Journal of Korea Multimedia Society Vol. 14, No. 10, October 2011(pp. 1303-1310)

인프라 클라우딩(Infra Clouding) 환경에서 자가조직 저장매체의 보안을 위한 3자간 협상 프로토콜 설계

이병관†, 정은희††

요 약

본 논문은 인프라 클라우딩 환경에서 데이터를 소유한 소유자 노드와 데이터를 보관하는 보관 노드 그리고 데이터를 검증하는 검증 노드로 구성된 자가 조직 저장 매체 보안을 위한 3자간 협상 프로토콜을 설계를 제안한다. 제안한 자가 조직 저장 매체의 보안기법은 보관 노드의 데이터 검증을 검증 노드에게 위임함으로써 데이터 검증의 효율성을 증가시키고, EC-DH 알고리즘을 이용하여 생성된 암호키와 저장 매체 내의 인증서로 보안을 강화시켰다. 또한, 자가 조직 저장 매체를 구성할 때, 3자간 인증키를 설정하여 외부적인 플러딩 공격 방지하고, 검증노드의 개수를 제한함으로써 내부적인 플러딩 공격을 방지하였다. 그리고 검증단계에서 발생할 수 있는 재전송 공격은 검증을 요청할 때마다 새롭게 생성된 Seed 값을 이용하여 자동적으로 재전송 공격을 탐지하도록 하였다.

A 3-Party Negotiation Protocol Design for the Security of Self-Organized Storage on Infra-Clouding Environment

Byung Kwan Lee†, Eun Hee Jeong††

ABSTRACT

This paper proposes the design of 3-party negotiation protocol for the security of self_organized storage which consists of the owner node possessing data, the holder node holding the owner's data and the verification node verifying the data of the holder node on infra-cloud environment. The proposed security technique delegating the data verification of the holder node to the verification node increases the efficiency of the self-organized storage. In addition, the encrypt key and certification of the storage created by EC-DH algorithm enhances the security much more. Also, when the self-organized storage is composed, the security technique not only prevents external flooding attack by setting a certification key among three parties, but also prevents internal flooding attack by restricting the number of verification nodes. And The replay attack which can occur in the step of verification is automatically detected by using the created seed value whenever the verification is requested.

Key words: Self-Organizing(자가 조직), Security(보안), Negotiation Protocol(협상 프로토콜)

1. 서 론

Gartner는 클라우드 컴퓨팅(Cloud Computing)을 중요한 IT 전략기술 1위로 선정하였다[1]. 그리고 클라우드 컴퓨팅 기술이 기업을 IT 인프라에 대한 유지보수 부담을 경감시킬 수 있다는 기대로 관심이 증대되고 있다. 또한, 스마트 폰 등 모바일 단말기의 발달로 활용 기능의 증가와 광대역 네트워크 발달로

※ 교신저자(Corresponding Author) : 정은희, 주소 : 강원도 삼척시 중앙로 1 (245-711), 전화 : 033)570-6646, FAX : 033)574-6640, E-mail : jeongeh@kangwon.ac.kr
접수일 : 2011년 6월 2일, 수정일 : 2011년 8월 4일
완료일 : 2011년 8월 29일
† 정회원, 관동대학교 컴퓨터학과 교수
(E-mail : bklee@kwandong.ac.kr)
†† 정회원, 강원대학교 지역경제학과 부교수

클라우드 컴퓨팅은 생활의 중심으로 부각되고 있다. 하지만, 클라우드 컴퓨팅은 IT 자원을 소유하는 것이 아니라 일부 또는 모두를 아웃소싱 하는 형태이므로 필연적으로 보안 문제가 제기 될 수밖에 없다. IDC에서 조사한 것에 의하면 244명의 IT 관련 임원들에게 IT 클라우드 서비스에서 해결해야 할 첫 번째 과제로 보안을 꼽고 있다[2].

특히, 인프라 클라우딩의 저장매체 서비스는 웹 하드처럼 클라이언트가 클라우드 컴퓨팅 서버의 저장장소를 임대해 사용하는 방식으로 데이터와 서비스의 중앙 집중화는 악의적인 공격에 치명적인 결합을 초래할 수 있으며, 서버에 문제가 발생하면 클라이언트는 데이터의 접속이 불가능하거나 데이터가 손실될 수 있다. 그리고 클라이언트의 데이터를 임의로 양도할 수 있으므로 인프라 클라우딩의 저장매체 서비스에 대한 좀 더 강력한 보안이 필요하다.

