정규논문 (Regular Paper)
방송공학회논문지 제20권 제4호, 2015년 7월 (JBE Vol. 20, No. 4, July 2015)
http://dx.doi.org/10.5909/JBE.2015.20.4.580
ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

비밀분산 기반의 2-Subset Difference 브로드캐스트 암호시스템

이 재 환[a], 박 종 환[a)‡]

2-Subset Difference 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 안전하게 메시지를 전송하는 기법이다. 제안된 브로드캐스트 암호시스템 중 가장 효율적인 것은 트리구조에서 Subset Difference(SD) 기법을 이용한 것으로, 이를 구체화하기 위해 유사난수생성기(PRG: Pseudo-Random Generator)와 비밀분산(SS: Secret Sharing) 방식을 이용한 두 가지 방법이 존재한다. 2-SD 기법은 SD 기법을 일반화하는 것으로 하나의 집합에서 두 개의 부분집합을 동시에 탈퇴시킬 수 있는 방법이다. 2-SD 기법의 장점으로는 SD 기법에 비해 전송량을 더 줄일 수 있다는 것이다. 그러나 현재까지 PRG나 SS 기반에서 2-SD 기법을 설계한 결과는 알려지지 않았다. 본 논문에서는 2014년 Jae Hwan Lee[9]등이 제시한 SS 기반의 SD 기법을 확장하여 SS 기반의 2-SD 기법을 설계한다. 제안된 기법은 기존 SS 기반의 SD 기법에서 요구하는 암호문 헤더 전송량의 약 25% 줄이는 효과가 있다. 또한 암호 이론적으로 본 논문의 결과는 증명 가능한 2-SD 기법을 최초로 제시한 것이다.

Abstract

Broadcast encryption system is a cryptographic primitive that enables a sender to broadcast a message to a set of receivers in a secure channel. Out of previous proposed broadcast encryption systems, the most effective is the one that uses the Subset Difference(SD) method in a binary tree structure. The SD method has been realized by two underlying approaches: Pseudo-Random Generator(PRG) and Secret Sharing(SS). 2-SD method is the generalized version of the SD method by which two subsets of revoked receivers can be dealt with by one subset (in an SD-based broadcast encryption system). The primary advantage of the 2-SD method is to further reduce the size of transmission overhead, compared to the SD method. Until now, however, there is no known broadcast encryption system that is based on such a 2-SD technique using either PRG or SS basis. In this paper, we suggest a new 2-SD broadcast encryption system using the SS-based technique that was suggested by Jae Hwan Lee et al. in 2014[9]. The new system can reduce the size of ciphertext by 25% of the one in the previous SS-based broadcast encryption system. Also, on a theoretical note, ours is the first 2-SD broadcast encryption system that is provably secure.

Keyword : broadcast encryption, subset difference, 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 한국연구재단의 지원을 받아 수행된 기초연구사업임 (NRF-2014R1A1A2059802).
· Manuscript received April 14, 2015; revised June 24, 2015; accepted June 24, 2015.

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. 이러한 접근 제어 기능은 위성 TV, PPV (pay-per-view TV), DRM(Digital Right Management) 등의 실제 시스템에 적용되고 있다.

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 사용자(예를 들어 10^6~10^8명)를 대상으로 하므로 암호문의 전송량이 전체 네트워크 트래픽에 주는 영향이 크기 때문이다. 물론 전송량을 줄이더라도 나머지 저장량과 연산량이 적당한 범위 내에서 용인되어야 한다. 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. 이 기법은 전체 사용자 수 n, 탈퇴자 수를 r이라 할 때 $O(r)$ 전송량, $O(\log^2 n)$ 저장량, 그리고 $O(\log n)$ 연산량을 갖는다. SD 기법은 사용자의 비밀키와 Subset에 따른 그룹키를 생성하기 위해 유사난수생성기(PRG: Pseudo-random generator)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]이 Shamir의 $(2, n)$-비밀분산(Secret Sharing) 기법을 이용하여 키 분배를 하는 SD 기법(이하에서는 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.

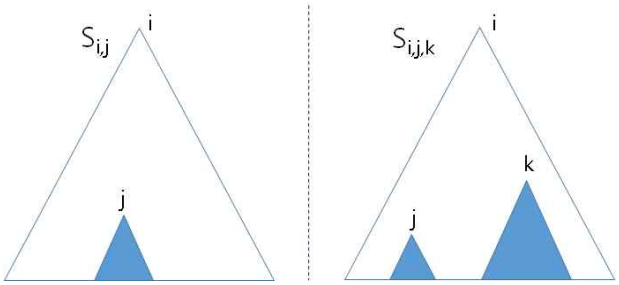

그림 1. SD 기법과 2-SD 기법에서의 부분집합 표현
Fig. 1. Subset representation in SD(left) and 2-SD(right) methods

SD 기법에서는 탈퇴자 집합이 [그림 1]의 왼쪽처럼 $S_{i,j}$ 형태의 부분집합들로 분할된다. 여기서 $S_{i,j}$ 의미는 i를 root로 하는 subtree에서 j를 root로 하는 subtree를 제외한 집합이 수신자들이란 것이다. 이를 일반화하면 [그림 1]의 오른쪽처럼 $S_{i,j,k}$ 형태의 부분집합들로 분할할 수 있다. 비슷하게 $S_{i,j,k}$ 의미는 i를 root로 하는 subtree에서 j와 k를 root로 하는 두 개의 subtree를 제외한 집합이 수신자임을 나타낸다. 2-SD 기법에서는 하나의 부분집합으로 두 개의 subtree를 처리하므로 직관적으로도 SD 기법에 비해서 전송량을 거의 절반으로 줄일 수 있을 것이라고 예상된다.

2006년 장지용 등[10]은 Naor 등이 제시한 SD^{PRG} 기법을 변형하여 2-SD 기법을 제안하였다. 그 2-SD 기법은 SD^{PRG} 기법에 비해 저장량과 연산량을 거의 희생시키지 않고 전송량을 1/2로 줄일 수 있다는 놀라운 결과를 제시하였다. 그러나 2014년 이재환 등[11]은 장지용 등이 제안한 2-SD 기법이 브로드캐스트 암호시스템 안전성 모델에서 기본적인 공격인 공모공격(collusion attack)에 전혀 안전하지 않다는 것을 보였다. 최근 Bhattacherjee 등[8]은 2-SD 기법과는 다른 접근방법으로, 이진트리를 k진트리로 확장하여

PRG 기반의 SD 기법을 일반화하는 방법을 소개하였다. [8]에서는 동일한 부모 아래 오직 같은 레벨에 있는 k개의 자손 중 임의의 두 개 자손이 탈퇴되는 경우를 처리한다는 점에서 2-SD 기법과 비슷하다. 그러나 2-SD 기법은 (k진트리로 확장하더라도) 서로 다른 레벨에 있는 자손 중 임의의 두 개 자손을 탈퇴시킬 수 있다는 점에서 [8]에서 제안한 기법보다 더욱 일반적이다. 더구나 [8]은 worst case에 SD^{PRG} 기법보다 전송량을 줄이는 효과를 전혀 내지 못하고 있다.

현재까지 2-SD 기법을 구현한 브로드캐스트 암호시스템은 제안되지 않고 있다. 본 논문에서는 [9]에서 소개된 비밀분산 방식의 키 분배 아이디어를 확장하여 최초로 2-SD 기법을 설계하고자 한다. 새롭게 제안되는 2-SD 기법(이하에서는 2-SD^{ss} 기법으로 표기함)은 $(2,n)$-비밀분산과 추가적으로 $(3,n)$-비밀분산 방식을 도입하고, $(3,n)$-비밀분산을 이진트리의 서로 다른 레벨에 적용하여 키를 분배하는 방식을 도입한다. 2-SD^{ss} 기법은 $O(r)$ 전송량, $O(\log^3 n)$ 비밀키 저장량, $O(1)$ 복호화 연산량을 갖는다. SD^{ss} 기법과 비교하면 비밀키 저장량은 증가하지만, SD^{ss} 기법에서 요구하는 전송량의 약 1/4을 줄일 수 있는 효과가 있다. 또한 SD^{ss} 기법과 마찬가지로 전체 사용자 수나 탈퇴자 수에 관계없이 항상 일정한 $O(1)$ 복호화 연산량을 제공한다. 예를 들어 SD^{PRG} 기법과 전송량을 비교하면, 전체 사용자수를 $n=2^{31}$, 탈퇴자 수를 $r=2^{12}$이라 할 때, SD^{PRG} : SD^{ss} : 2-SD^{ss} = 1 : 1.7 : 1.25의 비율을 보인다. [표 1]에서는 안전성과 효율성 측면에서 SD^{PRG}, SD^{ss}, 2-SD^{ss} 기법들을 비교한다. 암호 이론적으로는 기존 PRG 기반에서 설계할 수 없었던 2-SD 기법이 비밀분산 기반에서 설계가능하다는 것을 보이는 최초의 결과이다.

표 1. 기존 SD^{PRG}, SD^{SS} 기법과 새로운 2-SD^{SS} 기법의 비교
Table 1. Comparison between the previous SD^{PRG}, SD^{SS} schemes and our new 2-SD^{SS} scheme

기법	암호문 전송량	비밀키 저장량	복호화 연산량	안전성 손실	안전성 모델
SD^{PRG} [2]	$O(r)$	$O(\log^2 n)$	$O(\log n)$	$O(mw_1^2)$	CCA1
SD^{ss} [9]	$O(r)$	$O(\log^2 n)$	$O(1)$	$O(mw_2)$	weak CCA1
2-SD^{ss}	$O(r)$	$O(\log^3 n)$	$O(1)$	$O(mw_3)$	weak CCA1

m : Subset Cover를 구성하는 부분집합의 최대 개수, n : 전체 사용자 수, r : 탈퇴자 수,
$\{w_1, w_2, w_3\}$: 각 기법의 setup 과정에서 생성되는 전체 그룹키의 개수, weak CCA1 : [9]에서 제시된 안전성 모델로 이 모델에선 탈퇴자=배제자가 된다.

