韓國 컴퓨터情報學會 論文誌
第8卷 第3號, 2003. 9.

2003-8-3-3-3

RTP를 위한 보안 제어 프로토콜 구현

홍 종 준*

Implementation of Security Control Protocol for Real-Time Protocol

Jong-Joon Hong*

요 약

멀티미디어 데이터는 실시간 제약을 갖고 있기 때문에 암호화/복호화로 인한 지연이 실시간 제약에 미치는 영향을 최소로 하면서 암호화를 하기 위해서는 네트워크 트래픽과 부하에 적응하여 암호화 알고리즘을 변경하기 위한 방법이 필요하다. 또한 다수가 참여하는 멀티미디어 서비스 진행 중에 서비스 이용을 중지한 사용자는 RTP payload의 암호화 키를 알고 있기 때문에 이 사용자로부터 RTP payload를 보호하기 위해서는 암호화 키를 변경하기 위한 방법이 필요하다. 따라서 본 논문에서는 RTP payload의 암호화를 위해 암호화 알고리즘과 암호화 키를 변경하기 위한 RPT 보안제어 프로토콜을 설계하고 구현하였다.

Abstract

Encryption/decryption delay is minimized, because there are constraints in transporting a multimedia data through the Internet. Therefore, encryption algorithm is needed which is changed with considering network traffic and load. And during many users participate in the same multimedia service, an user who already left the service can receive and the method which decrypt the RTP payload is needed because of knowing the encryption key. Therefore in this paper, Security Control Protocol for RTP is designed and implemented for changing the encryption algorithm and the key.

▸ Keywords : RTP(Real-Time Protocol), Security Control Protocol

* 평택대학교 정보과학부 정보통신학전공 교수

Ⅰ. 서 론

RTP(Real-time Transport Protocol)는 실시간 정보를 전송하는 응용 프로그램에서 요구되는 네트워크의 양단간 멀티미디어 데이터의 전송을 위한 기능을 제공하기 위한 프로토콜이다. RTP는 모든 수송 계층의 기능을 수용하기 보다는 응용 프로그램의 측면에서 하부 망의 기능을 수정 보완하는 기능을 가지고 있다. 일반적으로 응용 프로그램은 UDP에서 제공하는 다중화 기능 및 체크섬 기능 등을 활용하기 위해 UDP상에서 RTP를 실행시킨다[1].

그러나 현재의 인터넷은 전송되는 데이터를 보호할 수 없기 때문에 주문형 비디오 서비스나 비공개 화상 회의와 같은 멀티미디어 서비스의 보호를 위해 RTP의 페이로드를 암호화해야 한다. RTP 페이로드의 암호화는 암호화/복호화 하는 시간으로 인해 멀티미디어 서비스의 실시간 특성에 영향을 미친다. 따라서 멀티미디어 서비스의 실시간 특성에 미치는 영향을 최소로 하면서 암호화를 하기 위해서는 네트워크의 트래픽과 부하에 적응하면서 암호화 알고리즘을 변경하기 위한 방법이 필요하다. 또한 다수가 참여하는 멀티미디어 서비스 진행 중에 서비스 이용을 중지한 사용자는 사용한 암호화 키를 알고 있기 때문에 멀티미디어 서비스를 위해 사용된 RTP 페이로드를 수신하여 복호화 할 수 있다. 그러므로 멀티미디어 서비스 진행 중에 서비스 이용을 중지한 사용자로부터 멀티미디어 서비스를 보호하기 위해 키를 변경하는 방법이 필요하다[4][9][11].

본 논문에서는 멀티미디어 정보 보호를 위해, RTP 페이로드의 암호화를 위한 암호화 알고리즘과 암호화 키를 변경하기 위한 프로토콜인 RTP 보안제어프로토콜을 제안한다.

