Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.34, NO.4, Aug. 2024 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2024.34.4.597

MyData Cloud: 개인 정보 통제 강화를 위한 안전한 클라우드 아키텍쳐 설계*

허 승 민,[1†] 권 용 희,[1] 김 범 중,[1] 전 기 석,[1] 이 중 희[2‡]
[1,2]고려대학교 (대학원생, 교수)

MyData Cloud: Secure Cloud Architecture for Strengthened Control Over Personal Data*

Seungmin Heo,[1†] Yonghee Kwon,[1] Beomjoong Kim,[1]
Kiseok Jeon,[1] Junghee Lee[2‡]
[1,2]Korea University (Graduate student, Professor)

요 약

마이데이터는 개인데이터 활용 체계의 새로운 패러다임으로, 데이터 주체가 자신의 데이터를 어떻게 사용하고 어디에 제공할 것인지 결정할 수 있다. 데이터 주체의 동의 하에 서비스 제공자는 여러 서비스에 걸쳐 흩어져있는 고객의 데이터를 수집하고 이를 바탕으로 고객 맞춤화된 서비스를 제공한다. 기존의 마이데이터 서비스 모델들에서, 데이터 주체는 데이터 스토리지에 저장된 자신의 개인 정보를 서비스 제공자 또는 제3자의 데이터 프로세서에게 판매할 수 있다. 하지만 개인정보가 한 번 제3자의 프로세서에게 판매되어 그들의 프로세서에 의해 처리될 경우 그 순간부터 데이터를 추적하고 통제할 수 없다는 문제가 발생한다. 따라서 본 논문에서는 기존 마이데이터 운영 모델들의 문제점들을 개선하여 데이터 주체에게 더 높은 통제권을 부여하는 클라우드 모델을 제시한다. 동시에, 클라우드 모델과 같이 데이터 스토리지, 컨트롤러, 프로세서가 모두 한 곳에 모여있는 경우 클라우드가 침해될 시 모든 데이터가 한 번에 침해될 수 있다는 점을 고려하여, 이러한 위험을 줄일 수 있도록 클라우드-디바이스 간 협력적 암호화와 클라우드 컴포넌트들 간 격리 기술을 적용한 클라우드 모델 아키텍쳐를 함께 제시한다.

ABSTRACT

MyData is an approach of personal data management, which grants data subjects the right to decide how to use and where to provide their data. With the explicit consent of the subjects, service providers can collect scattered data from data sources and offer personalized services based on the collected data. In existing service models, personal data saved in data storage can be shared with data processors of service providers or third parties. However, once personal data are transferred to third-party processors, it is difficult for data subjects to trace and control their personal data. Therefore, in this paper, we propose a cloud model where both data storage and processor are located within a single cloud, ensuring that data do not leave the cloud.

Keywords: Personal Data Control, Data Sovereignty, Cloud-Native Security, Cloud Architecture, MyData

Received(04. 29. 2024), Modified(1st: 05. 20. 2024, 2nd: 06. 07. 2024), Accepted(06. 14. 2024)
* 이 논문은 2021년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구 결과임(No. RS-2021-II210528, 하드웨어 중심 신뢰계산기반과 분산 데이터보호박스를 위한 표준 프로토콜 개발)
† 주저자, nolzaheo@korea.ac.kr
‡ 교신저자, j_lee@korea.ac.kr(Corresponding author)

I. 서 론

유럽연합이 일반 데이터 보호 규칙(General Data Protection Regulation)을 시행하면서, 수집된 개인정보를 단순히 소유하는 수준에서 그치지 않고 이를 적극적으로 활용하고자 하는 시도가 증가하고 있다. 개인정보 활용과 동시에 개인정보보호 또한 중요시됨에 따라, 이 두 가지 요인을 모두 보장하기 위해 GDPR은 데이터 주체에게 개인 정보에 대한 통제권을 부여하여 서비스 사업자가 데이터 주체의 명시적인 동의를 얻어야만 데이터를 수집 및 이용할 수 있도록 허용한다[1]-[3]. 이 개념을 실현한 서비스 형태가 '마이데이터'이다.

마이데이터 서비스는 다양한 유형으로 상용화되고 있다[4]. 접근권, 정정권, 삭제권과 같이 정보주체가 서비스에 행사할 수 있는 자기정보통제 권한에 관한 정보를 제공해주는 서비스, 정보주체 동의를 관리하는 서비스 그리고 정보주체가 본인 정보를 통합하여 조회할 수 있도록 해주는 개인정보 관리 서비스가 있다[5]-[11].

개인정보 관리 서비스를 운영하는 메커니즘은 개인정보를 보관해두는 위치에 따라 크게 두 가지 모델로 나뉘며 operator 모델과 device 모델이 있다. operator 모델은 중앙 집중식 데이터 컨트롤러가 모든 데이터 주체의 개인정보를 수집하여 사업자 측 서버에 저장한다. 이 경우 사업자 측 입장에서 고객들의 데이터에 쉽게 접근하고 이를 관리할 수 있다는 장점이 있지만, 중앙 집중된 컨트롤러가 공격자에 의해 침해되면 수집된 모든 데이터가 보안상 위험해진다는 단점이 있다. device 모델은, 수집된 데이터를 회사 측 서버가 아닌 스마트폰, 태블릿과 같은 고객의 기기에 저장하는 방식으로, operator 모델보다 더 높은 수준의 데이터 통제권을 데이터 주체에게 제공한다[11]-[13].

두 모델 모두 마이데이터 서비스 사업자는 수집된 데이터를 제3자인 데이터 이용자에게 판매할 수 있고, 그 수익을 정보주체 및 기타 관계자와 분배하여 이익을 취할 수 있다. 이는 정보주체의 명시적인 동의 하에 개인정보 활용을 용이하게 한다. 하지만, 수집된 데이터가 제3자에게 이전되는 순간 정보주체는 자신의 개인정보를 추적하고 통제하기 어려워진다는 문제가 발생한다.

따라서 이 논문에서는 흩어진 개인 데이터가 하나의 클라우드상에서 수집되고 처리되도록 하는 cloud 모델을 제시한다. 수집된 데이터는 클라우드상에만 머무르며 오직 분석 연산에 의해 처리된 결괏값만 클라우드 밖으로 나갈 수 있다. 데이터 분석을 위해 클라우드상에 수집된 데이터를 제3자에게 전송하는 대신, 제3자가 데이터 분석 연산을 클라우드로 전송하는 것이다. 이를 통해 데이터가 클라우드 밖을 떠나지 않게 함으로써 데이터 주체의 데이터 통제권을 높이고, 데이터 공격표면을 줄일 수 있다. 더 나아가 필요에 따라 데이터 분석에 필요한 컴퓨팅 리소스를 할당하고 해제함으로써 클라우드 확장성을 활용한 장점을 극대화할 수 있다. 하지만 클라우드가 침해될 경우 클라우드 상의 모든 데이터가 위험에 처할 수 있다는 점을 고려하여, 본 논문에서는 cloud 모델을 사용하여 마이데이터 서비스를 운영하기 위한 안전한 클라우드 아키텍처를 함께 제안한다.

제안하는 아키텍처는, cloud 모델이 데이터 스토리지, 컨트롤러, 프로세서가 동일한 클라우드상에서 동작하는 구조를 취함에 따라 발생할 수 있는 데이터 침해 위협을 방지하는 것을 목표로 한다. 기존의 마이데이터 운영 모델 중 device 모델은 데이터 스토리지를 고객의 기기에 둠으로써 데이터를 격리한다. 따라서 이 모델은 서버가 침해되더라도 데이터가 고객의 기기에 저장되어있어 침해로부터 데이터를 안전하게 유지할 수 있다. 또한, 클라이언트의 리퀘스트에 항상 응답해야하는 서버와 달리 고객의 기기는 네트워크에 항상 연결되어있지 않아도 된다는 점으로 인해 상대적으로 데이터 침해공격으로부터 더 안전하다. 고객의 기기를 활용하여 서비스 운영 컴포넌트의 일부를 격리시키는 것은 잠재적인 위협요소를 줄이는데 효과적이다. 따라서 본 연구에서는 cloud 모델을 운영하기 위한 아키텍처를 설계하는데에 고객 기기를 활용하여 그 장점을 극대화한다. 클라우드 인프라는 믿을 수 있다는 전제 하에 클라우드 서비스 침해 위험을 줄이기 위한 클라우드 플랫폼 아키텍처를 고안한다.

본 논문이 기술적으로 기여하는 바는 다음과 같다.

- ◉ 데이터 프로세서가 데이터 주체의 데이터를 활용하도록 하면서도 데이터 주체에게 더 높은 수준의 데이터 통제권을 부여하는 클라우드 모델을 제시한다.
- ◉ 클라우드-디바이스 협력적 암호화와 컴퓨팅 및 네트워킹 리소스 격리를 사용하여 데이터 침해를 줄이는 클라우드 플랫폼 아키텍처를 제시한다.
- ◉ 제안하는 클라우드 아키텍처를 오픈스택상에

구현하여 그 성능을 측정하고 보안성을 평가한다.

