이 일 구 교수 지도
석사학위 청구논문

저지연 암호화 통신 위한 네트워크 자가 최적화 기법

2023

성신여자대학교 대학원
미래융합기술공학과
심 혜 연

저지연 암호화 통신 위한 네트워크 자가 최적화 기법

이 일 구 교수 지도

이 논문을 석사학위논문으로 제출함

2022년 11월

성신여자대학교 대학원

미래융합기술공학과

심 혜 연

인 준 서

심혜연의 석사학위 논문으로 인준함

2022년 11월

심사위원장 김 성 민 (서명 또는 인)

심 사 위 원 이 일 구 (서명 또는 인)

심 사 위 원 김 경 진 (서명 또는 인)

성신여자대학교 대학원

논 문 개 요

통신 기술의 발전에 따라 다양한 기기가 네트워크에 연결되면서 멀티 홉 환경의 네트워크가 확산되었다. 사물인터넷(IoT)과 개인 기기가 대중화되면서 실시간으로 발생되는 데이터를 처리하여 활용하고자 하고 있다. 이 외에도 디지털 기기에 따른 데이터 생성량이 기하급수적으로 증가하였으며 개인의 민감한 데이터가 누적되면서 데이터의 보안성과 기밀성이 중요해지고 있다. 그러나 종래의 연구는 통신 환경에서 암호화 알고리즘을 적용하여 네트워크의 변화하는 환경을 반영하지 못했으며, 보안성과 저지연 전송을 동시에 충족할 수 없었다는 문제가 있다.

본 연구에서는 멀티 홉의 수, 전송 데이터 크기, DPI 여부 등으로 변경되는 네트워크 통신 환경을 고려하여 동형암호화 방식과 AES 암호 방식을 선택적으로 사용하는 Dynamic Cryptographic Selection을 제안하고자 한다. 제안된 아이디어를 통해 네트워크 통신 환경에 따라 동적으로 통신을 최적화하는 암호화 메커니즘을 제안했으며, 암호화 통신의 보안성을 높이면서도 저지연 전송을 수행할 수 있다. 시뮬레이션을 통해 제안된 아이디어가 네트워크 환경에 따라 암호화 방식을 다르게 적용함으로써, 변화하는 네트워크 환경을 반영하여 AES 및 TenSEAL를 단일로 사용하였을 때의 통신보다 성능이 유지되는 것을 보였다.

목 차

표 차 례

그림 차례

제 I 장 서론

1. 배경

클라우드 및 네트워크 기술이 발전함에 따라 사물인터넷(IoT)이나 디지털 기기가 대중화되면서 실시간으로 데이터를 처리하는 멀티 홉 네트워크 환경이 증가하고 있다. 산업 사물인터넷(IIoT) 에서 교환된 정보를 대량으로 수집하게 되면서 데이터의 볼륨, 다양성 및 복잡성이 증가했다[1]. 산업 사물인터넷 장치가 관리해야 하는 데이터의 양이 일반적인 사물인터넷 애플리케이션보다 훨씬 더 많다는 것을 의미한다. 다분화된 데이터 특성과 볼륨을 반영하여 산업 사물인터넷 환경의 통신에서는 저지연 전송이 보장되어야 한다. 이 외에도 스마트 시티의 센서 내장형 장치에서는 효율적인 방식으로 데이터를 저장 및 처리해야 한다[2]. 데이터 처리 및 통신에 있어 지연시간을 줄여야 하며 암호화를 적용할 때, 네트워크 통신 환경에 따라 성능이 다르게 나올 수 있는 문제를 고려해야 한다.

네트워크 환경의 발전에 따라 사물인터넷 서비스, WSN(무선 센서 네트워크) 서비스, 클라우드 서비스, WSMN(무선 센서 모니터링 네트워크) 서비스 등을 제공하는 네트워크 시스템은 관리 가능성과 책임, 데이터 무결성, 가용성 및 기밀성을 보장해야 하며[4], 산업체의 기밀을 보호하기 위해 내부망 데이터에 대한 무결성 및 기밀성을 확인해야 할 필요가 있다. 디지털 기기에 따른 데이터 생성량의 증가로 인해 민감한 데이터가 누적되면서 데이터 레이크 서비스 제공업체가 보안 공격의 대상이 되었고[3], 멀티 홉 형태의 네트워크에서 처리되는 데이터를 탈취 및 유출하려는 위협들이 증가하고 있다. 통신 환경에서의 보안이 중요하게 되면서 AES(Advanced Encryption

Standard) 및 3DES(Triple Data Encryption Standard)와 같은 암호 알고리즘을 통신에 적용해 보안성을 높이기 위한 연구가 진행되고 있다.

내부망 데이터의 검증을 위해 사용되는 DPI(심층 패킷 검사)는 패킷 헤더뿐만 아니라 페이로드까지 검사하는 기법이다. DPI 기반으로 동작하는 솔루션은 네트워크 사용량에 대한 전체적인 흐름을 제공한다. 많은 양의 트래픽을 소비하는 사용자를 식별하고 트래픽을 실시간으로 관리하여 서비스 환경을 최적화하며 서비스 품질을 개선 및 관리한다[5]. 일반적으로 DPI 시스템은 네트워크에서 가장 로드가 많은 부분과 서비스 관리가 필요한 지점에 설치되어 통과되는 패킷을 검사한다. DPI는 특정 세션이 속한 애플리케이션 유형을 결정하고 관리자가 정의한 규칙을 적용하여[4], 악성코드 유무를 분석한다. DPI를 이용하면 지정된 다른 시스템으로의 트래픽을 차단하거나 제한할 수 있으며 우선순위를 지정할 수 있다[4]. DPI 기반 솔루션은 기업의 내부망과 같은 기밀성이 중요시되는 환경에서 유용하기 때문에, 기업에서는 보안성 강화를 위해 DPI 기반 솔루션을 도입하고자 하고 있다. 예를 들어, 소켓 통신 환경과 같이 서버가 제어하는 통신 환경에서는 기업의 내부망에서 패킷을 복호화하여 DPI를 진행할 수 있다. 그러나 이 방법은 오버헤드 발생으로 통신 지연시간이 증가할 수 있다.

현재 대부분의 네트워크 통신에서 보안을 위해 암호화를 적용하고 있다. AES와 같은 기존의 암호화 방식의 경우 빠른 통신이 가능하다. 멀티 홉 네트워크 내 DPI가 진행될 때, 악성 페이로드를 탐지하지 못하여 보안성이 저하될 수 있다. DPI 검사를 진행하더라도 복호화 후 검사를 진행하기 때문에 오버헤드 및 통신 지연시간이 증가할 수 있다. 반대로 동형암호의 경우 DPI가 존재하더라도 중간에 복호화를 하는 과정을 거치지 않아 보안성이 높다. 하지만 AES 방식에 비해 상대적으로 암호화 및 복호화 속도가 느리고, 오버헤드 및 통신 지연시간이 높아 실제 통신 환경에 적용되기 어려운 문제가

있다. 종래의 네트워크 연구에서는 지연 및 전송 비율 측면에서 네트워크의 성능과 개발을 향상하기 위해 다양한 라우팅 알고리즘이 개발되었지만, 보안성을 높이면서 성능을 유지하는 연구는 부족했다. 네트워크 대부분에서는 단일 암호화의 사용으로 인해 멀티 홉 네트워크 상황을 반영하기 어려워 통신의 지연시간을 높이고 있다.

본 논문에서는 홉의 수, 전송 데이터 크기, DPI 여부 등 멀티 홉 네트워크 통신 환경을 고려하여 동형암호화 방식과 AES 암호화를 선택적으로 사용하는 Dynamic Cryptographic Selection을 제안하고자 한다. 제안된 Dynamic Cryptographic Selection의 모델은 암호 방식을 적응형으로 선택하여 사용함으로써 멀티 홉 네트워크 환경에서의 통신 효율성을 증가시킬 수 있음을 보인다.

2. 논문의 기여점

종래 연구와 비교할 때 본 논문의 기여점은 다음과 같다.

첫째, 네트워크 통신 환경에 따라 동적으로 통신을 최적화하는 Dynamic Cryptographic Selection을 제안함으로써 저지연 전송에 기여한다.

둘째, 암호화 통신 환경에서 동형암호와 AES 암호화를 동적으로 선택하여 보안성을 유지하는 모델을 제시한다.

셋째, 다양한 네트워크 통신 환경에서 암호화된 데이터의 전송을 수행하는 간단한 시뮬레이션을 통해 제안한 모델의 성능을 입증한다.

본 연구에서는 홉의 수, DPI 등 네트워크 통신 환경을 결정하는 요소를 고려하여 동형암호 및 AES 암호화를 선택적으로 사용한다. AES 암호화는 속도는 빠르지만 많은 DPI가 존재하는 상황에는 불리하다. 동형암호화는 통

신 속도는 느리지만 DPI가 자주 일어나는 상황에서 암호문에 연산이 가능하여 AES 암호화에 비해 좋은 성능을 보인다. 이에 따라 다양한 네트워크 통신 환경에 최적의 성능을 갖는 암호화를 선택할 수 있다. 네트워크 통신 환경이 변화하더라도 단일 암호화를 사용하는 경우보다 다양한 통신 상황에서도 일관적으로 낮은 전송 지연시간을 가지는 통신을 진행할 수 있다.

기존의 통신에는 빠른 통신 속도로 인해 AES 암호화를 사용하였지만, 중간에 DPI가 존재하면 DPI가 제대로 수행되지 못하는 문제가 존재한다. DPI를 수행되지 않으면 전송되는 악성 페이로드가 존재하더라도 탐지하지 못하고, DPI를 수행하기 위해 데이터를 복호화하면 오버헤드가 발생한다. 이는 보안성 저하 및 속도 저하 문제를 야기한다. 본 연구에서는 암호화 통신에 AES 암호화와 암호화된 데이터에 DPI를 수행할 수 있는 동형암호화를 선택적으로 사용함으로써 DPI가 존재하더라도 보안성을 유지할 수 있다.

