HTTPS 완전정복

김기영 / Founder and CEO
kiyoung.kim@flyhigh-x.com
FlyHigh Co., LTD. / 주식회사 플라이하이

인증서만 설정하면 HTTPS고

인증서만 바꾸면
HTTPS가 안전해지는가?

어떻게 하면 웹을 안전하게 구축하고 식별할까?

적어도 고객을 위한 서비스를 하려고 한다면 제대로 해야 한다.

Cipher Suites

Certificate

HTTPS

Browser

Web Server

HTTPS를 알기 전에 우리는 프로토콜을 알아야 한다.

Protocol

명사

1. [U] 외교 의례, 의전

a breach of **protocol**
외교 의례 위반

the **protocol** of diplomatic visits
외교 방문상의 의전

2. [C] (전문 용어) (조약의) 초안[원안]; (합의안·조약의) 보충 협약

the first Geneva **Protocol**
제네바 조약 초안

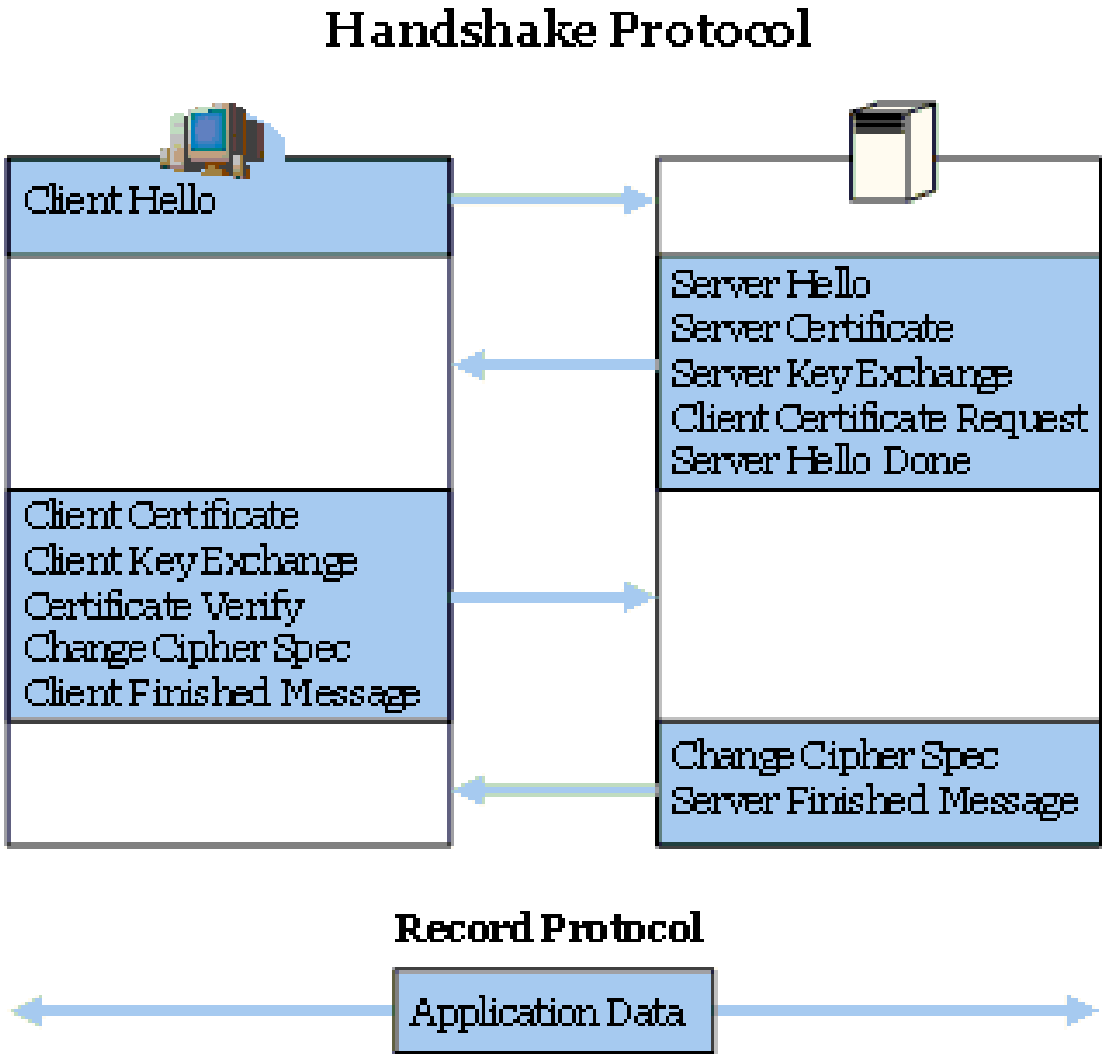
It is set out in a legally binding **protocol** which forms part of the treaty.
그것은 그 조약의 일부를 이루는, 법적 구속력이 있는 보충 협약으로 만들어진다.

