목차

TLS/SSL 3
암호화 종류 3
대칭키 3
공개키 3
해시 3
공개키 인증서 형식 3
인증서 포멧 변경 4
인증서 생성 4
CA 인증서 생성 4
서버 인증서 생성 5
CSR(인증요청서) 서명 5
Let's Encrypt / certbot 5
certbot 사용시 선택할 플러그인 5
디피-헬만 파라미터(dhparam) 키 는 무엇인가? 6

TLS/SSL

- https://www.ssllabs.com/ - 웹 보안점검
- https://sslmate.com/caa/ - CAA 생성기

- DNS CAA 레코드는 무엇일까?
- 디피 헬만 키 (Diffie-Hellman Key)
- Cipher Suite

암호화 종류

대칭키

- DES
- AES
- SEED
- ARIA

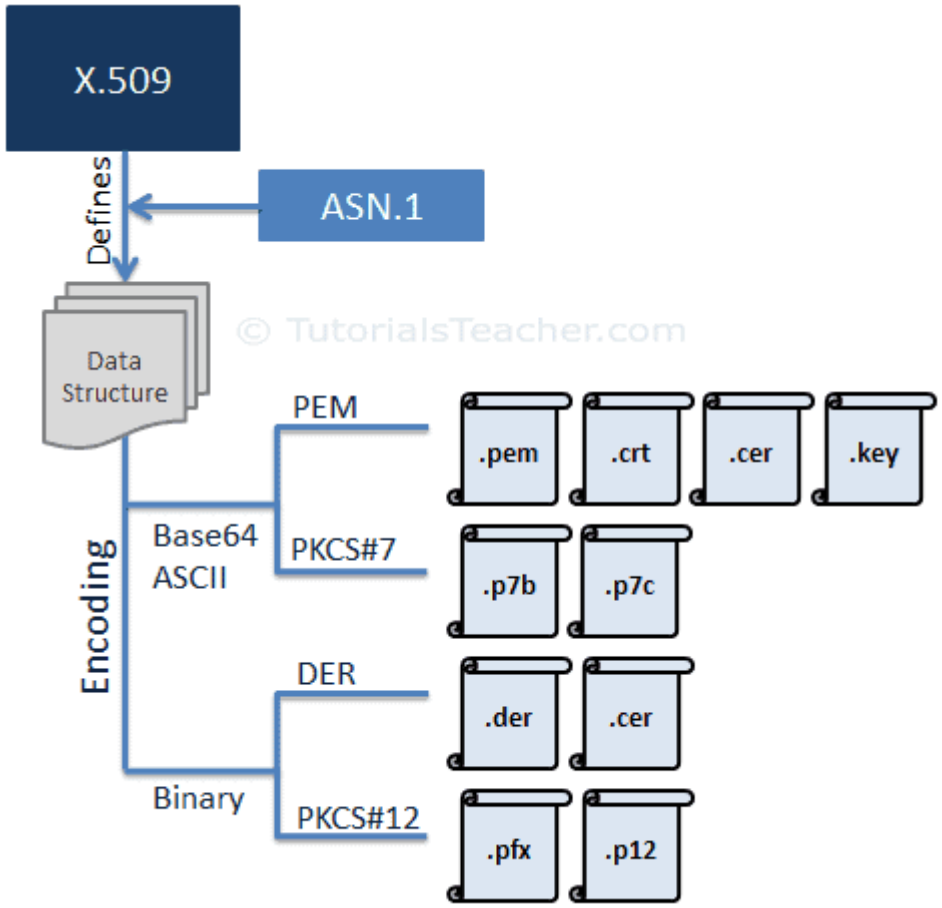
공개키

- RSA
- DSA
- Diffie-Hellman

해시

- MD5
- SHA1
- SHA2 (SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256)
- SHA3

공개키 인증서 형식



https://www.tutorialsteacher.com/https/ssl-certificate-format

- PEM(Privacy-enhanced Electronic Mail) : Base64 ASCII로 encode 된 파일로 .pem, .crt, .cer, .key등이 여기에 속한다. pem 파일은 서버 인증서와 중간자 인증서, private key가 해당 파일에 포함된다. crt 혹은 cer 파일로 서버 인증서나 중간자 인증서가 사용되고 개인키는 key파일이 될수있다.
- DER(Distingulished Encoding Rules) : binary 폼으로 .der 이나 .cer 파일들을 포함한다. 해당 인증서는 Java 기반 web 서버에서 주로 사용된다.
- pfx, p12 ... : PKCS#12(Public Key Cryptography Standards #12)

