

1. 보안의 3 가지 목표 기술하시오.

구 분	내 용
Confidentiality	· 비인가자에게 정보의 노출을 방지하는 것
Integrity	· 비인가된 정보의 변경을 방지
Availability	· 인가된 사용자가 원할 때 언제든지 시스템 서비스를 이용

2. Identification vs Authentication 차이를 기술하시오.

(각각에 알맞은 예제 1 개씩 서술)

- 식별 : 여러명 중에 한명을 식별하는 것.(1:N) (예 : 지문검색)
- 인증 : 식별한 한명이 그 사람이 맞는지 확인하는 것(1:1) (예 : 주어진 ID(신원)를 통해 해당 신원을 증명하는 방법(로그인))

*authorization : 인증과정을 거친 특정 사용자가 시스템에 의해 통제되는 보안 자원에 접근하기 위해 어떤 레벨을 가져야 하는지 결정하는 것

3. Reliability vs. Security

Why We Need Dependability? <ul style="list-style-type: none">■ Critical Infrastructures<ul style="list-style-type: none">■ Infrastructure systems for which continuity is so important that loss, significant interruption or degradation of service would have grave social consequences. <p>(Source : National Infrastructure Security Coordination Center, UK)</p>	Reliability v.s Security <ul style="list-style-type: none">■ Dependability = Reliability (Accidental Failures) + Security (Intentional Failures)■ Reliability and security are often strongly correlated in practice
--	---

소프트웨어공학(SE, Software Engineering)은 시스템을 설계 및 구현하는데 오류를 줄여주며, 이를 통해 Reliability 를 만족할 수 있다. 또한, 어떠한 시스템이 알려진 취약점에 모두 견디면, Security 를 만족한다고 할 수 있다. 이 두 가지를 모두 만족하면, Dependability 를 만족하는 시스템이다. 이러한 dependable 한 시스템은 금융, 정부기관, SCADA 시스템과 같은 critical infrastructure 로 분류되며, 이러한 중요 기반시설은 continuity 하여야 한다. 이는, dependability 를 만족하는 시스템이 필요로 한 이유이다.

4. 무어의 법칙을 이용하여 AES 128/192/256 이 2050 년까지 안전한지에 대해 논하시오.

무어의 법칙 : 18 개월마다 현재의 컴퓨팅 능력이 2 배가 되는 것.

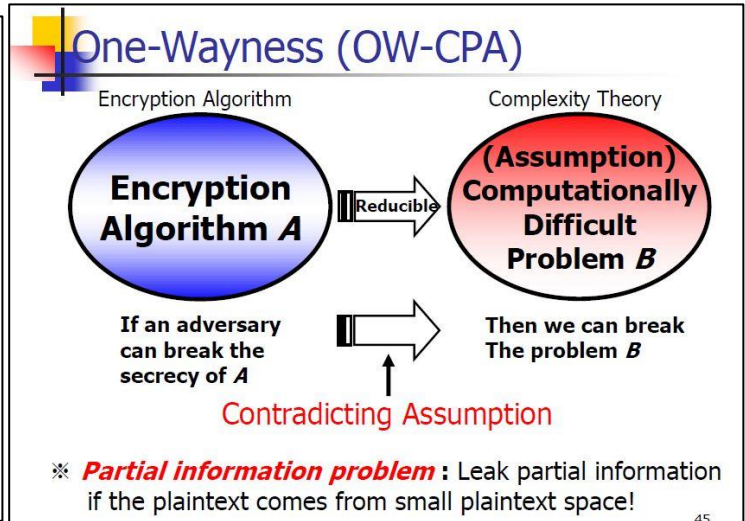
예를 들어, 현재의 컴퓨팅 능력이 2^{80} 이라 가정할 때, 18 개월 마다 2 배의 컴퓨팅 능력을 가지게 되니까, 2014 년 1 월을 기준으로 2015 년 6 월에 컴퓨팅 능력은 2^{81} 이 된다. 이를 쫓 계산해보면, 2050 년에는 AES 128 의 안전성을 가진 암호 시스템은 안전하지 않게 된다. AES 192/256 은 2050 년까지 안전하다.

