \| C \| (g^{s_i})_i \| A^* \| B^* \| K_c' \| K_c \| pwd_c \| i$$
$$K_c = (\frac{B^*}{v^{pwd_c}})^a, K_c' = (g^{x_c})^{s_i}$$

- **〔2 단계〕** 원래 K_c'의 값은 $K_c' = (g^{s_i})^{x_c} \cdot g^{W_i'}$로 계산되어야 하나 W_i'값이 생체정보의 비트 값이므로 0혹은 1이라는 제한된 값을 지닌다. 그러므로, $W_i' = 0$이면, $K_c' = (g^{x_c})^{s_i}$를 지니고 $W_i' = 1$이면, $K_c' = (g^{x_c})^{s_i} \cdot g$값을 지닌다. 결론적으로, $1 \le i \le N$에 대해 다음의 비교식이 성립하게 된다.

$$\text{if } \alpha_i'' = \alpha_i' \; then \; W_i'' = 0$$
$$else \; W_i'' = 1$$

여기서, α_i''값은 공격자 A가 직접 $K_c, K_i', H(K_i')$값을 통해 계산된 값이며, α_i'값은 실제 사용자가 서버에게 전달하는 값으로서 공격자가 α_i''값과 비교하려는 값이다.

〔그림 2〕 생체정보 유도 알고리즘

택해서 $(g^{s_i})_i$를 계산한다. 그리고 a값을 랜덤하게 선택해서 $A^* = g^a \cdot u^{pwd_c}$를 스스로 계산해서 사용자에게 보낸다. 사용자는 이에 대해서 검증하지 않고 $(\alpha_i')_i$를 계산해서 공격자에게 주게 된다. $(\alpha_i')_i$값은 사용자의 생체정보를 알 수 있는 중요한 비교 대상 값이 되는데 그 유도 알고리즘은 다음 〔그림 2〕와 같다.

3.2 비밀 값 유도 방법

공격자 A가 사용자의 패스워드를 미리 알고 있다고 가정 했을때, 3.1절의 방법을 통해 생체정보를 획득할 수 있으며, 이후 사용자의 비밀 값 유도도 가능하다. 먼저, 공격자 A가 g^{s_i}를 사용자에게 전달하지 않고 G상의 작은 그룹(small group) 원소를 선택해서 전달한다. 작은 그룹을 G_s라 했을 때, b_1을 G_s의 생성자라 하자. 프로토콜 〔2 단계〕에서 b_1값을 g^{s_i}값 대신 보내면, 사용자는 프로토콜 절차대로 K_c'값을 다음과 같이 계산하게 된다.

$$K_c' = (g^{x_c})^{s_i} \cdot g^{W_i'} = (b_1)^{x_c} \cdot g^{W_i'}$$

공격자 A는 $K_i' (= S \| C \| (g^{s_i})_i \| A^* \| B^* \| K_c' \| K_c \| pwd_c \| i)$의 연결되어 있는 메시지 중 K_c'을 제외하고는 모두 알 수 있다. 그런데, 프로토콜 〔3 단계〕에서 정당한 사용자로부터 받은 α_i'값은 높은 확률로 i번째 비트 값이 동일하다. 만약 W_i값이 1이면 g값을 나누어서 $b_1^{x_c}$값을 구할 수 있고, 0이면 직접 $b_1^{x_c}$값을 구할 수 있다. 이러한 $b_1^{x_c}$값은 작은 그룹에 속하기 때문에 전수조사를 통해 $a = b_1^{x_c}$값을 직접 찾을 수 있다. 즉, 공격자는 $b_1^{x_c}$값을 전수 조사하여 K_c'값을 먼저 구하고 나머지 값들을 이용해서 α_i'값을 계산한다. 이 값을 정당한 사용자로부터 받은 α_i'값과 비교해서 궁극적인 $b_1^{x_c}$값을 구한다. 이러한 작업을 b_2, b_3에 대해 여러 번 반복해서 구한 다음 중국인의 나머지 정리를 통해 x_c값을 찾게 된다.

