분산 TTP를 이용한 공정한 부인봉쇄 프로토콜*

최종권○ 이헌길
강원대학교 컴퓨터· 정보통신공학과
bbell@mirae.kangwon.ac.kr hglee@kangwon.ac.kr

A Fair Non-Repudiation Protocol Using Distributed TTP

Jong-Kwon Choi○ Heon-Guil Lee
Dept. of Computer Information and Communications Engineering,
Kangwon National University

요 약

통신 기술의 발달은 컴퓨터를 활용한 정보 교환을 수월하게 만들었지만, 더불어 정보 유출의 가능성 또한 증가하였다. 특히 전자상거래와 같이 정보의 보호가 필요한 응용들도 급격히 활성화됨에 따라, 정보의 보안은 반드시 필요하며, 보안 서비스 중에서도 공정한 부인 봉쇄 서비스는 필수적이다. 기존에 제안된 부인 봉쇄 기법들은 보통 공정성을 위해 송· 수신자는 중개자인 Trusted Third Party(TTP)를 두어 서로 통신한다. 하지만, 클라이언트수가 증가하면 중앙의 TTP에 네트웍 트래픽이 집중되어 효율적이지 못하다. 본 논문에서는 중앙 TTP에 집중되는 네트웍 트래픽량을 줄이기 위해 TTP를 분산시키는 새로운 부인봉쇄 프로토콜을 제시한다.

1. 서론

부인 봉쇄 서비스는 거래에 참여했던 한 개체가 특별한 상황이나 행동에 대해 부인할 때 거래에 참여한 또 다른 개체를 보호하는 서비스이다[1, 5].

기본적인 부인 봉쇄 서비스에서는 거래에 참여하는 두 개체에게 송신증명서(EOO: Evidence of Origin)와 수신증명서(EOR: Evidence of Receipt)를 요구하고 이 증명서를 통해 분쟁을 해결한다.

부인 봉쇄 서비스는 공정성(fairness)을 요구한다. 공정성이란 통신 중 어떠한 상황이 발생하였을 때, 통신에 참여하는 개체 중 어느 누구도 유리한 위치나 불리한 위치에 있지 않음을 말한다[2, 6]. 부인 봉쇄 서비스에서의 공정성을 위해 일반적으로 Trusted Third Party(TTP)를 이용하는 방식을 사용한다. 이는 통신에 참여하는 개체 사이에 신뢰할 수 있는 제 3의 서버를 두는 방법으로 서버를 유지하는 비용은 들지만 공정성을 유지할 수 있다는 장점이 있다[1].

Zhou와 Gollmann(ZG)는 TTP를 통해 송·수신자 사이에 정보를 중재하는 공정한 부인 봉쇄 프로토콜 모델을 제안하였다[3]. 그러나 이 방법에서는 부인 봉쇄 서비스를 요구하는 클라이언트수가 증가하면 중앙의 TTP에서 처리해야 하는 네트웍 트래픽량이 증가하여 중앙의 TTP에서의 처리량이 많아진다는 단점이 있다. 본 논문에서는 이를 해결하고자 TTP를 분산시켜 네트웍 트래픽을 분산시키는 새로운 부인봉쇄 프로토콜을 제시한다.

* 이 연구는 정보통신부 정보통신분야 우수대학원 지원 사업과제로 수행된 것임.

본 논문의 2장에서는 ZG의 공정한 부인 봉쇄 프로토콜을 소개하고 3장에서는 TTP를 분산시킨 새로운 프로토콜을 소개한다. 그리고 4장에서는 분쟁 해결 과정과 성능평가를 하고 5장에서 결론을 맺는다.

2. 관련연구

본 논문에서는 부인 봉쇄 프로토콜을 설명하기 위해 표 1과 같은 기호를 사용한다.

표 1. 부인봉쇄 프로토콜을 설명하기 위한 기호

· A : 송신자
· B : 수신자
· TTP : Trusted Third Party
· M : A가 B로 보내려는 원본 메시지
· K : 암호화 키
· S_A,S_B,S_{TTP} : A, B, TTP의 서명키, 즉 비밀키
· V_A,V_B,V_{TTP} : A, B, TTP의 검증키, 즉 공개키
· X, Y : 메시지 X와 Y를 연결
· H(X) : 메시지 X를 해싱함
· sK(X) : 키 K를 사용한 메시지 X의 서명
· eK(X) : 키 K를 사용하여 메시지 X를 암호화
· C : 메시지 M의 암호문(즉, C = eK(M))
· A ->B:X : A가 B에게 메시지 X를 전송
· A<->B:X : A가 B로부터 "ftp get"을 사용하여 메시지를 가져온다.
· L : 동일 세션의 메시지들을 구별하는 레이블, L = H(dK(C))
· EOO = $sS_A(f_{EOO},B,L,C)$: 암호문 송신증명서
· EOR = $sS_B(f_{EOR},A,L,C)$: 암호문 수신증명서