5. OW-CPA 증명하라 (2014. 08 월 종시 문제)

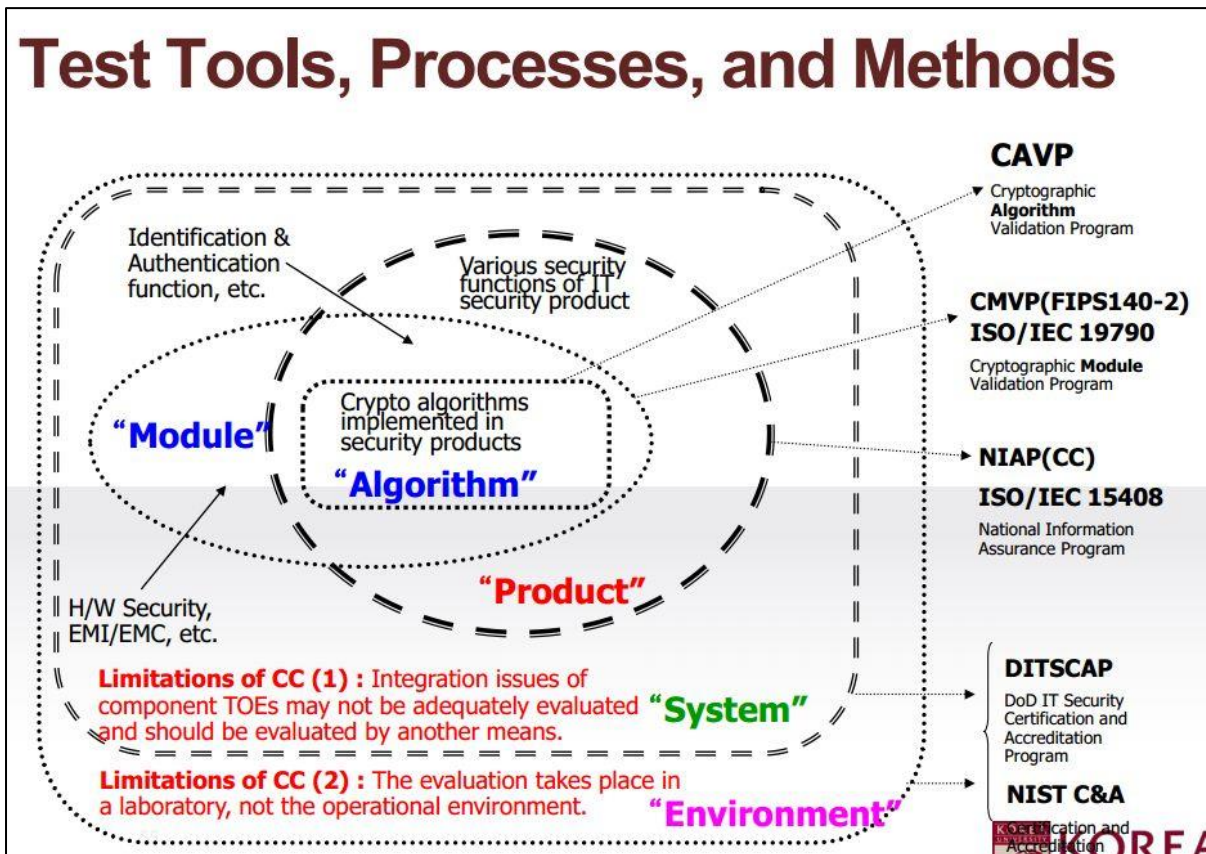
OW-CPA Example : Rabin Scheme

- **Private Key** : $p = q = 3 \pmod{4}$
- **Public Key** : $n = pq$
- **Encryption** : $C = M^2 \pmod{n}$
- **Decryption** :
 - $m_1 = C^{(p+1)/4} \pmod{p}$, $m_2 = (p - C^{(p+1)/4}) \pmod{p}$, $m_3 = C^{(q+1)/4} \pmod{q}$, $m_4 = (q - C^{(q+1)/4}) \pmod{q}$.
 - $a = q(q^{-1} \pmod{p})$, $p = p(p^{-1} \pmod{q})$.
 - $M_1 = (am_1 + bm_3) \pmod{n}$, $M_2 = (am_1 + bm_4) \pmod{n}$, $M_3 = (am_2 + bm_3) \pmod{n}$, $M_4 = (am_2 + bm_4) \pmod{n}$.
 - M is one of $\{M_1, M_2, M_3, M_4\}$

46



6. Draw CAVP, CMVP, CC & C&A



한국 : 사용역역이 민간인가 공공인가로 구분하여 평가한다.

미국 : 제품과 자료의 가치로 구분하여 평가한다.

→ 그림 통째로 외우시면 됩니다. CMVP 에서 한국은 K-CMVP 를 사용하는 차이점이 있습니다.

