http://dx.doi.org/10.3745/KIPSTC.2011.18C.6.369

안전한 클라우드 비즈니스를 위한 접근권한 분산관리

송 유 진[†] · 도 정 민[††]

요 약

최근 비즈니스 환경에서 공유되는 데이터의 기밀성과 유연성있는(fine-grained) 접근제어를 보장하기 위해서 KP-ABE(Key Policy-Attribute Based Encryption)와 PRE(Proxy Re-Encryption)를 활용한 시스템 모델이 제안되었다. 그러나 기존 방식은 클라우드 서버에 집중된 복호권한 때문에 데이터 기밀성을 침해하게 된다. 또한, 접근권한 관리에 대한 개념을 고려하지 않았으므로 악의적인 내부사용자의 공격에 취약하다. 이러한 문제를 해결하기 위해서 기존방식의 프로토콜 모델에서 권한 관리자 그룹을 두어 클라우드 서버에 저장되는 데이터 파일(data file)을 분산 저장하여 데이터 기밀성을 보장하고 AONT 기반의 XOR 임계치 비밀분산을 활용하여 접근권한 관리 모델을 구성하였다. 또한 XOR 쉐어를 활용하여 권한의 가중치를 부여할 수 있는 방법을 구체화했다. 4장에서 기존방식과 제안방식과의 비교 분석과 기능적 활용에 대해서 서술하여 제안방식의 차별화를 부각시켰다.

키워드 : AONT(All Or Nothing Transform), XOR 임계치 비밀분산, XOR 쉐어, 복원쉐어, 접근권한 관리

Distributed Access Privilege Management for Secure Cloud Business

You-Jin, Song[†] · Jeong-Min, Do[††]

ABSTRACT

To ensure data confidentiality and fine-grained access control in business environment, system model using KP-ABE(Key Policy-Attribute Based Encryption) and PRE(Proxy Re-Encryption) has been proposed recently. However, in previous study, data confidentiality has been effected by decryption right concentrated on cloud server. Also, Yu's work does not consider a access privilege management, so existing work become dangerous to collusion attack between malicious user and cloud server. To resolve this problem, we propose secure system model against collusion attack through dividing data file into header which is sent to privilege manager group and body which is sent to cloud server. And we construct the model of access privilege management using AONT based XOR threshold Secret Sharing, In addition, our scheme enable to grant weight for access privilege using XOR Share. In chapter 4, we differentiate existing scheme and proposed scheme.

Keywords : AONT(All Or Nothing Transform), XOR Threshold Secret Sharing, XOR Share, Recovery Share, Access Privilege Management

1. 서 론

컴퓨팅 자원을 서비스로 제공하는 클라우드 컴퓨팅은 학계와 산업계에서 주목되고 있는 컴퓨팅 패러다임이다. 예를 들면, 아마존(Amazon)사의 S3(Simple Storage Service) 데이터 스토리지 서비스는 데이터를 저장할 온라인 공간을 임대해 주고 한달에 기가바이트당 0.15달러, 약 170원도 안되는 금액으로 데이터 스토리지 서비스를 제공하고 있다[1]. 이러한 형태의 클라우드 컴퓨팅 서비스를 적절히 활용한다면 기업은 많은 IT 투자 비용을 절감할 수 있다. 기업 입장에서 효율적인 IT 투자는 기반 시설을 직접 소유하는 것보다 클라우드 컴퓨팅 환경에서 서비스로서 제공받음으로써 타 경쟁사보다 높은 경쟁우위를 달성할 수 있다. 이러한 장점을 활용하여 기존 비즈니스 환경은 클라우드 컴퓨팅 기술을 활용하는 클라우드 비즈니스 환경으로 점차 이동할 것으로 예상된다. 클라우드 비즈니스 환경에서 기업에게 손실을 주고 개인의 프라이버시를 침해할 수 있는 정보는 본인이 보관하는 것이 안전하다. 그러나 효율적인 업무를 위해서 기업의 기밀정보나 개인의 프라이버시가 담긴 정보의 공유·활용이 점차 증가할 것으로 예상된다.

※ 이 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2011-0027333).
† 정 회 원 : 동국대학교 정보경영학과 교수
†† 준 회 원 : 동국대학교 전자상거래협동과정 석사과정
논문접수 : 2011년 5월 14일
수 정 일 : 1차 2011년 9월 8일, 2차 2011년 10월 10일
심사완료 : 2011년 11월 2일

정보의 활용이 필요할 때 해당 정보 소유자에게 열람 동의를 거치는 것이 안전하지만 현재 개인 정보는 각 기관에 의해서 관리되고 있는 실정이다.. 의료환경을 실례로 들면 환자의 이름, 성별, 주민번호 등의 개인정보뿐만 아니라 통원내역, 병명 등과 같은 진료기록도 병원에서 관리하고 있다. 안전한 의료정보의 공유를 위해서는 환자가 자신의 진료기록을 보관하고 사용자(의료진, 의학연구기관, 보험기관 등)가 진료기록이 필요할 때 환자의 동의를 구해야 한다. 그러나 효과적인 의료서비스가 이루어지려면 의료정보는 사용자가 언제든지 접근 가능해야 한다. 즉, 서비스의 가용성을 위해서 개인정보를 의료 DB 서버에 보관한다. 하지만 이러한 DB 서버에 누구나 식별 가능한 평문형태로 정보를 저장·관리하게 되면 DB 서버에 대한 신뢰 문제에 직면하게 된다. DB 서버에 접근하는 악의적인 공격자에 의해 불법적인 2차 이용(DB 접근이 허용되지 않은 사용자와의 정보 공유) 등과 같은 문제가 발생한다. 따라서 암호방식을 도입하여 평문을 식별 불가능하도록 암호화하여 저장하는 것이 요구된다.

최근 일본 대지진으로 인한 기업 데이터의 소실로 데이터 백업에 대한 중요성이 부각되고 있다. 데이터 백업의 필요성 중 자연재해로 인한 데이터 소실은 현실적으로 발생할 확률이 적은 진부한 이유였다. 하지만 근래에 많은 자연 재해가 발생하고 있고 이로 인한 피해 상황도 속출하고 있다. 이와 더불어 농협의 사례[2]와 같이 악의적인 내부 공격자에 의한 데이터 삭제에 대비하여 기밀정보를 식별할 수 없는 형태로 데이터 센터에 파일을 분산·저장해야 한다. 또한, 악의적인 내부 공격자가 데이터 열람하기 이전에 시스템 접근에 대한 정당성을 평가하는 접근권한 관리 절차의 도입이 필요하다. 여기서 접근권한 관리는 환자가 접근을 허가한 사용자라 할지라도 데이터 열람에 대한 정당성을 재차 평가하여 접근허가 여부를 결정하는 것이다.

