

랜섬웨어 암호화 알고리즘

랜섬웨어 공격의 심각성

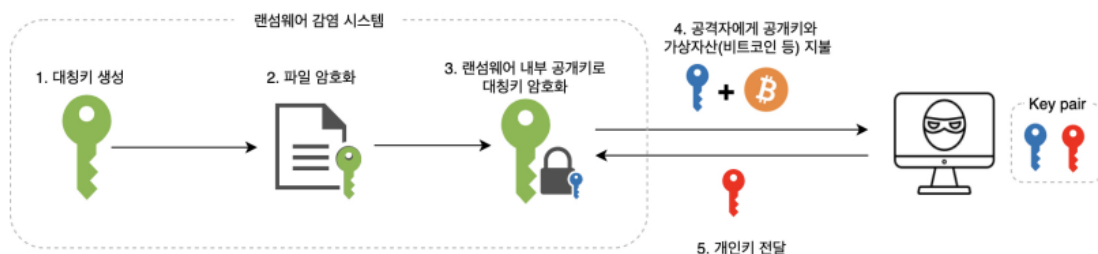
- 기업의 정보를 탈취 및 불법 유포
- 한 국가의 재정적 손실 초래

(랜섬웨어란? 컴퓨터 시스템을 감염시켜서 접근을 제한하고 그에 대한 값을 요구하는 악성 소프트웨어의 한 종류)

랜섬웨어 암호와 알고리즘

-랜섬웨어에서 사용하는 암호화 알고리즘은 크게 대칭키 암호와 공개키 암호로 나눌 수 있다.

구분	대칭키 암호	공개키 암호
장점	알고리즘이 단순해서 암호화 속도가 빠르다	암/복호화에 서로 다른 키를 사용해서 공개키로 암호화된 대상을 공격자가 가진 개인키로 복호화 하는 알고리즘 다수의 PC 를 감염시킬 때 용이
단점	암/복호화에 동일한 키를 사용	연산 알고리즘이 복잡해서 암호화 속도가 느리다



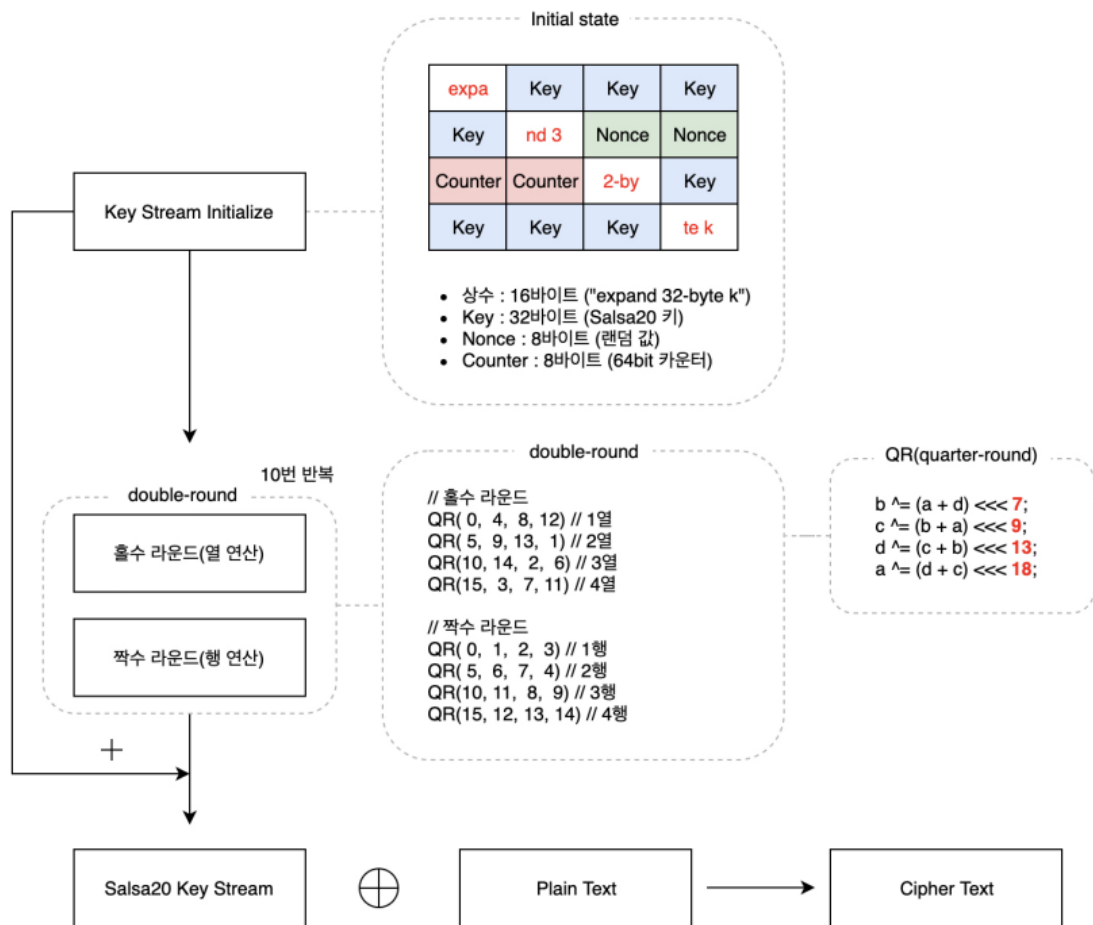
위 그림은 랜섬웨어가 파일을 암호화 시키는 과정을 단순화한 흐름도이다.