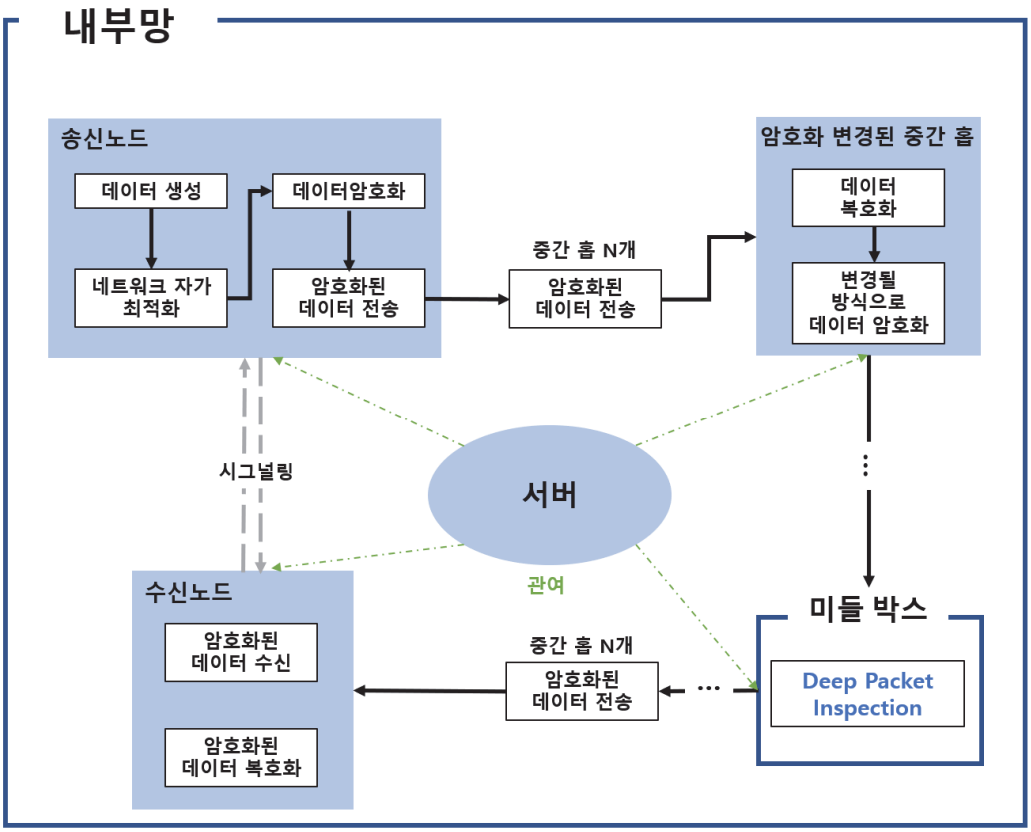
중간 홉 개수, DPI를 수행하는 중간 홉 비율, DPI 연산 횟수, 전송 데이터 크기 등 다양한 통신 매개변수를 변화시키고, 데이터셋을 구축하여 본 연구의 모델에 대한 시뮬레이션을 진행하였다. 이를 통해 멀티 홉 네트워크 통신 환경에서 제안한 알고리즘의 실효성을 입증했으며, 보다 신뢰성 높은 알고리즘의 성능 결과를 도출했다.

3. 논문 구성

본 논문의 구조는 다음과 같이 구성된다. 2장에서는 본 논문에서 주요하게 사용되는 배경 지식 및 시스템 개요에 대해 설명한다. 3장에서는 선행연구에 관해 기술하며, 4장에서는 본 논문에서 제시하는 아이디어와 실험 환경 및 평가 방법에 관해 기술한 후, 실험 결과를 분석한다. 5장에서는 결론을 내린다.

제 Ⅱ 장 PROBLEM SCOPE STATEMENT

본 장에서는 논문에서 주요하게 다루는 동형암호, 저지연 전송 기법, 네트워크 자가 최적화, DPI의 개념에 대해 정의하고, 암호화 방식에 대해 간단히 서술하고자 한다.



[그림 1] 시스템 개요

[그림 1]은 본 논문에서 고려하고 있는 시스템에서의 암호화 통신이 이루어지는 과정을 나타낸다. 전체 시스템은 서버가 관여하는 통신을 기반으로

하고, 기업이나 조직의 내부망이라고 가정한다. 시스템에 악성 페이로드가 담긴 패킷의 침입으로 인해 기업 기밀과 같은 중요한 사항이 유출되거나 시스템의 변경 등의 보안 문제가 발생할 수 있다. 기업은 악성코드 공격으로부터 시스템을 보호하기 위해 DPI를 수행한다. DPI는 컴퓨터 네트워크를 통해 전송되는 데이터를 패킷 헤더뿐만 아니라 페이로드까지 자세히 검사하여 패킷 차단, 경로 재지정 또는 로깅과 같은 대응할 수 있는 솔루션이다. DPI는 OSI 7 Layer 계층에서 2계층부터 7계층까지 동작할 수 있으며, 네트워크에서 가장 로드가 많은 부분과 서비스 관리가 필요한 지점에 설치되어 전송 데이터의 전체를 검사한다[4]. 유해 정보 및 외부 공격 위협을 차단함으로써 보안성을 높이기 위해 사용되고 있다. 이러한 DPI는 미들 박스에서 동작할 수 있다. 미들 박스는 다른 중간 홉과는 다르게 패킷을 전달뿐만 아니라 패킷 검사, 필터링 등을 수행하는 컴퓨터 네트워킹 장치이다. 서버는 DPI 결과를 확인하고 대응해야 하기 때문에 DPI를 수행하는 미들 박스에 관여한다.

송신 노드는 수신 노드로 데이터를 전송하려고 한다. 전송되는 데이터는 중요한 정보를 포함할 수 있으므로 수신 대상 및 보안을 담당하는 팀 외에는 확인하지 못하도록 암호화가 적용된다. 송신 노드가 전송을 수행하면 송신 노드와 수신 노드는 시그널링을 통해 통신 관련 정보를 획득한다. 이때 내부망의 서버가 관련하여 암호화 키에 대한 지정을 할 수 있다. 서버는 암호화키를 송신 노드 등의 암호화키가 필요한 홉에 분배 및 교환할 때 [6]과 같이 BIBD(Balanced Incomplete Block Design) 접근 방식이나 [7]과 같이 타원곡선 암호화를 통한 키 분배 방식의 이용을 통해 안전하고 빠르게 키 분배 및 관리를 진행할 수 있다. 송신 노드는 전송할 데이터를 빠른 속도로 보내기 위해 네트워크 상황에 최적화된 통신을 진행할 수 있는 각 통신 구간의 암호화를 선정한다. 송신 노드는 선정된 암호화를 중간 홉에 짧은 패

킷으로 전송하여 본 통신 이전에 암호화를 설정한다. 통신을 진행하면 설정된 암호화를 통해 데이터를 수신 노드로 전송한다. 중간 홉은 전송을 수행할 통신 구간에 설정된 데이터 암호화가 이전 통신 구간과 동일하다면 데이터 전송만 진행한다. 이전 통신 구간에서 사용한 암호화와 다른 암호화로 통신 구간이 설정되어있는 경우에는 암호화 알고리즘을 변경하게 되고 보안상의 문제로 서버가 관여하여 해당 과정이 수행된다.

본 환경에는 암호화 알고리즘 선택에 동형암호화와 기존 통신의 암호화가 이용된다.

동형암호화는 사용자가 먼저 암호를 해독하지 않고 암호화된 데이터에 대해 연산을 수행할 수 있도록 하는 암호화의 한 형태이다. 기존 암호화 방식과 달리, 암호문을 복호화하지 않아도 검색과 통계 처리 및 기계 학습에 활용할 수 있으며 데이터를 처리하는 중간 과정에서 복호화하지 않아도 되므로 데이터 유출 위험을 줄일 수 있다는 장점이 있다[8]. 본 논문에서는 동형암호화 패키지로 TenSEAL을 사용했다. TenSEAL은 Microsoft SEAL을 기반으로 구축된 동형암호화 라이브러리다. Python API를 통해 사용 편의성을 제공하는 동시에 C++를 사용하여 구현되어있으며 효율성을 유지하도록 구성되어 있다. 주요한 특징으로는 BFV(Brakerski, Fan, and Vercauteren)를 사용한 정수 벡터의 암·복호화, CKKS(Cheon, Kim, Kim and Song)를 이용한 실수 벡터의 암·복호화 등이 있다[9].

본 환경에서는 기존 통신환경에서 사용한 암호화 알고리즘 중 AES 암호화를 사용하였다. AES 암호화는 대칭 블록 암호로 NIST(National Institute of Standards and Technology)에서 무차별 대입 공격에 취약해지기 시작한 데이터 암호화 표준(DES)에 대한 대안이 필요하게 되어 만들어진 암호화 방식이다. AES 암호화의 키 길이는 128bit, 192bit, 256bit로 구성되어 있으며 128bit 블록의 데이터를 암호화하고 해독하도록 구성되어 있다.

시스템에서 중점적으로 최적화하는 성능지표인 전송 지연시간은 하나의 데이터 패킷을 한 지점에서 다른 지점으로 전송할 때 걸리는 시간을 의미한다. 지연시간이 적을수록 패킷 손실이 적어지므로, 저지연 전송 환경을 유지하는 것이 통신에서 중요하다. 5G와 같은 차세대 통신 시스템에서는 낮은 전송 지연시간을 유지하여 낮은 대기 시간을 가능하게 하도록 무선 액세스의 네트워크 엣지에 위치하기도 한다[10].

최근에는 낮은 전송 지연시간을 유지하기 위해 네트워크 자가 최적화(SON) 시스템을 도입하고 있다. 네트워크 자가 최적화는 모바일 무선 접속 네트워크의 계획, 구성, 관리, 최적화, 문제 수정을 더 간단하고 더 빠르게 만들기 위해 설계된 자동화 기술이다. SON 기술은 서로 다른 네트워크를 사용할 때 끊김이 없는 통신을 제공하도록 보장하기 위해 자체 최적화를 하는 것을 의미한다[11].

제 Ⅲ 장 관련 연구

본 장에서는 네트워크 통신 환경에서 암호 알고리즘을 적용하거나, 암호화 통신 환경에서 경량화 및 성능 개선, DPI의 초점으로 진행된 연구를 살펴보고자 한다.

네트워크 환경에서 동형암호를 사용한 선행연구는 다음과 같다.

동형암호화에 대한 이론과 동형암호화를 사용한 선행연구를 종합적으로 분석한 연구가 있다[12]. 해당 연구는 동형암호화를 사용한 연구를 정리하여 종합적으로 분석함으로써, 동형암호화를 활용한 연구 발전 방향에 대한 전체적인 개괄을 제시했다는 기여점이 있다.

이러한 종래 연구 외에도 무선 네트워크 환경에서 암호화를 직접적으로 활용하여 실험을 진행한 연구가 있다. 멀티 홉 네트워크 환경에서의 사물인터넷에 연결된 장치들은 높은 수준의 보안 요구 사항을 수용할 수 있어야 하며, 적은 양의 에너지 제약 문제를 처리하는 방식이 필요하다. 이에 따라, 사물인터넷에서 보안 공격으로부터 안전한 통신을 구현하기 위해 암호화를 적용하여 경량화를 진행한 연구가 있다[13, 14]. 해당 연구에서는 종단간(End-to-End)에서 통신을 보호하는 방안으로 암호화를 적용하고, 사물인터넷의 부담을 줄이기 위한 경량화를 진행한다. 단순 암호화를 적용하는 방식과 달리, 기계 학습 모델을 암호화 방식에 결합한 연구도 있다[15]. 해당 연구에서 사물인터넷 환경에서 인공지능 기반의 적응형 암호 시스템을 사용하여 새로운 경량 사물인터넷 장치 인증, 암호화 및 키 배포 접근 방식을 제안했다.

이 외에도 최근 인더스트리 4.0을 통해 산업체에서 활용되는 산업 사물인터넷의 보안 위협을 방지하기 위해 암호화를 사용한 연구도 있었다[16].

