이승용 - bongbongco.github.com

암호 기법의 소개

1. 목적

기초적인 암호 기법의 개념과 배경 지식에 대해서 알아보고자 함.

2. 개요

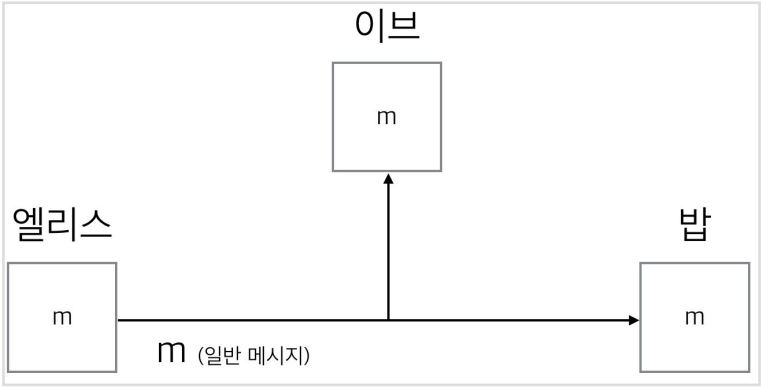
보안에서 암호 시스템은 빼놓을 수 없는 중요 요소입니다. 인증, 디지털 서명과 같은 많은 보안 관련 기능을 포함하면서 영역 또한 한층 더 확장되었습니다. 암호는 그 기법 자체만으로는 의미를 가지지 못하고 큰 시스템의 자물쇠로 역할을 수행합니다. 잊을만하면 한번 씩 발생하는 'Openssl' 취약점으로 단지 'Openssl' 을 적용한 암호화 통신을 하고 있다고 해서 능사가 아님을 느끼게 합니다. 이에 기초적인 암호 기법에서 부터 추후 암호 기법의 세부적인 내용까지 정리하고자 합니다.

3. 설명

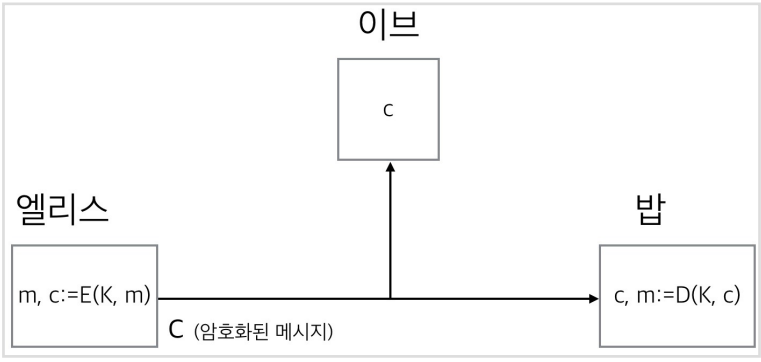
3.1. 암호화

암호화는 암호 기법의 기본 목표입니다. (예제에서 사용되는 이름은 엘리스와 밥, 이브이며, 이브가 공격자 역할을 맡습니다.)

- 엘리스와 밥이 통신하는 일반적인 방법



- 암호문 통신의 일반적인 설정



1. 이브는 밥과 엘리스 사이의 통신을 엿들을 수 있습니다. 엘리스와 밥의 대화를 이브가 엿듣는 것을 막기 위해서 위의 이미지와 같은 암호화 방법을 사용합니다.
2. 엘리스와 밥은 키 K 에 동의합니다. 이것은 이브가 듣지 못하는 다른 채널을 이용하여 주고 받습니다.
3. 엘리스가 메시지 m 을 보낼 때, 암호화 c := E(K, m)을 이용하여 암호문 c 를 보냅니다.
4. 밥은 복호화 기능을 하는 복호화 함수 D(K, c)를 이용하여 엘리스가 밥에게 보내기를 원했던 평문 m을 얻을 수 있게 됩니다.
5. 이브는 키 K 를 모르기 때문에 이브가 암호문 c 를 복호화할 수 없습니다.

3.1.1. 커차프스의 법칙

암호문을 해독하기 위해서는 두 가지를 알아야 합니다. 바로 복호화 알고리즘인 D 와 키인 K 입니다. 커차프스의 법칙에서 가장 중요한 사항은 다음과 같습니다.

