한국산학기술학회논문지
Vol. 8, No. 3, pp. 528-534, 2007

Peer-to-Peer 네트워크 상에서 XML 데이터의 효율적이고 안전한 배포 방식에 관한 연구

고혁진[1*], 강우준[2]

An Efficient Secure Dissemination of XML data in Peer-to-Peer Networks

Hyuk-Jin Ko[1*] and Woo-Jun Kang[2]

요 약 XML이 인터넷 상에서 수많은 정보의 표현과 교환의 표준으로 자리매김해감에, XML 데이터를 안전하고 효율적으로 배포하기 위한 방법의 강구가 강력히 요구 되는 추세이다. 특히 Peer-to-Peer와 같은 환경에서 그런 경향은 더욱 두드러지고 있다. 지금까지의 SDI 연구는 user profiling 에 대해 새로 입수되는 XML source에 대한 match (filtering 문제)에 관련된 문제에 집중, 효율적 배포와 관련된 연구가 드문 실정이며, 효율적 배포에 관련된 소수의 기존연구에서도 중앙집중식 관리방식을 사용함으로써 Peer-to-Peer와 같은 분산환경에는 바로 적용시키기가 어렵다. 본 논문에서는 이러한 특징을 가지고 있는Peer-to-Peer 환경에서 인가정책과 비밀분산을 이용함으로써 안전하고 확장이 용이한 XML 데이터의 배포 방법을 제안한다.

Abstract As XML is becoming a standard for representation and exchange of abundant information on the Web, solutions for a secure and selective dissemination of XML data, known as SDI, are strongly demanded. Such trends are more outstanding especially in distributed heterogeneous environment such as Peer-to-Peer. Although many approaches have been proposed to provide secure and efficient SDI mechanisms, almost previous approaches have focused only on filtering with user profile and they adopt center-oriented administration approaches. It is therefore difficult to adapt them directly to the distributed Peer-to-Peer environments characterized by dynamic participation. In this paper, we develop a novel dissemination method, which makesuse of authorization policy and secret sharing scheme. It provides more secure, scalable means for XML dissemination on Peer-to-Peer networks.

Key Words: XML, SDI(Selective Dissemination of Information), Secret Sharing, Peer-to-Peer Networks

1. 서론

XML 은 인터넷 상에서 정보교환의 수단으로 표준이 되어가고 있으며, XML 데이터를 안전하고 선택적으로 배포하는 선택적 정보 배포(SDI: Selective Dissemination of Information)에 대한 기술개발 요구도 증가해 가고 있는 추세이다. 특히 웹 서비스나 Peer-to-Peer(P2P)[1-3]와 같은 분산되고 개방된 환경에서 그러한 증가 추세는 더욱 두드러지고 있다. 현재까지의 SDI 연구는 사용자 프로파일링(profiling)에 대해 새로 입수되는 XML 소스에 대한 매칭(또는 필터링)에 관련된 문제에 집중되어 정작 효율적 배포 방식에 관련된 연구가 드문 실정이며, 관련 소수의 기존연구에서의 방법도 중앙집중식 관리방식을 사용함으로써 P2P와 같은 분산환경에 바로 적용하기가 쉽지 않다.

본 논문에서는 피어(peer)의 참가/탈퇴가 빈발하며, 중앙집중식 관리가 어려운 특징을 가지고 있는 P2P분산환경에서 안전(secure)하고 확장성(scalable)을 갖는 XML데이터의 배포 방법을 제안한다. 제안방법의 핵심은 접근제어의 인가정책 (authorization policy) 을 이용함으로써 사용자 인증 타당성 조건(authentication verification condition)[4] 검사를 미리 정해놓은 인증규칙에 의해 정적으로 수행하는 것이 아니라, 인가 정책에 포함되는 사

[1]성균관대학교 컴퓨터공학과
[2]그리스도대학교 경영정보학부
*교신저자: 고혁진(hjko@skku.edu)

용자 신임(credential) 정보를 이용하여 네트워크에 유입되는 사용자에 대해 동적으로 사용자 인증 타당성 조건 검사를 수행하도록 하는 것이다. 이 방식을 사용함으로써 피어의 빈발한 참가/탈퇴 가 허용되는 환경에서도 확장성을 보장할 수 있으며, 정교한 보안설정 전략을 수립 가능하게 한다. 또한 키 관리에 있어 비밀분산(Secret Sharing)을 적용함으로써 관리되는 키의 개수를 줄이는 방식에 대해서도 설명한다.

