aws re/start

예방: 데이터 보안

Security Fundamentals

보안 수명 주기: 예방 - 데이터 보안을 시작하겠습니다.

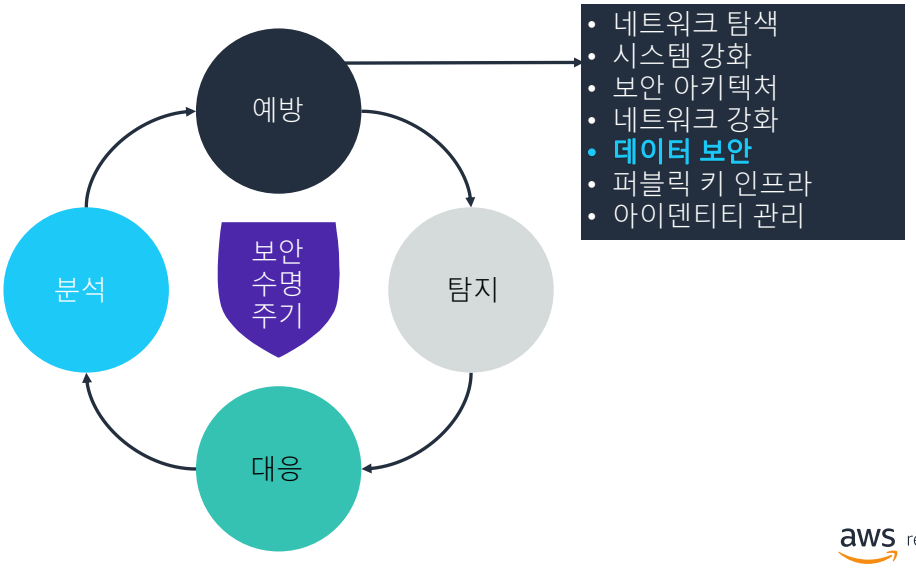
교육 내용

이 강의의 핵심

배울 내용은 다음과 같습니다.

- 데이터 기밀성을 보호하는 데 도움이 되는 암호화 기법을 설명합니다.
- 데이터 무결성 보장을 위한 해싱 방법을 설명합니다.
- 임의 액세스 제어와 역할 기반 액세스 제어를 구분합니다.

보안 수명 주기: 예방



aws re/start

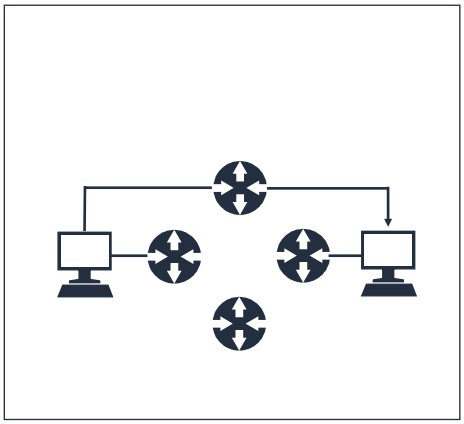
복습하자면 보안 수명 주기는 이렇게 구성됩니다.

- **예방** - 첫 번째 방어선입니다.
- **탐지** - 예방이 실패했을 때 수행됩니다.
- **대응** - 보안 위협이 탐지되었을 때 취해야 할 조치를 설명합니다.
- **분석** - 향후에 문제가 다시 발생하지 않도록 예방하는 새로운 조치를 구현하면서 주기가 완료됩니다.

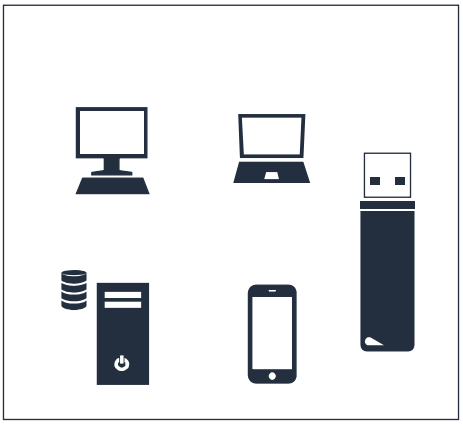
이 강의에서는 **데이터 보안**의 개념과 예방 단계에서 사용할 수 있는 방법에 관해 배웁니다.

