한국정보전자통신기술학회논문지(jkiiect)'18-02, Vol.11 No.1
논문 18-11-01-121 http://dx.doi.org/10.17661/jkiiect.2018.11.1.121

IoT 통신 환경을 위한 경량 IPsec 프로토콜 연구

송인아*, 오정현*, 이두원*, 이영석**

Lightweight IPsec protocol for IoT communication environments

In-A Song*, Jeong-Hyeon Oh*, Doo-Won Lee*, Young-Seok Lee**

요 약 사물인터넷(IoT) 기술은 사람과 장치 간 통신 기술 및 데이터 취득을 위한 센싱 기술, 가상의 프로세스, 저장 데이터 등 모든 것들이 인터넷으로 연결되어 활용 되는 기술이다. 최근 IoT 관련 프로토콜에 대해 연구가 많이 진행되고 있지만, IoT 보안을 위한 경량 프로토콜에 대한 연구는 미흡한 실정이다. IoT 프로토콜들의 보안성을 위해 추가로 연구되는 부분이 있지만 이는 메모리 사용 및 통신량의 단점을 갖고 있기 때문에 적합하다고 할 수 없다. 이러한 이유로 IoT의 자원 제약적인 특성상 기존의 프로토콜들이 경량화 추세로 가고 있으며 경량화 IPsec 연구는 필수적이라 할 수 있다. 본 논문에서는 기존 경량 IPsec을 분석하고 취약점을 보완한 새로운 경량 IPsec을 제안한다. 그리고 성능 분석 및 성능 평가를 통하여 기존 경량 IPsec 프로토콜과의 차이점을 분석하였다.

Abstract Internet of Things architecture connected to the Internet is a technology. However, Many paper research for the lightweight Protocol of IoT Environment. In these Paper excluded secure problem about protocol. So Light weight Protocol has weakness of secure in IoT environment. All of IoT devices need encryption algorithm and authentication message code for certain level of security. However, IoT environment is difficult to using existing security technology. For this reason, Studies for Lightweight IPsec is essential in IoT environment. For Study of Lightweight IPsec, We analyze existing protocols such as IPsec, 6LoWPAN for IEEE 802.15.4 layer and Lightweight IPsec based 6LoWPAN. The result is to be obtained for the lightweight IPsec protocols for IoT environment. This protocol can compatible with Internet network.

Key Words : IKEv2, IoT, IPsec, Lightweight, Security Protocol

1. 서론

사물인터넷(IoT)이 지향하는 스마트 세상을 만들기 위해서는 자원이 제한적인 일상의 사물들과 이들이 연결되는 저전력 네트워크 및 인터넷 연동을 수용하는 IoT 보안 기술이 제공되어야 한다.

최근 IoT 보안을 위해 경량화 프로토콜인 6LoWPAN을 중심으로 IPsec 적용가능성에 대한 연구가 진행되고 있다.

IPsec은 Network Layer에서 동작하므로 Application과는 무관하게 동작할 수 있으며 HTTP에서만 동작하는 SSL과는 다르게 IPsec은 표준화 프로토콜의 대해 보안을 보장할 수 있게 된다. 그리고 Ethernet, TokenRing, PPP등 모든 Network Topology에 사용가능하며 L2TP, L2F 등으로 구현될 수 있는 PPTP같은 Tunneling Protocol과는 다르게 IPsec은 표준화된 터

This work was supported in part by MSIT & NIPA.(No.C1605-17-1002, Development of low cost and high precision industrial asset management solution based on location service)
*Department of Information & Communicatin Engineering, Kunsan National University
**Corresponding Author : Department of Information and Communication Engineering, Kunsan National University(leeys@kunsan.ac.kr)
Received September 25, 2017 Revised October 11, 2018 Accepted February 31, 2018

널링, 인증, 암호화 방법을 사용한다.

대표 관련 연구로서 Shahid Raza는 6LoWPAN 프로토콜에서 적용할 수 있는 경량화 IPsec을 제안하였다.[1,2,3] Raza의 제안 방식은 IPsec의 대표모드인 Tunnel 모드를 지원하지 않고 Transport 모드가 지원하기 때문에 기존 IP 계층에서 사용하던 IPsec을 완벽하게 적용했다고는 할 수 없다.