3. [C] (컴퓨터) 프로토콜, 통신 규약

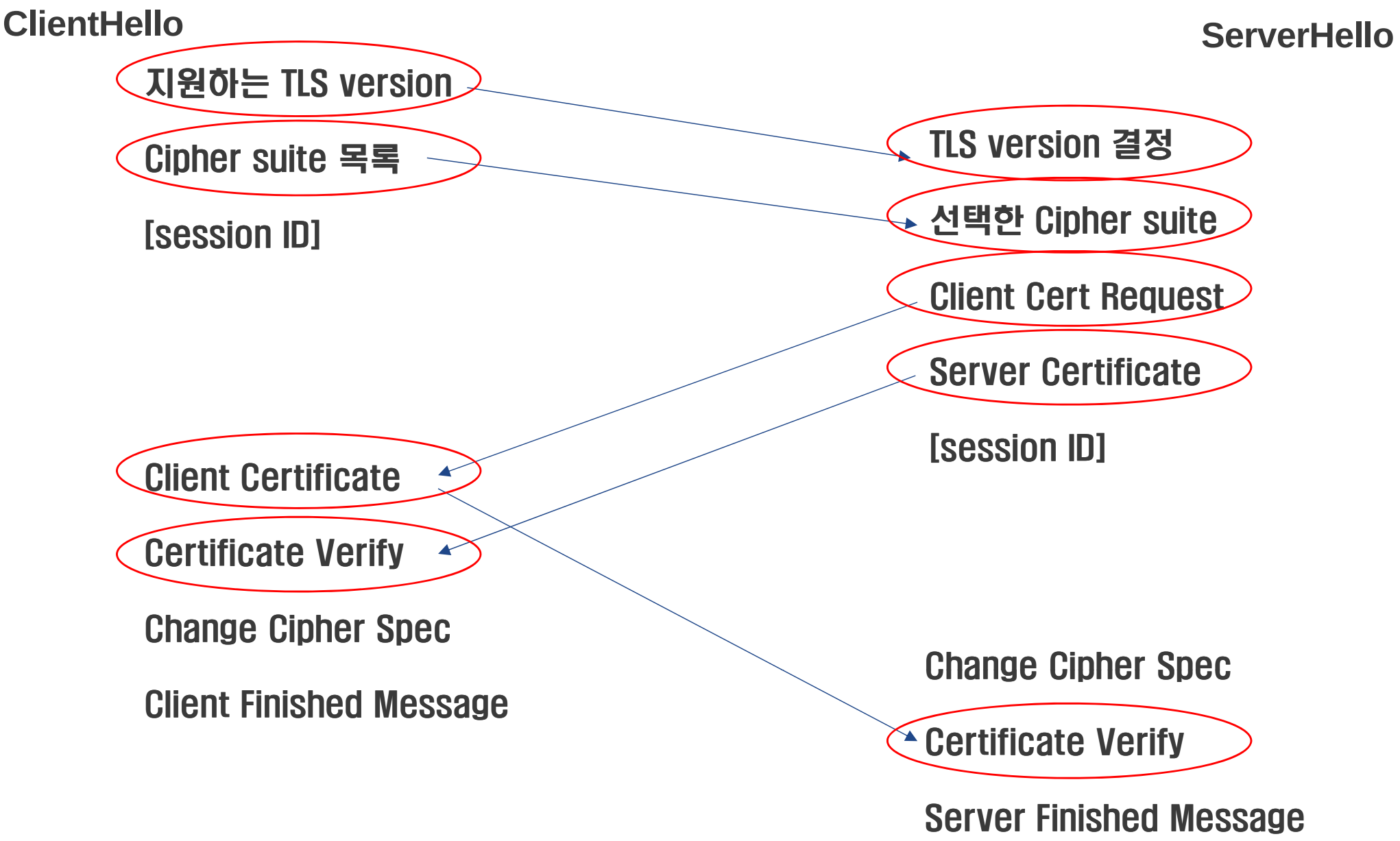
4. [C] (전문 용어) (과학 실험·의료 치료의) 계획서

출처:Oxford Advanced Learner's English-Korean Dictionary

Protocol



Cipher Suite



먼저 Version

HTTP over TLS

SSL 1.0
SSL 2.0 – 1995
SSL 3.0 – 1996

MD5

TLS 1.0 – 1999
TLS 1.1 – 2006
TLS 1.2 – 2008
TLS 1.3 – draft

SHA

주의하지 않으면 쓰나마나

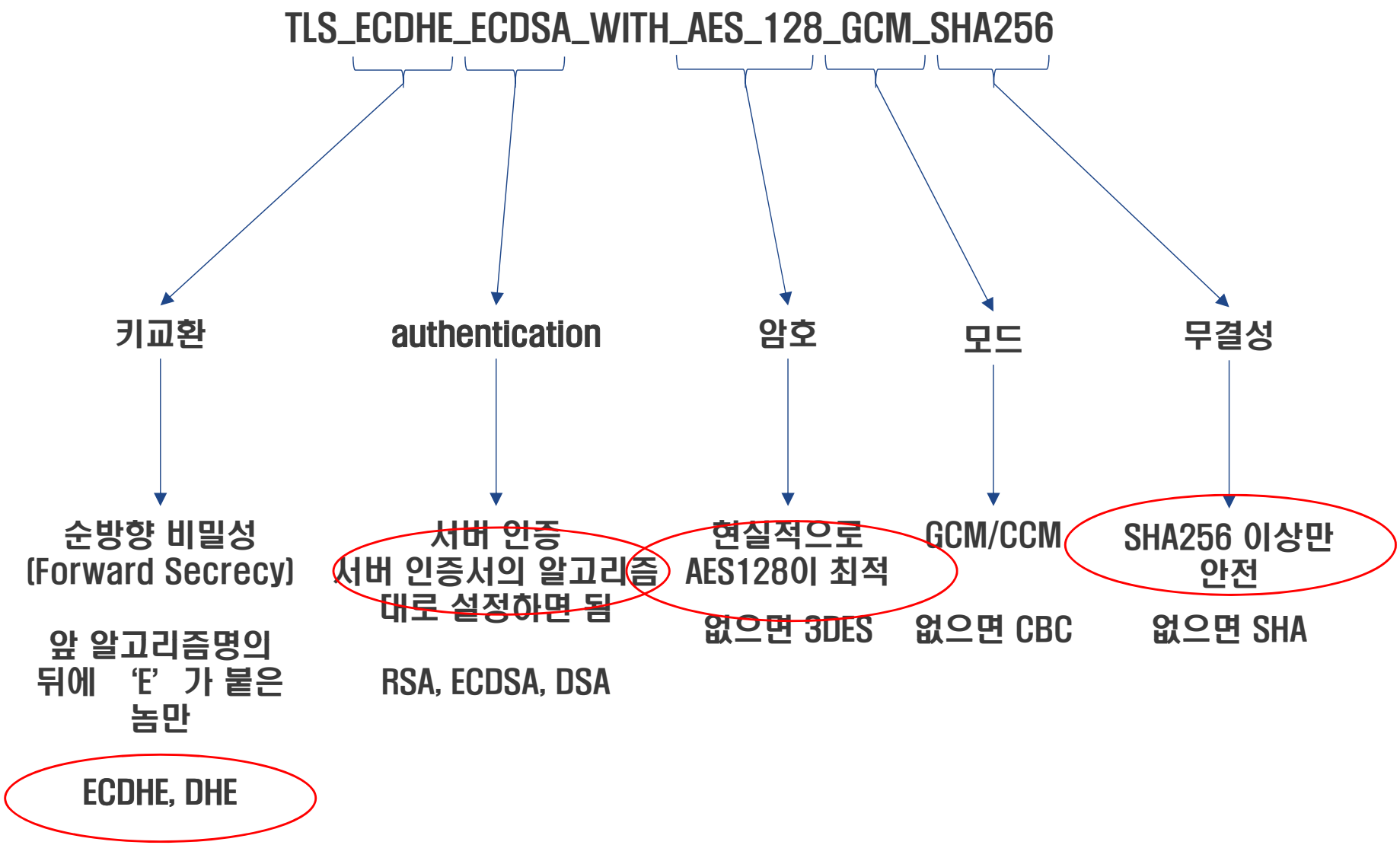
다음은 Cipher Suite

안전한 Cipher Suite - https://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml

Value	Description	DTLS-OK	Reference
0x00,0x9E	TLS_DHE_RSA_WITH_AES_128_GCM_SHA256	Y	[RFC5288]
0x00,0x9F	TLS_DHE_RSA_WITH_AES_256_GCM_SHA384	Y	[RFC5288]
0xC0,0x2B	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	Y	[RFC5289]
0xC0,0x2C	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	Y	[RFC5289]
0xC0,0x2F	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	Y	[RFC5289]
0xC0,0x30	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	Y	[RFC5289]
0xC0,0xA2	TLS_DHE_RSA_WITH_AES_128_CCM_8	Y	[RFC6655]
0xC0,0xA3	TLS_DHE_RSA_WITH_AES_256_CCM_8	Y	[RFC6655]
0xC0,0xAC	TLS_ECDHE_ECDSA_WITH_AES_128_CCM	Y	[RFC7251]
0xC0,0xAD	TLS_ECDHE_ECDSA_WITH_AES_256_CCM	Y	[RFC7251]
0xC0,0xAE	TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8	Y	[RFC7251]
0xC0,0xAF	TLS_ECDHE_ECDSA_WITH_AES_256_CCM_8	Y	[RFC7251]