2. 관련연구

Miklau[5]와 Bertino[6]는 공통적으로 XML 데이터 배포를 위해 암호를 사용하는 접근제어 방식을 취하였으나, 접근제어 정책을 기술하기 위해 Bertino는 인가모델을 사용한 반면 Miklau는 XQuery 언어를 확장하는 방식으로 접근하였다. 위의 두 가지 접근 방법 모두에서, 암호화 키를 사용하여 데이터의 일부분을 암호화한 후, 암호화된 XML 데이터를 사용자에게 배포한다. 사용자가 획득할 수 있는 데이터는 그가 보유하고 있는 키에 의해 복호화할 수 있는 데이터와 암호화 되지 않고 배포된 데이터로 제한된다. 또한 Miklau와 Suciu[13]는 XML 데이터 중 'EncryptedData' 태그 엘리먼트를 사용자가 소유하고 있는 키들로 우선적으로 복호화한 후, 복호화된 데이터 엘리먼트들이 제시된 쿼리에 부합되는지를 검사하는 쿼리 처리 방식을 제안하기도 하였다. 이외에도XML 데이터 배포를 위한 다양한 접근방식들, 사용자 프로파일에 대한 필터링 방식[7]이나 암호화된 데이터베이스에 대한 효율적인 인덱싱 방식[9] 그리고 비밀정보가 침입자에 게 비밀정보에 대한 단서를 제공할 수 있는 QI-속성(Quasi-Identifier attribute)을 제거하기 위한 일반화(generalization)[10]와 변환(transformation) [8] 방식 등의 연구가 있어왔다. 하지만 이런 대부분의 연구들은 미리 정해진 인증규칙에 따라 정적으로 타당성 검사를 수행하며, 중앙집중식으로 관리되므로Peer-to-Peer 환경에는 적합하지 않다.

3. 배경지식

3.1 Peer-to-Peer 네트워크

Peer-to-Peer(P2P)는 네트워크상의 모든 노드가 그들이 수행하는 기능이나 작업에 있어 완전히 동등한 분산환경이다. P2P 구조는 다음과 같은 중요한 성질을 갖는다. 첫째로 중앙서버의 중재를 요청하지 않고 직접 교환에 의해 컴퓨팅 자원을 공유한다. 때로 중앙서버는 새로운 노드가 네트워크에 첨가될 때, 데이터 암호화를 위한 키의 획득 등에 있어 특정한 작업을 담당하기도 한다. 둘째로 노드의 빈발한 참가/탈퇴에 의한 연결상의 불안정(instability)을 관리하며, 네트워크 연결상의 오류 시 자발적(autonomous) 이면서 순응적으로 (proactive) 적응한다. 끝으로 협업(collaboration) 환경에서 작업 그룹에 대한 유연한 관리를 할 수 있게 함으로서 협업비용을 줄일 수 있게 하며, 의사전달 처리를 가속화 시킬 수 있도록 한다.

