정보보안기사 서브노트

PART 01 : 정보보호 개요 — 정보보호관리의 개념

❑ 단원 개관

1. 정보보호의 목표 : 기밀성(C), 무결성(I), 가용성(A), 인증, 부인방지, 책임추적성
   ※ 인증 : 사용자 인증, 메시지 인증
2. 소극적 공격 vs 적극적 공격
3. 정보보호대책 : 관리적 보호대책, 물리적 보호대책, 기술적 보호대책
4. 시점별 통제(예방/탐지/교정)
   ※ 시간순서

❑ 보안에 자주 나오는 기관들

① TTA : Telecommunication Technology Association(한국 정보통신 기술 협회) – tta.or.kr
② KISA : Korea Internet & Security Agency(한국 인터넷 진흥원) – kisa.or.kr
③ NIST : National Institute of standards and Technology(미국 국립 표준 기술 연구소) – nist.gov

❑ 정보보호의 목표

❑ 정보보호 관리

① 정보보호 vs 정보보호 관리 시스템

↳ '관리'가 추가 → 2010년부터 공공기관 ISMS 인증 의무

↳ (KISA 주관)

② 정보보호 관리와 정보보호 대책

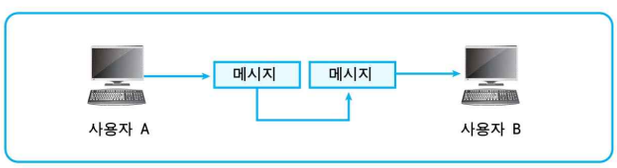
- 관리적 보호대책 :
- 물리적 보호대책 :
- 기술적 보호대책 :

※ 정보보호관리 → 생산성, 명성, 금전적 손실 : 비용/효익

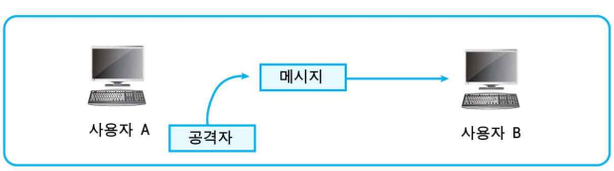
→ 수용가능한 위험 수준 결정(∵100% 완전 보안통제 불가능)

❑ 변조와 위조의 차이점

① 데이터 변조 : A가 생성하여 전송하는 데이터를 제3자가 가로챈 후 데이터의 일부 또는 전체를 변경하여 B에게 전송하는 경우이다. 이 상황에서 B는 수신한 데이터를 A가 전송한 데이터로 오인하게 된다.



② 데이터 위조 : A가 B에게 데이터를 전송한 적이 없는 상황에서 제3자는 A가 생성하여 전송한 데이터인 것처럼 위조한 후 B에게 전송할 수 있다. 이것은 데이터 변조와 유사하게 보이지만 A가 데이터를 생성하지 않았다는 점에서 데이터 변조와 다르다.



❑ 위험(Risk)

위험(Risk) = 자산 × 위협 × 취약점

↳Asset ↳Threat ↳Vulnerability → V.A.T(세금)

❑ 가장 약한 링크 원칙

보안에서는 'Chain Rule'이란 것이 있다. 쇠사슬 양쪽 끝에서 잡아당기면 가장 약한 부분이 끊어진다. 즉, 쇠사슬의 전체 강도는 가장 강한 부분이 아닌 가장 약한 부분에 따라 결정된다. 보안도 마찬가지다.

❑ Defense-in-Depth(약어 : D-i-D) : Multilevel(다단계) 보안

① COTS(Commercial off the shelf, 상용제품)는 완벽한 보안을 제공하지 못한다.

② 보안통제를 중복(overlap)

③ Work Factor의 증가를 유발 → 단점 : 복잡도↑

❑ 시점별 통제

❑ 정보보호의 목표

구분	특징	위협요소	공격종류	적극적/소극적	대응책	비고
기밀성	노출×, 인가된 것만 접근, 가로채기	도청, 사회공학	스누핑, 트래픽분석	소극적	접근통제,암호화	
무결성	불법생성, 변조, 위조	논리폭탄, 바이러스	변경, 가장, 재연, 부인	적극적	접근통제, 메시지 인증, (침입탐지, 백업)←변경후	해시함수
가용성	지체없이(즉시) 서비스	DoS, DDoS, 지진	DoS	〃	백업, 중복성 유지	BCP/DRP

- 인증 : 메시지 인증 → MAC, 사용자 인증 : 시스템 접근통제
- 부인방지 : 오리발 방지, SSL은 제공하지 못함. 종류 : 송신부인방지, 수신부인방지