즉, 클라우드 컴퓨팅 환경에서 기밀정보를 안전하게 공유할 수 있는 방식이 필요하다. Yu, Wang, Ren and Lou(2010)은 이러한 요구사항을 만족시키는 방식(이하 Yu의 방식)을 제안하고 있다[3]. KP-ABE(Key Policy-Attribute Based Encryption)를 활용하여 데이터에 대한 접근제어와 기밀성을 보장하고 PRE(Proxy Re-Encryption)를 이용하여 클라우드 서버에 업무를 위임함으로써 다수의 사용자 이용에 따른 키 생성 등의 계산상 과부하를 해결하고 있다. Yu의 방식[3]은 암호문을 임의의 비밀키 암호방식으로 암호화하고 비밀키를 KP-ABE로 암호화하여 클라우드 서버에 저장한다. 사용자의 열람 요청이 있을시 클라우드 서버는 암호문과 암호화된 비밀키를 전송한다. 이를 통해서 [3]에서는 안전한 비밀키 분배 문제를 해결하고 있다. 하지만 클라우드 서버가 암호문과 암호화된 키를 모두 보유하고 있는 상태, 다시 말하면 복호와 관련된 모든 권한을 소유하고 있으므로 클라우드 서버의 복호권한의 분산관리가 요구된다. 전적으로 신뢰할 수 없는 클라우드 서버에 복호권한이 집중되면 사용자와 서버간의 공모 공격에 취약하게 된다. 또한 [3]에서는 접근권한 관리의 개념을 고려하고 있지 않으므로 정당한 사용자라도 접근에 대한 정당성을 재차 판별할 수 있는 구체적인 절차가 명시되어야 한다. 이를 통해서 사용자가 데이터를 남용하거나 오용하는 것을 방지할 수 있다.

본 논문에서의 제안방식은 Yu의 방식에서 사용된 비밀키 분배와 암호문 저장 방식에 수정을 가한 방식이다. 먼저, 권한 관리자 그룹이라는 신뢰할 수 있는 기관을 두어 기존의 방식에서 클라우드 서버에 집중되는 복호권한(data file)을 분산 저장한다. 이러한 기능을 형식화하기 위해서 비밀키의 안전한 분배를 위해 암호화하는 데 활용되었던 KP-ABE를 대신하여 AONT기반의 XOR 임계치 비밀분산을 활용한다. 권한 관리자 그룹은 AONT기반의 XOR 임계치 비밀분산을 통해 분산된 쉐어(복원쉐어, XOR쉐어) 중 일부(복원쉐어)를 보관하고 사용자가 데이터에 접근시 정당성을 평가하여 비밀키를 복원하는 기능을 수행한다. 이를 통해서 접근권한 관리절차가 구현된다.

현재 스마트폰에 기반한 모바일 클라우드 서비스가 활성화되고 있다. 예를 들면 환자의 의료데이터를 서버에 저장하고 환자 자신은 물론 다수의 이용자가 스마트폰의 어플리케이션을 통해 언제 어디서든 접속하여 원하는 정보를 열람할 수 있는 서비스를 제공한다. 이러한 의료데이터를 공유·활용하고자 할 때 신뢰할 수 있는 기관을 통해 데이터 접근제어가 가능하다면 데이터의 소유자는 안심하고 자신의 데이터를 온라인 공간에 저장할 수 있다. 제안방식을 의료 분야에 도입하게 되면 데이터의 기밀성을 보장함과 동시에 접근권한 관리를 가능하게 될 것이다.

본문의 구성은 다음과 같다. 제 2장에서 제안방식을 구성하는 관련 연구를 소개한다. 그리고 제 3장에서 제안방식인 AONT 기반의 XOR 임계치 비밀분산의 설계 및 특징, 상세한 구조에 대해 서술한다. 제 4장에서 제안방식과 Yu의 방식을 비교·분석과 제안방식의 기능적으로 활용 가능한 부분에 대해서 서술한다. 마지막으로 제 5장에서 본 논문의 끝을 맺는다.

2. 관련 연구

2.1 KP-ABE와 PRE

1) KP-ABE

KP-ABE에서 사용자의 비밀키는 접근구조와 관련되며 암호문은 속성집합과 관련된다[4]. 즉, 암호문 내의 속성집합이 사용자 비밀키의 특정 복호정책을 만족하면 암호문이 복호되는 구조이다. 예를 들어 (대학교수, 연구팀, 연구팀장, 학과장) 속성으로 구성된 접근구조(=((대학교수∧연구팀)∨(연구팀장∨학과장)))가 있으면 복호정책은 (대학교수이고 연구팀 또는 연구팀장이나 학과장)이다('∧', '∨'는 각각 AND, OR 게이트). 예를 들어 의과대학에서 신종플루에 대한 연구를 수행한다고 가정한다. 연구데이터는 속성집합({연구팀, 학과장})으로 암호화되어 있고 비밀키는 접근구조와 관련되어 있다. 연구자가 그동안 연구해 왔던 데이터에

대한 접근을 원한다면 암호문은 대학교수이고 연구팀이라는 AND 게이트는 만족하지 못하지만 학과장이라는 OR 게이트를 만족하므로 데이터에 접근가능하다. KP-ABE는 이하 4개의 알고리즘으로 구성된다.

a. Setup(1^k) : 보안 파라미터 k 를 입력하여 그 값에 대응하는 공개파라미터 PK와 마스터키 MK를 출력하는 알고리즘.

① 속성 $U=\{1,2,...,n\}$를 정의한다.

② 각각의 속성 $i\in U$에 포함시키고 랜덤한 Z_P로부터 숫자 t_i를 균등하게 선택한다.

③ 랜덤한 Z_P에 y를 균등하게 선택.

④ PK는 $< T_1=g^{t_1},...,T_{|U|}=g^{t_{|U|}}, Y=e(g,g)^y >$ 고 MK는 $\langle t_1,...,t_{|U|},\ y\rangle$ 이다.

b. Encrypt(PK, M, γ) : 공개파라미터 PK와 접근구조 T와 평문 M을 입력하여 그 평문에 대응하는 암호문 E을 출력하는 알고리즘.

① 속성 γ의 집합아래 암호화 메시지 $M\in G_2$하고 랜덤값 $s\in Z_P$를 선택한다.

② 암호문은 $E=<\gamma, E'=MY^s, E_i=T_i^{s_{i\in\gamma}}>$이다.

c. KeyGen(MK, T) : 마스터키 MK와 공개 파라미터 PK을 입력하여 접근구조 A에 대응하는 비밀키 D을 출력하는 알고리즘.

① 만약에 $T(\gamma)=1$이면 복호화가 가능한 사용자에게 키를 출력한다.

② 다음의 비밀값 $D_x=g^{\frac{q_x(0)}{t_i}}$, $i=att(x)$를 사용자에게 준다.

d. Decrypt(CT, D) : 비밀키 D와 암호문 E을 입력하여 암호문에 대응하는 평문(대응이 없는 경우는$\perp$)을 출력하는 알고리즘.