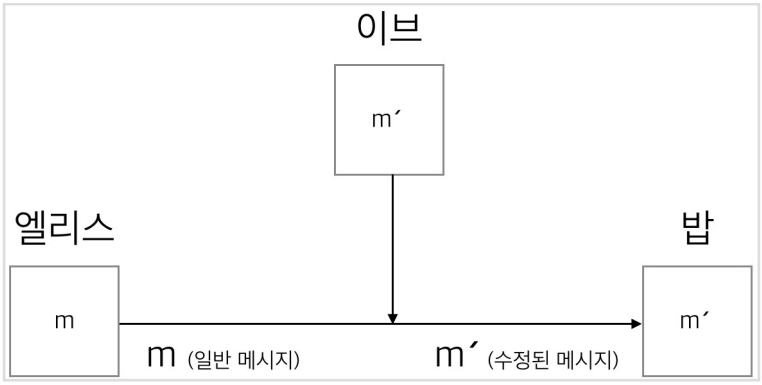
"암호화 방법의 보안성은 암호화 알고리즘의 보안성능이 아니라 키인 K 를 얼마나 잘 보호하는가에 달려있다."

커차프스의 법칙이 옳은 이유 첫 번째는 알고리즘을 바꾸기 어렵기 때문입니다. 두 번째는 키를 비밀스럽게 관리하는 것이 알고리즘의 비밀을 유지하는 것보다 쉽기 때문입니다. 어떤 시스템도 두 사람을 위해서 암호 시스템을 구축하지 않습니다. 시스템의 모든 관계자는 같은 알고리즘을 사용할 것입니다. 비밀스러운 알고리즘을 통해서 암호 시스템을 구축했을 때, 이브는 관계자 중 한명으로 부터 알고리즘을 획득하고 공격에 성공할 가능성이 높습니다.

이외에도 알고리즘을 공개해야하는 이유는 공격 당하기 전에는 알고리즘의 결점을 찾아낼 수가 없기 때문입니다.

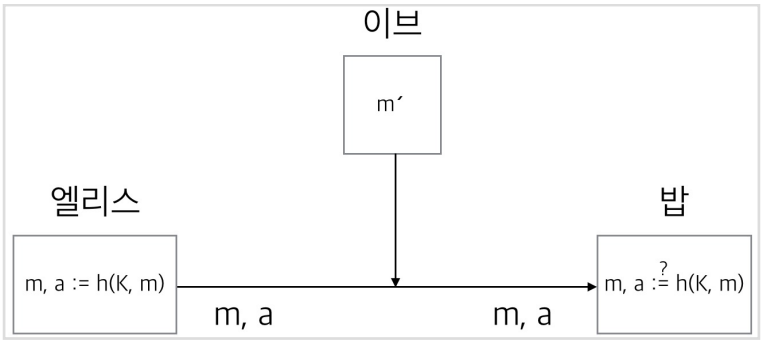
3.2. 인증

이브는 메시지를 중간에 가로채서 메시지를 바꾼 후에 밥에게 보낼 수 있습니다.



예를 들어 엘리스는 메시지 m 을 밥에게 보내려고 합니다. 그러나 이브는 이 통신을 간섭하고 있어서, m 을 받는 대신 m′ 를 받을 수 있습니다. 이밖에도 이브는 메시지를 지워서 밥이 아무 메시지도 받지 못하게 하거나, 새로운 내용을 메시지에 삽입하는 일, 메시지를 저장했다가 나중에 밥에게 보내는 일, 메시지의 순서를 바꾸는 일 등을 할 수 있습니다.

인증에 관한 문제를 해결하기 위해서, 암호화와 같이 인증에서 엘리스와 밥이 알고 있는 비밀 키를 이용합니다. (여기서 인증키 K는 메시지 인증 키와 구분됨) 위 그림에 메시지 m 의 인증과정이 있습니다. 엘리스가 메시지 m 을 보낼 때, 메시지 인증 코드(Message Authentication Code)를 계산합니다.



1. 엘리스는 메시지 인증 코드 a 를 a := h(K, m) 로 계산합니다. (h 는 인증 키를 검사하는 메시지 인증 코드 함수입니다.)
2. 엘리스는 메시지 m 과 메시지 인증 코드 a 를 함께 전송합니다.
3. 밥이 m 과 a 를 받았을 때, 인증 키를 이용하여 a 가 맞는 지 계산하고 a 가 엘리스가 보낸 것이 맞는 지 확인합니다.

이브는 현재 메시지 m 을 다른 메시지 m′ 로 수정하려합니다.