Ⅱ. 브로드캐스트 암호시스템과 안전성 정의

1. 브로드캐스트 암호시스템

양의 정수 n은 브로드캐스트 암호시스템이 수용하는 전체 사용자 수라고 하고, N을 전체 사용자 집합이라고 하자. 즉 $N=\{1,2,...,n\}$이다. 브로드캐스트 암호시스템에 속하는 사용자는 1부터 n 가운데 하나의 수로 특정된다. R은 N의 부분집합으로서 탈퇴자들의 집합이라 하자. 브로드캐스트 암호의 목표는 전송 메시지 M을 R에 속하는 탈퇴자는 복호화할 수 없도록 하고, $N \setminus R$에 속하는 사용자는 복호화할 수 있도록 하는 것이다. 약화된 안전성 모델에서는 탈퇴자=배제자임을 상기하자.

브로드캐스트 암호시스템은 세 가지 알고리즘 - 초기설정(setup), 암호화(Encryption), 복호화(decryption) - 으로 구성되어 있다.

1. Setup(λ, n): 시스템 파라미터 λ와 전체 사용자 수 n을 입력받은 후, 시스템에 속하는 사용자 u ($u=1,...,n$)에게 비밀키 d_u를 부여한다.

2. Encryption(R, M): 탈퇴자 집합 R과 메시지 M을 입력받은 후, R에 속한 탈퇴자들이 복호화하지 못하도록 암호문 CT를 생성한다. 이 경우 CT는 R을 포함하는 것으로 본다.

3. Decryption(CT, d_u): 사용자 u는 $N \setminus R$에 속한다고 하자. 암호문 CT와 사용자의 비밀키 d_u를 입력받은 후, 암호문을 복호화하여 메시지 M을 출력한다.

2. Subset Cover 방식에 기반한 브로드캐스트 암호시스템

Subset Cover 방식은 먼저 전체 사용자 집합 N에서 탈퇴자 집합 R을 제외한 집합, 즉 $N \setminus R$을 서로 겹치지 않은 disjoint 부분집합들의 모임인 $S_{i_1},...,S_{i_m}$으로 분리된다. 즉 $N \setminus R = \bigcup_{j=1}^{m} S_{i_j}$ 가 된다. 이러한 부분집합을 찾는 알고리즘을 CoverFinding 알고리즘이라 하고, R을 입력받아 부분집합을 출력하는 것을 $CoverFinding(R) = \{S_{i_1}, \ldots, S_{i_m}\}$으로 표현하자. 각각의 부분집합 S_j들은 그에 대응하는 그룹키 L_j가 할당되고, S_j안의 모든 수신자는 L_j를 유도해 낼 수 있다. 그리고 메시지를 암호화하기 위한 (일회용) 키 K는 $S_{i_1},...,S_{i_m}$들에 대응하는 그룹키 $L_{i_1},...,L_{i_m}$ 들로 암호화되고, 전송하고자 하는 메시지는 K를 이용하여 암호화한다. 보다 구체적으로 설명하면, Subset Cover 방식 하에서는 두 가지 종류의 대칭키 암호시스템이 사용된다.

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

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

2.1 초기설정(Setup)

모든 수신자 u는 비밀키 d_u를 받는다. S_i에 속한 모든 수신자 u는 자신의 비밀키 d_u를 이용하여 S_i에 대응되는 그룹키 L_i를 유도할 수 있다. 여기서 그룹키 L_i는 각각의 그룹 별로 독립적이고 랜덤한 값으로 할당하거나, 유사난수 값을 이용하여 할당할 수도 있다.

2.2 암호화(Encryption)

1. 메시지 암호키 K를 랜덤하게 선택한다.
2. 주어진 탈퇴자 집합 R에 대해 알고리즘 $Cover\ Finding(R)$을 수행하여 부분집합들 $S_{i_1},...,S_{i_m}$를 구하고, 각각의 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.

$$CT = < [i_1, i_2, \ldots, i_m, E_{L_{i_1}}(K), E_{L_{i_2}}(K), \ldots, E_{L_{i_m}}(K)], \ddot{E}_K(M) > \quad (1)$$

2.3 복호화(Decryption)

수신자 u는 다음과 같은 암호문을 수신하면, 자신의 비밀키 d_u를 이용하여 복호화 절차를 실행한다.

$$CT = < [i_1, i_2, \ldots, i_m, C_1, C_2, \ldots, C_m], C > \quad (2)$$

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 논문에서 고려하는 브로드캐스트 암호시스템의 안전성 모델은 [9]에서 제안하였고, 약화된 선택 암호문 중간공격(weak CCA1)이다. 선택 암호문 중간공격(CCA1: Chosen ciphertext and launch-time attack)[2]은 대부분의 브로드캐스트 응용환경에서 충분하다고 간주되는 공격이다. 이 공격에서는 정당한 수신자의 집합 $S = N \setminus R$에 암호문을 전송하는 경우, 집합 R에 속하는 탈퇴자들이 자신들의 비밀키를 이용하여 공모공격(collusion attack)을 하더라도 암호문에 대응하는 평문의 내용을 알 수 없어야 한다. 또한 선택 암호문 공격이므로 공격자는 평문에 대응하는 암호문을 수집할 수 있는 능력뿐만 아니라, 암호문에 대응하는 평문을 수집할 수 있는 능력까지 주어진다. 단,

weak CCA1 모델에서는 암호 질의 및 복호 질의에 사용된 탈퇴자들은 이후의 공격에서도 계속해서 탈퇴된 자들의 집합에 포함되어야 한다는 것이 일반적인 CCA1과 다른 점이다. 부연하면 R에 속한 사용자들은 (시스템이 유지되는 한) 계속해서 탈퇴자로 남고, 메시지 전송 시 정당한 수신자 집합은 $N \setminus R$로 결정되어야 한다. 따라서 수신자 집합 S를 $N \setminus R$에 속하는 부분집합으로서 동적으로 선택하는 것은 고려되지 않는다. 이러한 weak CCA1 안전성은 공격자 A와 챌린저 B사이의 (다음과 같은) 게임으로 정의된다.

- Setup. B가 Setup(λ,n) 알고리즘을 수행하여 사용자 u ($u \in U$) 각각에 대한 비밀정보를 생성한다.

- Adversarial Action. B는 초기 탈퇴자 그룹 R을 $\varnothing$로 설정한다. A는 다음의 세 가지 질의를 할 수 있다. (1) A가 사용자 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 경우에는 Encrypt(R^*, M^*)의 결과를 암호문으로서 A에게 준다. b=0인 경우는 M^*와 같은 길이를 갖는 랜덤 메시지 R_M를 선택하여 Encrypt(R^*, R_M)의 결과를 암호문으로서 A에게 준다.

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

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

각각의 부분집합에 대응하는 그룹키 하에서 메시지 암호키 K를 암호화하기 위해 필요한 대칭키 암호시스템 $SKE=(E, D)$의 안전성을 정의한다. CCA1 공격에 안전한 브로드캐스트 암호시스템을 설계하기 위해서는 SKE 역시 CCA1 공격에 안전해야 된다. 대칭키 암호의 CCA1 안전성은 공격자 A와 챌린저 B사이의 (다음과 같은) 게임으로 정의된다.

- Setup. B가 SKE의 비밀키 공간에서 랜덤한 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 $CGuess$로 하자. 안전성 상수 λ에 대해 A의 advantage는 $Adv^{SKE}_{A,CCA1}(\lambda) = |\Pr[CGuess] - 1/2|$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 암호키 K가 결정되면 메시지 M은 다른 대칭키 암호시스템 $\ddot{SKE}=(\ddot{E}, \ddot{D})$을 이용하여 암호화된다. CCA1 공격에 안전한 브로드캐스트 암호시스템을 설계하기 위해서는 $\ddot{SKE}$가 one-time 선택 평문 공격(CAP: chosen-plaintext attack)에 안전해도 충분하다. 대칭키 암호의 one-time CPA 안전성은 공격자 A와 챌린저 B사이의 (다음과 같은) 게임으로 정의된다.

- Setup. B가 SKE의 비밀키 공간에서 랜덤한 비밀키 K을 생성한다.
- Challenge. A는 메시지 m^*을 B에게 보낸다. B는 랜덤한 bit $b\in\{0,1\}$을 선택한다. b=1인 경우에는 $E_K(m^*)$를 A에게 준다. b=0인 경우는 m^*와 같은 길이를 갖는 랜덤 메시지 R_M를 선택하여 $E_K(R_M)$를 A에게 준다.
- Guess. A는 추측한 $b'\in\{0,1\}$를 내보낸다.

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

정의 3. 대칭키 암호시스템이 one-time CPA 공격 환경에서 공격자 A가 가지는 $Adv^{\ddot{SKE}}_{A,CPA^{OT}}(\lambda)$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 기반 $(2,n)$-SS과 $(3,n)$-SS기법이다. p를 다항식 설정을 위한 소수(prime)라 하자. $(2,n)$-SS의 경우, 1차 다항식 $f(x)=ax+K\in Z_p[x]$를 선택한다. 여기서 $a\in Z_p$는 랜덤하게 선택된 값이고, 상수항 $K\in Z_p$는 공유하고자 하는 비밀값이라 하자. 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 2개 이상의 값을 알면 복구할 수 있다. 여기서 K를 효율적으로 복구하기 위해 Lagrange 보간법을 사용할 수 있다. $(3,n)$-SS 기법에서는 2차 다항식 $g(x)=bx^2+cx+K\in Z_p[x]$를 선택한다. 여기서 $b,c\in Z_p$는 랜덤하게 선택된 값이고, 상수항 $K\in Z_p$는 공유하고자 하는 비밀값이다. 이 경우 n개의 share $g(x_1),..., g(x_n)$ 중 3개 이상의 값을 갖는 경우에만 비밀키 K값을 복구할 수 있고, 2개 이하의 share 값을 아는 경우에는 K에 대한 어떠한 정보도 정보 이론적으로 알 수 없다.