전송 중 데이터와 저장 데이터

전송 중 데이터



저장 데이터



민감 데이터가 네트워크와 시스템을 통해 이동 중일 때와 저장 상태일 때 모두 민감한 데이터를 보호합니다.

암호화 기법과 제어 조치를 사용하여 데이터가 이동 중인지 저장 상태인지에 따라 데이터를 보호합니다.

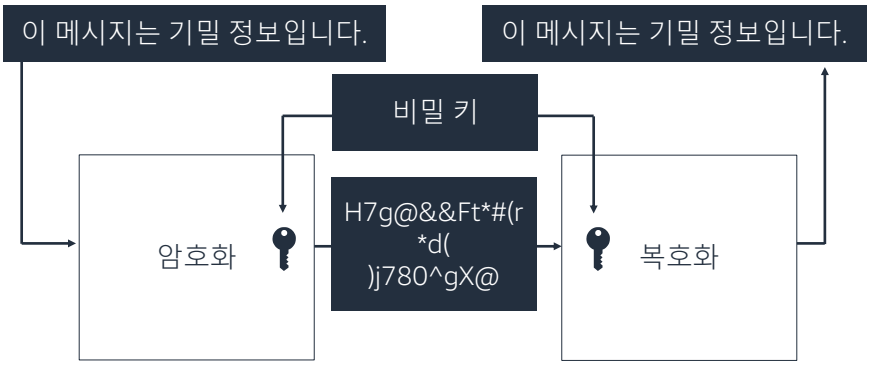
암호화 기법

- 기밀성과 데이터 무결성을 비롯하여 데이터 보안을 제공하는 원칙과 기법을 포함하는 규율입니다.

암호화 기법은 기밀성과 데이터 무결성을 비롯하여 데이터 보안을 제공하는 원칙과 기법을 포함하는 규율입니다. 암호화 기법은 민감한 정보에 대한 액세스를 보호하고 유지 관리하기 위해 정보에 특정 수준의 기밀성과 무결성을 제공하는 현대의 컴퓨팅 기법을 활용하는 관행의 집합입니다. 암호화 기법은 암호화가 아니며, 암호화는 암호화 기법 중 하나입니다.

암호화

- 용례
  - 데이터를 암호화하는 암호화 기법 알고리즘
  - 암호화된 데이터를 해독하는 비밀 키
- 암호화의 유형: 대칭, 비대칭, 하이브리드

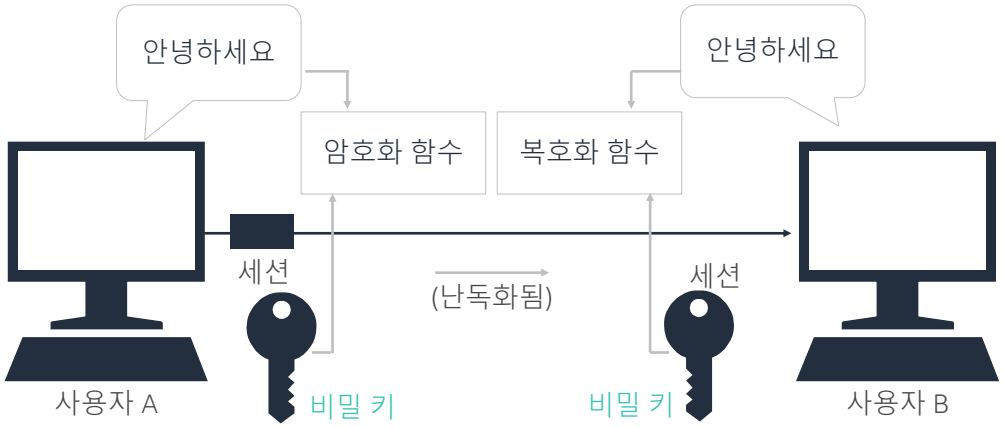


aws re/start

암호화의 목표는 데이터의 기밀성을 확보하는 것입니다.