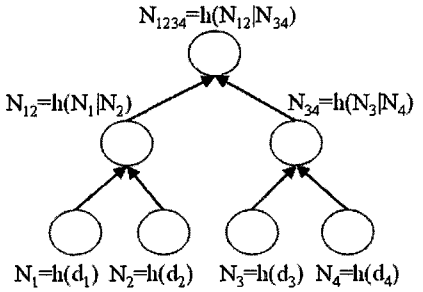

그림 1. Merkle 해쉬 트리의 예

Peer-to-Peer 네크워크를 구성하는 각 피어들은 데이터와 Merkle 해쉬[4] 값으로 구성되는 해당 데이터에 대한 인덱스를 함께 저장한다. 그림1은 Merkle 해쉬 트리의 구성 예를 보여주고 있다. 트리의 단말 노드에서는 데이터 파티션 d1 ~ d4의 해쉬 값을 계산하여 저장하며, 윗 단계의 부모 노드들에서는 자식 노드의 해쉬 값들을 연결하여 새로운 해쉬 값을 계산한다. 이러한 인덱스 정보는 다른 피어들에게 전달되어 그 인덱스에 인해 참조되는 데이터가 어떤 피어에게 저장되어 있는지의 위치 정보를 나타낸다. 또한 이러한 인덱스는 신뢰되는 피어에게 전달되기도 하여 계층적인 위치 정보로 관리되기도 한다. 데이터 검색을 위해서는 요청 피어의 신임 정보와 요청된 데이터의 인덱스 값들 그리고 타임아웃 스탬프를 포함하는 요청 메시지를 네트워크에 전송한다. 해당 인덱스 값을 보유하거나 참조할 수 있는 피어들은 요청 피어의 신임 정보를 검사하여 적절한 응답 메시지를 전송한다.

3.2 비밀분산 (Secret Sharing)

최근 정당하지만 정보를 독점하는 관리자에 의한 부정이용이 개인정보 노출의 가장 큰 원인이 된다는 사실이 점점 알려지고 있다. 관리자가 사용자의 기밀 문서를 몰래 복사하거나 고쳐 쓰는 것이 용이하기 때문에 이와 같이 접근권한을 한 곳에 집중시키면 관리는 쉽지만 그에

따라 보안 위험은 커지게 된다. 이러한 문제를 해결하기 위한 비밀분산은 하나의 비밀정보를 여러 개의 비밀조각 즉 구성키(share)로 분할시켜 다수의 참가자들에게 공유시키고 필요 시 참가자들의 합의에 의해 비밀정보를 복원하도록 하는 고급 암호 프로토콜이다. 비밀정보의 분할은 다항식과 같은 수학적 방법을 이용하여 마치 물리적으로 분할된 것과 같은 효과를 내는 논리적 분할을 의미한다.

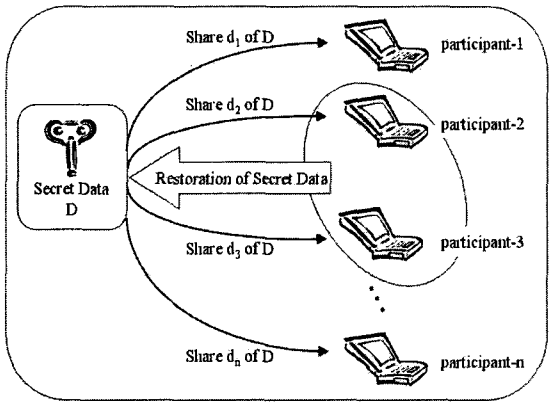

그림 2. Shamir의 (2,n)-임계치의 예

가장 대표적인 비밀분산은 Shamir가 제안한 (t,n)-임계치 [11]이다. 이 비밀분산은 비밀정보를 *n*개의 비밀조각으로 분할하고 그 중에서 임의의 *t*개 이상의 비밀조각을 모으면 원래 비밀정보를 복원할 수 있지만, *t*개 미만의 비밀조각으로는 비밀정보를 복원할 수 없게 한다. 그림2는 (2,n)-임계치의 개념을 보여준다.

3.3 접근제어(Access Control)

인가는 다음의 3-튜플 <S,O,P>로 표현되는데, 이 때 S는 시스템의 주체, Object O는 목표데이터, P는 목표데이터에 대한 접근권한을 나타낸다[13][14][15]. P는 다시 2-튜플 <sign, mode>로 세부 정의되며, 여기서 sign = {positive+, negative-} 이고, mode={create, delete, read, write }이다. S를 인증 확인하기 위해서 신임 정보를 이용한다. 신임정보로 S의 문맥 정보 즉, 접근IP 주소와 위치, 시간 등을 이용함으로써 해당 주체가 정당한 사용자임을 확인할 수 있다.

접근제어 정책은 인가의 집합이며 다음과 같이 분류된다[17]. 결정(decision) 정책은 인가 평가 시 개방세계(open-world) 전략을 따를 것인지 폐쇄세계(close-world) 전략을 따를 것 인지를 결정할 수 있도록 하며, 충돌해결(collision resolution) 정책은 인가 정책 간의 충돌 발생 시 금지 우선(negative take-precedence) 전략으로 충돌을 해결할 것인지 허용 우선(positive take-precedence) 전략으로 해결할 것인지를 결정하며, 전파(propagation) 정책은 주체 계층 구조 또는 데이터 계층 구조 상에서 권한 전파 전략을 전파 허용으로 할 것인지 전파 불가로 할 것인지를 결정한다. 제안 시스템에서는 결정 정책으로는 폐쇄세계 전략을, 충돌해결정책으로는 금지 우선 전략을, 전파정책으로는 전파허용전략을 기본으로 한다.