Ⅱ. RPT 데이터의 암호화

멀티미디어 데이터의 실시간 특성에 영향을 최소로 하기 위해서는 대칭키 알고리즘을 사용한다[5][7][8]. 그러나 멀티미디어 서비스의 처음부터 끝까지 동일한 암호화 알고리즘과 동일한 암호화 키를 사용하여 RTP 페이로드를 암호화하는 경우 페이로드의 암호화/복호화로 인한 일정한 지연이 추가된다. 이러한 지연은 네트워크 트래픽과 부하에 따라 멀티미디어 데이터의 실시간 제약에 많은 영향을 준다. 또한 멀티미디어 서비스 이용을 중지한 사용자가 RTP 페이로드의 암호화 키를 알고 있기 때문에 서비스 이용을 중지한 사용자로부터 멀티미디어 서비스를 보호하지 못하는 문제점이 있다. 따라서 문제를 해결하기 위해서는 암호화 알고리즘과 키를 변경하기 위한 방법이 필요하다.

Ⅲ. RTP 보안 제어 프로토콜 제안

RTP 세션 도중에 세션에서 나간 사용자에게 RTP 페이로드를 보호하기 위해 암호화 키를 바꿔야 한다. 그렇지 않으면 세션에서 나간 사용자가 RTP 페이로드를 받아 볼 수 있다. 이런 문제를 해결하기 위해 본 논문에서는 암호화 알고리즘과 암호화 키를 바꾸는 메카니즘을 제공하는 RTP 보안제어프로토콜을 제안하였다.

RTP 보안 제어 프로토콜은 RTP 세션에 참여하려는 사용자들과 TCP 연결을 설정하고, RTP 보안제어프로토콜의 채널을 보호하기 위해 RTP 보안제어프로토콜을 사용하는 응용이 암호화 알고리즘을 선택하여 암호화 하거나 TLS(Transport Layer Security)[6]을 사용하여 RTP 보안 제어 프로토콜의 채널을 보호한다. RTP 보안 제어 프로토콜의 채널 보호는 RTP 세션 보호와 밀접한 관계가 있고 RTP 보안 제어 프로토콜의 데이터는 실시간 제약성을 갖지 않으므로 RTP 보안제어프로토콜 채널은 암호화 알고리즘에 의해 보호된다.

1. RTP 보안 제어 프로토콜의 헤더

RTP 보안 제어 프로토콜의 헤더는 <그림 1>과 같고 각 필드에 대한 설명은 다음과 같다.

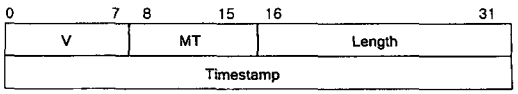

그림 1. RTP 보안 제어 프로토콜 헤더
Fig. 1 The Header of RTP security control protocol .

- 버전(V) : 현재 RTP 보안 제어 프로토콜의 버전
- 메시지 타입(MT) : 전송되는 메시지의 타입
- 길이(Length) : 실제 데이터의 바이트 수
- 타임스탬프(Timestamp) : 패킷의 라운드 트립 시간 계산과 재전송 공격[10]을 막기 위해 사용

2. RTP 보안 제어 프로토콜의 데이터 타입

RPT 보안제어 프로토콜의 각 타입에 대한 설명은 아래와 같다.

2.1 Authentication

사용자 인증 프로토콜에 의해 사용되고, RTP 보안 제어 프로토콜을 사용하는 응용에서 설정한다. 인증 과정에서 RTP 보안 제어 프로토콜 채널 보호를 위해 사용되는 암호화 알고리즘의 암호화 키를 받아온다.

2.2 SSRC Info

SSRC Info는 RTP 세션에서 사용할 클라이언트의 SSRC 식별자 를 전달하는 데이터 구조이다. SSRC 식별자를 각 클라이언트에서 임의의 값으로 선택하는 경우 RTP 세션에 참여한 클라이언트 사이에 SSRC 식별자의 충돌이 발생할 수 있다. 이와 같은 SSRC 식별자 충돌은 각 RTP 구현에서 검출하고 해결해야 한다[11].

2.3 Init Info

Init Info는 RTP 페이로드의 보호를 위해 사용하는 암호화 알고리즘에 의해 사용될 암호화키와 시작 알고리즘의 인덱스 번호를 사용자들에게 전달하는 메시지 타입이다. Init Info의 데이터 구조는 <그림 2>와 같다.

Init_index는 처음 RTP 페이로드를 암호화할 암호화 알고리즘에 대한 인덱스 번호이고 Key_number는 전송되는 Key_length와 Key의 개수를 나타낸다. Key의 개수와 암호화 알고리즘의 수는 같으므로 Key_number는 사용하는 암호화 알고리즘의 개수를 나타낸다.