본 논문에서 제안하는 방식은 Raza의 제안방식에서 적용하지 못한 Tunnel Mode를 적용하고 IoT 환경에서 사용 가능할 수 있는 경량 IPsec을 위해 헤더 압축과 필드 추가 · 삭제를 진행하였다.

본 논문의 2장에서는 Shahid Raza가 제안한 경량 IPsec을 살펴본다. 3장에서는 본 논문에서 제안한 경량 IPsec을 살펴보고, 4장에서는 제안 프로토콜의 효율성을 연산량, 통신량 측면에서 비교한다. 5장에서는 구성한 환경을 토대로 성능평가를 진행하고, 마지막으로 6장에서는 결론과 향후 연구 방향을 제시한다.

2. 관련 연구

기존방식에서는 6LoWPAN에서 AH에 대한 특정 비트를 지정하여 AH를 나타내려고 하였다. 이는 LOWPAN_NHC 안에 포함되어 있는 비트 지정방식을 이용하게 된다. 또한 payload Length field를 항상 제거하는 것을 제안하는데 그 이유는 Lower Layers (IEEE 802.15.4 헤더나 6LoWPAN 헤더)에서 길이는 추정 가능하기 때문이다.

ICV (Integrity Check Value)의 크기는 SPI 값에서 포함된다. 인증 데이터의 길이는 알고리즘에 따라 결정되며 출력 값은 알고리즘에 따라 항상 고정된 출력 값을 결정하기 때문이다.

0	1	2	3	4	5	6	7
1	1	0	1	SPI		SN	

그림 1. Razz 제안 AH 방식
Fig. 1. Proposed LOWPAN_NHC encoding for AH

기존 NHC에서 NH의 4bit에서의 Reserved 부분은 제거되며, Reserved 부분 중 하나가 AH를 나타내는 부분으로 사용된다. SPI와 SN은 6LoWPAN의 Compress Encoding 방식에 따라 압축된다.

처음 4비트인 1101은 Next Header Type에서 AH를 나타내며 SPI의 압축 방법은 2bit로 4가지의 경우로 나타내게 된다. 만약 SPI가 00일 경우 IEEE 802.15.4에서 사용하는 SPI 방식을 사용하며 이는 SPI 값이 모두 제거되는 것이며 모든 노드가 같은 SA (Security Association)를 사용하지 않는다는 것을 의미한다. 만약 SPI가 01일 경우 32bit의 SPI 중 8bits의 SPI가 NHC 다음에 나타나게 되며 24비트의 SPI는 제거된다. 만약 SPI가 10일 경우 32bit 중의 SPI 중 16비트의 SPI는 NHC 다음에 나타나게 되고, 나머지 16비트의 SPI는 제거된다. 만약 SPI가 11일 경우 32비트의 SPI는 모두 NHC 다음에 나타나게 된다.

SN(Sequence Number)의 표현 방식은 SPI의 방식과 같은 방식인 4가지 방식을 갖도록 2bit로 나타낸다. SN에 경우 00일 경우 32bit 중 8bits의 SN이 NHC 다음에 나타나게 되며 나머지 비트는 제거된다. SN이 01일 경우 16비트는 NHC 다음에 나타나게 되며 16비트는 제거된다. SN이 10일 경우 24비트의 SN은 NHC 다음에 나타나게 되고 8비트는 제거되게 된다. SN이 11일 경우 32비트의 SN은 모두 NHC 다음에 나타나게 된다. SN 부분은 AH 패킷에 포함되며 처음의 bit의 값이 1인 경우 IPsec의 SA를 위해 사용된다.

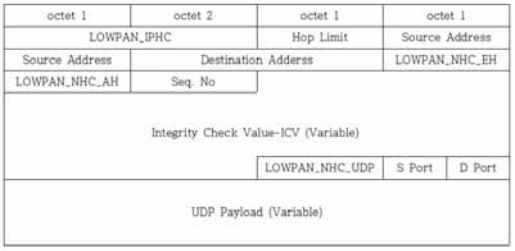

그림 2. AH를 적용한 압축 패킷
Fig. 2. Compressed Packet secured with AH

3. 제안 프로토콜

앞서 살펴 보았듯이 기존 경량 IPsec 프로토콜은 터널링과 호환성의 문제점을 갖고 있다. 이러한 문제점을 보완하고 IPsec 프로토콜을 경량화를 하기 위해 본 논문에서는 새로운 방식의 경량 IPsec 프로토콜을 제안

한다.