[표 1] 선행연구

항목	연구 주제	참고 문헌	기여점	한계점
암호 알고리즘 적용	통신 환경에서의 암호 알고리즘	[12]	완전동형암호화 체계의 보안에 대한 보완점 및 개선 방향에 대해 논의	개선 방향 논의로 그침 실제 시뮬레이션 부족
네트워크 통신	암호 경량화	[2] [13] [14] [17]	경량화를 통해 사물인터넷 통신 환경에서 효율적인 에너지 사용 조절, 저지연 네트워크 환경을 달성	멀티 홉의 수, DPI, 전송 데이터의 길이 등과 같은 요인으로 변화하는 네트워크 상황을 고려하지 못함 보안성과 저지연 전송을 동시에 충족할 수 없음
	성능 개선	[18] [19]	4K 해상도 또는 산업용 엣지 컴퓨터 환경에서의 최적화 진행을 위한 네트워크 자가 최적화 방식 제시	
		[20] [15] [16] [21]	공격자에게 노출되는 위협을 극복하기 위해 통신 보안성을 강화함	
네트워크 DPI	새로운 환경	[22]	통신 환경에서의 DPI 적용을 위해 새로운 DPI 접근 방식 메커니즘을 제안함	DPI의 보안성에 초점을 맞추어 통신 지연시간을 고려하지 못함
	성능 개선	[23]		

종래 연구의 주요 목적은 산업 사물인터넷 네트워크에서 보안성은 높이고 경량화된 암호화 알고리즘을 적용하기 위해 암호화 알고리즘을 경량화하거나 새로운 보안 프레임워크를 제안하고자 했다.

이처럼 무선 네트워크 환경에서 암호화 적용을 위한 연구는 지속되어 왔다. WSN환경 외에도 LTE(Long Term Evolution) 네트워크에서 사용자의 트래픽 암호화를 수행하기 위해 패킷 키 메커니즘을 사용하는 종단 간 보안 기법을 적용한 연구가 있다[21].

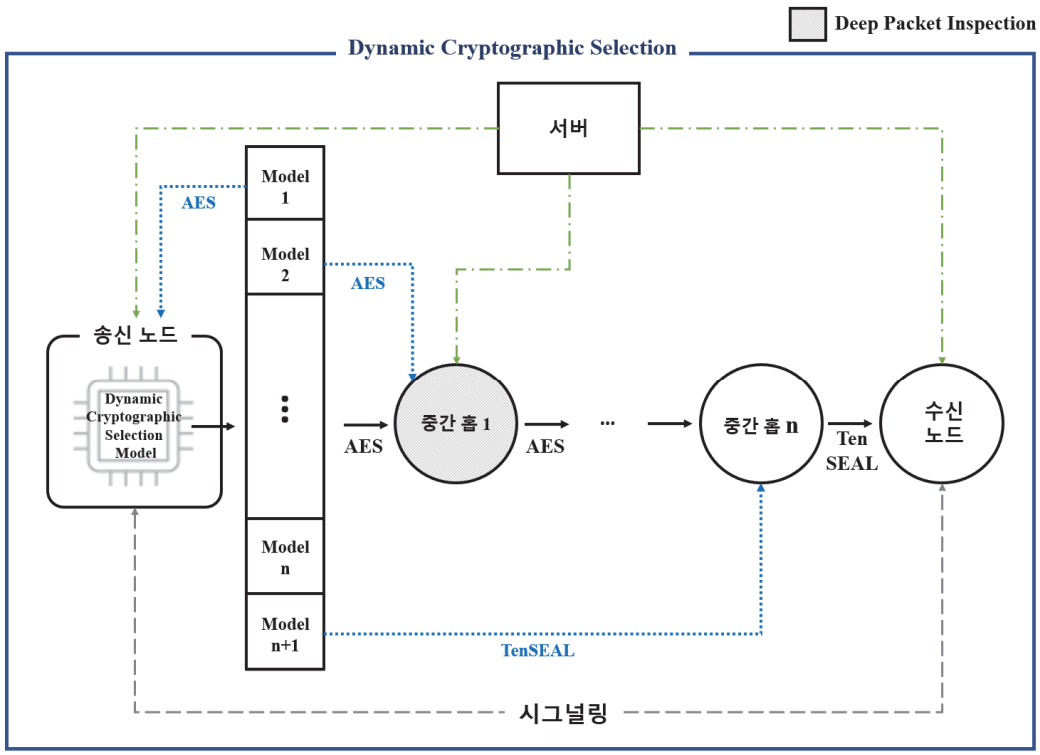
하지만 종래의 연구는 멀티 홉의 수, DPI 등과 같은 요인으로 인해 변화하는 네트워크 상황에 따라 암호화 알고리즘을 차등적으로 적용할 수 없어 보안성과 통신에 따른 전송 지연시간을 유지할 수 없었다.

따라서 본 연구에서는 네트워크 환경의 변화에 따라 동적으로 알고리즘을 사용할 수 있도록 하는 Dynamic Cryptographic Selection을 제안하고자 한다. 본 연구에서 제시한 알고리즘은 기존 단일 암호화 사용방식과 비교했을 때, 통신 환경에 최적화되어 전송 지연시간이 감소되는 것을 확인할 수 있다.

제 Ⅳ 장 DYNAMIC CRYPTOGRAPHIC SELECTION

본 장에서는 본 논문에서 제안하는 Dynamic Cryptographic Selection과 암호화 방식을 선정하는 Dynamic Cryptographic Selection 모델의 구성 및 흐름에 관해 서술하고자 한다.

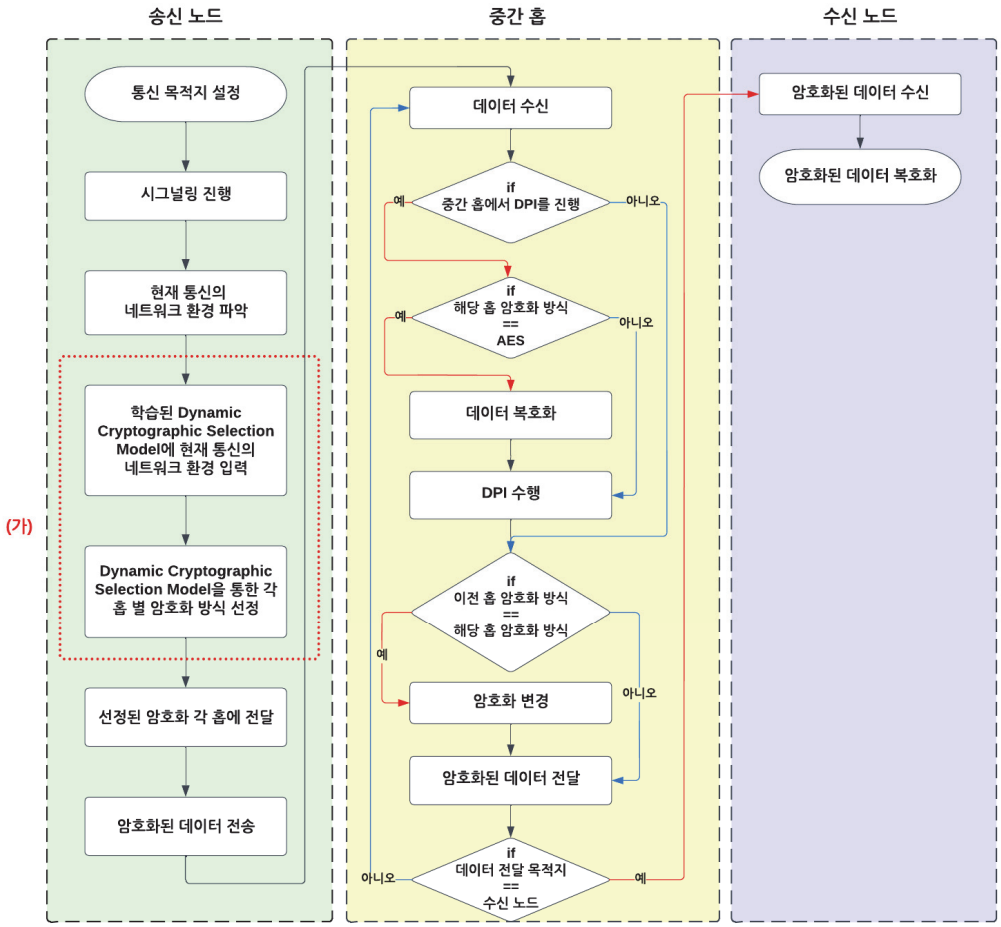
1. Dynamic Cryptographic Selection 시스템 구성



[그림 2] Dynamic Cryptographic Selection 시스템 구성도

[그림 2]는 Dynamic Cryptographic Selection이 기업의 내부망과 같은 통신을 진행하는 시스템에 적용되었을 때의 전체적인 동작을 나타낸다. 시스템에서는 수많은 송신 및 수신 동작이 이루어진다. 이때, 보안을 위해 데이터를 암호화하여 전송한다. 암호화 전송에는 Dynamic Cryptographic Selection 모델을 통해 선정된 암호화 방식이 사용된다. Dynamic Cryptographic Selection 모델은 머신러닝을 통해 생성되는 모델로, 네트워크 통신 진행 시 전송 지연시간을 줄이기 위해 송신 노드 단에서 암호화 선택을 수행한다. 선택 결과로 나오는 암호화는 AES 암호화와 TenSEAL 두 가지이다. Dynamic Cryptographic Selection 모델에서 암호화 선정은 수신 노드를 제외한 모든 홉 및 노드에 대해 각각 따로 진행된다. 선정된 암호화 방식은 데이터를 전달하는 중간 홉에 작은 패킷으로 보내진다. 중간 홉은 지정된 암호화 방식으로 데이터가 암호화되도록 설정하여 처리된 데이터를 전달한다. 각 홉이 전송을 수행할 구간에 선택된 암호화 방식을 통해 암호화된 데이터가 전송된다. 시스템의 전체 과정에는 서버가 개입하여 암호화 및 복호화가 존재하는 노드에 개입하여 암호화 키 등의 설정을 제공하거나 시스템의 네트워크의 상황을 파악하여 송신 및 수신 노드에 통신 환경에 대한 정보를 줄 수 있다.

Dynamic Cryptographic Selection의 시스템에서의 통신은 [그림 3]과 같이 송신 노드, 중간 홉, 수신 노드 순서로 진행된다. 송신 노드에서 데이터 전송을 하기 전에 통신 목적지를 설정한다. 이때, 설정되는 목적지는 수신 노드이다. 통신 설정이 완료되면 송신 노드는 수신 노드와의 시그널링을 통해 현재 통신 환경에 대한 정보를 획득한다. 이때, 서버를 통해 추가적인 통신 환경에 대한 정보를 획득하는 것이 가능하다. 작은 크기로 전송된 시그널을 통해 네트워크 통신 환경에 대한 정보를 구성하고 이를 사전에 학습된 Dynamic Cryptographic Selection 모델에 입력으로 전달한다. Dynamic