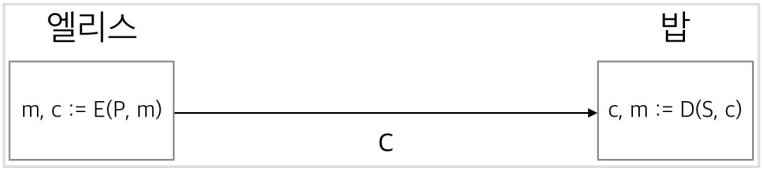
1. 이브는 간단히 m 을 m′ 로 바꾸었습니다.
2. 밥은 h(K, m′) 를 이용해서 a 값을 비교합니다.
3. 밥은 이비가 보낸 메시지가 틀린 메시지라는 것을 알게 됩니다.

하지만 이브가 인증키 K를 모른다 하여도 엘리스가 밥에게 메시지를 보낼 때, 이브가 메시지 인증 코드를 가로채서 활용할 수 있습니다. 따라서 순수한 인증은 완벽한 해답이 될 수 없습니다. 인증 수행 중에도 오래된 메시지를 반복해서 보내거나, 메시지의 순서를 바꾸는 등의 행동을 여전히 할 수 있기 때문입니다. (그래서 인증은 거의 항상 메시지를 순차적으로 셀 수 있게 번호 부여 방식과 혼합해서 사용됨) 메시지에 번호를 붙이는 것과 인증의 혼합은 일반적으로 사용되는 대부분의 문제 해결 방법입니다. 밥은 오직 올바른 메시지 인증 코드만을 확인하고 마지막 메시지 번호보다 큰 메시지만을 확인하면 됩니다.

인증과 암호화의 두 가지 개념을 절대로 혼동하지 말아야 합니다. 암호화된 메시지는 메시지의 내용을 변조하는 것을 막지못하며, 인증된 메시지는 메시지의 비밀을 지키지 못합니다.

3.3 공개 키 암호화 (Public-Key Encryption)

3.1 절에서의 암호화 방법을 이용하기 위해서 엘리스와 밥은 키 K를 어떻게 공유하여 알 수 있을까요. 만약 20명의 친구들과 서로 통신하려면, 각자 19개의 키를 교환해야 하고 모두가 교환하는 키를 생각하면 총 190개가 됩니다. 공개키 암호화 기법은 키 분배의 문제를 해결합니다. 아래 그림은 3.1 절의 '암호문 통신의 일반적인 설정' 과 유사하지만 엘리스와 밥이 같은 키를 사용하지 않는 다는 점에서 차이를 보입니다.



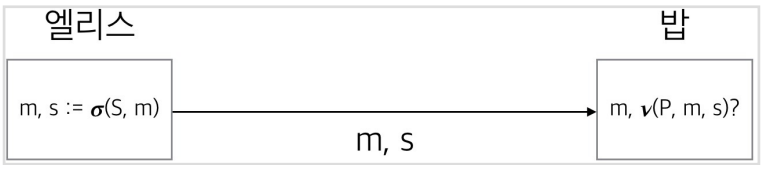
1. 밥은 먼저 두개의 키(S, P)를 알고리즘을 이용해서 생성합니다. (S: 비밀 키, P: 공개 키)
2. 엘리스가 메시지를 밥에게 보내려고 할 때, 밥이 공개한 공개 키 P 를 이용하여 메시지 m 을 암호화하여 c 를 만듭니다.
3. 암호문 c 를 밥에게 전송합니다.
4. 밥은 비밀키 S 를 이용해서 복호화합니다.

이러한 방식이 성공하기 위해서는 두 개의 키 알고리즘과 암호화/복호화 알고리즘이 암호 해독을 할 때 실제적으로 원본 메시지를 만들 수 있다는 것을 보증해야 합니다. (다른 말로는, D(S, E(P, m)) = m) 그리고 다른 요구사항은 공개 키로 비밀 키를 계산할 수 없어야 합니다. 이러한 암호화 기법을 비대칭 키 암호화 방법이나, 공개 키 암호화 방법이라고 합니다. 반대로 앞에서 본 방식들은 대칭 키 암호화 방법 혹은 비밀 키 암호화 방법이라고 부릅니다. 이러한 장점에도 비밀 키 암호화 방법이 여전히 사용되는 이유는 공개 키 암호화 기법은 내부적으로 비효율적인 구조를 지니고 있기 때문에 모든 부분에 적용하기에 적절하지 않기 때문입니다.