IV. 효율적인 해결책 제시

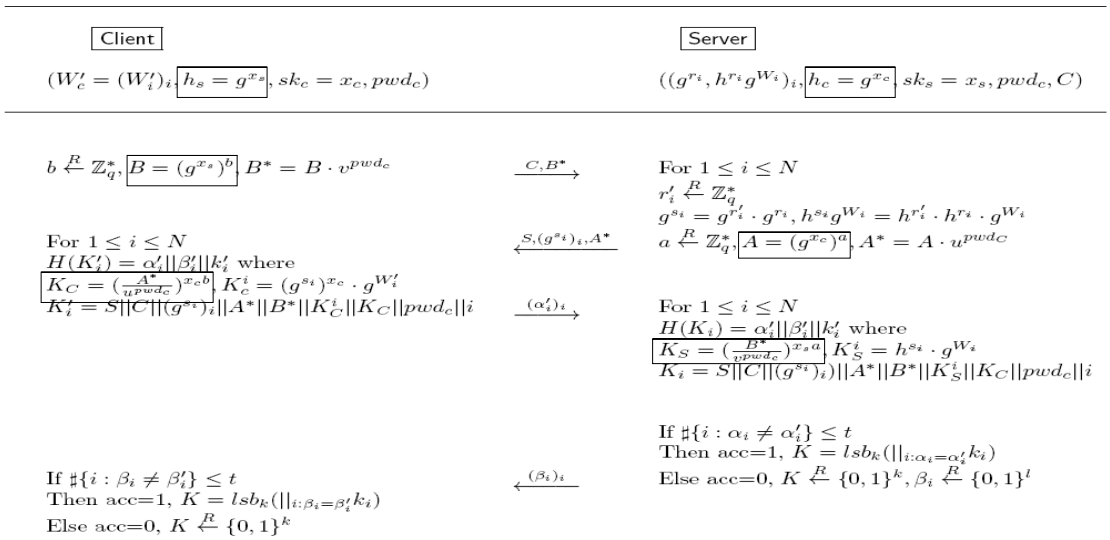
먼저, 두 번째 공격방법에 대해서 논의하자. 이 공격은 해결책이 존재한다. 즉, 프로토콜에 공개키 검증 절차를 추가하면 충분히 방어할 수 있다. 가령, 사용자가 서버로부터 받은 α_i'값에 대해 작은 그룹의 원소가 아님을 $(\alpha_i')^q \equiv 1 (\mathrm{mod}\, p)$를 검증함으로써 공격을 방어할 수 있는데, 이러한 검증 절차를 두지 않는 이유는 프로토콜의 효율성 저하 때문이다. 이러한 검증 절차에 대한 필요성은 참고문헌 〔6〕에도 언급되어 있다. 두 번째 공격에 대해서는 해결책이 존재하므로, 본 논문에서는 첫 번째 공격에 대한 방어법을 다룬다.

4.1 서버의 공개키, 개인키를 이용한 해결책

기존 프로토콜에는 사용자의 공개키 g^{x_c}, 개인키 x_c 쌍만 존재한다. 하지만, 첫 번째 공격의 방어를 위해, 서버의 공개키 g^{x_s}, 개인키 x_s를 새롭게 추가 정의한다. 이를 이용해서 기존 프로토콜을 〔그림 3〕처럼 변경하였으며, 특별히 변경된 부분은 네모 상자 처리하였고 나머지는 기존 프로토콜과 동일하다. 개선된 프로토콜의 가장 큰 특징은, A^*와 B^*를 만들 때 사용자는 서버의 공개키 g^{x_s}를 적용하고 서버는 사용자의 공개키 g^{x_c}를 적용하여 아래와 같이 계산하는 부분이다.

$$A^* = (g^{x_s})^a \cdot v^{pwd_c}, \; B^* = (g^{x_c})^b \cdot u^{pwd_c}$$

이로 인해 프로토콜에서 사용자와 서버는, K_c, K_s를 다음과 같이 각각 만들게 된다.