3.1 Dispatch의 활용

6LoWPAN의 Dispatch는 많은 부분이 Reserved 용도로 사용되고 있다. 이러한 Reserved 부분을 활용한다면 기존 방식에서 제안한 확장헤더 내의 4bit 부분을 사용할 필요가 없게 된다.

AH와 ESP의 전송 모드와 터널 모드를 지원하기 위해 Dispatch의 부분은 4가지로 설정해야하며 AH 전송모드, AH 터널모드, ESP 전송모드, ESP 터널모드이다. 추가적으로 압축이 필요할 경우 더욱 많은 모드가 필요할 가능성이 있지만 Reserved 부분이 Dispatch에서 다수 사용되고 있기 때문에 문제가 없다.

3.2 확장헤더의 구성

IPsec의 터널 모드를 사용하기 위해서는 기존 IP 헤더의 유출을 막아주는 확장헤더가 필수적으로 필요하게 된다. 6LoWPAN에 터널모드를 적용하기 위한 확장헤더의 형식은 기존 인터넷 환경과 호환성 유지를 위해 IPsec의 형식을 따라야 한다. [그림 3]는 기존 IPsec에서의 확장헤더 구성방식을 나타낸다.[4,5,6]

	<-- How Outer Hdr Relates Inner Hdr -->	
	Outer Hdr at	Inner Hdr
IPv6	Encapsulator	Decapsulat
Header fields:	----------	----------
version	6 (1)	no change
DS Field	copied from inner hdr (5)	no change
ECN Field	copied from inner hdr	constructe
flow label	copied or configured (8)	no change
payload length	constructed	no change
next header	AH,ESP,routing hdr	no change
hop limit	constructed (2)	decrement
src address	constructed (3)	no change
dest address	constructed (3)	no change
Extension headers	never copied (7)	no change

그림 3. 기존 IPsec 확장헤더 구성
Fig. 3. IPsec Extension Header Encoding

6LoWPAN의 호율적인 압축을 위해서는 필요 없는 부분 또는 다른 계층에서 알 수 있는 부분은 삭제하였다.

(1) Version 6 : 6LoWPAN에서는 IPv6를 사용하므로 따로 표시할 필요가 없다. 대신 SG에서는 6LoWPAN을 사용하는 패킷이 올 경우 압축된 패킷이라는 것을 알 수 있어야 하며 이는 Dispatch 또는 Layer 2에서 확인이 가능하다.

(2) DS Field : 기존 IPsec은 IPv6 헤더의 DS 필드를 복사하여 사용한다. 제안 헤더 또한 기존의 방식과 동일하게 필드를 복사하며 사용방식은 6LoWPAN 방식을 따른다.

(3) ECN Field : 기존 IPsec은 IPv6 헤더의 ECN 필드를 복사하여 사용한다. 마찬가지로 기존의 방식과 동일하게 복사하여 사용하며 6LoWPAN의 방식을 따른다. 추가적으로 매핑이 필요하며 이는 SG나 Router에서 매핑테이블을 생성하여 매핑 정보를 알 수 있어야 한다.

(4) Payload Length : 확장헤더 크기를 나타내기 위해 사용되며 기존 6LoWPAN과 패킷의 순서가 달라지기 때문에 필수적으로 사용되어야 한다.

(5) Next Header : Dispatch 부분을 사용하여 AH와 ESP를 명시할 수 있게 된다. 그러므로 명시할 필요가 없어지게 되므로 삭제한다.

(6) Hop Limit : 기존의 IPsec 방식을 따르며 압축을 진행하지 않는다.

(7) Source Address 및 Destination Address : 기존 6LoWPAN은 64bit와 16bit를 사용한다. 호환성을 위해 64bit를 사용해야한다. 16bit를 사용할 경우 Network Layer 2의 주소를 이용하므로 기존 패킷에서 보내는 주소가 나타나게 된다. 터널모드에서 기존 패킷의 주소가 나타나게 되면 보안상 취약점이 발생된다.

이러한 사항에 따라 생성되는 확장헤더는 [그림 4]과 같다.

TF	Payload Len	HL	Source Add	Dest Add
2 bits	16bits	2 bits	64 bits	64 bits

그림 4. Tunnel Mode를 위한 제안 확장헤더
Fig. 4. Proposed Extension Header for Tunnel Mode

3.3 Payload의 압축

효율적인 패킷 경량화를 위해 [표 1]과 같이 IKEv2 Notification IPCOMP Transforms IDs(Value 16387)의 알고리즘을 이용하여 Payload 압축을 진행한다.