암호화에는 **대칭, 비대칭, 하이브리드**의 세 가지 유형이 있습니다.

대칭 암호화



aws re/start

대칭 암호화는 같은 키를 사용하여 데이터를 암호화하고 복호화합니다. 키는 발신자와 수신자 간에 공유되는 비밀입니다. 대칭 암호화는 빠르고, 신뢰할 수 있으며, 대량의 데이터에 사용됩니다.

대칭 암호화를 사용하는 보안 표준은 다음과 같습니다.

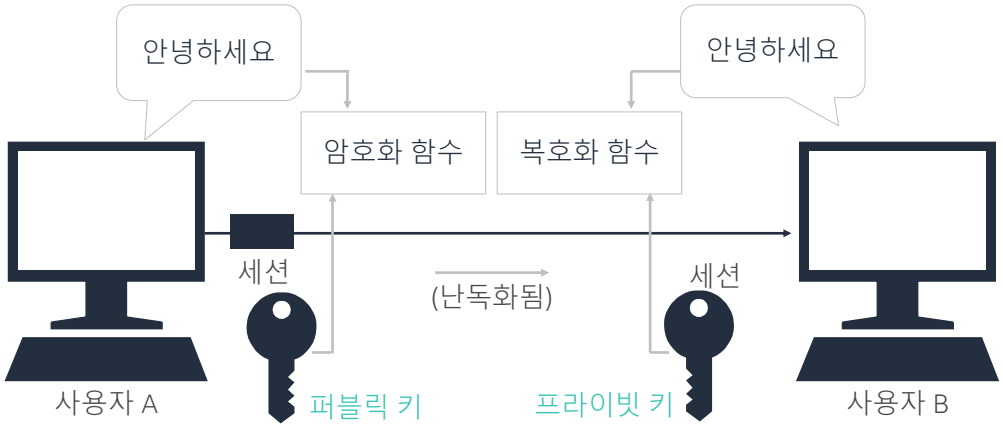
- Advanced Encryption Standard(AES) - 미국 National Institute of Standards and Technology(NIST)에서 확립한 표준으로, 기존의 Triple Data Encryption Algorithm (3DES) 표준을 대체합니다.

  AES에는 세 가지의 고정된 128비트 블록 암호가 있습니다. 암호화 키 크기는 128비트, 192비트, 256비트입니다. 블록 암호는 데이터 블록 하나에 한 번에 적용되는 키와 알고리즘입니다. AES를 사용하면 키 크기가 무제한이지만 블록 크기는 최대 256비트입니다. 예를 들어, Amazon Elastic File System(Amazon EFS)은 AES-256 암호화 알고리즘을 사용하여 저장 중 데이터를 암호화합니다.

- International Data Encryption Algorithm(IDEA) - 스위스에서 만들고 특허를 낸 표준으로, 비상업적으로 사용 시 무료입니다. 이 표준은 128비트 키로 블록 암호를 사용합니다.

- Twofish - AES보다 느린 퍼블릭 도메인 암호화 알고리즘입니다.

비대칭 암호화



aws re/start

비대칭 암호화는 프라이빗 키와 퍼블릭 키(**키 페어**)를 사용하여 데이터를 암호화하고 복호화합니다. 대화에 참여하는 모든 사용자가 키 페어를 가집니다. 비대칭 암호화는 대칭 암호화보다 더 복잡하며 훨씬 느립니다. 그러나 키 관리에 있어서는 더 많은 기능을 제공합니다.

비대칭 암호화를 사용하는 보안 표준은 다음과 같습니다.

- Rivest-Shamir-Adleman(RSA) - 소인수 분해를 기반으로 알고리즘을 사용합니다. 따라서 RSA 키를 추론하려면 시간과 프로세싱 파워가 많이 소요됩니다. 중요한 데이터, 특히 인터넷을 통해 전송되는 데이터에 표준으로 사용되는 암호화 방법입니다.

