일반논문 (Regular Paper)
방송공학회논문지 제21권 제3호, 2016년 5월 (JBE Vol. 21, No. 3, May 2016)
http://dx.doi.org/10.5909/JBE.2016.21.3.412
ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

비밀분산 기반의 효율적인 전송량을 갖는 브로드캐스트 암호시스템

이 재 환[a)], 박 종 환[a)‡]

A Transmission-Efficient Broadcast Encryption System Based on Secret Sharing Method

Jae Hwan Lee[a)] and Jong Hwan Park[a)‡]

요 약

브로드캐스트 암호시스템은 한명의 송신자가 다수의 수신자에게 메시지를 안전하게 전송하는 기법이다. 그 효율성은 암호문 전송량, 사용자 저장량, 복호화 연산량으로 측정되는데, 보통 대규모 수신자를 가정하므로 암호문 전송량을 줄이는 것이 가장 중요한 것으로 고려된다. 본 논문에서는 Shamir의 비밀분산 방식을 이용하여 전송량을 크게 줄이는 브로드캐스트 암호기법을 새롭게 제안한다. 기존의 Subset Difference (SD) 기법과 비교하면, 전체 사용자 수 n에 대하여 탈퇴자 수가 $\sqrt{n}$ 보다 작으면 SD 기법 전송량이 작지만, 탈퇴자 수가 $\sqrt{n}$ 보다 커지면 제안 기법의 전송량이 작게 된다. 이러한 장점은 사용자 저장량과 복호화 연산량을 약간 증가시키면서 얻을 수 있다.

Abstract

Broadcast encryption (BE) is a cryptographic primitive that enables a sender to broadcast a message to a set of receivers in a secure channel. The efficiency of BE is measured by three factors: ciphertext transmission cost, user storage cost, and computational cost for decryption. In general, BE is applied to the environments where a large number of receivers should be accommodated, so that the transmission cost is considered as being the most important factor. In this paper, we suggest a new BE system, using Shamir's secret sharing method, which considerable reduces the transmission cost. In comparison to the previous Subset Difference (SD) system, the transmission size of our BE is longer until $r \leq \sqrt{n}$, but get shorter when $r \geq \sqrt{n}$ for r number of revoked users and n number of total users. We show that the advantage can be achieved at the slight expense of both the storage and computational costs.

Keyword : broadcast encryption, secret sharing

a) 상명대학교 ICT융합대학 컴퓨터과학과 (Department of Computer Science, College of ICT Convergence, Sangmyung University)
‡ Corresponding Author : 박종환(Jong Hwan Park)
E-mail: jhpark@smu.ac.kr
Tel: +82-2-781-7589
ORCID: http://orcid.org/0000-0003-2742-6119
※ 이 논문은 2014년도 정부(교육부)의 재원으로 한구연구재단의 지원을 받아 수행된 기초연구사업임(NRF-2014R1A1A2059802).
· Manuscript received February 26, 2016; revised May 24, 2016; accepted May 27, 2016.

Ⅰ. 서 론

브로드캐스트 암호시스템[1]은 한명의 송신자가 다수의 권한이 있는 수신자에게 메시지를 안전하게 전송할 수 있는 방법이다. 정당한 권한이 없는 사용자는 메시지를 암호화한 암호문에 접근하더라도 내용을 알 수 없어야 하는데, 이를 위해 브로드캐스트 암호시스템에서는 정당한 권한이 있는 사용자들만 구할 수 있는 그룹키를 이용하여 메시지를 암호화한다.

브로드캐스트 암호시스템의 효율성은 암호문 전송량, 사용자의 비밀키 저장량, 그리고 그룹키 계산을 위한 연산량 측면에서 분석이 된다. 세 가지 효율성 요소 중 짧은 전송량을 갖는 것이 가장 중요한 것으로 간주되는데, 그 이유는 일반적으로 브로드캐스트 암호시스템이 대규모 사용자를 대상으로 하므로 암호문의 전송량이 전체 네트워크 트래픽에 주는 영향이 크기 때문이다. 또한 브로드캐스트 암호시스템은 (공개키 없이) 지정된 송신자만 전송할 수 있는 대칭키 기반 기법[2]과 공개키를 이용하여 누구나 송신자가 될 수 있는 공개키 기반 기법[3][4][5]으로 분류된다.

지금까지 제안된 대칭키 기반 브로드캐스트 암호시스템 중 가장 효율적인 것으로는 2001년 Naor, Naor, Lotspiech[2]가 제안한 이진트리 기반의 SD(Subset Difference) 기법(이하에서는 SD^{PRG} 기법으로 표기함)이다. SD 기법은 사용자의 비밀키와 Subset에 따른 그룹키를 생성하기 위해 PRG(Pseudo-random generator, 유사난수생성기)의 일방향성을 이용하여 키를 분배한다. 이후 PRG를 이용한 키 분배 방식을 변형하여 세 가지 효율성 요소 간에 trade-off를 제공하는 기법들[6][7][8]이 제안되었다.

최근에는 이재환 등[9][10]이 Shamir의 비밀분산(Secret Sharing) 기법을 이용하여 키 분배를 하는 SD 기법과(이하에서는 SD^{SS} 기법으로 표기함) 그것을 확장한 2-SD기법(이하에서는 $2-SD^{SS}$ 기법으로 표기함)을 제시하였다. 기존 PRG 기반에 비해 암호문 전송량의 길이는 다소 증가하지만 증명과정에서 나타나는 안전성 손실(security loss)이 적고, 그룹키 계산을 위한 연산량이 $O(1)$으로 전체 사용자 수나 탈퇴자 수에 무관하다는 장점이 있었다. 암호 이론적으로는 SD 기법의 키 분배를 위해 PRG가 아닌 새로운 접근방법 - 정보 이론적으로 안전한 비밀분산 기법 - 을 이용한 것이 의미 있는 결과였다.

본 논문에서는 [9]에서 소개된 비밀분산 방식의 키 분배 아이디어를 응용하여 효율적인 전송량을 갖는 브로드캐스트 암호시스템을 설계하고자 한다. 제안하는 기법은 SD^{ss} 기법과 같이 비밀분산 방식의 키 분배를 사용하지만 이진트리기반의 Subset Difference기법에 기반을 두지 않고 고정된 그룹 내에서의 키 분배를 통해 전송량을 낮추는 것을 목표로 하였다. 탈퇴자수 r과 전체 사용자수 n에 대하여 제안하는 기법의 전송량은 worst case에서 $r+\sqrt{n}$가 되는데 기존 SD^{PRG} 기법의 $2r-1$와 비교하여 탈퇴자 수가 $\sqrt{n}$ 이상 일 때 전송량 측면에서 유리해진다. 4장에서 제안하는 기법의 상세한 과정을 기술한다.

Ⅱ. 브로드캐스트 암호시스템과 안전성 정의

1. 브로드캐스트 암호시스템

양의 정수 n을 전체 사용자 수라고 하자. N을 전체 사용자 집합이라 하고, R은 탈퇴자들의 집합이라 하자. [2]를 따라서 브로드캐스트 암호시스템을 초기설정(setup), 암호화(Encryption), 복호화(decryption)의 세 개의 알고리즘으로 정의한다.

1. **Setup**(λ, n): 시스템 파라미터 λ와 사용자 수 n을 입력받고 사용자 $u\,(u=1,...,n)$에게 비밀키 d_u를 준다.
2. **Encryption**(R, M): 탈퇴자 집합 R과 메시지 M을 입력받고, 탈퇴자들이 복호화하지 못하도록 암호문 CT를 생성한다.
3. **Decryption**(CT, d_u): 사용자 u는 $N \setminus R$에 속한다고 하면, CT와 비밀키 d_u를 입력받고 암호문을 복호화하여 M을 출력한다.