(빨간색 설명까지 암기할 필요는 없습니다. 약어는 암기하여 쓸 필요까진 없을 것 같은데 참고로 저는 암기하여 작성하였습니다.)

- CAVP : 보안 제품에서 구현되는 암호 알고리즘에 대한 평가

- CMVP : 암호 모듈에 대한 평가

- CC : 보안 제품에 대한 평가

- DITSCAP : 보안 시스템의 운영, 유지보수에 대한 평가

- NIST C&A : 보안 환경의 운영, 유지보수에 대한 평가

6-1) CAVP, CMVP, CC, C&A, DITSCAP 및 국내와 미국 평가제도 비교설명

- CAVP : 암호화 관련, 암호 알고리즘 중심(미국 표준 → ISO)
 - CMVP : 암호 모듈 중심으로 테스트 및 제대로 구현되었는지 Validate 한다.
CAVP 는 CMVP 를 위해 사전에 실시해야 한다. Vender 가 제품을 개발하면 국가에서 지정한 Testing LAP 에서 기본적인 TEST 를 실시하고, NIST 나 CSEC 에서 Validate 한다(Certificate 제공).
그리고 User 는 이를 구매한다.
 - CC : 공통평가 기준, 정보보호 기능이 들어가 있는 모든 IT 제품을 평가한다.
 - CC 제한사항 : 실험실 환경에서 테스트 하는 것으로 현장과 환경이 맞지 않으면 보안사고가 발생하고, 단일제품별로 평가하는 개념으로 시스템 전체에 대한 평가하는 것이 아니다.
- ***** Information Assurance Point *****
- DITSCAP : System 중심으로 보안성 평가, 순정부품으로 만든 자동차는 안전할까?
 - NIST C&A : Environments 중심의 보안성 평가, 제품자체는 안전성이 있지만, 전쟁같은 복잡한 환경에서도 안전할까?

7. Security Goal

구분	내용
One-way	<ul style="list-style-type: none"> · 공격자가 암호문으로부터 평문(전체)를 해독하는 것이 불가능 · 부분 해독이 가능하다. 대통령 5시... 이런 것 조합 가능 하지만 OW 관점에서는 상관 없다.
Semantic Secure (Indistinguishability)	<ul style="list-style-type: none"> · 어떠한 암호문이 1비트의 정보도 노출이 안된 것임 (No information leakage) · 같은 평문에는 항상 다른 암호문이 나와야 한다. · 해결책 : random padding을 붙여야 한다. 그럼 모두 값이 바뀜 <div data-bbox="481 1312 1104 1585"> <p>※ Goldwasser Micali 가 증명 한 내용</p> <div> <div> <p>Random Padding → Polynomial Security (IND-CPA)</p> <p>Polynomial Security (IND-CPA) ← Ciphertext leaks NO information!</p> </div> <p>↕</p> <div> <p>Random한 Padding을 붙이면, Message Space가 작아도 안전하다</p> <p>Message Space가 작아도 안전하면 어떠한 부분 정보도 노출되지 않는 Schema을 만들 수 있다.</p> </div> </div> <p><주> Polynomial Security : Message Space 가 작아도 안전하다. Semantic Security : 어떠한 부분 정보도 노출되지 않는다.</p> </div> <p>→ Polynomial Security = Semantic Security</p>
Non-Malleability	<ul style="list-style-type: none"> · 반드시 암호문으로 하는 것이 해독밖에 없는가? · 실제 해독은 안하지만 암호문의 변경을 통해 목적 달성 · 해결책 : 고정 padding을 붙인다 → IND와 상충된다. 그래서 RSA-OAEP 같은 것을 사용한다. Random padding 과 Fixed padding을 동시 붙인다.

8. Non-repudiation, Availability, Accountability, Integrity, Confidentiality 용어 설명하시오.

9. Reference Monitor ?

- 접근 통제를 감시하는 것
- Security Kernel: Reference monitor 를 구현한 것
- TCB(Trust Computing Base): Security Kernel + 다른 부가 보안 장치
- kernel 에 넣기 위하여 꼭 필요한 기능만 탑재하여 초소형화 해야 함
- 즉, 무엇을 탐지할 수 있고 무엇을 탐지할 수 없는지 명확히 정의하고 간결해야 함

10. DAC(Discretionary Access Control) vs. MAC(Mandatory Access Control)

(나을 확률 높음)

