<대칭키 암호>

[블록 암호]

I) 구조

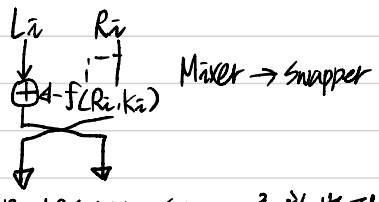
i) Feistel

① 보안 강도의 핵심

- 평문 블록 길이 ≥ 64 bit
- 키 길이 ≥ 128 bit
- 라운드수 ≥ 16 round (최소 3라운드, 짝수 권장)

② 특징

- 입력을 좌·우로 나누어 처리



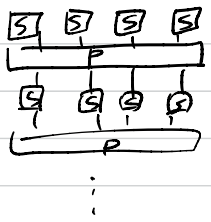
- 최종 라운드에서는 Swapper를 안 씀.
- 복호화 과정 = 암호화 과정, 단, 라운드키를 역순으로!

ii) SPN

① 보안 강도의 핵심

- S-box 와 P-box를 중첩함

② 특징



- 중첩구조
- 병렬처리↑

II) 블록 암호 공격 기법

① 차분 ② 선형 ③ 전수 ④ 통계 ⑤ 수학

[스트림 암호]

* 보안 강도 핵심
  - 키 길이↑ ≥128 bit
  - 키 주기↑
  - 키 랜덤성↑
  - 키가 평문과 독립적으로 생성 ∴ 암호문-키 증가한 셈.

I) 동기식 스트림 암호 : OTP, FCR, LFSR, NLFSR, OFB모드, CTR모드

i) OTP

* 특징
  - 무조건 안전
  - XOR
  - 1 bit / 1 time

ii) FCR

* 특징
  - OTP 절충안
  - H/W 구현이 더 수월
  - Feedback 레지스터 + Shift 레지스터

iii) LFSR

* 특징
  - H/W 구현 용이

- 키가 평문과의 관계를 가짐

II) 비동기식 스트림 암호 : CFB 모드

* 키 스트림이 평문 or 암호문과 관계 有.

*공개키 암호 알고리즘과 안전성 면에서 비교해 두기.

[대칭키 암호 알고리즘]

I) DES

i) 구조

*Feistel

* 64 bits → 64 bits (각 부분 32 bits)

16 Round key (48 bits)
Main key (64 bits = 56 + 8)

*16개 라운드.

ii) 함수

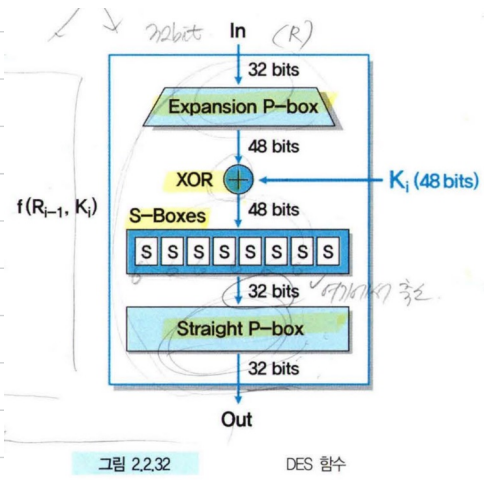


그림 2.2.32 DES 함수

iii) 블록, 라운드, 키, 라운드키

1블록 : 64 bit

키 : Seed : 64 (56+8) bit
└ Round : 48 bit

라운드 : 16

II) 3DES

i) 구조

*키가 2개인 DES

암호화 → 복호화 → 암호화
K1 K2 K1

*키가 3개인 DES

암호화 → 복호화 → 암호화
K1 K2 K3

ii) 함수

상동.

iii) 블록, 라운드, 키, 라운드키

상동.

키 크기 ┬ 128 bits (112 + 8×2)
└ 192 bits (168 + 8×3)

III) AES

i) 구조

*Feistel

* 10, 12, 14 round 有.

key : 128 192 256 bits

Round key : 128 bits.

* 128 bits → 128 bits

* 레인달이 제안함.

* 암호화 모듈 따로 있음.