사용되는 암호화 알고리즘의 종류는 SCPR을 사용하는 응용에서 설정한다. Key_Length n은 Key n의 길이를, Key n은 n번째 암호화 알고리즘에서 사용하는 키를 나타내고($1 \leq n \leq$ Key_number, n은 정수) Key 값의 설정은 응용에서 선택한다.

Init_Index (4 비트)	Key_number (4 비트)	Key_Length1 (8 비트)	Key 1
Key_Length 2 (8 비트)	Key 2		...

그림 2. Init Info의 데이터 타입
Fig. 2 Data type of Init Info

2.4 Algorithm Change Request

클라이언트가 자신이 사용하는 암호화 알고리즘을 변경하고자 할 때 RTP 보안 제어 프로토콜 서버에 전송하는 데이터 타입이다. 변경되는 암호화 알고리즘은 현재 네트워크 트래픽 상태와 부하에 따라 응용에 의해 가장 적합한 암호화 알고리즘이 선택된다. 암호화 알고리즘의 선택에 관한 세부 사항은 RTP 보안 제어 프로토콜을 사용하는 응용에 의해 구현되어야 한다. RTP 보안 제어 프로토콜 식별자는 이 데이터 타입을 보낸 송신자의 RTP 보안 제어 프로토콜 식별자이고 Change_Index는 변경하고자 하는 알고리즘의 인덱스 번호이다.

2.5 Algorithm Change

Algorithm Change는 모든 클라이언트들에게 특정 클라이언트나 모든 클라이언트에서 사용하는 암호화 알고리즘의 변화를 알려주기 위해 사용된다.

Waiting Time은 새로운 암호화 알고리즘의 사용 시점을 동기화하기 위해 사용된다. 클라이언트들은 Algorithm Change 메시지를 받은 시점부터 Waiting Time의 시간만큼 기다린 후에 Change_Index에 나타난 암호화 알고리즘을 사용한다. Waiting Time은 각 클라이언트의 라운드 트립 시간 중 가장 큰 값에서 각 클라이언트의 라운드 트립 시간을 뺀 값이다.

이 값은 각 클라이언트로 보내지는 패킷마다 다른 값을 가진다. 멀티미디어 데이터는 일부 데이터의 손실이 있어도 문제가 되지 않으므로 새 알고리즘 시작의 동기화가 정확하게 이루어지지 않아 몇 개의 RTP 패킷이 손실되더라도 문제가 되지 않는다. SSRC 필드는 암호화 알고리즘을 변경하려는 클라이언트의 SSRC 식별자이다. 이 값이 0이면 모든 클라이언트가 현재 사용하고 있는 암호화 알고리즘을 Change_Index에 명시된 알고리즘으로 바꾼다.

Waiting Time (32 비트)	SSRC (32 비트)	Change_index (4 비트)

그림 3. Algotithm Change의 데이터 구조
Fig. 3 Data structure of Algorithm Change

2.6 Key Change

Key Change는 암호화 알고리즘에서 사용하는 암호화 키를 변경하기 위해 사용되며 데이터 구조는 〈그림 4〉와 같다.

Waiting Time은 Algorithm Change에서와 같이 새 암호화 키 사용에 대한 동기화를 위해 사용된다.

Key_Index t는 암호화 키를 바꿀 암호화 알고리즘에 대한 인덱스 값, Key_Length t는 Key t필드의 길이(1〈= t 〈= n, n = Key_Num), Key는 변경할 키이다.

