Using

&

Optimizing **TLS**

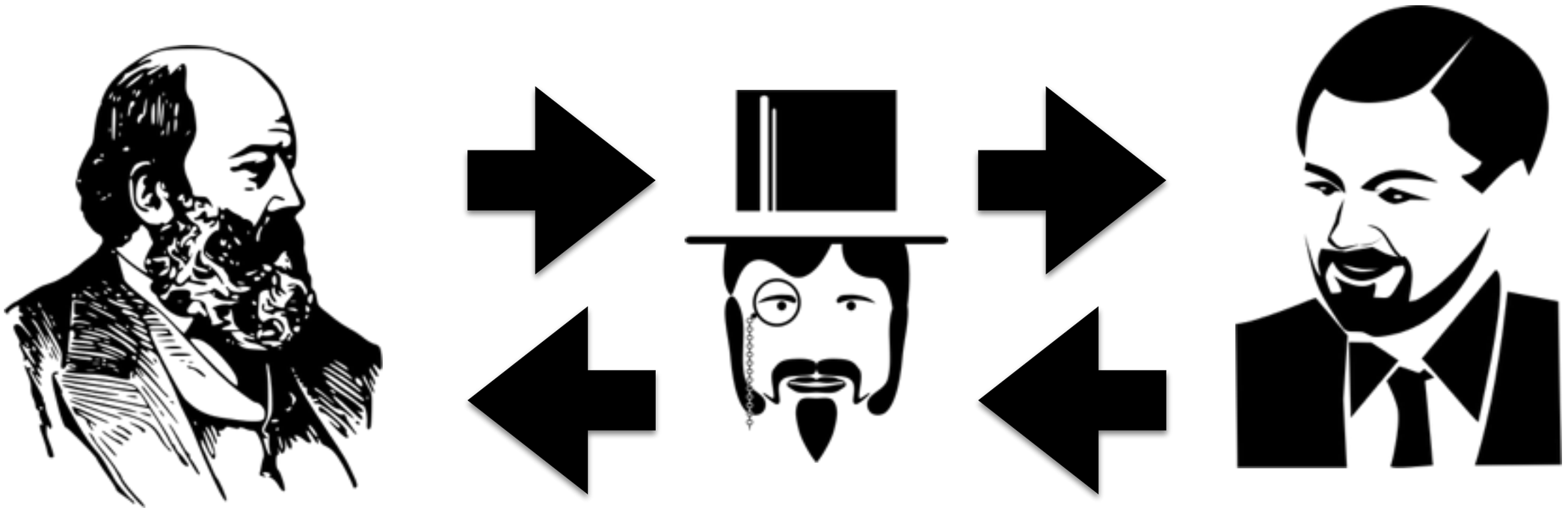
HTTP

HTTPS

E-commerce

Identity Theft

Man in the Middle



Observe and/or Manipulate Content

Content Injection

Mass Surveillance

credit: EFF