2. Subset Cover 방식에 기반 한 브로드캐스트 암호시스템

Subset Cover 방식은 $N \setminus R$을 서로 겹치지 않은 부분집합들 $S_{i_1},...,S_{i_m}$으로 분리한다. $N \setminus R = \bigcup_{j=1}^{m} S_{i_j}$ 가 된다. 이러한 부분집합을 찾는 알고리즘을 'CoverFinding 알고리즘'이라 하고, 입력 R에 대해 부분집합 출력하는 것을 $CoverFinding(R) = \{S_{i_1}, \ldots, S_{i_m}\}$로 표현하자. 각각의 부분집합 S_j들은 그에 대응하는 그룹키 L_j가 할당되고, S_j안의 모든 수신자는 L_j를 알 수 있다. 메시지 암호화에 사용하는 (일회용) 키 K는 $S_{i_1},...,S_{i_m}$들의 각 그룹키 $L_{i_1},...,L_{i_m}$로 암호화하고, 메시지는 K로 암호화된다. 즉, Subset Cover 방식에서는 두 종류의 대칭키 암호시스템이 사용된다.

1. $\ddot{E}_K : \{0,1\}^t \mapsto \{0,1\}^*$: M을 메시지 암호키 K로 암호화하는 대칭키 암호
2. $E_L : \{0,1\}^t \mapsto \{0,1\}^t$: 부분집합에 대응하는 그룹키 L로 메시지 암호키 K를 암호화하는 대칭키 암호

이러한 Subset Cover 방식에 기반 한 브로드캐스트 암호시스템은 다음의 세 알고리즘으로 설명할 수 있다.

2.1 초기설정(Setup)

수신자 u는 비밀키 d_u를 받는다. S_i에 속한 u는 d_u를 이용, S_i에 대응되는 그룹키 L_i를 유도한다. 여기서 그룹키 L_i는 각 그룹 별로 독립적이고 랜덤한 값으로 할당하거나, 유사난수 값을 이용하여 할당한다.

2.2 암호화(Encryption)

1. 랜덤하게 메시지 암호키 K를 선택한다.
2. 탈퇴자 집합 R에 대해 알고리즘 $CoverFinding(R)$을 수행하여 부분집합들 $S_{i_1},...,S_{i_m}$를 구하고, 각 부분집합에 대응하는 그룹키 $L_{i_1},...,L_{i_m}$를 구한다.
3. 메시지 암호키 K와 $L_{i_1},...,L_{i_m}$를 이용하여 메시지 M을 다음과 같이 암호화한다.

$$CT =< [i_1, i_2, \ldots, i_m, E_{L_{i_1}}(K), E_{L_{i_2}}(K), \ldots, E_{L_{i_m}}(K)], \ddot{E}_K(M) >$$

2.3 복호화(Decryption)

수신자 u는 암호문을 수신하면 d_u를 이용하여 암호문을 복호화한다.

$$CT =< [i_1, i_2, \ldots, i_m, C_1, C_2, \ldots, C_m], C >$$

1. $u \in S_{i_j}$가 되는 i_j를 찾는다. $u \in R$의 경우에는 i_j를 찾을 수 없다.
2. d_u로부터 대응하는 그룹키 L_{i_j}을 유도한다.
3. $D_{L_{i_j}}(C_j)$로 복호화하여 메시지 암호키 K를 얻는다.
4. $\ddot{D}_k(C)$로 복호화하여 메시지 M을 얻는다.

3. 브로드캐스트 암호시스템의 안전성

본 논문에서는 [9]에서 제시한 브로드캐스트 암호시스템의 안전성 모델을 따른다. 약화된 선택 암호문 중간공격(weak CCA1: Chosen ciphertext and launch-time attack)으로써, 대부분의 브로드캐스트 응용 환경에서 충분하다고 간주되는 공격이다. 이 공격의 특징은 암호 및 복호 질의에 사용된 사용자들은 모두 탈퇴자로 간주되어 이후의 공격에서도 계속해서 탈퇴된 자 집합에 포함된다는 것이다. 이러한 weak CCA1 안전성은 공격자 A와 챌린저 B사이의 다음 게임으로 정의된다.

- **Setup.** B가 Setup(λ,n) 알고리즘을 수행하여 $u(u \in U)$ 각각에 대한 비밀키를 생성한다.
- **Adversarial Action.** B는 초기 탈퇴자 그룹 R을 $\varnothing$로 설정한다. A는 다음의 세 가지 질의를 할 수 있다. (1) A가 u'의 비밀키 $d_{u'}$를 요청한다. 이때 $R \leftarrow R \cup u'$로 갱신된다. (2) A가 집합 R'과 메시지 M을 B에게 보내면, 대응하는 암호문을 받는다. 이때 $R \leftarrow R \cup R'$

로 갱신된다. (3) A가 집합 R' 하에서 생성된 암호문과 임의의 $u \in R'$를 B에게 보내면, u의 비밀키로 암호문을 복호화하여 얻은 메시지를 받는다. 이때 $R \leftarrow R \cup R'$로 갱신된다.

- **Challenge.** A는 메시지 M^*과 그때까지의 탈퇴자 집합 $R^* = R$을 B에게 보낸다. B는 랜덤한 bit $b \in \{0,1\}$을 선택한다. b=1인 경우 Encrypt(R^*, M^*)의 결과를 암호문으로 A에게 준다. b=0인 경우는 M^*와 같은 길이를 갖는 랜덤 메시지 R_M를 선택하여 Encrypt(R^*, R_M) 결과를 A에게 준다.

- **Guess.** A는 추측한 $b' \in \{0,1\}$를 내보낸다.

A가 bit b를 정확하게 추측한 상황을 $CorrectGuess$로 나타내자. 시스템 파라미터 λ에 대해 A의 advantage는 $Adv^{BE}_{A, wCCA1}(\lambda) = |\Pr[CorrectGuess] - 1/2|$로 정의된다.

정의 1. 브로드캐스트 암호시스템이 weak CCA1 공격 환경에서 공격자 A가 가지는 $Adv^{BE}_{A, wCCA1}(\lambda)$가 무시할 만한(negligible) 수준이라면, '*브로드캐스트 암호시스템이 weak CCA1 공격에 안전하다*'라고 말한다.

4. 메시지 암호키를 암호화하는 대칭키 암호의 안전성

메시지 암호키 K를 암호화하기 위해 필요한 대칭키 암호시스템 SKE=(E, D)의 안전성을 정의한다. 이러한 대칭키 암호의 CCA1 안전성은 공격자 A와 챌린저 B사이의 다음과 같은 게임으로 정의된다.

- **Setup.** B가 SKE의 랜덤 비밀키 L을 생성한다.

- **Adversarial Action.** A는 다음의 두 가지 질의를 할 수 있다. (1) A가 선택한 메시지m_i을 B에 보내서 그에 대응하는 암호문 $E_L(m_i)$을 받는다. (2) A가 선택한 암호문 C_i를 B에 보내서 복호화된 메시지 $D_L(C_i)$를 받는다.

- **Challenge.** A는 메시지 m^*을 B에게 보낸다. B는 랜덤한 bit $b \in \{0,1\}$을 선택한다. b=1인 경우에는 $E_L(m^*)$를 A에게 준다. b=0인 경우는 m^*와 같은 길이를 갖는 랜덤 메시지 R_M를 선택하여 $E_L(R_M)$를 A에게 준다.

- **Guess.** A는 추측한 $b' \in \{0,1\}$를 내보낸다.

A가 bit b를 정확하게 추측한 상황을 $CorrectGuess$로 하자. 안전성 상수 λ에 대해 A의 advantage는 $Adv^{SKE}_{A, CCA1}(\lambda) = |\Pr[CorrectGuess] - 1/2|$로 정의된다.