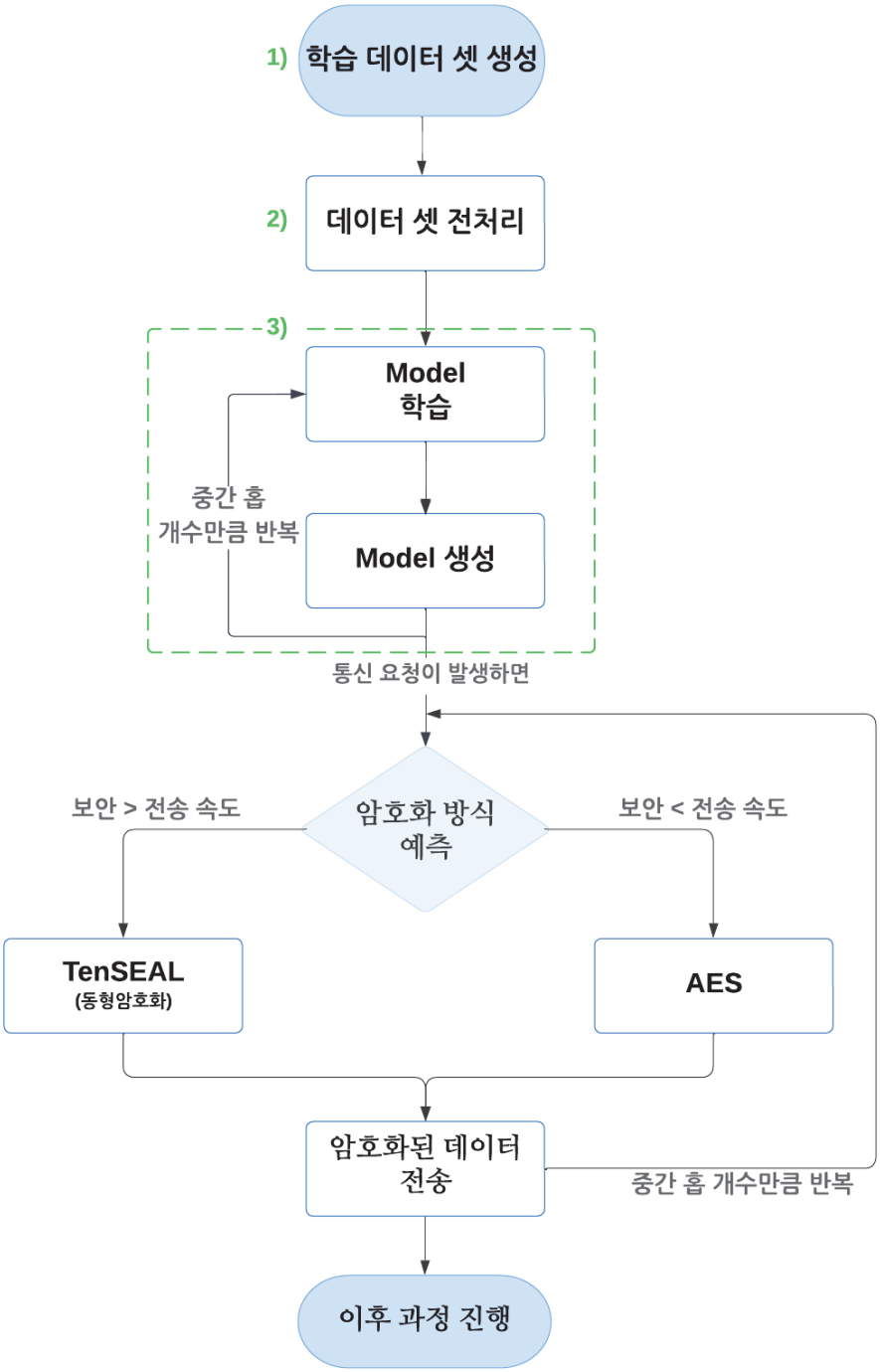
[그림 3] Dynamic Cryptographic Selection 통신 흐름도

Cryptographic Selection 모델은 현재의 통신 환경을 입력으로 받고 이를 토대로 현재 네트워크 통신 환경에서 낮은 지연시간을 가질 수 있는 암호화 방식을 통신을 선정한다. 암호화 방식 선정은 암호화를 진행하지 않는 수신 노드를 제외한 모든 노드 및 홉에 대해 이루어진다. 송신 노드는 Dynamic Cryptographic Selection 모델을 통해 선정된 각 홉에 대한 암호화 방식을 작은 사이즈의 패킷으로 중간 홉에 전송한다. 각 중간 홉은 전송된 작은 패킷을 통해 해당 홉에서 진행할 암호화 방식을 설정한다. 송신 노드는 선정

된 암호화 방식 중 송신 노드로 설정된 암호화 방식으로 데이터를 암호화하여 중간 홉으로 전송한다.

중간 홉은 송신 노드로부터 암호화되어 전송된 데이터를 수신하고 다시 데이터를 전송할 준비를 한다. 중간 홉이 DPI를 진행하는 홉이면 전송된 데이터에 대한 DPI 연산을 수행한다. 데이터가 AES로 암호화가 되어있으면 암호문에 바로 연산을 수행할 수 없으므로 암호문을 복호화한 후 DPI를 진행한다. TenSEAL은 암호문에 연산을 하는 것이 가능하므로 암호화된 데이터에 DPI를 진행한다. DPI 진행이 완료된 상태이거나 DPI가 존재하지 않는 홉이라면 이전 홉의 암호화 방식과 현재 홉에 설정된 암호화 방식을 비교한다. 두 가지 상황에서의 암호화 방식이 동일하면 데이터를 다음 홉 또는 노드로 전달한다. 이전 홉과 현재 홉에 설정된 암호화 방식이 서로 다르다면 전송된 데이터를 복호화한 후 현재 홉에 설정된 암호화를 통해 데이터를 재암호화한다. 과정이 완료되면 데이터를 전달한다. 중간 홉에서 데이터 전달은 수신 노드에 도착할 때까지 반복된다.

수신 노드는 중간 홉에서 전송된 암호화된 데이터를 획득한 후 해당 데이터를 복호화한다. 전송된 데이터가 정상적이라면 통신을 마친다.



[그림 4] Dynamic Cryptographic Selection 모델 흐름도

2. Dynamic Cryptographic Selection 모델 흐름

[그림 3]-(가)에서 네트워크 환경에 따라 최적의 암호화를 선정하는 Dynamic Cryptographic Selection 모델은 [그림 4]와 같은 흐름을 통해 생성된다. 암호화 알고리즘을 홉마다 선정하기 위해서는 머신러닝 학습을 통해 모델을 생성해야 한다. 모델은 데이터셋을 학습함으로써 생성되기 때문에 암호화 설정을 무작위로 하여 학습 데이터셋을 구성한다. 정확한 학습을 위해 학습 데이터셋 전처리를 진행한다. 전처리가 완료된 데이터셋을 통해 학습을 진행하고, 각 홉의 암호화를 선정하는 모델을 생성한다. 최종적으로 생성된 모델을 통해 현재 통신 환경에 최적화된 암호화를 각 홉마다 선정하고, 선정된 암호화를 통해 데이터를 전송을 수행한다.

Dynamic cryptographic algorithm은 예측을 위해 다음과 같은 과정을 거친다.

1) 학습 데이터셋 생성

예측은 학습된 머신러닝 모델을 통해 이루어지고, 머신러닝 모델의 학습을 위해 학습 데이터셋이 필요하다. 제안하는 Dynamic cryptographic algorithm의 머신러닝 모델은 네트워크 상황에 따라 유동적으로 가장 전송 지연시간이 낮을 수 있는 암호화를 예측하기 때문에 학습에 이용되는 데이터셋은 다양한 통신 상황 및 설정에 대한 값을 포함해야 한다.

그러나 머신러닝 모델 학습 이전에는 예측값을 통한 통신 진행이 불가하여 모델을 통한 통신 데이터셋을 생성할 수 없다. 따라서 초기 머신러닝 학습 데이터셋은 각 홉에 설정되는 암호화를 무작위로 지정한 통신 상황에서의 통신으로 생성한다. 머신러닝 학습 이후 학습된 모델로 생성한 각 구간의 암호화를 통신에 적용하여 학습에 사용할 데이터셋을 추가할 수 있다.

중간 홉	DPI 연산 횟수	DPI를 수행하는 중간 홉 비율	전송 데이터 크기	전송 지연시간	상위 20% 전송 지연시간
90	900	100	3500	8.520009	1
10	900	60	3500	0.640898	1
90	500	100	500	1.581494	1
90	500	100	500	2.721664	0
10	100	0	500	0.166203	0

1번 홉의 DPI 여부	2번 홉의 DPI 여부	...	n-1번 홉의 DPI 여부	n번 홉의 DPI 여부
1	1		1	1
0	1		0	1
1	1		1	1
1	1		1	1

1번 구간의 암호화	2번 구간의 암호화	...	n번 구간의 암호화	n+1번 구간의 암호화
TenSEAL	TenSEAL		TenSEAL	TenSEAL
TenSEAL	TenSEAL		TenSEAL	TenSEAL
AES	AES		AES	AES
TenSEAL	TenSEAL		AES	AES
AES	AES		AES	TenSEAL

[그림 5] Dynamic Cryptographic Selection 모델 학습에 사용되는 통신 정보 데이터셋의 예시

학습에 이용하는 데이터셋은 [그림 5]과 같이 통신 설정과 환경에 대한 정보로 구성된다.

- 중간 홉 개수 : 송신 노드에서 수신 노드로 데이터를 전송할 때 경로상에 존재하는 홉의 개수이다. 전송 시작 단계에서의 시그널링을 통해 획득한 값을 통해 입력된다.

- 전송 데이터 크기 : 송신 노드에서 수신 노드에게 보내고자 하는 데이터의 크기로 Byte 단위로 나타난다. 송신 노드에서 생성되어 데이터셋에 기록된다.

▪ DPI를 수행하는 중간 홉 비율 : 중간에 존재하는 홉 중 총 몇 퍼센트가 DPI를 진행하고 있는지를 나타내는 수치이다. 중간 홉 개수와 마찬가지로 전송 시작 단계에서의 시그널링을 통해 획득한 값을 통해 입력된다.

▪ 각 홉의 DPI 여부 : 각각의 홉이 DPI를 진행하고 있는지를 나타내는 컬럼이다. 0이면 해당 홉이 DPI를 진행하지 않는 것을, 1이면 DPI를 진행함을 나타낸다. 각 홉의 DPI의 여부는 중간 홉 개수만큼의 컬럼으로 생성된다. 전송 시작 단계에서의 시그널링을 통해 획득한 시그널을 통해 각각의 홉에 대해 입력된다.

▪ 각 구간에서 사용한 암호화 : 초기 학습 때는 무작위로 생성된 각 중간 홉의 암호화가 들어가 있으며, 머신러닝 모델 생성 이후 학습에는 모델 예측으로 발생한 각 홉의 암호화가 포함된다. 항목은 'TenSEAL'과 'AES' 두 가지가 존재한다. 각 구간에서 사용한 암호화는 송신 노드에서의 전송을 포함하기 때문에 중간 홉의 개수보다 한 개 더 많은 개수의 컬럼으로 생성된다. 또한 학습 및 예측의 레이블로 사용되기 때문에 예측에서는 해당 값을 도출한다.

▪ DPI 연산 횟수 : 중간 홉의 DPI 전체 프로세스 중 머신러닝을 통해 동작하는 DPI 동작이 수행되는 횟수를 나타낸다. DPI 연산 횟수는 DPI가 처리하는 프로세스 및 DPI 제품이 복잡한 정도에 따라서 차이가 존재할 수 있다. 서버의 시그널을 통해 DPI 연산 횟수를 사전에 입력된다.

▪ 전송 지연시간 : 송신 노드와 수신 노드에서 한 번의 통신이 진행될 때 발생되는 시간이며 초 단위로 측정된다. 전송 지연시간은 통신이 완전히

진행되어야 발생하는 값으로, 송신 노드에서 머신러닝 모델을 통한 예측을 진행하는 상황에서는 존재하지 않는다.

2) 데이터셋 전처리

생성된 학습 데이터셋을 모델에 입력으로 하기 위해 각 구간에서 사용한 암호화 컬럼의 'TenSEAL'과 'AES'를 0과 1로 대응하여 레이블링한다.

