안전한 브로드 캐스팅을 위한 Time-Bound Hierarchical Key Management 스킴 비교 분석

An Analysis of Time-Bound Hierarchical Key Management Scheme for Secure Broadcasting

김 현 철*, 구 우 권*, 이 준 호*, 이 동 훈*
(Hyun Cheol Kim, Woo Guen Goo, Jun Ho Lee, Dong Hoon Lee)

Abstract : Secure broadcasting is requirement for payment of TV systems, government or company. Hierarchical key management for access control provides efficient key management in those environment. Also, time-bound hierarchical key management technique generates different keys in each time period. In 2004, Tzeng proposed a time-bound cryptgraphic key assignment scheme for access control in a hierarchy and in 2008, Bertino et al proposed an efficient time-bound hierarchical key management scheme for secure broadcasting. *Tzeng's scheme and Bertino et al's scheme are organized in different environment and primitive. In this paper, we analysis above two time-bound hierarchical key management scheme.*

Keywords: Hierarchical key management scheme, Time-bound key management scheme, Access control, Secure broadcasting

I. 서론

hierarchical cryptographic key management 스킴은 security 레벨에 따른 액세스 컨트롤을 가능하도록 해준다. 예를 들어 흔히 알려진 멀티레벨 security 모델인 Bell-LaPadula 을 보면 security 레벨이 "unclassified", "confidential", "secret", "top-secret" 로 나누어져 있다. 이 클래스들은 "unclassified" ≤ "confidential" ≤ "secret" ≤ "top-secret" 로 나타낼 수 있다. 이때 "top-secret"은 K_4 로, "secret"은 K_3 로, "confidential"은 K_2 로, "unclassified"는 K_1 로 암호화된다. 그리고 $j \leq i$ 일때 K_i 를 이용해서 K_j 를 얻어낼수 있다. 즉 만약 K_2 를 가지고 있는 사용자는 "confidential" 과 "unclassified" 등급에 있는 컨텐츠를 복호화해서 정보를 획득 할 수 있다. 이러한 액세스 컨트롤은 컴퓨터 시스템 뿐만이 아니라 도청에 취약한 커뮤티케이션 시스템에도 유용하다.

또한 일반적인 access control에 time-bound개념을 도입하여 키의 유효기간 내에만 켄텐츠를 복호화해서 정보를 획득 할 수 있도록 하고 있다. 예를들어 컨텐츠의 유효기간이 t_1, t_2 로 주어졌을때 사용자는 $t1 \leq t \leq t2$ 사이일때만 컨텐츠를 복호화 할 수 있다.

hierarchical cryptographic key assignment는 1983년에 Akl과 Talor가 처음으로 연구하였고 이후에 다른 연구가 진행되었다[1][2][3][4]. 2002년도에 Tzeng이 time-bound 개념이 들어간 cryptographic key assignment scheme 논문을 발표하였고[5], 2008년도에 Bertino등이 tamper-proof장비를 사용하여 Tzeng의 문제를 보완한 논문을 발표하였다[6].

본 논문에서는 2004년도에 발표된 Tzeng의 논문과 2008년도에 발표된 bertino등의 논문을 비교분석한다. 이후 논문 구성은 다음과 같다. 2장에서는 Tzeng의 논문을 분석하고 3장에서 Bertino등의 논문을 분석한다. 4장에서는 두개 논문을 비교하고 5장에서 결론을 맺는다.

주저자: skycube99@gmail.com
*고려대학교 정보경영공학전문대학원
"본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음" (IITA-2008-(C1090-0801-0025)).

II. Tzeng 의 스킴

Tzeng의 스킴은 RSA에 기본하고 있다. 즉 DLP(Diffie Hellman Problem)문제를 이용해 다른 클래스에 있는 사람은 키를 구하지 못하도록 한다. 서로 다른 클래스는 C_1, C_2, C_3 ...C_m 으로 구분한다.

1. Tzeng 스킴의 설계

세팅. CA(Certification Authority)는 서로 다른 큰 소수 (p_1,q_1)과 (p_2,q_2) 쌍을 선택하고 $n_1=p_1*q_1$ 과 $n_2=p_2*q_2$를 계산한다. 그 다음 CA는 $\Box(n_1) = (p_1-1)(q_1-1)$ 을 만족하는 서로소인 $e_1,e_2\ldots,e_m,g_1,g_2$ 을 랜덤하게 선택하고 $e_id_i=1(\text{mod } \Box(n_1))$과 $g_jh_j = 1(\text{mod } \Box(n_1))$을 만족하는 $d_1,d_2,\ldots,d_m,h_1,h_2$를 결정한다.($i = 1,2,\ldots,m$ 그리고 $j=1,2$) CA는 1과 n_1 사이에서 랜덤하게 a를 선택하고 1 과 n_2 사이에서 b를 선택하고 유효기간으로 사용할 두 정수 f_1과 f_2를 선택한다. 그 후에 CA는 $g_1,g_2,f_1,f_2,e_1,\ldots,e_m,n_1,n_2$를 공개하고 그 외 값은 비밀 값으로 한다.