① 암호문 $E=<\gamma, E'=MY^s, E_i=T^s_{i\ i\in\gamma}>$와 비밀키 D, 노드 x의 입력으로 순환 알고리즘 DecryptNode (E,D,x)을 정의한다.

② G_2의 그룹 요소 또는 $\perp$를 출력한다.

③ 만약에 노드 x = LeafNode라면 DecryptNode $(E,D,x)=e(D_x,E_i)=e(g^{\frac{q_x(0)}{t_i}},g^{s\cdot t_i})=e(g,g)^{s\cdot q_x(0)}$를 계산하고, $i\in\gamma$이면 $\perp$로 정의한다.

④ $F_z\neq\perp$면

$$F_x=\prod_{z\in S_x}F_z^{\Delta i,S'_{z(0)}},\ S'_x=index(z):z\in S_x$$

$$=\prod_{z\in S_x}(e(g,g)^{r\cdot q_{z(0)}})^{\Delta_{i,S'_x}}$$

$$=\prod_{z\in S_x}(e(g,g)^{r\cdot q_{parent(z)'index(i)}})^{\Delta_{i,S'_x}}$$

$$=\prod_{z\in S_x}(e(g,g)^{r\cdot q_{z(0)}\cdot\Delta_{i,S'_x}}$$

$$=e(g,g)^{s\cdot q_x(0)}$$로 계산한다.

2) PRE

PRE는 프록시 서버가 평문에 대한 어떠한 정보의 습득없이 A의 암호문을 B가 복호 가능하도록 재암호화하는 방식이다[5]. 즉, A는 B에게 자신의 암호문에 대한 복호 권한을 위임하는 재암호화키(Re-encryption key)를 생성한 후, 프록시 서버에 송신한다. 그러면 프록시 서버는 재암호화키를 통해서 A의 암호문을 B의 비밀키로 복호 가능한 암호문으로 변환하여 B에게 전달한다. 그러면 B는 자신의 비밀키로 암호문을 복호한다. PRE는 Key Generation, Encryption, Re-encryption, Decryption의 4가지 알고리즘으로 구성된다. 각 알고리즘에 대한 설명은 아래와 같다.

a. Key Generation

- 소수 위수 q의 $<g>=G$
- 비밀키 $SK_a=a$와 $SK_b=b$를 랜덤하게 생성 $(a\in Z_q^*,\ b\in Z_q^*)$
- 공개키 $PK_a=g^a$와 $PK_b=g^b$를 생성
- 재암호화키 $RK_{A\to B}=b/a=b\cdot a^{-1}(\mathrm{mod}q)$를 생성

b. Encryption

- $r\in Z_q^*$을 랜덤하게 선택
- 공개키 PK_a로 평문 $m\in G$을 암호화

$$C_a=(g^r\cdot m, g^{ar})$$

c. Re-encryption

- 암호문 C_a를 재암호화키 $RK_{A\to B}$를 이용하여 C_b로 재암호화

$$C_a=(g^r\cdot m, g^{ar})$$

$$C_b=(g^r\cdot m,(g^{ar})^{RK_{A\to B}})=(g^r\cdot m,(g^{ar})^{b/a})$$

$$=(g^r\cdot m, g^{br})$$

d. Decryption

- 비밀키 SK_b에 의해 암호문 C_b를 복호

$$=\frac{g^r\cdot m}{(g^{br})^{1/b}}=\frac{g^r\cdot m}{g^r}=m$$

Yu의 방식에서는 데이터의 기밀성을 보장하기 위해서 KP-ABE로 정보를 암호화하고 데이터 소유자의 키 생성 등에 대한 계산상 과부하를 줄이기 위해서 PRE 방식을 활용하여 클라우드 사업자에게 업무를 위임한다.

2.2 Yu et al. 방식

[3]에서는 기존의 비즈니스 환경이 클라우드 컴퓨팅 환경으로 이동해갈 것이라고 전망하고 이러한 환경에서 데이터의 기밀성을 보장할 수 있는 시스템 모델을 제안하고 있다. 본 방식에서도 클라우드 컴퓨팅 환경에서의 데이터 기밀성 문제는 클라우드 사업자가 믿을만하지 못하다는 것을 지적한다. 각 분야와 관련된 문서들을 평문으로 수신하고 전송하게 되면 신속하고 편리하겠지만 전적으로 신뢰할 수 없는 클라우드 사업자를 통해서 전송된다면 데이터의 기밀성이 침해될 수 있다.

이에 대해서 데이터를 속성집합으로 암호화하고 데이터의 속성집합에 만족하는 접근구조와 관련된 비밀키를 소지하고 있으면 데이터를 열람할 수 있는 속성기반 암호화(KP-ABE)의 컨셉을 활용한다. 즉, 비밀키 DEK(Data Encryption Key)로 데이터를 암호화하여 바디(body)를 구성하고 속성집합으로 DEK를 암호화하여 헤더(header)를 구성한다. 이러한 헤더와 바디를 결합하여 데이터 파일(data file)을 구성하여 클라우드 사업자에게 전송한다. 사용자 그룹이 데이터에 접근하고자 할 때 소지하고 있는 비밀키가 헤더의 속성집합에 만족하면 DEK를 얻고 바디의 데이터를 열람할 수 있다.

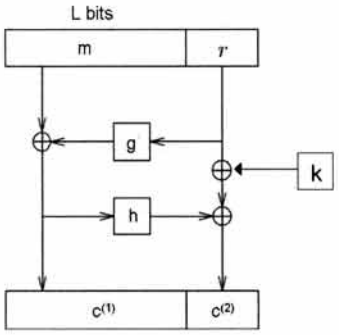
Yu의 방식은 데이터 소유자가 시스템 속성 집합의 정의, 시스템과 관련된 키 생성 및 설정을 하고 데이터의 암호화, 데이터를 암호화한 비밀키를 KP-ABE로 암호화하여 클라우드 사업자에게 전송하면 암호화된 키와 암호문의 관리는 전적으로 클라우드 사업자가 시행한다. 시스템 상의 사용자를 효율적으로 철회하기 위해서 PRE를 활용하여 속성의 버전을 업데이트하면서 사용자가 동일한 버전의 속성을 소지하고 있지 않으면 연산이 불가능하게 하고 있다. 이러한 기능을 클라우드 사업자가 수행하기 위해서 데이터 소유자는 접근구조를 Dummy Attribute가 포함된 접근구조를 생성하여 클라우드 사업자에게는 Dummy Attribute가 제외된 접근구조를 전달함으로써 시스템 마스터키의 불법적인 사용을 막는다.