↳ 기밀성, 상호인증, 무결성

❑ 암호학 파트 전체 학습 구조

1. 암호학의 개요
2. 대칭키 암호
3. 비대칭키 암호
   ※ 양방향 : 대칭키, 비대칭키
4. 해시함수와 응용
   ※ 일방향 : 해시함수
5. 전자서명과 PKI
6. 키와 난수

❑ 단원 개관

1. 암호시스템 분류
2. 암호 기법의 분류
   ⓐ 치환(substitution) vs 전치(transposition)
   ⓑ 링크 암호화 vs 종단간 암호화
   ⓒ 스트림 vs 블록
   ⓓ HW 기반 vs SW 기반
3. 암호공격의 분류
   ⓐ 암호문 단독공격(ciphertext only)
   ⓑ 기지평문 공격(known plaintext)
   ※ 기지 평문 공격 = 알려진 평문 공격
   ⓒ 선택 평문 공격(chosen plaintext)
   ⓓ 선택 암호문 공격(chosen ciphertext)

❑ 암호화 시스템

❑ 암호화, 복호화 키

키	=	키	: 대칭키 →

개인키	공개키	: 비대칭키 →

※ secret key(비밀키)와 private key(개인키)의 혼용

❑ 치환과 전치의 개념

※ 기법

1. 구성(조성비율 바꾸는 것) : Substitution {평문문자} ≠{암호문문자}
2. 배열(위치바꾸기) : Transposition, Permutation {평문문자} ={암호문문자}

❑ XOR (exclusive or)

0 xor 0 = 0 (짝+ 짝 = 짝)
0 xor 1 = 1 (짝+ 홀 = 홀)
1 xor 0 = 1 (홀+ 짝 = 홀)
1 xor 1 = 0 (홀+ 홀 = 짝)

❑ 링크 암호화와 종단간 암호화

구분	계층	암호화	암호화 범위	트래픽 분석	키 관리	암호화 주체
종단간 암호화	L7	• 송수신 말단에서 암/복호화 • 중계노드는 중계만	헤더(라우팅정보)는 암호화 하지 않음			사용자(S/W로 구현)
링크 암호화	L1~L2	• 중계노드에서 암/복호화 → 일시적 평문상태	헤더를 포함한 모든 데이터 암호화			ISP나 통신업자

❑ 하드웨어와 소프트웨어 암호시스템

구분	H/W기반	S/W기반
성능(속도)		
(도입)비용		
안정성		
방식		

❑ Steganography

① 메시지를 숨기는 기술 ex) 책상에 컨닝 메모

② 사례 : 저작권 보호 용도 ⇒ _____

③ 은닉채널(Covert channel)

④ 오사마 빈라덴이 사용(dead-drop(비밀 연락책)으로 사용)

※ 암호 → 크립토그래피(cryptography) : 메시지의 내용을 읽지 못하게 하는 기법
스테가노그래피(steganography) : 읽지 못하게 하는 것이 아니라 메시지의 존재 자체를 숨기는 기법

❑ DRM

① 디지털 미디어의 생명주기 동안 발생하는 사용권한 관리, 유통단계를 관리하는 기술

② 디지털 미디어의 불법 유통과 복제를 방지

③ 암호기술, 키관리 기술, 워터마킹 등 다양한 정보보호 기술 활용

④ 개념도

❑ 암호화 방법을 설계하는 목표

① Work Factor↑ : 공격자가 암호화 방법을 깨는데 걸리는 노력(리소스)

(= Work Function↑) ↳ 돈 or 시간

⇒ Computationally Infeasible ∝ 성능↓(속도가 떨어짐)

↳ 계산은 가능하지만 상당히 어려운.. → 설계 목표

② 암호를 깨는데 걸리는 시간과 비용이 많이 들게 설계하는 방법

ⓐ

ⓑ

ⓒ

#A&@ $\xrightarrow[\text{해독}]{100억/1년\ 소요}$ I Love You.

100억 〉〉〉 10억 ⇒ 설계가 잘되었다고 봄

공격비용 Data 값어치

키의 길이 $\begin{pmatrix} 64\,bits & \rightarrow & 2^{64}\,hr \\ 128\,bits & \rightarrow & ? \end{pmatrix}$

약 (2^{64})배의 효과가 있다.