다음 장에서는 마이데이터 서비스의 배경과 현존하는 마이데이터 서비스 모델, 그리고 관련 연구에 대해 설명한다. 제3장에서는 본 연구에서 제안하는 아키텍처의 개요를 설명한다. 제4장에는 개요를 바탕으로 구현된 아키텍처의 내부 기능에 대해 상세히 설명하며, 제5장에서 성능 및 보안성을 평가하고 제6장에서 본 논문을 마무리짓는다.

II. 배경지식

2.1 마이데이터

정보기술이 발전함에 따라 데이터 역시 대량으로 생성되어 이는 개인 맞춤형 서비스 제공을 위한 중요한 자원으로 사용되고 있다. 데이터의 가치가 지속적으로 증가함에 따라 데이터 주체가 자신의 데이터를 안전하게 관리하고 능동적으로 활용할 수 있도록 하는 데이터 주권이 더욱 강조되고있다. 세계 각국에서 정부는 데이터 주체가 자신의 데이터에 대한 통제권을 행사하고, 이를 바탕으로 맞춤형 서비스를 제공받을 수 있도록하는 정책을 시행하고 있다.

이를 위한 첫 번째 시도는 금융 분야에서 시도되었으며, 그 결과 오픈뱅킹 시스템이 등장하였다[14]. 오픈 뱅킹은 은행과 제3자 서비스 제공업체 간에 금융 데이터를 안전하게 공유하는 체계이다. 제3자 서비스 제공업체는 고객의 계좌, 거래내역, 대출잔액 등 공유된 데이터를 활용해 참여금융기관과 고객의 데이터를 통합해 유용한 마케팅 전략을 세울 수 있다.

오픈뱅킹의 개념은 2018년 영국에서 처음 도입되어 GDPR과 결제 서비스 지침(PSD2)의 시행으로 활성화 되었다[15]. 미국에서는 JP Morgan, BNY Mellon과 같은 금융 기관이 이 개념을 적극적으로 도입하였다[16, 17].

그 이후로도 데이터 주체가 자신의 데이터를 더 적극적으로 통제하고 활용할 수 있도록 하기 위한 시도가 계속되었고 그 결과로 마이데이터 개념이 생겨났다. 마이데이터는 오픈 뱅킹과 유사한 데이터 관리 개념이지만, 금융 분야 외에도 헬스케어, 교육 등 다양한 분야까지도 다룬다는 점에서 더 높은 범용성을 제공한다.

영국에서는 2011년 MiData initiative가 시작되어 Visa나 Google[18]과 같은 기업이 보유한 자신의 데이터에 고객들이 접근할 수 있도록 하였으며 미국에서는 정보 주체가 자신의 건강, 에너지, 교육 관련 데이터를 다운로드할 수 있는 스마트 공시 서비스가 시행되고 있다[19]. 한국 역시 2020년에 마이데이터 개념을 도입하여 다양한 분야에서 데이터 주체의 데이터 통제권을 강화하기 위한 법적 기반을 마련하고 있다.

2.2 기존 마이데이터 서비스 운영 모델

개인 데이터 수집 및 처리를 위한 마이데이터 서비스의 구현 모델은 크게 세 가지로 나뉘며, Fig.1에서 각 모델을 설명한다. 마이데이터 서비스는 컨트롤러와 프로세서의 두 가지 구성 요소로 이루어져 있다. 컨트롤러는 리소스 서버에서 데이터 스토리지로 데이터를 로드하고 각 데이터의 목적 및 용도를 정의한 후 데이터를 필요로 하는 곳으로 데이터를 전달한다. 프로세서는 컨트롤러로부터 전달받은 데이터를 사용해 이를 분석하여 의미있는 결과를 생성한다.

고객의 동의를 받아, 마이데이터 서비스상에 수집된 데이터를 제3자 서비스에 판매하여, 제3자 서비스에 존재하는 프로세서에서도 데이터가 이용될 수 있다.

데이터 스토리지 및 제3자 프로세서의 위치는 Fig.1에 설명된 대로 구현 모델에 따라 상이하며, 이에 따라 각 모델상에서의 마이데이터 서비스 동작 원리 및 보안성 역시 상이하다. Fig.2에서 각 모델에서의 데이터 흐름을 표현한다.

2.2.1 operator 모델

operator 모델에서, 고객들의 개인 데이터는 마이데이터 서비스의 내부 서버에 저장되며, 대부분의 마이데이터 서비스들이 이와 같은 모델을 기반으로 운영된다. 이 모델은 서비스 제공자 입장에서 데이터에 쉽게 접근하고 분석할 수 있어 편리하다. 해당 모델을 사용하여 운영되고 있는 마이데이터 서비스의 대표적인 예로는 Mint, Google Fit, Banksalad 등이 있다[21]-[23].

하지만 operator 모델은 제3자 데이터 프로세서가 마이데이터 서비스 외부에서 독립적으로 연산을 수행함에 따라, 고객의 입장에서 자신의 데이터가 제

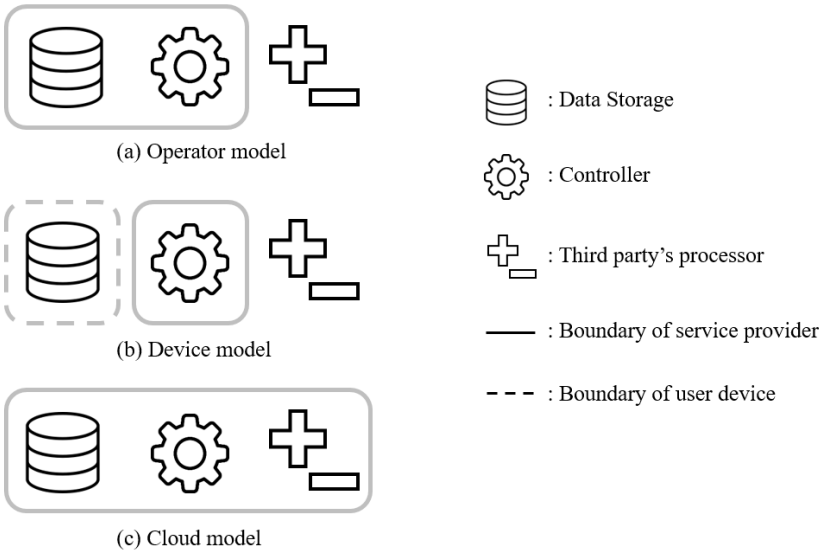


Fig. 1. Classification of MyData implementation.

3자 데이터 처리자에게 전송된 이후 데이터에 대한 통제권을 유지하는 것이 어렵다. 많은 법률과 규정에서 데이터 수집자와 처리자가 고객의 요청을 적절히 처리하도록 의무화하고 있지만, 아직 실제로는 완전히 이행되는데에 한계가 있다[24]. 또한 데이터 수집자/처리자가 증가할 수록 고객은 현재 자신의 데이터가 어떤 수집자/처리자에 의해 무슨 목적으로 이용되고 있는지 추적하는 것은 불가능해진다.

operator 모델은 데이터 컨트롤러가 고객 데이터를 수집해 이를 모두 서비스 내부 서버에 저장해두므로 세 가지 모델 중 데이터 유출 위험도가 가장 높다. 만약 컨트롤러가 침해될 경우 모든 고객의 정보가 위험해진다.

2.2.2 device 모델

device 모델은 operator 모델의 한계를 다루기 위해 고안된 모델이다. device 모델은 사내 서버에 데이터 스토리지를 두어 이 곳에 모든 고객의 데이터를 집중적으로 저장해두는 operator 모델과 달리, 데이터 스토리지를 사용자 기기에 둔다. 각 고객의 정보가 컨트롤러에 의해 수집되고, 각 사용자 기기로 전송되어 저장된다. 대표적인 서비스 운영 사례로 Digi-me[11], my:D[12], Meeco[13] 등이 있다.

device 모델은 데이터가 고객의 기기에 저장되므로 고객이 자신의 데이터를 손쉽게 통제할 수 있다는 점에서 operator 모델의 한계를 보완한다. 하지만 device 모델 역시 operator 모델과 마찬가지로 제3자 데이터 프로세서가 회사 서버 외부에서 독립적으로 동작하기 때문에, 데이터가 제3자의 프로세서로 전송된 후에는 어떤 프로세서가 어떤 데이터를 무슨 목적으로 수집 및 보관하는지를 추적하는 것은 여전히 고객, 즉 데이터 주체의 입장에서 어렵다.

이처럼 device 모델은 사용자 디바이스에 데이터 스토리지를 배치함으로써 데이터에 대한 사용자의 통제권을 강화하고 사내 서버가 침해되더라도 데이터는 사용자 디바이스에 저장되어 있기 때문에 데이터 유출 위험을 완화한다. 하지만 결론적으로 operator 모델과 device 모델 모두 데이터가 제3자 프로세서로 전송될 때 데이터 주체가 완전한 통제권을 행사할 수 없다는 한계가 있다.