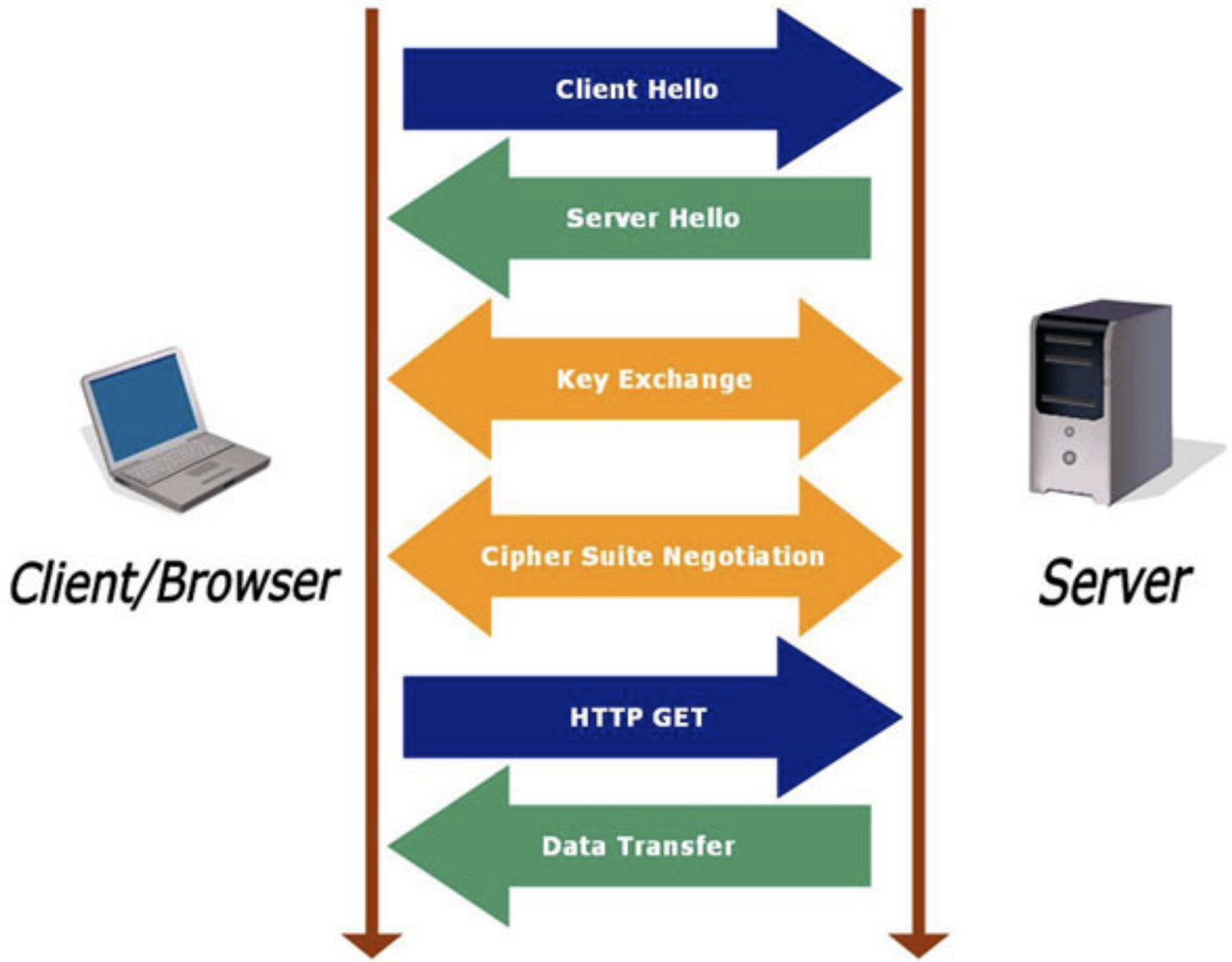
HTTPS Barriers

Cost

Technical

Motivation

SSL & TLS

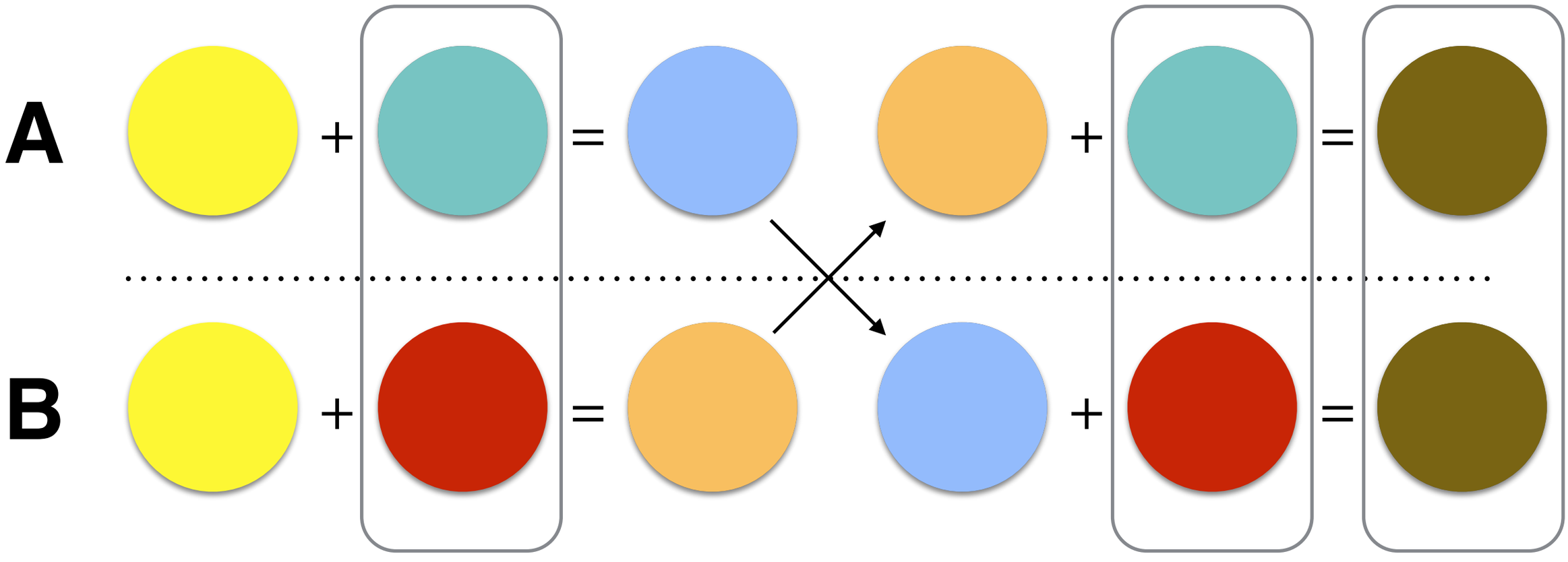

Client Hello
Server Hello
Key Exchange
Cipher Suite Negotiation
HTTP GET
Data Transfer
Client/Browser
Server