❑ 암호화 시스템

양방향	대칭키	Stream 방식	동기식	OTP, FSR, LFSR, NLFSR, OFB, RC4	난수열(키스트림) 독립성 생성
			비동기식(자기동기식)	CFB	난수열 종속적 생성
		Block 방식	Feistel	DES, SEED	암호화 = 복호화
			SPN	Rijndael, ARIA	암호화 ≠ 복호화
	비대칭키	RSA, ECC, ElGamal, DH			기밀성, 인증, 부인방지
일방향	해시함수	MDC	해시함수	MD5, SHA-1, RIPEMD, HAS-160	무결성
		MAC	해시함수+대칭키	축소 MAC, HMAC, CMAC	무결성+인증
		전자서명	해시함수+비대칭키	RSA 전자서명, DSS, ECDSA, KCDSA	무결성+인증+부인방지

❑ 단원 개관

1. 대칭키 암호 역사
   ⓐ 고대 : scytale, 시저
   ⓑ 근대 : 제1,2차 세계대전, Shannon
   ⓒ 현대 : DES, 3DES, AES
2. 스트림 암호 vs 블록암호 → (DES → 3DES → AES)
   ↳ 동기적 : 독립적 ↳ Feistel
   비동기적 : 종속적 SPN
3. 블록암호의 모드
   ⓐ ECB ⓑ CBC ⓒ CFB ⓓ OFB ⓔ CTR
   ↳ 스트림

❑ 대칭키 암호(블록 암호, 스트림 암호)

대칭키 암호는 크게 스트림 암호와 블록 암호로 나눈다. 스트림 암호는 매 시각마다 암호화하는 방법이 변하는 암호를 말하며 블록 암호는 키가 고정되면 암호화 함수가 고정되어 암호화하는 방법이 동일한 암호를 말한다. 이러한 관점에서 스트림 암호는 내부에 메모리가 있는 것으로 보며 블록 암호에는 메모리가 없는 것으로 고려한다. 이러한 분류는 개념적인 것으로 통상 Vernam Cipher처럼 의사난수열을 생성하여 평문과 XOR하는 형식으로 암호화하는 것을 스트림 암호라 하며, Substitution Cipher처럼 평문을 소블록으로 나누어 각 소블록에 동일한 암호 함수를 적용시키는 것을 블록 암호라 한다.

❑ P박스와 S박스

① P박스

구분	입력(n), 출력(m)	역함수	비트수
단순(straight)			-
축소(compression)			줄이고자
확장(expansion)			증가시키고자

② S박스
- 대치 암호의 축소 모형
- 입력과 출력의 개수가 달라도 됨
- 역함수 존재(입력=출력) or 비존재 가능

❑ 혼돈(Confusion)과 확산(Diffusion)

① 엔트로피 개념으로 설명(무질서도 증가 하는 방향으로 변함)

② shannon의 정의

③ 혼돈 : 암호문과 키 사이의 관계를 숨긴다.

암호문 이용 ↛ 키를 찾음

좌절시킴

④ 확산 : 암호문과 평문 사이의 관계를 숨긴다.(평문의 통계적 성질을 암호문 전반에 퍼뜨려 숨김)

암호문 통계테스트 이용 ↛ 평문 찾음

좌절시킴

❑ 스트림 암호

① 현대 스트림 암호는 한번에 r 비트 생성

$$P = P_1P_2P_3 \ldots \quad C = C_1C_2C_3 \ldots \quad k = (k_1, k_2, k_3, \ldots)$$
$$C_1 = E_{k1}(P_1) \quad C_2 = E_{k2}(P_2) \quad C_3 = E_{k3}(P_3) \quad \ldots\ldots$$

※ key를 어떻게 생성하는지가 주된 관심사

② 분류

구분	특징	종류
동기식		
비동기식		

❑ 스트림 암호와 블록 암호

① 인터넷의 컴퓨터 통신 암호화를 위해서는 실제 블록 암호가 스트림 암호보다 더 많이 사용된다.

② 스트림 암호는 작고 빠르기 때문에 휴대폰이나 작은 임베디드 장치와 같이 컴퓨팅 능력이 적은 응용 분야에 적합하다. 대표적인 스트림 A5/1 암호는 GSM 휴대폰 표준의 일부로 음성 암호화에 사용된다. 그러나 RC4와 같은 스트림 암호는 인터넷 트래픽 암호화에도 사용된다.

③ 일반적으로 스트림 암호가 블록 암호보다 더 효율적으로 암호화를 수행한다. 소프트웨어에 최적화된 스트림 암호가 효율적이라는 것은 평문 한 비트를 암호화 하는 데 보다 적은 프로세서 명령어(또는 프로세서 사이클)가 필요하다는 의미이다. 하드웨어에 최적화된 스트림 암호가 효율적이라는 것은 동일한 데이터율로 암호화하는 데 블록 암호보다 적은 게이트(또는 보다 작은 칩 영역)가 필요하다는 것이다. 그러나 AES와 같은 현대 블록 암호도 소프트웨어에서 매우 효율적이다.

④ 스트림 암호의 암호화 함수와 복호화 함수는 동일하다.