〔그림 3〕 개선된 Pointcheval-Zimmer 프로토콜

$$K_c = (\frac{A^*}{u^{pwd_c}})^{x_c b},\ K_s = (\frac{B^*}{v^{pwd_c}})^{x_s a}$$

정당한 사용자와 서버는 $K_c = K_s = g^{abx_s x_c}$를 형성해서 이를 이용해 K_i'을 계산하고 해시시켜 α_i', β_i'값을 유도해 서로 전달한다.

4.2 Hao와 Clarke 공격에 대한 안전성

공격자 A가 패스워드를 알았을 때 Hao와 Clarke이 제시했던 공격방법을 제안된 프로토콜에 적용해 보면 다음 〔그림 4〕와 같다. 먼저, A는 a, s_i값을 임의로 선택해서 사용자에게 $A^* = g^{x_c a} \cdot u^{pwd_c}$값을 계산해서 전달한다. 이후 〔그림 2〕의 알고리즘대로 수행하려 할 것이다. 하지만, 〔그림 4〕의 네모 상자를 계산하기 위해서는 반드시 공격자 A는 x_c값을 알아야 한다. 비록, K_c'값은 계산할 수 있지만 x_c값을 알 수 없으므로 K_c값은 계산할 수 없고, 이를 통해 궁극적으로 α_i''값을 유도할 수 없게 된다. 그러므로 사용자의 생체정보를 비교할 값이 존재하지 않는다.

- **〔1 단계〕** 공격자는 s_i, a값과 pwd_c값을 알고 있기 때문에, K_c, K_c'값을 계산해서, 1부터 N까지 K_i'값과 그 해시 값을 다음처럼 계산한다.

$$H(K_i') = \alpha_i'' || \beta_i' || k_i'$$
$$K_i' = S||C||(g^{s_i})_i||A^*||B^*||K_c'||K_c||pwd_c||i$$
$$K_c = (\frac{B^*}{v^{pwd_c}})^{ax_c},\ K_c' = (g^{x_c})^{s_i}$$

〔그림 4〕 개선된 프로토콜에 대한 생체정보 유도 알고리즘 적용

만약 A가 〔그림 3〕의 알고리즘 〔1 단계〕를 이용해 정확한 α_i''값을 유도할 수 있다면, 이는 이를 이용하여, CDH(computational diffie-hellman) 문제를 풀 수 있는 $\triangle_{cdh}$ 알고리즘을 쉽게 구성할 수 있다. CDH문제의 입력 인스턴스를 g, g^s, g^t이라 했을 때, $\triangle_{cdh}$는 그림 4의 알고리즘 〔1 단계〕의 K_c를 푸는 공격자 A가 높은 확률로 존재한다면, 이를 이용하여 〔그림 5〕처럼 알고리즘을 구성할 수 있다.

- **〔입력〕** g, g^s, g^t를 입력 받는다.
- **〔처리〕** $\triangle_{cdh}$는 사용자 및 서버의 공개키를 각각 g^s, g^t으로 대체해서 A의 프로토콜 수행을 시뮬레이션한다. 또한 공격자 A의 질의에 답하여, A^*, B^*를 시뮬레이션하며 이때 사용된 a,b값을 유지하고 있는다. A가 〔그림 3〕의 알고리즘을 이용하여 높은 확률로 $K_c = g^{stab}$를 구했을 때, $\triangle_{cdh}$는 K_c값을 이용해 CDH 입력 값에 대한 CDH 값을 다음과 같이 구한다.
- **〔출력〕** $(K_c)^{\frac{1}{b}\frac{1}{a}} = g^{st}$을 출력한다. 여기서, a,b값은 $\triangle_{cdh}$가 공격자 A의 질의를 위해 시뮬레이션할 때 사용한 랜덤 값이다.