이러한 K의 안전성은 공격자 A와 챌린저 B사이에서 이루어지는 다음과 같은 게임으로 정의된다. B는 랜덤 bit $b\in\{0,1\}$을 선택한다. $(2,n)$-SS의 경우, b=1인 경우에는 $(f(x_i), x_i, K)$를 A에게 주고, b=0인 경우는 랜덤한 $R_K\in Z_p$를 선택하여 $(f(x_i), x_i, R_K)$를 A에게 준다. 여기서 x_i는 A가 임의로 선택한 값이다. 이 게임에서 A가 bit b를 정확하게 추측한 상황을 $CGuess$로 하자. 다항식 f는 A에게 정보 이론적으로 K에 대한 어떠한 정보도 노출하지 않는다. 따라서 $|\Pr[CGuess]-1/2|=0$이 된다. 따라서 다음과 같은 식을 얻는다.

$$|\Pr[A(f(x_i), x_i, K)=1]-\Pr[A(f(x_i), x_i, R_K)=1]|=0 \quad (3)$$

마찬가지로 $(3,n)$-SS의 경우에도, $(g(x_j), g(x_k), x_j, x_k, K)$와 $(g(x_j), g(x_k), x_j, x_k, R_K)$이 주어지는 게임을 고려할 수 있다. 여기서 x_j와 x_k는 반드시 달라야 하고, x_i, x_j, x_k는 A가 임의로 선택한 값이다. 다항식 g는 정보 이론적으로 K에 대한 어떠한 정보도 노출하지 않으므로 $|\Pr[CGuess]-1/2|=0$이 된다. 따라서 다음과 같은 식을 얻는다.

$$\begin{aligned}&|\Pr[A(g(x_j), x_j, g(x_k), x_k, K)=1]\\&-\Pr[A(g(x_j), x_j, g(x_k), x_k, R_K)=1]|=0\end{aligned} \quad (4)$$

Ⅲ. Secret Sharing을 이용한 2-SD 기법 아이디어

1. 아이디어: 2차 다항식을 이용한 Secret Sharing으로 확장

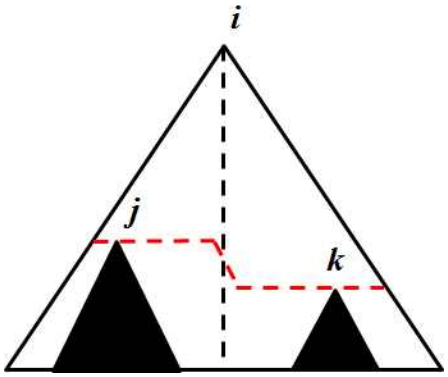

그림 2. SS 기반 2-SD 기법의 Subset 형태
Fig. 2. Subset form in 2-SD^{ss}

각 사용자는 이진트리 구조에서 leaf노드에 대응된다고 하자. 본래 Naor, Naor, Lotspiech[2]가 제안한 SD(Subset Difference) 기법은 임의의 subtree에서 그 내부에 있는 하나의 노드를 root로 하는 부분집합을 탈퇴시키는 것이었다. 그러나 2-SD 기법은 임의의 노드를 root로 하는 subtree에서 두개의 부분집합을 동시에 탈퇴시킬 수 있는 기법이다. [그림 2]에서 보면, i노드를 root로 할 때 j와 k노드를 root로 하는 두 개의 부분집합을 탈퇴시킬 수 있음을 의미한다. 2-SD 기법의 장점으로는 하나의 (i노드에 대응하는) 그룹키로 두 개의 부분집합을 처리할 수 있기 때문에, 그룹키에 대응하는 부분집합의 개수가 줄어들고 이는 결과적으로 SD 기법에 비해 암호문의 길이가 짧아진다는 것이다.

2-SD 기법이 SD 기법의 한 가지 확장된 형태로 볼 수 있으나, 2-SD 기법을 설계하는 것은 그리 간단하지 않다. 현재까지 SD 기법을 설계하는 방법으로는 유사난수생성기(PRG: Pseudo-random generator)를 이용한 것과 비밀분산(SS: Secret Sharing)을 이용한 두 가지 방법이 있다. SD 기법에 필요한 사용자의 비밀키 및 각 부분집합에 대응하는 그룹키를 생성하기 위해, PRG를 이용한 방법은 각 root노드에서 하위 노드로 향하는 경로 상에서 PRG의 일방향성을 이용하였고, SS를 이용한 방법은 각 root노드에서 하위 depth별로 독립적인 일차 다항식을 설정하고 $(2,n)$-SS를 이용하였다. 이에 비해 2-SD 기법을 설계하는 방법은 현재까지 제안되지 않고 있다. Bhattacherjee 등이 [8]에서 제시한 기법은 PRG를 이용한 매우 제한적인 형태의 2-SD 기법으로서, 삼진트리 위에서 하나의 노드 아래에 있는 동일한 레벨의 3개 자손노드 중 두 개의 노드만을 (또는 그 두 개의 노드를 root로 하는 부분집합들을) 탈퇴시키는 형태이다.

본 논문에서는 SS를 이용하여 2-SD 기법을 설계할 수 있음을 보인다. [9]에서 제시된 SD^{SS} 기법을 확장하여 2-SD 기법에 맞도록 새로운 키 분배방식과 Coverfinding 알고리즘을 제안한다. 제안된 2-SD^{SS} 기법은 전체 사용자 수를 n, 탈퇴자 수를 r이라 할 때, $O(r)$의 전송량, $O(\log^3 n)$의 비밀키 저장량, 그리고 $O(1)$의 복호화 연산량을 보인다. 이 결과는 SD^{SS} 기법이 $O(r)$의 전송량, $O(\log^2 n)$의 비밀키 저장량, 그리고 $O(1)$의 복호화 연산량을 갖는 것에 비하면, 저장량 측면에서 더 나빠진 성능을 갖는 것으로 보일 수 있다. 그러나 실제 전송량은 보안 파라미터 k에 대해 최악의 경우(worst case) 2-SD^{SS} 기법이 약 $3r \times k$ bits인데 비해, SD^{SS} 기법이 약 $4r \times k$ bits를 갖게 되어, 2-SD^{SS} 기법이 SD^{SS} 기법에 비해 약 25% 짧아진 전송량을 갖게 된다.

2-SD^{SS} 기법을 설계하는 핵심 아이디어는 이진트리 구조에서 2차 다항식 $f(x)=ax^2+bx+K \in Z_p[x]$를 이용한 $(3,n)$-SS를 도입하는 것이다. [그림 2]과 같이 i노드를 root로 하는 subtree에서 j노드와 k노드 아래의 후손들을 제외하는 경우를 생각하자. 이 경우 탈퇴자를 제외한 수신자 집합을 $S_{i,j,k}$로 표현하자. 여기서 중요한 점은 1) j노드와 k노드는 동일한 레벨에 존재하지 않아도 된다는 것, 2) j노드를 root로 하는 subtree는 i노드의 왼쪽 자손 쪽에, 그리고 k노드를 root로 하는 subtree는 i노드의 오른쪽 자손 쪽에 위치한다는 것이다. 즉 i노드를 기준으로 왼쪽과 오른쪽 자손에 각각 하나의 탈퇴자 집합이 있다.[1] $S_{i,j,k}$를 위해서는 j노드와 k노드의 후손들을 배제하는 하나의 그룹키를 생성

하는 것이 필요한데, 이를 위해 2차 다항식 $f(x)=ax^2+bx+K \in Z_p[x]$를 이용한다. 이 다항식은 [그림 2]에서 빨간색 점선으로 표시된, i노드의 왼쪽인 j노드 레벨과 i노드의 오른쪽인 k노드 레벨에 대응하는 다항식이다. 또한 상수항 K는 i노드에서 j노드와 k노드 레벨에 대응하는 그룹키로 설정된다. 각 사용자는 해당 레벨에 대응하는 노드의 함수값을 사전에 받게 되고, 암호문 생성 시 $S_{i,j,k}$를 위해서는 $[f(j), f(k), E_K(K')]$의 값이 계산된다. 여기서 K'는 메시지 암호화 키이다. 이 경우 $S_{i,j,k}$에 속하는 사용자는 자신이 받은 (j노드와 k노드에 대응되지 않는) 함수값과 $f(j)$, $f(k)$를 이용하여 그룹키 K를 복구할 수 있고, j노드와 k노드 아래의 탈퇴자들은 두 개의 함수값만 갖게 되어 K를 복구할 수 없게 된다.

위에서 기술한 아이디어는 기존 SD^{SS} 기법에서 1차 다항식을 이용한 $(2,n)$-SS을 확장한 것으로 볼 수 있다. SD^{SS} 기법이 root노드인 i노드에서 동일한 depth에 있는 레벨만을 $(2,n)$-SS로 다룬 것에 비해, 2-SD^{SS} 기법에서는 i노드 기준으로 좌우의 서로 다른 (또는 동일한) depth에 있는 레벨을 $(3,n)$-SS로 다루는 것이다. 당연히 i노드에서 동일한 depth에 있는 레벨을 ($(3,n)$-SS로) 다루는 것도 포함한다. 그러나 $S_{i,j,k}$ 부분집합은 $(3,n)$-SS을 이용하게 됨으로 $[f(j), f(k), E_K(K')]$에서 보듯, 복호화 연산을 위해 2개의 함수값을 전송해야 한다. 이는 SD^{SS} 기법이 $[f(j), E_K(K')]$처럼, $S_{i,j}$ 부분집합별로 하나의 함수값을 전송하는 것에 비해 $f(k)$와 같은 함수값을 추가로 전송하는 부담이 있다. 그러나 2-SD^{SS} 기법에서는 암호문 작성에 필요한 부분집합의 개수가 상당히 줄어들게 되어, 결과적으로 암호문 전송량이 줄어드는 효과를 가진다.

2. 2-SD^{SS} 기법에 필요한 Coverfinding 알고리즘

$S_{i,j,k}$처럼 $N \setminus R$을 disjoint한 부분집합으로 분할하기 위해서는 새로운 Coverfinding 알고리즘이 필요하게 된다. 새로운 Coverfinding 알고리즘 역시 기존 SD^{SS} 기법에서처럼 탈퇴자 집합 R을 입력받아 $N \setminus R$의 사용자들을 disjoint한 부분집합으로 분할한다. 기본적으로 새로운 Coverfinding 알고리즘이 $S_{i,j,k}$ 형태의 부분집합을 출력하지만, 특수한 경우에는 기존 SD^{SS} 기법에서처럼 $S_{i,j}$ 형태의 부분집합도 출력한다. 여기서 특수한 경우는 임의의 i노드를 root로 하는 subtree에서 i노드의 왼쪽 자식노드 ($i.lchild$로 표기) 또는 오른쪽 자식노드($i.rchild$로 표기)가 탈퇴자 집합의 노드를 가리킬 때이다. 이 경우는 $S_{i,j,k}$ 보다 $S_{i.rchild,j}$ 또는 $S_{i.lchild,j}$ 집합으로 암호문을 구성하는 것이 암호문 길이를 줄일 수 있기 때문이다.[2] 이것은 다음의 세 가지를 의미한다. 1) 사용자의 비밀키를 부여할 때, $S_{i,j,k}$ 형태에 대응하는 2차 다항식을 이용한 비밀키 뿐만 아니라 $S_{i,j}$ 형태에 대응하는 1차 다항식을 이용한 비밀키를 생성해야 한다. 2) 사용자의 비밀키를 부여할 때, i노드 기준으로 $i.lchild$ 노드 레벨과 오른쪽 편 레벨에 대응하는 2차 다항식들은 고려할 필요가 없다는 것이다. 마찬가지로 $i.rchild$ 노드 레벨과 왼쪽 편 레벨에 대응하는 2차 다항식들은 고려할 필요가 없다는 것이다.[3] 3) 2-SD^{SS} 기법의 Coverfinding 알고리즘은 $S_{i,j}$ 형태의 집합들($S1$으로 표기)과 $S_{i,j,k}$ 형태의 집합들($S2$로 표기)이 출력된다. 이를 $Coverfinding(R)=\{S1, S2\}$로 표현하기로 하자. 더 구체적으로 $S1=\{S_{i_1,j_1}, S_{i_2,j_2}, ..., S_{i_w,j_u}\}$이고 $S2=\{S_{i_1,j_1,k_1}, S_{i_2,j_2,k_2}, ..., S_{i_w,j_w,k_w}\}$의 모습으로 부분집합들을 포함하게 될 것이다.