❑ DES

1. 목적 : _____

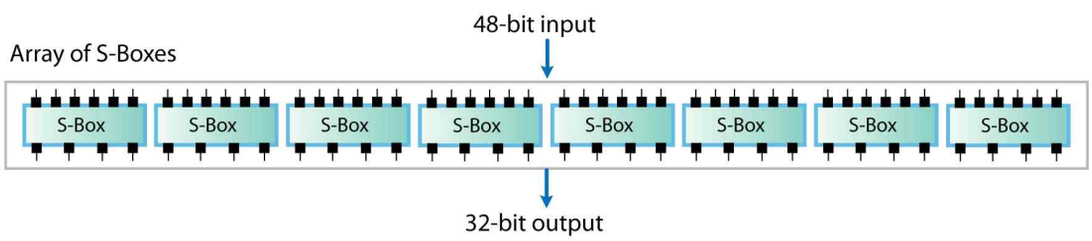
2. 방식 : _____

3. 사례 : Kerberos 버전 4(대칭 암호기법에 바탕을 둔 티켓 기반 인증 Protocol)
4. Round 횟수 :
5. Crack 이유 : 짧은 key size
6. key size
   - _____

   - 전체 : 64 bits

7. crack time 구하기
   WEP(Wired Equivalent Privacy) 40 bits 1 hr(work factor)
   DES crack time?
   풀이)

❑ S-박스



❑ Triple - DES

구분		암호-복호-암호	key 길이	Round
key	2개			
	3개			

❑ AES

① H/W, S/W에 구현 용이

② 사례 : smart card ⇒ 데이터 암호화

↳ 초경량 알고리즘임을 암시

③ SPN 구조 → 속도가 빨라짐(N/W 구조와 유사)

㉠ Linear Mixing layer : 안전성 높은 확산 효과(P-Box와 유사)

㉡ Non-Linear layer : S-BOX의 병렬 애플리케이션

㉢ Key addition layer : 중간 상태의 라운드 키와 XOR 연산

④ AES의 데이터 단위(DATA Units)

비트(bit), 바이트(byte), 워드(word), 블록(block), 스테이트(state) ⇒ 5가지 단위 사용 계산

↳블록의 나열 (행렬)

❑ AES의 라운드

DES와 달리 AES는 Feistel 구조가 아니다. Feistel 네트워크는 반복될 때 전체 블록을 암호화하지 않는다. 즉, DES의 경우 한 라운드에서 64/2=32 비트가 암호화된다. 하지만 AES는 하나의 반복에서 128비트 전체를 암호화한다. 그러므로 AES는 상대적으로 라운드의 수가 적다.

❑ SEED와 ARIA

→ SEED

→ ARIA

↳ Round(10,12,14) ↳ 국내 Round(12,14,16) ← AES에 비해 2회 추가됨.

❑ 블록 암호

❑ 모드(mode) 종류

구분	ECB	CBC	CFB	OFB	CTR
모양					
병렬처리					
초기화 벡터 ↳ 난수					
특징	• 간단 • 패턴 반복 • 재전송 공격 가능	• 암호문 블록이 파손되면 2개 평문 블록 영향 • 암호문 블록에서 비트 누락 → 평문 전체 영향(∵밀리기 때문)	• 재전송 공격 가능		• 카운터이용(비트+블록번호)
패딩	–				
Type of Result					

❑ 주요 대칭키 알고리즘

구분	키	블록	라운드수	구조	특징
DES	56	64	16	Feistel	안정성은 S-box에 의존, 16개의 48bit 라운드키
3DES	2개:112, 3개:168	64	48	〃	DES-EDE2, DES-EDE3
Rijndael	128, 192, 256	128	10, 12, 14	SPN	AES 최종 우승, 경쟁방식
SEED	128	128	16	Feistel	KISA+암호전문가
ARIA	128, 192, 256	128	12, 14, 16	SPN	한국, 경량환경, 하드웨어 구현에 최적화
IDEA	128	64	8	〃	유럽형(스위스)

❑ 단원 개관

1. 대칭키 vs 비대칭키

   ※ 대칭키 단점 : 키배송 문제 → 비대칭키 기법이 해결

2. 키배송 문제 해결

   ⓐ 사전공유 ⓑ KDC ⓒ D-H ⓓ 공개키

3. 공개키 알고리즘 상세 내용

   ⓐ D-H ⓑ RSA

4. 수학적 원리

   ⓐ 소인수 분해 : RSA, Rabin

   ⓑ 이산대수 문제, DH, ElGamal

❑ Diffie-Hellman 알고리즘

❑ Diffie-Hellman 키 교환

앨리스가 계산한 키 = $(G^B \bmod P)^A \bmod P = G^{B\times A} \bmod P = G^{A\times B} \bmod P$

❑ 공개키 사용 원칙

① 암호화키와 복호화키는 (　　)의 키쌍이어야 한다.

② 키는 암/복호화 중 (　　) 사용해야 한다.

③ 타인의 (　　)는 사용할 수 없다.