RSA Private Key Size

Strong	**>= 2048**
Weak	**< 2048**

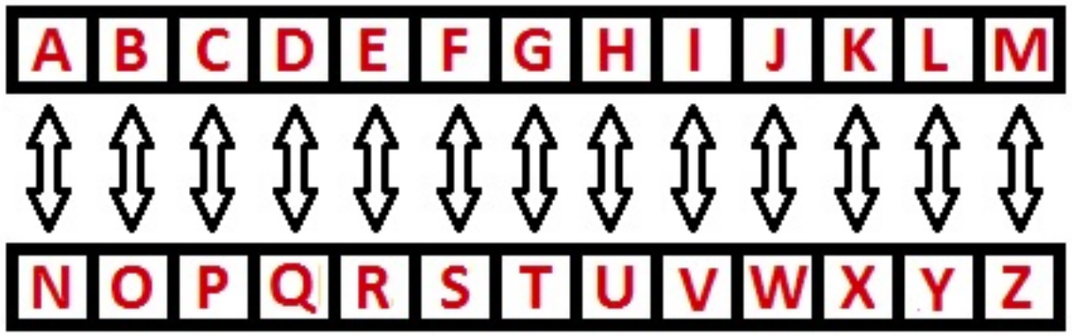
Forward Secrecy

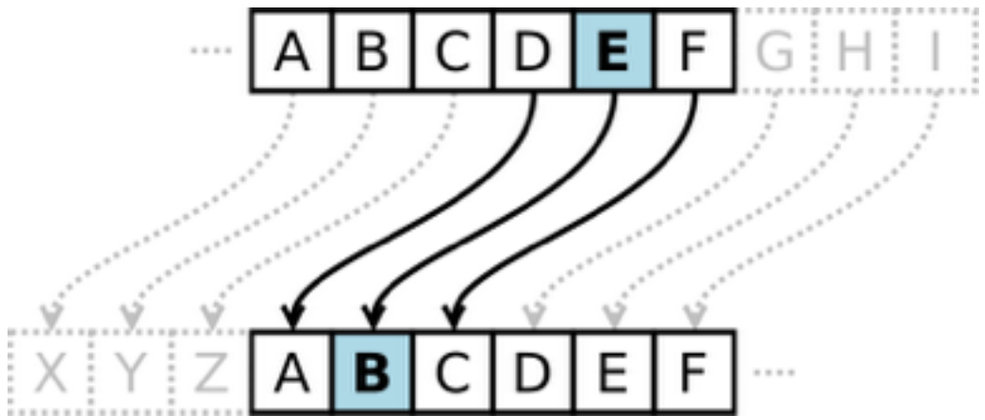
- DHE -Diffie-Hellman Ephemeral
- ECDHE - Elliptic Curve Diffie-Hellman Ephemeral