전송 지연시간은 통신이 끝까지 진행되지 않으면 발생할 수 없는 특성으로 인하여 학습 데이터셋에는 존재하지만, 예측하기 위해 입력으로 주는 데이터에서는 존재할 수 없다. 따라서 학습 및 예측에서 전송 지연시간을 그대로 사용하는 것이 아니라 처리를 통해 별도의 컬럼을 생성하여 사용한다.

본 연구는 암호화 알고리즘을 통해 전송 지연시간을 줄이는 것이 주요한 목적이므로 [그림 5]와 같이 낮은 전송 지연시간을 갖는 상위 데이터를 표시하는 상위 20% 전송 지연시간 컬럼을 생성한다.

상위 20% 전송 지연시간 컬럼은 동일한 중간 홉 개수, DPI를 수행하는 중간 홉 비율, 전송 데이터 크기, DPI 연산 횟수를 가지는 통신 상황에 대한 데이터셋 중 낮은 전송 지연시간 값을 가지는 상위 20%의 항목을 1로 표시한다. 그 외의 상위 20%에 해당하지 않는 값들은 0으로 표시된다. 해당 과정을 통해 학습에서는 낮은 전송 지연시간을 가지는 상황을 학습할 수 있고, 예측 상황에서는 단순 입력만으로도 낮은 전송 지연시간을 가지는 암호화를 획득할 수 있다.

3) 학습 및 모델 생성

머신러닝의 학습 및 모델 생성은 앞서 처리된 데이터셋을 이용하여 진행한다. 사용하는 머신러닝 모델은 방대한 네트워크 트래픽의 양을 고려하여 큰 데이터셋을 빠르고 정확하게 학습 및 예측을 진행할 수 있는

LightGBM(Light Gradient Boosting Machine)을 선택하였다.

학습은 [그림 5]에서의 중간 홉 개수, 전송 데이터 크기, DPI를 수행하는 중간 홉 비율, 각 홉의 DPI 여부, DPI 연산 횟수, 상위 20% 전송 지연시간 컬럼을 학습 매개변수로 입력하여 진행한다. 예측의 결과로는 통신 환경에 최적화된 각 노드의 암호화이므로, 머신러닝 학습 및 모델 생성에서는 각 노드의 암호화를 레이블로 설정한다.

학습은 송신 노드에서 암호화하여 전송하는 과정을 포함하기 때문에 [그림 2]처럼 중간 홉 개수보다 한번 많은 횟수로 진행된다. 머신러닝 모델도 마찬가지의 개수로 생성이 되어 통신이 진행되는 각 구간의 암호화를 예측할 수 있도록 한다. 각각의 학습에서는 해당 구간에서 사용한 암호화만을 가져와 레이블로 사용한다. 즉, 중간 홉의 개수가 10개면 암호화 통신 구간은 총 11개가 존재한다. 이때 학습은 각각의 통신 구간을 나누어 진행하기 때문에 머신러닝 모델은 각각의 통신 구간에 생성이 되어 총 11개가 발생하게 된다.

머신러닝 모델이 생성된 이후 Dynamic Cryptographic Selection을 이용한 통신이 시작되면, 송신 노드는 시그널링을 통해서 중간 홉 개수, DPI를 수행하는 중간 홉 비율, DPI의 연산 횟수, 각 홉의 DPI 여부 정보를 획득한다. 획득한 정보 및 송신 노드에서 생성한 전송 데이터 크기를 결합하고, 상위 20% 전송 지연시간 컬럼 추가 후 값을 1로 설정하여 모델에 입력될 수 있도록 재구성한다. 재구성된 데이터로 진행되는 각 홉의 암호화 예측은 해당 통신에 존재하는 모든 전송 구간에서 진행된다. 즉 현재 통신 상황에 최적이라고 예측된 암호화는 중간 홉 개수보다 한 개 더 많이 생성된다.

송신 노드는 최적의 암호화 설정을 전송을 진행할 모든 노드 및 홉에 작은 패킷으로 전송하여 사전에 알린다. 통신이 진행되는 노드들은 진행할 암호화를 설정하고 최종적으로 암호화 통신을 수행한다.

통신으로 인해 생성된 네트워크 통신 데이터셋은 본 알고리즘의 성능 개선을 위해 학습 데이터셋과 결합하여 다시 학습하는 과정을 반복한다.

제 V 장 실험 및 결과

1. 실험 환경

1) 전제 조건

본 논문의 실험의 가정 사항은 다음과 같이 정의하였다.

첫째, 통신은 실제 네트워크에서 진행하지 않는다.

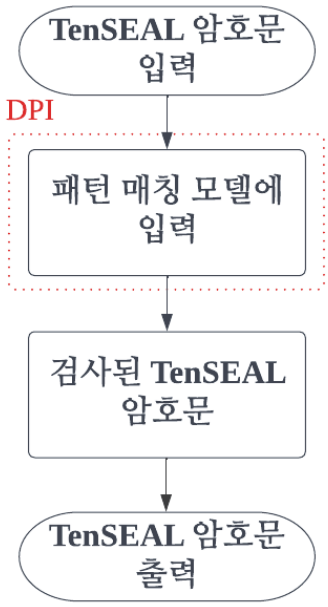
통신은 단일 컴퓨터에서 Python을 통해 구현된 통신 환경에서 진행된다. 따라서 실험의 통신은 실제 네트워크를 통한 패킷 교환이 아닌 하나의 환경에서 패킷을 주고받는 과정이다.

둘째, 실험의 통신에서 전송 실패는 존재하지 않는다.

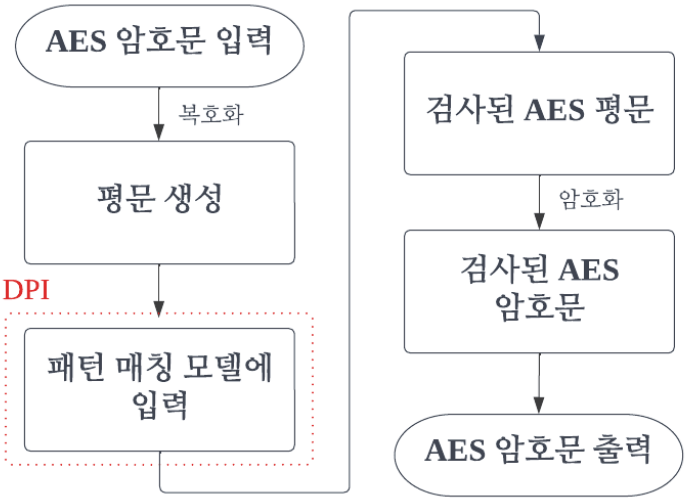
본 논문은 전송 지연시간을 중심으로 제안한 방식을 평가하기 때문에 전송 지연시간을 보다 원활히 나타낼 수 있도록 통신에서 재전송과 같은 불필요한 과정을 제외하였다. 따라서 송신 노드에서 전송된 모든 패킷이 중간 홉 및 수신 노드로 실패없이 전송이 됨을 가정한다.

셋째, 통신에서 암호화의 변경은 5개의 홉을 지나야 진행된다.

본 실험에서는 중간 홉에서 암호화 변경이 존재할 수 있다. 서버는 해당 과정에 관여하여 정상적인 암호화 변경이 이루어질 수 있도록 한다고 가정한다. 암호화가 변경되는 횟수는 예측에 따라 다르게 발생하고, 최악의 경우 중간 홉 개수만큼 암호화가 변경될 수 있다. 이는 지나친 오버헤드를 발생시키기 때문에 암호화를 일정 개수만큼 유지하도록 하여 오버헤드 문제를 예방하고자 한다. 본 실험에서는 예측을 통해 나온 각 통신 구간의 암호화를 홉의 순서대로 5개를 묶어 그 중 더 많이 발생한 암호화 방식을 다수결로 선택하여 해당 5개의 암호화를 지정한다.



[그림 6] TenSEAL 암호화에서의 DPI



[그림 7] AES 암호화에서의 DPI

넷째, DPI 과정 중 일부를 통해 DPI가 동작 된다.

각각의 암호화에서의 가정된 DPI를 수행하는 흐름은 [그림 6, 7]과 같다. DPI는 실제 환경에서 사용되는 완전한 제품이 아니라 DPI 과정 중 패턴 매칭만을 포함하여 간단한 동작으로 진행한다. 패턴 매칭은 머신러닝에 적용이 가능한 TenSEAL의 특성을 고려하여 [24]에서 제안된 DPI 모델과 같이 학습된 머신러닝 모델 예측을 통해 수행한다. DPI의 간소화 및 모델의 예측이 전송 지연시간 등의 결과에 영향을 주지 않는 실험의 특성으로 악성 행위 탐지는 예외 없이 모두 성공한다고 가정한다.

2) 시뮬레이션 환경

시뮬레이션은 Python으로 간단하게 구현한 통신 환경을 통해 실행되며, 사용한 구성요소와 버전은 [표 2]와 같다.

[표 2] 실험에 사용된 컴퓨터 환경

구성요소	사용한 환경
CPU	Intel(R) Core(TM) i9-10850K CPU @ 3.60GHz 3.60 GHz
RAM	32GB
Python	Python 3.8.12
TenSEAL Context	Polynomial : 8192 Coefficient : [60, 40, 40, 60]
AES Keysize	256bit

본 논문의 실험은 Intel사의 i9-10850K CPU와 32GB의 RAM을 가진 컴퓨터에서 진행되었다. 실험의 모든 환경의 구현은 Python 3.8.12 버전에서 진행하였다. 해당 환경에서 AES 암호화는 256bit의 키를 이용하였고

TenSEAL Context의 Polynomial 차수는 8192, Coefficient 계수는 [60, 40, 40, 60]로 구성하여 진행하였다.

2. 실험 결과

실험은 중간 홉의 개수, 전송 데이터 크기, DPI를 수행하는 중간 홉 비율, DPI 연산 횟수가 변할 때 전송 지연시간을 출력했다. 본 실험에서 사용된 전송 지연시간은 데이터가 송신 노드에서 전송을 시작하여 수신 노드까지 도착한 시간을 의미한다.

1) 중간 홉 개수

중간 홉의 개수가 변화할 때, 전송 데이터 크기가 500Byte인 경우와 3500Byte인 경우를 나누어 그래프를 출력했다. 다음 [표 3]은 중간 홉 개수에 따른 전송 지연시간 변화 시뮬레이션에서 사용된 매개변수를 정리한 것이다.