❑ 공개키 암호시스템의 활용

공개키 암호 시스템은 그 역사도 짧으며 현재 나와 있는 알고리즘도 그리 많은 편이 아니다. 또한 비밀키 암호 시스템보다도 데이터 암호화 속도가 매우 느리기 때문에 일반적으로 데이터 암호화에는 사용하지 않으며 키 분배나 디지털 서명 등에 많이 사용되고 있다.

❑ 공개키 암호

비대칭 암호방식은 트랩도어 일방향 함수를 이용하여 암호화키 E와 복호화키 D 사이의 관계가 독립이 되도록 설계되는 암호이다. 즉, 그들은 복호화키 D를 암호화키 E로부터 찾는 것이 계산적으로 불가능하도록 설계하여 암호화키 E를 공개하는 암호의 개념을 제시함으로써 키의 교환 없이 비밀 통신을 할 수 있도록 하였다. 즉, A가 서로 독립인 암호화키 E_A와 복호화키 D_A를 가지는 암호 방식을 설계하고 E_A를 등록한다. A에게 암호문을 보내려는 송신자 B는 등록된 암호화키 E_A를 이용하여 평문 P를 암호화한 암호문 C를 A에게 전송하면 A는 자신만이 가지고 있는 복호화키를 이용하여 암호문 C로부터 평문 P를 얻을 수 있다.

이와 같은 암호 방식을 복호화키 D_A가 암호화키 E_A로부터 쉽게 얻을 수 없다는 의미에서 비대칭 암호 방식(asymmetric cryptosystem)이라 한다. 또, 키 E_A를 등록한다는 것은 곧 공개한다는 의미로 이 암호계를 공개키 암

호 방식(public key cryptosystem)이라고도 한다. 한편, 키 E_A는 공개된다는 의미로 공개키 (public key)라 하고 키 D_A는 개인이 안전하게 관리해야 한다는 의미로 개인키(Private key)라고 한다.

이와 같이 공개키와 개인키가 독립인 공개키 암호 방식은 트랩도어 일방향 함수를 이용하여 설계된다.

❑ 비대칭키 암호화 시스템

〈알고리즘에 따른 분류〉

①　　: RSA, Rabin

②　　: Diffie-Hellman, ElGamal, DSA

↳ 원조, key 교환 전용　　↳ 디지털 서명 전용

③　　: ECC

〈목적과 보안서비스〉

① 목적 :

② 키 :　　(대칭키),　　(비대칭키)

↳ (개인키/공개키)

↳ N개 ↳ N개

③ 보안서비스 :

❑ 공개키 알고리즘에서 키 길이와 속도

3개의 알고리즘 계열의 계산적 복잡도는 대략 비트 길이의 세제곱 배로 증가한다. 예를 들어, RSA 서명 생성 소프트웨어에서 길이가 1024 비트에서 3076 비트로 증가하면 수행 시간은 $3^3 = 27$배가 느려진다. 현재 PC에서 수행 시간은 수 십 msec에서 100 msec 정도가 일반적이다. 그러나 공개키 성능은 휴대폰 또는 스마트카드와 같이 제약된 장치 또는 초당 많은 공개키 연산을 계산해야 하는 서버에서는 보다 심각한 장애 요인이 될 수 있다.

❑ RSA의 속도

일반적으로 특정 작업을 수행하는 데 소요되는 계산량은 MIPS년으로 나타낸다. 1 MIPS년은 1초당 100만 개의 명령어를 처리하는 컴퓨터로 1년간 수행해야 하는 계산량을 의미한다. 실제로 RSA-129의 소인수 분해는 5000 MIPS년이 소요되는 작업으로 인터넷을 통하여 전세계적으로 1600여 대의 워크스테이션, 대형컴퓨터 및 슈퍼컴퓨터를 연결하여 분담 작업하여 1994년 당시에 8개월 정도 걸렸다고 한다.

한편, 이와 같은 안전성에도 불구하고 암호화 및 복호화의 속도가 느리면 이용할 수 없다.

공개키를 선택한 후 개인키를 구하므로 보통 복호화 속도는 암호화 속도에 비해 느리다. 실제로 공개키가 8비트인 경우에 SPARC Ⅱ CPU에 의하여 측정된 RSA의 암호화 및 복호화의 속도는 다음과 같다.

	512 비트	768비트	1024 비트
암호화 속도	0.03	0.05	0.08
복호화 속도	0.16	0.48	0.93

일반적으로 RSA의 속도와 관련된 알고리즘의 계산복잡도를 살펴보면, n이 k비트의 수일 때 암호화 연산은 $O(k^2)$, 복호화 연산은 $O(k^3)$이고 키 생성은 $O(k^4)$이다. 이와 같은 암호화 연산 및 복호화 복잡도 차이는 서명과 인증에 응용된다.