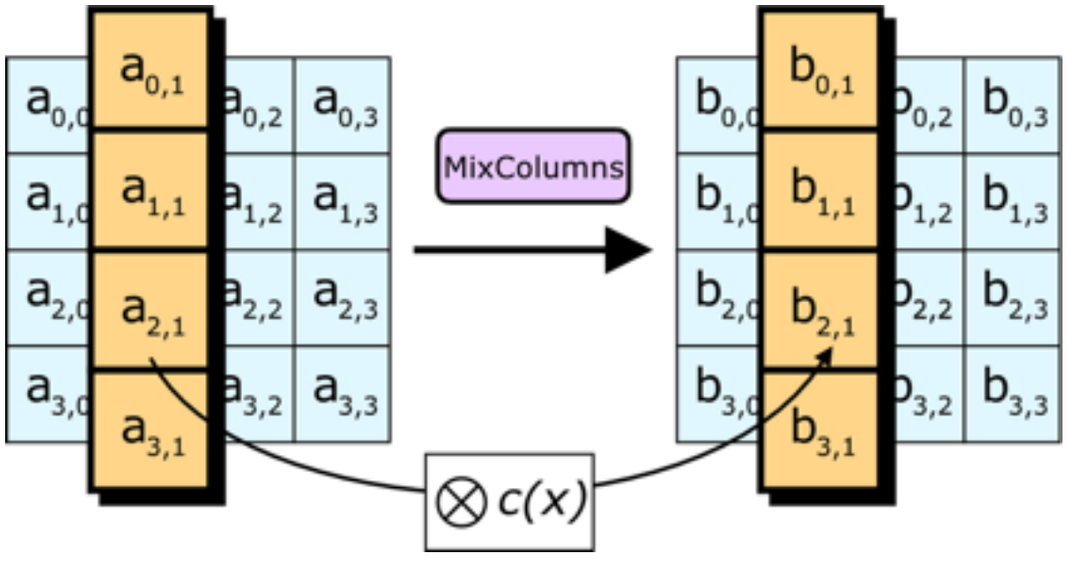


Confidentiality

Cyphers


A B C D E F G H I J K L M
N O P Q R S T U V W X Y Z


A B C D E F G H I
X Y Z A B C D E F


MixColumns
c(x)

Cipher Strength

Strong	**AES 128/256**
Weak	**3DES**
Broken	**RC4**

Recommended Ciphers

Block	AES-GCM/CCM/CBC*
Stream	ChaCha20-Poly1305*

Integrity

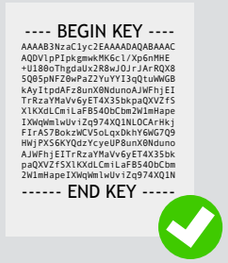
Message Authentication Code

Strong	**SHA-2+/ AEAD**
Weak	**SHA1**
Broken	**MD5**

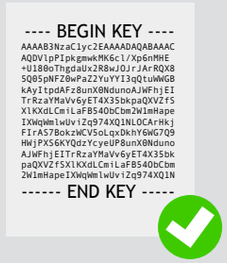
Authenticity

Root Certificates

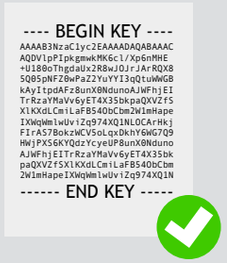
(Installed in your Browser)



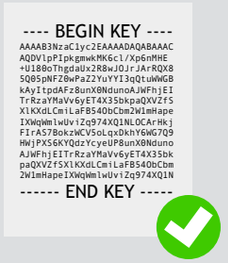
GoDaddy



Comodo



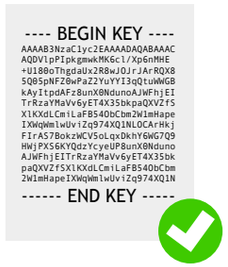
Symantec



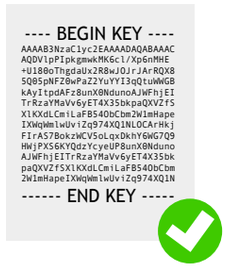
DigiCert

Intermediate Certificates

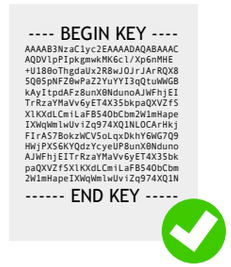
(Signed by a Root Certificate)



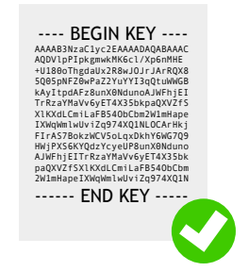
GoDaddy



Comodo



Symantec



DigiCert

Website Certificates

(Signed by an Intermediate Certificate)

openwest.org

eff.org

dc801.org

Certificate Lifecycle

Certificate Authority

Generate Private Key

Generate CSR

Submit CSR to CA

Validate Request

Install Certificate

Generate Certificate

Provide Certificate

Configure Webserver

Types of Validation

Domain	Controls the Domain
Organization	Prove Organization's Legal Entity
Extended	Manual Verification of Organization