3.4 전자 서명 (Digital Signature)

전자 서명은 메시지 인증에 사용되는 공개 키와 같은 개념입니다. (메시지를 전송하는 사람이 자신의 비밀 키로 메시지를 암호화해서 보냄. 공개 키 암호화 방식과는 다르게 공개 키와 비밀 키가 하는 역할이 바뀜.)

1. 엘리스는 키 생성 알고리즘을 이용해서 두 쌍의 키(S, P) 를 생성합니다.
2. 공개 키인 P 를 공개합니다.
3. s := σ(S, m) 을 이용하여 서명 s 를 만들어 냅니다.
4. 엘리스는 m 과 s 를 밥에게 보냅니다.
5. 밥은 서명을 검증하기 위해서 엘리스의 공개 키를 검증 알고리즘 ν(P, m, s) 를 통해서 검증합니다.



밥이 공개 키로 서명을 검증할 수 있다는 것을 제외하고는 메시지 인증 코드와 똑같이 동작합니다. 밥은 엘리스에게 받은 메시지를 확인하기 위해서 엘리스의 공개 키가 필요합니다. 여기서 공개 키는 누구나 가질 수 있고, 공개 키를 이용해서 메시지를 검증할 수 있으므로 디지털 서명이라고 불리게 됩니다. 그러나 실제 세계에서는 전자 서명이 생각만큼 유용하지 않습니다. 바로 본인 스스로가 서명을 만들 수 없다는 점 때문입니다.

3.5 공개 키 기반 구조 (PKI)

엘리스는 키가 밥의 공개 키 인지 다른 사람의 키인지 어떻게 알 수 있을까요. 일반적인 해결 방법은 PKI(Public Key Infrastructure) 입니다. 중앙 인증기관(Certificate Authority)를 이용하는 것입니다.

1. 각각의 사용자는 공개 키를 인증기관에 보내고 인증기관으로부터 사용자인증을 받습니다.
2. 인증기관은 사용자의 공개 키에 전자 서명으로 서명을 합니다.
3. 서명한 메시지 혹은 인증서는 인증기관의 검증, 유효기간 등 유용한 정보를 담게 됩니다.

인증서를 사용하는 것으로 엘리스는 밥의 키를 쉽게 찾을 수 있고 올바른 키임을 검증할 수 있게 되었습니다. 공개 키 기반 구조에서는 각각의 통신 참가자가 인증기관에서 검증 받은 공개 키를 가져야하고 인증기관의 공개 키를 이용해서 다른 통신 참가자의 인증서를 스스로 검증할 수 있어야 합니다. 이로써 공개 니 기반 구조의 가장 큰 이점인 '한번 등록하고 나면 어디서든 쓸 수 있는' 환경이 만들어지게 됩니다.

보통의 경우 공개 키 기반 구조는 여러 단계의 인증기관으로 구성됩니다. 맨 위 단계의 인증 기관을 루트라 부르고 루트는 아래 단계의 인증 기관들의 키를 보증합니다. 아래 단계의 인증 기관은 사용자의 키를 인증합니다. 이러한 공개 키 기반 구조에도 문제점이 존재합니다. 첫 번째로, 인증 기관은 모든 사람들에게 신뢰를 받을 수 있어야 한다는 것입니다. 여러 이해관계와 환경이 부딪혀 모든 회사나 모든 나라가 다 같이 신뢰하는 인증 기관은 존재할 수 없습니다. 두 번째로 책임의 문제가 있습니다. 인증 기관이 잘못된 인증서를 발급하거나, 전자 서명에 사용하는 비밀 키를 도난 당한다면 많은 피해가 발생할 것 입니다. 하지만 이에 대한 보상의 주체를 규정하고 금액을 산정하는 데 많은 어려움이 생깁니다.

4. 결론

지금까지 기초적인 암호 기법의 개념과 배경 지식에 대해서 정리해보았습니다. 추상적으로 생각했던 공개 키 암호화 방식과 비밀 키 암호화 방식을 정리하면서 이해도를 높일 수 있었던 것이 핵심이었습니다. 아직까지는 추상적인 내용들만 있어 활용도가 낮습니다. 다음에는 블록 암호화에 대해서 정리하도록 하겠습니다. 혹시 모르시는 분들은 읽고 도움이 되시면 좋겠습니다.