- Diffie-Hellman(DH) - 퍼블릭 채널을 통한 암호화 키의 키 교환 방법입니다. 이 방법에서는 암호화 키를 생성하기 위해 특정 지수만큼 거듭제곱한 숫자를 사용합니다.

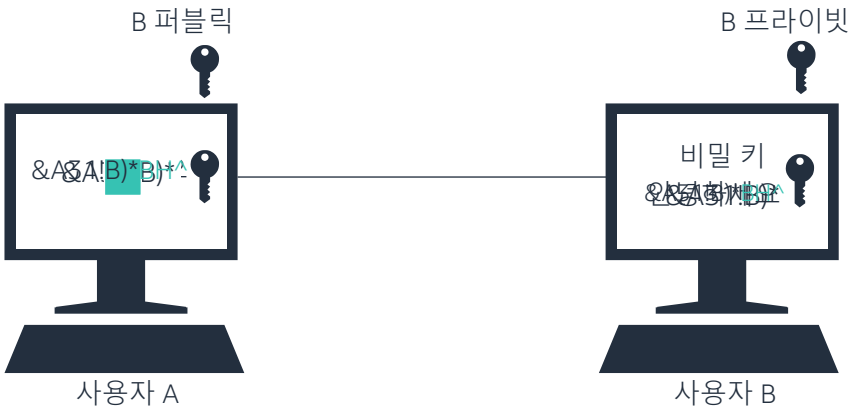
- ElGamal - DH 방법을 기반으로 한 알고리즘을 사용합니다.

암호화 유형 비교

	대칭	비대칭
프로세스 속도	빠름(데이터의 양이 많아도 빠름)	느림
보안 수준	매우 안전함	추가 보안 서비스를 제공함
관리 용이성	키가 많아지면 복잡해짐	키 시스템을 관리하기 쉬움
제공되는 보안 서비스	기밀성만	부인 방지, 인증 등

이 테이블에는 대칭 암호화와 비대칭 암호화가 비교되어 있습니다.

하이브리드 암호화



aws re/start

하이브리드 암호화 접근법에서는 대칭 암호화와 비대칭 암호화를 모두 사용하여 데이터를 한층 더 보호합니다.

하이브리드 암호화 방법을 적용하려면 예에 있는 이 단계를 따릅니다.

1. 사용자 A가 **안녕하세요**라는 메시지를 사용자 B에게 안전하게 보내려고 합니다.

2. 사용자 A는 사용자 B와 공유된 **비밀 키**를 사용하여 메시지를 **&A31!B)***로 암호화합니다.
   이 부분은 대칭 암호화입니다.

3. 그런 다음 사용자 A는 사용자 B의 **퍼블릭 키**를 사용하여 메시지를 **&A31!B)*BH^**로 한 번 더 암호화합니다.
   이 부분은 비대칭 암호화이며, 이제 메시지가 완전히 암호화되었습니다.

4. 암호화된 메시지가 사용자 B에게 전송되고 **&A31!B)*BH^**라는 형태로 도착합니다.

5. 사용자 B는 **프라이빗 키**를 사용하여 메시지를 **&A31!B)***로 복호화합니다.
   이 부분은 비대칭 복호화입니다.

6. 사용자 B는 사용자 A와 공유된 **비밀 키**를 사용하여 메시지를 **안녕하세요**로

복호화합니다.
이 부분은 대칭 복호화입니다.

하이브리드 암호화를 사용하는 프로토콜은 보안 소켓 계층(SSL)이라고 알려진 전송 계층 보안(TLS)입니다.