ii) 함수

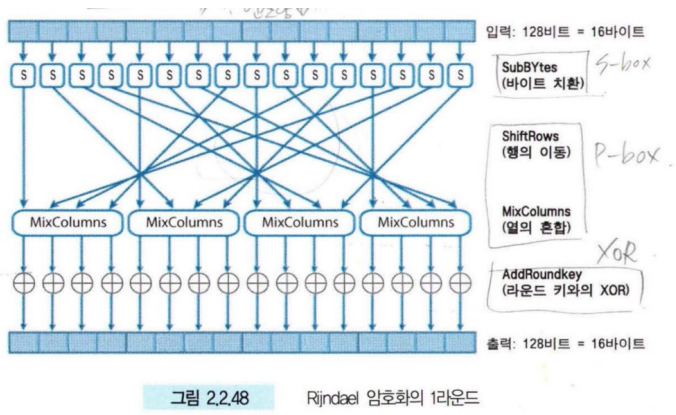


그림 2.2.48 Rijndael 암호화의 1라운드

iii) 블록, 라운드, 키, 라운드키

1블록 : 128 bits

키 : Seed : 128 192 256 bits
└ Round : 128 bits

라운드 : 10 12 14

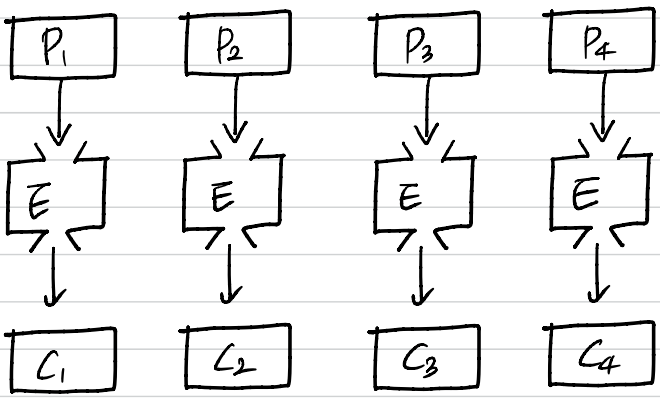
IV) 기타 대칭키 암호 알고리즘

구조 ┬ Feistel : DES, RC5, RC6, SEED (한국 기술)
├ IDEA
└ SPN : SPN, ARIA, S로 시작하면 웬만하면 애

[암호 알고리즘 운용 모드]

I) ECB

i) 암호화



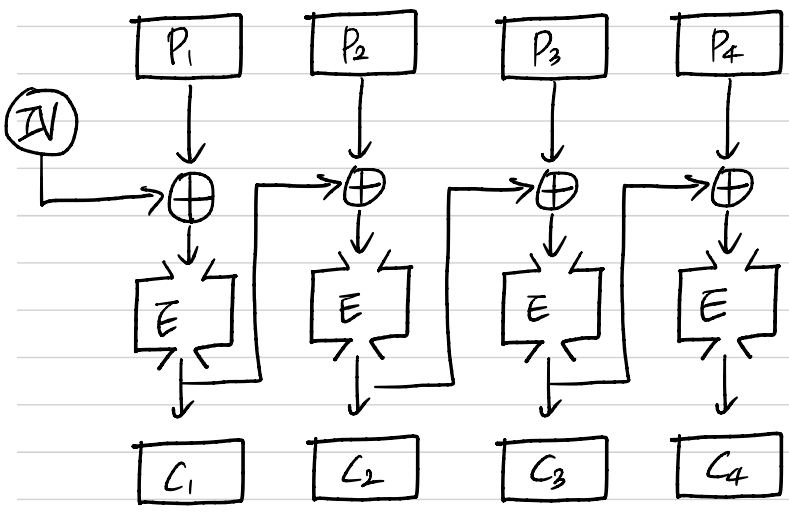
ii) 복호화

(그냥 뒤집으면 됨)