4. XML 데이터의 배포

4.1 기존 방식의 문제점

기존의 SDI 시스템에서는 XML 데이터를 보안 전략에 따라 여러 부분으로 나누고 각 부분에 대해 키를 할당함으로써 해당 키를 보유하는 사용자들만 문서에 접근할 수 있도록 한다. 네트워크 상에서 데이터의 안전한 전송을 위해 데이터는 키로 암호화되며, 수신 피어 에서는 해당 키(키 공유방식에 따라 대칭키 또는 비대칭 키가 될 수 있다)를 가지고 데이터를 복호화 한다. 키는 정적으로 정의된 인증규칙에 따라 설정되어 당사자들에게 분배되므로 피어의 참가와 탈퇴가 빈번한 P2P 환경에서는 이런 방식을 적용하기가 매우 어려워진다.

표 1. 기존 방식에서의 암호화된 데이터 인덱스

Key Id	Partition (XPath)
{ k1 }	d1: /hospital/patient
{ k1, k2 }	d2: /hospital/insurance
{ k1, k3 }	d3: /hospital/treat
{ k4 }	d4: /hospital /patient[/SSN="xxx"]

표 2. 신임 프레디킷으로 확장된 암호화된 데이터 인덱스

Subject credential predicate	Key Id	Part.
c1: IP(subject) in Medical_Domain	{ k1 }	d1
c2: Dept(subject)=Finance_Dept	{ k1, k2 }	d2
c3: Role(subject)=Doctor	{ k1, k3 }	d3
c4: Role(subject)= Manager	k4	d4

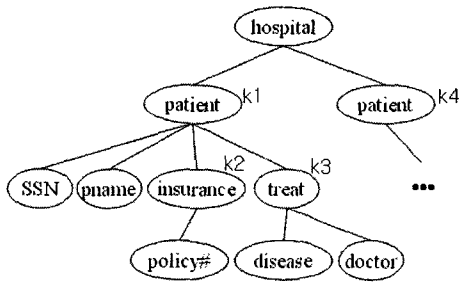

그림 3. XML 데이터 파티션의 예

표1은 기존방식으로 구성 되는 암호화된 데이터 인덱스이다. 그림3과 같이 나뉘어진 파티션에 대한 XPath 표현[16] (XML 데이터의 파티션은 XPath 표현으로 명시될 수 있다)과 키를 명시한 것이다. 그림에서 *hospital*의 두 번째 자식 *patient*는 특별한 환자의 정보이므로 첫 번째 자식 *patient*와는 다른 보안 정책이 적용 되었다. 본 연구에서는 이와 같은 정적인 사용자 인증 타당성 조건 검사 문제를 해결하기 위해 인가구조를 사용한다. 주체 신임 프레디킷(predicate)를 이용한 검사를 수행함으로써 이를 가능하게 한다. 즉, 특정 데이터를 요청하는 피어들은 자신의 신임 프레디킷을 제공함으로써 인증될 수 있는 것이다. 표2는 주체의 신임 프레디킷이 첨부된 암호화된 데이터 인덱스를 보여 준다. 또한 개인 정보 노출의 주요 원인이 되는 권한 집중을 방지하기 위해서 비밀분산 방식이 사용된다. 우리가 알고 있는 한, 분산 키의 구성 키들을 다수의 비밀분산 키 복원에 사용할 수 있도록 하는 방식이 가능할 것이라 추측되며 이에 대한 이론적 연구를 수행할 계획이다. 현재의 아이디어는, (n,K)-임계치 비밀분산체계에 있어 k-1 차 다항식의 미지 상수 계수(상수 계수의 값은 비밀 분산키가 된다)를 계산하는 데 필요한 K개의 구성 키를, K-1차 이하의 다항식의 미지 상수 계수를 알아내는 데에 있어 재 이용할 수 있다 라는 것이다. 물론 이 다항식은 비밀분산에 사용되는 다항식의 계수 성질을 계속 유지 하고 있어야 하며, 비밀분산의 보안 정책에 따라 최 하차 차수를 결정할 수도 있어야 한다. 현재 이론적 증명을 통한 명확한 결과를 제시하지는 못하지만 이 방식을 이용함으로써 관리 키의 개수를 줄일 수 있을 것이라고 사료된다.