하지만 철회된 사용자와 클라우드 사업자간의 공모를 통해서 클라우드 서버가 공모자를 위한 암호문의 속성 버전을 업데이트하지 않으면 데이터에 대한 접근이 가능하다. 즉, 프록시가 암호문의 속성 버전을 업데이트하지 않은 상태에서 기존에 발급받은 비밀키만 소지하고 있으면 암호문의 속성에 만족하기 때문에 평문으로 복호가 가능하다.

제안방식은 권한 관리자 그룹이라는 믿을 수 있는 공인기관을 두고 데이터 파일을 분산 저장하므로써 기존 방식의 공모 공격을 해결하고자 한다. 제안방식에서 바디는 AES로 암호화된 암호문, 헤더는 AES키를 d회 AONT 변환을 적용한 후의 복원쉐어다. 헤더는 권한 관리자 그룹, 바디는 클라우드 사업자에게 전송된다. 이러한 데이터 파일의 분산 저장으로 공모 공격에 대한 안전성을 확보하여 데이터의 기밀성을 보장하고 부가적으로 사용자 그룹이 데이터 접근시 권한 관리자 그룹에게 반드시 승인을 받아야 하는 접근권한 관리 기능을 구현하고자 한다.

2.3 AONT 암호화 모드

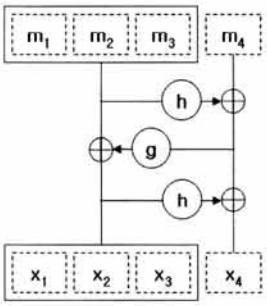
기존의 AONT(All Or Nothing Transform) 비분리 암호화 모드는 비밀키를 사용하지 않고 평문을 변환하는 스크램블(Scramble)부와 비밀키를 사용하여 평문을 암호화하는 암호화부로 구성된다(Rivest 1997). 여기서 $h : \{0,1\}^{\ell} \rightarrow \{0,1\}^{*}$는 hash function이고 $g : \{0,1\}^{*} \rightarrow \{0,1\}^{\ell}$는 generator로 한다. 데이터 m을 L 비트의 평문, 송신자와 수신자가 공유하는 비밀키를 k로 한다. 난수 r과 평문 m을 다음과 같이 암호화한다. 이때, 암호문은 $c = c^{(1)} \parallel c^{(2)}$ 이다. (그림 1)은 [6]에서 제안된 비분리 암호화 모드다.



$$c^{(1)} = m \oplus g(r),$$
$$c^{(2)} = r \oplus k \oplus h\left(c^{(1)}\right)$$

(그림 1) 비분리 암호화 모드

비분리 암호화 모드는 비밀키 암호방식과 같이 송·수신자간의 안전한 키 분배의 문제가 있다. 본 논문에서는 비밀키를 사용하지 않는 AONT를 활용한다[7]. (그림 2)는 [7]에서 제안된 AONT 방식이다.



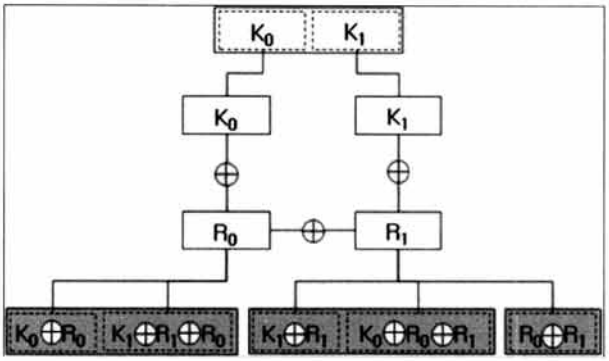
(그림 2) 제안방식에서 사용하는 AONT

[7]에서는 AONT 변환을 통해서 데이터의 가변성(Variability)을 갖지만 평문 복원시 변환된 모든 데이터가 있어야 한다는 점에서 용장성(Redundancy)이 결여되어 있다. 본 논문에서는 용장성을 제공하기 위해서 AONT변환과 XOR 임계치 비밀분산을 결합한 방식을 활용한다.

2.4 XOR 임계치 비밀분산

XOR(eXclusive OR) (k, n) 임계치 비밀분산 방식은 평문을 n개의 XOR 쉐어(Share)로 분산하여 k개의 XOR 쉐어만

으로 복원이 가능한 XOR 연산 기반의 비밀분산 방식이다[8]. 기존의 다항식 연산 기반의 비밀분산[9]은 복원시 계산상 과부하를 초래한다. 이와 비교하여 XOR 임계치 비밀분산은 비트연산이기 때문에 계산의 효율성 측면에서 우수하다. 하지만 평문을 n개의 XOR 쉐어로 분산시 평문이 n배만큼 용량이 증가한다. 즉, 평문의 소실에 대한 안전성을 위해서 XOR 쉐어의 개수를 증가시키면 평문의 n배만큼 저장공간이 필요하다. (그림 3)은 $n=3$, $k=2$일 때 XOR 임계치 비밀분산의 예를 나타낸다.



(그림 3) XOR (2,3) 임계치 비밀분산
(XOR threshold Secret Sharing)

(k,n) 비밀분산 방식의 일반적인 특징 중 하나는 n개의 쉐어 중 임의의 $k-1$ 이하의 쉐어가 주어졌을 때 비밀복원이 불가능하며 비밀에 대한 어떠한 정보도 획득할 수 없는 것이다. 비록 쉐어가 원래의 비밀정보의 조각(Share)이라고는 하지만 키 값(예를 들면 [그림 3]에서의 R_0, R_1)이 입력되어 변환되었기 때문에 쉐어 자체만으로는 비밀에 대한 아무런 정보도 유추할 수 없다. 이는 다항식 연산 기반의 비밀분산[9]의 안전성과 동일한 방식이다. XOR (2, 3) 임계치 비밀분산 방식을 비밀분산과 비밀복원 단계로 나누어서 설명한다.

a. 비밀분산

- 비밀정보 S를 $K_0 \| K_1 (K_i = \{0,1\}^l (i=1,2))$로 분할
- 2개의 난수 $R_0, R_1 \left(R_i = \{0,1\}^l (i=1,2)\right)$을 생성
- K_0, K_1와 R_0, R_1의 XOR 연산을 통해서 다음과 같이 $S_{i(i=1,2,3)}$를 생성

$$S_1=(K_0 \oplus R_0 \oplus R_1,\ K_1 \oplus R_1)$$
$$S_2=(K_0 \oplus R_0,\ K_1 \oplus R_0 \oplus R_1)$$
$$S_3=(R_0,\ R_1)$$