인증서 포멧 변경

```
openssl x509 -in cert.crt -outform der -out cert.der
```

인증서 생성

CA 인증서 생성

```
# key 생성
openssl genrsa -out ca.key 4096
# 인증서 생성
openssl req -x509 -new -nodes -sha512 -days 3650 \
 -key ca.key \
```

```
 -out ca.crt
```

서버 인증서 생성

```
# key 생성
openssl genrsa -out ia.key 4096

# CSR(인증요청서) 생성
openssl req -sha512 -new \
    -key ia.key \
    -out ia.csr
```

CSR(인증요청서) 서명

```
# 인증요청서 서명
cat > v3.ext <<-EOF
authorityKeyIdentifier=keyid,issuer
basicConstraints=CA:FALSE
keyUsage = digitalSignature, nonRepudiation, keyEncipherment,
dataEncipherment
extendedKeyUsage = serverAuth
subjectAltName = @alt_names

[alt_names]
DNS.1=yourdomain.com
DNS.2=yourdomain
DNS.3=hostname
EOF

openssl x509 -req -sha512 -days 3650 \
    -extfile v3.ext \
    -CA ca.crt -CAkey ca.key -CAcreateserial \
    -in ia.csr \
    -out ia.crt
```

Let's Encrypt / certbot

- https://certbot.eff.org/docs/using.html Certbot Manual
- Let's Encrypt(CertBot) SSL with HAProxy
- HAProxy SSL 설정
- Let's Encrypt(CertBot) Wildcard Certificates

certbot 사용시 선택할 플러그인

certbot으로 인증서 발급시 사용할 플러그인을 선택하는데 플러그인은 여러종류가 있지만 크게 http

방식과 dns방식이 있다.

- http 인증서를 적용할 머신에서 직접 발급받을때 사용하는 플러그인. 대부분의 경우에 사용가능하지만 와일드카드 *.domain.com 인증서 발급은 되지 않는다.
- dns 인증서를 사용할 머신이 아닌 다른곳에서 발급받을때 사용할 수 있다. 또한 와일드 카드 인증서 발급시 사용할 수 있다.

http 방식은 발급받는 머신에서 let's encrypt쪽으로 직접 응답을 하는 방식으로 확인을 하는 구조이다. 예를들어 my.domain.dom에 대한 인증서라면 해당 도메인으로 접근하는 서버에 대해서 소유권한이 있는지를 확인하는게 목적이기 때문이다. --standalone 옵션을 대표적으로 많이 사용한다. dns 방식은 발급받는 머신이 아닌 **도메인에 대한 전체 와일드카드 인증서를 발급**하기 때문에 해당 도메인에 대한 소유권을 확인하기 위해서 도메인에 TXT 레코드를 등록하여 확인하는 방식이다. --manual 옵션과 함께 --preferred-challenges dns 옵션을 추가로 같이 사용한다. --manual 옵션은 http 방식과 dns 방식 모두 사용할 수 있다.

디피-헬만 파라미터(dhparam) 키 는 무엇인가?

디피-헬만 파라미터(이하 dhparam)는 SSL 통신시 암호화를 도와주는 방식중 하나로 dhparam 키를 이용해 암호화 복잡도를 높여 보안을 강화하는것이 목적으로 각종 WAS의 SSL 통신에서 dhparam을 지원합니다. SSL 등급을 매기는 테스트 사이트에서 A등급 이상을 받으려면 이 설정은 필수로 해야합니다.

OpenSSL에서 dhparam 키를 생성하는 명령어는 다음과 같습니다.

```
openssl dhparam -out dhparam.pem <KEY-SIZE>
# KEY-SIZE 는 2048 또는 4096
```

보통 dhparam 키를 생성하는데 시간이 매우 오래 걸리기 때문에 누군가 만들어놓은 키값을 그대로 가져다 써도 무방하다. 다만 그 누군가는 믿을만한 사람이어겠다.

모질라https://ssl-config.mozilla.org/에서 공개적으로 제공하는 dhparam 파일이 있다.

- https://ssl-config.mozilla.org/ffdhe2048.txt
- https://ssl-config.mozilla.org/ffdhe4096.txt

참조링크

- https://rsec.kr/?p=242
- https://waspro.tistory.com/479

From:
https://atl.kr/dokuwiki/ - **AllThatLinux!**

Permanent link:
https://atl.kr/dokuwiki/doku.php/tls_ssl?rev=1623400211

Last update: **2021/06/11 08:30**