기밀성 및 공정성이 보장되는 웹 기반 Pay-Per-View

엄상용[O] 주한규
한림대학교 컴퓨터공학과
{syuhmn, hkjoo}@hallym.ac.kr

Private and Fair Web-Based Pay-Per-View

Saangyong Uhmn[0] Hankyu Joo
Dept. of Computer Engineering, Hallym University

요 약

인터넷이 활성화되면서 다양한 웹 기반 서비스가 제공되어진다. 그 가운데 하나가 VOD(Video On Demand) 서비스이다. VOD는 다양한 프로그램을 사용자가 원하는 시점에 볼 수 있도록 한다. VOD 서비스를 유료화할 경우 시청한 양만큼 요금을 지불하는 pay-per-view 방법을 생각할 수 있다. Pay-per-view 방식으로 서비스를 하는 경우 사용자는 각 프로그램을 시청한 양에 따라 요금이 부과되고 서비스 제공자는 부과된 요금이 정당함을 보증하는 공정성을 제공하여야 하다. 또한 사용자의 시청 내역을 타인으로부터 보호하는 기밀성이 보장되어야 한다.

이 연구에서는 기밀성과 공정성을 제공하는 웹 기반 pay-per-view 방식을 개발하였다. 이 논문에서 제안된 pay-per-view 방식은 대칭키 암호 기법을 이용하여 내용 및 자료 요청에 대한 기밀성을 제공하며 전자 서명과 다중 해쉬를 사용하여 사용자의 요금 부과에 대한 공정성을 제공한다. 또한 제안된 방식은 반복적으로 시청되는 내용에 요금을 차등하게 부과할 수 있는 유연성을 가지고 있다.

1. 서 론

인터넷이 활성화되면서 다양한 웹 기반 서비스가 제공되어진다. 그 가운데 하나가 VOD(Video-on-Demand) 서비스이다. VOD는 다양한 프로그램을 사용자가 원하는 시점에 볼 수 있도록 한다. VOD 서비스를 유료화할 경우 다양한 지원 방법이 필요하다. 이를 지원하는 방법은 월회비를 지불하고 자유롭게 이용할 수 있는 정액 방법과 시청한 양만큼 요금을 지불하는 pay-per-view 방식으로 나눌 수 있다. 정액 사용 방법의 경우 월 회비를 납부하는 회원만이 프로그램을 이용할 수 있도록 하는 기법과 회원의 사생활을 보호하는 기법이 필요하다. Pay-per-view 방식의 경우에는 허용된 사람만이 프로그램을 이용할 수 있도록 하여야 하며 사용자의 사생활 보호 뿐만 아니라 프로그램을 이용한 사람에게는 이용한 분량에 따라 요금을 부과할 수 있는 방법이 필요하다. Pay-per-view 방식의 경우 정액 방식보다 복잡하나, 서비스의 유연성이 훨씬 뛰어나다.

Pay-per-view 방식은 허가받지 않은 사람으로부터의 프로그램 보호와 사용자의 사생활 보호, 그리고 요금 부과 방법에 대한 고찰이 필요하다. 허가받지 않은 사람으로부터 프로그램을 보호하여야 하는 문제는 프로그램을 암호화하고 정당한 키를 소유한 사람만이 프로그램을 볼 수 있도록 하여 해결할 수 있다. 사용자의 사생활 보호를 위해서는 사용자가 프로그램을 선택하는 내용을 암호화함으로써 사용자의 선택 프로그램을 사용자와 프로그램 제공자 외에는 알 수 없도록 할 수 있다. 요금 부과 방식은 특별한 주의를 필요로 한다. 가장 쉬운 요금 부과 방식은 각 프로그램에 요금을 책정하고 사용자가 그 프로그램을 선택하면 그에 상응하는 요금을 사용자에게 부과하는 방법이다. 그러나 이 방법은 사용자가 프로그램의 일부분만을 본다거나, 프로그램을 보다가 도중에 중지할 경우에도 프로그램에 책정된 모든 요금을 내야하므로 사용자에게 불리할 수 있다. 경우에 따라 인터넷 라인 불량 또는 서버의 용량 부족에 의해 사용자가 선택한 프로그램을 끝까지 보지 못하는 경우 논란의 여지가 있다. 이러한 문제를 해결하기 위해서는 사용자가 본 양만큼 요금을 낼 수 있도록 할 필요가 있다. 사용자는 프로그램을 중간부터 보거나 중간에 시청을 중단할 수 있다. 또한 Pause(일시 정지) 및 fast-forward(빨리 감기), rewind(되감기) 등을 지원할 필요가 있다. 이러한 경우 또한 공정성이 필요하다. 사용자가 자신이 시청한 프로그램에 따라 요금이 부과되었을 때 이에 대하여 승복할 수 없는 경우 이를 해결할 수 있는 방법이 필요하다. 이러한 경우 제공자는 사용자가 그 프로그램을 얼마나 시청하였는지 증명할 수 있어야 한다.