〔그림 5〕 CDH 문제 솔루션 찾는 알고리즘

4.3 세션 키 안전성

본 절에서는 참고문헌 〔7〕에 정의되어있는 세션 키 구별가능성 성질에 대해서 논한다. 우선 본 논문에서 제시된 해법은 아래의 (1.1), (2.1) 부분을 제외하고 참고문헌 〔7〕과 동일하다. 즉, 기존 참고문헌 〔7〕의 프로토콜 (1), (2) 부분만 각각 (1.1), (2.1)로 수정되었다. 수식에서 보듯이 a와 b값이 유니폼하고 랜덤하게 선택된다면, (1.1)의 $(g^{x_s})^a, (g^{x_s})^b$ 값들도 G상의 임의의 원소 g^{x_s}, g^{x_c}에 대해서 a와 b승을 했으므로 유니폼하고 랜덤한 원소가 된다. 그러므로 (1.1)의 A^*, B^*도 유니폼하고 랜덤함은 자명하며, 이는 (2.1)의 K_c, K_s 경우도 마찬가지이다. 결론적으로, 세션 키 형성에 필요한 입력 값 중, K_c, K_s가 유니폼하고 랜덤하므로 세션 키의 안전성에 영향을 미치지 않는다.

$$(1)\ A^* = g^a \cdot v^{pwd_c},\ B^* = g^b \cdot u^{pwd_c}$$
$$(1.1)\ A^* = (g^{x_s})^a \cdot v^{pwd_c},\ B^* = (g^{x_c})^b \cdot u^{pwd_c}$$

$$(2)\ K_c = (\frac{A^*}{u^{pwd_c}})^b,\ K_s = (\frac{B^*}{v^{pwd_c}})^a$$
$$(2.1)\ K_c = (\frac{A^*}{u^{pwd_c}})^{x_c b},\ K_s = (\frac{B^*}{v^{pwd_c}})^{x_s a}$$

참고문헌 〔7〕에 정의되어 있는 안전성 모델은 세션 키에 대한 안전성만 다루고 사용자들의 비밀 값들에 대한 안전성은 정의하지 않았다. 예를 들어, 사용자가 pwd_c를 이용해서 x_c를 획득하려는 행위 혹은 생체정보를 획득하려는 행위에 대한 안전성을 확인할 수 없다. Hao와 Clarke의 공격이 가능했던 이유도 여기에 있다. 다음 절에 이러한 안전성을 분석한다.

4.4 사용자의 인증요소에 대한 안전성

제안된 해결책이 Hao와 Clarke이 제시한 공격에 안전함은 4.2절에 이미 보였다. 참고문헌 〔6〕 논문에서 의미 있는 공격으로 간주했던 것은, pwd_c와 같은 단일 인증 요소가 공격자에게 노출되었을 때 다른 인증 요소들 (생체정보, 비밀 값)을 공격자가 알 수 있다는 것이었다. 즉, 사용자의 인증요소의 노출이 다른 인증요소의 노출에 영향을 미치는지에 대한 부분은 참고문헌 〔7〕의 안전성 모델에서 다루지 않는 범위다. 본 절에서는 이에 대한 안전성을 다음의 세 경우에 대해서 분석한다.

- **공격자가 pwd_c를 알고 있으며, 이를 이용해서 생체정보 및 x_c, x_s를 유도하는 경우** : 공격자가 pwd_c를 이용해서, 생체정보를 유도할 수 없음을 4.2절에서 보였다. 생체정보 및 pwd_c를 모르기 때문에, 3.2절에서 언급했던 방법으로 x_c, x_s값을 유도할 수도 없게 된다. 물론, 유도할 수 있더라도, 쉽게 공개키 검증을 통해 방어가 가능함을 앞서 논의하였다.
- **공격자가 x_c를 알고 있으며, 이를 이용해서 생체정보 및 pwd_c를 유도하는 경우** : 프로토콜에서 패스워드 pwd_c는 $A^* = (g^{x_c})^a u^{pwd_c}, B^* = (g^{x_s})^b v^{pwd_c}$에 사용되고, x_c, x_s는 사용자, 서버의 공개키 g^{x_c}, g^{x_s}에 사용된다. 우선, 공격자가 x_c, x_s를 안다고 해도 사용자와 서버가 랜덤하게 선택한 a,b 값을 알지 못하는 한 A^*, B^*로부터 pwd_c를 유추할 수 없다.