지금부터 대칭키 암호화와 공개키 암호화의 주요 알고리즘들을 간략하게 알아보고, 대표 랜섬웨어에 대해서도 짧게 소개해 보고자 한다.

구분	알고리즘	대표 랜섬웨어
대칭키	Salsa20	Revil
	Chacha20	Conti
	RC4	Clop
	AES	Nemty
	RSA	Clop
공개키	Curve25519	Babuk

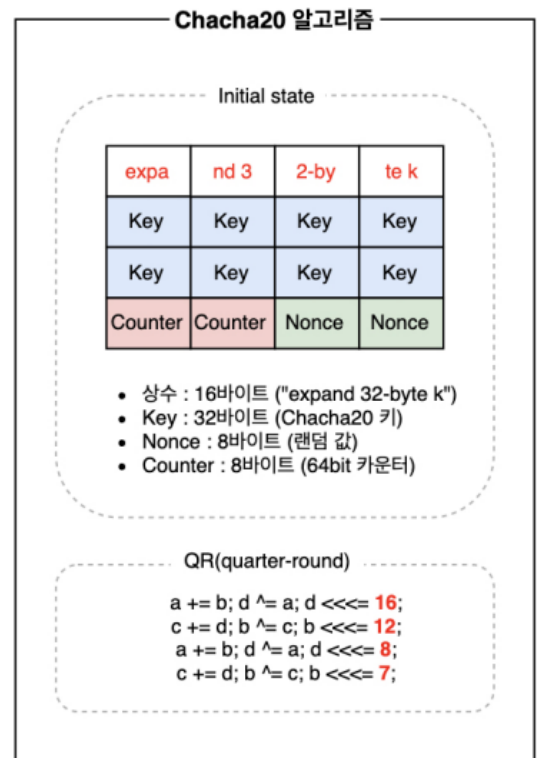
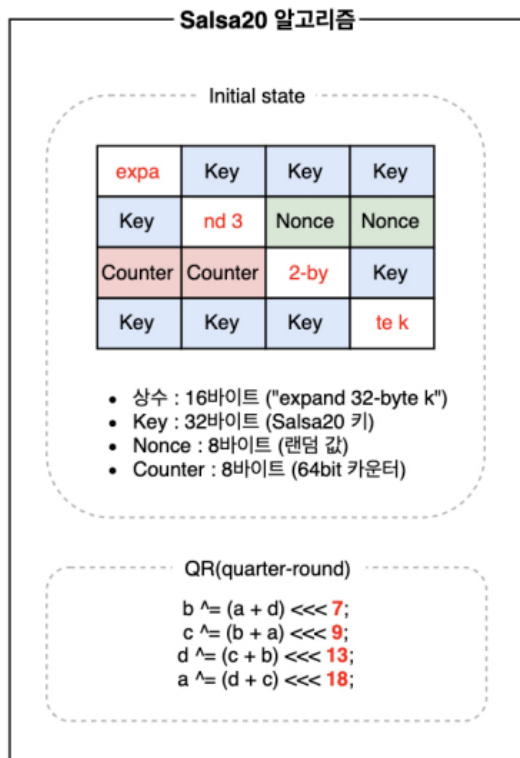
1. Salsa20 Algorithm