암호화 모듈 자리에 복호화 모듈

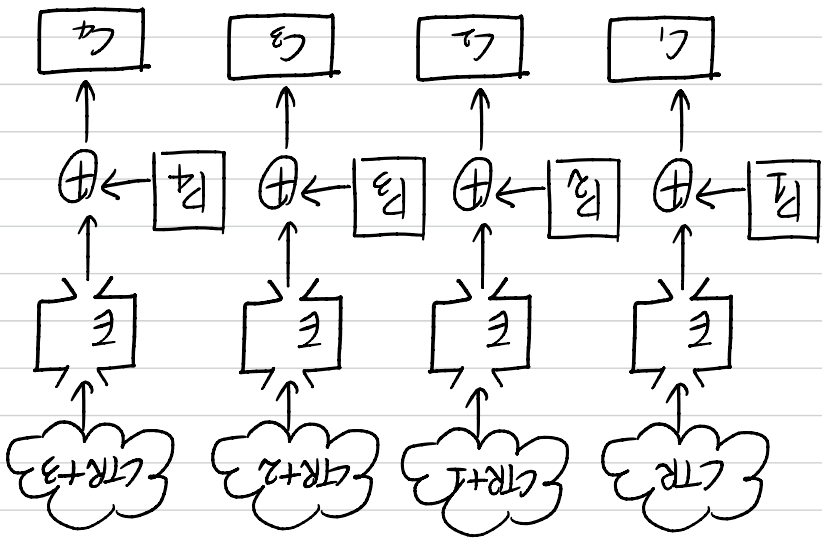
II) CBC

i) 암호화



ii) 복호화

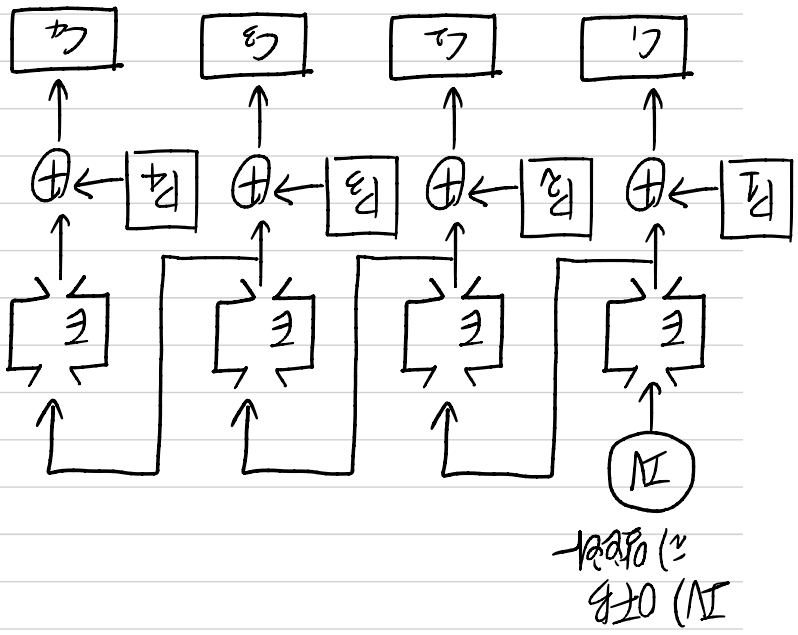
V) CTR 기반 암호화



VI) 복호화

병렬처리에 용이 // 암호화 속도 고려시

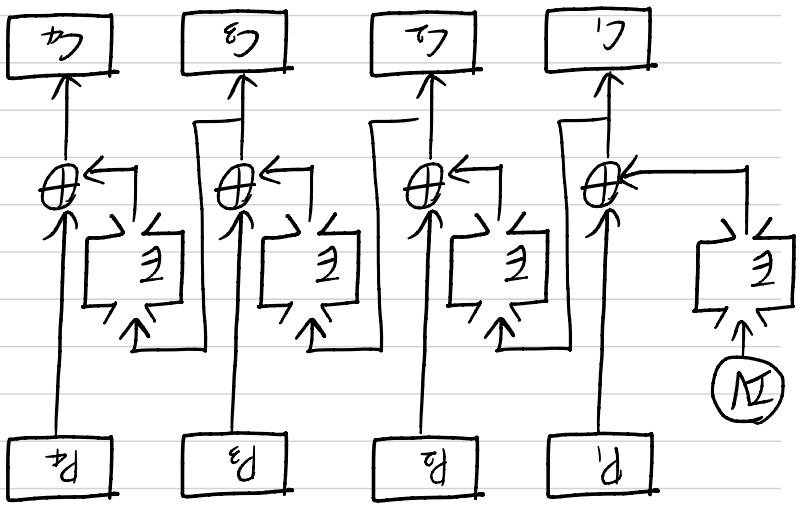
IV) OFB 기반 암호화



VII) 복호화

병렬처리에 용이 // 암호화 속도 고려시

III) CFB 기반 암호화



VIII) 복호화

병렬처리에 용이 // 암호화 속도 고려시

7) 블록 암호 정리

(가) 블록 암호 모드 비교 20.감리 19.감리 18.11회.기사

12.감리 18.12회.산기

17.9회.기사 13.2회.기사

중요체크

- 블록 모드: ECB, CBC
- 스트림 암호방식의 블록 암호모드: CFB, OFB, CTR

	이름	장점	단점	비고
ECB 모드	전자 부호표 모드 (Electronic CodeBook)	• 간단 • 고속 • 병렬 처리 가능 (암호화, 복호화 양쪽)	• 평문 속의 반복이 암호문에 반영된다. • 암호문 블록의 삭제나 교체에 의한 평문의 조작 가능 • 비트 단위의 에러가 있는 암호문을 복호화하면 대응하는 블록이 에러가 난다. • 재전송 공격이 가능	사용해서는 안 된다.
CBC 모드	암호 블록 연쇄 모드 (Cipher Block Chaining)	• 평문의 반복은 암호문에 반영되지 않는다. • 병렬 처리 가능(복호화만) • 임의의 암호문 블록을 복호화 가능	• 전송 도중 암호문 블록 C_j에서 한 비트 오류가 발생하면 평문 블록 P_j에는 대부분의 비트에서 오류가 발생되고, 평문 블록 P_{j+1}에서는 C_j의 오류비트와 같은 위치에서 한 비트 오류가 발생한다. • 암호화에서는 병렬 처리 불가능	Practical Crypto -graphy 권장
CFB 모드	암호 피드백 모드 (Cipher FeedBack)	• 패딩이 필요 없다. • 병렬처리 가능(복호화만) • 임의의 암호문 블록을 복호화 가능	• 암호화에서는 병렬 처리 불가능 • 전송 도중 암호문 블록 C_j에서 한 비트 오류가 발생하면 평문 블록 P_j에는 C_j의 오류 비트와 같은 위치에서 한 비트 오류가 발생된다. 하지만 C_j의 비트는 시프트 레지스터에 오류가 존재하는 한 다음 평문 블록의 대부분의 비트에 오류가 발생한다.(확률적으로 50%) • 재전송 공격이 가능	
OFB 모드	출력 피드백 모드 (Output FeedBack)	• 패딩이 필요 없다. • 암·복호화의 사전 준비 가능 • 암·복호화가 같은 구조 • 비트 단위의 에러가 있는 암호문을 복호화하면 평문의 대응하는 비트만 에러가 난다.	• 병렬 처리 불가능 • 적극적 공격자가 암호문 블록의 비트를 반전시키면 대응하는 평문 블록의 비트가 반전된다.	
CTR 모드	카운터 모드 (CounTeR)	• 패딩이 필요 없다. • 암·복호화의 사전 준비 가능. • 암·복호화가 같은 구조 • 비트 단위의 에러가 있는 암호문을 복호화하면 평문의 대응하는 비트만 에러가 난다. • 병렬 처리 가능(암·복호화 양쪽)	• 적극적 공격자가 암호문 블록의 비트를 반전시키면 대응하는 평문 블록의 비트가 반전된다.	Practical Crypto -graphy 권장