Certificate Authorities

Monopoly

Cartel


GoDaddy.COM


COMODO
Creating Trust Online®


Symantec


GlobalSign


digicert®


StartCOM®


NetworkSolutions®

Alternatives

Using

ACME

Automated Certificate Management Environment

- Prove to the CA that it controls one or more domains(s).
- Request, renew, and/or revoke certificates for the domain(s)

ACME Agents

- CertBot (Recommended - EFF)
- Caddy (Easy & Feature Rich)
- ZeroSSL (Browser)
- ACME Tiny (< 200 lines of Python)
- Letsencrypt-plugin (Ruby on Rails)
- Certify (Windows)
- Acme-client (Java)

A more comprehensive list can be found at:
https://github.com/certbot/certbot/wiki/Links

Certbot Installation

WGET	wget https://dl.eff.org/certbot-auto chmod a+x certbot-auto
Package	Varies depending on operating system and web server

Plugins

Module	Authentication	Installation
apache	**Yes**	**Yes**
standalone	**Yes**	No
webroot	**Yes**	No
nginx	**Yes**	**No**

Examples

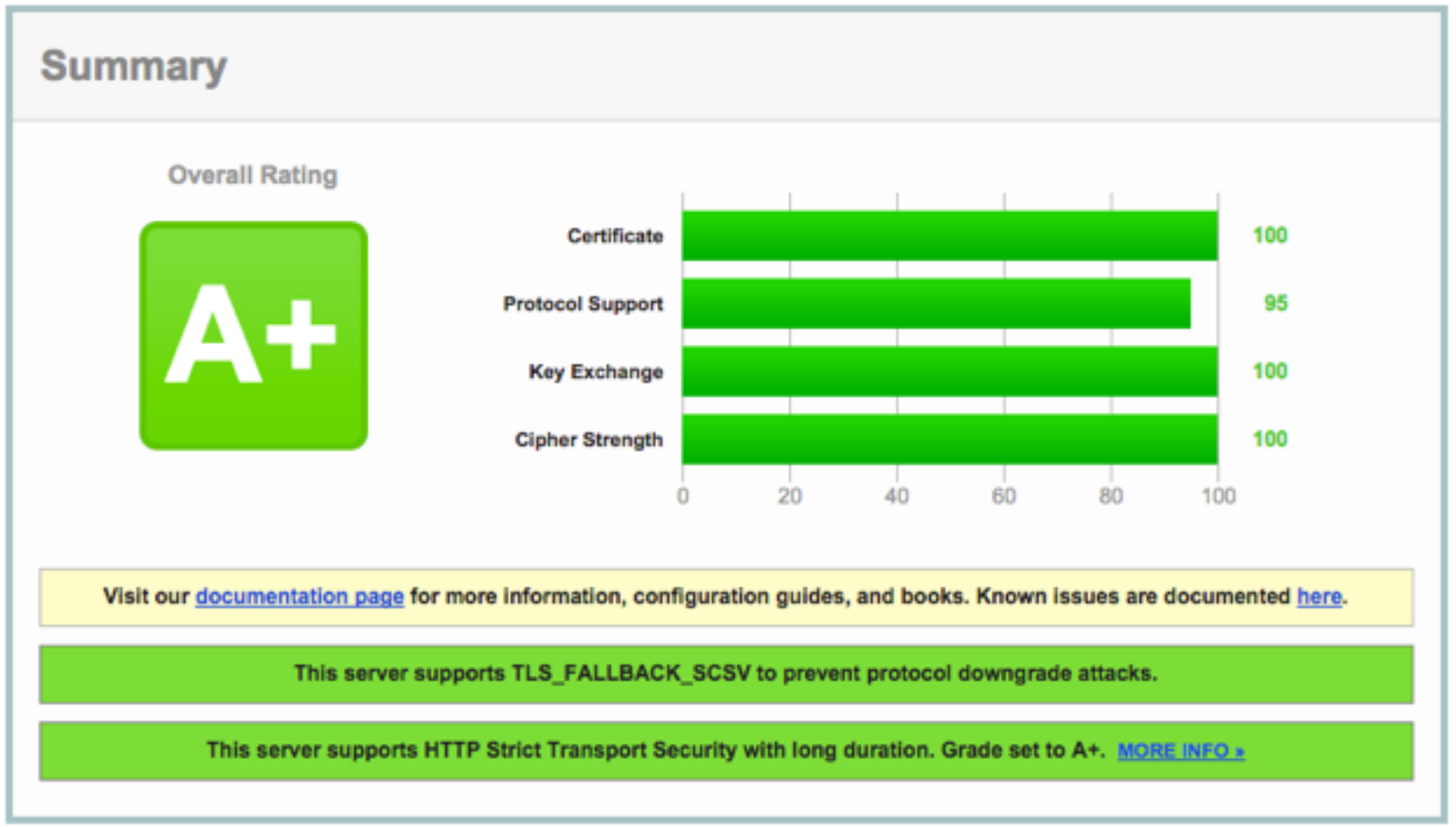
Basic/Apache	./certbot-auto
Standalone	./certbot-auto certonly --standalone
Webroot	./certbot-auto certonly --webroot
Nginx	./certbot-auto certainly
Renew	./certbot-auto renew
Help	./certbot-auto help