- Daniel J.Bernstein 이 개발한 스트림 암호 알고리즘
- 4x4 테이블로 구성된 64 바이트의 평문과 동일한 구성의 64 바이트 키 스트림을 XOR, 덧셈, 시프트 로 총 20 라운드의 연산 과정을 통해 암호화를 수행한다.
- 4x4 테이블에서 홀수 라운드에는 각 열을 OR 연산하고 짝수 라운드는 각 행을 OR 연산한다. 열 연산(홀수 라운드)와 행 연산(짝수 라운드)를 합친 Double round 를 10 라운드 반복하여 총 20 라운드의 연산을 수행한다.
- 초기 키 스트림과 20 라운드를 반복한 키 스트림을 덧셈 연산해서 최종 Salsa20 키 스트림을 생성한다.
- XOR 연산을 통한 평문 암호화 (→ Salsa20 hash function)을 거친 키 스트림과 평문을 XOR 연산하여 암호화한다.



2. ChaCha20 Algorithm

-Salsa20 과 동일한 초기 키 스트림을 구성한다. 다만 다른 점은 Initial state 의 구성요소의 배치 순서가 변경되었다.



Chacha20 Algorithm 을 사용한 Conti 랜섬웨어와 Salsa20 Algorithm 을 사용한 Sodinokibi 랜섬웨어를 비교하면 초기 키 스트림 구성 시 문자열 상수의 값 배치 순서가 다른 것을 볼 수 있다.

Hex	ASCII
65 78 70 61 6E 64 20 33 32 2D 62 79 74 65 20 6B	expand 32-byte k
AC B4 28 68 74 FA 2F 8E AF 6E 1E B4 AB 0F 36 44	~(htu/. n. «.6D
9F 45 B4 8F E6 37 95 EB 50 CB 82 9B 35 EB FA E3	.E'.æ7.ëPĚ..5ëúã
00 00 00 00 00 00 00 00 F6 D6 21 9A 2E 19 9E 0Aöö!.....

[Chacha20 알고리즘의 Key Stream Initialize(Conti 랜섬웨어)]

Hex	ASCII
65 78 70 61 D6 41 99 82 A7 9C 31 10 9F 5E A5 88	expa0A..\$.1..^¥.
1B 09 72 72 6E 64 20 33 5F D4 85 71 B2 B7 5B C6	..rrnd 3.0.q².[Æ
00 00 00 00 00 00 00 00 32 2D 62 79 D6 67 0A 402-by0g.@
CB 55 3F C9 CE E4 88 E2 8B A8 26 77 74 65 20 6B	ËU?Éîä.â.~&wte k

[Salsa20 알고리즘의 Key Stream Initialize(Sodinokibi 랜섬웨어)]

3. RC4 Algorithm

-Ron Rivest 가 발명한 스트림 암호 알고리즘이다.

→속도가 빠르다.

→네트워크 패킷 암호화, 파일 암호화 시 사용된다.

--RC4 Algorithm 동작 과정

-1) KSA(Key-scheduling algorithm)

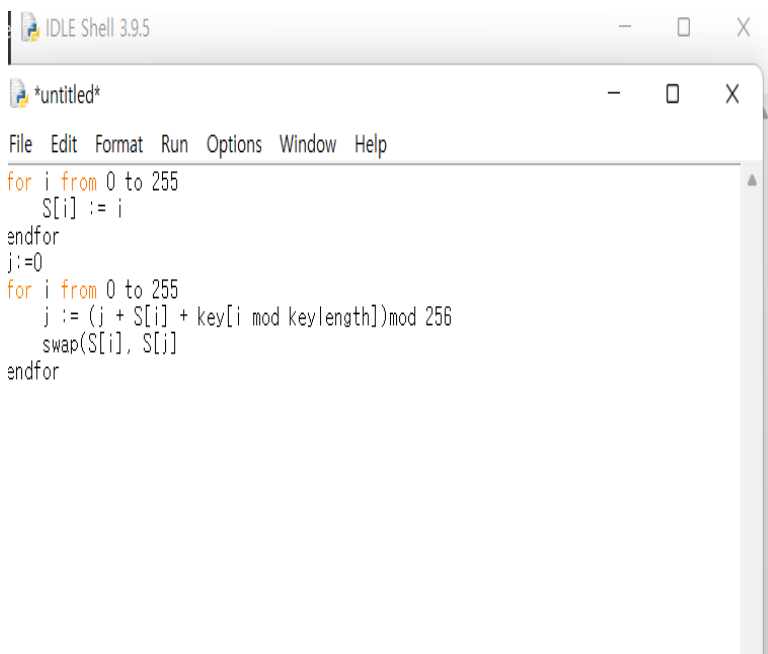
S-box 초기화하고, 키 배열 값을 더해서 임의의 S-box 위치에 저장하는 랜덤화 과정