표 2.2.6 블록 암호모드 비교표

<네트워크 장비의 이해>

데이터 → 세그먼트 → 패킷 → 프레임 → 비트스트림

I) 물리적 장비

i) L1 #수신자가 정상적으로 신호 복원

① 허브 : "브로드캐스트" "패턴 복사"

② 리피터 : "패턴 복사" "증폭"

③ 랜카드

ii) L2 #MAC(48) #네트워크 연결 #Learning #Flooding #Forwarding #Faltering #Aging → L2TA.

① 브릿지 : "충돌방지" "프레임간의 보정"

② 스위치 : "리피터+브릿지" "속도증대" "포트-VPI-포트" "포트 미러링" "학습 → 스위칭 테이블"

iii) L3 #IP(32 or 128) #LAN 연결

① 라우터 : "Fragmentation" → "패킷 내용 변경" "최소한의 anti-DDoS" "RIP, BGP4, OSPF" "인터페이스별 MAC & IP" "내 앞까지만 처리하면 돼"

② L3 스위치

iv) L4 #TCP/IP #well-known port num

① L4 스위치 : "로드밸런싱도 되용!"

v) L7 # all port num

① 게이트웨이 : "통역자" "다른 환경 중재" "또 계층 프로토콜 이해"

② L7 스위치 : "L4보다 똑이똑"

II) 논리적 장비

"VLAN" — Tagging (IEEE 802.1Q.)

By L2 스위치 ⇒ 브로드캐스트 도메인 제어.

VLAN - L3 - VLAN

Based on What?

① Port

② MAC addr X

③ Net addr

④ Protocol

⑤ Multicast IP.

⑥ Combination

III) 스위치 상황

스위칭 방법

① Store-and-Forward : 프레임 다 받고 처리
∴ 느림 But 오류대응↑

② Cut-Through : 처음 48bit (MAC) 만 보고 Forward
∴ 빠름 But 오류대응↓

③ Fragment-Free : ①+② 장점.
- ① : 512비트 확인 (오류대응↑)
- ② : 프레임의 Store 기다리지 X.

IV) 브리지 vs 라우터

L2	L3
X	패킷변경
MAC	IP
MAC 아님	인터페이스별(MAC, IP)
브로드캐스팅 O / 멀티캐스팅	X
테이블 O	테이블 O
MAC In → O	IP In → X

<원도우 서버 보안>

[원도우 5 rings 구조]

I) 5 rings

Applications
Manager
Macro kernel
HAL
H/W

II) Subsystem

POSIX
32bit
NTVDM (16bit)
Win32 Subsystem — 보안, OS/2

[원도우 파일시스템]

I) FAT16

ⅰ) 특징

① 파티션 MAX 16GB
② 2GB 개 클러스터
③ 클러스터 32KB

II) FAT32

ⅰ) 특징

① 파티션 크기↑↑↑
② 2TB 개 클러스터
③ 호환성↑
④ 보안성↓ (접근제어X)

III) NTFS

ⅰ) 특징

① 원도우 NT계열에서만
② 매우대용량 / 이름길이↑
③ 보안성↑ 접근제어O · 하위파일 자동상속
(개별 폴더·파일에 사용권한 O ·
④ Auditing

ⅱ) 디스크 구조

MBR + VBR + MFT + 시스템 파일 + 파일

MBR
: 섹터 0, 512 byte
디스크 부트 코드
디스크 메타데이터 (파티션테이블)
시그니처 (NT··)

VBR
OS 부팅 인스트럭션
파일시스템 메타데이터
(각 볼륨포맷크기)
MFT 시작주소
파티션 마지막 섹터에 백업.

MFT
각 파일·디렉토리 메타데이터
(이름·소유주·권한·MAC Time)
→ 개별 파일은 하나의
MFT Entry를 가짐.