Coverfinding 알고리즘이 탈퇴자 집합 R을 입력받게 되면 root노드와 탈퇴자에 해당하는 leaf노드들, 그리고 연결되는 내부 경로에 포함된 노드들로 구성된 Steiner Tree $ST(R)$를 구성하고, 최종적으로 탈퇴자 처리된 root노드만 남을 때까지 반복 작업을 수행하여 $N \setminus R$을 disjoint한 부분집합들로 분할하게 된다. 자세한 과정은 아래 2.1절에 유사코드로 서술되어 있다.

---

1) 만일 i노드를 기준으로 어느 한쪽 자손에만 두 개의 탈퇴자 집합이 있다면, i노드의 바로 아래 두 개의 자손노드 중 탈퇴자가 있는 쪽의 노드를 새로운 root노드로 설정하면 된다.

2) 이후에는 $i.lchild$ 또는 $i.rchild$가 아닌, i노드를 탈퇴자의 노드로 간주하고 상위 레벨로 이동하면서 Coverfinding 알고리즘을 적용한다.

3) 이 내용은 비밀키 생성하는 setup 알고리즘에서 다시 한 번 상기될 것이다.

2.1 2-SD^{SS} 기법의 Coverfinding 알고리즘 유사코드

$Coverfinding(R)$
let $T := ST(R)$
let $S1 := \varnothing$
let $S2 := \varnothing$
while($root.state \neq revocation$)
{
Find two revoked node j,k in T such that the least-common-ancestor i of j,k does not con tain any other revoked node

if $i.lchild \neq j$ and $i.rchild \neq k$ then
$S2 \leftarrow S2 \cup S_{i,j,k}$
else if $i.lchild \neq j$ and $i.rchild = k$ then
$S1 \leftarrow S1 \cup S_{i.lchild,j}$
else if $i.lchild = j$ and $i.rchild \neq k$ then
$S1 \leftarrow S1 \cup S_{i.rchild,k}$

set $i.state := revocation$
Remove nodes that are descendants of i
}
return $\{S1, S2\}$

2.2 Coverfinding 알고리즘 분석

위 Coverfinding 알고리즘에서 while 반복문을 한번 수행할 때마다 두 개의 탈퇴자노드를 처리하고 한 개의 탈퇴자노드가 새롭게 생성된다. 그러면서 1개의 부분집합이 $S1$ 또는 $S2$에 추가된다. 반복문은 root노드가 탈퇴자노드가 될 때까지 수행된다. root노드가 탈퇴자가 되려면 while문이 최소 r-1번 수행되어서 남아있는 탈퇴자수를 1개로 줄여야 한다. 그것이 root노드가 아니라면 한 번 더 수행하여 root노드를 탈퇴자 노드로 만들고 알고리즘을 종료한다. 따라서 while문은 최대 r번 수행될 수 있다. worst case로 반복문이 수행될 때마다 $S1$ 또는 $S2$에 부분집합이 1개 추가된다고 하면, r명의 탈퇴자에 대해 최대 r개의 부분집합이 출력된다.

Ⅳ. 제안하는 2-SD^{SS} 브로드캐스트 암호시스템

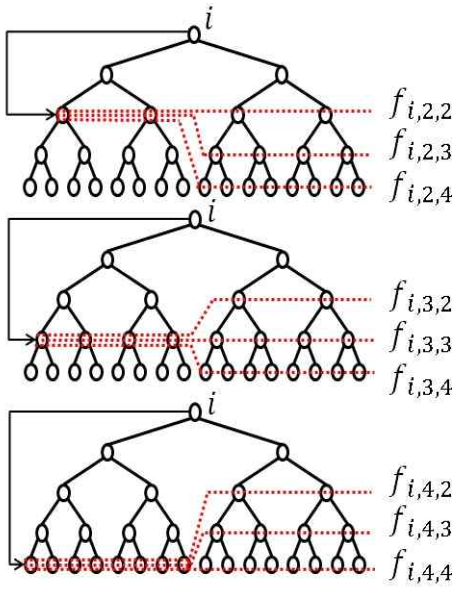

그림 3. Secret-sharing 기반 2-SD 기법의 비밀키 할당 예시
Fig. 3. Example of key assignment in 2-SD^{ss}

1. 기법 설명

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 대칭키 암호를 사용하기 위해) 128 비트의 소수(prime) p는 전체 사용자가 공유한다. 4) 전체 사용자 수는 $n = 2^k$이라 하자. 5) n명을 leaf노드에 대응시키는 완전이진트리(full binary tree)를 구성하고, 이진트리의 모든 노드들은 Z_p 상의 원소에 고유하게 대응된다.

1.1 초기설정(Setup)

1. i노드를 root로 하는 subtree에서 각각의 하위 레벨 depth l에 대응하는 1차 다항식 $f_{i,l}(x) = a_{i,l}x + k_{i,l} \in Z_p[x]$을 선택한다. 여기서 $a_{i,l}$과 $k_{i,l}$은 Z_p에서 랜덤하게 선택된다.

2. 1의 과정을 root 노드 i_0에서부터 하위 레벨의 depth가 $-1+\log n$이 될 때까지 중간단계의 모든 노드들에 대해 반복 수행한다.

3. i노드를 root로 하는 subtree에서 i노드의 왼쪽 subtree의 하위 레벨 depth l과 i노드의 오른쪽 subtree의 하위 레벨 depth m에 서로 대응되는 2차 다항식 $f_{i,l,m}(x)=a_{i,l,m}x^2+b_{i,l,m}x+k_{i,l,m}\in Z_p[x]$을 선택한다. 마찬가지로 $a_{i,l,m}$, $b_{i,l,m}$ 그리고 $k_{i,l,m}$은 Z_p에서 랜덤하게 선택된다. 여기서 l과 m의 범위는 2부터 leaf노드의 depth인 k까지다. 여기서 1이 빠지는 이유는 III.2절에서 언급한 대로 i노드의 바로 아래 자식노드에 대해서는 2차 다항식을 대응시키지 않아도 되기 때문이다. [그림 3]에서는 depth가 4인 i노드를 root로 하는 subtree에서 빨간색 점선으로 2차 다항식이 필요한 레벨의 조합을 나타내고 있다.

4. 3의 과정을 root 노드 i_0에서부터 하위 레벨의 depth가 $-2+\log n$이 될 때까지 중간단계의 모든 노드들에 대해 반복 수행한다.

5. root 노드 i_0에서 leaf 노드 사용자 u에 의해 결정되는 $Pnodes(i_0\rightarrow u)$를 구한다. 예를 들어 $Pnodes(i_0\rightarrow u)=\{i_1, ..., i_k\}$라 하자.

6. i_0에 대응되는 $(2,n)$-SS의 비밀값 $d_{i_0,u}^{degree1}=\{f_{i_0,1}(i_1), ..., f_{i_0,k}(i_k)\}$ 을 계산한다.

7. i_0에 대응되는 $(3,n)$-SS의 비밀값 $d_{i_0,u}^{degree2}=\{f_{i_0,2,2}(i_2), ..., f_{i_0,2,k}(i_k)\}$을 계산한다.
$\vdots \quad \vdots \quad \vdots$
$\{f_{i_0,k,2}(i_2), ..., f_{i_0,k,k}(i_k)\}$

8. i_0에 대응되는 비밀값은 $d_{i_0,u}=d_{i_0,u}^{degree1}\cup d_{i_0,u}^{degree2}$ 가 된다.

9. 노드 i_1부터 i_{k-1}을 subtree의 root 노드로 할 때의 비밀값 $d_{i_1,u}^{degree1}, ..., d_{i_{k-1},u}^{degree1}$ 과 $d_{i_1,u}^{degree2}, ..., d_{i_{k-2},u}^{degree2}$ 를 6-8의 과정으로 구한다.

10. 사용자 u에 부여되는 $(2,n)$-SS의 비밀값은 $d_u^{degree1}=\{d_{i_0,u}^{degree1}, d_{i_1,u}^{degree1}, ..., d_{i_{k-1},u}^{degree1}\}$가 되고 $(3,n)$-SS의 비밀값은 $d_u^{degree2}=\{d_{i_0,u}^{degree2}, d_{i_1,u}^{degree2}, ..., d_{i_{k-2},u}^{degree2}\}$가 되며 최종적으로 사용자 u가 보유하는 비밀값은 $d_u=\{k, d_u^{degree1}, d_u^{degree2}\}$로 구성된다. 여기서 $k\in Z_p$는 탈퇴자가 없는 경우에 사용되는 랜덤한 비밀키이다.

각 사용자가 저장하는 비밀값 d_u의 사이즈를 분석하면, $d_u^{degree1}$의 원소 개수는 $\log n-j$개가 되고, j는 0부터 $\log n-1$까지 포함하므로, 전체 개수는 $\sum_{d=1}^{\log n} d$ 로 구할 수 있다. $d_u^{degree2}$의 원소 개수는 $(\log n-j-1)^2$개가 되고, j는 0부터 $\log n-2$까지 포함하므로, $\sum_{d=1}^{\log n-1} d^2$로 구할 수 있다. 결과적으로 $1+\sum_{d=1}^{\log n} d+\sum_{d=1}^{\log n-1} d^2=\frac{1}{3}log^3 n-\frac{1}{3}log^2 n+\log n+1$, 즉 $O(\log^3 n)$의 비밀키 저장량을 가지게 된다.