2.2.3 cloud 모델

이 연구에서 제안하는 cloud 모델은 데이터 스토리지가 서비스 내부 서버에 위치한다는 점에서 operator 모델과 같다. 따라서 데이터 침해 위험수준이 operator와 동일한 수준으로 머무를 수 있지만, cloud 모델은 데이터 스토리지에 다음과 같은 데이터 보안을 위한 조치를 취함으로써 한계를 극복한다.

cloud 모델에서 데이터 스토리지 상의 데이터는 항상 암호화된 상태를 유지한다. 이 때 데이터 암호화 및 복호화는 사용자 기기에서 생성한 비대칭 키를 사용한다. 비대칭 키 중 공개키를 사용하여 데이터를 암호화하며, 이 암호화된 데이터는 데이터 주체의 기기에 저장된 비밀키를 통해서만 복호화될 수 있다. 따라서 데이터 스토리지에 저장된 데이터는 사용자 기기에 저장된 비밀키가 유출되지 않는 한 항상 암호화된 상태를 유지하여 데이터 침해로부터 안전하며, 이 때의 데이터 침해 위험 수준은 device 모델과 같다.

operator 모델과 device 모델 모두 서비스 외부의 제3자 프로세서로 데이터를 전송하지만, cloud 모델은 클라우드 내에서 제3자 프로세서의 연산을 호스팅하고 실행한 결괏값만 반환한다. 즉 분석을 위한 데이터가 클라우드 밖으로 나가는 대신 분석 연산이 클라우드 내로 전송된다. 이로써, 제3자 프로세서로 전송되는 데이터에 대해서는 cloud 모델이 기존의 operator 모델과 device 모델보다 더 높은 수준의 통제권을 제공한다.

이처럼 기존의 마이데이터 서비스 모델이 데이터 스토리지의 위치에 따라 operator 모델과 device 모델로 나뉘었다면, 본 연구에서 제안하는 cloud 모델까지 고려하였을 때 데이터 스토리지의 위치 뿐만 아니라 제3자 프로세서의 연산이 수행되는 위치까지 분류 기준에 포함되면서 세 모델이 서로 구분된다. operator 모델은 데이터 스토리지가 사내 서버에 존재하지만 제3자 프로세서의 연산은 서비스 외부에서 수행되며 device 모델은 데이터 스토리지가 사용자 기기에 위치하면서 제3자 프로세서 연산이 서비스 외부에서 수행된다. 마지막으로 cloud 모델은 데이터 스토리지가 사내 서버인 클라우드 상에 존재하고, 제3자 프로세서 연산 역시 클라우드 상에서 수행된다.

III. 관련 연구

클라우드 모델은 데이터 스토리지를 서비스 자체 내부 서버상에 두므로 데이터 스토리지를 보호하는 것이 매우 중요하다. 데이터를 보호하는 방법에는 크게 두 가지 방법이 있다. 한 가지는 데이터 자체를 침입으로부터 보호하는 것이고, 다른 한 가지는 침해가 발생했을 때 그 피해를 최소화하는 것이다. 이와 관련한 기술 및 연구는 다음과 같다.

3.1 데이터 스토리지 보호

데이터 자체가 노출되지 않도록 보호하기 위하여, 데이터 암호화, 데이터 접근제어, 모니터링 등의 기술을 적용할 수 있다.

Always-encrypted와 투명 데이터 암호화(TDE)같은 데이터 암호화 기술을 사용하여 민감한 데이터를 보호할 수 있다. Always-encrypted[25]는 암호화된 데이터와 암호화 키가 데이터베이스 서

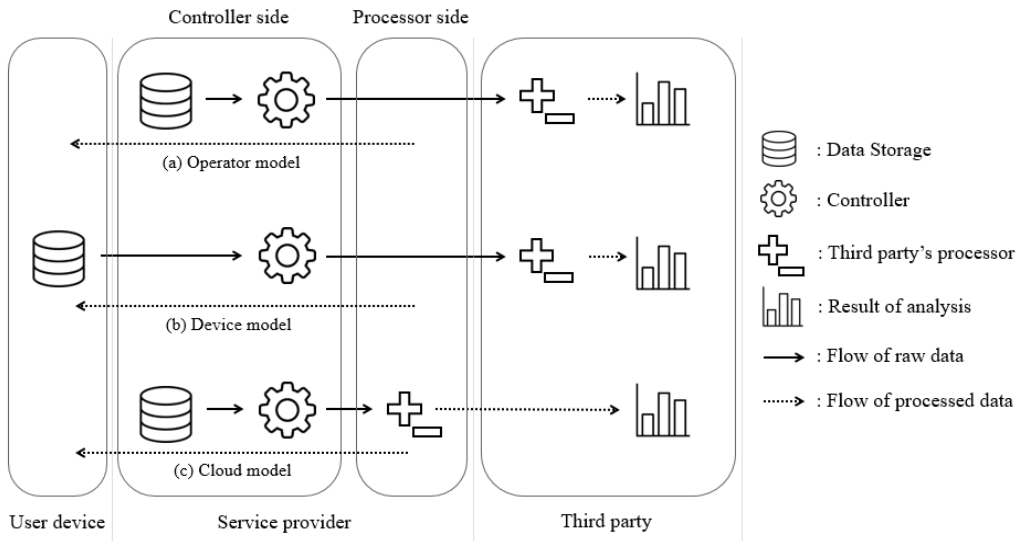


Fig. 2. Data flow of MyData models.

버로부터 절대 유출되지 않음을 보장하는 클라이언트 측 암호화 기술이다. 이를 통해 데이터베이스 엔진은 암호화된 데이터를 처리하면서도 실제 데이터 값을 조회할 수 없다. Always-encrypted 기술을 사용하여 사용자는 중요한 데이터를 클라우드에 안전하게 저장하고 악의적인 내부자에 의한 데이터 도난 피해를 줄일 수 있다.

TDE는 서버 측 암호화 기술로, 데이터가 저장 시에는 암호화된 상태로 존재하다가 데이터에 액세스할 시 자동으로 복호화하는, 끈김없는 보안성(seamless security)을 제공한다. 이 기술을 사용할 경우, 기존 애플리케이션을 수정할 필요 없이 확장성 있게 통합하여 사용이 가능하다. TDE는 2008년부터 Microsoft SQL Server에서 지원되고 있으며 Oracle, IBM, PostgreSQL 등에서도 지원하고 있다. TDE와 같이 애플리케이션의 수정이 없이 암호화를 투명하게 적용하기 위한 연구는 TDE 제품이 상용화되기 전부터 진행되어왔다. U. T. Mattson[26]은 데이터 필드 타입이나 길이를 변경할 필요가 없는, 컬럼 수준의 데이터베이스 암호화를 제시하였다. TDE 제품이 상용화되고 난 이후로는, 이를 개선하기 위한 추가적인 노력이 계속되었다. 기존의 TDE 기술은 저장된 데이터(data at rest)만 암호화할 수 있었지만, V. Sidorov 등[27]은 사용중인 데이터(data in use)도 암호화할 수 있는 투명한 데이터 암호화 방법을 제안한다.

접근 제어는 허용되지 않은 접근을 차단하고 데이터 세트 별로 서로 다른 접근 정책을 적용함으로써 데이터 유출을 방지한다. Kumar 등[28]은 보안 수준이 낮은 시스템으로 정보가 유출되는 것을 방지하기 위해 벨-라파둘라 모델(BLP)을 사용하는 방안을 제시한다. Yang 등[29]은 블록체인을 사용하여 중앙에서 접근을 통제하고 데이터가 승인되지 않은 사용자에 의해 데이터가 변조되는 방지하는 기술을 제안한다. Lounis 등[30]은 접근 정책의 논리 속성기반 암호화(ABE)를 사용한 세분화된 접근 제어를 제안하여, 접근 정책의 논리적 표현을 사용하여 접근 구조와 함께 데이터를 암호화한다.

데이터 손실 방지(DLP)는 다양한 채널에 걸친 데이터 흐름을 기록하고 통제하는 기술로, 데이터 보호를 위한 모니터링에 적용될 수 있다. DLP는 클라우드 기반 데이터 스토리지 및 서비스형 소프트웨어(SaaS) 애플리케이션으로 확장하여 적용할 수 있다. Cloud DLP 솔루션은 클라우드 조직 내에서 승인되지 않은 민감하고 중요한 데이터의 흐름을 식별하고 차단해낸다.

Y. J. Ong 등[31]은 머신러닝을 활용하여 민감한 정보를 탐지하는 문맥 기반의 DLP 시스템을 구축하였다. 이 시스템은 민감도 분석을 통해 데이터 모니터링을 수행한다. P. Han[32] 등도 머신러닝을 활용하여 민감한 데이터에 대해 안전성 검사를 자동으로 수행하는 인터넷 게이트웨이인 클라우드 DLP 기술을 제안한다.

3.2 데이터 유출 피해 최소화

데이터가 이미 탈취된 상황에서도 데이터 저장 위치를 분산하여 저장하거나, 데이터 단편화(fragmentation)를 통해 데이터 유출로 인한 피해를 최소화할 수 있다.

Kumar 등[33]은 클라우드 스토리지 서버를 개인 데이터와 공유 데이터의 두 섹션으로 나눈다. 사용자는 특정 사용자만 액세스할 수 있는 개인 데이터 섹션에 개인 데이터를 저장할 수 있으며, 공유 데이터 섹션은 신뢰할 수 있는 사용자 간에 공유해야하는 데이터를 저장하는 데 사용된다.