시스템 파일
※ NTFS의 핫팩스 상등
오류 복구용 데이터

파일
실 데이터

IV) FAT vs NTFS : 장·단점 위주로

	FAT	NTFS
장점	호환성↑ 안정성↑ 저용량에 효율↑	대용량 효율↑↑ 보안성↑↑ 파일 이름&길이↑↑
단점	보안성↓ 대용량 효율↓	호환성↓↓↓ 저용량 효율↓, (NTFS<FAT)

※ 리눅스&유닉스의 디스크 구조

부트 블록 + Super block + i-node list + data

Super block
해당 볼륨의 메타데이터

i-node list
파일 (일반, 디렉터리, 장치) 메타데이터
파일명은 X
MAC Time
- Modification
- Access
- Metadata Change

[원도우 부팅 순서]

① POST
※ 파워 켜고 H/W들의 셀프점검
→ ② BIOS의 CMOS 읽어 config 적용
→ ③ MBR 로드
→ ④ NTLDR
※ 최초의 파일시스템 on
※ boot.ini 읽어서 부팅옵션
구버전 (XP, 서버2003)
실행
신?버전 (Vista, 07, 08)
실행
④ 부트매니저
※ bootmgr.exe
※ BCD 읽음 → 어떤 OS 부팅.
수정불가 → 보안성↑
→ ⑤ OS로더 (Winload.exe)
※ 장치 드라이버 로드
→ ⑥ 하드웨어 인지 by NTDETECT.com
※ ∴ 장치드라이버 로드
→ ⑥ ntoskrnl.exe 실행
※ HAL, DLL, 커널로드
※ 서브시스템 on

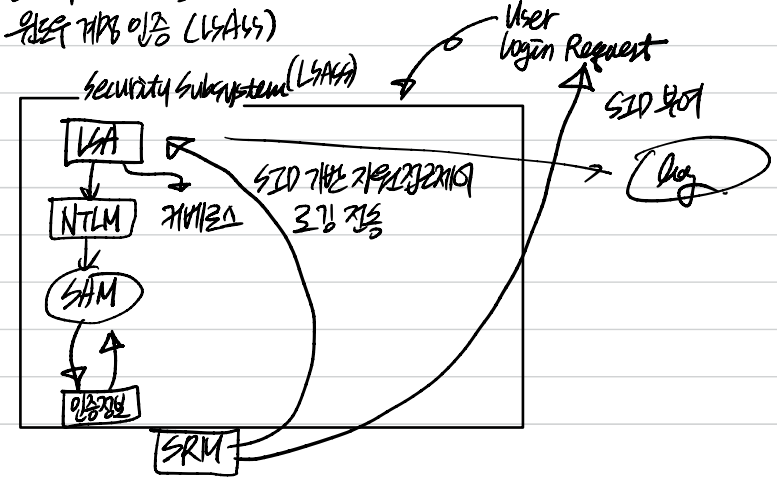
(부트설정데이터)

[원도우 계정, 인증, 권한]

I) 원도우 주요 계정과 SID

Administrator : S-1-5+도메인+500
Guest : S-1-5+도메인+501
User : S-1-5+도메인+1000↑
Anonymous : S-1-0-7
System service : S-1-5-18
Not known : S-1-0-0
Everyone : S-1-1-0

II) 원도우 계정 인증 (LSASS)



※ SAM files
[OS drive] / system32 / config / sam
보통 C:/Windows.

[윈도우 네트워크 공유 폴더]

I) 설정방법

net use [mount drive]: \\IP\[target drive]

II) 보안권한 설정

i) 파일&폴더 공통 ㄱ + 폴더 only1 (ALL)

① 모든 권한 : 접근권한&소유권 변경. 서브폴더&파일 삭제

② 수정 : 삭제권, ③ == ④ 해야 함

③ 읽기&실행 : 읽기권, dir/file mv 권.

④ 읽기 : Read-only

⑤ 쓰기 : 생성권, 권한 확인권.

⑥ 폴더 내용 보기 : 파일·서브디렉터리 이름 보권 -

ii) 폴더&파일 접근 권한 (CL)

① NTFS라면? 권한은 누적.

② 파일 접근권한 < 디렉터리 접근권한

③ 명백한 deny가 allow에 우선.

iii) 공유 폴더 공유. & 관리 → 숨김은 이름뒤$. 이걸 보려면 net share 쳐야 함.

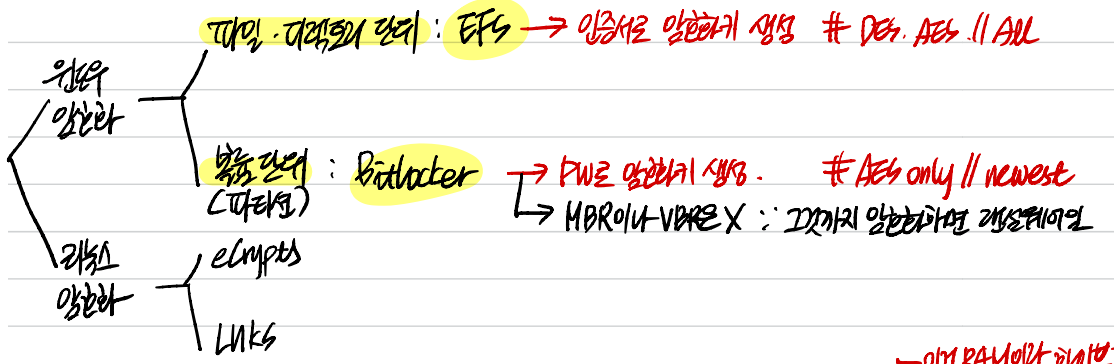
i) common 공유 폴더

① C$. D$

② ADMIN$ → 윈도우 설치폴더임.