- sub_K = $sS_A(f_{SUB},B,L,K)$: 키 제출 증명서
- con_K = $sS_{TTP}(f_{CON},A,B,L,K)$: 키 확인 증명서
- f_{EOO} : EOO를 가리키는 플래그 정보
- f_{EOR} : EOR를 가리키는 플래그 정보
- f_{SUB} : 키를 제출했다는 플래그 정보
- f_{CON} : 키를 확인했다는 플래그 정보

ZG의 공정한 부인봉쇄 프로토콜에서는 다음 단계별로 수행된다.

1. A→B : f_{EOO}, B, L, C, EOO
2. B→A : f_{EOR}, A, L, EOR
3. A→TTP : f_{SUB}, B, L, K, sub_K
4. B↔TTP : f_{CON}, A, B, L, K, con_K
5. A↔TTP : f_{CON}, A, B, L, K, con_K

1, 2번 단계에서 A와 B사이에 암호화된 메시지와 송신증명서 그리고 수신증명서를 직접 주고받으며 3, 4, 5번 단계에서 송신자가 중앙의 단일 TTP에 키를 제출하면 TTP는 con_K를 생성하고 A, B, L, K, con_K를 저장한다. A와 B는 "ftp get"를 사용해 con_K를 가져온다. 이 프로토콜에서는 단일 TTP를 사용하는 클라이언트수가 증가하면 TTP에서 저장할 저장공간이 증가하게 되며 TTP에서의 통신량의 증가로 TTP에 병목현상이 발생하게 된다.

3. 제안된 송신자 지역도메인 기반의 분산 TTP 프로토콜

본 논문에서는 중앙의 TTP에 네트워 트래픽이 집중되는 문제를 해결하기 위해 TTP들을 각 지역 도메인 내에 하나씩 분산시켰다(그림 1). 개체들은 송신측 지역 도메인의 TTP_i를 기반으로 통신한다(그림 2, 3). 각 개체들과 모든 TTP_i사이에서의 통신은 신뢰성과 안전성이 보장된다고 가정한다. 프로토콜의 수행 단계는 표 2, 3과 같다.

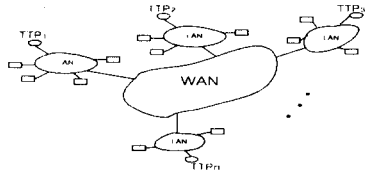

그림 1. 분산 TTP의 구조

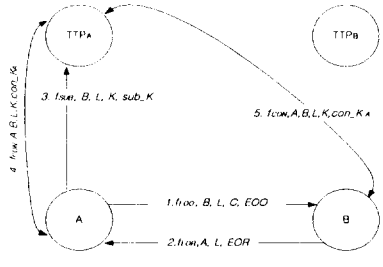

그림 2. A가 B에게 메시지를 보낼 때

A가 B에게 메시지를 보낼 때 1, 2번 단계에서는 A와 B사이에 암호화된 메시지 C와 송신 증명서(EOO) 그리고 수신 증명서(EOR)를 직접 주고받으며, 3번 단계에서 A가 지역 도메인의 TTP_A에 키 K와 키 제출증명서(sub_K)를 제출한다. 그러면 TTP_A는 키 확인 증명서(con_K_A)를 생성하고 A, B, L, K, con_K_A를 읽기만 가능한 공용의 디렉토리에 저장한다. 4번 단계에서 B는 "ftp get"을 사용하여 TTP_A로부터 con_K_A를 가져온다. 5번 단계에서 A는 "ftp get"을 사용하여 TTP_A로부터 con_K_A를 가져온다. B가 A에게 메시지를 보낼 때는 이와 반대로 수행된다(그림 3, 표 3).