또한, 패스워드는 공통의 세션 키를 만들 때 아래의 K_c, K_s값에 적용된다.

$$K_c = (\frac{A^*}{u^{pwd_c}})^{x_c b},\ K_s = (\frac{B^*}{v^{pwd_c}})^{x_s a}$$

이 경우에도 공격자가 x_c, x_s를 안다고 해도 사용자와 서버가 랜덤하게 선택한 a,b 값을 알지 못하므로 K_c, K_s로부터 pwd_c를 유추할 수 없다.

끝으로, 공격자가 x_c를 알고 있는 경우, 생체정보를 추측하는 경우를 고려해보자. 아래 수식에서 보듯이, 공격자는 x_c를 통해서 K_c'값의 범위를 $(g^{s_i})^{x_c} \cdot g$ 혹은 $(g^{s_i})^{x_c}$로 유추할 수 있다. 하지만, 생체정보를 올바르게 추측하기 위해서는, K_i'값을 정확히 도출해서 α_i'' 값을 계산하고 이 값은 정당한 α_i'값과 비교해서 생체

정보의 비트를 알 수 있게 된다. 하지만, 공격자는 K_c 값을 모르기 때문에 비교할 대상 값 α_i''를 계산할 수 없다. 그러므로, x_c의 노출은 생체정보와 상관이 없다.

$$K_i' = S \| C \| (g^{s_i})_i \| A^* \| B^* \| K_c' \| K_c \| pwd_c \| i$$
$$K_c = (\frac{A^*}{u^{pwd_c}})^{x_c b}, K_c' = (g^{s_i})^{x_c} \cdot g^{W_i'}$$

• **공격자가 생체정보를 알고 있으며, 이를 이용해서 pwd_c 및 x_c를 유도하는 경우** : 공격자가 생체정보를 알게 되면, 아래의 K_c', K_c값의 계산을 통해 K_i'값을 계산한 후 α_i''값을 유도해서 정당한 α_i'값과 비교해야 한다. 하지만, 공격자는 생체정보의 내용만 가지고 패스워드와 x_c, x_s를 알 수 없으므로 이를 비교할 α_i''값을 유도할 수 없다.

$$K_c = (\frac{A^*}{u^{pwd_c}})^{x_c b}, K_c' = (g^{s_i})^{x_c} \cdot g^{W_i'}$$

참고문헌 [7]에서는, 공격자가 생체정보를 얻지 못하도록 공격자의 공격 질의를 제한하였다. 즉, 살아있는 가정 (liveness assumption)을 새롭게 정의했는데, 엄밀히 말하면, 이 가정은 사용자의 생체정보는 그 사람만이 가지고 있는 것으로서, 프로토콜 수행을 위해서는 본인이 반드시 프로토콜에 참여해야함을 가정한다. 그러므로 본인이 아닌 다른 사람은 다른 사람의 생체정보를 절대 얻을 수 없음을 가정하고, 이러한 가정에 의하면, 공격자가 생체정보를 얻는다는 것 자체가 가정에 어긋나게 된다.

V. 결론 및 향후 연구 과제

본 논문에서는 Pointxcheval과 Zimmer의 다중인증 키 교환 프로토콜의 결점을 해결했다. 그 안전성 결점은 공격자가 하나의 인증 요소인 패스워드를 이용해서 다른 인증 요소 값들을 유도할 수 있다는 것으로 Hao와 Clarke에 의해 처음 밝혀졌지만 그 해결책은 아직 제시되지 못했다. 본 논문에서 제안된 해결책의 핵심은 공통된 키 값을 계산할 때 필요한 K_c값을 사용자, 서버의 개인키와 선택한 랜덤 값들을 이용해서 구성했다는 점이다. 이러한 해결책은 단순히 지수 승 2번을 더 요구하므로 매우 효율적인 방법이다. Hao와 Clarke이 제시한 공격에 대해 해결책을 제시했지만, 다중인증키교환 프로토콜에 대해서 추후 연구되어야 할 과제가 여전히 존재한다.