4.2 파티션, 메타 키 인덱스, 블록

이번 절에서는 앞에서 설명한 인가정책과 비밀분산이 어떻게 정보 배포에 사용되는지를 설명한다. 여러 파티션으로 나뉘어진XML 데이터들의 안전한 배포를 위해 부가적인 인가정보와 키의 해쉬 값 정보를 묶어 meta-key index를 정의한다.

정의 1. 파티션 (Partition)

정보 소유자의 보안정책에 의해 분할된 데이터를 파티션이라 한다. 분할된 데이터들은 XPath 표현에 의해 명시된다.

예제 1.

그림 3의 각 파티션을 표현하는 XPath 표현 d1, d2, d3, d4가 표2의 3번째 필드에 예시되어 있다.

정의 2. 메타 키 인덱스 (Meta-Key Index)

메타 키 인덱스는 3-튜플 <*policy, kid, Merkle-해쉬값*>로 구성되며, *policy* 필드는 데이터 파티션 별로 명시하여, 어떤 주체가 해당 파티션에 대해 어떠한 연산을 수행할 수 있는지에 대한 접근 권한을 명시한다. K_{id} 필드는 전송 상에서의 기밀성 보장을 위해 각 파티션 별로 할당된 키 식별자 리스트이며, 해당 키 값을 가지고 파티션을 암호화한다. *Merkle-해쉬값* 은 K_{id}의 해쉬 값으로 구성되는 리스트이다.

표 3. 블록 구성의 예

Block	Meta-key index			Partition
	policy	kid	*Key Hash	
block 1	p1	{ k1 }	H(k1)	d1
block 2	p2	{ k1, k2 }	H(k1), H(k2)	d2
block 3	p3	{ k1, k3 }	H(k1), H(k3)	d3
block 4	p4	{ k4 }	H(k4)	d4

* 'Key Hash' 컬럼에서 H(ki) = MkH(Value(ki))

*MkH()*는 매개변수로 키 값을 취하여 해당 키 값의 해쉬 값을 반환하는 함수이며 Value()는 키 식별자 인자에 대한 키 값을 반환하는 함수이다. 해쉬 값은 마치 디렉토리 인덱스와 같이 해당 키 식별자에 대한 키 값을 찾을 수 있도록 한다. 데이터 인덱스를 위한 해쉬 값은 Peer-to-Peer 네크워크의 기본 기능으로 제공되나, 키 인덱스를 위한 해쉬 값은 제공하지 않으므로 메타데이타에 포함시켜 이를 처리한다. 이와 같이 키 값을 찾기 위한 추가 절차가 필요하나 보다 안전한 배포를 보장함으로써 이러한 단점을 극복한다.

예제 2.

그림3과 표2에서 k3으로 암호화되는 파티션 d3에 대한 인가정책은 <c3,d3,<+,read>>로 의사로 식별되는 주체

에게 파티션 d3에 대해 읽기 연산을 허가한다. d3의 암호화, 복호화를 위한 키 식별자는 k3이며 MkH(Value(k3)) 값을 가지고 키 값을 찾을 수 있도록 한다. 표4에서의 k4는 k1, k2, k3로 구성되는 (2,3)-임계 비밀분산 키이다.

정의 3. 블록 (Block)

메타 키 인덱스와 암호화된 파티션으로 구성되며 네트워크 상에서의 요청과 응답에 대한 논리적 전송단위가 된다.

예제 3.

그림3 과 표2를 토대로 구성되는 블록들은 표3 와 같다. 표2의 신임정보와 파티션을 참조하여 인가정책을 예시하면,

p1 = < c1, d1, <+, read> >
p2 = < c2, d2, <+, read/write> >
p3 = < c3, d3, <+, read/write> >
p4 = < c4, d4, <+, read> > 와 같이 구성할 수 있다.