③ IPC$ → 프로세스통신 ∴ 함부로 비활성화 ⇒ 필수 ∴ Anonymous (CS-1-5-7)은 deny

ii) 관리 (암호화)



[레지스트리]

I) Master key : 하이브 파일을 공유하면 이 둘만 있음

SYSTEM.dat ① HKLM (HKEY-local-Machine) : 현재 컴퓨터에 의존한 HW 설정

USER.dat ② HKU (HKEY-USERS) : 시스템에 존재하는 모든 유저들의 프로파일

HARDWARE ← 이건 RAM이라 하이브 파일이 없음

SAM
SECURITY
SOFTWARE
SYSTEM

II) Derived key:

① HKCU (HKEY-Current-User) : "현재 로그온한" 사용자정보 . // 약간 utmp같은 거기

② HKCC (HKEY-Current-Config) : 하드웨어 프로파일; HKLM 서브셋에 모니터·프린터 정보only.

③ HKCR (HKEY-Classes-Root) : COM 오브젝트 등록정보, 확장자-응용 연결정보.

↓ 이건 어떻게 관리하는가?

보통 레지스트리는 "설정 저장" 용

가장 자주 악용 ∴ ① 시스템 시작 시 뭔가 자동 실행하는지

→ 전체사용자 — 암호성 : HKLM\ … \CurrentVersion\RunOnce
　　　　　　 — 지속성 : 〃 \Run

→ 개별사용자 — 암호성 : HKCU\ … \CurrentVersion\RunOnce
　　　　　　 — 지속성 : 〃 \Run

② 확장자-응용 연결은 어떻게 할지

→ HKCU를 변경;

권한 상승에 악용 → ③ 유저의 그룹이 뭐고 뭘 권한은 뭔지

→ HKU 키 변동 … 보통 이건 SAM 털어서 해쉬값 만들 때 변경.

공격자는 레지스트리키 변경후 어떤 행위를 할까?

HKCU\... \Explorer\RecentDocs

내부를 탐색하거나 (파일 Read) — 악성SW를 설치하거나 — 터미널 커넥션을 설정하거나 — 무언가 설정.

HKCU\... \Explorer\... \OpenSaveMRU

HKCU\Software

HKCU\TerminalServerClient\Default

정보 이외 또 어떻게 얻을 수 있을까?

파일 시스템은 어떻게 해야 할까?
수많은 구성 요소.
SYSTEM.dat, USER.dat
SYSTEM.ini, WIN.ini

① 아예 접근을 제한 ← UAC (아마 LSASS의 SRM이 하는 걸거임)

② 개별 레지스트리키 저장 (.reg인 텍스트파일)

③ 아예 스냅샷 찍기

[윈도우 서버 보안 설정]

서버의 보안 요소

① 계정
- 관리자 계정 이름 변경 — Need-to-know, 최고권한 최소한의, 계정아키텍처
- 암호 정책 설정 — 의사자 · 사용X 계정 최대한 빨리 식제하기
- Guest 계정 off
- 로그인 실패횟수 & 잠금기간 설정.

② 서비스
- 불필요 서비스 비활성화
- 공유권한 · 폴더 사용자 그룹 설정, ACL에서 Everyone 삭제
- 하드디스크 기본공유 제거 (레지스트리 CurrentControlSet\... \LanmanServer\parameters)
  ↳ net share /delete 근데 왜이름?
- 익명AA 취약점.
- Null session 접근차단

③ 패치.

① Anonymous FTP off. TFTP는 꼭 필요하면 chroot 같은거라도 사용하기
② Alerter
③ Clipbook
④ Messenger
⑤ Simple TCP/IP Service

[Win AD] ← 네트워크 서비스

특히 네트워크 디렉터리 공유서비스가 매우 exploitable 하므로 주의해야 한다.

I) Directory Database
  ① Network Directory Service
    ex) LDAP, AD, NDS ...

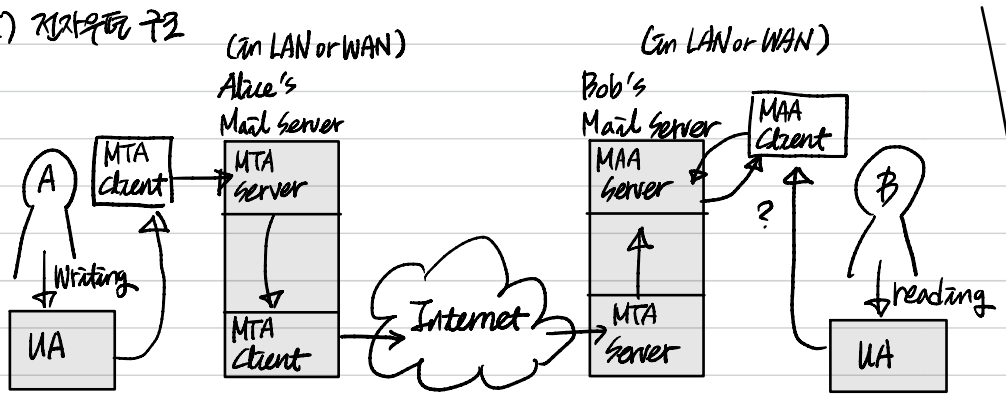
II) Active Directory
→ 보안정책 관리의 디렉터리.
디렉터리에 데이터 저장 — 공유자원 ↝ 서버, 보안, 프린터 ...
                        — 네트워크 사용자 & 컴퓨터 계정.