표 1. IKEv2 Notification IPCOMP Transforms IDs
Table 1. IKEv2 Notification IPCOMP Transforms IDs

Value	Compression Type	Reference
0	Reserved	RFC7296
1	IPCOMP_OUI	UNSPECIFIED
2	IPCOMP_DEFLATE	RFC2394
3	IPCOMP_LZS	RFC2395
4	IPCOMP_LZJH	RFC3051
5-240	Unssigned	-
241-255	Private Use	RFC7296

Value 0일 경우 Reserved, Value 1 일 경우 IPCOMP_OUI를 사용한다. 하지만 OUI 알고리즘은 구체적인 RFC가 존재하지 않아 실제로 적용을 하기 어렵다.

Value 2의 IPCOMP_DEFLATE와 Value3의 IPCOMP_LZS 그리고 Value 4의 IPCOMP_LZJH는 상당히 많이 사용되고 있는 알고리즘이다. 이러한 알고리즘들은 무 손실 압축 알고리즘으로서 데이터를 압축하여도 복구하였을 때 큰 문제가 없게 된다. Value 5 이후로는 Unssigned와 Private Use이기 때문에 고려할 필요가 없다. 최종적으로 Payload 압축을 위해 사용하는 알고리즘은 [표 2]와 같다.

표 2. Payload 압축을 위한 사용 알고리즘
Table 2. Payload Compress Algorithm in Proposed Protocol

Value	Compression Type	Reference
0	No Compression	-
1	IPCOMP_DEFLATE	RFC2394
2	IPCOMP_LZS	RFC2395
3	IPCOMP_LZJH	RFC3051

기존 IPv6나 압축 알고리즘을 사용하지 않는 패킷을 위해 Value 0에는 Not Compression을 설정한다. Value 1,2,3 번은 앞서 언급한 무 손실 알고리즘을 사용하여 압축을 진행하도록 한다.

3.4 제안 프로토콜의 패킷 구조

제안 프로토콜의 패킷에서 HC0는 Tunnel Mode 지원을 위한 확장헤더를 나타낸다. HC1는 6LoWPAN_IPHC, HC2는 6LoWPAN_NHC를 나타낸다. CA (Compression Algorithm) 필드는 [표 2]에 나타난 알고리즘을 사용하며 2bit로 나타낸다. SPI와 SN는 기존 경량 IPsec 방식에서 사용한 압축방식과 같은 방식을 사용하며 2bit로 나타낸다.

앞서 언급한 Extension Header의 처음 4bits는 Dispatch에서 AH와 ESP의 전송 모드 및 터널모드의 헤더의 타입을 나타낼 수 있기 때문에 삭제한다. Dispatch에 따라 뒤에 나오는 AH와 ESP의 Payload가 나오며, 이는 CA에서 설정한 압축 알고리즘에 따라 압축이 진행된다. 추가로 터널 모드가 가능하게 되면 UDP Checksum 부분은 삭제가 가능하게 되며 UDP 패킷에서의 압축이 가능하게 된다.

그림 5. 제안 프로토콜의 패킷 구조
Fig. 5. Packet Encoding in Proposed Protocol

4. 성능 분석

프로토콜 분석을 위해 6LoWPAN, 기존 경량 IPsec 프로토콜 그리고 제안 경량 IPsec을 연산량과 통신량 관점으로 성능 분석을 진행한다.

4.1 연산량

6LoWPAN의 경우 Preamble 생성, 802.15.4 Layer를 이용한 Mac Header 생성, Dispatch의 생성, IPv6를 압축하여 생성하는 6LoWPAN_IPHC의 생성, Next Header를 압축하여 생성하는 6LoWPAN_NHC의 생성이 있다. NHC안에 확장헤더와 관련된 파라미터인 EID가 포함되어 있다. 이후 UDP와 관련된 UDP Header 및 UDP Payload를 생성하며 마지막으로 FCS를 생성한다.

이를 파라미터에 접목하여 나타내면 Preamble 생성 P, 802.15.4 Mac Header 생성 M, 6LoWPAN_IPHC 생성 A, 6LoWPAN_NHC 생성 B, 그리고 UDP Header 및 UDP Payload 생성 C, 마지막으로 FCS생성 F로 나타내며 'P+M+A+B+C+F'의 패킷 생성 연산 요구량을 필요로 한다.