여기서 주목할 점은 블록 전송 시 키 값의 누출을 방지하기 위하여 키 식별자만을 포함하고 키 값은 포함하지 않는다는 사실이다. 키 값을 얻기 위해서는 키 식별자와 그에 해당되는 해쉬 값이 필요하다. 기본 전제 조건으로, 보유하고 있는 데이터나 키의 해쉬 값을 다른 피어들에게 제공하여 데이터나 키 값을 보유하고 있는 피어의 위치를 알려 줄 수 있도록 서비스 디렉토리 기능이 존재한다고 가정한다. 디렉토리의 기능 구성 방식은 신뢰 피어가 존재하지 않는 순수 P2P와 존재하는 하이브리드 P2P에서의 구성이 각각 다를 수 있다. 자세한 사항은 [1]을 참조 바란다. 예제 3에서 k4는 k1, k2, k3로 구성되는 (k,n)-임계 비밀분산 키(단, k=2,n=3)이다. K4 키 값을 복원하기 위해서는 k1, k2, k3 개의 구성 키 값 중 2개 이상만 획득하면 된다. k4의 해쉬 값은 트리 구조 상에서k1, k2, k2의 해쉬 값을 갖는 노드들을 자식으로 갖는다. 인가된 피어는 정보 소유자에 의해 제공되는 이러한 키 식별자, 키 값, 해쉬 값을 획득하여 저장하고, 저장된 정보를 이용함으로써 다른 피어의 요청에 응답할 수 있게 된다.

4.3 네트워크상에서 메타정보의 처리

이번 절에서는 메타정보가 어떻게 Peer-to-Peer 네트워크 상에서 XML데이터의 안전한 배포를 위해 사용되는지를 그림 4의 네트워크 구성을 들어 설명한다. 또한 메타정보를 사용하는 배포를 위한 두 개의 알고리즘을 보여준다. 정보 생성자는 피어 S이고, 사용자 정의 설정 정보를 통해 데이터를d1 ~ d4의 4개의 파티션으로 분할 구성한 후, 이에 대한 메타 키인덱스를 표 2에서와 같이 구성한다. 여기서 d4는 비밀분산 키 K4가 적용되는 파티션이며 d4의 복호화를 위해서는 k1, k2, k3 중 2개 이상을 보유하여야 한다. 초기 상태의 신뢰 피어는 어떠한 정보도 가지고 있지 않다. 파티션 d1을 S의 인접 피어 A 에서 요청 시(요청 메시지에는 데이터 파티션에 대한 식별자와 자신의 신임정보가 포함 된다), S 에서는 A의 신임 프레디킷을 요청된 d1의 인가정책 p1를 가지고 평가한 후, 접근이 허가되면 블록 b1을 A로 전송한다. A 에서는 수신된 블록에 포함된 k1의 해쉬 값을 이용하여 k1의 키 값을 가지고 있는 피어들을 검색한다. 현재는 S에서만 k1의 키 값을 가지고 있으므로 S로부터 k1키 값을 획득 후 암호화된 d1을 복호화 한다. A는 보유하게 된 k1에 대한 해쉬 값을 저장하고 신뢰 피어에게 전송한다. 신뢰 피어는 k1을 보유하는 피어가 A임을 나타내는 정보가 저장된다. 이후 다른 피어 들의 d1 요청에 대해 S 뿐만 아니라 A도 응답할 수 있게 된다. 이와 같이 인가정책에 포함되어 있는 신임 정보를 이용하여 해당 파티션에 대한 접근허가를 결정할 수 있다.

알고리즘 Evaluation

Input: 파티션 id, credential
Output: If 접근이 허용되면 블록 전송
Else 접근 거부
1 요청되는 데이터 파티션 식별자를 이용하여 관련 블록 b_i를 결정하고, 인가정책 p_i를 선택한다.
2 요청 피어의 신임 정보를 이용하여 인가정책 p_i 상에서 평가한다.
2.1 결과가 "허용" 이면 블록 b_i을 요청 피어에게 전송한다.
2.2 결과가 "불가" 면 거부 메시지를 전송한다.