2. 관련 연구

A. Furche와 G. Wrightson에 의해 pay-per-view 프로토콜인 SubScrip이 제안되었다[1]. SubScrip은 소액 결제를 위한 선불 방식이다. 사용자는 제공자 시스템에 일정액을 지불하고 계좌를 개설한다. 매번 자료를 이용할 때마다 사용자의 계좌에서 일정액이 인출된다. SubScrip은 간단하다는 장점이 있으나 다음과 같은 여러 가지 문제점을 가지고 있다.

① 안전성
통신 내용이 암호화 되지 않으므로 타인에 의해 사용자의 계좌가 사용될 수 있다. 프로그램 자체도 또한 도청이 가능하다.
② 공정성
사용자로부터 일정한 금액이 인출되었을 때 그 인출된 금액에 대하여 사용자가 승복할 수 없는 경우가 생긴다. 이 경우 처리할 수 있는 방법이 없다.
③ 사생활 보호
통신 내용이 누구에게나 노출될 수 있으므로 사용자의 사생활이 보호되지 않는다.

J. Zhou 와 K-Y. Lam에 의하여 pay-per-view 방법이 제안되었고 이 방법은 공정성을 제공한다[2]. Zhou와 Lam의 방법은 다중 해쉬[3]를 사용하여 사용자의 시청양을 제공자가 증명할 수 있는 방법이다. 이 방법은 공정성을 제공한다는 장점이 있으나 몇 가지 미흡한 점을 가지고 있다.

① 사생활 보호
사용자가 무엇을 보았는지가 보호되지 않는다.

② 동기화
인증에 타임 스탬프를 이용하므로 제공자 시스템과 사용자 시스템의 시간 동기화가 필요하다.
③ 유연성
Fast-forward, rewind 등의 위치 변경을 처리할 수 없다.

3. 고려 사항

사용자는 프로그램을 선택하고 그 프로그램을 시청한다. 제공자는 사용자가 시청한 양(시간)만큼 요금을 부과한다. 이 과정에서 사용자는 자신이 시청한 양보다 더 많이 요금이 부과되었다고 주장하여 논란이 있을 수 있다. 이러한 경우에 제공자는 요금이 부과된 근거를 보일 수 있어야 한다.

사용자가 프로그램 시청을 요청하는 정보와 제공자로부터 사용자로 전송되는 프로그램은 타인에게 누출되지 말아야 한다. 이는 사생활 보호와 자료 보호의 측면에서 필요하다.

Fast-forward와 rewind 중에는 요금을 부과하지 않는다. 그리고 rewind 후 시청한 내용을 다시 시청하는 경우에는 그 부분에 대하여 요금을 부과하나 차등 요금을 부과할 필요가 있다. 즉, 사용자가 이미 한 번 본 내용을 다시 보는 것이므로 보다 낮은 요금을 부과하는 것이 필요하다.