기존 경량 IPsec 프로토콜의 패킷 생성 연산 요구량을 살펴보면 기본적인 패킷생성인 Preamble 생성, 802.154 Layer를 이용한 Mac Header 생성, Disaptch의 생성, 6LoWPAN_IPHC의 생성은 동일하다. AH와 ESP를 나타내기 위한 EID 또한 기본 4bit를 이용하고 SPI 및 SN의 표시도 기본 6LoWPAN_NHC의 비트수와 같으므로 6LoWPAN에서의 NHC 생성 요구량과 같다고 할 수 있다. ESP모드를 진행할 경우 IV 생성 및 ESP의 Checksum 그리고 알고리즘 사용을 위한 Pad와 Pad의 길이를 나타내는 Pad Length의 생성을 요구하며 ICV 또한 무결성 검사를 위해 연산이 추가로 요구된다. 또한 암호화를 위한 암호화 알고리즘 사용 연산이 필요로 하게 된다. 이러한 생성 요구량을 파라미터에 접목하며 나타내면'P+M+A+B+C+F+I+S+L+Z'의 패킷 생성 연산이 요구된다.

표 3. 프로토콜의 연산량 비교
Table 3. Comparison of calculation amount

	6LoWPAN	Existed Lightweight IPsec	Proposed Lightweight IPsec
Calucul ation Amount	α	α+I+S+L+Z	α+I+S+L+Z+D+E

- α : 기본 6LoWPAN 패킷 생성 연산량
- I : IV 생성
- S : ESP Checksum 생성
- L : Pad 및 Pad length 생성
- Z : ESP 암호화 및 무결성 알고리즘 연산
- D : 터널모드를 위한 헤더 생성
- E : Payload 압축 연산

제안 경량 IPsec 프로토콜의 패킷 생성 연산 요구량와 기존 경량 IPsec 프로토콜의 연산 요구량의 큰 차이점은 터널모드를 위한 새로운 헤더 생성과 Payload의 압축을 위한 압축 알고리즘 연산이다. Compression Algorith의 헤더 필드는 기본적인 6LoWPAN_NHC에서 포함되고 IPsec을 위한 SPI와 SN 또한 포함되기 때문에 6LoWPAN_NHC의 생성 연산 요구량과 같다. 이러한 요구량을 파라미터로 표현하면 'P+M+A+B+C+F+I+S+L+Z+D+E'의 연산량이 요구된다.

[표 3]은 연산량에 대해 나타낸다. 기본 6LoWPAN의 패킷 생성 연산 요구량은 기존 경량 IPsec 프로토콜과 제안 경량 IPsec 프로토콜에 포함되며 효율적인 비교 분석을 위해 6LoWPAN의 패킷 생성 연산 요구량을 α로 표시하였다.

4.2 통신량

6LoWPAN의 통신량을 살펴보면 연산량과 마찬가지로 Preamble, 802.15.4 Mac Header, Dispatch, 6LoWPAN_IPHC, 6LoWPAN_NHC, UDP 헤더 및 UDP Payload, FCS의 통신량을 요구한다.

기존 경량 IPsec 프로토콜은 6LoWPAN의 통신량에서 IPsec 환경을 위한 IV 패킷, Pad 및 Pad Length 패킷, ICV의 패킷의 통신량이 추가된다. 기존 6LoWPAN의 패킷헤더 통신량을 log(A)로 나타내고 6LoWPAN의 페이로드 통신량을 log(B)로 나타낼 경우 추가로 요구되는 통신량인 IV 패킷 log(C), Pad 및 Pad Length 패킷의 통신량 log(D), 그리고 ICV 패킷의 통신량 log(E)로 log(A)+log(B)+log(C)+log(D)+log(E)=log(ABCDE)의 통신량이 요구된다.

