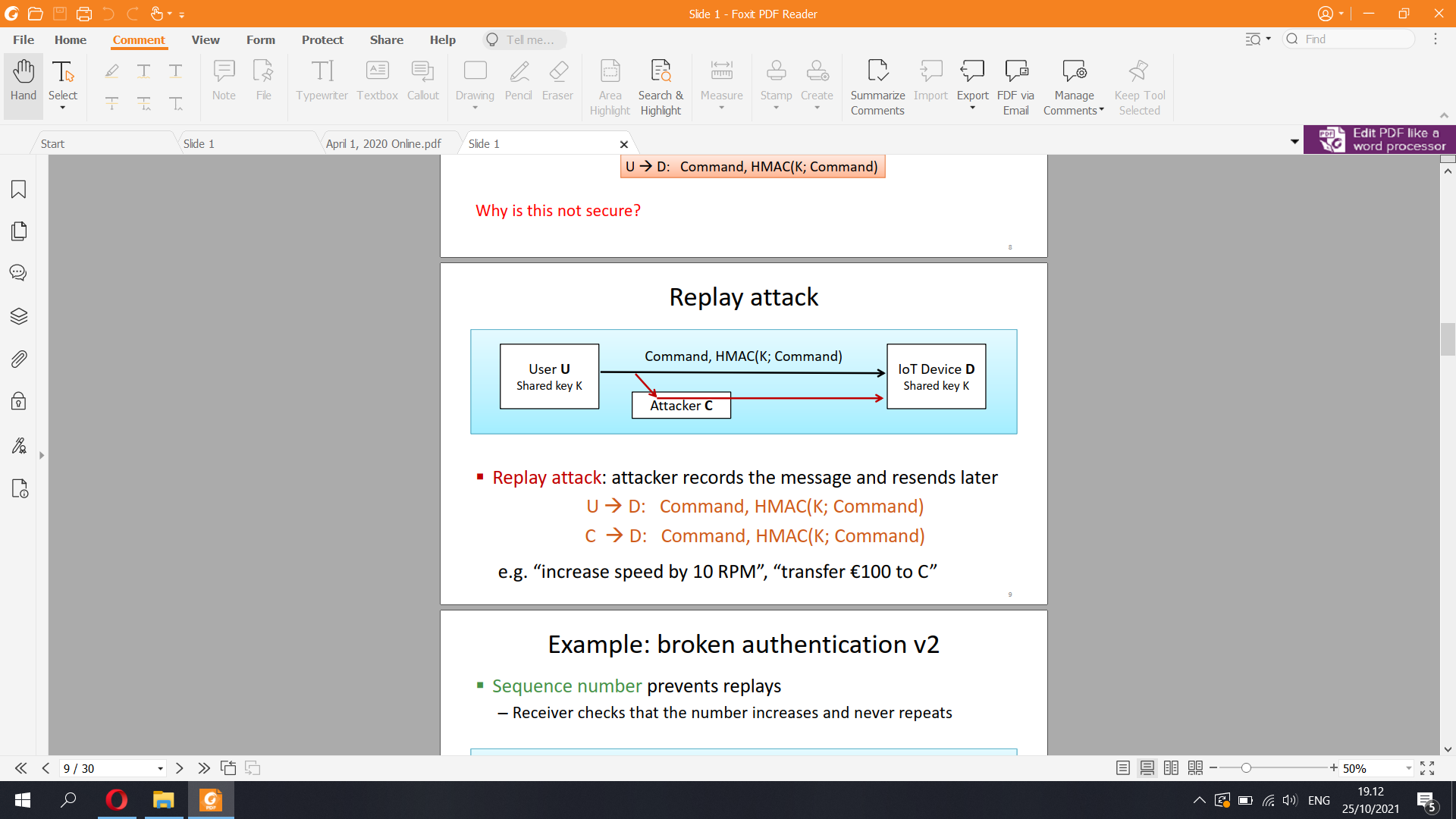
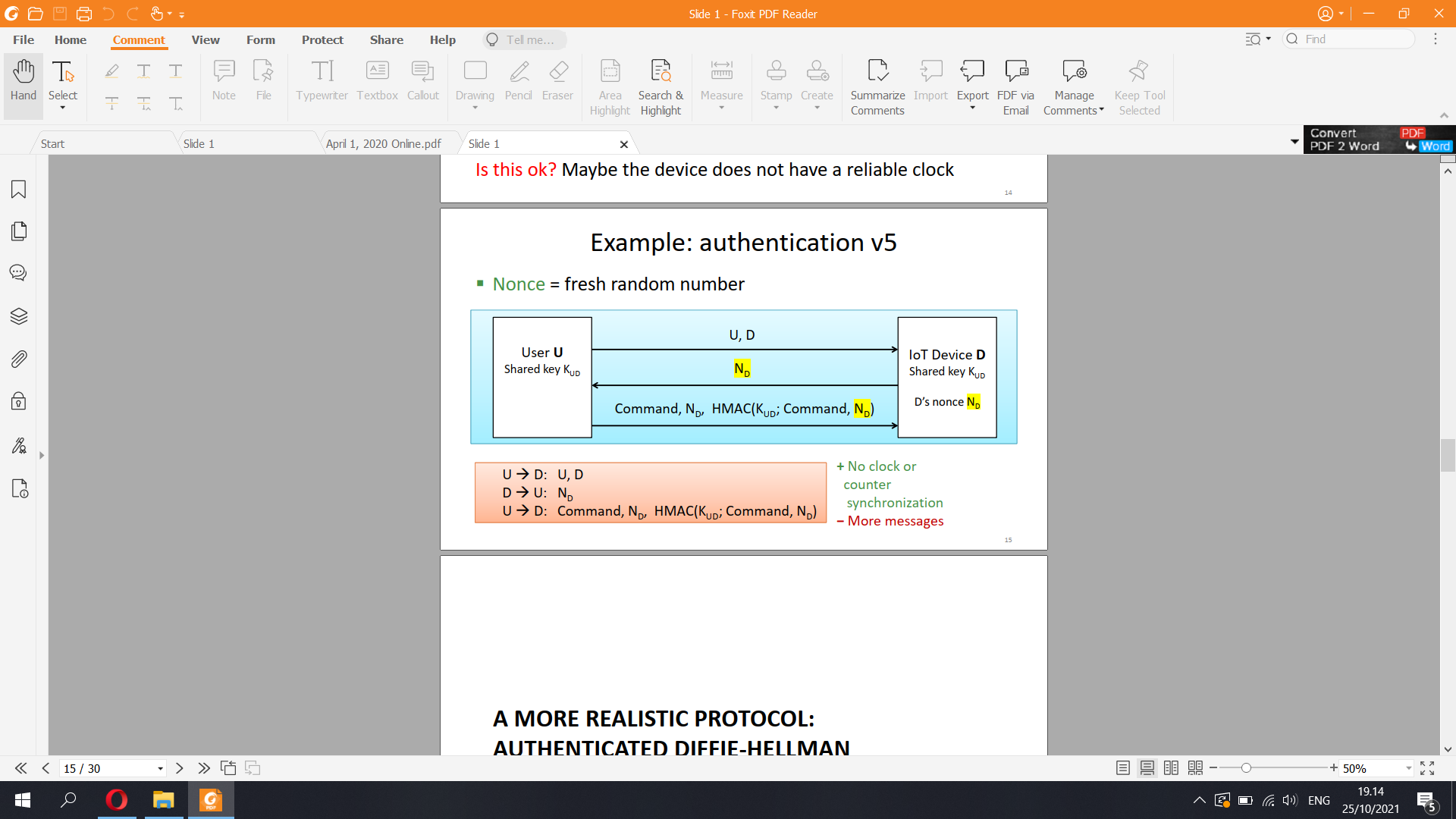


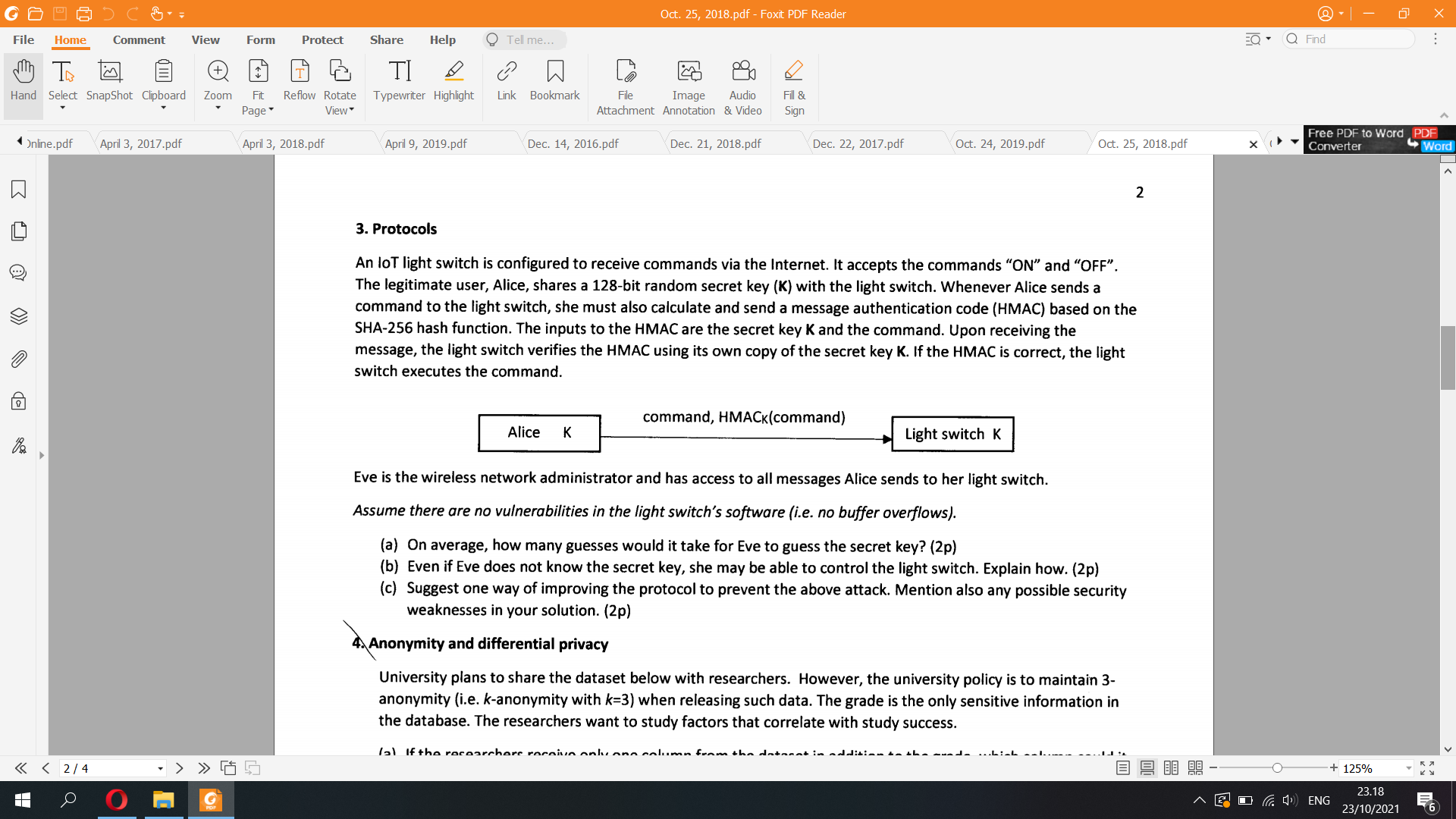
a) In protocol (a), there is absolutely no security at all. The attacker can easily read the command and temperature as plain text. As a consequence, the attacker can modify the content of the command over the insecure network, such as sending the wrong temperature to Alice’s phone or sending another command to the temperature sensor

b) In protocol (b), although there is HMAC that guards the temperature from being read, the action is not being encrypted to be verified by the temperature sensor, allowing the attacker to also send different commands to the sensor. Furthermore, the protocol is subject to replay attack, which the hacker can delay the response of the command or the reply of the HMAC, thus read command can receive later temperature instead of current temperature

c) In protocol (c), there is a fresh random number nonce that can help protect data transfer from being replayed or delayed, making it far more secure than protocol a and b. The only drawback is that encryption of longer concatenated strings of T and R increases the message size.