또한 데이터 단편화를 통해, 데이터가 유출되더라도 그 일부만 유출되도록 하여 전체 데이터 손실을 방지할 수 있다[34]. 데이터 단편화 외에도 전송 전 민감한 데이터 탐지 및 제거와 같은 다른 방식의 전처리를 적용할 수 있다[35].

3.3 컨피덴셜 서버리스 컴퓨팅

본 연구에서 제안하는 cloud 모델에서는 제3자 프로세서의 연산을 클라우드에서 호스팅하고 결괏값만 반환한다. 이와 같이 연산 아웃소싱 형태로 운영되는 서비스가 증가함에 따라, 특히 퍼블릭 클라우드에서 서버리스 컴퓨팅의 기밀성 및 신뢰성을 확보하고자 하는 노력이 계속되고있다[36]. W. Qiang[37]은 SGX enclave를 통해 API 게이트웨이를 보호하고, 이중 샌드박스 접근방식을 통해 서비스 런타임을 보호하는 새로운 서버리스 컴퓨팅 프레임워크인 Se-Lambda를 소개하며, SGX enclave를 WebAssembly 샌드박스 환경과 통합한다. F. Alder[38]는 강력한 보안과 책임성을 제공하기 위해 Intel SGX를 사용한 서비스형 기능(FaaS)을 구현한 S-FaaS를 제안한다.

D-Auth[39]는 서버 기반 OTP와 토큰 인증을 통합하여 D-Auth 토큰을 생성함으로써 서버리스 컴퓨팅을 위한 인증 및 권한 부여를 제공한다.

IV. 설계목표 기반 보안 조치

이번 장에서는 설계 목표 및 위협 모델을 설정하여, cloud 모델을 안전하게 동작시키기 위한 아키텍쳐를 도출한다.

4.1 설계 목표

제안하는 아키텍처의 설계 목표는 데이터 스토리지, 컨트롤러, 프로세서를 하나의 클라우드상에 배치함으로 인해 발생할 수 있는 데이터 유출 위험을 줄이는 것이다. cloud 모델은 operator 모델과 같이 데이터 스토리지를 서버상에 두고 있다는 점에서 컨트롤러가 침해될 경우 모든 고객의 데이터가 위험에 취할 수 있다. 따라서 제안하는 아키텍쳐는 cloud 모델의 서버에 집중된 데이터를, 클라우드상에서 호스팅되는 제3자의 프로세서 연산이나 외부에서 발생하는 데이터 침해 공격으로부터 안전하게 보호하는 것을 목표로 한다.

4.2 위협 모델

위협 모델은 다음과 같다: 공격자는 제3자 프로세서 연산 혹은 마이데이터 서비스 제공자에 의해 제공되는 서비스를 포함한 가상머신 상의 모든 서비스들을 공격 대상으로 삼을 수 있다. 클라우드 외부로부터, 서버 상의 데이터 스토리지를 노린 침해 공격이 시도될 수 있으며 내부에서도 데이터 분석용 연산에 악의적인 코드를 주입하거나 데이터 스토리지에 승인되지 않은 접근을 시도하는 등의 공격이 가능하다. 다만, 본 연구에서는 하이퍼바이저, 방화벽, 네트워크 스위치와 같은 인프라는 신뢰할 수 있다고 가정한다. 클라우드 아키텍쳐는 크게 물리 계층과 가상화 계층으로 구분된다[40]. 클라우드 환경 구축의 물리적 자원이 되는 서버, 스토리지, 네트워크와 같은 물리 인프라가 물리적 계층을 이루며, 물리적 계층 상에 가상화 계층이 존재한다. 가상화 계층은 다시, 하이퍼바이저와 가상 네트워크, 가상 머신 등을 포함하는 가상 인프라 계층과 가상 머신 상에서 동작하는 소프트웨어들을 포함하는 가상 어플리케이션 계층으로 나뉜다. 본 연구는 물리 인프라 계층 및 가상 인프라 계층은 신뢰할 수 있다고 가정한다. 공격자는 하이퍼바이저의 격리 메커니즘을 방해하여 다른 가상머신을 침해할 수 없다.

◉ 공격 가능한 대상 : 가상머신 상의 서비스

공격자는 가상머신에서 실행 중인 서비스를 침해하기 위해 잘못 설정된 가상머신 또는 게스트 운영체제의 취약점을 이용하거나, 악성 코드를 배포할 수 있다.

◉ 공격 불가능한 대상 : 물리 및 가상 인프라

하이퍼바이저와 내부 네트워크를 포함한 인프라는 신뢰할 수 있다. 하이퍼바이저가 물리적인 자원 격리를 제공하며 가상 네트워크는 트래픽 격리를 제공하는데, 이는 애플리케이션 수준의 공격이 인프라로 확산되어 다른 VM이나 물리 자원을 침해하는 것을 방지한다.

이처럼 클라우드 아키텍처의 계층을 물리 인프라, 가상 인프라, 가상 어플리케이션 계층으로 분류할 때, 본 연구는 인프라는 신뢰할 수 있지만 가상머신 상에서 동작하는 어플리케이션 서비스는 신뢰할 수 없다고 가정하여 데이터 유출 위험을 줄이기 위한 플랫폼 설계에 중점을 둔다. 데이터 위협 탐지는 본 연구의 연구 범위 밖으로, 이에 관해서는 기존의 기술을 채택한다고 가정한다.

4.3 클라우드-디바이스 간 협력적 암호화

위와 같은 위협 모델을 전제로 할 때, 데이터 유출로 인한 잠재적 손상을 최소화할 수 있는 가능한 접근 방식으로써 마이데이터 서비스 고객의 디바이스를 활용하여 always-encrypted 데이터 스토리지를 구축할 수 있다.

device 모델에서, 고객의 디바이스는 서버와 달리 항상 네트워크에 연결되어있지 않아도 되며, 모든 리퀘스트에 응답할 필요가 없어 서버보다 더 안전하다고 간주된다. cloud 모델에서도 마찬가지로, 이러한 디바이스의 장점을 살려 이를 활용한 클라우드와 디바이스간 협력적 접근방식을 고안할 수 있다. 고객 디바이스에 데이터 스토리지를 두는 device 모델과 달리, cloud 모델에서 제안하는 아키텍처는 데이터 스토리지를 서버에 두면서도, 암호화 된 데이터를 복호화할 수 있는 복호화 키는 고객 디바이스에 두어

데이터가 침해되더라도 암호화된 데이터로부터 고객의 개인정보가 유출되는 것을 막는다.

세부적인 동작 원리는 다음과 같다. 컨트롤러가 고객의 데이터를 수집할 때, 해당 고객의 디바이스에서 생성된 비대칭 키페어 중 공개키로 암호화된 채로 데이터 스토리지에 저장된다. 복호화에 사용되는 비밀키는 고객 디바이스에 저장되어, 고객의 동의가 있을 때에만 데이터가 복호화될 수 있다. 프로세서가 데이터 분석을 위해 데이터 주체에게 데이터를 요청할 경우, 고객의 동의 하에 암호화된 데이터가 기기에서 클라우드 상의 서버로 전송된다.

4.4 컴퓨팅 및 네트워킹 자원 격리

컴퓨팅 및 네트워킹 자원을 격리하는 것 역시 가상머신상에서 동작하는, 사내 서버용 데이터 스토리지로의 비정상적인 접근을 막는 방안이 된다.

컴퓨팅 리소스를 격리하는 것은 가상화 기술을 활용하여 각 프로세서를 서로 다른 가상머신상에 두어 독립된 작동을 보장할 수 있다. 각 가상머신은 논리적으로 격리되어 있어 다른 가상머신에 영향을 주지 않고 작동한다.

동시에, 네트워킹 리소스 격리는 특정 가상머신이 클라우드 환경 외부로 트래픽을 전송하거나 가상머신으로 접근하지 못하도록 제한한다. 이는 내부 네트워크에서만 통신이 가능하도록 제한하고, 가상머신을 프라이빗 서브넷에 배치하여 서버의 데이터 스토리지로부터 격리하는 방식으로 구현될 수 있다. 네트워크 보안 그룹과 방화벽 설정 등의 네트워크 보안 조치는 외부 접근을 차단하고 가상머신이 서버의 데이터 스토리지에 접근하는 것을 차단한다.

종합적으로 살펴보았을 때, 본 논문에서는 각 프로세서를 서로 다른 가상머신에 배치하고 프라이빗 서브넷에 둠으로써 외부 인터넷으로부터 격리하는 것과, always-encrypted 데이터 스토리지를 구축하여 암호화된 데이터를 복호화할 수 있는 키를 고객 디바이스에 저장하도록 구현하는 것을 데이터 유출 위험을 완화하는 해결책으로 제시한다. 이를 통해 데이터 스토리지, 컨트롤러, 프로세서가 하나의 클라우드 환경 내에 집중됨으로써 발생할 수 있는 데이터 유출 위험을 효과적으로 줄일 수 있다.

V. 제안 아키텍쳐: MyData Cloud

이 장에서는 개요에서 고안한 기술들을 바탕으로 구현한 아키텍쳐의 구조적 특징과 내부 기능을 소개한다.