본 논문에서는 이러한 보안 문제점들을 해결할 수 있는 인프라 클라우딩(Infra-Clouding) 환경에서 자가 조직 저장매체의 보안을 위한 3자간 협상 프로토콜을 설계하고자 한다. 본 논문에서 설계한 3자간 협상 프로토콜은 데이터 소유자(Owner) 노드, 데이터를 저장하는 저장매체 서버인 보관(Holder) 노드와 보관 노드의 데이터를 검증하는 검증(Verification) 노드로 자가 조직을 구성하여 데이터를 분산시켜 저장하고, 저장된 데이터를 보관 노드가 아닌 별도의 검증 노드가 검증하도록 함으로써, 인프라 클라우딩 저장매체의 중앙 집중화 문제, 저장 서버 문제로 인한 데이터 손실 문제, 클라이언트 데이터의 임의 양도 문제 등을 해결하고자 한다. 또한, 자가 조직된 구성 요소 간에 필요한 보안 프리미티브를 설계하여 검증 노드가 데이터의 무결성을 검증하도록 함으로써 자가 조직 저장매체 보안을 강화시키고, 데이터 신뢰성과 가용성을 향상시키고자 한다.

본 논문의 구성은 다음과 같다. 2장에서 관련 연구를 살펴보고, 3장에서 자가 조직 저장매체 보안을 위한 3자 협상 프로토콜을 설계한다. 그리고 4장에선 3장에서 설계한 3자 협상 프로토콜을 평가하고, 5장에서 결론을 맺는다.

2. 관련연구

2.1 자가 조직 보안

자가 조직된 시스템(Self-organized system)은 구조화 측면에서 요소들이 특정한 방법으로 조직화되고 상호작용 하면서 기능 측면에서 전체 시스템이 특정한 목적을 만족시켜야 하는 시스템으로 중앙 집중적인 제어 없이 요소들 사이의 peer-to-peer의 분산화된 상호작용을 함을 의미한다[3,4].

P2P와 같이 자가 조직 매체는 분산컴퓨팅 기술을 기반으로 하고 있으므로 분산 환경에서의 IP Spoofing, 노드 간에 전송되는 데이터 스트림의 불법 수정이나 거짓 데이터 스트림생성을 통한 신분위장(masquerade), 재전송(replay), 그리고 서비스 부인(denial of service) 등의 공격과 같은 보안 취약성을 가진다[5,6]. 이러한 보안 취약성을 해결하기 위한 클라우드 컴퓨팅 환경에서의 보안 제어는 대부분 일반적인 IT 환경에서의 보안 제어와 유사하지만, 클라우드 서비스가 동작하도록 하는 운용방법과 기술들이 이용됨으로써, 전통적인 IT에서 만족했던 해결만으로 클라우드 컴퓨팅의 보안을 해결하기 어려운 여지가 있다[7].

클라우드 컴퓨팅 환경의 자가 조직이나 인프라 클라우딩 스토리지 서비스에서 사용하는 모든 보안 기술들을 정리해보면 다음과 같다[8,9].

- 사용자 인증 : 사용자 인증 기술의 가장 대표적인 방법으로 ID/Password, PKI 이 있으며, 보안강도를 높이기 위해 몇 가지를 혼합해 사용하는 Multi-factor(홍채, 지문, OTP), 한국에서 사용하는 i-PIN 등이 있다. 그리고 네트워크에서 사용하는 사용자 인증기법으로는 통합인증서버, ID 연계기반, url 기반 등이 있다.
- 접근제어 : 운영체제상의 한 프로세스가 다른 프로세스의 영역(파일 혹은 메모리)에 접근하는 것을 통제하는 기술로 DAC, MAC, RBAC 등이 있다.
- 검색 가능 암호시스템 : 스토리지에 대한 대표적인 보안 기술로서 기존의 암호 기술과 같이 암호화된 정보에 대한 기밀성을 보장하면서 동시에 특정 키워드를 포함하는 정보를 검색할 수 있도록 고안된 암호 기술이다. 이 보안 기술은 암호화된 데이터 외에 검색에 사용할 인덱스(index)를 추가

로 생성하여 저장함으로써 사용자가 특정 키워드를 포함하는 자료를 검색할 때 이 인덱스를 사용한다. 즉, 사용자가 키워드와 비밀키를 사용하여 키워드의 정보를 포함한 트랩도어(trapdoor)를 생성해 서버에 전달하면, 서버는 사용자가 전해준 트랩도어와 저장된 인덱스를 이용하여 검색을 수행하여 검색의 결과를 사용자에게 전달한다. 이 과정에서 인덱스와 트랩도어로부터 저장된 자료 또는 사용자가 검색한 키워드에 대한 정보의 유출을 최소화함으로써 데이터를 보호하는 것이다[9].