정의 2. 대칭키 암호시스템이 CCA1 공격 환경에서 공격자 A가 가지는 $Adv^{SKE}_{A, CCA1}(\lambda)$이 무시할 만한(negligible) 수준이라면, 우리는 '*대칭키 암호시스템이 CCA1 공격에 안전하다*'라고 말한다.

5. 메시지를 암호화하는 대칭키 암호의 안전성

메시지 M은 다른 대칭키 암호시스템 $S\ddot{K}E$=($\ddot{E}$, $\ddot{D}$)을 이용하여 암호화된다. CCA1 공격에 안전한 브로드캐스트 암호시스템을 설계하기 위해서는 $S\ddot{K}E$가 One-time 선택 평문 공격(CPA: chosen-plaintext attack)에 안전해도 충분하다. 대칭키 암호의 One-time CPA 안전성은 공격자 A와 챌린저 B사이의 다음과 같은 게임으로 정의된다.

- **Setup.** B가 $S\ddot{K}E$의 랜덤 비밀키 K을 생성한다.

- **Challenge.** A는 메시지 m^*을 B에게 보낸다. B는 랜덤한 bit $b \in \{0,1\}$을 선택한다. b=1인 경우에는 $\ddot{E}_K(m^*)$를 A에게 준다. b=0인 경우는 m^*와 같은 길이를 갖는 랜덤 메시지 R_M를 선택하여 $\ddot{E}_K(R_M)$를 A에게 준다.

- **Guess.** A는 추측한 $b' \in \{0,1\}$를 내보낸다.

A가 bit b를 정확하게 추측한 상황을 $CorrectGuess$로 하자. 안전성 상수 λ에 대해 A의 advantage는 $Adv^{S\ddot{K}E}_{A, CPA^{OT}}(\lambda)$

$Adv^{SKE}_{A,\,CPA^{OT}}(\lambda) = |\Pr[CorrectGuess] - 1/2|$로 정의된다.

정의 3. 대칭키 암호시스템이 one-time CPA 공격 환경에서 공격자 A가 가지는 $Adv^{SKE}_{A,\,CPA^{OT}}(\lambda)$이 무시할 만한(negligible) 수준이라면, 우리는 '*대칭키 암호시스템이 one-time CPA 공격에 안전하다*'라고 말한다.

6. Shamir의 비밀분산 기법

본 논문에서 필요한 비밀분산(SS: Secret Sharing) 기법은 Shamir의 다항식 기반의 $(k,n)-SS$ 기법이다. p를 다항식 설정을 위한 소수(prime)라 하자. $k-1$차 다항식 $f(x) = a_{k-1}x^{k-1} + ... + a_2x^2 + a_1x + K \in Z_p[x]$를 선택한다. 여기서 계수 $a_i \in Z_p (i = 0,1,...,k-1)$는 랜덤하게 선택된 값이고, 상수항 $K \in Z_p$는 공유하고자 하는 비밀값이다. 서로 다른 n개의 값 $x_1, ..., x_n$에 대하여 $f(x_1),..., f(x_n)$을 share라 하면, 비밀값 K는 전체 n개의 share 중 k개 이상의 값을 알면 Lagrange 보간법을 사용하여 복구할 수 있다. k개 미만의 share를 아는 경우 K에 대한 어떠한 정보도 정보 이론적으로 알 수 없다.

이러한 K의 안전성은 공격자 A와 챌린저 B사이에서 이루어지는 다음의 게임으로 정의된다. $(k,n)-SS$의 경우, B는 $k-1$차 임의의 다항식 $f(x) \in Z_p[x]$를 선택한다. A는 $k-1$개의 $\{x_i\}$ 값들을 선택하여 B에게 주면, B는 함수값 $\{f(x_i)\}$ 값들을 계산한다. B는 랜덤 bit $b \in \{0,1\}$을 선택한다. $b=1$인 경우에는 $\left(\{f(x_i), x_i\}_{(i=1,2,...,k-1)}, K\right)$를 A에게 주고, $b=0$인 경우는 랜덤한 $R_K \in Z_p$를 선택하여 $\left(\{f(x_i), x_i\}_{(i=1,2,...,k-1)}, R_K\right)$를 A에게 준다. 이 게임에서 A가 bit b를 정확하게 추측한 상황을 $CorrectGuess$로 하자. 이 경우 다항식 f는 A에게 정보 이론적으로 K에 대한 어떠한 정보도 노출하지 않는다는 것을 쉽게 알 수 있다. 따라서 $|\Pr[CorrectGuess] - 1/2| = 0$이 된다. 따라서 다음과 같은 식을 얻는다.

$$|\Pr[A(\{f(x_i), x_i\}_{(i=1,2,...,k-1)}, K) = 1] - \Pr[A(\{f(x_i), x_i\}_{(i=1,2,...,k-1)}, R_K) = 1]| = 0$$

Ⅲ. Secret Sharing을 이용한 브로드캐스트 암호시스템의 아이디어

1. Secret Sharing을 이용한 방법

본 논문에서는 이진트리에 기반을 둔 SD^{SS}[9]기법과는 다르게 비밀분산기법을 활용하고자 한다. SD^{SS}기법이 1차 다항식 $f(x) = ax + K \in Z_p[x]$를 이용한 $(2,n)-SS$을 이용하였다면 본 논문에서는 다항식의 차수를 다양하게 두어 1차 다항식부터 $\lceil n/2 \rceil$ 차 다항식까지 이용한 $(2,n)-SS$, $(3,n)-SS$, ... , $(\lceil n/2 \rceil, n)-SS$을 이용한다.

먼저 소수 p는 전체 사용자에게 공유하고 $N = \{u_1,u_2,...,u_n\}$에 속하는 n개의 사용자를 Z_p 상의 (서로 다른) 값들로 표현된다고 가정하자. 송신자는 1차 다항식부터 $\lceil n/2 \rceil$ 차 다항식 $f_1(x), f_2(x), ..., f_{\lceil n/2 \rceil}(x)$을 각각 하나씩 선택한다. 이때 각 다항식의 계수와 상수항은 Z_p에서 랜덤하게 선택한다. 다항식이 선택되면 각각의 사용자 u_i는 비밀정보 $d_{u_i} = \{f_1(u_i), f_2(u_i), ..., f_{\lceil n/2 \rceil}(u_i), K_{u_i}, K_{all}\}$을 받는다. 이때 K_{u_i}는 $N \setminus R$에 속하는 수신자수가 탈퇴자수 $|R| = r$ 보다 적을 때 사용되는 key이고 K_{all}는 탈퇴자 집합에 속하는 사용자가 없을 때($R = \varnothing$) 사용되는 key이다.