1.2 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. 새로운 Coverfinding 알고리즘을 사용하여 탈퇴자 집합 R에 대해서 $CoverFinding(R)=\{S1,S2\}$을 구한다.

3. $S1$의 원소 각 부분집합 $S_{i,j}$에서 i노드를 시작으로 하는 subtree에서 j노드에 이르는 depth l에 정의된 함수 $f_{i,l}(x)=a_{i,l}x+k_{i,l}\in Z_p$의 상수항 $k_{i,l}$를 $S_{i,j}$의 그룹키 $K_{i,l}$로 하고, j노드에 할당된 값을 대입한 함수값

$f_{i,l}(j)$을 구한다.

4. 3의 과정을 $S1$의 원소 S_{i_1,j_1}, S_{i_2,j_2}, ..., S_{i_w,j_w}에 각각 적용한다.

5. 메시지 암호화키 K를 그룹키 $K_{i,l}$로 각각 암호화하고 각 $S_{i,j}$마다 구한 함수값 $f_{i,l}(j)$과 암호문 $E_{K_{i,l}}(K)$을 암호문에 포함시킨다.

6. $S_{i_1,j_1}, S_{i_2,j_2}, \ldots, S_{i_w,j_w}$을 특정할 수 있도록 (i_1, j_1), ... , (i_w, j_w)를 index로 암호문에 포함시킨다. 사용자는 자신이 속한 부분집합을 바로 확인하고 복호화할 수 있다.

7. $S1$에 대한 암호문 헤더를 식 (5)와 같이 구성한다.

8. $S2$의 원소 각 부분집합 $S_{i,j,k}$에서 i노드를 시작으로 하는 subtree에서 왼쪽 j노드에 이르는 depth l과 오른쪽 k노드에 이르는 depth m에 정의된 함수 $f_{i,l,m}(x) = a_{i,l,m}x^2 + b_{i,l,m}x + k_{i,l,m} \in Z_p[x]$의 상수항 $k_{i,l,m}$를 $S_{i,j,k}$의 그룹키 $K_{i,l,m}$로 하고, j노드와 k노드에 할당된 값을 대입한 2개의 함수값 $f_{i,l,m}(j)$, $f_{i,l,m}(k)$을 구한다.

9. 8의 과정을 $S2$의 원소 S_{i_1,j_1,k_1}, S_{i_2,j_2,k_2}, ..., S_{i_w,j_w,k_w}에 각각 적용한다.

10. 메시지 암호화키 K를 그룹키 $K_{i,l,m}$로 각각 암호화하고 각 $S_{i,j,k}$마다 구한 2개의 함수값 $f_{i,l,m}(j)$, $f_{i,l,m}(k)$과 암호문 $E_{K_{i,l,m}}(K)$을 암호문에 포함시킨다.

11. S_{i_1,j_1,k_1}, S_{i_2,j_2,k_2}, ..., S_{i_w,j_w,k_w}을 특정할 수 있도록 (i_1, j_1, k_1), ... , (i_w, j_w, k_w)를 index로 암호문에 포함시킨다. 사용자는 자신이 속한 부분집합을 바로 확인하고 복호화할 수 있다.

12. $S2$에 대한 암호문 헤더를 식 (6)과 같이 구성한다.

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

14. 최종적인 암호문은 $CT = <[Header^{\text{degree}1}, Header^{\text{degree}2}], \ddot{E}_K(M)>$가 된다.

전송량 측면에서 기존 SD^{SS} 기법과 비교할 때, 우선 Coverfinding 알고리즘이 달라짐으로 암호문 생성에 필요한 부분집합이 달라진다. 2-SD^{SS} 기법의 Coverfinding 알고리즘의 결과로 탈퇴자수 r에 대하여 최악의 경우 r개의 부분집합이 발생한다. 기존 SD^{SS} 기법의 $2r-1$개에 비해 부분집합 개수를 대폭 줄였지만, $S2$에 속하는 $S_{i,j,k}$ 형태의 부분집합은 $(f_{i,l,m}(j), f_{i,l,m}(k), E_{K_{i,l,m}}(K))$로 2개의 함수값을 전송하는 것을 고려해야한다.

1.3 복호화(Decryption)

수신자 u는 다음과 같은 브로드캐스트 암호문을 수신하고 복호화 절차를 실행한다.

$$< [Header^{\text{degree}1}, Header^{\text{degree}2}], \widetilde{C} > \tag{7}$$

1. 사용자 u는 암호문 헤더의 $Header^{\text{degree}1} = <(i_1, j_1), \ldots, (i_w, j_w), f_{i_1,l_1}(j_1), C_1, \ldots, f_{i_w,l_w}(j_w), C_w>$에서 index $(i_1, j_1), \ldots, (i_w, j_w)$를 통해 자신이 속한 부분집합 $S_{i,j}$를 확인한다. 자신이 속한 부분집합을 찾지 못한 경우 5과정으로 넘어간다.

$$Header^{\text{degree}1} = <(i_1, j_1), \ldots, (i_w, j_w), f_{i_1,l_1}(j_1), E_{K_{i_1,l_1}}(K), \ldots, f_{i_w,l_w}(j_w), E_{K_{i_w,l_w}}(K)> \tag{5}$$

$$Header^{\text{degree}2} = <(i_1, j_1, k_1), \ldots, (i_w, j_w, k_w), f_{i_1,l_1,m_1}(j_1), f_{i_1,l_1,m_1}(k_1), E_{K_{i_1,l_1,m_1}}(K), \ldots, f_{i_w,l_w,m_w}(j_w), f_{i_w,l_w,m_w}(k_w), E_{K_{i_w,l_w,m_w}}(K)> \tag{6}$$

2. $S_{i,j}$가 확인되면 대응하는 $[f_{i,l}(j), C]$를 가져온다.

3. $f_{i,l}(j)$와 d_u에서 (i,j)에 대응하는 함수값 $f_{i,l}(\hat{j})$값 (여기서 $j \neq \hat{j}$임)을 가지고 Lagrange 보간법을 이용하여 그룹키 $K_{i,l}$를 구한다. 참고로 $x_1 \neq x_2$인 두 개의 1차 다항식 함수값 (x_1,y_1), (x_2,y_2)를 통해 상수항인 그룹키 $f(0)=K_{i,l}$를 구하는 식은 아래와 같다.

$$f(0)=y_1\frac{(0-x_2)}{(x_1-x_2)}+y_2\frac{(0-x_1)}{(x_2-x_1)} \pmod p \quad (8)$$

4. $K_{i,j}$를 복구해낸 후 $D_{K_{i,j}}(C)$로 복호화하여 메시지 암호키 K를 얻은 후 9번 과정으로 넘어간다.

5. 사용자 u는 암호문 헤더의 $Header^{degree2}=<(i_1,j_1,k_1), \ldots,(i_w,j_w,k_w),f_{i_1,l_1,m_1}(j_1),f_{i_1,l_1,m_1}(k_1), C_1, \ldots,f_{i_w,l_w,m_w}(j_w), \ldots,f_{i_w,l_w,m_w}(j_w), f_{i_w,l_w,m_w}(k_w), C_w>$에서 index $(i_1,j_1,k_1),\ldots,(i_w,j_w,j_w)$를 통해 자신이 속한 부분집합 $S_{i,j,k}$를 확인한다. 자신이 속한 부분집합을 찾지 못한 경우 복호화 과정을 중단한다.

6. $S_{i,j,k}$가 확인되면 대응하는 $[f_{i,l,m}(j), f_{i,l,m}(k), C]$를 가져온다.

7. $f_{i,l,m}(j)$, $f_{i,l,m}(k)$와 d_u에서 (i,j,k)에 대응하는 함수값 $f_{i,l,m}(\hat{j})$값 (여기서 $j \neq \hat{j}$임)을 가지고 Lagrange 보간법을 이용하여 그룹키 $K_{i,l,m}$를 구한다. 참고로 $x_1 \neq x_2 \neq x_3$인 세 개의 2차 다항식 함수값 (x_1,y_1), (x_2,y_2), (x_3,y_3)을 통해 상수항인 그룹키 $f(0)=K_{i,l,m}$을 구하는 식은 아래와 같다.

$$\begin{aligned} f(0)=&y_1\left(\frac{(0-x_2)}{(x_1-x_2)}\right)\left(\frac{(0-x_3)}{(x_1-x_3)}\right)+\\ &y_2\left(\frac{(0-x_1)}{(x_2-x_1)}\right)\left(\frac{(0-x_3)}{(x_2-x_3)}\right)+\\ &y_3\left(\frac{(0-x_1)}{(x_3-x_1)}\right)\left(\frac{(0-x_2)}{(x_3-x_2)}\right)(\mathrm{mod}\, p) \end{aligned} \quad (9)$$

8. $K_{i,j,k}$를 복구해낸 후 $D_{K_{i,j,k}}(C)$로 복호화하여 메시지 암호키 K를 얻는다.

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

$N \diagdown R$에 속하는 권한을 가진 사용자의 경우 2-4 과정 또는 6-8 과정 둘 중 하나를 통하여 그룹키를 구할 수 있으며, 두 개의 과정 모두 복호화 시 전체 사용자 수 n이나 탈퇴자 수 r에 관계없이 간단한 modular 연산을 필요로 한다.

Ⅴ. 제안하는 2-SD^{SS} 브로드캐스트 암호시스템의 안전성 증명

정리 4. 대칭키 암호시스템이 SKE=(E, D)이 CCA1 공격에 안전하고, 대칭키 암호시스템 $S\ddot{K}E$= $(\ddot{E}, \ddot{D})$이 one-time CPA 공격에 안전하다고 가정하자. 이 경우 제안한 브로드캐스트 암호시스템은 weak CCA1 공격에 안전하다.

구체적으로 A는 브로드캐스트 암호시스템에 대한 weak CCA1 공격 알고리즘이고, B는 SKE에 대한 CCA1 공격 알고리즘 또는 $S\ddot{K}E$에 대한 one-time CPA 공격 알고리즘이라 하면,

$$Adv^{BE}_{A,wCCA1} \leq 2vw\, Adv^{SKE}_{B,CCA} + Adv^{S\ddot{K}E}_{B,CPA^{OT}} \quad (10)$$

여기서 v은 A가 선택하는 탈퇴자 집합에 의해 생성되는 부분집합의 최대 개수이고, w는 2-SD의 setup 하에서 설정되는 전체 그룹키 개수이다.