첫째로, 단일 인증요소의 노출이 다른 단일 인증요소의 노출에 영향을 미칠 수 있는지에 대해, 앞 절에서 분석되었다. 하지만, x_c, pwd_c를 모두 지닌 공격자를 가정했을 때 생체정보까지 안전하도록 프로토콜을 설계할 수 있는지에 대한 의문은 여전히 남는다. 이 문제의 해결은 쉽지 않다. 왜냐하면, 사용자의 생체정보는 제한된 비트 값을 지니고 있으므로 x_c, pwd_c값을 알고 있는 공격자에 대해 생체정보 비트를 노출하지 않도록 프로토콜을 설계하는 것은 쉽지 않기 때문이다. 현재까지 이와 같은 프로토콜은 제안되지 않았으며, 이를 위해서는 Pointcheval과 Zimmer 프로토콜 [7]의 근본적인 프로토콜 수정, 재설계를 요구한다. 이 문제는 시급히 연구되어야 할 중요한 주제이다. 또한, 단일 인증요소의 노출로 인한 다른 인증요소의 노출에 대한 안전성 모델이 고려되지 않았다. 참고문헌 [7]에서는 기존의 세션 키 안전성과 사용자 인증에 대해서만 고려하고 있다. 물론, 공격행위는 corrupt질의를 이용해서 단일 인증요소들을 얻을 수 있는 행위를 모델링하였지만, 인증요소의 안전성에 대해서는 고려되지 않았다. 이 부분은 다중인증요소 키 교환 프로토콜에서 충분히 중요한 이슈가될 수 있으므로 관련 안전성 모델 연구가 필요하다.

참고문헌

[1] J. W. Byun, D. H. Lee, and J. I. Lim, "EC2C-PAKA: an efficient client to client password authenticated key agreement," Information Science, vol 177, no. 19, pp. 3995-4013, Oct. 2007.

[2] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," Advances in Cryptology Eurocrypt'00, LNCS 1807, pp. 139-155, May. 2000.

[3] E. Bresson, O. Chevassut, and D. Pointcheval, "Group diffie-hellman key exchange secure against dictionary attacks," Advances in Cryptology Asiacrypt'02, LNCS 2501, pp. 497-514, Dec. 2002.

[4] S. Bellovin and M. Merrit, "Encrypted key exchange: password based protocols

secure against dictionary attacks," Proceedings of the 1992 IEEE Computer Society Symposium on Research in Security and Privacy, pp.72-84, May. 1992.

〔5〕 L. Lamport, "Password authentication with insecure communication," Communications of the ACM vol. 24 no. 11 pp. 770 - 772, Nov. 1981

〔6〕 Feng Hao and Dylan Clarke, "Security Analysis of a Multi-Factor Authenticated Key Exchange Protocol," Proceedings of Applied Cryptography and Network Security (ACNS'12), LNCS 7341, pp. 1-11. June. 2012.

〔7〕 D. PointCheval and S. Zimmer, "Multi-Factor Authenticated Key Exchange," Proceedings of Applied Cryptography and Network Security (ACNS'08), LNCS 5037, pp. 277-295, June. 2008.

〔8〕 변진욱, 정익래, 이동훈, "서로 다른 패스워드워드를 가진 사용자간의 패스워드 인증 키 교환 프로토콜," 정보보호학회논문지, 13(1), pp. 27-38, 2003년 2월

〔9〕 변진욱, "새로운 C2C-PAKA 프로토콜의 안전성 연구," 정보보호학회논문지, 22(3), pp. 473-483, 2012년 6월

〈저자소개〉

변 진 욱 (Jin Wook Byun) 정회원
2001년 2월: 고려대학교 전산학과 이학사
2003년 2월: 고려대학교 정보보호대학원 정보보호 전공, 공학 석사
2006년 8월: 고려대학교 정보보호대학원 정보보호 전공, 공학 박사
2006년 11월~2007년 12월: 영국 런던대학교, ISG 박사후 연수
2008년 03월~현재: 평택대학교 정보통신학과 조교수
〈관심분야〉 사용자 인증, 프라이버시 보호 기술, 데이터베이스 보안, 암호 프로토콜