https://certbot.eff.org/docs/

Optimizing

TLS

Making the Grade



https://ssllabs.com

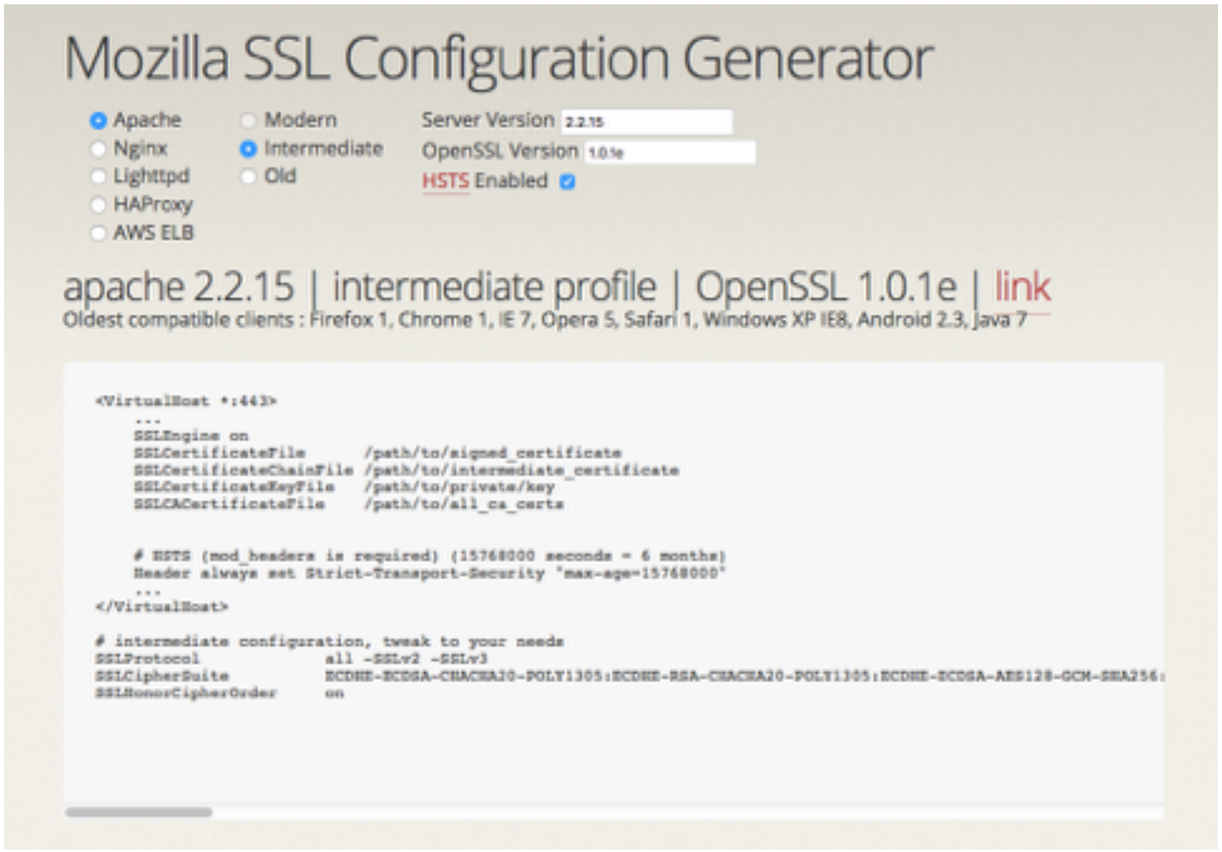
Other Scanners

- **HT Bridge** - https://www.htbridge.com/ssl
- **nmap** --script ssl-enum-ciphers -p 443 HOSTNAME
- **openssl** s_client -connect HOSTNAME:443 -prexit -showcerts -state -status -tlsextdebug -verify 10