키 생성 및 할당. time-bound가 $(1 \leq t \leq z)$일 때 C_i 클래스는 다음과 같은 키를 할당 받는다.

$$K_{i,t} = H(K_i^{h_1^t h_2^{z-t}} (\text{mod } n_1), w_t)$$

H는 one-way 해쉬 function을 의미한다. 여기서 K_i는 다음과 같다.

$$K_i = a^{\prod_{C_l \leq C_i} d_l} (\text{mod } n_1)$$

그리고 w_t는 다음과 같다.

$$w_t = V_{f_1^{z-t} f_2^t}(b)(\text{mod } n_2)$$

여기서 V는 Lucas function[7]을 의미한다. Tzeng은 Lucas function이 가지고 있는 성질중 $V_i(V_j(b))=V_{ij}(b)(\text{mod } n_2)$를 이용한다. 위 식을 이용하여 CA는 $K_{i,t}$를 계산하고 다음과 같은

키 정보를 클래스 C_i 에 있는 사람에게 전달한다.

$$I(i,t_1,t_2)=(K_i^{h_1^{t_2}h_2^{z-t_1}}(\bmod n_1),V_{f_1^{z-t_2}f_2^{t_1}}(b)(\bmod n_2))$$

키 추출. Ci에 있는 사용자는 $I(i,t_1,t_2)$를 이용하여 자신의 키를 알 수 있고 자신보다 하위에 있는 ($C_j \le C_i$)클래스의 키를 추출할 수 있다. 다음과 같은 방법으로 C_j 클래스의 키를 추출한다.

$$(K_i^{h_1^{t_2}h_2^{z-t_1}})^{g_1^{t_2-t}g_2^{t-t_1}\prod_{C_l\le C_i, C_l\not\le C_j}e_l}$$

g와 h는 $g_ih_j=1(\bmod \Box(n_1))$ 이므로 g를 h로 변환하면

$$=(K_i^{\prod_{C_l\le C_i, C_l\le C_j}e_l})^{h_1^{t_2}h_2^{z-t_1}h_1^{-t_2+t}h_2^{-t+t_1}}=K_i^{h_1^t h_2^{z-t}}(\bmod n_1)$$

이다. 또 w_t값은 다음과 같이 구한다.

$$V_{f_1^{t_2-t}f_2^{t-t_1}}(V_{f_1^{z-t}f_2^t}(b))=V_{f_1^{t_2-t}f_2^{t-t_1}f_1^{z-t}f_2^t}(b)$$
$$=V_{f_1^{z-t}f_2^t}(b)=w_t(\bmod n_2)$$

이렇게 구한 K_i 값과 w_t 값을 이용하여 $K_{j,t}$ 값을 구할 수 있다.

2. 시나리오

CA: 클래스가 그림 1 과 같이 나누어져 있을 때 CA는 각 클래스마다 K를 생성한다. 즉, C_1 에 K_1, C_2에는 K_2, C_3에는 K_3, C_4 에는 K_4를 생성한다. 또한 각 클래스마다 $I(i,t_1,t_2)$를 생성해서 각 클래스 사용자에게 분배한다. z가 5일 때 time period는 0,1,2,..,5까지 여섯개가 된다. 컨텐츠는 각 클래스의 키로 암호화된다.

사용자: $C_j(C_j \le C_i)$ 클래스에 있는 사용자는 $I(i,t_1,t_2)$를 통해서 $K_{i,t}$를 알 수 있고 키 추출 과정을 통해서 $K_{j,t}(C_j$ 의 키)를 알 수 있다. 따라서 자신이 위치한 클래스에 제공된 컨텐츠를 복호화 할 수 있고 자신보다 아래에 위치한 클래스에 제공된 컨텐츠도 복호화 하여 정보를 획득 할 수 있다.

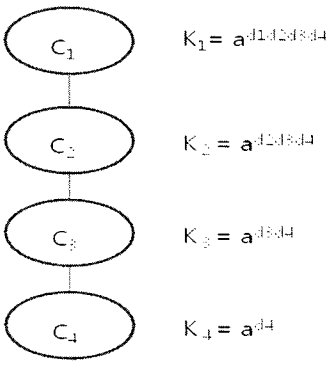

그림 1. 클래스 구조

III. Bertino 등의 스킴

Bertino등의 스킴은 HMAC의 키에 안전성을 두고 있다. 이 스킴은 Tzeng의 스킴을 보완하고자 tamper-proof 장비에 비밀값들을 저장하여 안전하게 보관되어 있다고 가정한다. Tzeng의 스킴과 크게 다르지 않고 ECC와 tamper-proof 장비를 사용한다.