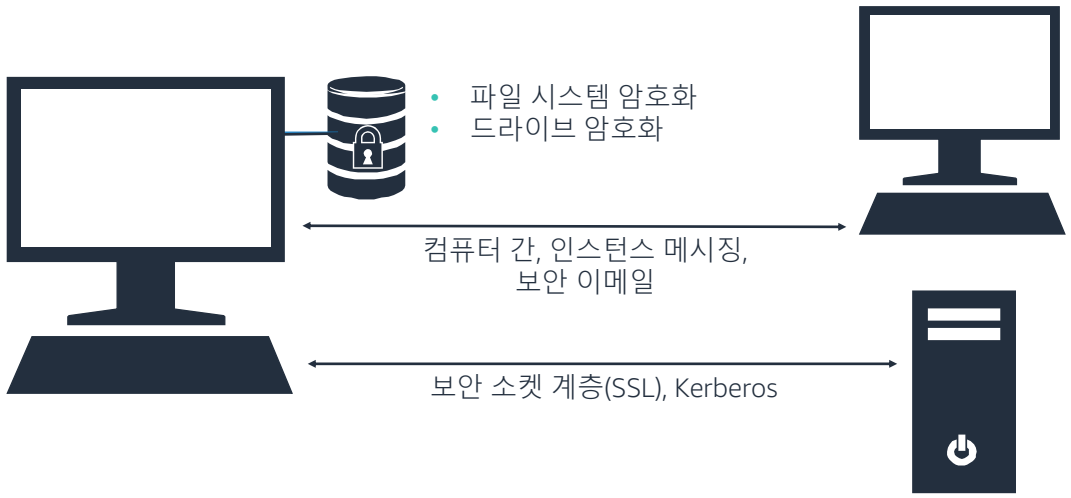
토론

보안을 위해 암호화해야 하는 데이터는 무엇입니까?

aws re/start

어떤 유형의 데이터를 암호화할지 생각해 보고 다른 사람들과 답변을 논의해 보십시오.

실용적인 용례: 암호화 기법



aws re/start

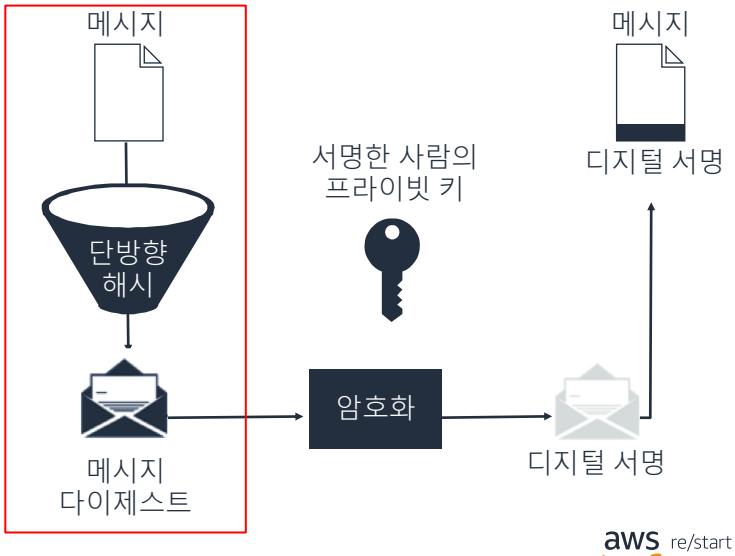
다음 예에서는 데이터 암호화를 사용합니다.

- **저장 데이터 암호화 -**
  - **파일 시스템 암호화**: 예를 들어 Windows New Technology File System(NTFS)을 사용하여 하나의 파일 또는 전체 파일 세트를 암호화할 수 있습니다.
  - **드라이브 암호화**: 드라이브 전체와 그 콘텐츠를 암호화합니다. 예에는 BitLocker와 VeraCrypt가 있습니다. VeraCrypt는 오픈 소스 드라이브 암호화를 위한 무료 솔루션입니다.
- **전송 중 데이터 암호화 -**
  - **보안 소켓 계층(SSL)/전송 계층 보안(TLS)**: 유선으로 전송되는 데이터를 보호하는 데 사용되는 프로토콜입니다. URL의 시작 부분에 **https**가 있는 보안 웹 사이트를 방문할 때 **s**는 **SSL**을 의미합니다. TLS는 SSL의 뒤를 이은 프로토콜입니다.
  - **Kerberos**: 두 디바이스 간의 모든 통신을 암호화하는 데 사용되는 암호화 기술입니다.
  - **IP Security(IPsec)**: 이 프로토콜은 인터넷 프로토콜 버전 6(IPv6)의 핵심적인 구성 요소로 도입되었으며, 인터넷 프로토콜 버전 4(IPv4)에서 빌려서 사용합니다. 두 IP 모두 각 디바이스의 식별을 통해 네트워크 디바이스 간에 논리적인 연결을 제공합니다. IPsec은 가상 프라이빗 네트워크(VPN)를 보호하는 데 사용됩니다.
  - **인스턴스 메시징 또는 보안 이메일**: 클라이언트와 서버 간의 통신을 보호하기 위해 TLS를 사용하는 데 더해, 메시지 자체를 보호하고 싶을 수도 있습니다. Secure/Multipurpose Internet Mail Extensions(S/MIME)과