[표 3] 중간 홉 개수 시뮬레이션 매개변수

매개변수	설정 값
중간 홉 개수	10 ~ 90
전송 데이터 크기(Byte)	500, 3500
DPI를 수행하는 중간 홉 비율(%)	40
DPI 연산 횟수	900

DPI를 수행하는 중간 홉 비율은 40%로 DPI 연산 횟수는 연산 900회로 고정하여 실험을 진행하였다. 전송 데이터 크기는 500Byte와 3500Byte 두 가지의 경우로 나누어 결과를 보였다. 중간 홉 개수는 10개부터 90개까지

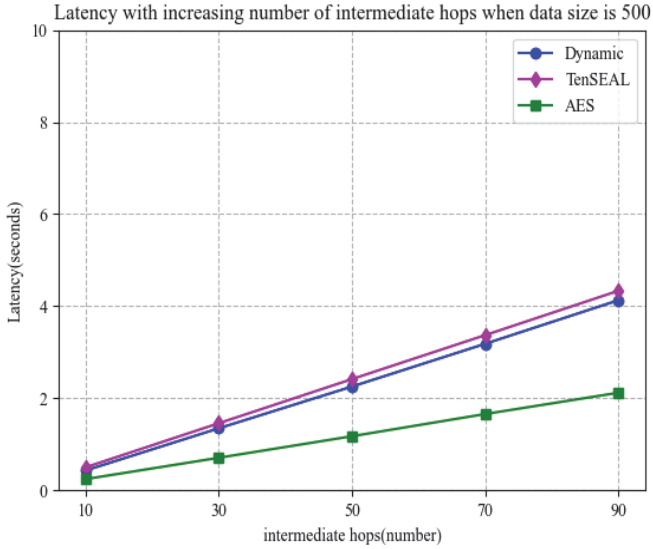
20개씩 증가시켜 개수에 따른 경향을 확인하고자 하였다.

[그림 8]은 전송 데이터 크기가 500Byte인 경우 중간 홉 개수가 10부터 90까지 20개의 간격으로 증가했을 때를 출력한 것이며 [그림 9]는 전송 데이터 크기가 3500Byte일 때 중간 홉 개수가 10개부터 90개까지 20개의 간격으로 증가했을 때를 출력하여 나타낸 것이다.

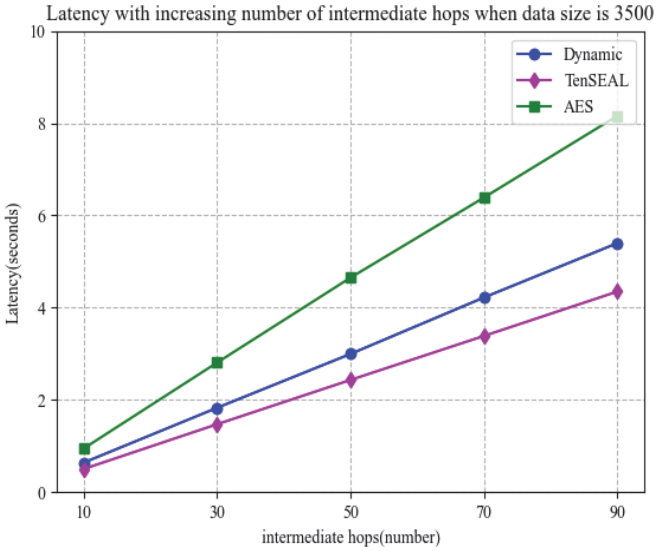
전송 데이터 크기가 500Byte인 경우에서 중간 홉 개수가 90개인 경우 지점을 보면 AES 암호화 단일 적용 통신의 전송 지연시간은 약 2.11초였으며 TenSEAL 암호화 단일 적용 통신의 전송 지연시간은 약 4.33초였다. 제안하는 방식인 Dynamic Cryptographic Selection은 약 4.1초였다.

데이터가 500Byte인 경우 Dynamic Cryptographic Selection이 AES 암호화 단일 적용 통신보다 약 2배 높은 전송 지연시간을 보였으며, TenSEAL 암호화 단일 적용 통신과는 유사한 전송 지연시간을 보였다.

전송 데이터 크기가 3500Byte일 때의 중간 홉 개수가 90개인 경우에는 AES 암호화 단일 적용 통신 지연시간은 약 8.1초였으며 TenSEAL 암호화 단일 적용 통신은 약 4.3초였다. Dynamic Cryptographic Selection은 약 5.3초로 AES 암호화 단일 적용 통신보다 약 1.5배 적은 전송 지연시간을 보였다. 이를 통해 중간 홉 개수 및 전송 데이터 크기의 변화가 존재하여도 다른 방식들과는 달리 Dynamic Cryptographic Selection이 일관되게 좋은 성능을 보임을 확인할 수 있다.



[그림 8] 전송 데이터 크기가 500Byte일 때 멀티 홉 수에 따른 전송 지연시간



[그림 9] 전송 데이터 크기가 3500Byte일 때 멀티 홉 수에 따른 전송 지연시간

2) 전송 데이터 크기

전송 데이터 크기가 변화할 때 DPI 연산 횟수에 따른 전송 지연시간을 확인했다. [표 4]는 전송 데이터 크기 변화 시뮬레이션에서 사용된 매개변수를 정리한 것이다.

[표 4] 전송 데이터 크기 시뮬레이션 매개변수

매개변수	설정 값
중간 홉 개수	90
전송 데이터 크기(Byte)	500 ~ 3500
DPI를 수행하는 중간 홉 비율(%)	40
DPI 연산 횟수	100, 900

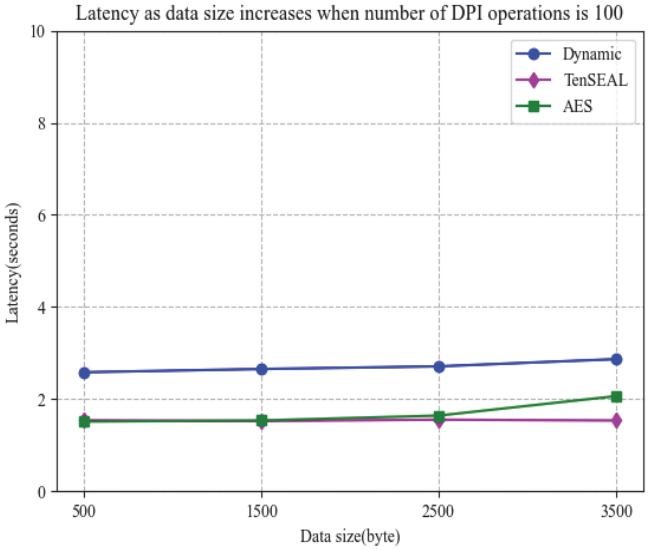
DPI를 수행하는 중간 홉 비율은 40%로 중간 홉 개수는 90으로 고정하여 실험을 진행하였다. DPI 연산 횟수는 연산이 100회 존재할 때와 900회일 때로 나누어 결과를 보였다. 전송 데이터 크기는 500Byte부터 3500Byte까지 1000Byte씩 증가시켜 크기 증가에 따른 경향을 보이고자 하였다.

다음 [그림 10]은 전송 데이터 크기가 변화할 때 DPI 연산 횟수가 연산 100회일 경우를, [그림 11]은 DPI 연산 횟수가 연산 900회일 때를 나타낸 것이다.

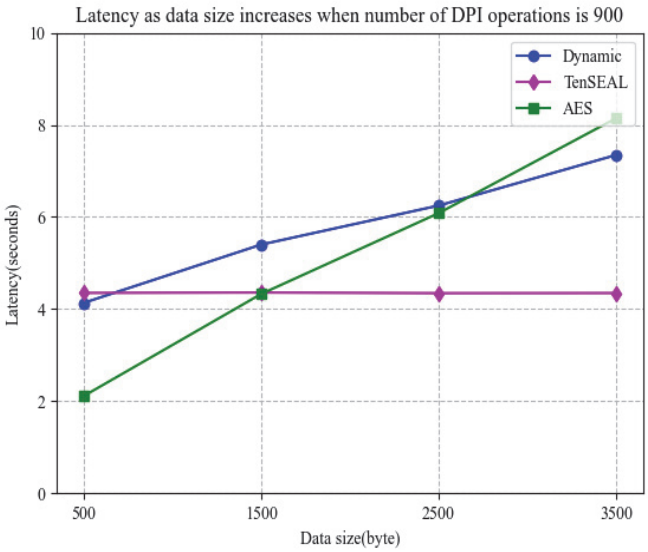
DPI 연산 횟수를 100회의 연산 횟수로 고정하고 전송 데이터 크기가 3500Byte 일 때, AES 암호화 단일 적용 통신은 약 2.0초의 전송 지연시간이 출력되었다. TenSEAL 암호화 단일 적용 통신은 약 1.5초이며 Dynamic cryptographic algorithm은 약 2.8초의 전송 지연시간을 출력했다. DPI 연산 횟수가 100일 때는 전송 데이터 크기에 상관없이 Dynamic Cryptographic Selection이 AES 암호화 단일 적용 통신 및 TenSEAL 암

호화 단일 적용 통신의 방식보다 전송 지연시간이 높으나, 출력된 전송 지연시간의 차이가 미미하다.

DPI 연산 횟수를 900회로 증가시켰을 때 전송 데이터 크기가 3500Byte인 경우 AES 암호화 단일 적용 통신은 약 8.1초의 전송 지연시간을 출력했다. TenSEAL 암호화 단일 적용 통신은 약 4.3초의 전송 지연시간을 출력했으며, Dynamic cryptographic algorithm은 약 7.3초의 전송 지연시간이 출력되었다. DPI 연산 횟수가 900일 때, AES 암호화 단일 적용 통신과 TenSEAL 암호화 단일 적용 통신 전송 데이터 크기가 1500Byte인 지점에서 교차 지점 존재하는 것을 확인할 수 있다. 교차 지점을 기준으로 Dynamic Cryptographic Selection의 모델은 암호화 방식을 다르게 선택하기 때문에 AES 및 TenSEAL 암호화 단일 적용 통신의 특성이 모두 반영되기 때문에 두 암호 알고리즘에 비해 Dynamic Cryptographic Selection은 성능이 유지된다.



[그림 10] DPI 연산 횟수의 연산이 100회일 때 멀티 홉 수에 따른 전송 지연시간



[그림 11] DPI 연산 횟수의 연산이 900회일 때 멀티 홉 수에 따른 전송 지연시간

3) DPI를 수행하는 중간 홉 비율

DPI를 수행하는 중간 홉 비율이 변화할 때, 전송 데이터 크기 500Byte와 3500Byte에 대한 전송 지연시간을 출력했다. 다음 [표 5]는 DPI를 수행하는 중간 홉 비율 시뮬레이션에서 사용된 매개변수를 정리한 것이다.

[표 5] DPI를 수행하는 중간 홉 비율 시뮬레이션 매개변수

매개변수	설정 값
중간 홉 개수	90개
전송 데이터 크기(Byte)	500, 3500 Byte
DPI를 수행하는 중간 홉 비율(%)	0 ~ 100 %
DPI 연산 횟수	900회

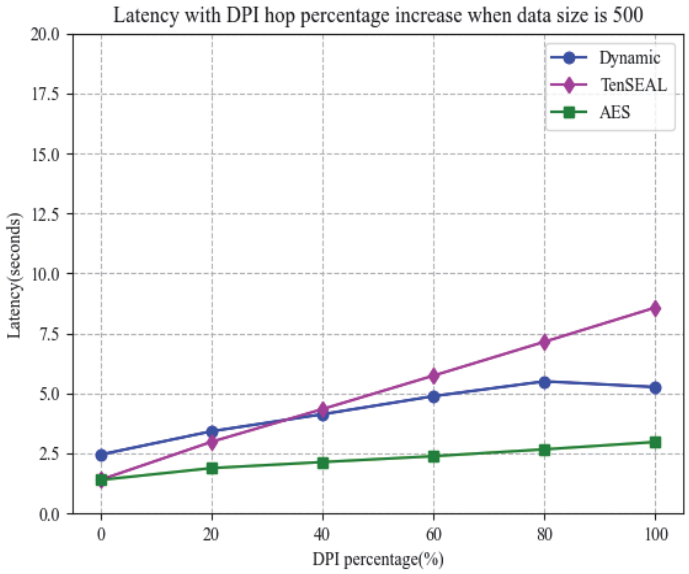
DPI 연산 횟수는 900회의 연산으로 중간 홉 개수는 90개로 고정하여 실험을 진행하였다. 전송 데이터 크기는 500Byte일 때와 3500Byte일 때로 나누어 결과를 보였다. DPI를 수행하는 중간 홉 비율은 0%부터 100%까지 20%씩 변화시켜 비율 증가에 따른 경향을 보이고자 하였다.