b. 비밀복원

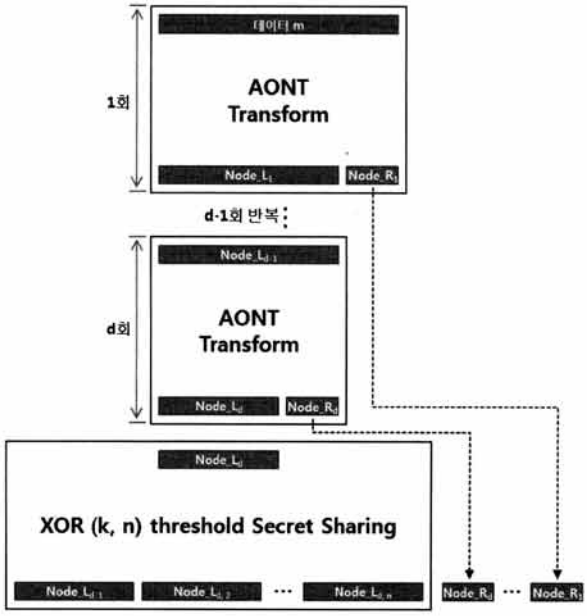
- 임계치 k개 만큼의 S_i를 모아 XOR 연산을 통해서 아래와 같이 비밀정보 S를 복원(설명을 위해서 S_1과 S_2를 사용)

$$K_0 = K_0 \oplus R_0 \oplus R_0$$
$$K_1 = K_1 \oplus R_0 \oplus R_1 \oplus R_0 \oplus R_1$$
$$S=(K_0 \| K_1)$$

3. 제안 방식

3.1 AONT와 XOR 임계치 비밀분산을 활용한 프로토콜 구성

AONT[7]와 XOR 임계치 비밀분산[8]을 결합하여 데이터의 가변성과 용장성을 모두 보장하는 방식을 제안한다[10]. (그림 4)는 AONT 기반의 XOR 임계치 비밀분산의 구성도이다.

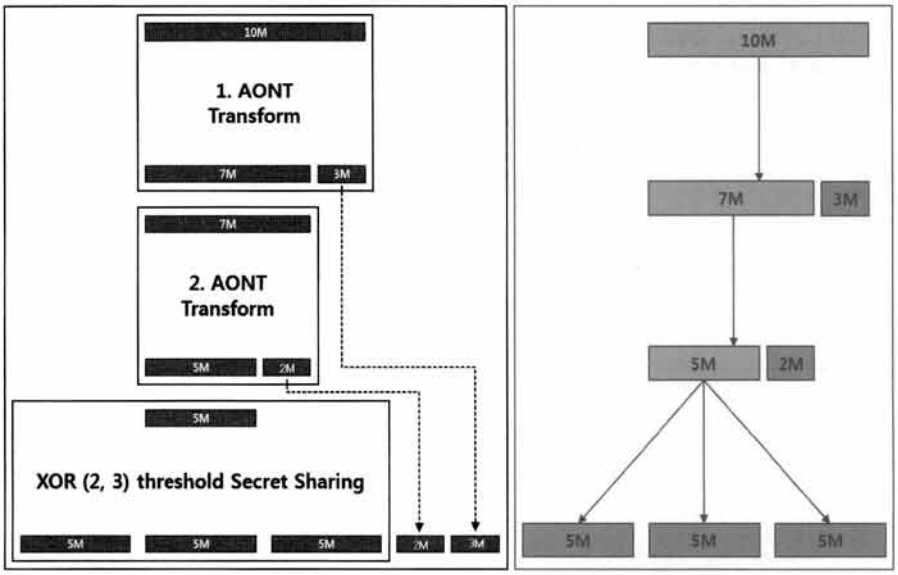


(그림 4) AONT 기반의 XOR (k,n) 임계치 비밀분산

(그림 4)와 같이 비밀정보에 AONT를 반복 적용함으로써 Node_L_i(i=1,...,d-1)의 크기를 줄인 후 Node_L_d(d회의 AONT 변환이 수행된 쉐어)에 XOR 임계치 비밀분산을 적용한다. 비밀정보에 AONT 변환을 적용하여 데이터를 원하는 블록크기로 분할하는 가변성과 Node_L_d에 XOR 임계치 비밀분산을 적용함으로써 데이터 소실에 대비한 용장성을 확보할 수 있도록 설계하고 있다.

여기서 AONT 변환시 생성되는 쉐어로서 복원할 때 필수적으로 필요한 쉐어(Node R_1, Node R_2,..., Node R_d)를 복원쉐어(Recovery shares)라고 하고 XOR 임계치 비밀분산으로 생성되는 쉐어로서 용장성을 가지는(k개의 쉐어만 있으면 나머지 쉐어는 소실되도 상관없는) 쉐어(Node $L_{d,1}$, Node $L_{d,2}$,..., Node $L_{d,n}$)를 XOR 쉐어라고 한다.

제안방식의 프로토콜에서는 복원쉐어를 권한 관리자 그룹이, XOR 쉐어를 사용자가 소지한다. 사용자가 데이터 열람을 원할 때 k개의 XOR 쉐어로 Node_L_d를 복원하고 권한 관리자 그룹으로부터 복원쉐어를 받아서 AES키를 복원하는 것을 접근승인 절차(접근권한 관리)로 활용한다.



(그림 5) AONT 기반의 XOR (2,3) 임계치 비밀분산

본 적용에 대한 구체적인 예는 (그림 5)와 같다.

10M(Mega byte)의 평문을 AONT 변환으로 분할한다. 1번째 수행시 10M를 7M와 3M로 분할하고 2번째 수행시 7M를 5M와 2M로 분할한다. 그리고 5M에 XOR (2,3) 임계치 비밀분산을 사용하여 3개의 XOR 쉐어로 분산하고 2개의 XOR 쉐어만 있으면 원래대로 복원 가능하게 한다.

XOR (2,3) 임계치 비밀분산만으로 10M의 평문을 분산하면 30M의 저장공간이 필요한 것에 비해 AONT 기반의 XOR (2,3) 임계치 비밀분산을 사용하면 20M의 저장공간이 필요하다. 또한 AONT를 반복 적용하면 비밀분산으로 분산된 쉐어의 용량을 더 줄일 수 있다. 즉, 효과적으로 저장공간을 활용하고 데이터 소실에 대한 안전성도 보장한다.

1) 개요

제안방식은 기존방식과 다르게 비밀키와 관련된 정보(헤더)와 암호문(바디)을 분산 저장한다. 이로 인해 복호권한의 분산관리와 동시에 공모 공격에 대한 안전성을 확보할 수 있다. 또한, 제안방식은 KP-ABE를 대신해서 AES (Advanced Encryption Standard)[11]와 AONT 기반의 XOR 임계치 비밀분산을 활용하여 데이터 기밀성의 보장과 안전한 키 분배 문제 해결하고 접근권한 승인절차를 실행 가능하게 한다.