Pretty Good Privacy(PGP)는 이를 위해 사용하는 프로토콜입니다.

해싱을 통한 데이터 무결성 보장

- 해싱은 데이터 무결성을 보장하는 데 사용됩니다.
- 해시 함수는 파일 또는 메시지 콘텐츠로부터 **고유한 해시 값** 또는 **메시지 다이제스트**를 생성합니다.
- 해당 파일 또는 메시지의 수신자는 해시 값을 사용하여 전송 중에 콘텐츠가 변경되지 않았는지 확인할 수 있습니다.



aws re/start

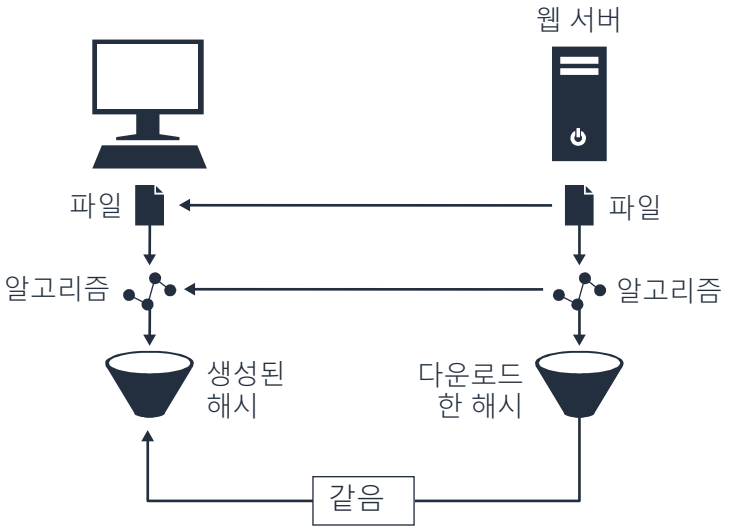
해싱은 **데이터 무결성**을 보장하는 데 사용되는 기법입니다. 데이터 무결성이란 여러분이 받는 데이터가 전송된 데이터와 같은 것임을 보장하는 것입니다. 또한 데이터가 변경되거나 조작되지 않았다는 것을 의미합니다.

해싱은 메시지 또는 파일 속의 데이터를 읽고, 데이터로부터 **해시 값** 또는 **메시지 다이제스트**라고 하는 고유한 텍스트 문자열 값을 생성하는 함수를 사용합니다. 같은 해싱 함수를 사용하여 같은 파일을 해시 복사하면 항상 같은 해시 값이 생성됩니다. 해싱을 통해 예를 들면 파일이 네트워크를 통해 전송된 후 파일의 콘텐츠가 변경되지 않았음을 확인할 수 있습니다. 해싱 함수는 일반적으로 보안 해시 알고리즘 버전 1(SHA1)과 메시지 다이제스트 버전 5(MD5)와 같은 표준 알고리즘을 사용합니다.

이 슬라이드의 다이어그램에는 수신자에게 전송될 메시지의 예시가 있습니다. 메시지는 먼저 데이터 무결성을 보장하기 위해 해시 처리된 후 데이터 기밀성을 보장하기 위해 암호화됩니다. 구체적으로 설명하면 다음과 같습니다.

- 메시지는 해시되어 메시지 다이제스트를 생성합니다. 이 다이제스트는 메시지의 수신자가 메시지가 전송 중에 조작되지 않았음을 확인하는 데 사용할 수 있습니다.
- 메시지는 프라이빗 키를 사용하여 암호화되고 수신자가 디지털 서명을 한 상태입니다.