알고리즘 Decryption

Input: 블록 (meta-key 인덱스와
암호화된 데이터 파티션 식별자 di)
Output: 복호화된 데이터 파티션
1 블록에서 meta-key 인덱스 구성정보 p_i와 k_{id} 그리고 k_{id}의 해쉬 값 $H(k_{id})$를 추출한다.
2 $H(k_{id})$를 이용하여 k_{id}의 키 값을 제공하는 피어를 검

색한다.

3 검색된 피어로부터 키 값 value_k_{id} 을 얻는다.

4 value_k_{id} 을 사용하여 데이터 파티션 d_i를 복호화 한다.

5 k_{id}, $H(k_{id})$, value_k_{id} 값을 저장하고 다른 피어가 요청 시 정보를 제공한다

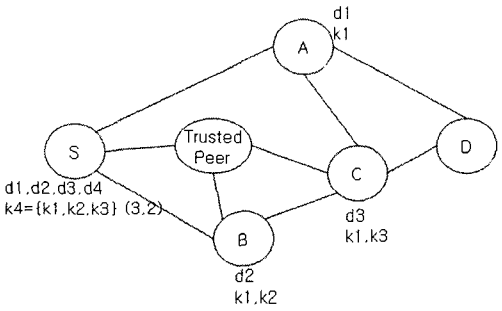

그림 4. 네트워크 구성의 예

다음은 비밀분산의 방식 처리 예를 살펴 보자. 일정 시간 경과 후, 데이터 생성자 S가 아닌 A, B, C가 그림 4에서 처럼 각각 k1, k2, k3를 포함하는 블록 b1, b2, b3을 획득했다고 가정하자. 신뢰 피어에는 A,B,C 에서 제공한 키의 해쉬 값이 저장되어 있어 k1, k2, k3 키 값을 포함하는 피어가 각각 A, B, C 임을 나타내고 있다. 이때 D에서 비밀분산이 적용되는 데이터 파티션 d4를 요청할 시, 인접 피어인 A, B, C에서는 인증 검사의 결과가 '허용' 이 아니므로 각자 자신의 인접 피어 에게 요청을 중계하게 되고, 결국 S가 D의 요청을 받게 된다. S는 D의 신임 프레디킷을 평가하고 접근이 '허용' 되면 블록 b4 를 D에게 전송하게 된다. D는 d4을 복호화 하기 위한 키 식별자가 k4임을 확인 하고, k4의 키 해쉬 값을 가지고 검색한다. 검색 결과로 k4가 구성키 k1, k2, k3로 복원되는 비밀분산 키라는 것과 k1, k2, k3 구성 키 값을 저장하고 있는 피어가 A, B, C 임을 확인 후 (피어 S로 부터 k1, k2, k3를 획득할 수도 있지만 가장 근접한 피어가 우선 응답한다), 구성 키 값들 중 2개 이상을 획득하게 되면 k4 키 값을 구성하고 이 키 값을 가지고 데이터 파티션 d4를 복원한다. 4.1절에서 설명했듯이 특정 비밀분산 키의 구성 키를 다른 비밀분산 키의 구성 키로 재 사용할 수도 있으므로 네트워크 상에서 관리되어야 하는 키의 총 개수를 줄일 수 있을 것이다.

5. 결론

본 논문에서는 Peer-to-Peer네트워크와 같이 구성 피어의 참가, 탈퇴가 빈발하고 중앙 집중식 관리가 비 효율적인 분산환경에서, XML데이터를 안전하고 확장성 있게 배포 할 수 있도록 하는 방법을 제안하였다. 제안 방법에서는 인가 정책을 이용함으로써 정적 사용자 인증 검사 대신 동적인 인증 검사를 수행할 수 있게 하여, 빈발한 피어의 참가/탈퇴 경우에도 확장성을 보장할 수 있도록 하였으며, 비밀분산 적용 시 구성 키를 다른 비밀분산 키의 구성키로 재 사용할 수도 있을 것이라는 아이디어를 제시하였고, 이를 바탕으로 관리 키의 개수 또한 줄일 수 있을 것이라고 추측하였다. 향후 과제로는 비밀분산 구성 키의 재 이용에 관한 이론적 토대에 관한 연구를 수행할 계획이며 XACML과 SAML을 제안하는 시스템에 적용시켜 보다 표준화된 방식으로 메타 키 인덱스를 정의하는 방식과, 해쉬 키 기반의 보다 효율적인 키 검색 방법에 대해 연구할 계획이다.