데이터 소유자는 데이터 m을 AES로 암호화하고 AES키를 AONT 기반의 XOR 임계치 비밀분산으로 분할·분산한다. AES로 암호화된 암호문은 클라우드 서비스 제공자 (Cloud Service Provider ; CSP)에게, AES키를 AONT 변환으로 분할된 모든 복원쉐어는 권한 관리자 그룹에 전송한다. 마지막 AONT 변환으로 분할된 Node_L$_d$는 XOR 임계치 비밀분산을 통해서 XOR 쉐어로 분산하여 사용자에게 나눠준다. 사용자가 암호문 C를 복호하고자 한다면 우선 다른 사용자가 소지하고 있는 XOR 쉐어로 Node_L$_d$를 복원하고 권한 관리자 그룹에게 정당한 사용자임을 인증(접근 승인) 받아서 모든 복원쉐어를 입수한다. AONT 역변환 과정을 수행하여 AES키를 입수하고 암호문 C에서 데이터 m을 복호한다. (그림 6)은 제안방식의 프로토콜 구성도이다.

2) 상세

① System Setup : 시스템을 설정하는 단계

- h: $\{0,1\}^{\ell(s-1)} \rightarrow \{0,1\}^{\ell}$는 해쉬함수,
  g: $\{0,1\}^{\ell} \rightarrow \{0,1\}^{\ell(s-1)}$는 생성원함수
- ℓ은 1개의 블록 크기, s는 블록의 수

② Encryption : 데이터 m을 AES로 암호화하는 단계

- 데이터 m을 AES로 암호화하여 암호문 C를 구성

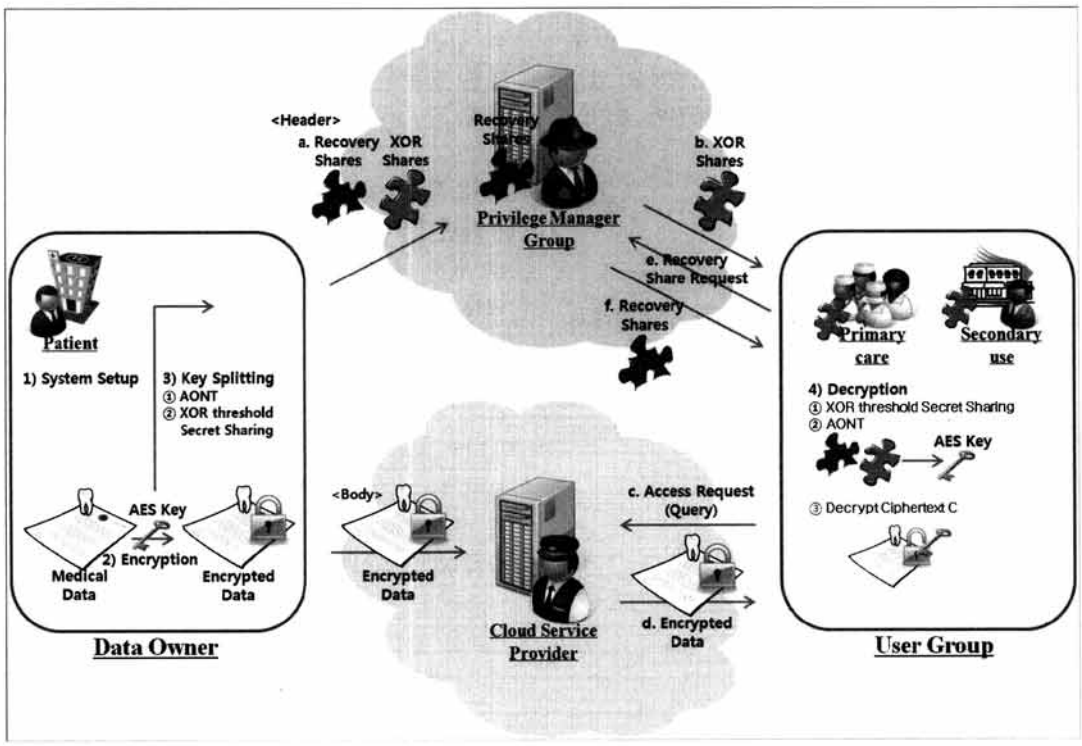
③ Key Splitting : AES키를 AONT 기반의 XOR (k,n) 임계치 비밀분산으로 분할하는 단계(설명을 위해서 $k=2$, $n=3$으로 지정)

a. AONT 변환

- AONT 변환을 수행하기 위해 AES키 k를 $k_1, k_2, ..., k_s (k_i \in \{0,1\}^{\ell}, i=1,...,s)$로 분할
- 해쉬함수 h를 이용해 $\mu_s = h(k_1 \| k_2 \| ... \| k_{s-1})$ 를 계산
- $\mu_s \oplus k_s$를 생성원함수 g를 이용해서 $g(\mu_s \oplus k_s)$를 계산
- $k_1 \| k_2 \| ... \| k_{s-1}$와 $g(\mu_s \oplus k_s)$를 XOR 연산해서 $x_1 \| x_2 \| ... \| x_{s-1}$ (Node_L$_i$(i=1,...,d))를 계산

  $x_1 \| x_2 \| ... \| x_{s-1} = (k_1 \| k_2 \| ... \| k_{s-1}) \oplus g(\mu_s \oplus k_s)$
- 해쉬함수 h를 이용해서 $h(x_1 \| x_2 \| ... \| x_{s-1})$를 계산
- $h(x_1 \| x_2 \| ... \| x_{s-1})$과 $(\mu_s \oplus k_s)$ 을 XOR 연산해서 x_s (Node_R$_i$)를 계산

$$x_s = (\mu_s \oplus k_s) \oplus h(x_1 \| x_2 \| ... \| x_{s-1})$$



(그림 6) 제안방식의 프로토콜 구성

· Node_L_i에 AONT 변환을 d회 반복 수행
· 마지막 d회 수행 후 생성된 모든 Node_R_i (i=1,...,d) (복원쉐어)는 권한 관리자 그룹에 저장
· 권한 관리자 그룹에 저장된 복원쉐어는 사용자가 데이터 복호 요구시 일괄적으로 전송

b. XOR (2,3) 임계치 비밀분산([그림 3])
· Node_L_d를 $K_0 \parallel K_1 (K_i = \{0,1\}^l (i=1,2))$로 분할
· 2개의 난수 $R_0, R_1 \left(R_i = \{0,1\}^l (i=1,2)\right)$을 생성
· K_0, K_1와 R_0, R_1의 XOR 연산을 통해서 다음과 같이 Node_$L_{d,i(i=1,2,3)}$를 생성

$$\text{Node\_}L_{d,1}=(K_0 \oplus R_0 \oplus R_1,\ K_1 \oplus R_1)$$
$$\text{Node\_}L_{d,2}=(K_0 \oplus R_0,\ K_1 \oplus R_0 \oplus R_1)$$
$$\text{Node\_}L_{d,3}=(R_0,\ R_1)$$