Cipher Suite

구조



Cipher Suite

내 브라우저는? https://www.ssllabs.com/ssltest/viewMyClient.html

안전

취약

Cipher Suites (in order of preference)

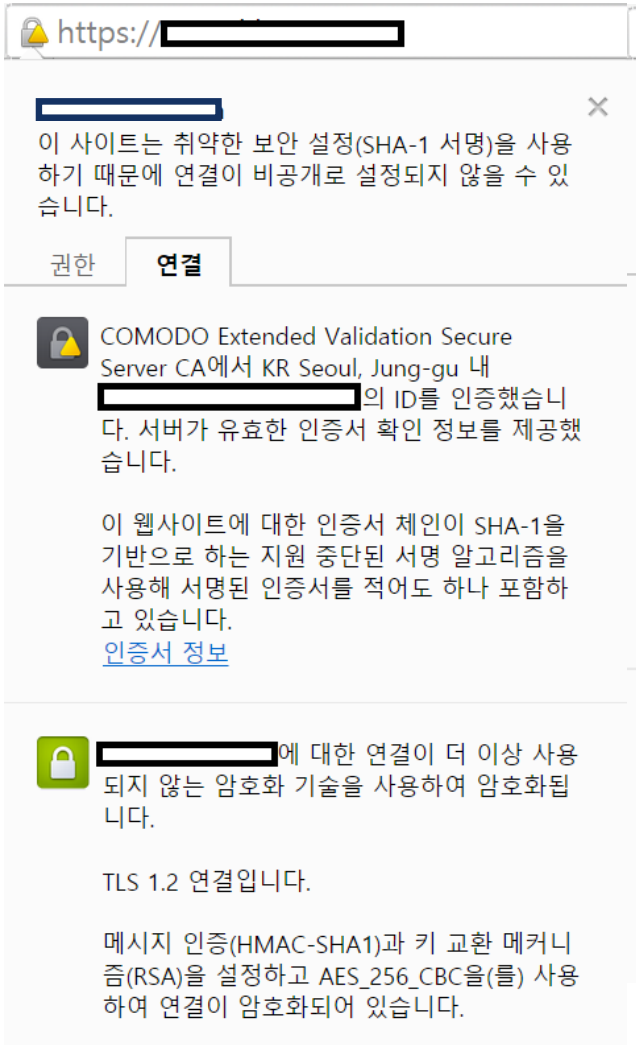
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)	Forward Secrecy	128
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)	Forward Secrecy	128
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (0x9e)	Forward Secrecy	128
TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xcc14)	Forward Secrecy	256
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcc13)	Forward Secrecy	256
TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcc15)	Forward Secrecy	256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a)	Forward Secrecy	256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)	Forward Secrecy	256
TLS_DHE_RSA_WITH_AES_256_CBC_SHA (0x39)	Forward Secrecy	256
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009)	Forward Secrecy	128
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)	Forward Secrecy	128
TLS_DHE_RSA_WITH_AES_128_CBC_SHA (0x33)	Forward Secrecy	128
TLS_RSA_WITH_AES_128_GCM_SHA256 (0x9c)		128
TLS_RSA_WITH_AES_256_CBC_SHA (0x35)		256
TLS_RSA_WITH_AES_128_CBC_SHA (0x2f)		128
TLS_RSA_WITH_3DES_EDE_CBC_SHA (0xa)		112