❑ RSA 알고리즘

❑ RSA 알고리즘 예시

〈키 생성〉

① p=5, g=11

n=5×11=55

② ø(n)=(5−1)×(11−1)=40

③ gcd(e, 40)=1

e=3, 7, 9, …

if) e=3 일 경우

④ 3 × d ≡ 1 mod 40

∴ d=27

∴ 공개키는 (55, 3) 비밀키는 (55, 27)

〈암호문 구하기〉

① m=9(평문)인 경우 암호문? $m^e \bmod n$ $9^3 \bmod 55$

❑ 결정적(deterministic) vs. 확률적(probabilistic)

① 결정적 암호알고리즘: 암호키와 메시지가 같으면 항상 같은 암호문으로 암호화되는 암호알고리즘

② 확률적 암호알고리즘: 암호키와 메시지가 같아도 항상 다른 암호문으로 암호화되는 암호알고리즘

※ ElGamal은 확률적 암호 스킴이다. 즉, 두 개의 동일한 메시지를 암호화하더라도 그 결과는 동일하지 않다.

❑ ECC

① 키 길이

ECC	보안수준	RSA
160 bits	=	1024 bits
1 bit	〉	1 bit

- ECC는 RSA에 비해서 적은 키로 높은 보안 수준을 제공한다. : False(∵ 동일한 보안수준을 제공)

→ ECC는 RSA에 비해서 bit당 높은 보안 수준을 제공한다. : True

② 특징

ⓐ

ⓑ

ⓒ

③ 사례 : 스마트 카드→ 서명, 인증 용도

❑ 암호 해독을 위한 계산 비용에 따른 키 사이즈 비교

Symmetric Scheme (key size in bits)	ECC-Based Scheme (size of n in bits)	RSA/DSA (modulus size in bits)
56	112	512
80	160	1024
112	224	2048
128	256	3072
192	384	7680
256	512	15360

출처 : Certicom

❑ 대칭키 vs 비대칭키

- 대칭키 : 기호를 대체하거나 치환하는 것, 비밀키 : 기호열(비트열)
- 비대칭키: 숫자를 다른 숫자로 변환하는 것, 개인키 : 하나의 숫자 or 둘 이상의 숫자

→ 대칭키와 비대칭키는

❑ 하이브리드 암호

❑ 키 배송문제 해결

구분		특징	
키의 사전공유에 의한 해결		TA(Trusted Authority)이용 가능, $\frac{n(n-1)}{2}$ ($\because {}_nC_2$)	
KDC에 의한 해결		온라인 키분배(세션키), 커버로스, Needham-schroeder 프로토콜	
DH 키교환에 의한 해결		공개키의 효시, 이산대수문제, 키교환(키합의), 중간자공격가능 → 국대국 protocol로 해결	
공개키 암호에 의한 해결	RSA	소인수분해	키생성/암.복호화 이해(계산), OAEP, p,q조건
	ECC	타원곡선상 이산대수문제	160비트 ECC≒1024비트 RSA, 빠르다, (자원)효율적 → 스마트카드, 무선
	ElGamal	이산대수	암호문이 평문의 2배, RSA 활용과 비슷
	Rabin	소인수분해	이차잉여류, RSA와 유사

※ DSS : 서명 전용

PART 02 : 암호학 — 해시함수와 응용

❑ 단원 개관

1. 해시함수의 특징
   ⓐ 일방향 ⓑ 다대일 → 충돌
2. 해시함수의 보안적 요구사항
   ⓐ 역상저항성 ⓑ 두번째 역상저항성 ⓒ 충돌저항성
3. 해시함수의 종류 및 특징
   ⓐ MD5 ⓑ SHA ⓒ HAS-160
4. MAC - 키가 있는 해시함수
   ↳ 대칭키

❑ 해시함수의 일반적 개념

❑ 해시의 일방향성에 대한 예시(나눗셈을 이용한 해시값 획득)

① 먼저 12345**5**6789라는 수와 이것과 한 자리 수만 다른 12348**8**6789라는 수가 있다 가정해보자. 두 수를 다음과 같이 가운데를 기준으로 둘로 나누고 큰 수를 작은 수로 나눈다.

$\frac{12345}{6789}$ = 1.81838 2677861...

$\frac{12348}{6789}$ = 1.81882 4569115...

② 결과값은 각각 1.818...이다. 앞 6자리 숫자를 버리고, 나머지 값을 해시의 결과 값이라고 생각한다면 12345**5**6789의 해시 값은 2677861, 12348**8**6789의 해시 값은 4569155다. 하지만 두 해시 값만으로 해시 전의 원래 수를 알아내는 것은 불가능에 가깝다. 로직을 알더라도 버려진 1.81838과 1.81882를 알아낼 수 없기 때문이다.