· Node_$L_{d,i}$를 각 사용자에게 전송

④ Decryption : AES키를 AONT 기반의 XOR (k,n) 임계치 비밀분산으로 복원하고 복원된 AES키로 암호문 C를 복호하여 데이터 m을 도출하는 단계

a. XOR 비밀복원
· 사용자가 임계치 k개 만큼의 Node_$L_{d,i(i=1,2,3)}$를 모아 XOR 연산을 통해서 아래와 같이 Node_L_d를 복원(설명을 위해서 Node_$L_{d,1}$과 Node_$L_{d,2}$를 사용)

$$K_0 = K_0 \oplus R_0 \oplus R_0$$
$$K_1 = K_1 \oplus R_0 \oplus R_1 \oplus R_0 \oplus R_1$$

· Node_L_d=$(K_0 \parallel K_1)$를 구성

b. AONT 역변환
· 사용자가 권한 관리자 그룹에게 인증된 사용자임을 승인받고 모든 Node_R_i(i=1,...,d)를 전송받아서 Node_L_d와 함께 AONT 역변환 과정을 d회 수행하여 AES키 복원

c. 암호문 C의 복호
· AONT 기반의 XOR 임계치 비밀분산으로 복원된 AES키로 암호문 C를 복호

3) 쉐어에 대한 관리

제안방식에서 권한 관리자 그룹은 신뢰할 수 있는 기관이다. 권한 관리자 그룹은 모든 복원쉐어를 보관하고 있고 일부의 XOR 쉐어도 소지하고 있다. 사용자가 암호문을 복호하기 위해서 비밀키를 복원하고자 할 경우, 권한 관리자 그룹은 사용자가 정당한 사용자인지를 판별하고 그에 따라 비밀키에 대한 접근 가부가 정해진다.

병원 시나리오를 예로서 설명하면 권한 관리자 그룹은 병원의 보안 관리자 그룹이고 사용자는 의사와 간호사 등이다. 환자 A가 AONT 기반의 XOR (2, 3) 임계치 비밀분산을 사용하여 복원쉐어와 3개의 XOR 쉐어를 생성하고 보안 관리자 그룹에게 모든 복원쉐어와 1개의 XOR 쉐어를, 나머지 2개의 XOR 쉐어를 각각 의사 B와 간호사 C에게 지급한다. C는 A에 대한 정보가 담긴 암호문을 복호하기 위해서 B의 동의를 얻어 XOR 쉐어를 조합한다. 또한 보안 관리자 그룹에게 비밀키 복원을 요청하고 동의를 거쳐 복원쉐어를

통해서 비밀키를 복원한다. 만약 B가 XOR 쉐어를 분실하였다면 보안 관리자가 소지하고 있는 XOR 쉐어를 이용할 수 있다. 모든 복원쉐어를 보안 관리자 그룹이 보관하고 있는 이유는 하나의 복원쉐어만 없어도 비밀키의 복원이 불가능하기 때문이다.

4. 분 석

본 장에서는 복호권한 분산관리, 용장성, 효율성 관점에서 기존 방식과 제안방식을 비교·분석한다. 그리고 제안방식의 XOR 쉐어를 활용한 새로운 기능인 권한의 가중치 부여에 대해서 검토한다. 아래의 <표 1>은 기존방식과 제안방식의 특징 비교표이다.

〈표 1〉 기존방식과 제안방식의 특징 비교표

특징	기존방식	제안방식
복호권한 분산관리	복호권한이 클라우드 서버에 집중관리	복호권한이 권한 관리자 그룹과 클라우드 서버에 의해 분산관리
용장성	사용자의 비밀키 소실에 무력하므로 용장성 결여	XOR 임계치 비밀분산을 활용하여 XOR 쉐어가 소실되어도 k개만 있으면 복원 가능하므로 용장성 확보
효율성	페어링 연산으로 제안방식에 비해 느린 암복호화	XOR 연산으로 기존방식에 비해 빠른 암복호화
권한의 가중치 부여	권한의 가중치 부여에 대한 개념을 제공하지 않음	XOR 쉐어 개수에 따른 권한의 가중치 부여 가능

4.1 기존 방식과의 비교

클라우드 컴퓨팅 환경에서의 안전한 키 분배와 암호문 공유 문제를 해결하기 위해서 제안된 [3]의 프로토콜 구성은 제안방식과 유사하다. [3]에서는 데이터를 암호화한 비밀키를 속성기반 암호방식(Attribute Based Encryption; ABE)으로 암호화하고 ABE의 비밀키를 사용자에게 전송하여 키 분배 문제를 해결하고 있다. 하지만 클라우드 서버가 암호문과 암호화된 비밀키를 모두 보유함으로써 복호권한의 집중화 문제가 있다. 이러한 집중화로 인해서 악의적인 사용자와 클라우드 서버간의 공모 공격에 취약하다.

제안방식은 복호권한인 data file을 header와 body로 분산하여 저장한다. 또한, AES로 데이터를 암호화하고 AONT 기반의 XOR 임계치 비밀분산으로 AES키를 분할·분산(d개의 복원쉐어, Node_L_d에 대한 n개의 XOR 쉐어)한다. 클라우드 서버에 암호문, 신뢰기관인 권한 관리자 그룹에 d개의 복원쉐어, 사용자에게 Node_L_d에 대한 n개의 XOR 쉐어를 각각 분산·저장한다. 권한 관리자 그룹은 정당한 사용자를 인증하여 암호문에 대한 복호권한을 제공한다. 즉, 복호권한의 집중화로 공모 공격에 대한 위협이 존재하고 사용자의 인증 절차를 기대할 수 없는 [3]의 프로토콜 구성보다 안전하다.

4.2 제안방식의 기능적 활용

제안방식은 XOR 임계치 비밀분산을 활용하여 Node_L_d의 XOR 쉐어 개수에 따른 권한의 가중치 부여가 가능하다. 예를 들어 병원 A에 '의사 이상의 권한이어야 데이터를 열람할 수 있다'는 복호정책이 있으면 XOR (3,4) 임계치 비밀분산을 활용하여 (그림 7)과 같이 XOR 쉐어를 배치할 수 있다. 병원장과 의사는 3개 이상의 XOR 쉐어를 소지하고 있어서 복호정책에 만족하므로 데이터를 열람할 수 있는 권한이 있다. 그에 반해 간호사, 약사, 간호조무사는 복호정책에 만족하지 못하는 XOR 쉐어를 소지하고 있으므로 3개 이상의 XOR 쉐어를 소지하기 위해서 다른 구성원과의 협력이 필요하다. 즉, 간호사가 간호조무사와의 협력을 통해서 복호정책을 만족할 수 있다. 이와 같이 XOR 쉐어 개수를 조절함으로써 권한의 계층화가 구현가능하다.