다음으로 탈퇴자를 제외하는 복호화 과정을 설명하기 위해, 임의의 사용자 r명 $R = \{u_1,u_2,...,u_r\}$이 탈퇴되었다고 가정하자. $r \le \lceil n/2 \rceil$ 일 때 송신자는 r차 다항식 $f_r(x)$에 각 탈퇴자들이 대응되는 값 $f_r(u_1), f_r(u_2), ... f_r(u_r)$을 전송하고 $f_r(x)$의 상수항 k_r을 비밀키로 사용한다. 즉 암호문을 복호화하기 위해서는 k_r을 복구해야 한다. 각 사용자는 암호문을 복호화하기 위해 자신들이 보유한 비밀정보 중 f_r에 대응하는 값과 전송된 $f_r(u_1), f_r(u_2), ... f_r(u_r)$을 이용하여 $(r+1,n)-SS$을 풀 수 있고, 상수항 k_r을 복구하게 된다. 그러나 탈퇴자들은 자신이 보유한 f_r에 대응되는 값이 암호문 헤더에 전송된 값에 포함되기 때문에 $(r+1,n)-SS$을 풀 수가 없게 된다. 이후에 탈퇴자가 추가적으로 발생하면 탈퇴자집합에 $R \leftarrow R \cap \{w_{r+1}\}$로 탈퇴자가 추가되고 같은 방법으로 탈퇴자들을 제외할 수 있다. 이 방법의 경우 암호문 헤더에 탈퇴자 r명에 대해 r개의 함수 값만

포함하면 됨으로 타 기법에 비해 효율적이다. 그러나 각 사용자는 사전에 $f_1(x), f_2(x), ..., f_{\lceil n/2 \rceil}(x)$의 $\lceil n/2 \rceil$개의 다항식에 대응되는 값과 추가적인 2개의 key까지 $\lceil n/2 \rceil$개의 비밀값을 저장하여야 하고 메시지 복호화를 위해 r명의 탈퇴자에 대해 $(r+1, n)-SS$을 푸는 연산을 해야 함으로 $O(r)$의 모듈러 연산이 수행되어야한다. 결과적으로 r의 전송량, $O(n)$의 비밀키 저장량, $O(r)$의 복호화 연산량을 보이게 된다. 전송량측면에서는 효율적이지만 비밀키 저장량과 복호화 연산량은 비효율적인 문제점이 있다.

2. 그룹설정을 통한 tradeoff의 아이디어

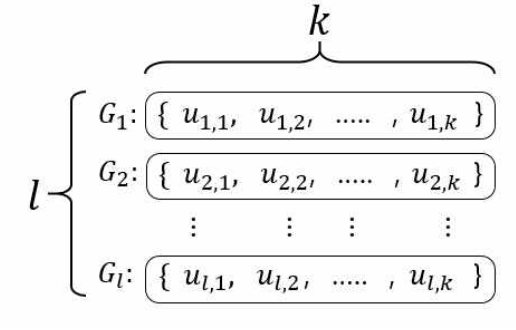

그림 1. 그룹 분할 예시
Fig. 1. Group division example

3장 1절에서 설명한 비밀분산을 이용한 방법은 효율적인 전송량을 가졌지만 비밀키 저장량과 복호화 연산량에서 비효율적 이였다. 이러한 문제점을 개선하고자 사용자들을 일정크기의 그룹으로 분할하여 각 그룹 내에서 비밀분산을 활용하는 방법을 보인다.

먼저 n명의 사용자를 $n=k\times l$이 되도록 k명씩 그룹화하여 l개의 그룹$(G_1, G_2, ..., G_l)$으로 분할하면 각 그룹 G_i는 k명의 사용자 $G_i = \{u_{i,1}, u_{i,2}, ..., u_{i,k}\}$가 포함되어 N에 속하는 전체 사용자를 i번째 그룹 G_i의 j번째 사용자 $u_{i,j}$으로 표현할 수 있다. 그룹이 설정되면 송신자는 각 그룹 G_i마다 1차 다항식부터 $\lceil k/2 \rceil$차 다항식까지 $\lceil k/2 \rceil$개의 다항식 $f_{i,1}(x), f_{i,2}(x), ..., f_{i,\lceil k/2 \rceil}(x)$ 을 하나씩 선택한다. 마찬가지로 각 다항식의 계수와 상수항은 Z_p에서 랜덤하게 선택한다. G_i에 속하는 각각의 사용자 $u_{i,j}$는 비밀정보 $d_{u_{i,j}} = \{f_{i,1}(u_{i,j}), f_{i,2}(u_{i,j}), ..., f_{i,k-2}(u_{i,j}), K_{u_{i,j}}, K_{i,all}\}$를 받는다. 이때 $K_{u_{i,j}}$는 그룹내 탈퇴자집합 $G_i \cap R = R_i$를 제외한 $G_i \setminus R_i$에 속하는 정당한 사용자수 n_i가 그룹 내 탈퇴자 수 $|R_i| = r_i$보다 적을 때 사용되는 key이고 $K_{i,all}$는 그룹 G_i에 탈퇴자 집합에 속하는 사용자가 없을 때$(R_i = \varnothing)$ 사용되는 key이다.

다음으로 탈퇴자를 제외하는 복호화 과정을 설명하기 위해, 임의의 사용자 r명이 탈퇴되어 그룹 G_i에 r_i명의 탈퇴자 $G_i \cap R = \{u_{i,1}, u_{i,2}, ..., u_{i,r_i}\}$가 포함된다고 가정하자. 송신자는 그룹 G_i에 대하여 암호문 헤더에 r_i차 다항식 $f_{i,r_i}(x)$에 각 탈퇴자들이 대응되는 r_i개의 함수값 $f_{i,r_i}(u_{i,1})$, $f_{i,r_i}(u_{i,2}), ... f_{i,r_i}(u_{i,r_i})$을 전송하고 $f_{i,r_i}(x)$의 상수항 k_{i,r_i}을 G_i의 그룹키 K_i로 사용한다. 즉 암호문을 복호화하려면 k_{i,r_i}을 복구해야 한다. 그룹 G_i에 속하는 사용자는 자신들이 보유한 비밀정보 중 f_{i,r_i}에 대응하는 값과 전송된 $f_{i,r_i}(u_{i,1}), f_{i,r_i}(u_{i,2}), ... f_{i,r_i}(u_{i,r_i})$을 이용하여 라그랑주 보간법으로 $(r_i+1, n)-SS$을 풀 수 있고, 상수항 k_{i,r_i}을 복구하게 된다. 그러나 탈퇴자들은 자신이 보유한 f_{i,r_i}에 대응되는 값이 암호문 헤더에 전송된 값에 포함되기 때문에 $(r_i+1, n)-SS$ 을 풀 수가 없게 된다. 이후에 $G_i \cap R$에 탈퇴자가 추가되어도 같은 방식으로 제외시킬 수 있으며 탈퇴자가 소속하지 않는 그룹$(G_i \cap R = \varnothing)$의 경우 $K_{i,all}$을 그룹키 K_i로 설정하고 그룹내 탈퇴자수가 절반을 넘어가는 $r_i > k/2$인 상황에서는 권한 있는 사용자가 보유한 각각의 개인키 $K_{u_{i,j}}$을 메시지 전송에 이용한다. 이러한 그룹화를 통해 각 사용자는 그룹원 수 k에 대해 $\lceil k/2 \rceil$개의 다항식값과 2개의 비밀값까지 총 $\lceil k/2 \rceil$개의 비밀정보를 저장해야하고 복호화 연산량은 그룹내에 속하는 탈퇴자수 r_i에 따라 $(r_i+1, n)-SS$의 연산을 필요로 한다. 비밀분산에 사용되는 다항식의 차수가 최대 $\lceil k/2 \rceil$임으로 최악의 경우에도 $O(k)$의 연산을 필요로 하게 된다. 그러나 암호문 헤더 전송량 측면에서는 메시지 암호화용 세션키 K를 l개의 그룹 각각에 대해 그룹키로 암호화한 $E_{K_i}(K)$을 암호문 헤더에 포함시켜야 하며 각 그룹마다 탈퇴자수 r_i명에 대해

($r_i < \lceil k/2 \rceil$ 일 때) r_i의 함수값을 전송해야한다. R에 속하는 전체 탈퇴자는 $\sum_{i=1}^{l} r_i = r$임으로 총 r개의 함수값과 l개의 세션키 암호문이 전송되어야한다. 함수값의 길이와 암호문의 길이가 동일하다고 가정하면 $r+l$의 암호문 헤더 전송량을 필요로 하게 된다. N에 속하는 전체사용자수 n에 대해 $n = k \times l$이 성립함으로 그룹의 수 l과 그룹원수 k가 서로 반비례하게 된다. 결과적으로 비밀키저장량과 복호화 연산량은 k에 의존하고 암호문 헤더전송량은 l에 의존함에 따라 tradeoff 관계를 가지게 된다. 간단하게 $\sqrt{n} = k = l$이라고 했을 때 $r+\sqrt{n}$의 암호문 헤더 전송량, $O(\sqrt{n})$의 비밀키저장량, $O(\sqrt{n})$의 복호화 연산량을 필요로 하게 된다.