③ 이처럼 해시는 로직을 알고 있을 경우 해시의 결과 값을 구하는 건 쉽지만, 그 해시 결과 값을 통해 해시를 생성하기 전의 원래 값은 알기는 어렵다. 그리고 값이 아주 조금만 다르더라도 해시 결과 값은 무척 상이하게 생성된다.

❑ 해시함수의 안정성 사항들 간의 관계

❑ 약한 충돌 내성과 강한 충돌 내성

: $h(x)=y$가 주어졌을 때, $h(x')=y$가 되는 다른 x' (단, $x \neq x'$)를 찾기 힘듦.

: $h(x)=h(x')$가 되는 입력쌍 x와 x' (단, $x \neq x'$)를 찾기 힘듦.

❑ Hash 알고리즘의 응용 분야

① 디지털 서명의 효율성 증대

② 중요정보의 무결성 확인

③ OS의 변동 유무 체크 : H-IDS 이용 → ⓐ log 이용 ⓑ Hash 이용

↳ 침입탐지 2가지 방법 중 하나

❑ 비둘기집 원리(Pigeon-hole principle)

↳ ( ) 발생

(N+1) 비둘기 $\xrightarrow{\text{넣는다}}$ (N)개 비둘기집

❑ 생일 공격

① $1-\frac{365\times364\times363\times\cdots(365-N+1)}{365^N}$ N=23명 있으면 $\frac{1}{2}$ 이상의 확률로 적어도 2명의 생일이 일치한다.

② 생일 역설의 가장 중요한 점은 충돌을 찾기 위해 해시하는 데 필요한 메시지의 수는 가능한 결과 값 수의 제곱근 즉, $\sqrt{2^n}=2^{n/2}$과 대략 동일하다는 것이다. 즉, 생일 공격에 의해 키 공간의 크기가 2^n이면 $2^{n/2}$의 계산량으로 해독될 수 있다.

❑ Hash

$\begin{cases} MDC(Modification\ Detection\ Code) \\ MAC(Message\ Authentication\ Code) \end{cases}$

❑ MDC

① _____ ② ③

❑ MAC

① _____ ② ③ _____

❑ MDC, MAC, 전자서명 Process

❑ MDC, MAC, DS

구분	특징	무결성	인증	부인방지
MDC	Hash	○	×	×
MAC	Hash+대칭키	○	○	×
DS	Hash+비대칭키	○	○	○

❑ 해시함수의 충돌

구분	동일 용어	설명	깨는 방법	포함관계	전자서명
역상저항성	프리이미지저항성	y=h(x) ⇒ x?	–	포함×	약 일방향성
두 번째 역상저항성	제2프리이미지저항성, 약한충돌내성	$h(x)=h(x')$ ⇒ 다른 입력값 $x'(\neq x)$	무차별공격 (전수공격)	∪ 충돌저항성은 두 번째 역상저항성을 보장	강 일방향성
충돌저항성	충돌회피성, 강한충돌내성	$h(x)=h(x')$ ⇒ 두 입력값 x, x'	생일공격		충돌회피성

❑ 단원 개관

1. 전자서명의 개념

ⓐ 무결성 ⓑ 인증 ⓒ 부인방지

※ 기밀성(x)

2. 전자서명의 제공기능

ⓐ 변경불가 ⓑ 서명자 인증 ⓒ 부인방지 ⓓ 위조불가 ⓔ 재사용불가

※ 서명자 인증(송/개 →송/공)