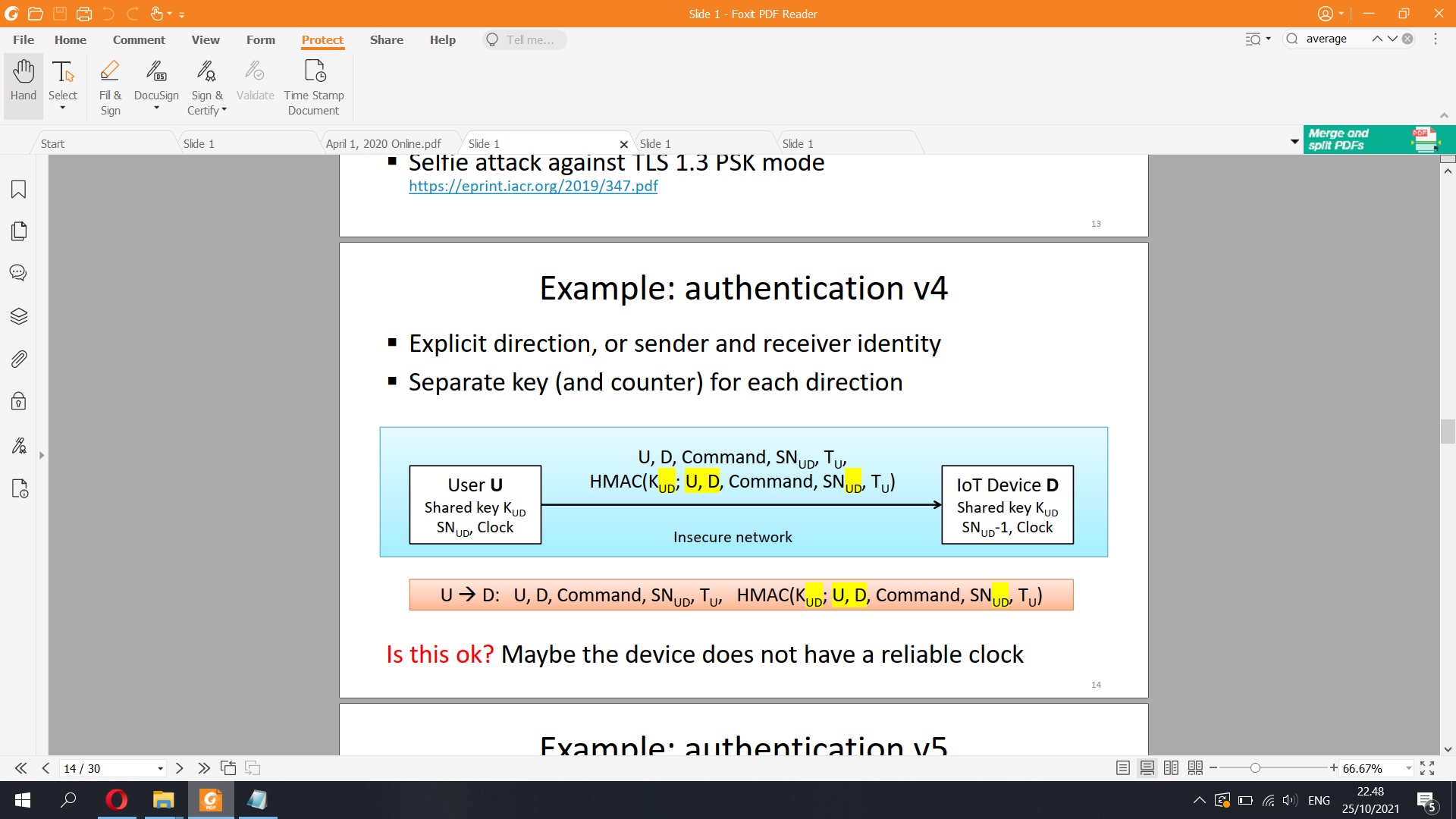