구분	DAC	MAC
특징	<ul style="list-style-type: none"> · 자원의 접근을 사용자 계정에 기반 (Need to know) · 사용자는 자원과 관련된 ACL이 변경되어 권한을 부여받는다. · Data Owner가 누가 그 데이터에 접근할 것인지 결정한다. 	<ul style="list-style-type: none"> · 자원에 대한 접근권한을 관리자가 부여한다. · 자원에 대한 접근은 사용자의 보안등급과 주어진 동안에 대상의 보안레벨에 기반 · Multi-level Security : 보안등급을 계층화 해서 나누어 놓은 것 → 최초 Formalizing : BLP Model (No read up, No write down)
제한사항	<ul style="list-style-type: none"> · Identity 도용시 자료 유출이 가능하다. (A가 B, C에 권한을 주면, B, C가 다른이에게 자료를 줄 수 있다.) - 강력한 통제가 필요한 곳에서는 쓸 수 없음 	<ul style="list-style-type: none"> · 객체 단위의 세밀한 권한 설정 불가 · Security Level 개수 : $m \cdot (2^n)$ - Security level : m - Categories : n
비고		<p>MLS(Multi Level Security)</p> <ul style="list-style-type: none"> - partial ordering 기반(lattice 구조)으로 강제적 접근 제어 MAC 모델의 예 - 서로 다른 보안 등급을 가지는 두 객체에 대하여, 한 주체가 두 객체를 모두 읽기 위해 반드시 가져야 하는 최소한의 비밀 취급 권한은 무엇인가? - 서로 다른 비밀 취급 권한을 가지는 두 주체에 대하여, 한 객체가 두 주체에게 읽혀지면서 가질 수 있는 최대한의 비밀 등급은 무엇인가?

11. BLP(Bell-Lapadula Model) ? (나올 확률 높음)

· 최초의 MLS 을 Formalizing 한 모델 → 최초 수학적으로 증명됨. → - 자연어로 쓰인 정책: 모호함, 불일치, 누락

Simple Security Property (No Read Up)	<ul style="list-style-type: none"> • A state (b, M, f) satisfies the SS-property if <ul style="list-style-type: none"> • $\forall (s, o, a) \in b$, such that $a \in \{\text{read}, \text{write}\}$ • $f_O(o) \leq f_S(s)$ • I.e. a subject can only observe objects of lower classification <p>Subject는 단지 자신보다 동급이거나 낮은 레벨의 Object를 읽을 수 있다.</p>
* - Property (No Write Down)	<div> <p>■ *-Property (Star-Property)</p> <ul style="list-style-type: none"> • A state (b, M, f) satisfies the *-property if <ul style="list-style-type: none"> • $\forall (s, o, a) \in b$, such that $a \in \{\text{append}, \text{write}\}$ • $f_O(s) \leq f_O(o)$ • and <ul style="list-style-type: none"> • if $\exists (s, o, a) \in b$ where $a \in \{\text{append}, \text{write}\}$, • then $\forall o', a' \in \{\text{read}, \text{write}\}$, such that $(s, o', a') \in b$ • $f_O(o') \leq f_O(o)$ </div> <p>Subject 는 자신보다 동급이거나 높은 레벨의 Object에 쓸 수 있다. 그리고, Writing 중에는 그 Object level 보다 높은 레벨을 읽을 수 없다.</p>

장점 for BLP	단점 for BLP
<ul style="list-style-type: none"> - 최초로 Formalizing 된 것이다. - 여기서 언급된 state machine model 로 다른 모델을 언급 가능하다 : Biba 	<ul style="list-style-type: none"> - 기밀성 위주 초점(Not Integrity) - Change access right에 대한 언급이 없다. - Covert Channel이 존재 : Information Flow는 Security Mechanism 으로 컨트롤이 안됨 <p>-----</p> <p># covert channel</p> <ul style="list-style-type: none"> - 설계 당시의 입출력 통로 외의 통로 - 높은 레벨의 보안 등급을 가지는 object를 외워서 나가거나, 사진 찍어 나가는 경우 - 높은 레벨의 비밀취급인가를 받은 subject의 컴퓨터가 악성코드에 감염되어, 높은 레벨의 object를 낮은 레벨의 비밀취급인가를 받은 subject에게 몰래 전송하는 경우