-2)PRGA(Pseudo-random generation algorithm)

랜덤화된 S-box 값을 교환하는 과정을 거쳐 키 스트림 배열 생성

-3)XOR

위 과정을 거친 키 스트림 배열 값을 평문과 바이트 단위로 XOR 연산해 암호문 생성



```
IDLE Shell 3.9.5
*untitled*
File Edit Format Run Options Window Help
for i from 0 to 255
    S[i] := i
endfor
j:=0
for i from 0 to 255
    j := (j + S[i] + key[i mod keylength])mod 256
    swap(S[i], S[j])
endfor
```

KSA 단계

```
File Edit Format Run Options Window Help
i := 0
j := 0
while GeneratingOutput:
    i := (i+1)mod 256
    j := (j + S[i])mod 256
    swap(S[i], S[j])
    K := S[(S[i]+S[j])mod 256]
    output K
endwhile
```

PRGA 단계

$\text{XOR} \rightarrow \text{ciphertext}[1] = \text{plaintext}[1] (\text{XOR}) K[1]$

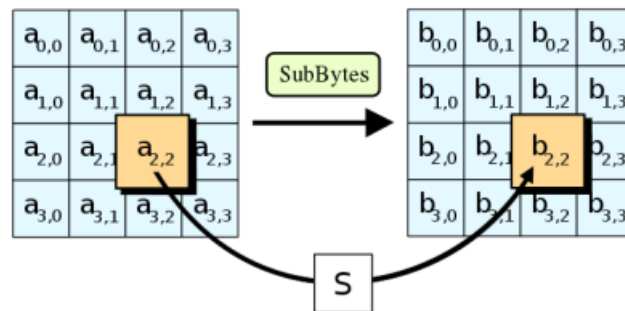
4. AES Algorithm

NIST 에서 DES 를 대체할 목적으로 암호 기법을 공모한 결과로 생성됨.

- 암호화와 복호화 과정에서 동일한 키를 사용하는 대칭키 알고리즘이다.
- 동작 과정
- 4x4 배열 형태의 입력블록이 주어진다.
- 키 길이에 따른 라운드 수 만큼 SubBytes(S-box 치환), ShiftRows, MixColumns, AddRoundKey(XOR)을 수행한다.

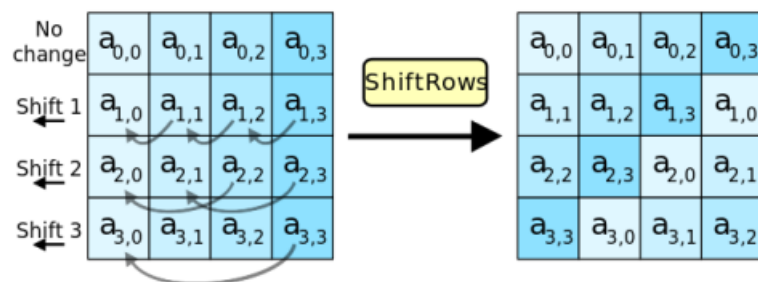
1) SubBytes(State, S-box)