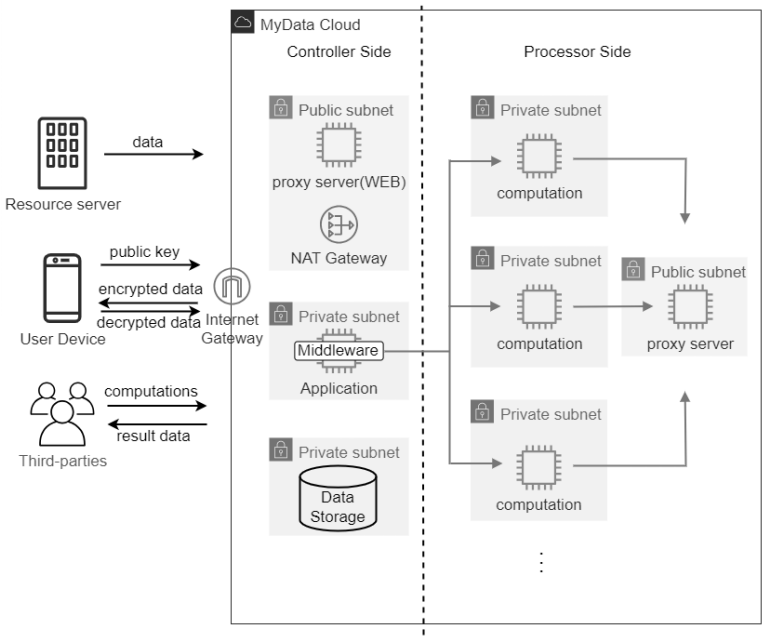


Fig. 3. The proposed cloud architecture.

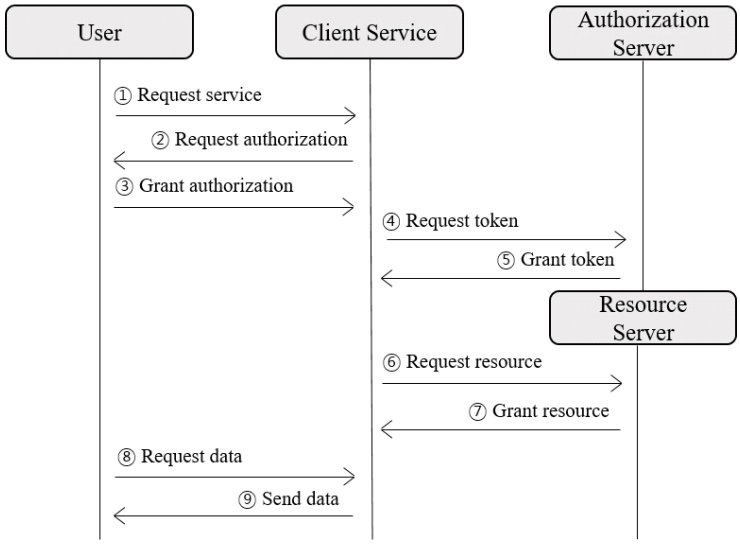


Fig. 4. The flow of OAuth 2.0 protocol.

5.1 아키텍쳐 구성요소

제안하는 아키텍쳐는 Fig.3과 같이 컨트롤러 측과 프로세서 측으로 나뉜다. 컨트롤러 측은 데이터 수집을 담당하며 프로세서 측은 데이터를 연산하여 유의미한 분석 값을 도출해낸다. 컨트롤러 측은 프록시서버, 웹서버, 웹 애플리케이션 서버, 미들웨어, 그리고 데이터베이스 서버로 구성된다. 프록시 서버는 퍼블릭 서브넷에 배치되어 데이터 주체 및 OAuth2.0 프로토콜에 참여하는 각종 리소스 서버, 그리고 제3자 서비스들과 통신한다. 나머지 웹서버, 웹 애플리케이션 서버, 그리고 데이터베이스 서버는 프라이빗 서브넷에 배치되어 마이데이터 서비스 운영을 위한 기본적인 3계층 구조를 이룬다. 추가적으로 미들웨어가 웹 애플리케이션 서버상에 구현되어 데이터를 암호화하고 프로세서 측으로 데이터를 전송하는 기능을 수행한다.

5.2 동작 흐름

제안하는 아키텍쳐에서의 cloud 모델은 다음과 같이 운영된다. 이 때 컴포넌트간 전송되는 데이터는 암호화 통신 프로토콜을 통해 전송됨을 가정한다.

사용자 등록. 데이터 주체는 MyData Cloud 서비스에 가입한다. 데이터 주체, 즉 고객은 자신의 디바이스로 서비스에 접속해 동의 여부를 미리 설정할 수 있으며, 미리 설정된 동의는 언제든지 변경될 수 있다. 고객은 MyData Cloud 서비스에 다른 서비스에 흩어져있는 본인의 데이터를 수집해줄 것을 요청할 수 있다. 예를 들어, 고객은 다른 금융 기관의 거래 내역, 병원 의료 기록, 그리고 정부가 관리하는 개인 기록 등을 마이데이터 서비스상에서 한곳에 모아 확인할 수 있다.

프로세서 등록. MyData Cloud 서비스는 자체 데이터 프로세서를 운영할 수도 있으며 또는 제3자 프로세서의 연산을 호스팅하여 데이터를 분석할 수 있다. 어떤 경우라도, 분석 연산은 클라우드 플랫폼상에서 호스팅되어야한다. 프로세서 측에 연산들이 호스팅될 때, 분석을 위한 데이터가 필요함에 따라 컨트롤러로부터 기존 데이터를 요청할 수 있다. 이 경우 컨트롤러는 암호화된 데이터를 고객에게 보내고, 고객의 동의 하에 데이터가 복호화되어 프로세서 측으로 전송된다.

데이터 수집. 개인 정보 수집을 위한 인증 절차에서 다양한 프로토콜이 사용될 수 있다. 정보주체의 개인신용정보 전송요구를 위한 인증 API의 대표적인 예로 OAuth2.0가 있으며 그 흐름이 Fig.4에 서술되어있다. 그림에 서술된 OAuth2.0는 다음와 같은 절차를 따른다.

사용자가 MyData Cloud 서비스 측에 데이터 수집을 요청한다. 이와 동시에 사용자 기기에서 비대칭키를 생성하며 공개키는 MyData Cloud 측으로 전달된다. Fig.4의 'Client Service'에 해당하는 MyData Cloud 서비스는 사용자에게 인가를 요청

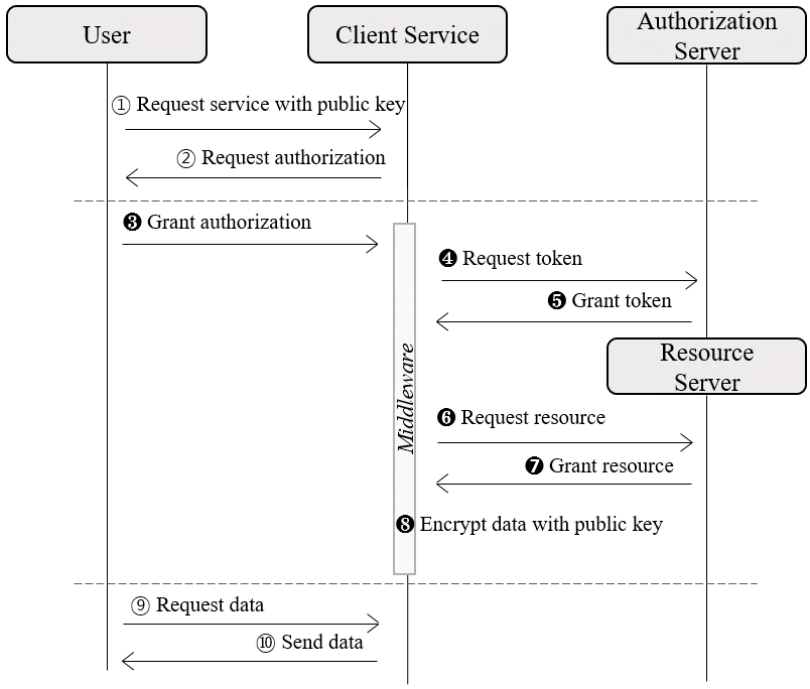


Fig. 5. Data collection with OAuth 2.0 protocol on the proposed platform

한다. 사용자가 인가하면, MyData Cloud 서비스는 사용자 자격 증명을 받아 이를 활용해 사용자가 로드하고자 하는 데이터를 소유하고 있는 서비스의 인가 서버에 토큰을 요청한다. 인가서버는 자격 증명을 확인한 후 토큰을 반환하고, MyData Cloud 서비스는 이 토큰을 사용해 리소스 서버에서 데이터를 얻는다. 이 데이터는 앞서 사용자 기기로부터 전달받은 공개키로 암호화되어 데이터 스토리지에 저장된다.

데이터 조회. 데이터 주체, 즉 사용자는 컨트롤러 측에 수집되어 데이터 스토리지에 저장되어있는 자신의 데이터를 조회할 수 있다. 사용자의 요청이 있을 시, 컨트롤러는 요청된 데이터를 암호화된 채로 사용자 기기로 전송한다. 사용자 기기는 기기에 저장된 복호화용 비밀키를 사용하여 이를 복호화하고 해당 내용을 화면에 디스플레이한다.