4. 제안된 프로토콜

4.1 개요

제안된 프로토콜은 기밀성과 공정성을 보장하기 위하여 대칭키 암호 기법(RC4)[4], 공개키 암호 기법(RSA)[5], 전자 서명(RSA, SHA-1), 그리고 해쉬 알고리즘(SHA-1)[6]을 사용한다. 대부분의 실 자료 교환은 대칭키 알고리즘에 의하여 보호된다. 이 대칭키 알고리즘의 암호 키는 세션 키로 불리며 공개키 알고리즘에 의하여 보호되어 전달되고 전자 서명에 의하여 인증된다. Pedersen[2]에 의하여 제안되었으며 Zhou[3]에서도 사용된 기법인 해쉬 함수를 m번 수행한 다중 해쉬 기법을 요금 부과의 근거로 사용한다. 사용자는 프로그램을 처음 시청할 때 랜덤하게 생성된 정수 n을 m번 해쉬한 $H^m(n)$을 제공자에게 보낸다. n은 사용자가 보관하고 $H^m(n)$은 제공자가 보관한다. 프로그램을 시청하면서 $H^i(n)$를 전송한 후 일정 시간마다 사용자는 $H^{i-1}(n)$를 제공자에게 전송한다. 즉 서비스를 선택하여 제일 처음 프로그램을 보기 시작할 때에 $H^m(n)$을 보내며 그로부터 일정한 시간이 흐르면 $H^{m-1}(n)$을 보낸다. $H^{m-1}(n)$을 보낸 후 다시 일정한 시간이 흐르면 $H^{m-2}(n)$을 보낸다. $H^{m-j}(n)$를 제공자가 받았을 경우 이는 j만큼의 프로그램을 사용자가 시청하였음을 의미한다. 따라서 시청 시간에 논란이 있을 경우 제공자는 자신이 가지고 있는 $H^{m-j}(n)$를 j번 해쉬하여 $H^m(n)$을 생성함으로써 사용자가 j만큼의 프로그램을 사용자가 시청하였음을 증명할 수 있다. 이미 한 번 play가 된 내용을 replay하는 경우 차등된 요금을 부과하게 된다. 이를 위하여 사용자와 제공자 모두 한 번 play 된 위치(시작, 끝)를 기억하고 있을 필요가 있다. 논란에 대비하여 fast-forward나 rewind를 이용한 위치 변경이 있는 경우 사용자는 제공자에게 play가 중지된 위치와 새로 play가 될 위치를 전자서명하여 전송한다. 제공자는 사용자로부터 받은 중지된 위치와 자신이 계산한 중지된 위치와 일치하는가를 확인한다. 제공자가 계산하는 중지된 위치는 그 play가 시작된 위치와 위치 변경이 요청되는 시점까지 받은 해쉬값의 수에 의하여 계산된다.

프로토콜은 크게 서비스 선택 및 요청, play, fast-forward와 rewind를 이용한 위치 변경, rewind 후 play 시에 일어나는 replay로 나누어 생각할 수 있다.

서비스 선택 및 요청시에 사용자는 자신이 원하는 프로그램을 선택하여 제공자에게 서비스를 요청한다. 이 때에 이 세션에서 사용될 세션키 설정이 이루어진다. 키 설정이 완료되면 play를 시작하게 된다. Play 중에 제공자는 그 프로그램의 내용을 대칭키 알고리즘을 이용하여 세션키를 키로 하여 암호화하여 전송한다. 사용자는 세션키를 이용하여 복호화하여 그 프로그램을 시청한다. Fast-forward와 rewind는 동일한 위치 변경으로 취급한다. 사용자가 위치 변경을 요청할 때 사용자는 현재 play를 중단하는 위치와 새로운 play를 시작할 위치를 전자서명하여 제공자에게 전송한다. Replay 중에도 play와 마찬가지로 제공자는 그 프로그램의 내용을 대칭키 알고리즘을 이용하여 세션키를 키로 하여 암호화하여 전송한다. Fast-forward와 rewind를 반복하면 play 도중 replay 부분을 만나거나 replay 도중 play 부분을 만나게 된다. 이러한 경우 각각 play 모드와 replay 모드로 변환하여 필요한 작업을 수행하게 된다. 따라서 play와 replay 중에도 이미 play 되었던 영역을 다시 play 하는지 확인할 필요가 있다.

4.2 상세 프로토콜

프로토콜은 다음과 같다. 프로토콜 기술을 위하여 다음 표기법을 사용하기로 한다.

C->S: m : 사용자로부터 제공자로 메시지 m을 보낸다.
S->C: m : 제공자로부터 사용자로 메시지 m을 보낸다.
Id: 프로그램의 Title
Pr: 프로그램의 가격 (price)
randomC: 클라이언트가 생성한 랜덤 정수
randomS: 서버가 생성한 랜덤 정수
L: time slice의 크기 (default 1분으로 사용)
E: 암호화
K_s: session key
E_{ks}: 세션 키로 암호화
E_{ps}: 서버의 공개캐로 암호화
S: 전자 서명, 전자 서명은 메시지 부착형을 사용한다. 즉, SHA-1으로 해쉬한 후 RSA 비밀키로 암호화 함을 의미한다.
S_{sc}: 클라이언트의 비밀키로 전자 서명
S_{ss}: 서버의의 비밀키로 전자 서명
H: 해쉬
$H^i(n) = H(H^{i-1}(n))$, $H^0(n) = n$, 즉 $H^i(n)$는 n을 i번 해쉬 함을 의미한다.