실용적인 용례: 해싱을 통한 무결성



aws re/start

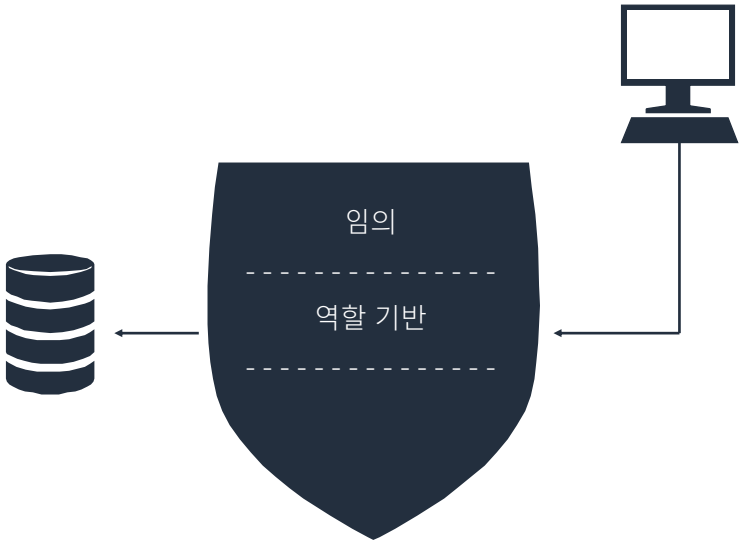
일반적으로 해시는 웹 사이트에서 다운로드한 파일의 무결성을 확인하는 데 사용됩니다.

이 다이어그램에는 다음과 같은 단계가 포함된 일반적인 흐름이 나와 있습니다.

1. 사용자가 온라인에서 파일과 파일의 해시를 찾습니다.
2. 사용자가 파일을 다운로드합니다.
3. 사용자가 로컬 도구를 다운로드한 데이터에 실행하여 해시를 생성합니다.
4. 사용자가 온라인 해시 값을 다운로드하여 로컬에서 생성한 해시와 비교합니다.
5. 두 해시가 같으면 무결성이 유지된 것이며, 파일이 디지털 서명 후 변경되지 않았음을 나타냅니다. 해시가 일치하지 않으면 파일의 무결성이 침해된 것이며, 파일을 신뢰해서는 안 됩니다.

권한의 원칙

- 권한은 리소스에 대한 명시적 액세스를 허용하는 데 사용됩니다.
- 주체는 객체에 대한 권한을 부여받습니다.
- 일반적인 권한의 유형:
  - 임의
  - 역할 기반



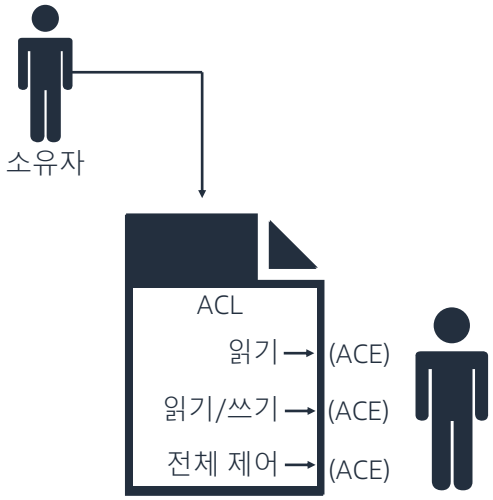
aws re/start

권한은 리소스에 특정 유형의 액세스를 부여합니다. 예를 들면 파일에 대한 쓰기 액세스를 부여하는 것입니다. 권한은 두 가지 유형으로 분류됩니다. 임의 권한과 역할 기반 권한입니다.

권한은 주체(예: 사람, 디바이스, 시스템)에 할당되어 주체에게 권한에 정의된 리소스 액세스 능력을 제공합니다.