12. Biba Model (나올 확률 높음)


- 무결성(내용의 진위성) 측면에서의 접근 제어 모델을 정의
- 무결성의 관점에서 정보의 흐름이 아래로 향해야 함

Simple Integrity Property (No Write Up)	<ul style="list-style-type: none"> • Corresponds to ss-property in Bell-LaPadula • If subject s can modify object o, then $f_O(o) \leq f_S(s)$. • no write-up <p>Subject는 자신보다 동급이거나 낮은레벨의 Object에만 변경 가능하다</p>
Integrity * - Property (No Read Down)	<ul style="list-style-type: none"> • Corresponds to *-property in Bell-LaPadula • A subject s can read an object o only if $f_S(s) \leq f_O(o)$ • No read down <p>Subejct는 자신보다 동급이거나 높은레벨의 Object만 읽을 수 있다.</p>

13. RBAC(Role-based Access Control) (나을 확률 높음)

- Subject-Object 외에 Role 이라는 개념을 도입
- Hierarchical RBAC : Role 에 계층을 만들어 운용
 - General role hierarchy : 상급자는 하급자의 모든 역할을 담당(Multiple inheritance)
 - Limited role hierarchy : 한 두명에 제한된 인원에게만 역할을 넘김
- Constrained RBAC
 - Static Separation of Duty : 상호배제 효과(감사자, 은행원 관계)..
Role 이 충돌되는 경우에는 둘이 양립할 수 없다.
 - Dynamic Separation of Duty : 충돌되는 역할(Role)을 동시에 수행할 수 없다.
감사하는 기간에는 은행원이 하는 역할을 할 수 없다.
- Consolidated RBAC : Hierarchical + Constrained RBAC

14. Access Control Structure

Access Control Matrix	<div><div> Capabilities & ACL</div><div>■ Example</div><table><tr><td></td><td>R₁</td><td>R₂</td><td>R₃</td><td>R₄</td></tr><tr><td>S₁</td><td>rw</td><td>rwX</td><td></td><td></td></tr><tr><td>S₂</td><td></td><td>X</td><td>rwX</td><td>rwX</td></tr><tr><td>S₃</td><td>rwX</td><td>r</td><td></td><td>r</td></tr></table><div><div>Capabilities:</div><div>Access Control lists:</div></div><div><div>S₁: {(R₁, rw), (R₂, rwX)}</div><div>R₁: {(S₁, rw), (S₃, rwX)}</div></div><div><div>S₂: ...</div><div>R₂: ...</div></div></div>		R ₁	R ₂	R ₃	R ₄	S ₁	rw	rwX			S ₂		X	rwX	rwX	S ₃	rwX	r		r
	R ₁	R ₂	R ₃	R ₄																	
S ₁	rw	rwX																			
S ₂		X	rwX	rwX																	
S ₃	rwX	r		r																	
Rule based Access Control	<div>· MAC처럼 구분되어 있음</div> <div>· 어떤 룰을 미리 규정해 놓고 Subject와 Object 간에 일어나는 일을 확인, 통제한다</div> <div>· Firewall에서 미리 Rule을 규정하는 것</div>																				
Constrained User Interface	<div>· User의 입력을 제한한다</div>																				
Context Dependent Access Control	<div>· Object에 대한 접근은 Object의 Contents에 달려 있다. E-mail이나 DB에서 특정 단어 검색 같은 것</div>																				

15. C&A ?

- Certifying(인증): 주어진 환경에서 시스템이 안전하다는 인증
- Accreditation(인정): 제품의 lifecycle 이 끝날 때까지 계속적으로 모니터링해서 시스템에 문제가 없다는 것을 인정
- FISMA(법)에 의해 명시된 요구사항을 만족시키기 위해 사용되는 방법론: DITSCAP, NIACAP, NIST C&A

16. One-Way, Semantic Secure, Non-Mellability 용어 설명

구분	내용
One-way	<ul style="list-style-type: none"> · 공격자가 암호문으로부터 평문(전체)를 해독하는 것이 불가능 · 부분 해독이 가능하다. 대통령 5시... 이런 것 조합 가능 · 하지만 OW 관점에서는 상관 없다.
Semantic Secure (Indistinguishability)	<ul style="list-style-type: none"> · 어떠한 암호문이 1비트의 정보도 노출이 안된 것임 (No information leakage) · 같은 평문에는 항상 다른 암호문이 나와야 한다. · 해결책 : random padding을 붙여야 한다. <p>그럼 모두 값이 바뀜</p> <div data-bbox="437 568 1059 837" data-label="Diagram"> <p>※ Goldwasser Micali 가 증명 한 내용</p> <p>Random Padding → Polynomial Security (IND-CPA) → Semantic Security</p> <p>Ciphertext leaks → NO information!</p> <p><주> Polynomial Security : Message Space 가 작아도 안전하다. Semantic Security : 어떠한 부분 정보도 노출되지 않는다.</p> </div> <p>→ Polynomial Security = Semantic Security</p>
Non-Malleability	<ul style="list-style-type: none"> · 반드시 암호문으로 하는 것이 해독밖에 없는가? · 실제 해독은 안하지만 암호문의 변경을 통해 목적 달성 · 해결책 : 고정 padding을 붙인다 <p>→ IND와 상충된다. 그래서 RSA-OAEP 같은 것을 사용한다. Random padding 과 Fixed padding을 동시 붙인다.</p>

17 (나올 확률 높음)

(a) 접근제어에서 capability 란?