2. Bertino등의 스킴 설계

세팅. 벤더는 유한체(IF_q)상에서 타원곡선 E를 선택하고 큰 소수 위수 p와 점 $Q \in E(IF_q)$를 선택한다. 또 $n_ig_i(\bmod p)$ 값이 서로 다른 $n_1,n_2,\ldots,n_m,g_1,g_2,\ldots,g_m$을 선택한다. 그 다음 벤더는 $E(IF_q)$상에서 $P_i = n_iQ$ 와 $g_ih_i \equiv 1(\bmod p)$를 만족하는 h_i를 계산한다. 벤더는 랜덤하게 정수 a,b를 선택하고 HMAC의 비밀키를 정한다. g_i, h_i, a, b값은 공개하지 않는다.

키 생성 및 할당. 벤더는 $K_i = g_iP_i$를 계산하고 C_i 클래스 사용자에게 안전한 채널로 전달한다. 또 $R_{i,j} = g_iK_j+(-k_i)$를 계산하여 공개한다. ($C_j \le C_i$) 벤더는 Ci 클래스의 키를 다음과 같이 구한다.

$$K_{i,t}=H_K((K_i)_Y \oplus H^t(a) \oplus H^{Z-t}(b) \oplus ID_i)$$

H_k 는 HMAC을 나타낸다.

키 추출. 사용자는 tamper-proof 장비를 이용해서 암호화 정보($EncInf_i$)를 획득한다. tamper-proof장비에는 H_K, E, IF_q, ID_i, h_i와 $EncInf_i$가 저장되어 있다. $EncInf_i$는 다음과 같이 구성된다.

$$EncInf_i=(H^{t_1}(a),H^{Z-t_2}(b))$$

사용자는 K_i와 $EncInf_i$를 이용해서 자신의 키 $K_{i,t}$를 생성할 수 있다. $C_j \le C_i$일 때 $K_{j,t}$를 구하기 위해서 tamper-proof 장비에 K_i와 공개값인 ID_j를 입력한다. 그러면 tamper-proof장비는 다음을 계산한다.

$$K_j=h_i\cdot(R_{i,j}+K_i)$$

또한 해쉬값을 구하기 위하여 다음을 계산한다.

$$H^t(a)=H^{t-t_1}(H^{t_1}(a)),H^{Z-t}(b)=H^{t_2-t}(H^{Z-t_2}(b))$$

위와 같이 K_j, $H^t(a)$, $H^{Z-t}(b)$를 이용하면 $K_{j,t}$를 구할 수 있다.

2. 시나리오

벤더: 각 클래스 마다 K_i와 $K_{i,t}$를 구하고 tamper-proof 장비에 H_K, E, IF_q, ID_i, h_i 와 EncInf를 입력한다. 각 클래스의 사용자에게 tamper-proof장비를 분배한다. 또 $R_{i,j}$값을 공개하고 K_i 값을 안전한 채널로 전달한다.

사용자: 사용자는 자신의 tamper-proof장비에 K_i를 입력하면 클래스 키($K_{i,t}$)를 획득할 수 있다. 또한 tamper-proof 장비에 K_i와 자신보다 하위 클래스에 있는 사용자의 ID_j를 입력하면 $K_{j,t}$를 획득 할 수 있다.($C_j \le C_i$)

IV. 비교 분석

1. 키 설계

Tzeng의 스킴은 사용자들의 공모 공격에 취약하다[8]. 사용자 A, B와 C가 있다고 가정하고 $C_k \leq C_j \leq C_i$ 이고 $I(i,t_1,t_2)$, $(j,t_3,,t_4)$, (k,t_5,t_6)을 가질 때 $K_{j,t}$를 구할 수 있다. 이미 Kj,t를 가지고 있는 B가 참여해서 공모에 가담하였으므로 이 공격은 크게 의미가 있다고 보기 힘들다. 하지만 직접적으로 키를 전달하지 않으면서 하위 클래스에 있는 C가 상위 클래스의 키($K_{j,t}$)를 구할 수 있다는 점에서 취약점으로 볼 수 있다.

반면 Bertino등의 스킴에서는 tamper-proof 장비를 이용해서 키를 관리하기 때문에 공모공격을 원천적으로 차단하고 있다. 단, tamper-proof 장비가 안전하다는 가정에 기반하고 있다.

또 Tzeng의 스킴은 처음에 CA가 $I(i,t_1,t_2)$를 이용해서 각 클래스에 맞는 키를 안전하게 전달하는 반면에 Bertino등의 스킴에서는 $EncInf_i$ 를 이용해서 K_i만 안전하게 전달하고 각 클래스에 있는 사용자들이 직접 자신의 키를 계산해서 사용하도록 하고 있다. 이러한 측면에서는 Bertino등의 스킴이 더 안전하다고 할 수 있다.