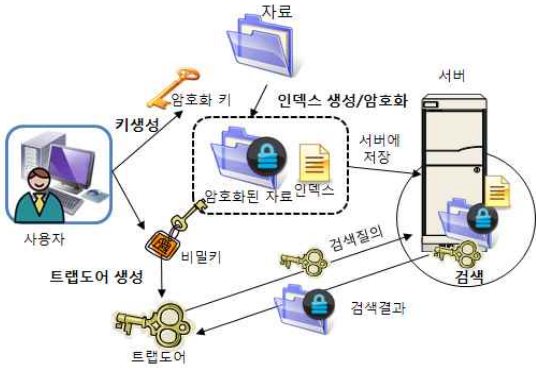


그림 1. 검색 가능 암호 시스템

- 프라이버시 보존형 데이터 마이닝(PPDM) 기술 : 스토리지에 대한 대표적인 보안 기술로 데이터 소유자의 프라이버시를 침해하지 않으면서 유용한 정보를 추출할 때, 정보를 공유하느냐와 프라이버시를 유지하느냐의 trade-off 문제를 해결하고자 개발된 프라이버시 보존형 데이터 마이닝 기술이다. 하지만 PPDM 보다 현재 상용되고 있는 데이터 마이닝 소프트웨어에 제공되는 알고리즘으로는 연관 규칙(association rules), 분류(classification),순차 패턴(sequential patterns), 군집화(clustering) 등이 있다.
- 네트워크 보안 기술 : 인터넷의 발전과 함께 성장한 대표적인 네트워크 보안 기술로 통신상의 기밀성을 보장하는 SSL과 IPsec 기술, 그리고 네트워크를 통한 공격을 차단하는 application firewall과 DDoS 방지 기술 등 이 있다.

2.2 암호 알고리즘

데이터를 안전하게 저장하거나 전달하기 위하여 적용 가능한 대표적인 알고리즘은 비밀 키 알고리즘으로 DES(Data Encryption Standard)와 AES(Advanced Encryption Standard)가, 공개키 암호 알고리즘으로 RSA(Rivest-Shamir-Adleman) 방식과 ECC(Eliptic Curve Cryptography)가 있다.

ECC는 유한체(finite field) 상의 타원곡선 점들 간의 연산에서 정의되는 이산대수 문제의 어려움을 이용하는 것으로 전자서명과 키 교환 알고리즘에 주로 이용된다[10,11]. 그리고 EC-DH는 Diffie-Hellman 알고리즘을 타원곡선위로 옮긴 것으로 X9.63으로 표준화 되어 있다. X9.63에서 사용하는 도메인 파라미터는 ECDSA에서의 도메인 파라미터와 같고, X9.62에서 선택 파라미터로 사용하는 cofactor $h=|E|/n$는 small subgroup 공격을 막기 위해 필수적으로 사용된다. 그리고 각 사용자는 파라미터 n, G,$E(F_p)$, h를 모두 알고 있다고 가정한다[12]. 그림 2는 EC-DH를 이용한 키 교환 과정을 설명한 것이다.

본 논문에서는 EC-DH 키 교환 알고리즘을 이용해 생성된 공유 비밀키를 데이터 소유자 노드, 보관 노드, 검증 노드 간에 자가 조직 저장매체의 인증서로 사용한다.

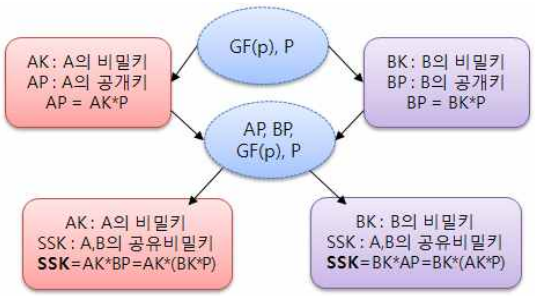


그림 2. EC-DH 키 교환 흐름도

3. 3자간 협상 프로토콜 설계

본 논문에서 제안하는 인프라 클라우딩 환경에서 자가 조직 저장매체의 보안을 위한 3자간 협상 프로토콜은 데이터를 인프라 클라우딩 환경의 저장매체에 저장할 때, 기존의 클라우드 스토리지 서비스처럼 한 곳에 집중하여 저장하는 것이 아니라, 데이터 소유자(Owner) 노드가 데이터를 저장하는 저장매체인 보관(Holder) 노드들과 데이터의 인증을 위임받는 검증(Verification) 노드들을 선정하여 자가 조직

을 구성하여 소유자 노드, 보관 노드, 그리고 검증 노드 간에 자료를 저장하고 검증하도록 3자간 협상하는 프로토콜을 설계한다.