증명) 브로드캐스트 암호시스템의 weak CCA1 안전성을 공격하는 공격자를 A라 하자. 증명에서는 A를 이용하여 SKE=(E, D)의 CCA1 안전성을 공격하거나 $S\ddot{K}E$= $(\ddot{E}, \ddot{D})$의 one-time CPA 안전성을 공격하는 알고리즘 B

를 설계할 것이다. 증명의 편의상 공격자가 선택한 탈퇴자 집합 R^*에 대해 $CoverFinding(R)=\{S2\}$라고 하자.[4] A가 받는 챌린지 암호문 CT^*의 변화를 이용하여 다음 식 (11)과 같은 하이브리드 게임을 고려한다.

여기서 R_{i_1,l_1,m_1}, ..., R_{i_v,l_v,m_v}은 SKE의 비밀키 공간에서 선택한 난수들이고, R_1, ... , R_m은 SKE의 메시지 공간에서 선택한 난수들이다. R_{M^*}은 챌린지 메시지 M^*와 같은 길이를 갖는 난수 메시지이다.

각 게임 G_i에서 공격자 A가 정확하게 추측할 확률을 $\Pr[G_i]$라 하자. 게임 G_0는 메시지 M^*에 대한 암호문이 주어진 것으로 브로드캐스트 암호시스템의 weak CCA1 공격 환경이다. 반면 게임 G_F는 난수 메시지 R_{M^*}에 대한 암호문이 주어진 것으로 A는 메시지에 대한 어떠한 정보도 얻을 수 없는 공격 환경이다. 안전성 증명은 G_0에서 G_F로 전이하는 과정에서 A가 가지는 advantage가 무시할 만한 (negligible) 것임을 보이는 것으로 이루어진다.

Claim 1. $\Pr[G_0]-\Pr[G_1]=0$

증명) A는 브로드캐스트 암호시스템에 대한 weak CCA1 공격을 수행한다. B는 S_{i_1,j_1,k_1}에 대응하는 그룹키를 제외하고 setup 과정과 동일하게 비밀키를 생성한다. B는 챌린지로 받은 $(f(j_1), f(k_1), j_1, k_1, T)$ 값을 S_{i_1,j_1,k_1}에 대응하는 질의에 이용한다. (여기서 j_1, k_1은 B가 선택한 값이다.) S_i은 i을 root로 하는 subtree의 leaf 노드 사용자 집합이라 하자. 1) 사용자 u가 $u\notin S_{i_1}$인 경우, u의 비밀키 질의는 B가 (관련된 키를 알고 있으므로) 쉽게 응답할 수 있다. u가 $u\in S_{i_1}$이고 ($u\in S_{j_1}$ 또는 $u\in S_{k_1}$)인 경우, u의 비밀키 질의는 j_1 노드가 속한 레벨에 대응하는 다항식 값으로 $f(j_1)$을 이용하고 k_1 노드가 속한 레벨에 대응하는 다항식 값으로 $f(k_1)$을 이용한다. 2) u가 $u\notin S_{i_1}$인 경우, A의 암호(복호) 질의는 B가 u의 비밀키를 생성하여 대응한다. u가 $u\in S_{i_1}$이고 ($u\in S_{j_1}$ 또는 $u\in S_{k_1}$)인 경우, A의 암호(복호) 질의는 T를 그룹키로 응답한다.

A가 u의 비밀키 질의 시 $u\in S_{i_1}$이고 ($u\notin S_{j_1}$ 그리고 $u\notin S_{k_1}$)인 경우, j_1 노드와 k_1 노드 레벨에 동시에 대응하는 (계수를 모르는) 다항식 $f(\bullet)$를 이용하여 비밀키를 응답해야 한다. 이 경우 B는 $a\times(j_1)^2+b\times j_1+c=T\in Z_p$와 $a\times(k_1)^2+b\times k_1+c=T\in Z_p$를 만족하는 $a, b, c\in Z_p$를 랜덤하게 선택한 후, j_1과 k_1 노드 레벨에 대응하는 다항식을 $f(x)=ax^2+bx+c\in Z_p[x]$로 사용한다. 이러한 비밀키

$$
\begin{aligned}
G_0:\ CT^*=<&[(i_q,j_q,k_q)_{q=1}^{v},\ f_{i_1,l_1,m_1}(j_1),f_{i_1,l_1,m_1}(k_1),E_{K_{i_1,l_1,m_1}}(K),\dots,f_{i_v,l_v,m_v}(j_v),f_{i_v,l_v,m_v}(k_v),E_{K_{i_v,l_v,m_v}}(K)],\ \ddot{E}_K(M^*)>\\
G_1:\ CT^*=<&[(i_q,j_q,k_q)_{q=1}^{v},\ f_{i_1,l_1,m_1}(j_1),f_{i_1,l_1,m_1}(k_1),E_{R_{i_1,l_1,m_1}}(K),\dots,f_{i_v,l_v,m_v}(j_v),f_{i_v,l_v,m_v}(k_v),E_{K_{i_v,l_v,m_v}}(K)],\ \ddot{E}_K(M^*)>\\
G_{\hat{1}}:\ CT^*=<&[(i_q,j_q,k_q)_{q=1}^{v},\ f_{i_1,l_1,m_1}(j_1),f_{i_1,l_1,m_1}(k_1),E_{R_{i_1,l_1,m_1}}(R_1),\dots,f_{i_v,l_v,m_v}(j_v),f_{i_v,l_v,m_v}(k_v),E_{K_{i_v,l_v,m_v}}(K)],\ \ddot{E}_K(M^*)>\\
G_{\tilde{1}}:\ CT^*=<&[(i_q,j_q,k_q)_{q=1}^{v},\ f_{i_1,l_1,m_1}(j_1),f_{i_1,l_1,m_1}(k_1),E_{K_{i_1,l_1,m_1}}(R_1),\dots,f_{i_v,l_v,m_v}(j_v),f_{i_v,l_v,m_v}(k_v),E_{K_{i_v,l_v,m_v}}(K)],\ \ddot{E}_K(M^*)>\\
&\ \ \vdots\qquad\qquad\qquad\vdots\qquad\qquad\qquad\vdots\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad(11)\\
G_v:\ CT^*=<&[(i_q,j_q,k_q)_{q=1}^{v},\ f_{i_1,l_1,m_1}(j_1),f_{i_1,l_1,m_1}(k_1),E_{K_{i_1,l_1,m_1}}(R_1),\dots,f_{i_v,l_v,m_v}(j_v),f_{i_v,l_v,m_v}(k_v),E_{R_{i_v,l_v,m_v}}(K)],\ \ddot{E}_K(M^*)>\\
G_v:\ CT^*=<&[(i_q,j_q,k_q)_{q=1}^{v},\ f_{i_1,l_1,m_1}(j_1),f_{i_1,l_1,m_1}(k_1),E_{K_{i_1,l_1,m_1}}(R_1),\dots,f_{i_v,l_v,m_v}(j_v),f_{i_v,l_v,m_v}(k_v),E_{R_{i_v,l_v,m_v}}(R_v)],\ \ddot{E}_K(M^*)>\\
G_{\hat{v}}:\ CT^*=<&[(i_q,j_q,k_q)_{q=1}^{v},\ f_{i_1,l_1,m_1}(j_1),f_{i_1,l_1,m_1}(k_1),E_{K_{i_1,l_1,m_1}}(R_1),\dots,f_{i_v,l_v,m_v}(j_v),f_{i_v,l_v,m_v}(k_v),E_{K_{i_v,l_v,m_v}}(R_v)],\ \ddot{E}_K(M^*)>\\
G_F:\ CT^*=<&[(i_q,j_q,k_q)_{q=1}^{v},\ f_{i_1,l_1,m_1}(j_1),f_{i_1,l_1,m_1}(k_1),E_{K_{i_1,l_1,m_1}}(R_1),\dots,f_{i_v,l_v,m_v}(j_v),f_{i_v,l_v,m_v}(k_v),E_{K_{i_v,l_v,m_v}}(R_v)],\ \ddot{E}_K(R_{M^*})>
\end{aligned}
$$

4) $S1$이 포함되는 경우에도 [9]에서 전개한 것처럼, 1차 다항식을 이용한 secret sharing 기법의 안전성을 이용하여 하이브리드 게임을 적용하면 된다.

질의가 발생하면 S_{i_1, j_1, k_1}는 더 이상 챌린지 단계에서 사용되지 않고, S_{i_1, j_1, k_1}가 두 개의 부분집합으로 분할된다. 그러므로 이후의 게임진행은 (S_{i_1, j_1, k_1}가 두 개의 부분집합으로 분할된) G_0와 G_1이 B의 챌린지 값 $(f(j_1), f(k_1), j_1, k_1, T)$ 과는 무관하게 동일한 (statistically identical) 게임이 된다.[5] 또한 암호(복호) 질의 시 수반되는 탈퇴자 집합들을 누적하는 과정에서 그때까지의 총 탈퇴자 집합이 $u \in S_{i_1}$이고 ($u \notin S_{j_1}$ 그리고 $u \notin S_{k_1}$)인 사용자 u를 적어도 한명 포함하는 경우에는, 위와 같이 임의의 $a, b, c \in Z_p$로 다항식을 $f(x) = ax^2 + bx + c \in Z_p[x]$로 결정한 후 게임을 진행한다. 이 경우에도 weak CCA1 안전성 모델에서는 G_0와 G_1이 B의 챌린지 값 $(f(j_1), f(k_1), j_1, k_1, T)$ 과는 무관하게 동일한 게임이 되기 때문이다.