가. 서비스 선택 및 요청

C->S: $E_{ps}(K_s)$, E_{ks}(Id, Pr, m, k, L, randomC)
S->C: E_{ks}(randomS, S_{ss}(Id, Pr, randomC, randomS))
C->S: E_{ks}(Id, $H^m(n)$, $H^k(r)$,
S_{sc}(Id, Pr, m, k, L, randomC, randomS))

여기에서 m*L = *length of the program* 의 관계가 성립한다.

서비스 선택 및 요청시에 사용자는 자신이 원하는 프로그램을 선택하여 제공자에게 서비스를 요청한다. 사용자는 웹에 공개된 제목, 표지, 가격, 예고편 등을 참조하여 프로그램을 선택하고 그 선택된 내용을 제공자에게 전송한다. 이 때에 이 세션에서 사용될 세션키 설정이 이루어진다. 서비스 요청 마지막 단계에서 사용자는 랜덤 정수 두 개를 생성하여 다중 해쉬하여 제공자에게 보낸다. 그 중 하나는 정상적인 play를 위하여 사용되고 다른 하나는 replay를 위하여 사용된다.

사용자는 세션키(K_s)를 제공자의 공개키로 암호화하여 전송한다. 또한 제목(Id), 가격(Pr), play를 위한 다중해쉬 횟수(m), replay를 위한 다중 해쉬 횟수(k), 요금 부과 시간 간격(L), 랜

덤 정수(randomC)를 세션키로 암호화하여 제공자에게 전송한다. 제공자는 자신의 비밀키를 이용하여 세션키를 복원한 후 그 세션키를 이용하여 제목 등 그 외의 정보를 획득한다. 제공자는 랜덤 정수(randomS)를 생성하고 제목, 가격, randomC, randomS를 전자 서명한 후 이들을 세션키를 이용하여 암호화하여 사용자에게 전송한다. 사용자는 복호화 후 전자 서명을 확인하여 키 교환이 성립하였음을 확인한다. 사용자는 제공자에게 두 랜덤 정수 n과 r을 생성하여 각각 m번과 k번 다중 해쉬한 값과 자신의 비밀키를 이용하여 제목, 가격, randomC, randomS를 전자 서명하여 제공자에게 보낸다. 제공자는 전자 서명을 통하여 올바른 사용자로부터 받은 내용임을 확인하여 키 설정을 완료한다. 사용자와 제공자가 모두 상대방이 생성한 랜덤 정수에 서명함으로 전자 서명에 기반한 challenge-response 기법[7]으로 사용자 인증이 키 설정과 동시에 수행된다. 키 설정이 완료되면 프로그램 내용을 세션키로 암호화하여 제공자에게 전송을 시작하고 play 모드로 전환한다.

나. Play

S->C: E_{ks}(contents)
C->S: E_{ks}(Id, P_flag, k, $H^{m-j}(n)$) (매 L 당)
여기에서 k는 프로그램의 현재 위치를 나타낸다.

제공자는 사용자에게 공유한 세션 키인 K_s로 암호화된 프로그램 내용을 전송한다. 사용자는 자신이 가지고 있는 K_s로 복호화하여 프로그램을 시청한다.

매 L 당 사용자는 제공자에게 $H^{m-j}(n)$을 전송하여 자신이 현재 프로그램을 시청하고 있음을 알린다. 즉 서비스를 선택하여 제일 처음 프로그램을 보기 시작할 때에 $H^m(n)$을 보내며 그로부터 L 만큼 시간이 흐르면 $H^{m-1}(n)$을 보낸다. $H^{m-1}(n)$을 보낸 후 다시 L 만큼 시간이 흐르면 $H^{m-2}(n)$을 보낸다. 제공자는 항상 가장 최근의 해쉬값을 보관하며, $H^{m-j+1}(n)$을 받은 후 L 만큼 시간이 경과하면 $H^{m-j}(n)$를 기대한다. $H^{m-j+1}(n)$을 받은 후 L 만큼 시간이 흐른 다음 $H^{m-j}(n)$을 받지 못하면 사용자가 더 이상 프로그램을 시청하지 않는 것으로 간주하여 프로그램 전송을 중단한다. 또한 $H(H^{m-j}(n)) = H^{m-j+1}(n)$ 이 성립하지 않는 경우 또한 올바른 사용자로부터 받은 값이 아니므로 전송을 중단한다. 그러나 매번 본 메시지를 받을 때마다 현재 위치(CP)를 L씩 증가시킨다. 메시지에서 받은 k와 계산된 CP는 일치해야 한다.