→ AD 서버
AD는 도메인별로 서비스.: — 도메인 컨트롤러 ← DB에 실제 계층구조 디렉터리 有.
                          AD를 실행하는 주체, 로그인 인증 컨트롤.
                          ∴ AD 정보 복제, 저장, 계정 로그인처리.
                          도메인 보안 정책 저장.
                        — 구성원 서버

∴ 도메인 A ↑ : 독립.

[이메일 보안]

I) 전자우편 구조



II) MTA : SMTP ← DNS를 사용해서 전방으로 전송 (Dst 메일서버까지)

i) 대표적 SMTP 소프트웨어

- UNIX : Sendmail
- MS : MS Exchange

ii) 보안상 문제

* 이메일 암호화 기능X

iii) 명령 순서

EHLO → (AUTH) → MAIL → RCPT → DATA → QUIT.

⊕ 특수명령

HELP. RSET. NOOP. VRFY. EXPN.

III) MAA : POP3, IMAP4

i) POP3

① 110번 포트 TCP

② 서버 ← 클라이언트 해서 명령들을 보고 받을수O

③ USER, PASS 명령어로 인증.

ii) IMAP4

① 143번 포트 TCP

② POP3 보다 기능↑

- 메일 다운 전 헤더 검사
- 메일 다운 전 문자열 검색
- 메일 일부만 다운
- 메일 서버에 메일 생성 관리 (온라인)

IV) 이메일 구성요소 보안

PEM	PGP	S/MIME
◦ 보통 군용. 공공용 (보안성↑)	◦ 일반용 (보안성↓)	◦ 상용 표준
◦ 공인기관의 키 인증	◦ 분산형 키 인증 구조	◦ X.509 인증서 지원
◦ ∴ 구현 어려움	◦ ∴ 구현 수월	◦ PEM & PGP 개선

i) PGP

5개 서비스

① 기밀성 (암호화) → 3DES, AES, IDEA, CAST ← CBC로 운용. ← 비밀키로 암호화 → 상대 공개키로 비밀키 암호화] + → 인증 이후 암호화 과정

② 인증 (전자서명) ∴ HASH + 공개키인증 → SHA-2, MD5, RSA + DSS → 메시지 해시를 내 개인키로 암호화

(DF. ElGamal)

③ 압축 → 근IP

④ 전자 우편 호환성 → Base64 ∴ ASCII로 변환 (다른 전자메일서비스는 이게 on/off?)

⑤ 분할 & 재합성 → 최대 메시지 크기 제한 예시

* 송신 부인방지 O 수신 부인방지 X

키링 — 한 쌍! # Web of Trust (플랫데이터)

Pair 01. 여러 User의 공개키 — Public Keyring

Pair 02. 내가 쓴 개인키 — Private Keyring

이메일 보안기술 보장할 것인가?

① 컨텐츠 보안 : PEM, PGP, S/MIME

② 인프라 보안 : Anti-SPAM ; 메일서버 보안 — 메일서버 수신차단 / 이외

메일 클라이언트 보안

iii) S/MIME

MIME를 통해 ASCII 외 포맷도 송신할 수 있어용!!!

4개 서비스

① 기밀성 (암호화) : AES-128 // CBC로 운영.

② 인증 (전자서명) : RSA & SHA-256.

③ 이메일 호환성 : Base64 (Radix64)

④ 압축 : *

X.509 인증서 지원 → 개선된 서비스

V) 인프라 보안

i) 메일 서버 보안

① 메일 서버 수신 차단 — 콘텐츠 필터링 / 송신자 필터링 / 네트워크 레벨 필터링 / 바이러스 기준 차단 | 시그니쳐를 차단.

② 메일 서버 보안 — Relay 스팸 방지 ← SendMail.

Anti-SPAM ~ 스팸 필터 솔루션 — 메일헤더 / 제목 / 본문 / 첨부파일 | 필터링.

ii) 메일 클라이언트 보안

① 콘텐츠 ② 송신자] 필터링.

iii) 거버넌스적 보안

* 메일 서버 등록제 (SPF)

Sender의 서버를 DNS에 등록.