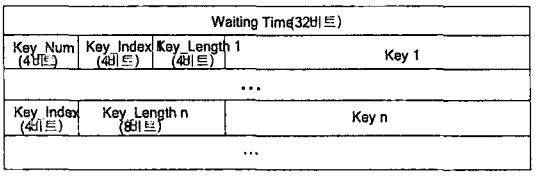

그림 4. Key Change의 데이터 구조
Fig. 4 Data Structure of Key Change

2.7 Packet Ack

Packet Ack는 서버와 각 클라이언트 사이의 라운드 트립 시간을 측정하기 위해 모든 클라이언트에 의해 서버로 보내지며 이의 데이터 구조는 〈그림 5〉와 같다. MT는 이 패킷이 어떤 패킷에 대한 응답인지를 나타내기 위해 사용되는 필드로 Init Info, Algorithm Change와 Key Change의 값을 가질 수 있다. SSRC 식별자는 어떤 클라이언트가 데이터를 보냈는지 구별하고 각 클라이언트의 라운드 트립 시간을 계산하기 위해 사용된다. Sender RTT는 서버가 보낸 패킷이 클라이언트에게 도착하는데 걸리는 시간으로 전 패킷을 받은 시간에서 전 패킷의 Timestamp 필드값을 뺀 값이다. 평균 라운드 트립 시간은 다음과 같이 계산한다.

$$RTT = \frac{t_1 + t_2}{2}$$

(RTT는 평균 라운드 트립 시간, t1은 Packet Ack 패킷이 서버에 도착한 시간에서 Packet Ack 패킷의 Timestamp 필드 값을 뺀 값, t2는 Sender RTT 필드 값)

MT (8 비트)	SSRC (32 비트)	Sender RTT (32 비트)

그림 5. Packet Ack의 데이터 구조
Fig. 5 Data structure of Packet Ack

2.8 Session Leave Request, Leave, End

Session Leave는 클라이언트가 RTP 세션을 떠나기 전에 서버에 전달하는 메시지이다. Session Leave는 서버가 특정 클라이언트가 RTP 세션에서 떠났다는 것을 다른 클라이언트들에게 알려주는 메시지이다. Session End는 RTP 세션이 종료될 때 서버가 현재 세션에 참여하고 있는 모든 클라이언트에게 보내는 메시지이다.

3. RTP 보안 제어 프로토콜의 동작 절차

3.1 클라이언트가 RTP 세션에 참여하기 전의 절차

RTP 세션에 참여하기 전에 클라이언트는 인증 프로토콜을 사용하여 인증을 받은 후, RTP 보안 제어 프로토콜의 SSRC Info와 Init Info 패킷을 서버로부터 받아 SSRC 식별자, 처음 사용할 암호화 알고리즘의 인덱스와 각 암호화 알고리즘에서 사용할 암호화 키 값을 얻는다. 클라이언트는 Init Info 패킷을 받은 후에 서버에 Packet Ack 패킷을 보내 서버가 클라이언트와의 라운드 트립 시간을 계산하도록 한다. 그 후에 클라이언트는 RTP 세션에 참여한다.

3.2 클라이언트의 암호화 알고리즘 변경 절차

암호화 알고리즘을 변경하려는 클라이언트는 서버에 Algorithm Change Request 패킷을 보내고 이 패킷을 받은 서버는 모든 클라이언트에게 Algorithm Change 패킷을 보낸다. 패킷을 받은 클라이언트는 서버에 Packet Ack를 보내고 이를 받은 서버는 각 클라이언트와의 라운드 트립 시간을 계산한다. 클라이언트가 Algorithm Change 패킷을 받고 Waiting Time 만큼 기다린 후 새 암호화 알고리즘을 사용한다.

3.3 암호화 키 변경 절차

암호화 키를 변경하는 경우에는 서버가 모든 클라이언트에게 Key Change 패킷을 전송하고 패킷을 받은 클라이언트는 서버에 Packet Ack를 보낸다. 이를 받은 서버는 각 클라이언트와의 라운드 트립 시간을 계산한다. 클라이언트가 Key Change 패킷을 받은 후에 Waiting Time 만큼 기다린 후 새 암호화 키를 사용한다.

3.4 세션 종료와 클라이언트가 세션을 떠나는 절차

RTP 세션을 떠나려는 클라이언트는 서버에 Session Leave Request 패킷을 보낸다. 이를 받은 서버는 클라이언트들에게 Session Leave 패킷을 보내 RTP 세션을 떠났

다는 것을 알려준다. 그 후 서버는 Key Change 패킷을 보내 클라이언트의 암호화 키를 변경하도록 한다. 새 라운드 트립 시간을 계산하기 위해 패킷을 받은 클라이언트들은 서버에 Packet Ack 패킷을 보낸다. 또한 RTP 세션이 끝난 경우에 서버는 세션에 참여한 모든 클라이언트들에게 Session End 패킷을 보내 RTP 세션이 끝났다는 것을 알려준다.