데이터 처리. 컨트롤러 측에서 위와 같은 과정을 통해 데이터를 수집하면, 프로세서 측에서는 데이터 주체의 동의 하에 복호화 된 데이터를 전달받아 미리 등록된 분석 연산들을 수행한다. 이 때 이 분석 연산은 MyData Cloud 서비스 자체에서 운영하는 연산, 혹은 제3자 프로세서가 등록해 둔 분석 연산이다. 분석 결괏값은 프록시 서버로 전송되고, 프록시 서버가 다시 이 값을 사용자 기기, 혹은 연산을 의뢰한 제3자 프로세서 등 해당 값을 필요로 하는 위치로 전송한다. 이 때 제 3자 프로세서의 분석 결괏값은 개인정보 보호법 및 관련 규정에 따라 비식별화되어야한다. 개인식별 요소 삭제, 가명처리, 총계처리, 랜덤화, 데이터 삭제, 데이터 범주화, 데이터 마스킹 등의 비식별 조치를 취한 후 제 3자 프로세서로 전달된다.

데이터 폐기. 컨트롤러 측에서의 프로세서 연산이 완료되면, 해당 프로세서 연산을 위해 할당되었던 컴퓨팅 자원은 해제된다. 이에 따라, 연산 시 활용되었던 데이터가 동시에 폐기된다.

5.3 미들웨어

앞서 설명한 4.2장에서, 컨트롤러는 세 가지 역할을 수행한다: (1)데이터 주체의 요청이 있을 시 데이터를 수집하고, (2)수집한 데이터를 암호화하여 데이터 스토리지에 저장하며 (3)분석에 필요한 데이터를 프로세서 측으로 전송한다.

이 중 암호화 작업(2)과 데이터 전송(3) 작업을 위해, 제안하는 아키텍쳐는 컨트롤러 측에 미들웨어를 두어 해당 작업을 수행하도록 한다.

데이터 암호화. 미들웨어는 사용자 기기에서 생성된 공개키를 사용하여 데이터를 암호화하고, 이 암호화된 데이터를 컨트롤러의 데이터 스토리지에 저장한다. 데이터 암호화를 위해, 사용자 기기는 비대칭키를 생성하여 컨트롤러 측으로 공개키를 전달하고, 비밀키는 기기에 저장한다. 따라서 컨트롤러의 데이터 스토리지에 암호화된 채로 저장된 데이터는 데이터 주체의 동의 없이 복호화될 수 없다.

암호화된 데이터는 이후 사용자 기기에 디스플레이 되거나, 프로세서가 연산을 위해 요청할 경우 이에 따라 전송될 수 있다. 이 때 사용자 기기에 저장된 비밀키로 복호화된다. 비밀키는 사용자 기기에 저장되어있으며 절대 기기를 떠나지 않는다. 따라서 복호화를 위해 비밀키를 클라우드로 전송하지 않고 암호화된 데이터가 비밀키가 저장되어있는 사용자 기기로 전송된다.

데이터 전송. 미들웨어는 컨트롤러가 수집한 데이터를 프로세서 측으로 전송하는 역할 역시 수행한다. 프로세서 측의 각 분석 엔진마다 연산에 사용하는 데이터가 다르므로, 미들웨어는 각 엔진에 필요한 데이터를 식별하여 전송해준다. 추가적으로 데이터를 전송하기 전, 미들웨어는 사용자의 약관 동의 여부를 확인한다. 프로세서 측 분석 엔진이 증가할 경우 데이터 주체로부터 매번 데이터 전송을 승인받는 것이 어려울 수 있으므로, 제안하는 아키텍처는 사용자가 데이터 전송에 대한 동의를 미리 설정할 수 있게 한다. 데이터 수집이 데이터 주체에 의해 승인되면, 미들웨어는 자동으로 데이터 전송을 위한 미리 설정된 동의를 확인하고 필요한 경우에만 승인을 요청한다 (예: 동의 만료, 새로운 분석엔진 등록, 또는 정책 변경).

5.4 격리된 컴퓨팅 및 네트워킹 자원 구성

cloud 모델이 동작하는 클라우드 플랫폼은, 프로세서 측의 연산들을 격리함으로써 연산들 중 일부가 침해됨으로 인해 연쇄적으로 발생할 수 있는 위협을 방지해야한다. 연산들 간의 격리를 위해, 제안된 아키텍처는 데이터 처리 연산을 서로 다른 가상머신에 둠으로써 격리한다. 추가적으로, 모든 프록시 서버를 제외한 모든 가상머신은 프라이빗 서브넷에 배치되어 망분리되고, 네트워크 흐름이 통제된다. 이를 통해 프로세서 측의 모든 가상머신은 외부 인터넷과의 통신이 불가능하여, 클라우드 밖으로 데이터를 유출하는 것을 방지할 수 있다.

VI. 성능 분석 및 보안성 평가

이 장에서는 제안된 아키텍처의 성능과 보안수준을 평가한다.

6.1 성능 평가

성능 평가의 목적은 본 연구에서 제안하는 아키텍처에서 동작하는 컨트롤러의 오버헤드를 분석하는 것이다. 주요단계별로 소요되는 연산 수행시간을 측정하고 그 결과를 대조군의 수행시간과 비교한다. 마이데이터 서비스에서 컨트롤러는 OAuth2.0 프로토콜에 따라 리소스 서버에서 데이터를 수집하여 정보를 사용자 기기에 디스플레이 한다. 이처럼 제안된 아키텍처의 컨트롤러가 수행하는 OAuth2.0 절차는, 대조군이 수행하는 기본적인 OAuth2.0 단계에 더해 명확히 구분되는 추가적인 단계들을 수행하므로 이 추가적인 단계들을 바탕으로 대조군과 제안된 아키텍처 간의 성능 비교가 가능하다.

6.1.1 대조군 및 측정 범위 설정

성능 비교를 위해 네트워크 격리를 위한 프록시와 데이터 암호화를 위한 미들웨어가 존재하지 않는 기본적인 아키텍처를 대조군으로 설정한다. 대조군과 제안된 아키텍처 모두 데이터 수집을 위해 OAuth2.0 프로토콜을 사용한다. 두 아키텍처 모두 Fig.4에 표시된 것과 동일한 단계를 따르지만, 제안된 아키텍처는 보안성 강화를 위한 추가적인 단계를 거쳐 더 많은 오버헤드가 발생한다. 컨트롤러가 리소스 서버로부터 데이터를 수신하면(Fig.5의 단계 7) 미들웨어는 이를 암호화하여 데이터 스토리지에 저장한다. 사용자가 데이터 조회를 원하는 경우 사용자 기기에서 데이터를 복호화하여 디스플레이한다 (Fig.5의 단계 9).

본 연구의 실험에서 다음 두 연산의 경과 시간을 측정하고 비교한다.

◉ 연산 1: 데이터 수집
- 1-1: 리소스 서버로부터 데이터 수집
- 1-2: 데이터 암호화
- 1-3: 데이터 스토리지에 데이터 저장

◉ 연산 2: 데이터 시각화
- 2-1: 데이터 스토리지에서 요청 데이터 로드
- 2-2: 데이터 복호화
- 2-3: 복호화된 데이터를 기기에 디스플레이

연산 1-1은 Fig.4의 단계 1부터 7까지를 포함한다. 연산 1-2는 Fig.5의 단계 8에 해당하며, 이는 제안된 아키텍쳐에 추가된 미들웨어가 수행하는 연산으로, 대조군에서는 수행되지 않는다. 연산 1-3은 제안된 아키텍쳐와 대조군 모두 동일하게 수행한다.

연산 2-1은 Fig.4의 단계 8부터 9까지를 포함한다. 연산 2-2는 사용자 기기상에서 제안된 아키텍쳐만 수행하며 대조군에서는 수행되지 않는다. 연산 2-3은 제안된 아키텍쳐와 대조군 모두 동일하게 수행한다.

6.1.2 실험 환경

본 연구에서는 제안하는 아키텍쳐와 대조군 아키텍쳐를 동일한 조건으로 구성하여, 오픈소스 클라우드 컴퓨팅 플랫폼인 Openstack[41]을 사용하여 구현하였다. QEMU 하이퍼바이저[42]를 사용하여 머신을 가상화하였으며 호스트 머신과 가상머신 및 실험에 사용된 사용자 기기의 사양을 Table 1에서 설명한다.

호스트 머신은 Dell Power Edge R240이며 CPU는 Intel Xeon E-2224 3.4GHz, 메인 메모리는 16GB, Ubuntu 16.04.7 Server를 운영체제로 둔다. 그 위에 호스팅되는 가상머신은 2vCPU, 와 4GB 메인 메모리를 갖추며, Ubuntu 16.04 LTS 운영체제를 사용한다. 사용자 기기는 Samsung Galaxy S21로 Octa-Core 2.9 GHz CPU와 8GB의 메인메모리 환경에서 Android 운영체제가 동작한다.

Table 1. Specification of machines used for performance evaluation.