Sendmail

IPSec

① 기밀성(암호화) : ESP only (ESP 헤더&트레일러)
② 제한적 트래픽 흐름 기밀성 : ESP-tunnel only (암호화+보안 게이트웨이)
③ 송신처 인증 (ESP Auth(ICV) AH Auth data(ICV))
④ 재전송 공격 방어 (ESP, AH header seq #)
⑤ 무결성 보장 (비연결형) — msg 순서의 상관도 보장. (ESP Auth & AH ICV).
⑥ SA 사용 ∴ 정보제어 (SPI)

제공 서비스 6개

VPN에 用

기존의 IP헤더 상태로 ∴ 인트-인트 보호 ← 전송 모드 (Next IP header & ESP)

2개 종단 모드 — IPSec — 2개 보안 모드

인증에는 MAC 사용. ∴ 무결성+송신처인증.

Authentication Header
AH 「인증」
- 전송 모드 — IP 헤더 | AH | L4 세그먼트 (전송 영역 (단, IP 헤더의 mutable field 제외))
- 터널 모드 — New IP 헤더 | AH | IP 헤더 | L4 세그먼트 (전송 영역 (단, New IP 헤더의 mutable field 제외))

AH 헤더:

Next Header	Parameter Length	Reserved
SPI		← SA 정보.
Seq. num.		← 재전송 공격 방어
Auth data (ICV). 32*n bit		

보호 목적 (Goal) : 흐름? 개별적인 IP 패킷? 인증용 · 암호화용 ×

「인증 & 암호화」
ESP
- 전송 모드 — IP 헤더 | ESP 헤더 | L4 세그먼트 | ESP 트레일러 | ESP AUTH (인증 영역, 암호화 영역, 인증 정보, 암호화 정보)
- 터널 모드 — New IP 헤더 | ESP 헤더 | IP 헤더 | L4 세그먼트 | ESP 트레일러 | ESP AUTH (암호화 영역, 인증 영역, 암호화 정보, 인증 정보)

→ 암호화 + New IP header (보안 게이트웨이 개방)
⇒ 보안 게이트웨이 이후의 라우팅은 안 수 X
∴ "완전한 터널링 기법"

ESP 헤더 · 트레일러 · AUTH :
보안화
ESP 트레일러 (암호화 essential 부수적 데이터)
Padding data + Padding length + next header (캡슐화된 프로토콜 헤더)
↳ 블록 암호화 과정에 필수적으로
ESP 헤더 (어떤 규칙으로 암호화? 재전#)
SPI num + Seq num ← 재전송공격 방지
ESP Auth — 인증 ∴ 암호화 ICV (MAC-based) 32*n bit

SA
* SAD, SPD 개념X
* Inbound 는 SAD에 따른 SPD 적용
* Outbound는 SPD 확인 후 PROTECT 정책이면 SAD 검색.

- 하나의 SA는 한 방향 통신만 지원. ∴ 양방향 통신은 두개의 SA 필요.
- SA는 AH나 ESP, ISAKMP 中 택1
  ∴ 하나의 SA가 세 프로토콜 모두 지원X
- SA 구분 기준 : SPI + IP dst addr + AH·ESP 헤더

키 관리? — 키 생성 : OAKLEY
키 교환 (IKE) — 키를 직접 교환하는 프로토콜 : SKEME (DHKD)
키관리 프레임워크 : ISAKMP 프로토콜

IKE는 공개키로도 대칭키로도 최적화해 줌.

TLS / SSL

- ① 기밀성(암호화) → cipher suite
- ② 메시지무결성 → 해시, MAC
- ③ 서버-클라이언트 間 상호 인증 → 전자서명 (X.509)
- ④ 재전 제어 → 인증을 통해 구현

제공 서비스

- "고개 계층의 캡슐 프로토"
  L4 ~ L7 間 new 계층 개념체계 등장.
- 보안 파라미터 set TLS session ≠ TLS connection

이게 어떻게 암호화 인증을 제공하냐!!
1 byte + 1 msg (R/W)

TLS Handshake	TLS Change cipher spec	TLS Heartbeat	TLS Alert	HTTP
TLS Record (↑ 위: 애플 계층 메세지 / ∴ Presentation 계층역할: 복호화→암호화, 암호화→복호화)				
TCP / UDP (Transport)				
IP (Network)				

TLS Record pro.

암호화 & 인증 & 무결성 보장. // 제공 서비스: ① Confidentiality ② Integrity → 이 둘의 사용자 재정의 가능

* Sender: Fragmentation → Zip → Mac → Encryption → Add header
  ∴ outcome:

TLS Record header
Zipped & Encrypted data
MAC

* receiver: Remove & Read header → Decryption → Verify MAC → Unzip content → Defragmentation

proto. — TLS

TLS
- Cipher Suite
  Format: SSL/TLS_[KE&AUTH]_WITH_[Cipherspec]

취약점
- ① Heartbleed (OpenSSL) — 64KB data 탈취 by TLS heartbeat
- ② POODLE — SSL 3.0 의 downgrade 공격 (∵ 통신 spec은 client가 제공하므로)
- ③ FREAK — PF 공격에 취약한 RSA 타협레이트 → 키유출

↔ PFS (Perfect Forward Secrecy)
* 서버 개인키 유출되면 암호화된 과거 정보 기밀성 X.
  (이전 트래픽, 세션키 / 비밀키 기밀성 X)

4-way handshaking
* 서버가 먼저 인증받고, 클라이언트가 인증받음