(b) explain an advantage of access control list over capability lists

(c) Explain an advantage of capagility list over access control list

답)

(a)

- 주체 S_i 와 객체 O_j 의 교점으로 만나는 매트릭스의 접근권한을 기술 한 것을 의미한다.

(b)

- ACM 에서 행을 중심으로 접근권한을 리스팅함,
- 자원별로 객체의 권한을 각각 설정하여 저장함.
- 자원별로 주체의 접근권한을 설정 가능,
- 윈도우 NTF 나 F/W, 라우터 등에서 많이 사용

(c)

- ACM 에서 행을 중심으로 접근권한을 리스팅함,
- 주체가 각 자원에 대해 어떠한 접근권한을 가질 수 있는지 설정가능,
- 커버로스 안드로이드 등이 대표적

Question 1. Access control

(a) In access control systems, what is a *capability*? [10%]

A capability is a piece of data possession of which proves authorisation to access a resource.

(b) (i) Explain an advantage of access control lists over capability lists. [5%]

Access control lists make it easy for an administrator to see who has access to a given resource. (This is harder with capabilities.)

(ii) Explain an advantage of capability lists over access control lists. [5%]

Capabilities may be transferred offline between users. This is generally not possible with access control lists.

18. We consider 1 and 0 the two possible ballots for an election. A server publishes his public RSA key (N, e) . Each voter encrypts his vote 0 or 1, as $\text{RSA}(N, e) 0$ or $\text{RSA}(N, e) 1$, respectively. At the end of the election the server decrypts all received messages and counts the votes. Show how an attacker eavesdropping on the network can learn everybody's vote.

(나을 확률 높음)

답)

이에 가능한 공격방법은 NM(Non-Malleability), IND(Indistinguishability)가 존재한다.

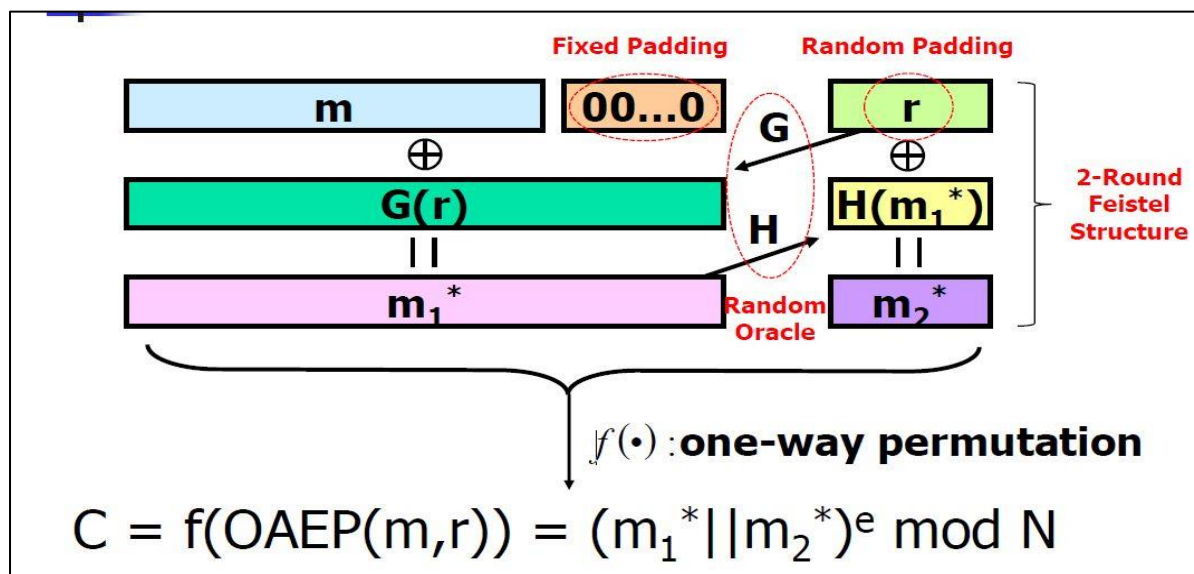
- IND 라는 Security Goal 은 암호문을 보고 공격자가 평문 M 을 어떤 것이었는지 구별 불가능해야 한다는 속성이지만, IND-CCA 공격을 통해 (임의의 난수 값을 추가하여 연산) 얻어진 C 값으로 평문을 유추할 수 있게 된다.