참고문헌

[1] S. Androutsellis-Theotokis and D. Spinellis, "A Survey of Peer-to-Peer Content Distribution Technologies," *ACM Computing Surveys*, Vol. 36, No. 4, 2004.

[2] I. Foster and A. Iamnitchi, "On death, taxes, and the convergence of peer-to-peer and grid computing," *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems*, IPTPS'03, 2003.

[3] D. Schoder and K. Fischbach, "Peer-to-peer prospects," CACM , 46, 2 , 2003.

[4] R. C. Merkle, "A Certified Digital Signature," *Advances in Cryptology*, Crypto'89, 1989.

[5] G. Miklau and D. Suciu, "Controlling Access to Published Data Using Cryptography," Proc. *Of the 29th VLDB Conference*, 2003.

[6] E. Bertino and E. Ferrari, "Secure and Selective Dissemination of XML Documents," *ACM Transactions on Information and System Security (TISSEC)*, 5(3), 2002.

[7] H. Wang and L. Lakshmanan, "Efficient Secure Query Evaluation over Encrypted XML Database," *VLDB'06*, 2006.

[8] X. Xiao and Y. Tao, "Anatomy: Smiple and Effective Privacy Preservation," *VLDB'06*, 2006.

[9] T. W. Yan and H. Gracia-Molina, "The SIFT Information Dissemination System," *ACM Transactions on Database Systems,* Vol. 24, No. 4, 1999.

[10] M. Altiel and M. Franklin, "Efficient Filtering of XML Documents for Selective Dissemination of Information," *Proceedings of the 26th VLDB Conference*, 2000.

[11] A. Shamir, "How to Share a Secret," *CACM*, Vol. 22, 1979.

[12] G. J. Simmons, "How to (Really) Share a Secret," *Advances in Cryptology -CRYPTO'88, Lecture Notes in Computer Science*, Vol. 403, 1990.

[13] R. S. Sandhu and E. J. Coyne and H. L. Feinstein and C. E. Youman, "Role Based Access Control Models," *IEEE Computer*, Vol. 29, No. 2, February 1996.

[14] R. Sandhu and D. Ferraiolo and R. Kuhm, "The NIST Model for Role-Based Access Control: Towards A Unified Standard," *Proceedings of the fifth ACM workshop on Role-based access control*, 2000.

[15] E. Bertino and S. Castano and E. Ferrari, "Securing XML Documents with Author-X," *IEEE Internet Computing*, Vol.5, No.3, 2001.

[16] J. Clark and S. DeRose, "XML Path Language (XPath) Version 1.0," *W3C Recommendation*, 1999.

[17] S. Jajodia and P. Samarati and M. L. Sapino and V. S. Subrahmanian, "Flexible support for multiple access control policies,". *ACM Trans. Database System*, Vol.26, No.2, 2001.

[18] S. Boag et al, "XQuery 1.0: An XML Query Language," *W3C Working Draft*, 2003.

고 혁 진(Hyuk-Jin Ko) **[정회원]**

- 1991년 2월: 성균관대학교 정보공학과(학사)
- 1994년 2월: 성균관대학교 정보공학과(석사)
- 1994년~2002년: (주) LG전자 연구소
- 2003년 3월~현재: 성균관대학교 전기전자 및 컴퓨터공학부 박사과정

<관심분야>: 데이터베이스 보안, 개인정보 보호, 데이터 마이닝, 문맥인지, 시멘틱 웹

강 우 준(Woo-Jun Kang) **[정회원]**

- 1984년 2월 연세대학교 전자공학과 (공학사)
- 1984년~ 1999년 한국 IBM S/W 연구소
- 1992년 연세대학교 전자계산학 (공학 석사)
- 1994년 연세대학교 경영정보학 (경영학 석사)
- 1999년 3월 ~ 2001년 2월 안산공과대학 교수
- 2001년 성균관대학교 컴퓨터 공학 (공학박사)
- 2001년 3월 ~ 현재 그리스도대학교 경영정보학부 교수

<관심분야>: XML/Web 마이닝, 접근제어, 전자상거래보안, DRM