Ⅳ. 제안하는 브로드캐스트 암호시스템

4장에서는 3장 2절의 그룹설정의 아이디어를 이용하여 전체사용자수 n에 대해 그룹의 수 l과 그룹원수 k를 $\sqrt{n} = k = l$으로 고정하고 구체적인 브로드캐스트 암호 시스템의 알고리즘에 대하여 기술한다. 먼저 설명의 편의를 위해 다음을 가정한다. 1) 안전성 파라미터 λ는 전체 사용자가 공유한다. 2) 메시지 암호키를 암호화하기 위한 대칭키 암호시스템 $SKE=(E, D)$와 메시지를 암호화하기 위한 대칭키 암호시스템 $\ddot{SKE}=(\ddot{E}, \ddot{D})$는 전체 사용자가 공유한다. 3) (AES-128 대칭키 암호를 사용하기 위해) 128 비트의 소수(prime) p는 전체 사용자가 공유한다. 4) 전체 사용자 수는 $n = 2^d$ (d는 2의 배수)이라 하자. 5) n명의 사용자는 Z_p 상의 원소에 고유하게 대응된다. 6) 각 사용자는 i번째 그룹 G_i에 속하는 j번째 사용자 $u_{i,j}$로 표현될 수 있다.

1. 초기 설정(Setup)

(1) n명의 사용자를 $\sqrt{n}$명의 사용자를 포함하는 $\sqrt{n}$개의 그룹 $G_1, G_2, ..., G_{\sqrt{n}}$ 으로 분할한다.

(2) 분할된 각 그룹 G_i ($i = 1,2,..., \sqrt{n}$)에 대하여 1 부터 $\sqrt{n}/2$차 다항식까지 $\sqrt{n}/2 = h$개의 다항식 $f_{i,1}(x), f_{i,2}(x), ..., f_{i,h}(x)$을 각각 하나씩 선택한다. 여기서 다항식의 계수와 상수항은 Z_p에서 랜덤하게 선택한다.

(3) G_i에 속하는 각 사용자 $u_{i,j}$에 대하여 함수값 $f_{i,1}(u_{i,j}), f_{i,2}(u_{i,j}), ..., f_{i,h}(u_{i,j})$를 계산한다.

(4) $u_{i,j}$에게 부여되는 비밀값 $d_{u_{i,j}}$는 $d_{u_{i,j}} = \{f_{i,1}(u_{i,j}), f_{i,2}(u_{i,j}), ..., f_{i,h}(u_{i,j}), K_{i,j}, K_{i,all}\}$로 구성된다. 여기서 $K_{i,j} \in Z_p$는 각 사용자마다 고유하게 할당되고 그룹내 탈퇴자수가 절반 이상일 때($r_i > h$) 사용되는 비밀키이고 $K_{i,all} \in Z_p$는 그룹 G_i에 탈퇴자가 없을 때 사용되는 랜덤한 비밀키이다.

(5) (2-4)의 과정을 각각 $\sqrt{n}$개의 그룹내 모든 사용자에게 적용한다.

각 사용자가 저장해야 하는 비밀키 $d_{u_{i,j}}$의 사이즈는 그룹원 수 $\sqrt{n}$에 대해 $\sqrt{n}/2$개의 다항식값과 2개의 비밀값까지 총 $(\sqrt{n}/2)+2$개의 비밀정보를 저장해야한다.

2. 암호화(Encryption)

탈퇴자 집합 R과 메시지 M을 입력받으면, 송신자는 $\ddot{E}_K : \{0,1\}^t \mapsto \{0,1\}^*$와 $E_L : \{0,1\}^l \mapsto \{0,1\}^l$를 사용하여 아래와 같이 암호문을 생성한다.

(1) 메시지 암호화용 세션키 K를 $\{0,1\}^t$에서 랜덤하게 선택한다.

(2) 설 각 그룹 G_i에 대해서 그룹내 탈퇴자수가 절반 이하($r_i \le h = \sqrt{n}/2$)인 경우 G_i를 S_1집합에 포함시키고 그 외 $r_i > h = \sqrt{n}/2$ 경우에 대해서는 G_i를 S_2에 포함시킨다.

(3) S_1에 속하는 각 그룹 G_i에 대하여 r_i차 다항식 $f_{i,r_i}(x)$의 상수항 k_{i,r_i}를 G_i의 그룹키 K_i로 하고 $f_{i,r_i}(x)$에 각 탈퇴자들이 대응되는 r_i개의 함수값

$f_{i,r_i}(u_{i,1}), f_{i,r_i}(u_{i,2}), \ldots f_{i,r_i}(u_{i,r_i})$을 구한다. 탈퇴자가 없는 그룹($r_i = 0$)의 경우 $K_{i,all}$를 그룹키 K_i로 한다.

(4) 메시지 암호화키 K를 그룹키 K_i로 각각 암호화하고 각 G_i마다 구한 함수값 $f_{i,r_i}(u_{i,1}), f_{i,r_i}(u_{i,2}), \ldots, f_{i,r_i}(u_{i,r_i})$과 암호문 $E_{K_i}(K)$을 암호문에 포함시킨다.

(5) S_2에 속하는 각 그룹 G_i에 그룹에 대해서는 $G_i \setminus R_i$에 속하는 남아있는 권한 있는 사용자들이 보유한 개인키 $K_{i,j}$를 사용한다. 메시지 암호화키 K를 $K_{i,j}$로 각각 암호화한 $E_{K_{i,j}}(K)$를 암호문에 포함시킨다.

(6) 메시지 M을 메시지 암호키 K로 암호화한 $\ddot{E}_K(M)$를 암호문에 포함시킨다.

(7) 각 그룹 내부의 탈퇴자 집합을 $R_1, R_2, \ldots, R_{\sqrt{n}}$을 암호문 헤더 앞쪽에 포함시킨다. 각 그룹의 탈퇴자 집합 R_i에는 r_i명의 탈퇴자정보가 포함되어있어 사용자는 자신이 속한 그룹의 탈퇴자들을 바로 확인할 수 있다.

(8) 최종적으로 암호문은 아래 (1)과 같이 구성된다.

3. 복호화(Decryption)

수신자 u는 아래 (1)과 같이 브로드캐스트 암호문을 수신하고 복호화 절차를 실행한다.

(1) 사용자 u는 자신이 속한 그룹 G_i 내부의 탈퇴자 집합 R_i에 포함된 r_i명의 탈퇴자를 확인한다. 사용자 u가 R_i에 포함되어있다면 복호화 과정을 중단한다.

(2) 그룹내 탈퇴자가 없는 경우($r_i = 0$) 사용자가 속한 그룹 G_i은 S_1에 속하고 $K_{i,all}$을 그룹키 K_i로 사용한다. 그룹 G_i에게 전송된 C_i를 $D_{K_i}(C_i)$로 복호화하여 메시지 암호키 K를 얻고 (7)번 과정으로 넘어간다.

(3) 그룹안의 탈퇴자 수 r_i가 $0 < r_i \le h = \sqrt{n}/2$인 경우에도 사용자가 속한 그룹 G_i은 S_1에 속하고 그룹 G_i에 대응하는 $[f_{i,r_i}(u_{i,1}), f_{i,r_i}(u_{i,2}), \ldots f_{i,r_i}(u_{i,r_i}), C_i]$를 가져온다.