Forward secrecy

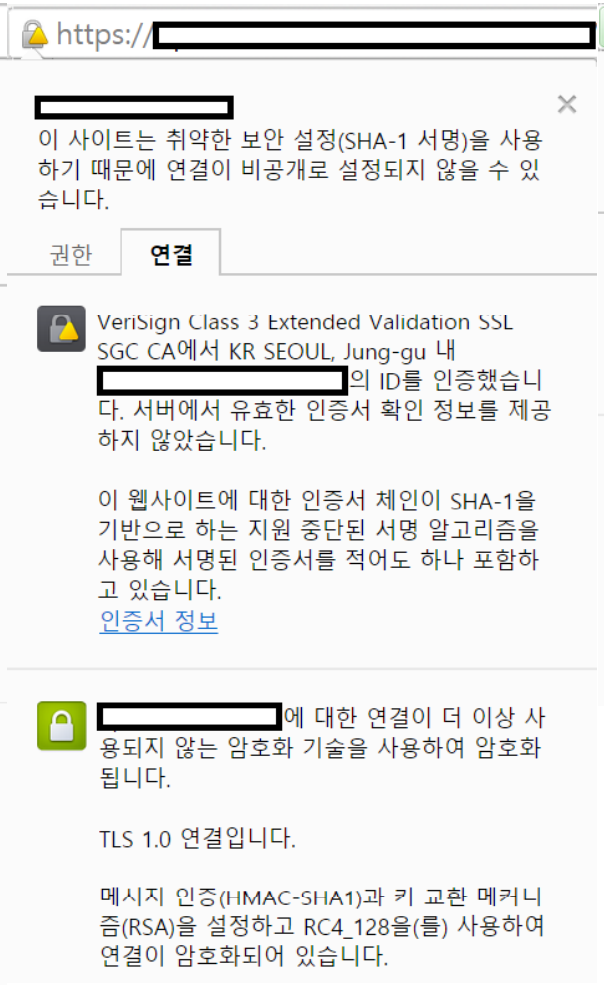
서버에서도 이와 같이 설정 해야 …

국내에서는 HTTPS를 어떻게 하고 있나?

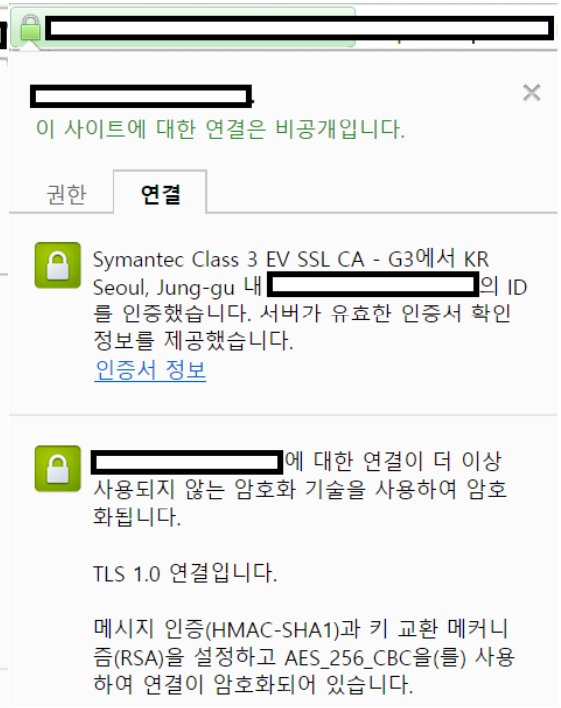
클라이언트와 무관한 Cipher Suites 설정, 들쑥날쑥한 보안 강도 설정, Forward Secrecy 무시(은행)



SHA1 인증서
AES256_CBC, SHA1, RSA



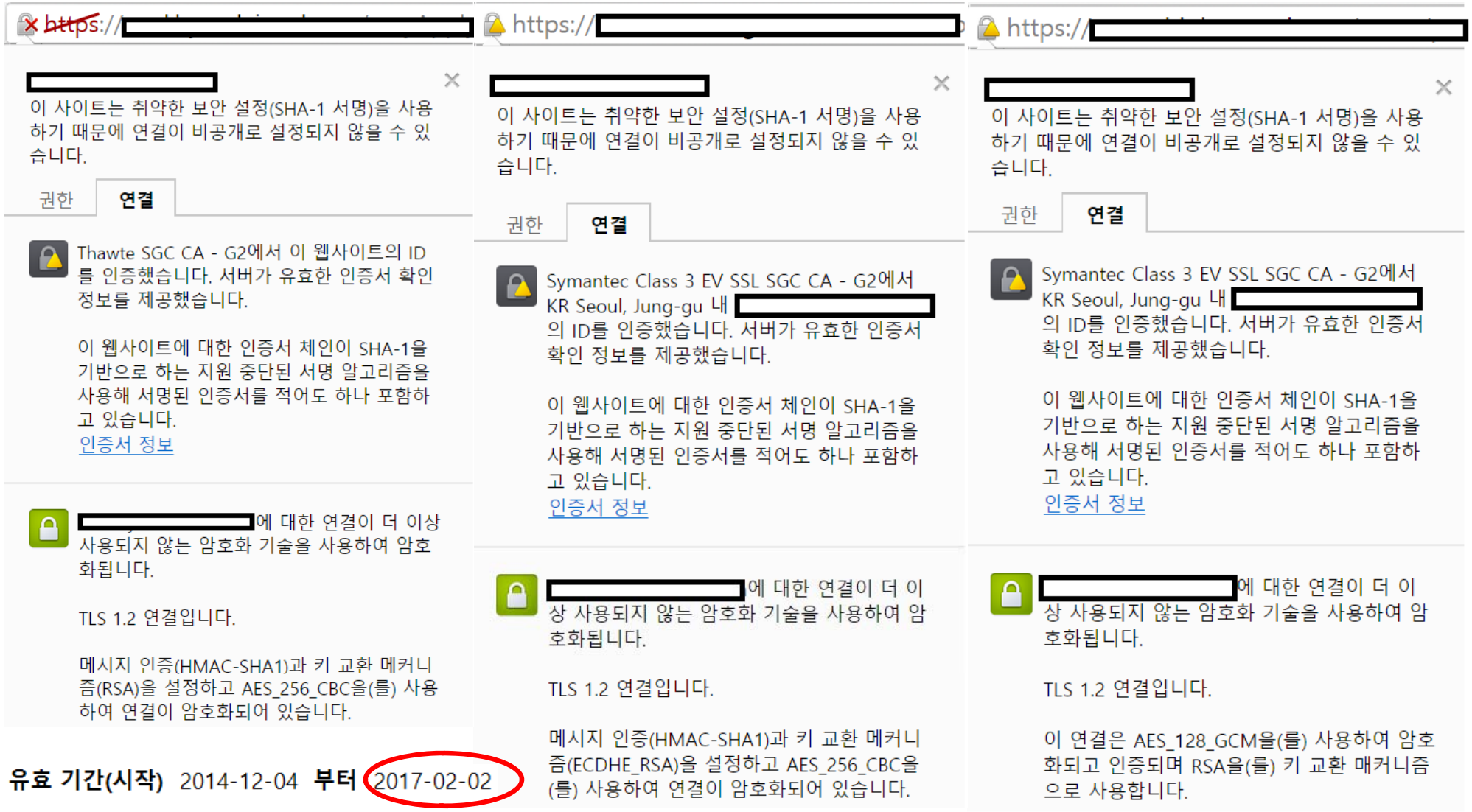
SHA1 인증서,
TLS1.0, RC4, SHA1, RSA



TLS1.0
AES256_CBC, SHA1. RSA

국내에서는 HTTPS를 어떻게 하고 있나?