그림 3은 본 논문에서 설계한 3자간 협상 프로토콜의 전체적인 구성과 각 구성요소간의 데이터 저장 및 검증에 대한 흐름을 설명한 것으로, 3자간 협상 프로토콜은 Init_Setup 프리미티브, Personal_Save 프리미티브, MetaGen_Delegation 프리미티브, Verification 프리미티브로 구성된다.

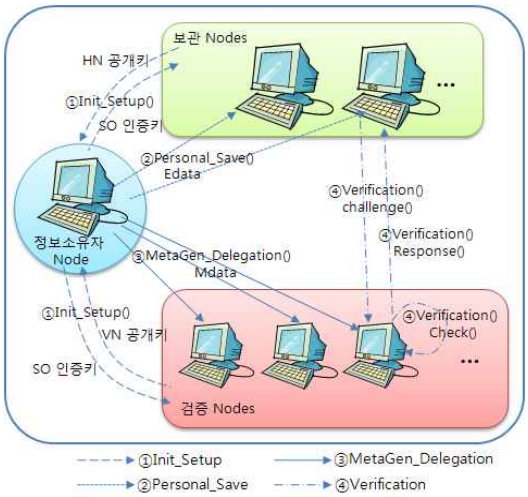


그림 3. 자가 조직 저장 매체 구성 및 협상 프로토콜의 흐름도

3.1 Init_Setup 프리미티브 설계

자가 조직 저장매체에 속해 있는 모든 노드들은 데이터 소유자 노드, 보관 노드, 검증 노드가 될 수 있으며, 자가 조직 저장매체를 구성할 때, 보관 노드와 검증 노드의 개수를 미리 정의하여 설계한다.

```
Init_Setup(S, V){
  S : 보관노드의 개수, V : 검증노드의 개수
  self_organizing(S, V); // 자가 조직 구성
  request_id(S, V);
        // 보관노드들과 검증노드들의 ID 요청
  set_elliptic_curve(E, a, b, n, P);
        // 타원곡선 선정
  OPkey=compute_public_key(E,a,b,n,P, OSkey);
        // 공개키 생성
  CSign = ECC(OSkey, HPkey, VPkey);
        // 3자간의 인증키 생성
}
```

그림 4. Init_Setup 알고리즘

그림 4는 Init_Setup 알고리즘을 설명한 것으로, 보관 노드와 검증 노드의 수를 정해 자가 조직 저장 매체를 구성한다. 그리고 보관 노드와 검증 노드의 아이디를 요청하고, 인증서 생성에 필요한 타원곡선을 선정하고, 저장매체 구성 노드에 타원곡선의 a, b, P, n을 공개한다. 데이터 소유자 노드의 비밀키는 비밀로 유지하고, 보관 노드와 검증 노드 또한 자신의 비밀키를 이용해 공개키를 생성해 자가 조직 저장 매체 내에 공개한다. 이 공개키를 이용해 자가 조직 저장 매체내의 서로의 신분을 확인하는 인증서를 생성하여 데이터를 암호화, 검증 그리고 서로의 신분을 확인할 때 사용한다.

그림 5는 자가 조직 저장 매체의 노드간의 인증서인 Csign을 생성하는 과정을 설명한 것이다. 데이터 소유자는 타원곡선 암호 알고리즘을 이용해 암호화에 필요한 파라미터 생성하고, 저장 매체 내의 구성 노드에 공개하면, 각 노드들은 자신의 비밀키를 이용해 공개키를 생성한다. 각 노드는 EC-DH 알고리즘을 이용해 상대방의 공개키를 자신의 비밀키로 한번 더 연산을 하면, 3자간 공유 비밀키가 생성된다. 이 공유 비밀키를 자가 조직 저장 매체의 인증서인 Csign으로 사용한다.