[그림 12]는 전송 데이터 크기가 500Byte인 경우 DPI를 수행하는 중간 홉 비율이 0부터 100까지 20의 간격으로 증가했을 때를 출력한 것이며 [그림 13]은 전송 데이터 크기가 3500Byte일 때 DPI를 수행하는 중간 홉 비율이 0%부터 100%까지 20%의 간격으로 증가했을 때를 출력하여 나타낸 것이다.

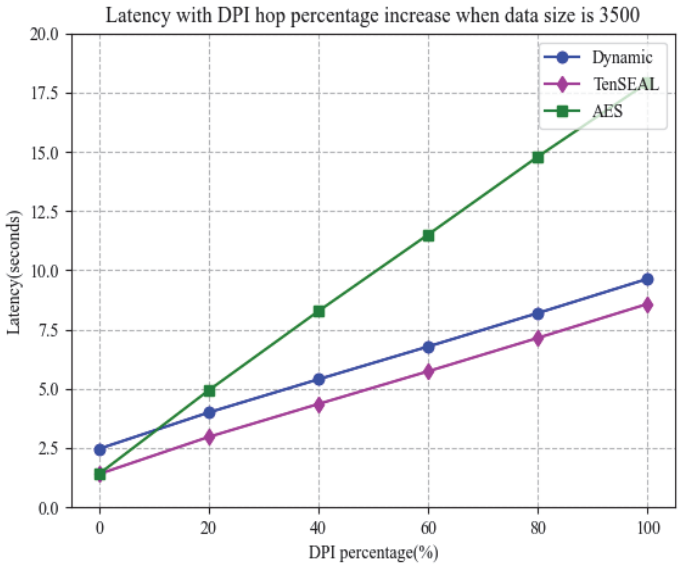
전송 데이터 크기가 500Byte일 때 DPI를 수행하는 중간 홉 비율이 100%인 지점을 보면, AES 암호화 단일 적용 통신은 약 2.9초의 전송 지연시간이 출력되는 것을 확인할 수 있다. TenSEAL 암호화 단일 적용 통신은 약 8.5초의 전송 지연시간이 출력되며, Dynamic Cryptographic

Selection은 약 5.2초가 출력되는 것을 확인할 수 있다. [그림 13]은 전송 데이터 크기가 3500Byte일 때, 변화되는 DPI를 수행하는 중간 홉 비율을 나타낸 것이다. [그림 13]을 통해 전송 데이터 크기가 3500Byte인 경우, DPI를 수행하는 중간 홉 비율이 100%인 지점을 보았을 때 AES 암호화 단일 적용 통신의 전송 지연시간은 약 17.9초가 출력되었다.

TenSEAL 암호화 단일 적용 통신은 약 8.5초가 출력되었으며, Dynamic Cryptographic Selection은 약 5.2초가 출력되었다. 전송 데이터 크기가 각각 500Byte, 3500Byte일 때 AES 및 TenSEAL의 암호화 방식은 DPI를 수행하는 중간 홉 비율 및 전송 데이터 크기에 의해 성능 차이가 발생한 것을 확인할 수 있다. 이와 달리 Dynamic Cryptographic Selection은 DPI를 수행하는 중간 홉 비율 및 전송 데이터 크기에 영향을 받지 않아 성능이 유지되는 것을 확인할 수 있다.



[그림 12] 전송 데이터 크기가 500Byte일 때 DPI를 수행하는 중간 홉 비율에 따른 전송 지연시간



[그림 13] 전송 데이터 크기가 3500Byte일 때 DPI를 수행하는 중간 홉 비율에 따른 전송 지연시간

4) DPI 연산 횟수

전송 데이터 크기가 500Byte와 3500Byte일 때 DPI 연산 횟수 변화량을 확인하고자 했다. 다음 [표 6]은 DPI 연산 횟수 변화량 시뮬레이션에서 사용된 매개변수를 정리한 것이다.

[표 6] DPI 연산 횟수 시뮬레이션 매개변수

매개변수	설정 값
중간 홉 개수	90개
전송 데이터 크기(Byte)	500, 3500Byte
DPI를 수행하는 중간 홉 비율(%)	40%
DPI 연산 횟수	100 ~ 900회

전송 데이터 크기는 500Byte일 때와 3500Byte일 때로 나누어 결과를 나누어 출력했다. 전송 데이터 크기는 500Byte일 때와 3500Byte일 때로 나누어 결과를 보였다. DPI 연산 횟수는 연산 횟수가 100회부터 900회까지 200회씩 증가시켜 횟수에 따른 경향을 보이고자 하였다.

[그림 14]는 전송 데이터 크기가 500Byte인 경우 DPI 연산 횟수가 100회부터 900회까지 200의 간격으로 증가했을 때를 출력한 것이며 [그림 15]는 전송 데이터 크기가 3500Byte일 때 DPI 연산 횟수가 100회부터 900회까지 200회의 간격으로 증가했을 때를 출력하여 나타낸 것이다.

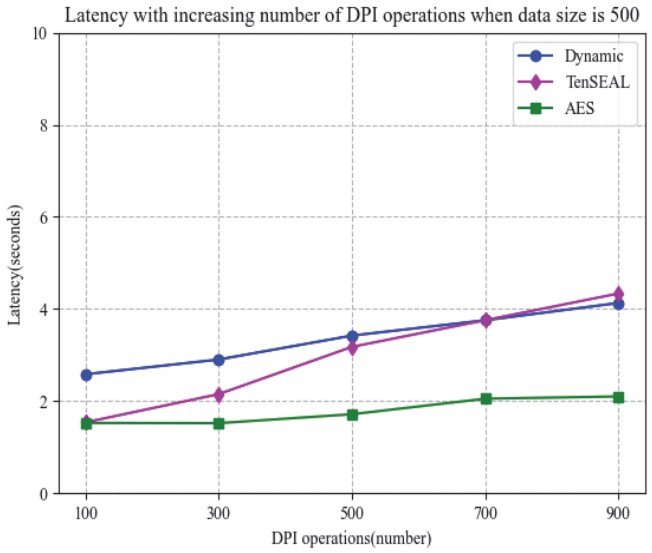
DPI 연산 횟수가 연산 900회일 때, AES 암호화 단일 적용 통신의 전송 지연시간은 약 2.1초이다. TenSEAL 암호화 단일 적용 통신은 약 4.3초이며, Dynamic Cryptographic Selection은 약 4.1초의 전송 지연시간이 출력되었다. 그림12는 전송 데이터 크기가 3500Byte일 때, DPI 연산 횟수를 출력한 것이다. DPI 연산 횟수가 900인 지점에서 AES 암호화 단일 적용 통

신의 전송 지연시간은 약 8.1초이다. TenSEAL 암호화 단일 적용 통신은 약 4.3초이며, Dynamic Cryptographic Selection은 약 5.3초의 전송 지연시간이 출력되었다.

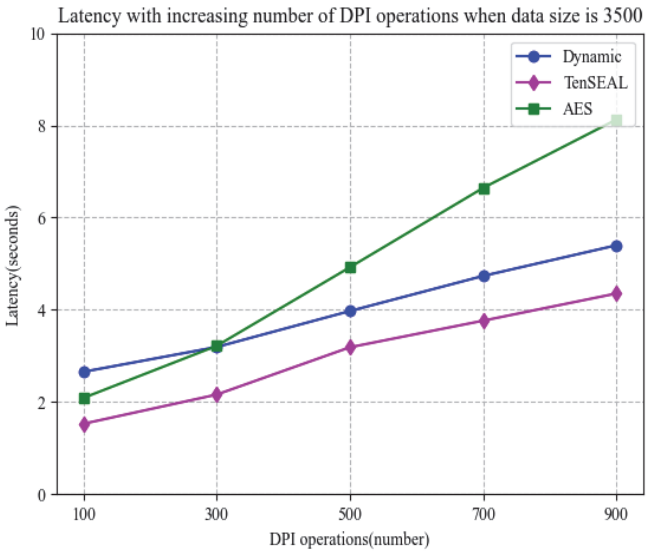
전송 데이터 크기가 각각 500Byte 및 3500Byte 일 때 DPI 연산 횟수가 연산 900회인 지점을 비교하면, AES 암호화 단일 적용 통신의 전송 지연시간은 약 3.8배 증가한 것을 확인할 수 있다. TenSEAL 암호화 단일 적용 통신은 전송 데이터 크기가 변하더라도 성능이 유지되었으며, 두 가지 암호화 알고리즘의 특성을 모두 반영하는 Dynamic Cryptographic Selection의 경우 약 1.2배의 전송 지연시간이 증가한 것을 확인할 수 있다.

이를 통해 AES 암호화 단일 적용 통신은 전송 데이터 크기 및 DPI 연산 횟수에 따라 성능이 조절되는 것을 알 수 있었으며, TenSEAL 암호화 단일 적용 통신 및 Dynamic Cryptographic Selection은 AES 암호화 단일 적용 통신에 비해 상대적으로 성능이 유지되는 것을 확인할 수 있었다.

시뮬레이션 결과를 통해 Dynamic Cryptographic Selection의 성능을 확인하고자 했다. 시뮬레이션을 통해 AES 및 TenSEAL 암호화 단일 적용 통신의 암호화 방식보다 제안된 Dynamic Cryptographic Selection이 네트워크 환경에 따라 암호화 방식을 다르게 적용함으로써, 성능이 유지되는 것을 확인할 수 있었다. 이를 통해 Dynamic Cryptographic Selection은 종래의 암호 알고리즘보다 네트워크 환경에 따른 영향을 적게 받으면서, 보안성과 통신 지연시간을 유지할 수 있다는 것을 암시한다.



[그림 14] 전송 데이터 크기가 500Byte일 때 DPI 연산 횟수에 따른 전송 지연시간



[그림 15] 전송 데이터 크기가 3500Byte일 때 DPI 연산 횟수에 따른 전송 지연시간

제 Ⅵ 장 결론

네트워크 기술의 발달로 통신 장치의 수가 기하급수적으로 증가하기 시작하면서 멀티 홉 환경의 통신이 대중화되었다. 멀티 홉 형태의 네트워크가 확산되면서 중간 홉에서 데이터를 처리 및 전송할 때 데이터를 탈취하거나 유출하려는 위협이 증가하고 있다. 종래의 통신 방식은 데이터의 기밀성 및 무결성을 보호하고 통신 환경의 저지연 전송을 확보하기 위해 AES 암호화를 적용하고 있다. 이 외에도 암호화 상태에서 연산이 가능한 동형암호화를 통신 환경에서 발생되는 데이터에 적용하거나 반동형암호를 적용하여 데이터의 보안성을 높이고자 했다. 그러나 종래의 연구는 데이터의 DPI 여부나 데이터의 크기 등과 같은 네트워크 환경에 영향을 주는 요소를 고려하지 못해 수시로 변화하는 멀티 홉 네트워크 환경을 고려하지 못했다.