(4) $f_{i,r_i}(u_{i,1}), f_{i,r_i}(u_{i,2}), \ldots f_{i,r_i}(u_{i,r_i})$와 사용자가 보유한 d_u에 포함된 함수값 $f_{i,r_i}(u)$값 까지 r_i+1개의 함수값으로 Lagrange 보간법을 이용하여 $f_{i,r_i}(x)$의 상수항 그룹키 K_i를 구한다. 탈퇴자 집합에 포함된 사용자들은 암호문헤더에 자신이 보유한 함수값이 포함됨으로 K_i를 복구할 수 없다.

(5) K_i를 복구해낸 후 $D_{K_i}(C_i)$로 복호화하여 메시지 암호키 K를 얻고 (7)번 과정으로 넘어 간다.

(6) 그룹 G_i안의 탈퇴자 수 r_i가 $r_i > h = \sqrt{n}/2$인 경우 암호문헤더의 $\{(C_{i,j})_{(j \in G_i \setminus R_i)}\}_{(i \in S_2)}$부분에서 사용자에게 할당된 $C_{i,j}$를 가져와 자신이 보유한 개인키 $K_{i,j}$을 이용하여 $D_{K_{i,j}}(C_{i,j})$로 복호화하여 메시지 암호키 K를 얻고 다음과정으로 넘어간다.

(7) $\ddot{D}_K(\tilde{C})$로 복호화하여 메시지 M을 얻는다.

$0 < r_i \le h$의 경우 r_i+1개의 함수값으로 Lagrange 보간법을 수행하는 $(r_i+1, n) - SS$을 풀어야한다. Lagrange 보간법은 함수값 개수의 제곱에 비례하는 연산을 필요로 하지만 5장에서 수신자의 연산량을 줄이는 방법을 소개한다.

Ⅴ. 기존 기법과의 비교분석

1. 복호화 효율성을 위한 변형

비교분석에 앞서 제안하는 기법의 효율성 증대를 위한 기법의 변형을 소개한다. 4장에서 제안하는 기법은 복호화 연산을 위해서 Lagrange 보간법을 수행하여야 한다. 그러나 Lagrange 보간법은 함수값 개수의 제곱에 비례하는 연산을 필요로 함으로 각 사용자는 상당한 연산량 부담을 가

진다. 이러한 문제는 제안하는 기법을 일부 변형함으로써 (안전성의 변화 없이) 개선시킬 수 있다.

1.1 암호문의 변형

각 사용자 u는 $0 < r_i \le h = \sqrt{n}/2$의 상황에서 자신이 포함된 그룹 G_i의 탈퇴자 수 r_i개 만큼의 함수값을 전송받고 사용자가 보유하는 d_u에 포함된 함수 값까지 r_i+1개의 함수값으로 그룹키를 복구해야한다. Lagrange 보간법을 이용하여 r_i+1개의 함수값으로 r_i차 다항식의 상수항을 복구하는 $(r_i+1,n)-SS$ 연산은 다음과 같다.

$$f(0)=\sum_{k=1}^{r_i+1}\left(y_k\times\prod_{\substack{j=1\\(j\neq k)}}^{r_i+1}\left(\frac{0-x_j}{x_k-x_j}\right)\right)(\mathrm{mod}\,p) \tag{2}$$

위 연산대로면 함수값 개수의 제곱에 비례하는 만큼 연산을 수행하여야한다. 그러나 4장 3절의 복호화과정을 살펴보면 암호문헤더로 전송받은 r_i개의 함수값 (x_1,y_1), $(x_2,y_2),...,(x_{r_i},y_{r_i})$은 그룹내 사용자들이 전부 같은 값으로 연산을 수행하고 각 사용자 u가 보유한 함수값 (x_u,y_u)만 사용자마다 다른 값으로 계산 하게 됨으로 암호문 헤더로 전송받은 r_i개의 함수값만으로 계산되는 부분만 분리하여 송신자가 미리 계산하여 전송시킬 수 있다. 그러므로 송신자가 사용자에게 전송될 r_i개의 함수값 대신에 아래의 식을 통해 공유값 $s_{i,k}$들을 암호문 전송전에 계산할 수 있다.

$$y_k\times\prod_{\substack{j=1\\(j\neq k)}}^{r_i}\left(\frac{0-x_j}{x_k-x_j}\right)=s_{i,k}\ (\mathrm{mod}\ p)\ \text{ for }\ k=1,2\cdots r_i \tag{3}$$

그러면 암호문 헤더에 r_i개의 함수값 대신에 위 식에서 계산한 r_i개의 $s_{i,k}$들을 포함 시킬 수 있다. 함수값과 $s_{i,k}$ 둘 다 Z_p에 속한 원소임으로 암호문 헤더의 길이의 증가는 일어나지 않는다. 이제 식(2)에서 송신자가 연산하여 전송하는 부분을 $s_{i,k}$값들로 치환하면 다음과 같은 식을 얻을 수 있다.

$$\sum_{k=1}^{r_i}\left(s_{i,k}\times\left(\frac{0-x_u}{x_k-x_u}\right)\right)+y_u\times\prod_{k=1}^{r_i}\left(\frac{0-x_k}{x_u-x_k}\right)\ (\mathrm{mod}\ p) \tag{4}$$

식(2)과 비교해보면 결과적으로 사용자는 그룹키 복구를 위한 연산량을 상당부분 줄일 수 있다. 이러한 변환은 S_1에 속하는 모든 그룹에 대해 진행되고 최종적으로 암호문은 다음과 같이 변형된다.

$$\begin{aligned}CT=<&[R_1,R_2,...,R_{\sqrt{n}},\{s_{i,1},s_{i,2},...,s_{i,r_i},E_{K_i}(K)\}_{(i\in S_1)}\\&,\{\{E_{K_{i,j}}(K)\}_{(j\in G_i\setminus R_i)}\}_{(i\in S_2)}],\ddot{E}_K(M)>\end{aligned} \tag{5}$$

1.2 사용자 저장값의 변형

복호화 연산량을 좀 더 개선시키기 위해 복호화 연산에서 이루어지는 연산중 일부분을 사전에 미리 계산하여 보유하면 복호화 연산에 활용할 수 있다.

$$\frac{0-x_u}{x_k-x_u}=t_{u,k}(\mathrm{mod}\ p)\ \text{ for }\ k\in G_i\wedge k\neq u \tag{6}$$

$$\frac{0-x_k}{x_u-x_k}=t_{k,u}(\mathrm{mod}\ p)\ \text{ for }\ k\in G_i\wedge k\neq u \tag{7}$$

x_k는 자신을 제외한 그룹내 각각의 사용자들에게 고유하게 할당된 값이고 x_u는 사용자 자신에게 고유하게 할당된 값이다. 위의 두 식을 이용하여 계산된 $\sqrt{n}-1$개의 $t_{u,k}$값들과 $\sqrt{n}-1$개의 $t_{k,u}$값들을 각 사용자에게 초기설정단계에서 미리 저장하도록 하면 Lagrange 보간법에 활용할 수 있다. 식(4)에서 사용자가 추가적으로 보유하는 부분을 $t_{u,k}$, $t_{k,u}$으로 치환하면 다음과 같은 식을 얻을 수 있다.

$$\sum_{k=1}^{r_i}(s_{i,k}\times t_{u,k})+y_u\times\prod_{k=1}^{r_i}t_{k,u}\ \ (\mathrm{mod}\ p) \tag{8}$$

사용자는 위 식을 통해 효율적으로 그룹키를 복구할 수 있다.

2. 안전성 측면의 비교

제안하는 브로드캐스트 암호시스템의 안전성은 [9], [10]에서 증명한 방법과 매우 유사하게 증명할 수 있다. 메시지

암호키를 암호화하는 대칭키 암호시스템 E가 CCA1에 안전하고, 메시지를 암호화하는 대칭키 암호시스템 $\ddot{E}$ 역시 one-time CCA1에 안전하고, 다항식의 차수가 확장된 (II. 6절에서 정의된) Shamir의 비밀분산 기법이 성립한다면, 제안되는 기법은 CCA1에 안전함을 증명할 수 있다.