클라이언트와 무관한 Cipher Suites 설정, 들쑥날쑥한 보안 강도 설정, Forward Secrecy 무시, 부적절한 유효기간(카드)



지금까지 전용 솔루션을 써서 안전했다고?

HTTPS만이 문제가 아님도 알아야 한다 …
보안업체들이 제공하는 프로토콜은 아직도 10년전 …

```
TLSv1 Record Layer: Handshake Protocol: Multiple Handshake Messages
  Content Type: Handshake (22)
  Version: TLS 1.0 (0x0301)
  Length: 4845
⊟ Handshake Protocol: Server Hello
    Handshake Type: Server Hello (2)
    Length: 84
    Version: TLS 1.0 (0x0301)
  ⊞ Random
    Session ID Length: 46
    Session ID: 433041383832324131383945303135323039323244454545...
    Cipher Suite: Unknown (0x0103)
    Compression Method: null (0)
⊞ Handshake Protocol: Certificate
⊟ Handshake Protocol: Certificate Request
    Handshake Type: Certificate Request (13)
    Length: 919
    Certificate types count: 1
  ⊞ Certificate types (1 type)
    Distinguished Names Length: 915
  ⊟ Distinguished Names (915 bytes)
      Distinguished Name Length: 76
    ⊞ Distinguished Name: (id-at-commonName=yessignCA,id-at-organiz
```

Mitigation 가능

SEEDCBC
HAS 160

다행히 인증서는 공인을 사용하기 때문에
SHA256
그러나 **순방향 비밀성은 없음**

SHA1은 장기사용은 X

중 여성과학자, 미국 암호체계 뚫었다

[란보라의 중국속으로] 40세 여교수, 암호체계 연속격파, 정보업계 경악

2005년, 대자보

란보라

[뉴스로 보는 중국인들의 심성_24] 세계를 놀라게 한 중국 과학계의 괘거

세계 암호영역의 양대 보루가 모두 중국 산둥대학 정보연구소의 여성 과학자 40세의 왕샤오윈(王小雲)소장이 이끄는 연구팀에 의해 격파되었다.

세상에 알려지지 않았던 왕 소장은 하루 만에 세계 명인이 되었고, 암호연구 영역에 알려지지도 않았던 산둥대학 정보연구소는 급기야 세계 암호연구영역에서 가장 주목을 받는 연구소로 되었으며, 국제 암호연구계는 충격에 휩싸여있다.

미국에서는 미국의 암호영역이 중국 전문가에 의해 격파되었음을 시인하고 미국의 정보안전이 위험에 처했다고 했다.

국제 전문가들은 정보안전에 대해 다시 연구하고, 암호계산법을 새로이 만들어야 할 일이 긴박한 시점에 와있음을 분분히 주장했다.

▲금년에 40세에 나는 왕샤오윈(王小雲) 산둥대학 정보연구소 소장. 그는 자기의 연구팀을 이끌어 세계 암호영역의 2대 보루인 MD5와 SHA-1 암호표준을 격파했다. ©자료사진

충돌쌍을 찾는데 1/2048. 170년 걸릴 것이 1달 소요

10년 이상 유지되는 SHA1 CA인증서?
그런 CA에서 발급된 서버 인증서는?

그래서 해외에서는 SNS도 …

금융보다 소중하지 않은 SNS는 어떻게 하고 있을까? 상호인증 빼고는 …

SHA256 인증서 사용. TLS1.2 지원, SHA256 사용
순방향비밀성 지원, AES_128_GCM사용

정확하게 알고 가장 효율적이고 안전하게 사용

국내는 공인인증서는 2010년에 SHA256으로 업글

서명 알고리즘	sha1RSA
서명 해시 알고리즘	sha1
발급자	KISA RootCA 1, Korea Certi...
유효 기간(시작)	2005년 8월 24일 수요일 오...
유효 기간(끝)	2025년 8월 24일 일요일 오...
주체	KISA RootCA 1, Korea Certi...
공개 키	RSA (2048 Bits)

- Sequence (of)
  - Object Identifier iso(1) member-body(2) US(840) rsadsi(113549) pkcs(1) pkcs-1(1) rsaEncryption(1)
  - NULL
- Bit String
  - Sequence (of)
    - Integer
    - Integer 3

서명 알고리즘	sha256RSA
서명 해시 알고리즘	sha256
발급자	KISA RootCA 4, Korea Certi...
유효 기간(시작)	2010년 7월 12일 월요일 오...
유효 기간(끝)	2030년 7월 12일 금요일 오...
주체	KISA RootCA 4, Korea Certi...
공개 키	RSA (2048 Bits)

- Sequence (of)
  - Object Identifier iso(1) member-body(2) US(840) rsadsi(113549) pkcs(1) pkcs-1(1) rsaEncryption(1)
  - NULL
- Bit String
  - Sequence (of)
    - Integer
    - Integer 65537

국내 금융권은 2016년 서버인증서를 SHA256으로 업글 예정

“어쩔 수 없이”가 아닌 “고객과 비즈니스의 영속성을 위해서” 하면 훨씬 더 좋을 텐데 …

Symantec.