따라서 본 논문에서는 멀티 홉의 수, DPI 여부, 전송 데이터 크기 등 네트워크 상황에 따라 동적으로 동작하는 Dynamic Cryptographic Selection을 제시하고자 한다. 제안된 알고리즘을 통해 멀티 홉 네트워크 상황에 최적화된 암호화 알고리즘을 선택적으로 적용함으로써, 저지연 전송 환경을 확보할 수 있다. 시뮬레이션에서는 동형암호화와 AES 암호화 단일 적용 통신 각각의 경우에서 노드의 개수와 전송 데이터 크기의 길이의 변화에 따른 전송 시간의 차이를 비교하였다.

시뮬레이션 결과를 통해 완전동형암호화 방식인 TenSEAL과 AES 암호화 단일 적용 통신의 전송 지연시간이 교차하는 조건을 찾았다. 노드의 개수와 데이터의 크기가 큰 환경에서는 TenSEAL을 사용하는 것이 더 적은 전송 지연시간을 확보할 수 있었다. 제안된 Dynamic Cryptographic Selection의 모델은 네트워크 환경에 따라 암호화 방식을 다르게 적용함으로써, AES 및

TenSEAL 암호화 단일 적용 통신보다 성능이 유지되는 것을 확인할 수 있었다. 이를 통해 네트워크 통신 환경의 변화가 잦고 다양한 변수를 고려해야 하는 멀티 홉 환경에 Dynamic Cryptographic Selection이 적합한 것을 보였다.

본 연구에서는 네트워크 시뮬레이션을 통한 두 암호화 방식에 대한 비교로 각각의 암호화에서 적합한 환경을 파악하였다. 후속 연구에서는 본 연구의 결과를 학습하여 데이터 전송 시 자동으로 의사 결정을 진행하는 모델을 제안하는 연구를 진행할 계획이다.

참 고 문 헌

[1] Tao, Fei, Qinglin Qi, Ang Liu, and Andrew Kusiak. “Data-Driven Smart Manufacturing.” *Journal of Manufacturing Systems* 48 (2018): 157 - 69. https://doi.org/10.1016/j.jmsy.2018.01.006.

[2] Jan, Mian Ahmad, Wenjing Zhang, Muhammad Usman, Zhiyuan Tan, Fazlullah Khan, and Entao Luo. “SmartEdge: An End-to-End Encryption Framework for an Edge-Enabled Smart City Application.” *Journal of Network and Computer Applications* 137 (2019): 1 - 10. https://doi.org/10.1016/j.jnca.2019.02.023.

[3] Burkhalter, Lukas, Nicolas Küchler, Alexander Viand, Hossein Shafagh and Anwar Hithnawi. “Zeph: Cryptographic Enforcement of End-to-End Data Privacy.” *USENIX Symposium on Operating Systems Design and Implementation* (2021).

[4] Song, Wenguang, Mykola Beshley, Krzysztof Przystupa, Halyna Beshley, Orest Kochan, Andrii Pryslupskyi, Daniel Pieniak, and Jun Su. “A Software Deep Packet Inspection System for Network Traffic Analysis and Anomaly Detection.” *Sensors* 20, no. 6 (2020): 1637. https://doi.org/10.3390/s20061637.

[5] MICHAŁOWSKA, Joanna. “Prediction of the Parameters of Magnetic Field of CNC Machine Tools.” *PRZEGLĄD ELEKTROTECHNICZNY* 1, no. 1 (2019): 136 - 38. https://doi.org/10.15199/48.2019.01.34.

[6] Louw, J., G. Niezen, T. D. Ramotsoela, and A. M. Abu-Mahfouz.

"A Key Distribution Scheme Using Elliptic Curve Cryptography in Wireless Sensor Networks." *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*, 2016. https://doi.org/10.1109/indin.2016.7819342.

[7] Moharana, Soumya Ranjan, Vijay Kumar Jha, Anurag Satpathy, Sourav Kanti Addya, Ashok Kumar Turuk, and Banshidhar Majhi. "Secure Key-Distribution in IOT Cloud Networks." *2017 Third International Conference on Sensing, Signal Processing and Security (ICSSS)*, 2017. https://doi.org/10.1109/ssps.2017.8071591.

[8] Bonnoron, Guillaume, Caroline Fontaine, Guy Gogniat, Vincent Herbert, Vianney Lapôtre, Vincent Migliore, and Adeline Roux-Langlois. "Somewhat/Fully Homomorphic Encryption: Implementation Progresses and Challenges." *Codes, Cryptology and Information Security*, 2017, 68 - 82. https://doi.org/10.1007/978-3-319-55589-8_5.

[9] OpenMined. "OpenMined/TenSEAL: A Library for Doing Homomorphic Encryption Operations on Tensors." GitHub. Accessed December 27, 2022. https://github.com/OpenMined/TenSEAL.

[10] Sachs, Joachim, Gustav Wikstrom, Torsten Dudda, Robert Baldemair, and Kittipong Kittichokechai. "5G Radio Network Design for Ultra-Reliable Low-Latency Communication." *IEEE Network* 32, no. 2 (2018): 24 - 31. https://doi.org/10.1109/mnet.2018.1700232.

[11] Alhammadi, Abdulraqeb, Mardeni Roslee, Mohamad Yusoff Alias, Ibraheem Shayea, Saddam Alraih, and Khalid Sheikhidris Mohamed. "Auto Tuning Self-Optimization Algorithm for Mobility

Management in LTE-A and 5G Hetnets." *IEEE Access* 8 (2020): 294 - 304. https://doi.org/10.1109/access.2019.2961186.

[12] Marcolla, Chiara, Victor Sucasas, Marc Manzano, Riccardo Bassoli, Frank H.P. Fitzek, and Najwa Aaraj. "Survey on Fully Homomorphic Encryption, Theory and Applications," 2022. https://doi.org/10.36227/techrxiv.19315202.v3.

[13] Bettoumi, Balkis, and Ridha Bouallegue. "LC-Dex: Lightweight and Efficient Compressed Authentication Based Elliptic Curve Cryptography in Multi-Hop 6LoWPAN Wireless Sensor Networks in Hip-Based Internet of Things." *Sensors* 21, no. 21 (2021): 7348. https://doi.org/10.3390/s21217348.

[14] Lizardo, André, Raul Barbosa, Samuel Neves, Jaime Correia, and Filipe Araujo. "End-to-End Secure Group Communication for the Internet of Things." *Journal of Information Security and Applications* 58 (2021): 102772. https://doi.org/10.1016/j.jisa.2021.102772.

[15] Sun, Yingnan, Frank P. Lo, and Benny Lo. "Lightweight Internet of Things Device Authentication, Encryption, and Key Distribution Using End-to-End Neural Cryptosystems." *IEEE Internet of Things Journal* 9, no. 16 (2022): 14978 - 87. https://doi.org/10.1109/jiot.2021.3067036.

[16] Mosteiro-Sanchez, Aintzane, Marc Barcelo, Jasone Astorga, and Aitor Urbieta. "Securing IIoT Using Defence-in-Depth: Towards an End-to-End Secure Industry 4.0." *Journal of Manufacturing Systems* 57 (2020): 367 - 78.

https://doi.org/10.1016/j.jmsy.2020.10.011.

[17] Haseeb, Khalid, Naveed Islam, Ahmad Almogren, Ikram Ud Din, Hisham N. Almajed, and Nadra Guizani. "Secret Sharing-Based Energy-Aware and Multi-Hop Routing Protocol for IOT Based Wsns." *IEEE Access* 7 (2019): 79980 - 88. https://doi.org/10.1109/access.2019.2922971.

[18] Salva-Garcia, Pablo, Jose M. Alcaraz-Calero, Qi Wang, Miguel Arevalillo-Herraez, and Jorge Bernal Bernabe. "Scalable Virtual Network Video-Optimizer for Adaptive Real-Time Video Transmission in 5G Networks." *IEEE Transactions on Network and Service Management* 17, no. 2 (2020): 1068 - 81. https://doi.org/10.1109/tnsm.2020.2978975.

[19] Dai, Wenbin, Hiroaki Nishi, Valeriy Vyatkin, Victor Huang, Yang Shi, and Xinping Guan. "Industrial Edge Computing: Enabling Embedded Intelligence." *IEEE Industrial Electronics Magazine* 13, no. 4 (2019): 48 - 56. https://doi.org/10.1109/mie.2019.2943283.

[20] Sengupta, Jayasree, Sushmita Ruj, and Sipra Das Bit. "End to End Secure Anonymous Communication for Secure Directed Diffusion in IOT." *Proceedings of the 20th International Conference on Distributed Computing and Networking*, 2019. https://doi.org/10.1145/3288599.3295577.

[21] Saini, Akanksha, Enrique Festijo, and Younchan Jung. "Proposing Packet-Key Based End-to-End Security Architecture for the Post-LTE Networks." *ICT Express* 5, no. 2 (2019): 124 - 30. https://doi.org/10.1016/j.icte.2018.08.002.

[22] De La Torre Parra, Gonzalo, Paul Rad, and Kim-Kwang Raymond Choo. "Implementation of Deep Packet Inspection in Smart Grids and Industrial Internet of Things: Challenges and Opportunities." *Journal of Network and Computer Applications* 135 (2019): 32 - 46. https://doi.org/10.1016/j.jnca.2019.02.022.

[23] Ren, Hao, Hongwei Litt, Dongxiao Liu, and Xuemin Sherman Shen. "Toward Efficient and Secure Deep Packet Inspection for Outsourced Middlebox." *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019. https://doi.org/10.1109/icc.2019.8761954.

[24] Trivedi, Uday, and Munal Patel. "A Fully Automated Deep Packet Inspection Verification System with Machine Learning." *2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, 2016. https://doi.org/10.1109/ants.2016.7947802.

ABSTRACT

Network Self-Optimization Techniques for Low Latency Encryption Communication

Shim Hye Yeon
Department of Future Convergence Technology Engineering
Graduate School of Sungshin University

With the development of communication technology, the network of multi-hop environments has spread as various devices are connected to the network. As the Internet of Things (IoT) and personal devices become popular, it is intended to process and utilize data generated in real time. In addition, the amount of data generated by digital devices has increased exponentially, and as individual sensitive data is accumulated, the security and confidentiality of data are becoming important. However, there is a problem that conventional studies have not reflected the changing environment of the network by applying encryption algorithms in a simple communication environment, and have not been able to meet both security and low-latency transmission.

This study proposes Dynamic Cryptographic Selection, which selectively uses homomorphic encryption and AES encryption methods,

considering the network communication environment that changes with the number of multi-hops, transmission data size, and DPI status. Through the proposed idea, we propose an encryption mechanism that dynamically optimizes communication according to the network communication environment, and can perform low-latency transmission while increasing the security of encrypted communication's security. The simulation shows that the proposed idea applies different encryption methods depending on the network environment, thereby maintaining performance over communication when AES and TenSEAL are used alone, reflecting the changing network environment.

ACKNOWLEDGEMENTS

본 논문을 지도해주신 이일구 교수님과 제안 아이디어의 구체화를 위한 논의와 초기 시뮬레이션 평가 개발 및 관련 연구 분석에 기여해 준 박태림 학생에게 감사드립니다.