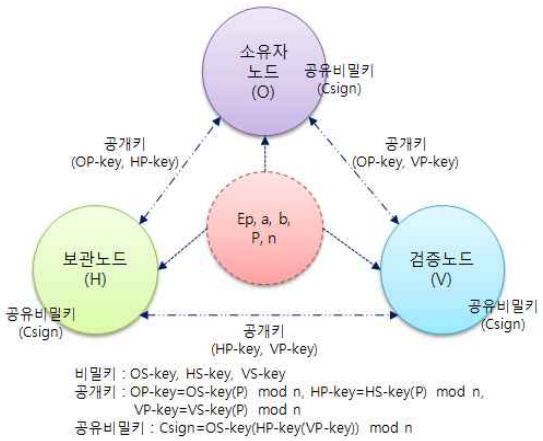


그림 5. 저장 매체 내의 인증서 생성 흐름도

3.2 Personal_Save 프리미티브 설계

Personal_Save 프리미티브는 Init_Setup 프리미티브에서 설정된 자가 조직 저장매체의 인증서를 이용해 데이터의 소유자 노드가 데이터를 저장할 보관 노드와 검증 노드의 신분을 확인하고, 데이터를 보호

하기 위해 보관 노드만이 풀 수 있도록 암호화를 시켜 전달한다. 이때, 데이터 소유자 노드는 데이터가 전송한 보관 노드가 누구인지에 대한 정보를 기억해야 한다. 보관 노드는 데이터를 변조 없이 안전하게 보관하기만 하면 되므로, 소유자 노드 데이터 암호화 키로 암호화된 데이터를 그대로 보관한다.

그림 6는 Personal_Save 프리미티브의 처리 절차를 설명한 것이다.

1단계 : 데이터 소유자 노드는 자신의 비밀키로 보관 노드에 저장할 데이터를 암호화한다.

```
Owner_Chiper(data, OS-key){
  OS-key : Owner Secret Key;
  Edata = OS-Key⊕ data;
  return(Edata);
}
```

2단계 : 암호화된 데이터를 일정한 크기로 나눈다. 이때, 맨 마지막 데이터 조각이 일정한 크기가 되지 않을 때에는 데이터의 뒷부분에 0을 채워 일정한 크기가 되도록 조정한다.

```
Slice_Data(Edata, m, Sdata){
  m : Data Slice size;
  Edata : encrypt data;
  cnt : Data Slice count;
  cnt = SliceData(Edata, Sdata, m);
   // 데이터를 m크기로 나눔.
  for(i=0 ; i<m ; i++){
   if (Sdata[cnt, i] == NULL)
     Sdata[cnt, i] = 0 ;
       // 데이터의 뒷부분에 0을 채움.
 return(Sdata);
}
```

3단계 : 데이터를 저장할 보관 노드의 고유 ID와 공개키를 가져와 암호화시켜 암호화된 데이터 조각을 각각의 보관 노드에 전달한다. 그리고 소유자 노드는 데이터를 저장한 보관 노드 정보를 기억한다.

```
Personal_Holder(Sdata, HID, HP-key, m, cnt){
  HID : Holder's ID;
  HP-key : Holder's public key;
  Pdata : Personal data;
  for(i=0 ; i<cnt ; i++){
    for(j=0 ; j< m ; j++){
      Pdata[i,j] = HP-key(Sdata[i,j])||HID;
   // 데이터 조각을 Holder의 공개키로 암호화
  }
  return(Pdata);
}
```

3.3 MetaGen_Delegation 프리미티브 설계

데이터 소유자 노드는 보관 노드에게 암호화하여 전송한 데이터로 meta-정보를 생성하여 데이터를 검증할 검증 노드에 전송한다. 물론, 데이터 소유자 노드는 어떤 검증 노드에 meta-정보를 전송했는지에 대해 기억해야 한다.

그림 7은 MetaGen_Delegation 프리미티브의 처리과정을 설명한 것이다.

1단계 : 데이터의 소유자 노드가 자신의 암호키로 암호화한 데이터로 검증 노드에 전송할 meta 정보를 생성한다.

```
MetaGen(Sdata){
  mdata = hash(Sdata);
  return(mdata);
}
```