표 4. 프로토콜의 통신량 비교
Table 4. Comparison of communication amount

	6LoWPAN	Existed Lightweight IPsec	Proposed Lightweight IPsec
Commu nication Amount	log(AB)	log(ABCDE)	$\log(AB_CCDE_CZ)$

- A : 6LoWPAN 패킷 헤더
- B : 6LoWPAN 페이로드
- C : IV
- D : Pad 및 Pad Length
- E : ICV
- BC : 압축 페이로드
- EC : 압축 ICV
- Z : 터널모드를 위한 추가 헤더

제안 경량 IPsec 프로토콜에서는 6LoWPAN 패킷 헤더의 통신량 log(A)와 압축알고리즘이 적용된 압축 페이로드 통신량 log(BC), IPsec을 지원하기 위한 IV 패킷 log(C), Pad 및 Pad Length 패킷의 통신량 log(D), 그리고 압축된 ICV 패킷의 통신량 log(EC) 그

리고 터널모드를 위한 추가헤더인 log(Z)로 log(A)+log(BC)+log(C)+log(D)+log(EC)+log(Z)=log(ABCCDECZ)의 통신량이 요구된다. [표 4]에서 각 방식에서의 통신량을 보여준다.

5. 성능 평가

5.1 성능 평가 환경

IoT 환경을 구성하기 위해 IoT Node와 IoT Gateway 그리고 IPv6 Device로 환경을 구성하였다. IoT Node와 IoT Gateway는 IEEE 802.15.4의 환경을 위해 무선 네트워크를 사용하며 IoT Gateway와 IPv6 Device는 기존 IPv6 네트워크를 사용한다. [그림 6]는 성능평가를 위한 환경을 나타낸다.

그림 6. 성능 평가 환경
Fig. 6. Performanc Evaluation Environment

IoT Node는 Raspberry Pi를 이용하여 구성하였으며 IoT Gateway와 IPv6의 Device는 일반적인 PC를 사용하여 구성하였다. 통신환경은 IoT Node와 IoT Gateway는 Bluetooth 4.0 위에 6LoWPAN 프로토콜을 이용하여 구성하였으며 거리는 약 3m를 두어 성능평가를 진행하였다. IoT Gateway와 IPv6 Device는 IPv6를 이용한 기존 네트워크 환경을 이용하여 구성하였다.

5.2 Payload 압축 후 응답시간

6LoWPAN은 기본적인 데이터를 보내며 기존 경량화 IPsec 또한 IPsec의 AH 전송모드를 이용하여 데이터를 전송하여 응답시간을 측정하였다. 제안 경량화 IPsec의 경우 기존 경량화 IPsec과 같은 AH 전송모드에서 데이터를 전송하지만 Payload 부분을 압축하여 데이터를 전송한 응답시간을 측정하였다.

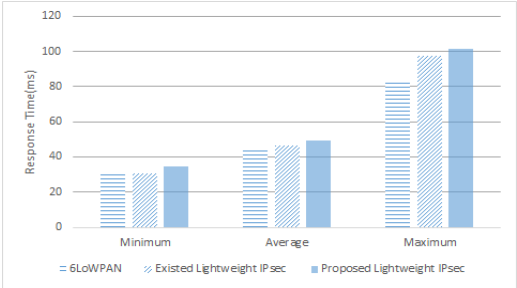

그림 7. Payload 압축 후 응답 시간
Fig. 7. Response Time after Payload Compress

평균 응답시간은 6LoWPAN은 약 45ms, 기존 경량화 IPsec은 약 46ms로 1ms가 차이가 나며 제안 경량화 IPsec은 약 49ms로 6LoWPAN보다 약 4ms가 차이나는 것을 알 수 있다. 기존 경량화 IPsec과 3ms의 응답시간이 차이가 나는데 Payload 압축 시간으로 인한 응답시간 차이인 것을 확인할 수 있다.

5.3 Payload 압축 해제 후 응답시간

6LoWPAN은 기본적인 데이터를 보내며 기존 경량화 IPsec 또한 AH 전송모드로 데이터를 전송한다. 제안 경량화 IPsec은 AH 전송모드로 Payload 압축을 진행하여 데이터를 전송하며 응답자는 압축 해제 후 응답시간을 측정하였다.

6LoWPAN 및 기존 경량화 IPsec의 응답시간은 [그림 8]과 동일하며 제안 경량화 IPsec의 경우 약 142ms로 응답시간이 증가하였다. 이는 압축 해제로 인한 응답시간 증가의 결과이며 평균적으로 93ms가 압축 해제에 사용되는 시간인 것을 확인 할 수 있다.