- NM 이란 Security Goal 은 공격자가 암호문에 대한 변조를 불가능하게 하여야 한다는 속성이지만, 암호문 값에 임의의 값을 추가로 연산함으로써, 전자투표의 무결성을 해칠 수가 있게 된다.

18-1) Propose a solution in order to avoid the above attack

RSA-OAEP 를 사용하면 됨(NM – Fixed Padding, IND – Random Padding)

- Random padding과 Fixed padding을 함께 사용하여, indistinguishability와 Non-Malleability를 달성함



19. (나을 확률 높음) Suppose that a receipt containing payment information has security level(secretary, personal),

A prescription for antibiotics has security level(Doctor, Emergency)

The list of medical tools necessary for an operation has security level(Nurse, Operating room) and the file containing the home address of patients in the hospital has security level(secretary, emergency).

Place all these documents (in the paragraph above) on the preceding lattice.

Dave- the surgeon has clearance (Doctor operation room)

Nancy- Nurse has clearance(Nurse, emergency room)

Shari- the secretary has clearance(Secretary, emergency)

Paul - the patient has clearance(Patient, personal)

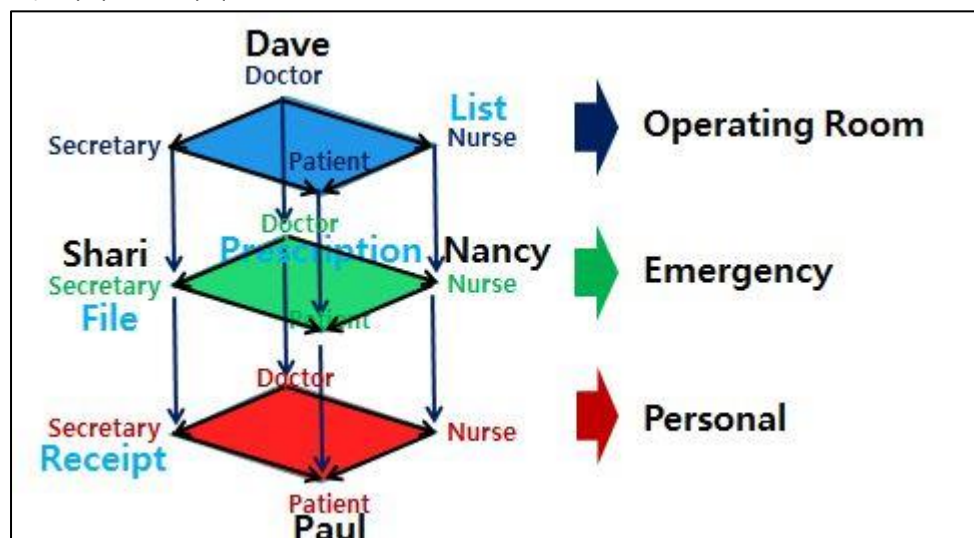
BLP, if the following actions are allowed, explaining each time why that is.

- (a) Dave writes on the list
- (b) Nancy read the files
- (c) paul writes on the prescription
- (d) shari read the receipt

BIBA

- (a) Dave writes on the list
- (b) Nancy read the files
- (c) Dave writes on the file
- (d) shari read the prescription

답) 래티스 그리기



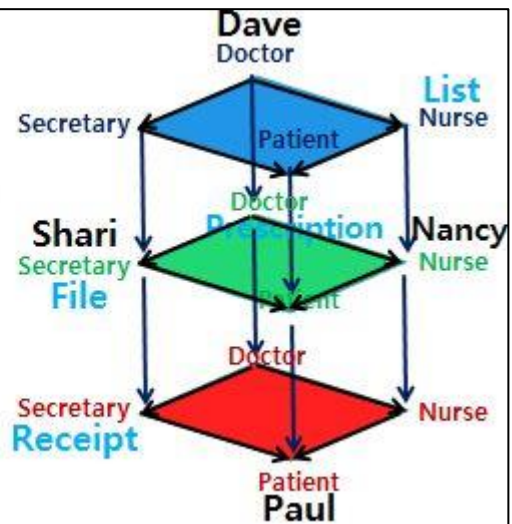
답) BLP 와 BIBA 일 때 WRITE/READ 가능여부

▪ In BLP Model

- Dave write on the List? **No**
- Nancy reads the File? **No**
- Paul writes on the Prescription? **Yes**
- Shari reads the receipt? **Yes**