이 단계에서는, 각각의 바이트가 s-box 에 따른 다른 바이트로 대체



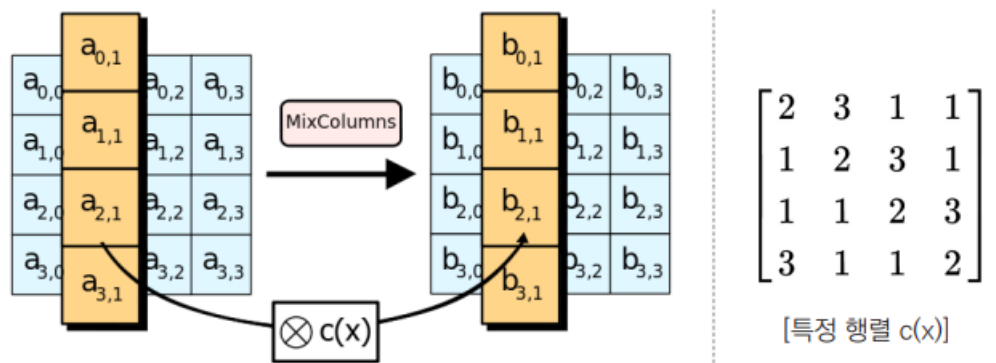
2) ShiftRows

행 이동 자리바꿈 단계에서는, 행의 각 바이트가 순서대로 왼쪽으로 이동



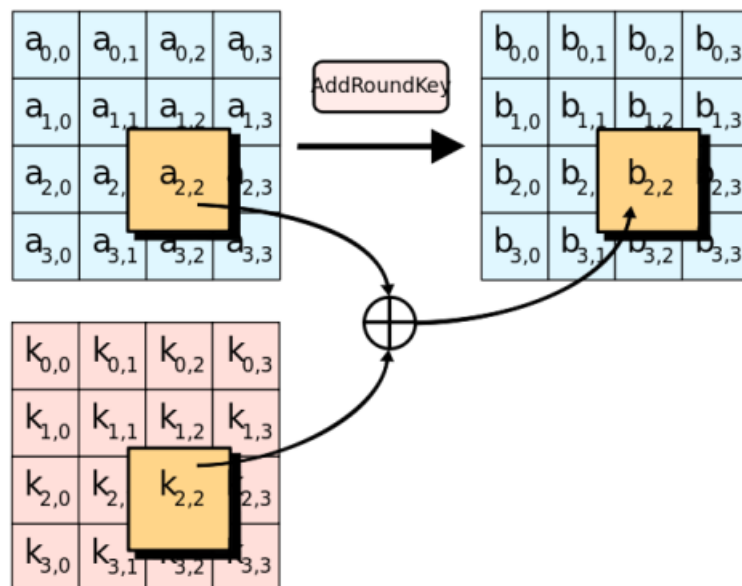
3) MixColumns

열을 섞고 치환 하는 단계이다. State 를 특정 행렬 $c(x)$ 와 곱셈 연산을 수행



4) AddRoundKey(Sate, Key)

XOR 단계이다, 평문과 Key 행렬 바이트를 XOR 연산한다.



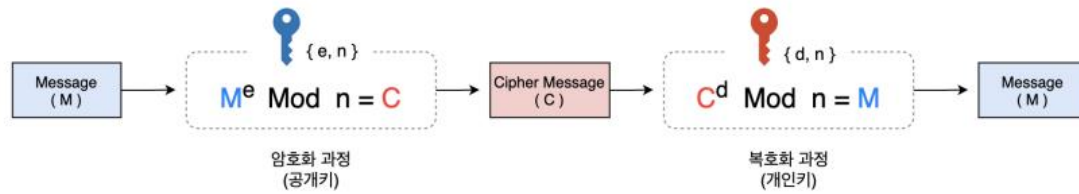
다음으로는 공개키 알고리즘이다.

1) RSA Algorithm

U.S MIT University Rivest, Shamir, Adleman 이 구현함

한 쌍의 (공개키-개인키)를 가지며, 공개키는 모두에게 알려진 암호화이고, 개인키를 가진 자만이 이 암호화를 열 수 있다.

주요 큰 틀은 “소인수 분해의 난해함”이다.



Ex) “2 개의 큰 소수의 곱으로 이루어진 수”, “인수가 아니면서, 서로소인 정수의 개수보다 작고 1 보다 큰 수”

-주요 동작 과정

2) Curve25519 Algorithm

타원곡선 암호기술(ECC) RSA 암호 방식에 대한 대안이다.

RSA 와 비교했을 때 더 작은 키 사이즈로 더 높은 보안 강도를 제공하는 장점이 있다.

키 사이즈와 계산량이 반비례관계다.

→동작 알고리즘

공격자

1) 비밀키 생성

-공격자는 32 바이트의 랜덤 값을 생성 후 248, 127, 64 상수 값을 사용해 연산한다.

2) 공개키 생성

-생성한 비밀키와 basepoint(9)를 사용해 32-byte 공개키 생성, 랜섬웨어 내부에 하드 코딩 후 랜섬웨어 유포

피해자

1) 비밀키 생성

-랜섬웨어에 감염된 피해자는 32 바이트의 랜덤한 값을 생성 후 248, 127, 64 상수 값을 사용해 연산한다.

2) 공개키 생성

-생성한 비밀키와 basepoint(9)를 사용해 32 바이트의 공개키를 생성한다

3) 공유 값 생성

-Shared Secret 인 공유 값은 피해자의 비밀키와 랜섬웨어 내부 하드코딩된 공격자의 공개키를 통해 생성된다.

공유 값은 파일 암호화 알고리즘인 Chacha 알고리즘 키 값으로 사용된다.

출처:CONTENTS-금융보안원