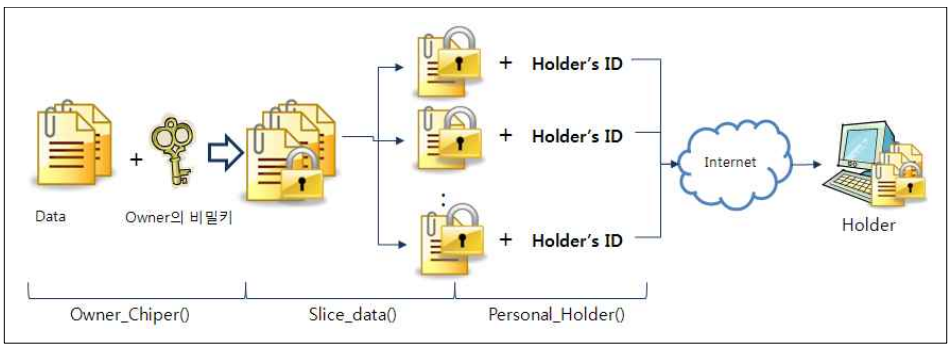


그림 6. Personal_Save 프리미티브 처리 과정

2단계 : meta 정보에 소지자 노드의 ID를 연접해 검증 노드의 공개키로 암호화시켜 검증 노드에 전달한다.

```
Delegation(mdata, VP-key){
  VP-key : verifier's public key;
  metaInfo = VP-key(mdata || HID);
  return(metaInfo);
}
```

3.4 Verification 프리미티브 설계

Verification 프리미티브는 자가 조직 저장매체에서만 사용되는 인증서와 Seed 값을 이용해 보관 노드와 검증 노드 간에 데이터 검증을 요청하는 요청(Challenge) 메시지와 응답(Response) 메시지를 생성하도록 설계하였다. 이때 Seed 값은 검증을 요청할 때마다 새로운 값으로 생성되도록 함으로써, 플러딩 공격과 재전송 공격을 방지하고자 하였다.

그림 8은 Verification 프리미티브 처리과정을 설명한 것이다.

1단계 : 검증 노드는 소유자 노드로부터 전송받은 meta 정보에서 보관 노드의 ID를 찾아 요청 신호 Q와 저장매체 내의 인증서인 Csign, 그리고 Seed 값을 보관 노드에 전송한다.

```
Challenge(HP-key, S, Csign){
  Csign : self-organized storage certification
  S : Seed value;
  HP-Key : Holder's public key;
  S = Random_Seed(); //seed값을 랜덤하게 생성
  Q = HP-key(S)||Csign;
    // 보관 노드의 공개키로 암호화
  return(Q);
}
```

2단계 : 보관 노드는 검증 노드의 인증서 Csign은 확인하고 맞으면 자신의 비밀키로 검증 요청 신호 Q를 복호화시켜 검증 노드가 전송한 Seed 값을 찾는다. 그리고 보관하고 있던 데이터를 해쉬한 값과 Seed 값를 검증 노드의 공개키로 암호화 시킨 응답 메시지인 R을 검증 노드에 전송한다.

```
Response(Q, HS-key, Csign){
  Q : Challenge message;
  HS-key : Holder's Secret Key;
  if (compare(Csign)){
    S = HS-key(Q);  // 전송 받는 Seed값 복원
    Hdata = hash(Pdata);
    R = VP-key(S, Hdata);
    return(R);
  }else{ return("Csign error");}
  }
}
```