▪ In Biba Model

- Dave write on the List? **Yes**
- Nancy reads the File? **No**
- Dave write on the File? **Yes**
- Shari reads the prescription? **Yes**



20. (나올 확률 높음)

Hospital patient record system provide login account for nurse.

It is desire to implement the following policy:

- 1) When a nurse register a new patient, the nurse is granted access to the patient records for a period of 90 days.
- 2) A nurse passing the right to access a patient record and give that right to another user. This may be done office.

The implement this policy, the system works as follow, when an nurse register new patient, a capability to access the patient record for the following 90 days in generated.

The nurser stores it on a usb stick, and may copy it onto other usb stick to give to other user.

When a user attempt to access patient records, she is promoted to upload the relevant capability.

The capability has the following format

patient- id, issue-date, HMAC(k, patient-id, issue-date)

Where HMAC(k,...) denote a suitable keyed hash function with key K. The key K is a secret key.

Known only the patient record system, Any user in possession of this capability is able to access the records of the patient with patient- id, provided the date is with in 90 days after issue date

(a) Suppose nurse register a patient and receives such a capability.

A passes it to B, B passes it C, and C passes it to D. Is D able to use the capability?

(b) In order to stop long-lived capabilities being distributed widely, the hospital decides to adopt the policy that the nurse that initially registers the patient will have access to the record for 90 days, as before, if she passes the capability to any other user. the validity should be 10 days from it issue-date.

Explain a new format for capability which would support this new policy.

답) (a)

A 가 D 까지에게 파일이 이전된 이후에도 Issue-date 에 의해 90 일간 접근권한이 유지되므로 D 는 사용가능하다.

답) (b)

Nurse ID, Patient ID, Issue-date, hmac(K,(patient-id, issue-date))로 설정 후 if(90 일 사용가능) else

21. (나올 확률 높음)

1) 레티스를 만족하려면?

2) 그리고 올바른 레티스 그림을 고르고 이유를 말하시오.

1) 답

"Lattice 구조를 갖는다" -> 유한체의 어떤 set(변수)이든 최대치(GLB)와 최소치(LUB)를 구할 수 있어야 한다는 뜻이다.

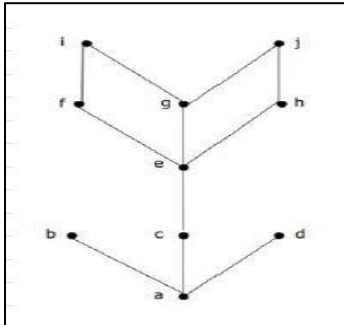
Lattice 구조란 다음의 두가지 조건을 만족해야 함을 정의하고 있다.

■ Partially ordered set (S, \leq) and two operations :

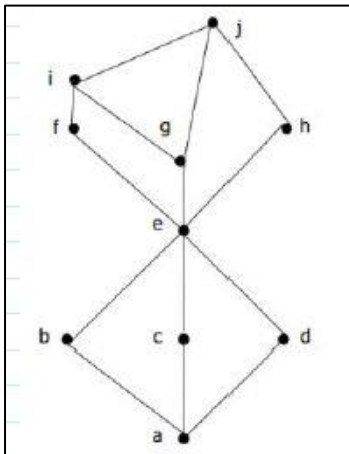
- greatest lower bound (glb X)
 - Greatest element less than all elements of set X
- least upper bound (lub X)
 - Least element greater than all elements of set X

■ **Lattice** : A finite set together with a partial ordering on its elements such that for every pair of elements there is a least upper bound and a greatest lower bound.

2) 답



이것은 Lattice 구조가 아니다. b와 c가 서로 접근할 수 있는 최대치와 최소치가 존재하지 않기 때문이다. (어떤 변수들 간에 접근되지 못하는 곳이 있으면 안 된다)



즉, 위와 같은 그림으로 변환될 수, 모든 set(변수)들은 접근제어가 가능하게 된다. 그러므로 이와 같은 lattice 구조가 될 때 접근제어 정책이 타당하게 수립 될 수 있다