IV. RTP 보안 제어 프로토콜 구현 및 실험 결과

구현한 RTP 보안 제어 프로토콜 서버와 클라이언트의 동작을 확인하기 위하여 TCP 패킷을 tcpdump프로그램을 사용하여 dump한 결과는 다음과 같다.

〈그림 6〉은 B의 요청에 의해 암호화 알고리즘을 변경하는 절차를 보인다.

```
• 단계 1 : B가 D에게 알고리즘 변경 요청
 14:05:58.657415 B.2040 > D.500: P 21:37(16) ack
77 win 8684 (DF)
• 단계 2 : D는 모든 클라이언트들에게 알고리즘 변경통보
 14:05:58.658145 D.500 > B.2040: P 77:97(20) ack
37 win 8724 (DF)
 14:05:58.658200 D.500 > A.2775: P 77:97(20) ack
21 win 8716 (DF)
• 단계 3 : 클라이언트들이 D에게 응답 메시지 송신
 14:05:58.659118 A.2775 > D.500: P 21:41(20) ack
97 win 8640 (DF)
 14:05:58.659530 B.2040 > D.500: P 37:57(20) ack
97 win 8664 (DF)
 14:05:58.844028 D.500 > A.2775: . ack 41 win
8696 (DF)
 14:05:58.844087 D.500 > B.2040: . ack 57 win
8704 (DF)
```

그림 6 Algorithm Change Request와 Algorithm Change의 실험결과
Fig. 6 Simulation result of Algorith, Change Request and Algorithm Change

〈그림 7〉는 암호화 키를 변경하는 절차를 보인다.

```
• 단계 1 : D가 클라이언트들에게 키 변경 통보
 14:06:08.314173 D.500 > B.2040: P 97:165(68) ack
57 win 8704 (DF)
 14:06:08.314250 D.500 > A.2775: P 97:165(68) ack
41 win 8696 (DF)
• 단계 2 : 클라이언트들이 D에게 응답메세지 송신
 14:06:08.315191 A.2775 > D.500: P 41:61(20) ack
165 win 8572 (DF)
 14:06:08.315919 B.2040 > D.500: P 57:77(20) ack
165 win 8596 (DF)
 14:06:08.468622 D.500 > A.2775: . ack 61 win
8676 (DF)
 14:06:08.468684 D.500 > B.2040: . ack 77 win
8684 (DF)
```

그림 7. Key Change의 실험결과
Fig. 7 Simulation result of Key Change

〈그림 8〉는 B가 세션을 떠나는 절차를 보인다.

```
• 단계 1 : B가 D에게 Session Leave Request 메시지 전송
 14:06:14.620299 B.2040 > D.500: P 77:89(12) ack
165 win 8596 (DF)
• 단계 2 : D는 모든 클라이언트들에게 B의 session Leave를
알리기 위해 Session Leave 메시지 전송 및 연결 해지
 14:06:14.620929 D.500 > B.2040: P 165:177(12)
ack 89 win 8672 (DF)
 14:06:14.620984 D.500 > A.2775: P 165:177(12)
ack 61 win 8676 (DF)
 14:06:14.621253 D.500 > B.2040: F 177:177(0) ack
89 win 8672 (DF)
 14:06:14.621602 B.2040 > D.500: . ack 178 win
8584 (DF)
 14:06:14.804743 A.2775 > D.500: . ack 177 win
8560 (DF)
• 단계 3 : D는 클라이언트에게 Key Change 메시지 전송
 14:06:14.804920 D.500 > A.2775: P 177:245(68)
ack 61 win 8676 (DF)
 14:06:14.805821 A.2775 > D.500: P 61:81(20) ack
245 win 8492 (DF)
 14:06:14.921479 D.500 > A.2775: . ack 81 win
8656 (DF)
```

그림 8. Session Leave Request, Session Leave의 실험결과
Fig. 8 Simuation result of Session Leave Request and Session Leave

구현 결과, 클라이언트의 암호화 알고리즘 변경 요청에 의해서 서버는 모든 클라이언트들에게 특정 클라이언트의 암호화 알고리즘의 변화를 전달됨을 확인하였다. 또한, 특정 클라이언트가 멀티미디어 서비스 진행 중에 서비스 이용을

중지하는 경우 RTP 보안 제어 프로토콜 서버가 다른 클라이언트에서 사용하는 암호화 키를 변경함을 확인하였다.