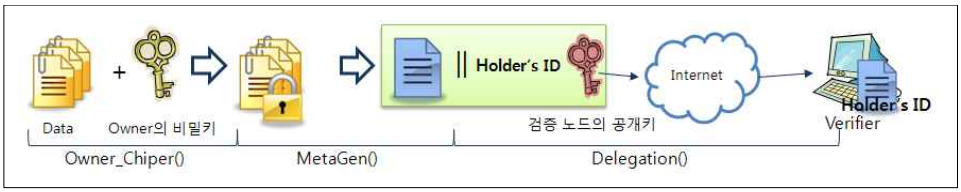


그림 7. MetaGen-Delegation 프리미티브 처리 과정

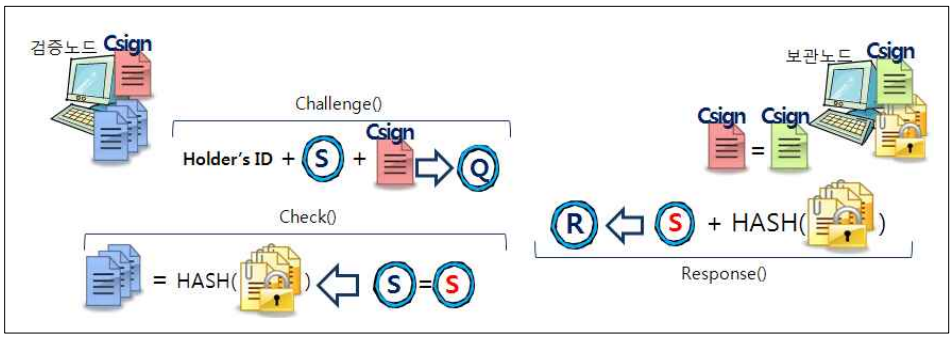


그림 8. Verification 프리미티브 처리과정

3단계 : 검증 노드는 보관 노드로 전송받은 인증서를 확인 후, 인증서가 맞으면 응답 메시지를 검증 노드의 비밀키로 복호화시켜 검증 노드의 Seed 값과 보관 노드가 전송한 Seed 값이 일치하는지 검사한다. Seed 값이 일치하면, 데이터 삭제 및 변조 여부를 검증한다.

```
Check(R, Csign){
  VS-key : Verifier's Secrete Key;
  SV : received seed value;
  if(compare(Csign)){
    Vdata(SV, hdata) = VS-key(R);
    if (S != SV) return("Seed value error");
    if (compare(mdata, hdata))
        msn = "TRUE";
    else
        msn = "FALSE";
    return(msn);
 else{
   return("Csign error");
 }
}
```

4. 보안기법 평가

본 논문에서 제안하는 자가 조직 저장 매체의 보안을 위한 3자간 협상 프로토콜은 설정단계에서 자가 조직 저장 매체를 구성할 때, 보관 노드와 검증 노드를 일정한 수 이내로 조직하도록 미리 설정하였고, 저장 매체 내에서만 사용가능한 인증서를 생성하였다.

4.1 상호 인증

본 논문에서 제안하는 3자간 협상 프로토콜 설계에서 3자간 협상을 하기 전에 3자간의 신분을 확인해야 하는데 EC-DH 알고리즘을 이용하였다. 즉, EC-DH 알고리즘을 이용해 자가 조직 저장 매체의 공유 비밀키를 생성하고, 공유 비밀키를 인증서인 Csign으로 사용해 서로의 신분을 확인하도록 하여 악의의 사용자가 보관노드에 데이터를 임의로 저장하는 것을 방지하였고, 플로딩 공격을 방지하는데 이용하였다.

그리고 본 논문에서 제안한 사용자 인증 모듈은 3회의 사용자 인증 연산 횟수를 가지지만 EC-DH 알고리즘을 이용해 노드의 공개키를 이용해 노드 인증서를 별도로 생성해 인증서만 전송하므로 최대 160bit의 데이터를 송신하므로 대표적인 공개키 암호화 알고리즘인 RSA 보다 데이터 전송량이 적으며, 보안적인 면에서는 더 강화되었다. 또한, 사용자 인증에 상대방의 공개키를 가져와서 인증서를 생성하므로 사용자 인증단계에는 별도의 암·복호화과정이 필요 없다.

4.2 플로딩 공격 방지

본 논문에서 설계한 3자간 협상 프로토콜은 검증 노드가 보관 노드에 저장된 데이터의 삭제 및 무결성을 검증하기 위해 보관 노드에 요청 메시지를 전송하는데, 자가 조직 저장 매체 내의 구성 노드가 아닌 악의적인 노드가 데이터 검증을 요청하는 요청 메시지를 보관 노드에 반복적으로 전송하는 플로딩 공격이 발생할 수 있다.

본 논문에서 설계한 3자간 프로토콜에선 데이터 검증 단계를 시작하기 전에 먼저 검증 노드와 보관 노드는 자가 조직 저장 매체내의 구성 노드임을 서로간에 인증을 하고, 보관 노드가 저장 매체의 인증서인 Csign을 가진 검증 노드만의 요청 메시지를 처리하도록 함으로써 외부적인 플로딩 공격을 방지한다. 또한, 자가 조직 저장 매체를 구성할 때, 보관 노드와 검증 노드의 수를 제한함으로써 저장 매체 내부 플러딩 공격을 방지하도록 하였다.

4.3 재전송 공격 방지

검증 노드는 보관 노드의 응답 메시지를 수신해 보관 노드의 데이터의 변조 여부를 검증 하는데, 보관 노드가 응답 메시지를 반복적으로 전송하는 악의적인 재전송 공격이 발생할 수 있다.

그림 8에서 설명하고 있듯이, 본 논문에서 제안하는 자가 조직 저장매체의 3자간 협상 프로토콜 설계에선 검증 노드는 보관 노드에 요청 메시지를 전송할 때, 랜덤하게 생성된 Seed 값을 포함해 전송하고, 보관 노드는 전달받은 Seed 값을 재전송하도록 설계하였다. 따라서 검증노드는 재전송된 Seed 값을 비교함으로써 악의적인 재전송 공격을 방지할 수 있다. 또한, Seed값은 요청 메시지를 작성할 때마다 다르게 생성함으로써 재전송공격을 검출할 수 있다.