핵심적인 증명과정을 간단히 설명하면 다음과 같다. 공격자에게 주어지는 암호문의 차이에 따라 [9], [10]과 유사한 하이브리드 게임을 설계한다. 먼저 각 그룹키를 랜덤으로 변화시키는 게임은 secret sharing 게임의 안전성에 의해 안전성 손실 없이 이전된다. 다음 해당 그룹 내에서 메시지 암호키를 랜덤하게 바꾸는 게임은 대칭키 암호시스템 E의 안전성 게임으로 이전된다. 이 경우, 시스템 내에서 전체 가능한 메시지 암호키는 총 $2n$개 이므로 $2n$개 중 하나의 대칭키를 선택하는 $1/2n$의 안전성 손실이 발생한다. 이 과정을 전체 그룹에 대해 진행한다. 이 진행에서 등장하는 그룹의 총 개수는 각 $\sqrt{n}$개의 그룹에서 다항식 차수에 의존하므로 각 그룹마다 $\sqrt{n}/2$개가 있다. 결국 최대 그룹 개수는 $n/2$가 가능하게 된다. 마지막으로 메시지 암호키로 메시지를 암호화하는 게임은 대칭키 암호시스템 $\ddot{E}$의 one-time CCA1 안전성으로 이전된다. 종합하면 제안하는 브로드캐스트 암호시스템의 안전성은 $Adv^{BE}_{A,wCCA1} \leq n^2\, Adv^{SKE}_{B,CCA} + Adv^{\ddot{S}KE}_{B,CPA\,OT}$로 증명된다.

3. 효율성 측면의 비교

일반적으로 브로드캐스트 암호시스템의 효율성은 1) 암호문 전송량, 2) 수신자의 비밀키 저장량, 3) 복호화 시 필요한 계산량 측면에서 분석된다. Table 1는 CS, SD[2], SD^{ss}[9], $2-SD^{ss}$[10] 기법과 본 논문에서 제안한 SS 기반의 기법의 효율성을 세 가지측면에서 비교한 것이다.

3.1 암호문 헤더 전송량

제안하는 기법의 암호화 알고리즘은 (R, M)을 입력받아 아래와 같은 암호문을 생성한다.

$$CT = < [R_1, R_2, ..., R_{\sqrt{n}}, \{s_{i,1}, s_{i,2}, ..., s_{i,r_i}, E_{K_i}(K)\}_{(i \in S_1)}, \{\{E_{K_{i,j}}(K)\}_{(j \in G_i \setminus R_i)}\}_{(i \in S_2)}], \ddot{E}_K(M) > \quad (9)$$

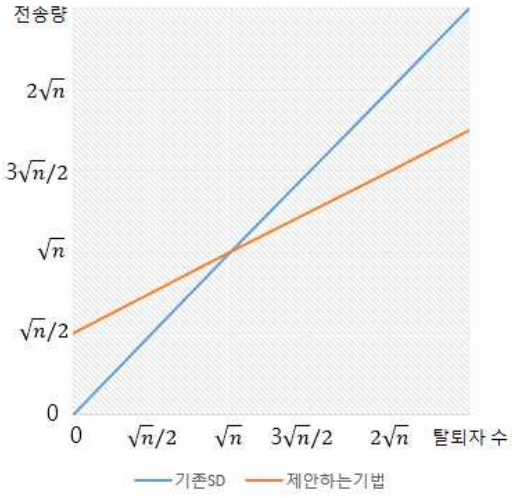

그림 2. 탈퇴자 증가에 따른 전송량 변화
Fig. 2. Changes in transmission costs due to the increasing number of revoked user

여기서 대괄호 [] 안의 값을 암호문 헤더(header)라 한다. 전송량 비교에서는 메시지 M에 대한 암호문 $\ddot{E}_K(M)$을 제외하고, 암호문의 헤더 길이만을 고려한다. 암호문 헤더를 살펴보면 각각의 그룹 내 포함된 탈퇴자 집합을 나타내는 $R_1, R_2, ..., R_{\sqrt{n}}$이 포함된다. 그리고 S_1에 소속되는 $0 < r_i \leq h = \sqrt{n}/2$경우의 그룹에 대하여 그룹마다 세션키 K의 암호문 $E_{K_i}(K)$과 추가적으로 각 그룹내 탈퇴자 수 r_i만큼 공유 값 $s_{i,k}$가 포함된다. S_2에 포함되는 그룹은 $r_i > h$의 경우이고 그룹 내 권한있는 사용자 수 $|G_i \setminus R_i| = \sqrt{n} - r_i$개 만큼 암호문 $E_{K_{i,j}}(K)$이 암호문 헤더에 들어간다. $r_i > h = \sqrt{n}/2$의 경우임으로 S_2의 경우는 항상 탈퇴자수 r_i보다 적은 암호문이 포함된다. 비밀분산에 사용되는 공유값 또는 함수값의 크기는 세션키 암호문의 길이와 동일해야 함으로 전체 탈퇴자수 r가 고정되어있다면 S_2에 속하는 그룹이 많을수록 전송량 측면에서 유리해진다. 최악의 상황을 가정하여 모든 그룹이 $r_i \leq h$의 상황으로 전부 S_1에 소속한다고 가정하면 탈퇴자수와 관계없이 항상 고정적으로 $\sqrt{n}$개의 세션키 암호문이 포함되며 전체 탈퇴자 수 $r = \sum_{i=1}^{\sqrt{n}} r_i$명에 대해 총 r개의 공유값 $s_{i,k}$가 암호문에 포함

됨을 알 수 있다. 결국 $\sqrt{n}$ 개의 세션키 암호문과 r개의 공유값이 암호문헤더에 포함된다. 결국 $\sqrt{n}+r$의 전송량을 보이게 된다. SD기법의 $2r-1$과 비교하면 그림 2에서 보이는 것과 같이 탈퇴자수가 $r<\sqrt{n}$일 때는 탈퇴자수와 상관없이 기본적으로 $\sqrt{n}$ 개의 암호문이 전송됨으로 전송량 측면에서 불리하지만 탈퇴자가 증가함에 따라 추가되는 정보가 (Worst case의 비교에서) SD기법의 절반에 해당함으로 $r>\sqrt{n}$ 가 될 때부터 전송량 측면에서 유리해진다.

3.2 저장량

각 사용자 u가 저장해야하는 비밀키 d_u에 포함되는 값은 다음과 같다.

$$d_u=\{f_{i.1}(u), f_{i.2}(u), \ ..., f_{i.h}(u), K_u, K_{i.all}\} \tag{10}$$

사용자는 그룹원 수 $\sqrt{n}$에 대하여 $h=\sqrt{n}/2$개의 다항식값과 추가적인 2개의 비밀키 까지 합하여 총 $\sqrt{n}/2+2$개의 비밀정보를 저장해야한다. 이외에도 5장 1절에서 언급한데로 복호화 연산의 효율성을 위해 비밀분산에 활용되는 값들 중 $\sqrt{n}-1$개의 $t_{u,k}$값과 $\sqrt{n}-1$개의 $t_{k,u}$값들을 사용자가 초기설정 단계에서 미리 저장해야한다. $t_{u,k}$와 $t_{k,u}$의 정보들도 Z_p에 속하는 값임으로 함수값의 길이와 동일하다. 메시지 암호화용 세션키 암호화에 128bit의 키 길이를 가지는 암호화 알고리즘을 사용한다고 가정하면 함수값의 크기도 128bit로 동일하고 $\sqrt{n}/2+2$개의 비밀정보와 추가적인 $2(\sqrt{n}-1)$개의 정보까지 합쳐 사용자가 저장해야하는 정보의 크기는 $(5/2\times\sqrt{n})\times 128$bit가된다. 이는 2^{28}의 사용자수의 환경에서 640kByte의 저장량이 요구된다.