V. 결 론

본 논문에서는 암호화 알고리즘과 암호화 키를 변경할 수 있는 RTP 보안 제어 프로토콜을 설계 구현하였다. 클라이언트는 적당한 암호화 알고리즘을 선택하여 서버에 Algorithm Change Request 메시지를 보내고 서버는 모든 다른 클라이언트들에게 Algorithm Change 메시지를 보내 특정 클라이언트의 암호화 알고리즘의 변경을 다른 참가자들에게 전달하여 암호화 알고리즘을 변경하였다. 또한, 특정 클라이언트가 멀티미디어 서비스의 이용을 중지한 후에 서버는 Key Change 메시지를 다른 클라이언트들에게 전송하여 암호화 키를 변경하도록 하였다. 따라서 멀티미디어 서비스 이용을 중지한 클라이언트로부터 서비스를 보호할 수 있었다. 실제 망에서 실험한 결과 본 논문에서 제안한 방식이 네트워크 트래픽과 컴퓨터 부하에 적응하기 위해 암호화 알고리즘을 변경하고, 암호화 키를 변경하여 RTP 데이터를 보호할 수 있음을 확인하였다.

참고문헌

[1] 이상현, 도미선, 이재용, "인터넷에서의 멀티미디어 통신과 서비스," 한국통신학회지, 제 13권, 제 2호, pp. 65 - 81, 1996.

[2] Sousa P. Freitas V, "A Framework for the D-evelopment of Tolerant Real-Time Application," Computer Networks & Isdn Systems, Vol. 30 No. 16-18, pp. 1531 - 1541, Sept. 1998.

[3] Raj Jain, "Multimedia over IP : RSVP, RTP, RTCP, RTSP," http://www.cis.ohio-state.edu/~jain/ci-s788-97/ip_multimedia/index.htm.

[4] Chi-Sung Laith, Sung-Ming Yen, "On the Design of Conference Key Distribution Systems for the Broadcasting Networks," Proc. of the IEEE Infocom'93, Vol. 3, pp. 1406-1413, May. 1993.

[5] Aikawa M, Takaragi K, Furuya S, Sasamoto M, "Lightweight Encryption Method Suitable for Copyright Protection," IEEE Transactions on Consumer Electronics, Vol .44, No. 3, pp. 902-910, Aug. 1998.

[6] G. Mamais, M. Markaki. M. H. Sherif, G. Stassinopoulos, "Evaluation of the Casner-Jacobson Algorithm for Compressing the RTP/UDP/IP Headers," Proc. of the Third IEEE Symposium om Computers and Communications, pp. 543 - 548, Jun. 1998.

[7] Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone, Handbook of Applied Cryptography, CRC Press, 1996.

[8] Charles P. Pfleeger, Security Computing, Prentice Hall, 1997.

[9] Oppliger R, Albanese A, "Participant Registration, Validation, and Key Distribution for Large-Scale Conferencing Systems," IEEE Communication Magazine, Vol. 35 No. 6, pp. 130-134, Jun. 1997.

[10] H. Schulzrinne, "RTP : A Transport Protocol for Real-Time Application," RFC1889, Jan. 1996.

[11] Shoichiro Seno, Akira Watanabe, Yuuji Koui, Masayuki Yabe, and Tetsuo Ideguchi, "Intranet Packet Encryption with Minimum Overheads," Proc. of the 13th ICCC, pp. 517-521, Nov. 1997.

저 자 소 개

홍 종 준

1991 인하대학교 전자계산공학과 졸업

1993 인하대학교 대학원 전자계산공학과(공학석사)

2002 인하대학교 대학원 전자계산공학과(공학박사)

2003 ~ 현재 평택대학교 정보과학부 정보통신학전공 교수

〈관심분야〉 네트워크보안, QoS라우팅, 분산시스템