5. 결 론

본 논문에선 인프라 클라우딩 환경에서 자가 조직 저장매체의 보안을 위한 3자간 협상 프로토콜을 설계하였다. 자가 조직 저장매체를 위한 보안 기법은 데이터 소유자가 데이터를 소유자 노드의 암호키로 암호화하여 보관 노드에 저장하고, 데이터의 검증은 검증 노드에게 위임하여 검증하도록 하여, 보관 노드가 소유자 노드의 데이터를 여전히 소유하고 있는지를 검증하였으며, 보관노드가 임의로 소유자 노드의 데이터를 위임하더라도 소유자의 암호화키로 암호화되어 있으므로 데이터의 노출 및 임의 양도를 방지하였다. 그리고 3자간 협상을 하기 전에 서로의 신분을 확인하는데 EC-DH 알고리즘을 이용해, 자가 조직 저장 매체의 공유 비밀키를 생성하고, 공유 비밀키를 인증서인 Csign으로 사용해 서로의 신분을 확인하도록 하여 저장 매체 내의 보안을 강화시켰다.

또한, 자가 조직 저장매체를 구성할 때 그룹간의 인증서를 설정하여 외부적인 플로딩 공격을 방지하였으며, 검증노드의 개수를 제한함으로써 내부적인 플로딩 공격을 방지하였다. 그리고 검증단계에서 발생할 수 있는 재전송 공격은 검증을 요청할 때마다 새롭게 생성된 Seed 값을 이용하여 자동적으로 재전송 공격을 탐지하도록 하여, 자가 조직 저장매체의 보안을 한층 강화시켰다.

참 고 문 헌

[1] 이정아, “모바일 클라우드 서비스 국내외 정책 추진현황,” KT 경제경영연구소, pp. 1-15, http://digieco.co.kr

[2] 민옥기, 김학영, 남궁한 “클라우드 컴퓨팅 기술 동향,” 전자통신동향분석, 제24권, 제4호, pp. 1-13, 2009.

[3] C. Prehofer and C. Bettstetter, “Self-organization in communication networks: Principles and design paradigms,” *IEEE Communications Magazine*, Vol.43, No.7, pp. 78-85, 2005.

[4] http://en.wikipedia.org/

[5] 권혁찬, 문용혁, 구자법, 고선기, 나재훈, 장종수, “P2P 표준화 및 기술동향,” 전자통신동향분석, 제22권 제1호, pp. 11-23, 2007.

[6] 문용혁, 권혁찬, 나재훈, 장종수, “P2P 사용자 인증과 OTP 분석,” 정보보호논문지, 제17권 제3호, pp. 32-40, 2007.

[7] 김진형, 김윤정, 박춘식, “클라우드 컴퓨팅에서 신뢰하지 않는 서버 데이터의 안전한 접근,” 정보과학회지, 제28권, 제12호, pp. 67-74, 2010.

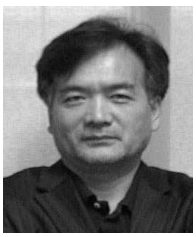
[8] 임철수, “클라우드 컴퓨팅 보안 기술,” 정보보호학회지, 제19권, 제3호, pp. 14-17, 2009.

[9] 은성경, 조남수, 김영호, 최대선, “클라우드 컴퓨팅 보안 기술,” 전자통신동향분석, 제24권, 제4호, pp. 79-88, 2009.

[10] N. Koblitz, “Elliptic Curve Cryptosystems,” *Mathematics of Computation*, Vol.48, No.177, pp. 203-209, 1987.

[11] V. Miller, "Uses of Elliptic Curves in Cryptography,"advances in Cryptology Proceedings of Crypto'85, LNCS(218), pp.417-426, 1986

[12] 이완복, 노창현, 류대현, “공개키 연산기의 효율적인 통합 설계를 위한 임계경로분석,” 멀티미디어학회논문지, 제8권 제1호, pp.79-87, 2005



이 병 관

1979년 2월 부산대학교 기계설계학과 공학사
1986년 2월 중앙대학교 전자계산공학과 공학석사
1990년 2월 중앙대학교 전자계산학과 공학박사

1988년 3월～현재 관동대학교 컴퓨터학과 교수
관심분야 : 네트워크 보안, 컴퓨터 네트워크, 전자상거래

정 은 희

1991년 2월 강릉대학교 통계학과 이학사
1998년 2월 관동대학교 전자계산공학과 공학석사
2003년 2월 관동대학교 전자계산공학과 공학박사

2003년 9월～현재 강원대학교 지역경제학과 부교수
관심분야 : 네트워크 보안, 웹 프로그래밍, 전자상거래