하위 클래스에 있는 사용자가 상위 클래스의 키를 구할 수 없도록 하기위해서 두 스킴 모두 DLP 를 사용하고 있다.

2. Time-bound 요소

Tzeng의 스킴에서 time-bound를 구하기 위해 I(i,t1,t2)을 받았을 때 w_t를 다음과 같이 구한다.

$$w_t = V_{f_1^{z-t_2} f_2^{t_1}}(b)$$

이때 $t'<t_1$인 시간에 대해서 $w_{t'}$를 구하기 위해서는 f_2의 역함수 즉, $f^{-1}{}_2\,(p^2-1)(q^2-1)$을 알아야 한다. 또 $t_2<t'$인 시간에 대해엇 $w_{t'}$를 구하기 위해서는 f_1의 역함수, $f^{-1}{}_1(p^2-1)(q^2-1)$을 알아야한다. 이것은 계산적으로 불가능 하므로 유효기간을 벗어나 합법적인 키를 생성할 수 없다.

Bertino등의 스킴에서 time-bound를 구하기 위해 $EncInf_i = (H^{t_1}(a), H^{Z-t_2}(b))$ 를 사용한다. 예를들어 벤더의 z가 70이고 사용자의 유효기간이 t_1=8, t_2=14 일때 사용자는 EncInf를 통해서 $H^8(a)$, $H^{56}(b)$를 알수 있다. 이때 사용자가 t=10일 때 키를 구하기 위해서 $H^{10}(a)$, $H^{60}(b)$를 알아야한다. $H^{10}(a)=H^8(H^2(a))$, $H^{60}(b)=H^{56}(H^4(b))$이므로 키를 생성할 수 있다. 하지만 t'<t1 일때나 t2<t' 일때에 키를 생성하려고 할 때 해쉬함수의 one-wayness 성질 때문에 키를 생성할 수 없다.

V. 결론

2008년에 Bertino등은 이전 hierarchical key management 스킴의 문제점을 해결하면서 더 안전한 스킴을 제안하였다. 하지만 Bertino등의 스킴은 tamper-proof 장비의 안전성에 의존하고 있다. 상용 서비스를 위한 tamper-proof 장비는 안전한 브로드 캐스팅 시스템을 구축 하는데 비교적 비용이 많이 드는 부분이다. 저가의 tamper-proof 장비는 안전성이 떨어질 수 있고 고가의 tamper-proof 장비를 사용하기에는 비용이 너무 많이 드는 것이다. 데이터의 기밀성이 중요시 되는 분야에서는 tamper-proof 장비를 사용한 hierarchical key management는 효과적인 수단이 될것이다. 하지만 기밀성이 덜 중요한 분야에서는 tamper-proof 장비 없이 안전성을 유지하는 현실적인 스킴이 필요하다.

참고문헌

[1] C. C. Chang, R. J. Hwang, and T.C. Wu, "Cryptographic Key Assignment Scheme for Aceess Control in a Hierearchy," Information Systems,vol. 17, no. 3, pp.243-247, 1992.

[2] L. Harn and H. Y. Lin, "A cryptographic Key Generation Scheme for Multilevel Data Security," Computers and Security, vol. 9, no. 6, pp. 539-546, 1990

[3] S. J. Mackinnon, P.D. Talor, H. Meijer, and S. G Akl, "An Optimal Algorithm for Assigning Cryptographic Keys to Control Access in a Hierarchy," Trans. Computers, vol. 34, no. 9, pp. 797-802, 1985.

[4] R. S. Sandhu, "Cryptographic Implementation of a Tree Hierarchy for Access Control," Information Processing Letters, no. 27, pp.95-98, 1988

[5] W.G Tzeng, "A Time-bound Cryptographic Key Assignment Scheme for Access control in a Hierarchy," IEEE Trans. Knowledge and Data Eng. vol. 14, no. 1, pp. 182-188, Jan, 2002.

[6] E. Bertino, N. Shang, S. S. Wagstaff, "An Efficient Time-bound Hierarchical Key Management Scheme for Secure Broadcasting", IEEE Trans. Dependable and secure computing, vol. 5, no. 2, 2008.

[7] S. M. Yen and C. S. Laith, "Fast Algorithm for LUC Digital Signature Computation," IEE Proc.-Computers and Digital Technique, vol. 142, no.2, pp.165-169, 1995.

[8] Security of Tzeng's Time-bound Key Assignment Scheme for Access Conrol in a Hierarchy", IEEE Trans. Knowledge and Data Eng, vol. 15, no. 4, 2003.