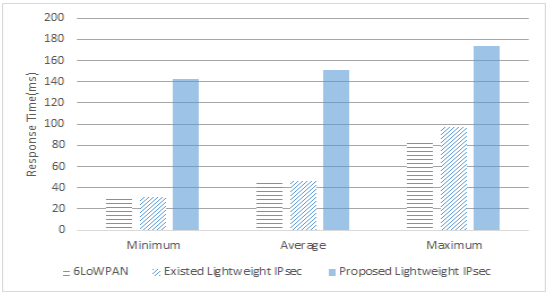

그림 8. Payload 압축 해제 후 응답 시간
Fig. 8 Response Time after Payload Uncompress

6. 결론

본 연구는 기존 경량화 연구들을 분석하여 단점을 도출하여 기존의 인터넷 환경과 호환성을 유지할 수 있도록 연구하였다. 제안 경량화 IPsec은 Dispatch의 이용 및 통신량 감소를 위해 압축 알고리즘을 이용하여 Payload의 압축을 적용하였으며 터널 모드의 추가를 위한 헤더 추가를 통해 IoT 환경에서의 안전할 수 있는 프로토콜을 제안하였다.

하지만 제안 경량 IPsec 프로토콜은 기존 경량 IPsec 프로토콜보다 압축 및 압축 해제로 인해 소요시간이 약 100ms가 추가되지만 압축되는 크기가 약 30-50byte로 6LoWPAN의 MTU가 127byte인 점을 고려하면 효율적인 압축이라 할 수 있다. 제안하는 경량 IPsec 프로토콜은 기존 경량화 프로토콜의 비해 압축 및 압축해제 시간이 소요되지만 패킷 통신량을 크게 절감할 수 있어 MTU의 사이즈가 작은 IoT 환경에 적용한다면 보안성이 향상된 IoT 통신 환경을 구축 할 수 있을 것으로 사료된다.

REFERENCES

[1] Shahid Raza, Thiemo Voigt, Utz Roedig, “6LoWPAN Extension for IPsec”, Swedish Institute of Computer Science, March. 2011.

[2] Shahid Raza, “Lightweight Security Solutions for the Internet of Things”, Department of Computer Science and Engineering Malardalen University, 2013.

[3] Shahid Raza, “Securing Communication in 6LoWPAN with Compressed IPsec“, Lancaster University School of Computing and Communications, 2011.

[4] S. Kent, “Security Architecture for the Internet Protocol”, RFC 4301, 2005.

[5] G. Montenegro, N. Kushalnagar, J. Hui, D. Culler, “Transmission of IPv6 Packets over IEEE 802.15.4 Networks”, RFC 4944, 2007.

[6] J. Hui, Ed, P. Thubert, “Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks”, RFC 6282, 2011.

저자약력

송 인 아(In-A Song) **[학생회원]**

- 2015년 2월 : 군산대학교 정보통신공학과(학사)
- 2017년 2월 : 군산대학교 컴퓨터정보통신공학부(석사)
- 2017년 3월 ~ 현재 : (주)다윈 연구원

<관심분야>
네트워크 보안, 네트워크 프로토콜

오 정 현(Jeong-Hyeon Oh) **[정회원]**

- 2005년 2월 : 목포해양대학교 정보통신공학과(학사)
- 2017년 2월 : 군산대학교 컴퓨터정보통신공학부(석사)
- 2017년 3월 ~ 현재 : 군산대학교 컴퓨터정보통신공학부(박사과정)
- 2013년 11월 ~ 현재 : (주)드림시큐리티 연구원

<관심분야>
네트워크 보안, 운영체제 보안

이 두 원(Hong-Sung Kim) **[정회원]**

- 1991년 2월 : 숭실대학교 전자계산학과(학사)
- 2005년 8월 : 연세대학교 경영대학원(석사)
- 2015년 8월 : 군산대학교 컴퓨터정보통신공학부(박사수료)
- 2016년 5월 ~ 현재 : ㈜아니스트 대표

<관심분야>
블록체인, 사물인터넷, 디지털 포렌식

이 영 석(Young-Seok Lee) **[종신회원]**

- 1992년 2월 : 충남대학교 컴퓨터공학과(학사)
- 1994년 2월 : 충남대학교 컴퓨터공학과(석사)
- 2002년 2월 : 충남대학교 컴퓨터공학과(박사)
- 2002년 3월 ~ 2004년 8월 : 한국전자통신연구원 선임연구원
- 2004년 9월 ~ 현재 : 군산대학교 컴퓨터정보통신공학부 교수

<관심분야>
정보보호, 사물인터넷, 이동컴퓨팅