챌린지 단계에서 A가 탈퇴자 집합 R^*와 M^*를 보내면, B는 $CoverFinding(R^*)$를 수행하여 $S_{i_1,j_1,k_1}, \ldots, S_{i_v,j_v,k_v}$을 구한다. S_{i_1, j_1, k_1}에 대한 그룹키로 암호화할 때 B는 자신이 챌린지로 받은 T 값을 이용한다. 다른 부분집합들에 대해서는 정상적인 그룹키로 메시지 암호키 K를 랜덤하게 선택한 후 다음과 같이 CT^*를 구성한다.

$$CT^* = < [(i_q, j_q, k_q)_{q=1}^{v},\ f_{i_1,l_1,m_1}(j_1), f_{i_1,l_1,m_1}(k_1), E_T(K), \ldots, f_{i_v,l_v,m_v}(j_v), f_{i_v,l_v,m_v}(k_v), E_{K_{i_v,l_v,m_v}}(K)],\ \ddot{E}_K(M^*) > \quad (12)$$

T가 $(f(j_1),\ f(k_1))$의 상수항이면 A는 S_{i_1, j_1, k_1}의 그룹키가 $T = K_{i_1, l_1, k_1}$인 G_0에서 공격을 수행하는 것이고, T가 다항식의 상수항과 관계없는 난수이면 A는 S_{i_1, j_1, k_1}의 그룹키가 R_{i_1, j_1, k_1}인 G_1에서 공격을 수행하는 것이다. 따라서 G_0와 G_1을 구별하는 A의 공격능력은 2차 다항식 비밀분산 기법의 상수항을 구분하는 B의 능력으로 전환된다. 그러나 비밀분산 기법의 상수항을 구분하는 문제는 A의 능력과 관계없이 advantage의 차가 0이다. (II. 6절 참고)

Claim 2. $|\Pr[G_1] - \Pr[G_1']| \leq w\, Adv_{B,\,CCA1}^{SKE}$

증명) A는 브로드캐스트 암호시스템에 대한 weak CCA1 공격을 수행한다. B는 S_{i_1, j_1, k_1}에 대응하는 그룹키를 제외하고 setup 과정과 동일하게 비밀키를 생성한다. S_{i_1, j_1, k_1}에 대응하는 그룹키는 B가 CCA1 공격을 하는 대칭키 암호시스템 SKE의 비밀키로 한다. 1) 사용자 u가 $u \notin S_{i_1}$인 경우, u의 비밀키 질의는 B가 (관련된 키를 알고 있으므로) 쉽게 응답할 수 있다. u가 $u \in S_{i_1}$이고 ($u \in S_{j_1}$ 또는 $u \in S_{k_1}$)인 경우, u의 비밀키 질의는 j_1노드의 레벨 l_1과 k_1노드의 레벨 m_1에 대응하는 임의로 선택된 다항식 $f_{i_1, l_1, m_1}(x)$을 이용하여 응답한다. 여기서 레벨 l_1과 레벨 m_1에 대응하는 그룹키는 SKE의 비밀키임을 상기하자. u가 $u \in S_{i_1}$이고 ($u \notin S_{j_1}$ 그리고 $u \notin S_{k_1}$)인 경우, B가 모르는 SKE의 비밀키가 상수항이 되도록 다항식 $f_{i_1, l_1, k_1}(x)$의 값을 결정할 수 없으므로 게임을 중단(abort)한다. 2) u가 $u \notin S_{i_1}$인 경우, A의 암호(복호) 질의는 B가 u의 비밀키를 이용하여 대응한다. u가 $u \in S_{i_1}$이고 j_1노드와 k_1노드의 레벨에 대응하는 그룹키를 이용하여 암호(복호)화 할 경우, B는 자신의 챌린저에게 암호(복호) 질의를 하고 그 결과를 A에게 보낸다. u가 $u \in S_{i_1}$이고 j_1노드와 k_1노드의 레벨에 대응하지 않는 그룹키를 이용할 경우 B는 (관련된 키를 알고 있으므로) 쉽게 응답할 수 있다.

챌린지 단계에서 A가 탈퇴자 집합 R^*와 M^*를 보내면, B는 $CoverFinding(R^*)$를 수행하여 $S_{i_1,j_1,k_1}, \ldots, S_{i_v,j_v,k_v}$을 구한다. S_{i_1, j_1, k_1}에 대한 암호문을 생성하기 위해 B는 메시지 암호키 K를 랜덤하게 선택한 후 B의 챌린저에게 K를 챌린지 값으로 보낸다. 그 결과로 받은 암호문을 T라 하자. 이를 이용하여 B는 CT^*를 다음과 같이 구성한다.

$$CT^* = < [(i_q, j_q, k_q)_{q=1}^{v},\ f_{i_1,l_1,m_1}(j_1), f_{i_1,l_1,m_1}(k_1), T, \ldots, f_{i_v,l_v,m_v}(j_v), f_{i_v,l_v,m_v}(k_v), E_{K_{i_v,l_v,m_v}}(K)],\ \ddot{E}_K(M^*) > \quad (13)$$

T가 K를 암호화한 것이면 A는 G_1에서 공격을 수행하는

5) 또한 더 엄밀하게는 이러한 게임진행이 2-SD 기법의 setup과정에서 분배되는 그룹키 개수와 같은 모든 w개의 부분집합에 대해서, 각 그룹키를 (independently) 랜덤한 값으로 변화시키는 하이브리드 게임진행을 생각하는 것이다.

것이고, T가 K와 같은 길이의 난수 R_1을 암호화한 것이면 A는 $G_{\hat{1}}$에서 공격을 수행하는 것이다. 따라서 위의 게임중단(abort)이 발생하지 않으면, G_1와 $G_{\hat{1}}$을 구별하는 A의 공격능력은 SKE에 대한 CCA1공격을 하는 B의 능력으로 전환된다. 여기서 abort가 발생하지 않을 확률은 2-SD의 setup 하에서 전체 그룹키 개수 w 중 한 개를 SKE의 비밀키로 선택하는 것이므로 $1/w$가 된다. 즉 $(1/w)|\Pr[G_1]-\Pr[G_{\hat{1}}]|$일 경우 A의 공격능력을 B의 공격능력으로 전환할 수 있다.

Claim 3. $\Pr[G_{\hat{1}}]-\Pr[G_1]=0$

증명) Claim 1과 유사하게 증명할 수 있다.

탈퇴자 집합 R^*로 생성되는 부분집합의 최대 개수를 v개 하면, v개의 부분집합 각각에 대해 위 Claim 1, Claim 2, Claim 3의 증명을 유사하게 적용할 수 있다. 그 결과 이후의 하이브리드 게임들에 대해 아래의 Claim 4, Claim 5, Claim 6을 쉽게 증명할 수 있다.

Claim 4. $\Pr[G_{\hat{j}}]-\Pr[G_{j+1}]=0$ (for $j=1,...,v-1$)

Claim 5. $|\Pr[G_j]-\Pr[G_{\hat{j}}]| \le w\,Adv^{SKE}_{B,CCA1}$ (for $j=2,...,v$)

Claim 6. $\Pr[G_{\hat{j}}]-\Pr[G_{\tilde{j}}]=0$ (for $j=2,...,v$)

Claim 7. $|\Pr[G_{\tilde{v}}]-\Pr[G_F]| \le Adv^{\ddot{SKE}}_{B,CPA^{OT}}$

증명) A는 브로드캐스트 암호시스템에 대한 weak CCA1 공격을 수행한다. B는 setup 과정과 동일하게 비밀키를 생성한다. 메시지를 암호화하는 $\ddot{SKE}$의 비밀키에 대해 B가 one-time CPA 공격을 하는 것임을 상기하자. B가 (관련된 키를 알고 있으므로) 사용자 u에 대한 비밀키 질의와 암호(복호)와 질의를 쉽게 응답할 수 있다.

챌린지 단계에서 A가 탈퇴자 집합 R^*와 M^*를 보내면, B는 $CoverFinding(R^*)$를 수행하여 $S_{i_1,j_1,k_1},....,S_{i_v,j_v,k_v}$을 구한다. B는 각 부분집합에 대응하는 메시지 암호키 $R_1,...,R_m$을 선택한 후, M^*를 B의 챌린저에게 챌린지 값으로 보낸다. 그 결과로 받은 암호문을 T라 하자. 이를 이용하여 B는 CT^*를 다음과 같이 구성한다.

$$\begin{aligned} CT^* = < & [(i_q,j_q,k_q)_{q=1}^{v},\ f_{i_1,l_1,m_1}(j_1), f_{i_1,l_1,m_1}(k_1), \\ & E_{K_{i_1,l_1,m_1}}(R_1),...,f_{i_v,l_v,m_v}(j_v), f_{i_v,l_v,m_v}(k_v), \\ & E_{K_{i_v,l_v,m_v}}(R_v)],\ T> \end{aligned} \quad (14)$$

T가 M^*를 암호화한 것이면 A는 $G_{\tilde{v}}$에서 공격을 수행하는 것이고, T가 랜덤 메시지 R_{M^*}을 암호화한 것이면 A는 G_F에서 공격을 수행하는 것이다. 따라서 $G_{\tilde{v}}$와 G_F을 구별하는 A의 공격능력은 $\ddot{SKE}$에 대한 one-time CPA공격을 하는 B의 능력으로 전환된다.

위에서 전개한 하이브리드 게임은 랜덤 메시지 R_{M^*}을 암호화한 것을 유지하면서 다시 각 부분집합에 대응하는 메시지 암호키를 R_i에서 K로 전환시키는 게임을 추가함으로써 완성된다. 이는 랜덤 메시지 R_{M^*}을 암호화한 것을 유지하면서 G_0에서 G_F로 변환되는 과정을 반대로 전개하면 된다. 이 경우 G_i에 대응하는 게임을 $\widetilde{G}_i$라 하자. 마지막 게임 $\widetilde{G}_0$은 G_0와 같으면서도 M^*대신 R_{M^*}을 암호화한 암호문이 된다. 결국 위 하이브리드 게임들을 종합하면 다음과 같은 식 (15)를 얻는다.

$$\begin{aligned} & \left|\Pr[G_0]-\Pr[\widetilde{G}_0]\right| \le \left|\Pr[G_0]-\Pr[G_1]\right| \\ & + \left|\Pr[G_1]-\Pr[G_{\hat{1}}]\right| + \cdots + \left|\Pr[G_{\tilde{v}}]-\Pr[G_F]\right| \\ & + \left|\Pr[G_F]-\Pr[\widetilde{G}_{\tilde{v}}]\right| + \left|\Pr[\widetilde{G}_{\tilde{v}}]-\Pr[\widetilde{G}_{\hat{v}}]\right| + \cdots \\ & + \left|\Pr[\widetilde{G}_{\hat{1}}]-\Pr[\widetilde{G}_0]\right| \\ & \le 2v(w\,Adv^{SKE}_{B,CCA}) + Adv^{\ddot{SKE}}_{B,CPA^{OT}} \end{aligned} \quad (15)$$

즉, $Adv^{BE}_{A,wCCA1} \le 2vw\,Adv^{SKE}_{B,CCA} + Adv^{\ddot{SKE}}_{B,CPA^{OT}}$ 으로 정리 4의 증명이 완성된다.