대한민국

제품 및 솔루션 지원 및 커뮤니티 보안 연구소 Try & Buy

/ 1024비트 인증서 지원

SSL 및 Code Signing 인증서를 위한 SHA-1 해시 알고리즘 마이그레이션

SHA-1을 SHA-2 인증서로 대체

Microsoft 및 Google은 이르면 2015년 12월 31일에 만료되는 SHA-1 인증서를 사용하는 웹 사이트에 영향을 미칠 수 있는 HA-1 사용 중단 계획을 발표했습니다.

"SHA-1 점진적 하위 집합화"에 대한 Google 블로그에 따르면 Chrome 버전 39 이상은 2016년 1월 1일 이후까지 유효한 SHA-1 SSL 인증서를 사용하는 사이트에 시작적 보안 표시기를 표시합니다. Chrome 39의 프로덕션 릴리스는 2014년 11월에 발표될 것으로 예상됩니다. 사이트들은 다음의 표시기 중 하나로 취급됩니다. "안전하지만 경미한 오류가 있음"(노란색 삼각형 안의 자물쇠), "중립, 보안 결여"(검은색 페이지 아이콘) 및 "확실히 안전하지 않음"(빨간색 X 표시의 자물쇠). Chrome 버전 39 이상을 사용하는 온라인 사용자에게 이러한 표시기가 표시되는 것을 방지할 수 있도록 2015년 12월 31일 이후에 만료되는 HA-1 SSL 인증서는 SHA-256 (SHA-2) 인증서로 대체되어야 합니다.

이미 오래 전부터 했어야 하지만 …

정책적으로 이미 정해진 일정 …

SHA-1서명,2년,3년 인증서 주문불가.
SHA-1 인증서 발급의 유효기간은 2016년 12월 31일로 제한
11. 2014

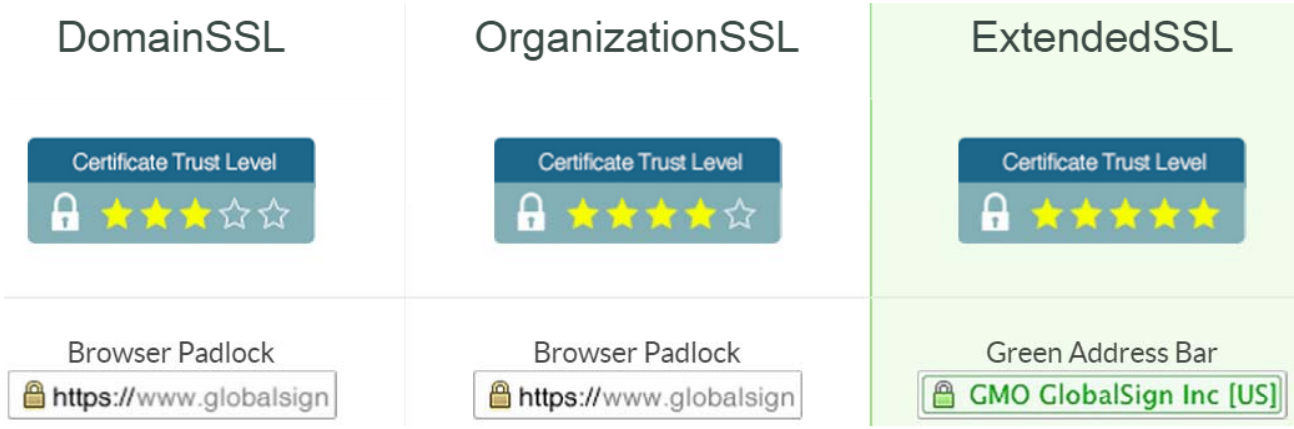
4년, 5년 다년인증서 발급중단.
인증서 재발급은 최대 39개월로 제한
03 2015

SHA-1 인증서 발급 및 재발급 중단
12. 2015

Microsoft는 SHA-1 서명발급된 모든 SSL, CA 인증서 신뢰중단
01. 2016

인증

서버 인증서



서명 알고리즘	sha256RSA
서명 해시 알고...	sha256
발급자	DigiCert SHA2 High Assurance Server
유효 기간(시작)	2014년 8월 28일 목요일 오전 9:00:00
유효 기간(끝)	2015년 12월 31일 목요일 오후 9:00:0(
주체	*.facebook.com, Facebook, Inc., Menl
공개 키	ECC (256 Bits)
공용 키 매개 변수	ECDSA_P256

서명 알고리즘	sha256RSA
서명 해시 알고...	sha256
발급자	COMODO RSA Organization Validation Se...
유효 기간(시작)	2015년 4월 28일 화요일 오전 9:00:00
유효 기간(끝)	2016년 4월 25일 월요일 오전 8:59:59
주체	works.naver.com, Unified Communications,...
공개 키	RSA (2048 Bits)

SSLCipherSuite ECDHE-ECDSA-AES128-SHA256:AES128-GCM-SHA256:HIGH:+TLSv1.2:!MD5:!aNULL

or

SSLCipherSuite ECDHE-RSA-AES128-SHA256:AES128-GCM-SHA256:HIGH:+TLSv1.2:!MD5:!aNULL

SSLCertificateFile "${SRVROOT}/conf/ssl/flyhigh.com.crt“
SSLCertificateKeyFile "${SRVROOT}/conf/ssl/flyhigh.com.key"

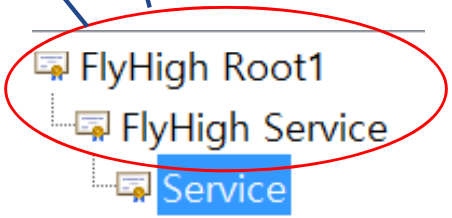
사용자 인증서

```
#   Certificate Authority (CA):

SSLCACertificateFile "${SRVROOT}/conf/ssl/service.crt"

SSLVerifyClient require

#length of chain
SSLVerifyDepth  2
```