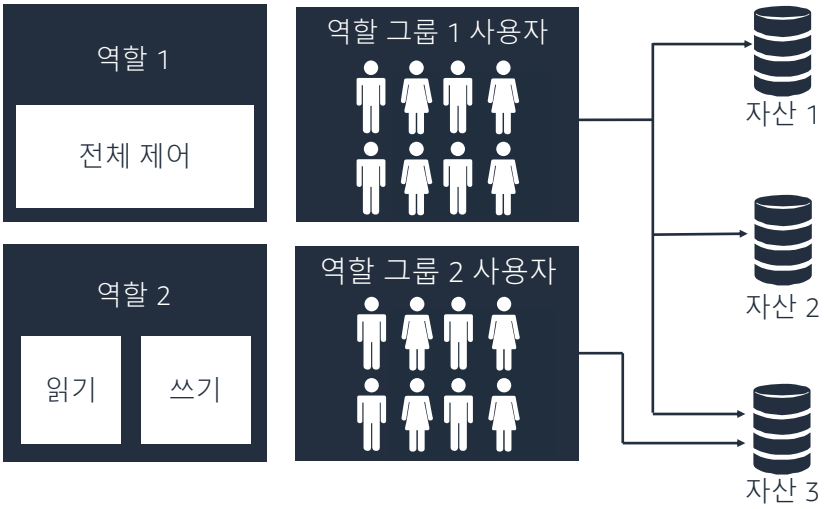
임의 액세스 제어

- DAC 속성:
  - 분산된 액세스
  - 액세스 제어 목록(ACL)
  - 액세스 제어 항목(ACE)
  - 감사
  - 동적 액세스 제어



임의 액세스 제어(DAC) 유형에서는 개인에게 리소스에 대한 특정 수준의 액세스가 할당됩니다. 액세스 수준 정보는 액세스 제어 목록(ACL)에 저장되어 있습니다.

역할 기반 액세스 제어



aws re/start

역할 기반 액세스 제어 유형에서는 리소스에 대한 액세스 수준이 역할에 할당됩니다. 즉, 권한이 역할을 기반으로 분배됩니다. 그런 다음 리소스에 대한 액세스가 필요할 때 개인이 각기 다른 역할에 할당됩니다.

- 변경이 필요할 때마다 상위 수준의 상호 작용을 요구하지 않는 현대적인 권한 접근법
- 보다 단기적인 프로젝트와 태스크에서 직원 이직률이 높고 채용이 잦을 때 효율적임
- 많은 부분에서 사용자 지정이 가능함
- 상업 분야에서 많이 사용됨
- 세분화 수준이 충분하기 때문에 사용이 확장되고 있음

예를 들어 한 회사에서 직무 역할에 따라 권한을 정의하기로 결정합니다. 새로운 직원을 채용했을 때 이 직원을 적절한 역할 그룹에 배치하기만 하면 이 직원은 자동으로 그 역할에 맞는 모든 권한을 갖게 됩니다.

핵심 사항

- 암호화는 데이터의 기밀성을 보호합니다.
- 암호화에는 대칭, 비대칭, 하이브리드의 세 가지 유형이 있습니다. 하이브리드는 TLS/SSL 프로토콜과 같은 인터넷 통신 프로토콜에서 널리 사용됩니다.
- 해싱은 데이터의 무결성을 보호합니다.
- 권한은 리소스에 누가 어떻게 액세스할 수 있는지를 정의합니다. 액세스 제어 목록(ACL) 또는 역할 기반 접근법으로 권한을 구현합니다.

aws re/start

이 강의에서 다룬 핵심 사항은 다음과 같습니다.

- 암호화는 데이터의 기밀성을 보호합니다.
- 암호화에는 대칭, 비대칭, 하이브리드의 세 가지 유형이 있습니다. 하이브리드는 TLS/SSL 프로토콜과 같은 인터넷 통신 프로토콜에서 널리 사용됩니다.
- 해싱은 데이터의 무결성을 보호합니다.
- 권한은 리소스에 누가 어떻게 액세스할 수 있는지를 정의합니다. 액세스 제어 목록(ACL) 또는 역할 기반 접근법으로 권한을 구현합니다.