3. 전자서명 알고리즘

RSA, ElGamal, ECDSA, DSS, Schnorr, KCDSA

4. 특수 전자서명

부인방지, 은닉서명, 다중서명

5. PKI 개념

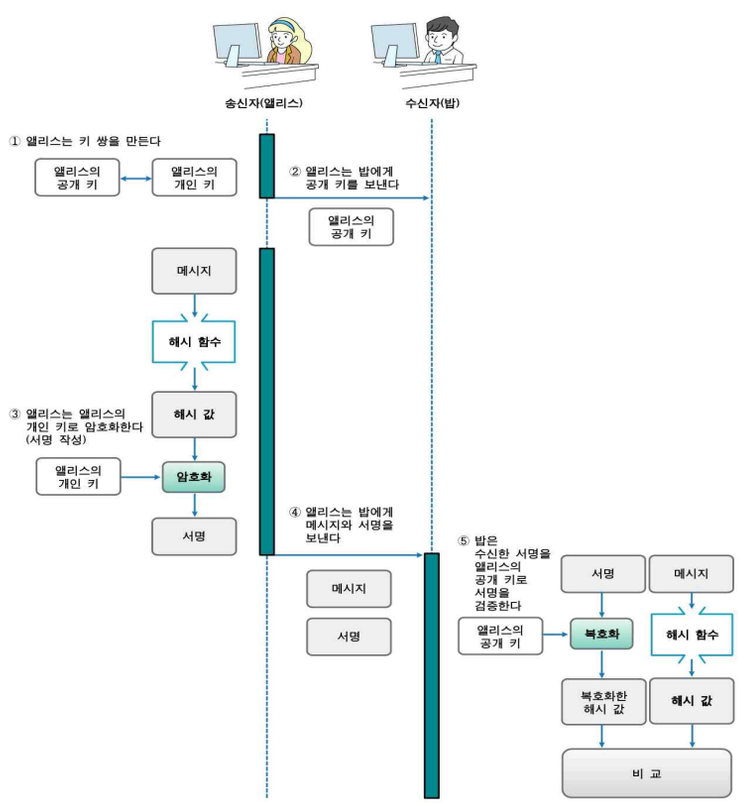
※ 공개키 알고리즘 → 중간자 공격 가능

PKI → 중간자 공격 막는다. 기밀성 제공

6. PKI의 구성요소

CA, RA, VA(검증기관), 디렉터리, X.509(인증서)

❑ 부가형 전자서명(시간적 흐름)



❑ 은닉서명(특수 전자 서명)

↳ Blind signature : ( ) 보장

❑ 인감도장

〈주택 매매〉

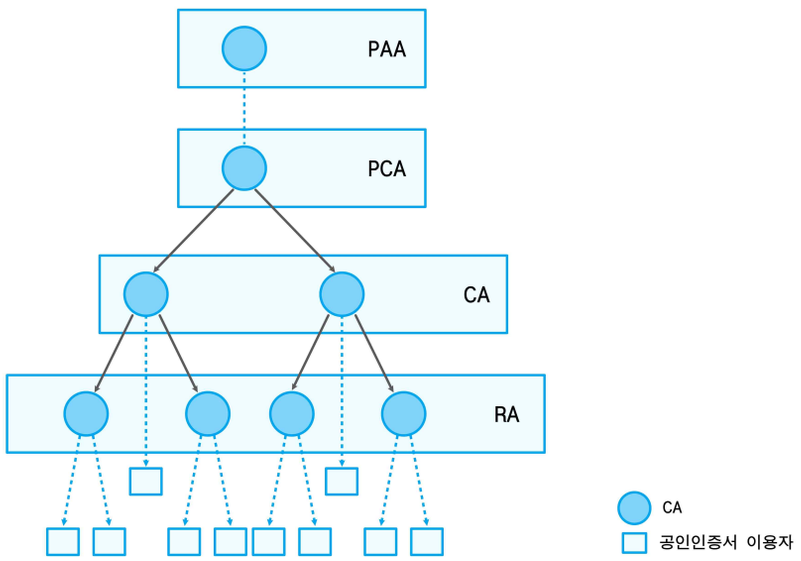
인감증명서 + 인감도장(개인키)

↳ 대한민국이 보증한다는 뜻(공개키 확인서)

※ 인증서는 인증된 공개키의 역할과 더불어 신원을 보증해주는 일종의 신분증 역할을 하기도 한다.

❑ PKI 개념 및 메커니즘

❑ 공인인증서 운영을 위한 계층 구조



① PAA(Policy Approval Authorities, 정책승인기관) : 공인인증서에 대한 정책을 결정하고 하위 기관의 정책을 승인하는 기관이다. 우리나라는 미래창조과학부가 담당한다.

② PCA(Policy Certification Authorities, 정책인증기관) : 기본 정책을 수립하는 기관으로, 우리나라의 KISA가 여기에 해당한다.

③ CA(Certification Authority, 인증기관) : PCA의 하위 기관으로 인증서 발급과 취소 등의 실질적인 업무를 하는 기관이다. yessign(금융결제원), NIA(한국정보화진흥원) 등이 이에 속하며, 상호 간 신뢰한다.

④ RA(Registration Authority, 등록기관) : 사용자의 신분을 확인하고, PKI를 이용하는 Appliation과 CA 간 인터페이스를 제공하는 기관이다.

❑ 전자서명 정리

구분	이론 근거	특징
RSA	소인수분해	RSA 아이디어 이용, 송신자 개인키/공개키 이용(∵ 공개키와 다름)
ElGamal	이산대수	RSA 서명 길이의 2배(암호문=2×평문), 속도 느림
Schnorr	〃	ElGamal에 기반 → 서명크기 줄임
DSS	〃	ElGamal 서명 개량(서명과 검증에 소요되는 계산량 획기적으로 줄임), 미국전자서명 표준, 서명전용
ECDSA	타원곡선상의 이산대수	DSA를 타원곡선위의 알고리즘으로 변환
KCDSA	이산대수	HAS-160(해시함수)이용, 한국
EC-KCDSA	타원곡선상의 이산대수	SHA-224이상(추가), 2014년 개정, 한국, KCDSA를 타원곡선위의 알고리즘으로 변환