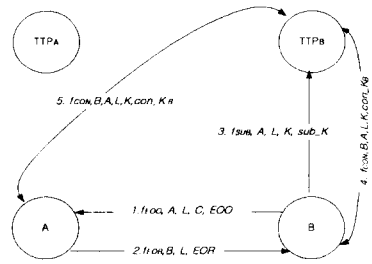

그림 3 . B가 A에게 메시지를 보낼 때

표 2. A가 B에게 메시지를 보낼 때

1. A→B : f_{EOO}, B, L, C, EOO
2. B→A : f_{EOR}, A, L, EOR
3. A→TTP_A: f_{SUB}, B, L, K, sub_K
4. TTP_A↔B: f_{CON}, A, B, L, K, con_K_A
5. TTP_A↔A: f_{CON}, A, B, L, K, con_K_A

표 3. B가 A에게 메시지를 보낼 때

1. B→A : f_{EOO}, A, L, C, EOO
2. A→B : f_{EOR}, B, L, EOR
3. B→TTP_B: f_{SUB}, A, L, K, sub_K
4. TTP_B↔B: f_{CON}, B, A, L, K, con_K_B
5. TTP_B↔A: f_{CON}, B, A, L, K, con_K_B

본 논문에서 제안한 분산 TTP 프로토콜은 기존에 제안된 프로토콜에서의 단점인 중앙 TTP에서 발생되었던 네트워 집중과, 증명서를 저장해야 할 저장공간을 송신측 지역 도메인의 TTP로 분산시킴으로써 성능향상을 이룬다.

4. 분쟁 해결 과정 및 성능평가

4.1 분쟁 해결 과정

분쟁은 송·수신자가 메시지의 송·수신 사실을 부인하는 경우에 일어난다. 송신자가 메시지 M을 수신자에게 보냈는데도 불구하고 수신자가 이를 부인할 때나, 그 반대로 수신자는 메시지를 수신했는데 송신자가 송신 사실을 부인하는 경우이다. 이럴 때 판사(Judge)가 중재를 하여 분쟁을 해결한다.

본 논문에서는 분쟁 해결 과정을 표현하기 위해 SVO Logic[4]에 기반을 둔 방법으로 표기한다. SVO 로직에서 사용하는 기호는 표 4에 나타내었다.

표 4. 분쟁 해결을 위한 SVO 로직 기호

- *PK(A,K)* : K는 A의 서명확인 공개키이다.
- *P received X from Q* : P가 Q로부터 X를 수신한다.
- *P checks X* : P가 X를 검증한다.
- *P believes X* : P가 X를 인정한다.
- *P don't believes X* : P가 X를 인정하지 않는다.
- *P said X* : P가 X를 전송했다.
- *P sees X* : P가 X를 수신했다.
- *P::W* : P의 결과가 W다. (W는 True 혹은 False)

- $P=>W$: P일 때 W가 된다. (P는 True 혹은 False)
- $P \mid Q$: P이거나 Q(OR를 나타냄)
- $P \wedge Q$: P이면서 Q(AND를 나타냄)
- $P \equiv Q$: P와 Q는 같다.
- $P \supset Q$: P라는 것은 Q를 뜻한다.

A가 B에게 메시지를 전달했을 때 분쟁해결을 위한 목표는 다음과 같다.

송신 부인 봉쇄의 목표: *J believes (A said M)*

수신 부인 봉쇄의 목표: *J believes (B see M)*

분쟁해결을 위한 전제는 다음과 같다.

J believes PK(A, V_A)
J believes PK(B, V_B)
J believes PK(TTP_i, V_{TTPi}) {i = A, B }
J received (M,C,K,L,EOR | con_K_A)
J received (M,C,K,L,EOO | con_K_A)
TTP_A said (A,B,L,K)⊃A said(A,B,L,K)∧B see(A,B,L,K)
A said (A, B, L, eK(M)) ∧ A said (A, B, L, K) ⊃ A said M
B sees (A, B, L, eK(M)) ∧ B sees (A, B, L, K) ⊃ B sees M

A가 송신한 사실을 부인할 때 B는 M,C,K,EOO,con_K_A를 판사에게 증거로 제시한다. 판사는 EOO가 A의 서명인지, con_K_A가 TTP_A의 서명인지 검사한다. 그리고 통신이 동일한 세션에서 일어났는지, M = dK(C)인지 검사한다. 아래는 SVO 로직을 사용하여 분쟁해결 과정을 나타낸 것이다.

1. *J received (f_{EOO}, M, C, K, L, EOO | con_K_A) from B*
2. *J checks (A said EOO)*
3. *2::True => J believes (A said EOO)*
4. *J checks (TTP_A said con_K_A)*
5. *4::True=>J believes (TTP_A said con_K_A)*
6. *J checks (L ≡ H(dK(C)))*
7. *6::True=> J believes (L ≡ H(dK(C)))*
8. *J checks (M ≡ dK(C))*
9. *8::True=>J believes (M ≡ dK(C))*
10. *J believes (A said M)*

B가 수신한 사실을 부인할 때 A는 M,C,K,EOR,con_K_A를 판사에게 증거로 제시한다. 판사는 EOR이 B의 서명인지, con_K_A가 TTP_A의 서명인지 검사한다. 그리고 통신이 동일한 세션에서 일어났는지, M = dK(C)인지 검사한다. 아래는 SVO 로직을 사용하여 분쟁해결 과정을 나타낸 것이다.