Compatibility

Configuration

Mozilla SSL Configuration Generator



https://mozilla.github.io/server-side-tls/ssl-config-generator/

Nginx

```
server {
    listen *:80;
    return 301 https://$host$request_uri;
}
```

Nginx

```
server {
    listen *:443 default ssl;
    ssl_certificate      /etc/letsencrypt/live/domain.com/fullchain.pem;
    ssl_certificate_key  /etc/letsencrypt/live/domain.com/privkey.pem;

    ssl_protocols      TLSv1.2;
    ssl_ciphers'ECDHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES128-GCM-
SHA256:ECDHE-RSA-AES256-SHA384:ECDHE-RSA-AES128-SHA256';
    ssl_ecdh_curve secp384r1;
    ssl_prefer_server_ciphers   on;
```

Nginx

```
ssl_stapling on;
ssl_stapling_verify on;
```

```
add_header Strict-Transport-Security "max-age=31536000;
    includeSubdomains; preload";
add_header Public-Key-Pins
       'pin-sha256="YLh1dUR9y6Kja30RrAn7JKnbQG/uEtLMkBgFF2Fuihg=";
        pin-sha256="klO23nT2ehFDXCfx3eHTDRESMz3asj1muO+4aIdjiuY=";
        max-age=518400';
```

Apache

```
<VirtualHost *:80>
        ReWriteEngine on
        ReWriteCond %{SERVER_PORT} !^443$
        ReWriteRule ^/(.*) https://%{HTTP_HOST}/$1 [NC,R,L]
</VirtualHost>
```

Apache

```
<IfModule mod_ssl.c>
<VirtualHost _default_:443>
     SSLEngine on
     SSLCertificateFile    /etc/letsencrypt/live/domain.com/cert.pem
     SSLCertificateKeyFile /etc/letsencrypt/live/domain.com/privkey.pem
     SSLCertificateChainFile /etc/letsencrypt/live/domain.com/chain.pem

    SSLProtocol TLSv1_2
    SSLHonorCipherOrder On
    SSLCipherSuite AES256+EECDH:!aNULL

     SSLUseStapling on
```