다. 위치 변경(Rewind, Fast-Forward)

C->S: E_{ks}(Id, C_flag, o, n, S_{sc}(Id, C_flag, o, n))

Fast-forward와 rewind는 동일한 위치 변경으로 취급한다. 위치 변경을 원할 때 사용자는 현재 play를 중지하고자 하는 위치(o)와 새로 play를 시작하고자 하는 위치(n)를 제공자에게 전자서명하여 전송한다. 사용자가 전송한 중지 위치는 제공자가 해쉬를 받은 횟수에 의하여 계산된 중지 위치와 일치하여야 한다. 제공자는 play의 시작위치와 다중 해쉬를 받은 횟수를 이용하여 현재 위치(중지 위치)를 계산한다. 이 두 위치는 일치하여야 한다. 제공자는 중지 위치를 기억함으로써 지금까지 한 번 이상 play된 프로그램의 위치를 기억한다. 그리고 새로 play될 위치를 기억하여 새로운 play 영역을 계산할 수 있도록 한다. 이러한 play된 영역 저장을 이용하여 특정 부분이 play에 사용되는지 replay에 사용되는지 판별한다. 사용자도 동일한 위치 정보 가진다.

라. Replay

S->C: E_{ks}(contents)
C->S: E_{ks}(Id, R_flag, k, $H^{m-j}(r)$) (매 L 당)
여기에서 k는 프로그램의 현재 위치를 나타낸다.

기본적으로 play와 동일하다. 차이점은 play의 경우 play를 위한 랜덤 정수의 다중 해쉬($H^{m-j}(n)$)를 사용하나, replay의 경우 replay를 위한 다중 해쉬($H^{k-i}(r)$)를 사용한다.

만약 현재 위치(CP)가 replay 영역을 벗어나면 play 모드로 변환하게 된다. Play 모드를 수행하던 중 replay 영역을 만나면 마찬가지로 replay 모드로 변환된다. Fast-forward와 rewind를 반복하면 play 도중 replay 부분을 만나거나 replay 도중 play 부분을 만나게 된다. 이러한 경우 각각 play 모드와 replay 모드로 변환하여 필요한 작업을 수행하게 된다. 따라서 play와 replay 중에도 이미 play 되었던 영역을 다시 play 하는지 확인할 필요가 있다.

5. 결론

이 연구에서는 pay-per-view 방식의 유료 웹 서비스를 위한 프로토콜이 개발되었다. Pay-per-view 방식에서 유의해야 할 것은 기밀성과 요금 부과의 공정성이다. 제안된 프로토콜은 대칭키 암호 방식을 이용하여 기밀성을 제공하고 전자 서명과 다중 해쉬를 이용하여 공정성을 제공한다. 대칭키 암호 기법의 키 설정은 공개키 암호 방식을 이용하며, 전자 서명에 기반한 challenge-response 기법으로 사용자 인증이 키 설정과 동시에 수행된다. 또한 다양한 위치 변경 및 play와 replay에 대한 차등 요금 적용 등 유연성을 제공한다.

참고문헌

[1] Furche, A. and Wrightson, G., "SubScrip - An Efficient Payment Mechanism for Pay-Per-View Services on the Internet," Proceedings of the *5th IEEE International Conference for Computer Communication and Networks*, pp. 369-373, 1996, Maryland, USA.

[2] Zhou, A. and Lam, K-Y., "A Secure Pay-per View Scheme for Web-Based Video Service," *Proceedings of the 2nd International Workshop on Practice and Theory of Public Key Cryptography, LNCS 1560*, pp. 315 - 326, 1999, Kamakura, Japan.

[3] Pedersen, T., "Electronic Payments of Small Amounts," *Proceedings of Cambridge Workshop on Security Protocols, LNCS 1189, pp. 59-68*, 1996, Cambridge, U. K.

[4] B. Schneier, *Applied Cryptography*, Wiley, 1996.

[5] R. Rivest, A. Shamir, and L. M. Adleman, "A method for digital signature and public-key cryptosystems," *Communications Of the ACM*, vol. 21, pp. 120-126, 1978.

[6] NIST, *FIPS 180-1, Secure hash standard, Federal Information Processing Standards Publication 180-1*, U.S. Dept. of Commerce / NIST, 1995.

[7] ISO/IEC, *ISO/IEC 9798-3, Information Technology Security techniques Entity authentication mechanism Part 3: Entity authentication using a public-key algorithm*, International Organization for Standardization, 1993.