Machine	Parameter	Value
Physical	Model	Dell Power Edge R240
	CPU	Intel Xeon E-2224 3.4GHz
	Main Memory	16GB
	Operating System	Ubuntu 16.04.7 Server
Virtual	vCPUs	2vCPUs
	Main Memory	4GB
	Operating System	Ubuntu 16.04 LTS
Device	Model	Samsung Galaxy S21
	CPU	Octa-Core 2.9 GHz
	Main Memory	8GB
	Operating System	Android

6.1.3 실험 결과

수행시간 측정을 위해 두 아키텍쳐에서 OAuth2.0에 따른 데이터 로드 연산을 100회씩 수행하였으며 각 리퀘스트마다 3KB 크기의 데이터를 송신한다. 마이데이터 서비스가 제3자 서비스 제공업체로부터 로드하는 고객의 데이터 크기는 서비스 유형 및 시스템에 따라 달라진다. 이 실험에서는 고객의 금융 데이터를 로드하는 상황을 가정하였으며, 로드하는 데이터 크기는 금융 보안원에서 규정한 금융분야 마이데이터 표준 API 규격[43]에 따라 응답 메시지의 평균 길이인 3KB로 설정하였다. 두 아키텍쳐상에서 연산별 수행시간은 Table 2에 정리되어 있다.

연산 1-1 수행결과, 제안된 아키텍쳐의 수행시간(58.55 ms)이 대조군 대비 (46.05ms) 27.14% 증가하였다. 이는 주로 네트워크 망분리로 인해 발생한 오버헤드로 해석된다. 연산 1-2는 데이터 암호화 연산으로 제안된 아키텍쳐에서만 수행되어, 이로인한 수행시간인 8.05ms가 해당 아키텍쳐에만 추가적으로 발생한다. 연산 1-3의 수행시간은 대조군과 제안된 아키텍쳐 각각 12.00ms, 11.94ms로 서로 거의 동일한 수준이다. 전체적으로 연산 1의 수행시간은 각각 대조군 58.05 ms, 제안된 아키텍쳐 78.54ms

Table 2. Average elapsed time of operations (ms).

Operation	Baseline	Proposed
1-1	46.05	58.55
1-2	-	8.05
1-3	12.00	11.94
Operation 1 total	58.05	78.54
2-1	68.34	76.18
2-2	-	94.69
2-3	0.72	0.89
Operation 2 total	69.06	171.76

로, 제안된 아키텍쳐가 대조군 대비 35.29% 증가하였다.

마찬가지로, 연산 2-1에서 네트워크 망분리로 인해 제안된 아키텍쳐의 수행시간(76.18ms)이 대조군(68.34ms) 대비 11.47% 증가하였다. 연산 2-2는 데이터 복호화 연산으로, 연산 1-2와 같이 제안된 아키텍쳐에서만 수행되어 94.69ms의 추가 수행시간이 발생하였다. 이 때 복호화 연산(연산 2-2)은 모바일 디바이스에서 실행됨에 따라 서버에서의 암호화 연산(연산 1-3)보다 훨씬 더 많은 시간이 소요된다. 연산 2-3에서는 대조군(0.72ms)과 제안된 아키텍쳐(0.89ms)간 수행시간이 거의 동일하였다. 종합적으로 연산 2의 수행시간은 각각 대조군 69.06ms, 제안된 아키텍쳐 171.76ms로 대조군 대비 148.71% 증가하였다.

6.2 보안성 평가

제안하는 아키텍쳐상에서 수집 및 처리되는 데이터에 대한 보안성을 평가한 결과는 다음과 같다. 보안성 평가 기준에는 1)서비스 내부 서버가 침해되었을 경우 데이터 스토리지에 대한 위협 수준, 2)제3자 프로세서가 처리할 데이터에 대한 통제 가능성 등이 포함된다.

먼저, 첫 번째 기준인 '서비스 내부 서버가 침해되었을 경우 데이터 스토리지에 대한 위협 수준'에 대해 Fig.1의 세 모델을 대상으로 평가한다. operator 모델의 경우 데이터 스토리지가 서비스 내부 서버에 위치하기 때문에 서버가 침해되면 모든 고객의 데이터가 손상된다. 데이터가 암호화되어있더라도, 키 역시 서버에서 관리하기 때문에 완벽한 보호가 어렵다. 따라서 데이터 유출 방지가 보장되지 않는다. 반면, device 모델에서는 데이터 스토리지가 사용자 기기에 존재하므로 서버가 침해되더라도 고객 데이터는 영향을 받지 않는다.

cloud 모델은 operator 모델과 마찬가지로 데이터 스토리지가 서비스 내부 서버에 위치하기 때문에 서버가 침해되면 스토리지 역시 위험에 노출된다. 하지만 사용자 기기에서 생성된 공개키로 데이터를 암호화하고, 비밀키는 사용자 기기에만 두고 있으므로 데이터 스토리지가 침해되더라도 내부 데이터가 노출되지 않아 device 모델과 동일한 수준의 데이터 보호가 보장된다.

다음으로, 두 번째 기준인 '제3자 프로세서가 처리할 데이터에 대한 통제 가능성'에 대한 평가는 다음과 같다. operator 모델과 device 모델은 고객의 동의 하에 데이터를 제3자 프로세서에 전송한다. 이 경우 데이터가 제3자에게 전송되면 이후 데이터 추적이 불가능하므로 데이터가 수집 목적 범위 내에서 사용됨을 보장할 수 없다.

반면 cloud 모델에서는 마이데이터 서비스 제공자가 제3자를 대신해 연산을 호스팅하여 분석을 진행하기 때문에 데이터가 클라우드 밖으로 유출되지 않아, 분석되는 데이터에 대한 완벽한 통제가 가능하다. 따라서 보안성을 종합적으로 평가할 때 cloud 모델은 기존의 모델들에 비해 수집 및 처리되는 데이터에 대한 안전성과 통제권을 모두 보장한다.

다만, 본 연구는 cloud 모델 상에서 데이터 주체의 개인정보가 담긴 데이터가 분석을 위해 일시적으로 복호화되는 시점의 데이터 안전성 보장은 다루지 않고 있어, 수집 단계와 처리 단계 사이에 일시적으로 복호화된 데이터의 공격 표면을 줄이기 위한 후속 연구가 필요하다.

VII. 결 론

개인데이터 활용체계의 새로운 패러다임인 마이데이터를 바탕으로, 서비스 제공자는 데이터 주체의 동의 하에 여러 서비스에 걸쳐 흩어져있는 데이터 주체의 데이터를 수집하고 이 데이터를 바탕으로 고객 맞춤화된 서비스를 제공할 수 있다. 더 나아가 데이터 주체는 동의 하에 자신의 데이터를 제3자의 프로세서에 판매하여 데이터 분석에 자신의 데이터를 사용하도록 할 수 있다. 하지만 개인정보가 한 번 제3자의 프로세서에게 판매되어 그들의 프로세서에 의해 처리될 경우 그 순간부터 데이터를 추적하고 통지할 수 없다는 문제가 발생한다.

이러한 문제를 해결하고자 본 연구에서는 고객의 데이터가 클라우드상에만 머무르고, 데이터 수집 및 처리가 동일한 하나의 클라우드 환경상에서 이루어지도록 하여 고객의 데이터 통제권을 강화하고 공격표면을 줄이는 마이데이터 운영 메커니즘인 클라우드 모델을 제시한다. 이 모델은 데이터 연산에 필요한 컴퓨팅 자원을 필요에 따라 할당하고 해제할 수 있는 클라우드 확장성의 장점을 극대화하여 활용한다. 또한 클라우드-디바이스 간 협력적 암호화와 컴퓨팅 및 네트워킹 자원 격리를 적용한 클라우드 아키텍쳐를 함께 제시한다.

결론적으로 서비스 내부 서버가 침해되었을 경우 데이터 스토리지에 대한 위협 수준과 제3자 프로세서가 처리할 데이터에 대한 통제 가능성을 바탕으로 보안성을 평가했을 때, 제안된 아키텍처 상의 cloud 모델은 기존 모델들 대비 수집 및 처리되는 데이터에 대한 더 높은 수준의 안전성과 통제권을 보장한다.

References

[1] S. Alessi, "Eternal Sunshine: The Right to be Forgotten in the European Union after the 2016 General Data Protection Regulation," Emory International Law Review, vol. 32(1), pp. 145-171, 2017.

[2] G. Malgieri and G. Comandé, "Why a right to legibility of automated decision-making exists in the general data protection regulation," International Data Privacy Law, vol. 7(4), pp. 243-265, 2017.

[3] C. B. Olsen, "To track or not to track? Employees' data privacy in the age of corporate wellness, mobile health, and GDPR," International Data Privacy Law, vol. 10(3), pp. 236-252, 2020.

[4] J. Sim, B. Kim, K. Jeon, M. Joo, J. Lim, J. Lee, and K. K. R. Choo, "Technical Requirements and Approaches in Personal Data Control," ACM Computing Surveys, vol. 55(9), 2023.

[5] ACCOUNTKILLER, "AccountKiller: A Service for Deleting Online Accounts," https://www.accountkiller.com/en/, April 17, 2024.

[6] Eliminalia, "Eliminalia: Online Reputation and Privacy Services," https://eliminalia.com/en/, April 17, 2024.

[7] RemoveOnlineInformation, "#1 Online Information Removal Solution," https://removeonlineinformation.com/, April 17, 2024.

[8] Mydex, "Mydex: Personal Data Management," https://mydex.org/, April 17, 2024.