Apache

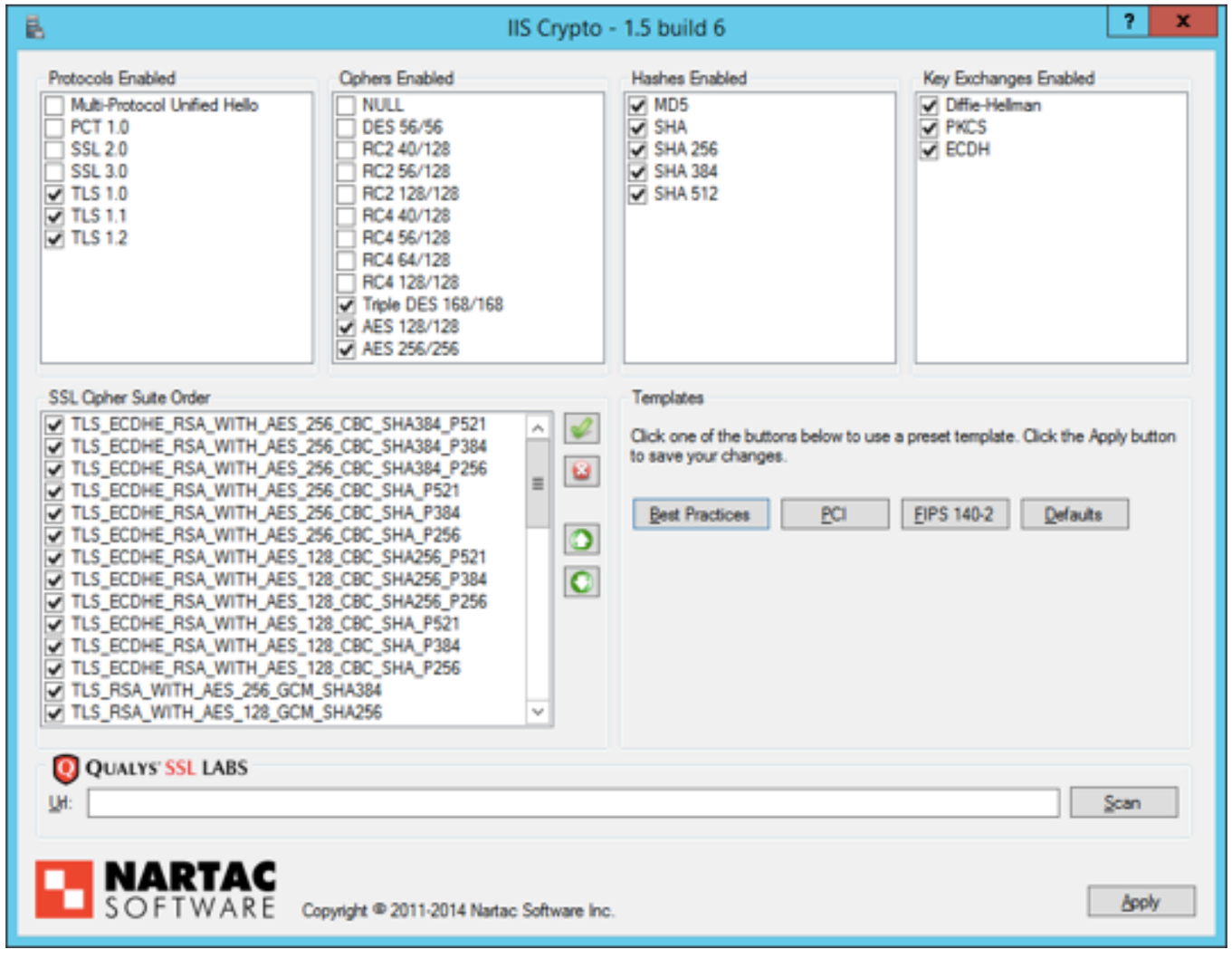
```
LoadModule headers_module modules/mod_headers.so

Header set Public-Key-Pins "pin-sha256=
    \"klO23nT2ehFDXCfx3eHTDRESMz3asj1muO+4aIdjiuY=\"; pin-sha256=
    \"633lt352PKRXbOwf4xSEa1M517scpD3l5f79xMD9r9Q=\"; max-age=2592000;
    includeSubDomains"

    Header always set Strict-Transport-Security "max-age=63072000;
includeSubDomains"

</VirtualHost>
</IfModule>
```

IIS



https://www.nartac.com/Products/IISCrypto

Extra Credit

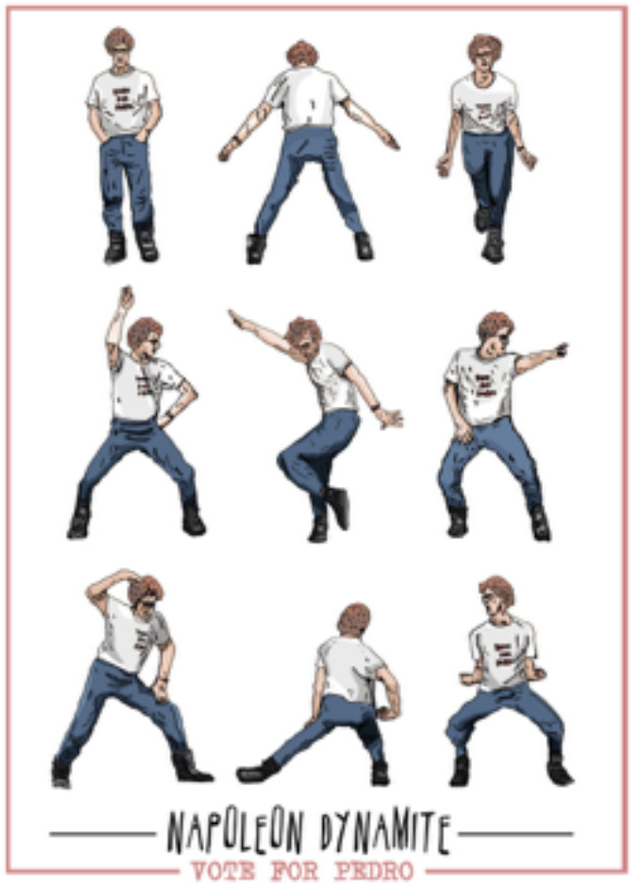
HTTP Strict Transport Security

HSTS Preloading
https://hstspreload.appspot.com

OCSP Stapling

HTTP Public Key Pinning (HPKP)

Dance like no one is watching



Encrypt like everyone is