3.3 복호화에 필요한 연산량

각 사용자 $u_{i,j}$는 소속한 그룹 G_i가 S_2에 소속하거나 또는 $r_i=0$일때 사용자가 보유하고 있는 $K_{i,j}$ 또는 $K_{i,all}$을 통해 단 한 번의 복호화 알고리즘를 수행하여 메시지 암호화용 세션키 K를 복구 할 수 있다. 반면에 그룹 G_i가 S_1에 소속하고 $r_i\neq 0$인 경우에 Lagrange 보간법을 통해 그룹키를 복구해야한다. 5장 1절에서의 변형을 통해 아래의 식을 이용하여 그룹키를 복구하게 된다.

$$\sum_{k=1}^{r_i}(s_{i,k}\times t_{u,k})+y_u\times\prod_{k=1}^{r_i}t_{k,u} \pmod p \tag{11}$$

여기서 $s_{i,k}$는 암호문을 통하여 그룹내 사용자들에게 전송된 공유값이고 $t_{u,k}$와 $t_{k,u}$는 사전에 할당된 미리 연산된 값이다. 위 식을 보면 왼쪽 항에서 r_i번의 곱셈이 이루어지고 오른쪽 항에서는 r_i+1번의 곱셈연산이 이루어지게 된다. 그러므로 그룹 키를 복구하기 위해 $O(r_i)$의 연산을 필요로 함을 알 수 있다. 위 연산은 $0<r_i\leq h$경우에서 이루어지고 $h=\sqrt{n}/2$ 임으로 최악의 경우에도 $O(\sqrt{n})$의 연산을 필요로 하게 된다.

이해를 돕기 위해 위 복호화 연산량을 RSA 복호화 지수승과 비교한다. 위 복호화에서는 최악의 경우 약 $2r_i(=\sqrt{n})$번의 곱셈 연산을 수행해야 한다. 여기서의 곱셈은 128bits 안전성을 고려할 때, 128 bits 소수 p위에서 수행된다. 곱셈 한번이 $128^2(=2^{14})$의 bit 연산을 요구하므로, $n=2^{28}$일 때, 총 $2^{14}\sqrt{n}$의 bit 연산을 수행하고, 결국 2^{28}번의 bit 연산을 수행한다. RSA 모듈러스 N에서 지수승 한번이 약 $(\log N)^3$의 bit 연산을 수행하므로, 결국 1024

표 1. 기존 기법들과의 효율성 비교
Table 1. Performance comparison to the previous broadcast encryption methods

기법	암호문 헤더 전송량	비밀키 저장량	복호화 연산량
CS [2]	$r\log\lceil n/r\rceil$	$O(\log n)$	$O(1)$
SD [2]	$2r-1$	$O(\log^2 n)$	$O(\log n)$
SD^{ss} [9]	$4r-2$	$O(\log^2 n)$	$O(1)$
2-SD^{ss} [10]	$3r$	$O(\log^3 n)$	$O(1)$
제안기법	$r+\sqrt{n}$	$O(\sqrt{n})$	$O(\sqrt{n})$

bits를 갖는 RSA 모듈러스 N에서 복호화 지수승 한번보다 작게 된다.

Ⅵ. 결 론

본 논문에서는 비밀분산기법을 활용하여 기존 PRG기반의 SD기법에 비해 전송량을 약 절반 가까이 줄일 수 있었다. 하지만 비밀키 저장량과 복호화 연산량이 $O(\sqrt{n})$로 상당히 증가하였다는 문제점이 있었다. 그러나 실제응용환경에서는 전송량 비용이 가장 중요한 요소로 간주되고 있고 최근 디바이스의 성능을 고려하면 대규모 사용자 환경을 고려해도 늘어난 저장량과 연산량은 크게 문제되지 않는다. 추가적인 단점은 안전성 모델이 [9]에서 제안한 weak CCA1로 약화되었다는 것이다. 위성 TV등의 대부분의 응용환경에서는 weak CCA1의 안정성 모델을 적용하는 것이 문제가 없다. 그러나 일부 응용환경에서는 문제가 될 가능성이 존재한다. 이후의 연구는 안전성 모델을 기존 SD기법과 동일한 CCA1을 유지하면서 SD기법의 전송량을 개선시킬 수 있느냐 하는 것이 될 것이다.

참 고 문 헌 (References)

[1] A. Fiat and M. Naor, "Broadcast encryption," Proceedings of the CRYPTO'93, volume 773 of LNCS, pp. 480-491, Aug. 1993.
[2] D. Naor, M. Naor and J. Lotspiech, "Revocation and tracing schemes for stateless receivers," Proceedings of the CRYPTO 2001, vol. 2139 of LNCS, pp. 41-62, Feb. 2001.
[3] Y. Dodis and N. Fazio, "Public key broadcast encryption for stateless receivers," Proceedings of the Digital Rights Management Workshop, vol. 2696 of Lecture Notes in Computer Science, pp. 61-80, 2002.
[4] D. Boneh, C. Gentry and B. Waters, "Collusion resistant broadcast encryption with short ciphertexts and private keys," Proceedings of the CRYPTO 2005, vol. 3621 of LNCS, pp. 258-275, Aug.2005.
[5] C. H. Kim, Y. H. Hwang and P. J. Lee, "An efficient public key trace and revoke scheme secure against adaptive chosen ciphertext attack," Proceedings of the ASIACRYPT 2003, vol. 2894 of LNCS, pp. 359-373, Nov/Dec. 2003.
[6] D. Halevy and A. Shamir, "The LSD broadcast encryption scheme," Proceedings of the CRYPTO 2002, vol. 2442 of LNCS, pp. 47-60, Aug. 2002.
[7] M.T. Goodrich, J.Z. Sun and R. Tamassia, "Efficient tree-based revocation in groups of low-state devices," Proceedings of the CRYPTO 2004, vol. 3152 of LNCS, pp. 511-527, Aug. 2004.
[8] S. Bhattacherjee and P. Sarkar, “Tree based symmetric key broadcast encryption”, IACR Cryptology ePrint Archive, Report 2013/786, 2013.
[9] J. H. Lee and J. H. Park, “Broadcast encryption system using secret sharing and subset difference methods”, Journal of Broadcast Engineering, 20(1), pp.92-109, Jan. 2015.
[10] J. H. Lee and J. H. Park, “2-Subset Difference Broadcast Encryption System Based on Secret Sharing Method”, Journal of Broadcast Engineering, 20(4), pp.93-110, Jul. 2015.

저 자 소 개

이 재 환

- 2009년 3월 ~ 2016년 2월 : 상명대학교 컴퓨터과학과 졸업
- 주관심분야 : 브로드캐스트 암호, 전자서명 등

저 자 소 개

박 종 환

- 1999년 2월 : 고려대학교 이과대학 수학과 (학사)
- 2004년 2월 : 고려대학교 정보보호대학원 정보보호학과 (석사)
- 2008년 8월 : 고려대학교 정보보호대학원 정보보호학과 (박사)
- 2009년 6월 ~ 2011년 5월 : 경희대학교(국제) 응용과학대학 학술연구교수
- 2011년 6월 ~ 2013년 8월 : 고려대학교 BK21정보보호사업단 연구교수
- 2013년 9월 ~ 현재 : 상명대학교 컴퓨터과학과 조교수
- ORCID : http://orcid.org/0000-0003-2742-6119
- 주관심분야 : 인증암호, ID-based 암호, 브로드캐스트 암호, 암호프로토콜 등