[9] Cookie Information, "The cookie banner that supports your marketing goals," https://cookieinformation.com/, April 17, 2024.

[10] CookieFirst, "Cookie Consent GDPR, ePR, CCPA, LGPD compliant," https://cookiefirst.com/, April 17, 2024.

[11] digi.me, "Medical Records Viewer," https://digi.me/, April 17, 2024.

[12] SNPLab, "SNPLab Service," https://snplab.io/service, April 17, 2024.

[13] Meeco, "Meeco: Personal Data Management," https://www.meeco.me/, April 17, 2024.

[14] L. Brodsky and L. Oakes, "Data sharing and open banking," McKinsey & Company, pp. 1105, 2017.

[15] Konsentus, "GDPR, PSD2, and Open Banking: Navigating Regulatory Waters," https://www.konsentus.com/insights/articles/gdpr-psd2-and-open-banking/, April 17, 2024.

[16] Openbanking,https://developer.openbanking.privatebank.jpmorgan.com, April 17, 2024.

[17] BNY Mellon Marketplace, "Open Banking APIs Payment Service Directive (PSD2)," https://marketplace.bnymellon.com/app/open/solutions-set/detail/psd2-open-api, April 17, 2024.

[18] GOV.UK, "The midata vision of consumer empowerment," https://www.gov.uk/government/news/the-midata-vision-of-consumer-empowerment, April 17, 2024

[19] D. S. Sayogo, J. Zhang, T. A. Pardo, G. K. Tayi, J. Hrdinova, D. F. Andersen, and L. F. Luna-Reyes, "Going beyond open data: Challenges and motivations for smart disclosure in ethical consumption," Journal of Theoretical and Applied Electronic

Commerce Research, vol. 9(2), pp. 3-4, 2014.
[20] W. Choi, J. W. Chun, S. J. Lee, S. H. Chang, D. J. Kim, and I. Y. Choi, "Development of a MyData platform based on the personal health record data sharing system in Korea," Applied Sciences, vol. 11(17), p. 8208, 2021.
[21] Mint, "Budget Tracker & Planner | Free Online Money Management," https://mint.intuit.com/, April 17, 2024.
[22] PYMNTS, "FinTech BankSalad Launches HealthTech Service," https://www.pymnts.com/healthcare/2022/korean-fintech-banksalad-launches-healthtech-service/, April 17, 2024.
[23] XDA Developers, "What is Health Connect: How Google combines fitness data from Samsung, Fitbit and others," https://www.xda-developers.com/health-connect/, April 17, 2024.
[24] Computer Weekly, "Most firms will not be GDPR-ready by compliance deadline," https://www.computerweekly.com/news/252439872/Most-firms-will-not-be-GDPR-ready-by-compliance-deadline, April 17, 2024.
[25] P. Antonopoulos, A. Arasu, K. D. Singh, K. Eguro, N. Gupta, R. Jain, R. Kaushik, H. Kodavalla, D. Kossmann, N. Ogg, R. Ramamurthy, J. Szymaszek, J. Trimmer, K. Vaswani, R. Venkatesan, and M. Zwilling, "Azure SQL Database Always Encrypted," Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 1511-1525, 2020.
[26] U. T. Mattsson, "A practical implementation of transparent encryption and separation of duties in enterprise databases: protection against external and internal attacks on databases," in Proceedings of the Seventh IEEE International Conference on E-Commerce Technology (CEC'05), pp. 559-565, Jul. 2005. IEEE.
[27] V. Sidorov and W. K. Ng, "Transparent data encryption for data-in-use and data-at-rest in a cloud-based database-as-a-service solution," in Proceedings of the 2015 IEEE World Congress on Services, pp. 221-228, Jun. 2015. IEEE.
[28] N. Kumar, V. Katta, H. Mishra, and H. Garg, "Detection of data leakage in cloud computing environment," in Proceedings of the 2014 International Conference on Computational Intelligence and Communication Networks, pp. 803-807, Nov. 2014. IEEE.
[29] C. Yang, L. Tan, N. Shi, B. Xu, Y. Cao, and K. Yu, "AuthPrivacyChain: A blockchain-based access control framework with privacy protection in cloud," IEEE Access, vol. 8, pp. 70604-70615, 2020.
[30] A. Lounis, A. Hadjidj, A. Bouabdallah, and Y. Challal, "Healing on the cloud: Secure cloud architecture for medical wireless sensor networks," Future Generation Computer Systems, vol. 55, pp. 266-277, 2016.
[31] Y. J. Ong, M. Qiao, R. Routray, and R. Raphael, "Context-aware data loss prevention for cloud storage services," in Proceedings of the 2017 IEEE 10th International Conference on Cloud Computing (CLOUD), pp. 399-406, Jun. 2017. IEEE.
[32] P. Han, C. Liu, J. Cao, S. Duan, H. Pan, Z. Cao, and B. Fang, "CloudDLP: Transparent and scalable data sanitization for browser-based cloud storage," IEEE Access, vol. 8,

pp. 68449-68459, 2020.

[33] A. Kumar, B. G. Lee, H. Lee, and A. Kumari, "Secure storage and access of data in cloud computing," in Proceedings of the 2012 International Conference on ICT Convergence (ICTC), pp. 336-339, Oct. 2012. IEEE.

[34] A. Alsirhani, P. Bodorik, and S. Sampalli, "Improving database security in cloud computing by fragmentation of data," in Proceedings of the 2017 International Conference on Computer and Applications (ICCA), pp. 43-49, Sep. 2017. IEEE.

[35] C. J. Chae, Y. Shin, K. Choi, K. B. Kim, and K. N. Choi, "A privacy data leakage prevention method in P2P networks," Peer-to-Peer Networking and Applications, 9(3), pp. 508-519, 2016.

[36] X. Zhao, M. Li, E. Feng, and Y. Xia, "Towards a secure joint cloud with confidential computing," in Proceedings of the 2022 IEEE International Conference on Joint Cloud Computing (JCC), pp. 79-88, Aug. 2022. IEEE.

[37] W. Qiang, Z. Dong, and H. Jin, "Se-lambda: Securing privacy-sensitive serverless applications using SGX enclave," in Security and Privacy in Communication Networks: 14th International Conference, SecureComm 2018, pp. 451-470, Aug. 2018. Springer International Publishing.

[38] Alder, F., Asokan, N., Kurnikov, A., Paverd, A., & Steiner, M. (2019). S-FaaS: Trustworthy and accountable function-as-a-service using Intel SGX. Proceedings of the ACM Conference on Computer and Communications Security, 185 - 199.

[39] P. Padma and S. Srinivasan, "DAuth –Delegated Authorization Framework for Secured Serverless Cloud Computing," Wireless Personal Communications, vol. 129, no. 3, pp. 1563-1583, 2023.

[40] A. Koo, Y.-G. Kim, and S. H. Lee, "Design of Security Architecture for the Cloud-Based Korea Military Command and Control System," The Journal of Korean Institute of Communications and Information Sciences, vol. 45, no. 2, pp. 400-408, 2020.

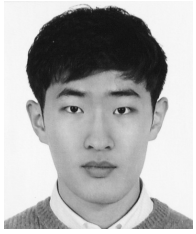
[41] OpenStack Documentation, https://docs.openstack.org/2024.1/, April 17, 2024.

[42] QEMU, "QEMU documentation," https://www.qemu.org/docs/master/, April 17, 2024.

[43] MyData Korea, "Standard API Specification of MyData in Financial Scope," https://developers.mydatakorea.org/m, April 17, 2024

〈저자소개〉

허 승 민 (Seungmin Heo) 학생회원
2022년 2월: 숭실대학교 컴퓨터학부 학사 졸업
2022년 3월~현재: 고려대학교 정보보호대학원 정보보호학과 석사과정
〈관심분야〉 정보보호, 클라우드 보안



권 용 희 (Yonghee Kwon) 정회원
2021년 2월: 고려대학교 정보보호학부 학사 졸업
2022년 3월~현재: 고려대학교 대학원 정보보안학과 석·박사통합과정
〈관심분야〉 클라우드 및 하드웨어 시스템 보안, 모의해킹

김 범 중 (Beomjoong Kim) 학생회원
2021년 2월: 고려대학교 물리학과 학사 졸업
2021년 3월~현재: 고려대학교 정보보호대학원 정보보호학과 석·박사통합과정
〈관심분야〉 암호자산보안, 블록체인, 사이버보안

전 기 석 (Kiseok Jeon) 학생회원
2017년 6월: Shanghai jiaotong University Media and Editing 학사 졸업
2021년 9월~현재: 고려대학교 정보보호대학원 정보보호학과 석·박사통합과정
〈관심분야〉 정보보호,시스템 소프트웨어 보안, 블록체인 마이닝 소프트웨어

이 중 희 (Junghee Lee) 종신회원
2000년 2월: 서울대학교 컴퓨터공학과 졸업
2003년 2월: 서울대학교 컴퓨터공학과 석사
2013년 12월: Georgia Institute of Technology 전자공학과 박사
2003년 2월~2008년 8월: 삼성전자 연구원
2014년 9월~2019년 1월: University of Texas at San Antonio 전자공학과 조교수
2019년 3월~현재 고려대학교 정보보호대학원 조/부교수
〈관심분야〉 하드웨어 보안, 블록체인