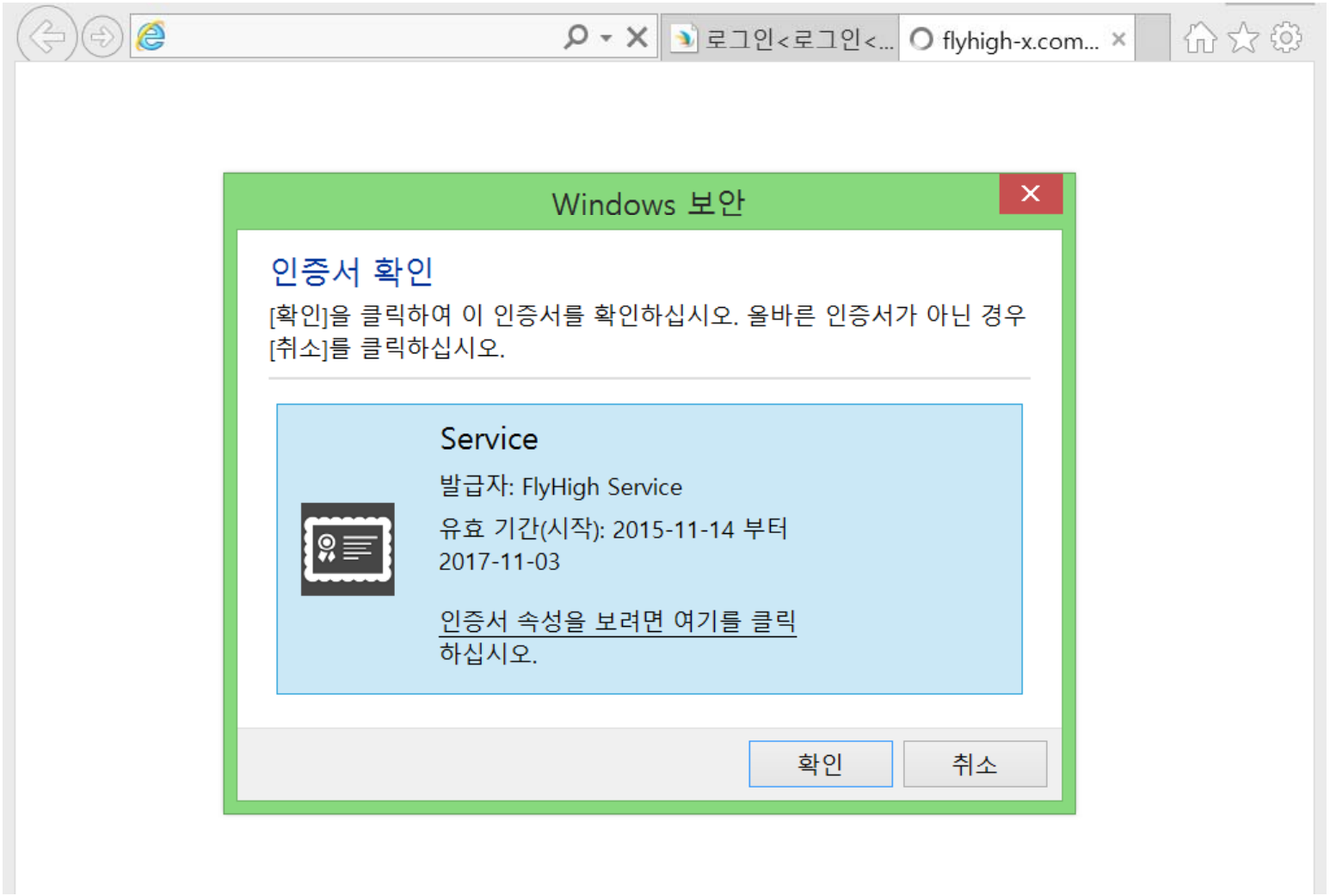
```
SSLVerifyClient optional
SSLVerifyDepth 3
RewriteEngine On
RewriteCond %{SSL:SSL_CLIENT_VERIFY} !^SUCCESS$
RewriteRule .* /help/ssl-client-auth-required.html
```

인증서 없으면 바로 에러

인증서 없으면 안내 페이지로 이동

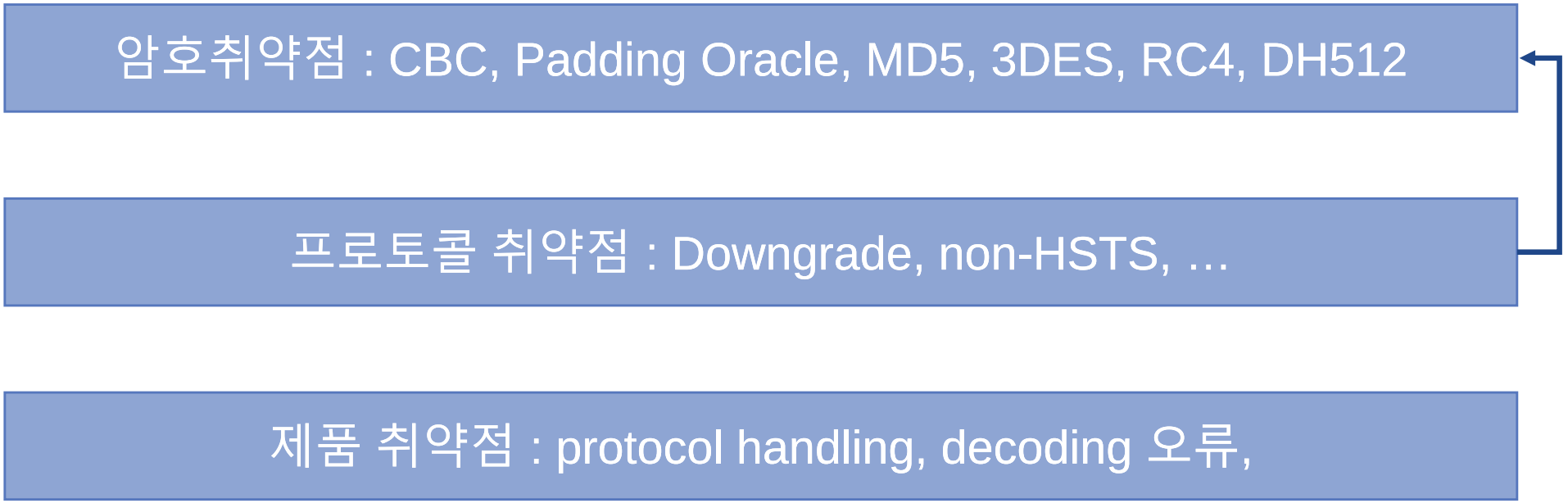


사용자 인증서 : ActiveX, Plugin, Exe 없이도 상호인증 가능(가장 안전한 HTTPS 사용 방법)