Ⅵ. 기존 기법과의 효율성 비교분석

제안한 2-SD 기법의 효율성을 기존 기법들과 비교한다.

편의상 PRG(Pseudo-Random Generator) 기반의 SD 기법 [2]을 SD^{PRG}, SS(Secret Sharing) 기반의 SD 기법을 SD^{ss} [9], 그리고 SS(Secret Sharing) 기반의 2-SD 기법을 2-SD^{ss}로 표기한다. [표 2]는 1) 암호문 전송량, 2) 수신자의 비밀키 저장량, 3) 복호화 시 필요한 계산량 측면에서 SD^{PRG}, SD^{ss}, 2-SD^{ss} 기법들 간의 효율성을 비교한 것이다.

1. 암호문 헤더 전송량

2-SD^{ss} 기법의 암호화 알고리즘은 (R, M)을 입력받아 $CoverFinding(R)=\{S_1,S_2\}$을 수행한 후 아래와 같은 암호문을 생성한다.

$$CT=<[\,Header^{\text{degree}1},\ Header^{\text{degree}2}],\ \ddot{E}_K(M)> \quad (16)$$

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. SD^{PRG}나 SD^{ss} 기법과 달리, 2-SD^{ss} 기법은 변형된 Coverfinding 알고리즘을 사용함으로써 탈퇴자 수 r에 대해 최대 r개의 부분집합이 (worst case로) 발생한다. r개의 부분집합은 $S1$ 집합에 포함되는 $S_{i,j}$형태로 또는 $S2$ 집합에 속하는 $S_{i,j,k}$형태로 결정된다. $S_{i,j,k}$형태의 부분집합은 $S_{i,j}$에 비해 암호문 헤더에 1개의 함수값을 더 전송해야하며 index도 1개 더 전송되므로 overhead가 커지게 된다. 2-SD^{ss} 기법에 가장 불리하도록 암호문 길이를 분석하기 위해 부분집합이 모두 $S_{i,j,k}$형태가 된다고 가정하자. 이 경우 암호문 헤더는 위 식 (17) 로 결정된다. $S_{i,j,k}$형태의 부분집합은 $S_{i,j,k}$마다 이를 표현하는 index는 i노드와 j노드 그리고 k노드를 묶어 (i,j,k)으로 표현한다. index 정보들 $\{(i_q, j_q, k_q)\}$는 [9]와 마찬가지로 암호문 헤더에 포함되어 전송된다고 하자.

$$Header^{\text{degree}2}=<\{(i_q, j_q, k_q)\},\ \left\{f_{i_q,l_q,m_q}(j_1), f_{i_q,l_q,m_q}(k_1), E_{K_{i_q,l_q,m_q}}(K)\right\}> \quad (17)$$

암호문 헤더에 포함되는 각 원소의 사이즈는 다음과 같다. 먼저 각 index 원소를 표현하기 위해서는 $\lceil \log(2n-1) \rceil$의 bits가 필요하다. 그 이유는 사용자 n명이 leaf노드로 대응되는 완전이진트리에서 총 노드의 수는 $2n-1$개가 되므로 하나의 노드를 표현하는데 $\lceil \log(2n-1) \rceil$ bits면 충분하기 때문이다. 따라서 index (i,j,k)는 $3\lceil \log(2n-1) \rceil$ bits로 표현된다. 다음으로 각 부분집합에 주어지는 $(f_{i,l,m}(j), f_{i,l,m}(k), E_{K_{i,l,m}}(K))$의 길이를 분석하면, 현재 상용화된 AES-128 대칭키 암호시스템을 고려하여 메시지 암호키 K를 128bits로 할 때, K를 표현하는 함수값도 동일한 128bits가 된다. 또한 $E_{K_{i,l,m}}(K)$은 AES를 이용하여 128bits 길이의 한 개 block 암호화로 해결할 수 있으므로 128bits가 된다. 결과적으로 index (i,j,k)와 $(f_{i,l,m}(j), f_{i,l,m}(k), E_{K_{i,l,m}}(K))$의 길이는 $(3\lceil \log(2n-1) \rceil + 2\times 128)$ bits가 되고 Cover- finding 알고리즘이 worst case로 r개의 부분집합을 생성할 수 있으므로 2-SD^{ss} 기법의 총 헤더 전송량은 $(\,3\lceil \log(2n-1) \rceil + 384)\times r$ bits를 갖는다.

2-SD^{ss} 기법의 이러한 암호문 헤더길이는 SD^{SS} 기법에 비해 확실히 더 짧아진 값이다. 이를 확인하기 위해 전체 사용자수를 $n=2^{31}$이고 탈퇴자 수를 $r=2^{12}$이라 하자. 이 경우 SD^{PRG} 기법은 1572672bits(≈192KB), SD^{SS} 기법은 2621120bits(≈320KB)인데 비해 2-SD^{ss} 기법은 1966080 bits(≈240KB)이다. 즉 2-SD^{ss} 기법은 SD^{SS} 기법과 비교할 때, SD^{SS} 기법의 약 (320-240=80)/320 = 1/4의 암호문 헤더

표 2. 기존 기법들과의 효율성 비교
Table 2. Performance comparison to the existing methods

기법	암호문 헤더 전송량	비밀키 저장량	복호화 연산량
SD^{PRG}[2]	$(\,2\lceil \log(2n-1) \rceil + 128)\times(2r-1)$	$O(\log^2 n)$	$O(\log n)$
SD^{SS}[9]	$(\,2\lceil \log(2n-1) \rceil + 256)\times(2r-1)$	$O(\log^2 n)$	$O(1)$
2-SD^{ss}	$(\,3\lceil \log(2n-1) \rceil + 384)\times r$	$O(\log^3 n)$	$O(1)$

를 줄이는 효과를 볼 수 있다. 이러한 비율은 암호문 헤더 전송량의 기울기 값이 SD^{PRG}<SD^{SS}<2-SD^{ss} 이므로 탈퇴자의 수가 증가함에 따라서 더 커지게 된다. SD^{PRG} 기법과 비교하면, SD^{SS} 기법에서 약 320/192=1.7배 길었던 암호문 헤더를 2-SD^{ss} 기법에서는 약 240/192=1.25배 긴 암호문 헤더로 줄이게 된다.

2. 저장량

사용자 u가 저장하는 비밀키 사이즈는 3장에서 기술한 대로 $\frac{1}{3}log^3 n-\frac{1}{3}log^2 n+\log n+1$개의 함수값을 저장해야 한다. 이 값은 SD^{PRG}와 SD^{ss} 기법이 저장하는 $\frac{1}{2}log^2 n+\frac{1}{2}log\, n+1$ 개에 비해 증가한 값이다.

실제 저장량을 비교하기 위해 전체 사용자수를 $n=2^{31}$이라 하고, 각각의 함수값을 128bits라 하자. SD^{PRG}와 SD^{ss} 기법은 63616bits(≈7.8KB)인데 비해 2-SD^{ss} 기법은 1234176bits(≈151KB)로 약 151/7.8=19.4배의 저장량을 가져야 한다.

3. 복호화에 필요한 연산량

복호화에 필요한 연산량은 기존 SD^{PRG} 기법이 최대 $\log n$번의 PRG 함수를 계산해야 하는 것에 비해, SD^{ss}와 2-SD^{ss} 기법은 Lagrange 보간법을 이용하여 Z_p에서 $(2,n)$-SS 또는 $(3,n)$-SS 연산을 수행하면 된다. 여전히 전체 사용자 수 n이나 탈퇴자 수 r에 관계없이 $O(1)$의 복호화 연산량을 요구하므로 매우 빠른 복호화 시간을 보장할 수 있다.

Ⅶ. 결 론

본 논문에서는 [9]에서 소개된 $(2,n)$-비밀분산과 SD (Subset Difference) 기법의 아이디어를 확장하여 2-SD 기법을 설계하는 방법을 제시하였다. 2-SD 기법 설계를 위해서는 $(2,n)$-비밀분산에서 확장된 $(3,n)$-비밀분산 기법을 이용하여 그룹키를 분배하는 아이디어를 새롭게 도입하였고, 2-SD 기법에 맞도록 Coverfinding 알고리즘을 수정하였다. 효율성 측면에서 새로운 2-SD 기법은 기존 SD^{SS} 기법에 비해 전송량을 약 25% 줄일 수 있는 효과를 보였다. 이는 SD^{PRG} 기법에 비해 (worst case로) 약 1.25배 긴 전송량을 갖는 것으로서 기존 SD^{SS} 기법이 1.7배 긴 전송량을 가진 것에 비해 개선된 결과이다. 이렇게 전송량을 줄이기 위해서는 저장량이 증가하는 단점이 있으나 최근 디바이스의 저장량을 고려하면 큰 문제가 되지는 않는다. 사용자의 복호화 연산량은 [9]와 마찬가지로 여전히 $O(1)$으로 사용자 수나 탈퇴자의 수에 관계없다는 장점을 가진다.

이론적으로 본 논문의 결과는 비밀분산 기법을 이용하여 (안전성을 약화시키면서) 2-SD 기법을 설계할 수 있음을 보인다. 이후의 흥미 있는 연구는 유사난수생성수(PRG)를 이용하여 2-SD 기법을 설계할 수 있느냐 하는 것으로, 이 결과는 SD^{PRG} 기법을 새로운 각도에서 일반화하는 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] ChongHee Kim, YongHo Hwang and PilJoong 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] Jae Hwan Lee and Jong Hwan Park, "Broadcast encryption system using secret sharing and subset difference methods", Journal of Broadcast Engineering, 20(1), pp.92-109, Jan. 2015.

[10] Ji Yong Jang, Dae Hun Nyung, and Joo Seok Song, "2-Subset Difference scheme for broadcast encryption", Journal of the Korea Institute of Information Security & Cryptology, 16(4), pp.1-5, Aug. 2006.

[11] Jae Hwan Lee and Jong Hwan Park, "Security analysis of broadcast encryption system based on 2-subset difference method", Journal of Broadcast Engineering, 19(4), pp.502-509, July. 2014.

저 자 소 개

이 재 환

- 2009년 3월 ~ 현재 : 상명대학교 컴퓨터과학과 학사과정
- 주관심분야 : 브로드캐스트 암호, 전자서명 등

박 종 환

- 1999년 2월 : 고려대학교 이과대학 수학과 (학사)
- 2004년 2월 : 고려대학교 정보보호대학원 정보보호학과 (석사)
- 2008년 8월 : 고려대학교 정보경영공학전문대학원 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