1. *J received (f_{EOR}, M, C, K, L, EOR | con_K_A) from A*
2. *J checks (B said EOR)*
3. *2::True => J believes (B said EOR)*
4. *J checks (TTP_A said con_K_A)*
5. *4::True=> J believes (TTP_A said con_K_A)*
6. *J checks (L ≡ H(dK(C)))*
7. *6::True => J believes (L ≡ H(dK(C)))*
8. *J checks (M ≡ dK(C))*
9. *8::True=>J believes (M ≡ dK(C))*
10. *J believes (B see M)*

4.2 성능평가

본 논문에서는 TTP를 분산한 경우와 중앙에 하나를 두었을 경우에 따른 평균 수행시간을 시뮬레이션 하였다. LAN과 WAN에서의 지연시간의 비율은 1:3으로 하였으며[7], 각각의 TTP는 동일한 처리량을 갖는다고 가정하였다. 이 가정 하에 클라이언트 수의 증가에 따른 프로토콜의 수행시간을 비교하였다.

그림 4에서 볼 수 있듯이 클라이언트수가 작을 때는 중앙 TTP와 분산 TTP에서의 프로토콜 수행시간의 차이가 적지만, 클라이언트 수가 증가할수록 프로토콜 수행시간의 차이가 증가함을 볼 수 있다. 이는 중앙 TTP에 모이는 네트워 트래픽량을 분산화 시킴으로써 중앙 TTP에서 부담해야 하는 처리량을 덜어 주는 효과를 가져온다.

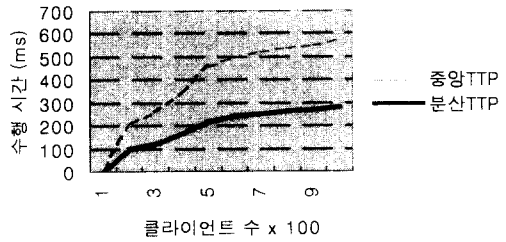

그림 4. 클라이언트수에 따른 프로토콜 수행시간

5. 결론 및 향후 과제

본 논문에서는 공정한 부인 봉쇄 프로토콜에서 클라이언트수가 증가함에 따라 중앙의 TTP에 집중되는 네트워 트래픽량을 줄이고자 TTP를 분산시킨 공정한 부인 봉쇄 프로토콜을 제시하였다. 제안된 분산 TTP를 이용한 부인봉쇄 프로토콜은 송·수신자 모두에게 공정성을 제공하며, 중앙의 TTP를 이용한 부인봉쇄 프로토콜에 비해 수행시간이 감소하였다. 향후 과제로는 통신채널이 안전하다는 가정 없이도 공정하게 부인 봉쇄 서비스가 수행되는 분산 TTP 프로토콜에 대한 연구이다.

5. 참고문헌

[1] ISO/IEC DIS 133888-1. Information technolody - Security techniques - Non Repudiation - Part 1: General model. ISO/IEC JTC1/SC27 N1503,November 1996.

[2] M.Ben, O.Gold, S.Micali and L.Rivest. "A Fair Protocol for Signing Contracts", IEEE Transaction in Information Theory, Vol 36. January 1990

[3] J.Zhou and D.Gollmann. "A fair non-repudiation protocol." In Proceedings of 1996 IEEE Symposium on Security and Pivacy, pages 55-61,Oakland, California, May 1996.

[4] P.Syverson and P.C. van Oorschot. A unified some cryptographic protocol logic. Draft (work in progress), March 1996.

[5] R.kailar. Accountability in Electronic Commerce Protocols. IEEE Transactions on Software Engineering, 22(5):313 - 328, May 1996.

[6] T.kamoto and K. Ohta. How to simultaneously exchange secretes by general assumptions. In Proceedings of 2nd ACM Conference on Computer and Communications Security, pages 184-192. Fairfax, virginia, November 1994.

[7] 김주영, 박준희, 송화선, 정영준 "ATM 망에서 ABR 서비스를 위한 개선된 전송률 기반의 폭주 제어 알고리즘", 한국정보과학회, 98' 가을 학술발표논문집(III) pages. 374-376