HTTPS에 대한 공격

주요 공격 패턴



HTTPS에 대한 공격

주요 취약점

취약점	내용	대상
취약한 키 유도	MD5	~SSL 3.0
Cipher Suite Downgrade	handshake	~SSL 3.0
POODLE Attack	CBC + Downgrade	SSL 3.0
RC4 Attack		SSL/TLS
Truncation attack	로그아웃 차단	SSL/TLS
FREAK attack **Logjam attack**	OpenSSL 512-bit DH 무기수출통제정책	SSL/TLS
Heartbleed bug **BERserk attack**	OpenSSL 일부 제품의 ASN.1 decoding 오류	SSL/TLS
Timing attacks on padding	Padding Oracle Attack	~TLS 1.1 AES_GCM만 안전

https://en.wikipedia.org/wiki/Arms_Export_Control_Act

그래서 TLS Spec을 계속 수정하고 있다.

TLS History : ActiveX를 도입했던 이유가 뭐였었는지도 잊었는지 …
이제는 제발 흐름을 따라 가기만이라도
서버와 클라이언트를 안전하게 연결하기 위한 보안 프로토콜

TLS 1.0(2246) ---------- 1999(암호강화, 효율)
AES Ciphersuites for TLS(3268) ---------- 2002
TLS Extensions (3546) ---------- 2003

TLS 1.1(4346) ---------- 2006(IV명시)
TLS Extensions (4366) ---------- 2006
ECC Cipher Suites TLS(4492) ---------- 2006

TLS 1.2(5246) ---------- 2008(SHA256도입)
TLS Authorization Extensions(5878) ---------- 2010
TLS Renegotiation Indication Extension(5746) ---------- 2010
Prohibiting Secure Sockets Layer (SSL) Version 2.09(6176) ---------- 2011
TLS Fallback SCSV for Preventing Protocol Downgrade Attacks(7507) ---- 2015
Prohibiting RC4 Cipher Suites(7465) ---------- 2015
Deprecating SSL Version 3.0(7568) ---------- 2015
TLS Session Hash and Extended Master Secret Extension(7627) ---------- 2015

TLS 1.3(draft 9)---------- 2015(RSA-PSS, SHA1 X)

The SHA-3 standard was released by NIST on August 5, 2015. 조만간 반영되겠죠..

HTTPS만 가지고는 안전하게 지킬 수가 없다.

웹을 더욱 안전하게 하기 위해서 W3C와 IETF는 지속적으로 새로운 보완기술을 도입하고 있다

Considerations for Web Transaction Security(2084)------- 1997

The Web Origin Concept(6454)---------------------------- 2011

HTTP Strict Transport Security(6797)--------------------- 2012

Public Key Pinning Extension for HTTP(7469)-------------- 2015

System for Cross-domain Identity Management(7642)------- 2015

Content Security Policy Pinning ------------------------ 2015

Content Security Policy Level 2 ------------------------ 2015

HTTPS에 대한 공격

SSL Strip/SSL Proxy과 방어

HSTS(HTTP Strict Transport Security)

Strict-Transport-Security: max-age=15768000 ; includeSubDomains

SSL Proxy

HPKP(HTTP Public Key Pinning)

HTTPS 성능과 Mixed Contents

30~40%는 느려진다. 특히 CPU의 부하가 크다

가속장비 사용 : 컨텐츠 관리 …

Image를 http로 처리 : 모든 브라우저에서 허용했음, iFrame 랩핑

의도하지 않은 정보유출

Type	Cookie stealing	Request forgery	DOM data leakage	JavaScript execution
Image	x	x		
iframe	x	x		
CSS	x	x	x	
JavaScript	x	x	x	x
Flash	x	x	x	x

Table 1: Impact of mixed content attacks

http://www.securitee.org/files/mixedinc_isc2013.pdf

HTTPS 성능과 Mixed Contents

보안을 위해서 하는 것이면 조금 더 주의하자

의도하지 않은 정보유출

HSTS는 별 효과 없음. iFrame랩핑 금지(경고가 뜨지 않기도 함)

CSP(Content Security Policy) : 원하지 않는 컨테츠 유입 차단

```
EXAMPLE 5
Content-Security-Policy: default-src 'self' http://example.com http://example.net;
                         connect-src 'none';
Content-Security-Policy: connect-src http://example.com/;
                         script-src http://example.com/
```

최신 W3C Spec의 특징 활용

보안 제품으로만 보안을 하는 것이 아님

```
Content-Security-Policy-Pin: max-age: 10886400;
                             includeSubDomains;
                             default-src https:;
                             form-action 'none';
                             frame-ancestors 'none';
                             referrer no-referrer;
                             report-uri /csp-endpoint/pinned
```

3.2. Content-Security-Policy-Report-Only-Pin Header Field

The ***Content-Security-Policy-Report-Only-Pin*** header field is the mechanism for delivering a pinned policy that the user agent MUST monitor for any resource which is not delivered with a `Content-Security-Policy-Report-Only` header (as described in the §4.1.3 Pin a policy to response algorithm).

The ABNF grammar is as follows:

```
"Content-Security-Policy-Report-Only-Pin:" 1#<policy-token production from CSP, Section 4.1>
```

HTTPS 설정

1) 먼저 서버 업그레이드를 하세요

2) 최신 브라우저 사용자를 배려하세요

3) 최소 보안의 한계를 정하세요

4) 낮은 버전 브라우저
사용자에게 안내하세요.

잘 못 사용하면
보안에 아무런 도움이 되지 않습니다.

HTTPS 보안 강화

1) 서버 인증서 관리를 안전하게 하세요

2) 컨텐츠를 섞지 마세요

3) HSTS, HPKP, CSP를 잘 활용하세요

4) CORS를 사용할 때 점검을 잘 하세요

보안은 보안제품만으로 하는 것이 아닙니다.

Thank you.

We Make You *FlyHigh*