(그림 7) XOR 쉐어 개수에 따른 권한의 가중치

4.3 AONT 기반의 XOR 임계치 비밀분산의 특징

본 논문에서는 비밀키에 대해서 AONT를 반복 적용하여 데이터의 크기를 작게 분할하고, 그 중 하나의 쉐어(복원쉐어를 제외한 쉐어)에 대해서만 XOR 임계치 비밀분산을 적용한다. 기존 방식[3]에서는 페어링 연산 기반의 ABE와 PRE가 사용되었다. 페어링 연산보다 XOR 연산이 효율성 측면에서 월등히 우수하다는 것은 보편적으로 알려져 있는 사실이다. 또한, 비밀분산 방식만을 단독으로 사용하였을 경우 원래의 데이터에서 분할된 쉐어 개수의 배수만큼 저장공간이 비효과적으로 요구되지만 AONT 기법을 활용하면 분할될 데이터의 크기를 줄일 수 있기 때문에 효과적이다.

또한, [3]에서는 사용자가 소지하고 있는 ABE 비밀키의 소실에 따른 융장성 결여 문제가 있다. 다시 말해서 ABE 비밀키가 소실되면 재발급받아야 하며 이에 따른 계산상 과부하 문제가 예상된다. 기업 입장에서 보면 비밀키 재발급까지 서비스가 중단되므로 비즈니스 연속성 유지 실패와 시간적 비용 손실이 있다. 제안방식은 사용자가 소지하고 있는 Node_L_d에 대한 n개의 XOR 쉐어 중 임계치 k개만 존재하면 비밀키를 복원할 수 있다. 즉, XOR 임계치 비밀분산을 활용하여 n개의 XOR 쉐어 중 Node_L_d 복원에 필요한 k개의 XOR 쉐어 정보량에 따라 복원 가능성이 결정된다.

5. 결 론

본 논문에서는 클라우드 상에 신뢰할 수 있는 기관인 권한 관리자 그룹을 두어 접근권한 관리와 복호권한의 분산관리(data file를 header와 body로 분산하여 저장)를 통해서 공모공격에 대한 안정성을 보장하고 있다. 또한, 기존 방식과 달리 KP-ABE를 대신해서 AONT 기반의 XOR 임계치 비밀분산을 사용하여 키를 분산·저장하고 데이터 소유자가 분산한 복원쉐어(권한 관리자 그룹) 및 XOR 쉐어(사용자)를 사용자가 입수해야 암호화된 의료데이터를 복원할 수 있는 방식과 복호 권한에 대한 가중치 적용 가능한 프로토콜을 구성하였다. 4장에서 복호권한 분산관리, 용장성, 효율성 관점에서 기존 방식과 제안방식을 비교·분석하였고 제안방식의 XOR 쉐어를 활용한 새로운 기능인 권한의 가중치 부여에 대해서 검토하였다. 이를 통해서 제안방식이 기존 방식에 비해 보다 안전(데이터의 기밀성을 보장)하고 XOR 비트 연산으로 빠른 연산이 가능함을 정성적으로 보였다.

본 논문의 연구결과는 개인의 프라이버시 정보를 담고 있는 의료데이터, 기업의 주요 정보 등의 대용량 데이터를 안전하고 효율적으로 관리하는 클라우드 서비스 환경에서의 이용이 기대된다. 향후 과제로써 효율성 측면에서 실제 구현을 통해 기존 방식과 제안방식의 비교분석을 통한 증명이 필요하다. 또한, 환자가 자신의 의료데이터를 직접 제어할 수 있는 환자 중심의(Patient-controlled) 접근제어에 대한 연구가 필요하다[12][13].

참 고 문 헌

[1] Amazon, http://aws.amazon.com/s3/ (2010.)

[2] 미디어다음, http://media.daum.net/breakingnews/view.html?newsid=20110418211512938/ (2011.)

[3] S. C. Yu, C. Wang, K. I. Ren and W. J. Lou, "Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing," INFOCOM, 2010 Proceedings IEEE, pp.321-334, 2010.

[4] V. Goyal, O. Pandey, A. Sahai and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," inProc. Of CCS'06, pp89-98, 2006.

[5] M. Blaze, G. Bleumer and M. Strauss, "Divertible protocols and atomic proxy cryptography," In Advances in Cryptology. EUROCRYPT'98, volume 1403 of LNCS, pp.127-144, 1998.

[6] R. L. Rivest, "All-or-nothing encryption and the package transform," Fast Software Encryption FSE '97, Lecture Notes in Computer Science, Vol.1267, pp.210-218, 1997.

[7] 桑門秀典, 神戸大學, "暗号システムの安全性を向上させる暗号化モードに關する研究(継續)," 電氣通信普及財団, 研究調査報告, No.19, 236, 2004.

[8] 石津晴崇, 荻原利彦, "電子データの長期保管に關する一考察," 電子情報通信學會2004年總合大會講演論文集, D-9-10, 2004.

[9] A. Shamir, "How to Share a Secret," Communication of the ACM, Vol.22, No.11, pp.612-613, 1979.

[10] 송유진, 박광용, "대용량 e-비즈니스 데이터 분산 보안관리 모델", e-비즈니스연구, 제11권 제1호, pp.325-342, 2010.

[11] J. Daemen and V. Rijmen, 'AES Proposal: Rijndael', AES Algorithm Submission, 1999.

[12] 長澤 悠貴, 毛利 公美, 福田 洋治, 白石 善明, 岩田 彰, "グループ秘密鍵の分散管理によるPHR の医療消費者主導型開示先制御," SCIS2011, 2011.

[13] 須賀祐治, "クラウド環境に適した秘密分散共有法の初期検討", SCIS2011, 2011.

송 유 진

e-mail : song@dongguk.ac.kr

1982년 한국항공대학교 전자공학과(학사)

1987년 경북대학교(공학석사)

1995년 일본 Tokyo Institute of Technology(공학박사)

1988년~1996년 한국전자통신연구원 선임연구원

2003년~2005년 미국 University of North Carolina at Charlotte 연구교수

2006년~2006년 일본 정보보호대학원대학 객원교수

1996년~현 재 동국대학교 정보경영학과 교수

2005년~현 재 동국대학교 부설 전자상거래연구소장

1998년~현 재 한국정보보호학회 부회장(영남지부장)

2006년~현 재 국제 e-비즈니스학회 이사

2006년~현 재 한국사이버테러정보전학회 이사

2001년 ICISC2001 운영위원장

2003년 하계CISC2003 프로그램위원장

2006년 CISC-S2006 공동프로그램위원장

2007년 한국정보시스템학회 추계학술발표대회 공동 조직위원장

관심분야 : Privacy Protection, Secret Sharing, 전자상거래응용보안(Location Privacy, 디지털컨텐츠 보호, SCM/CRM 보안 등), Context Aware Application Security 등

도 정 민

e-mail : havdrim@hotmail.com

2009년 동국대학교 정보경영학과(학사)

2010년~현 재 동국대학교 전자상거래 협동과정 석사과정

관심분야 : 정보보호, 암호이론(Secret Sharing, Attribute-Based Encryption), Context Aware Application Security 등