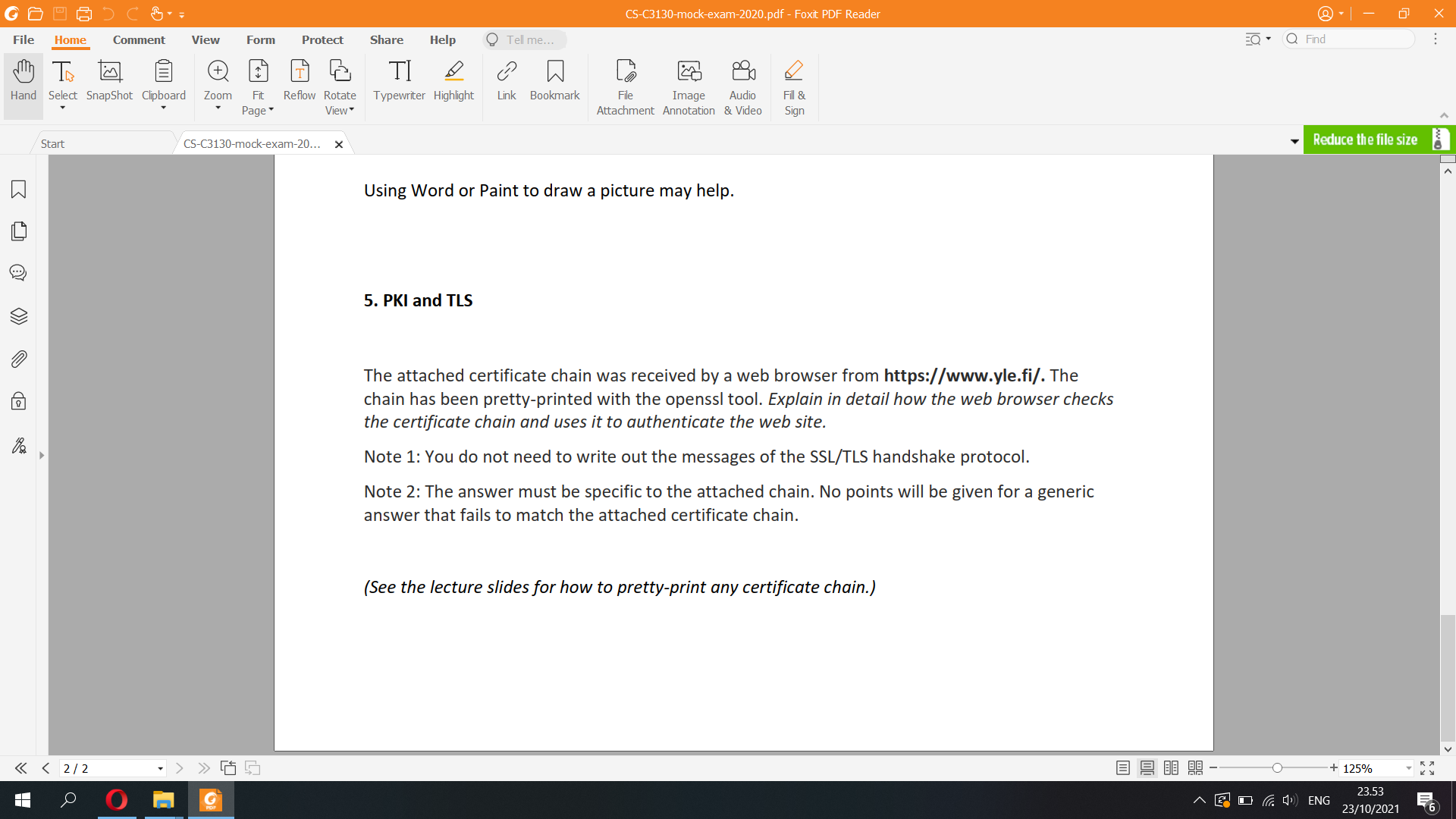


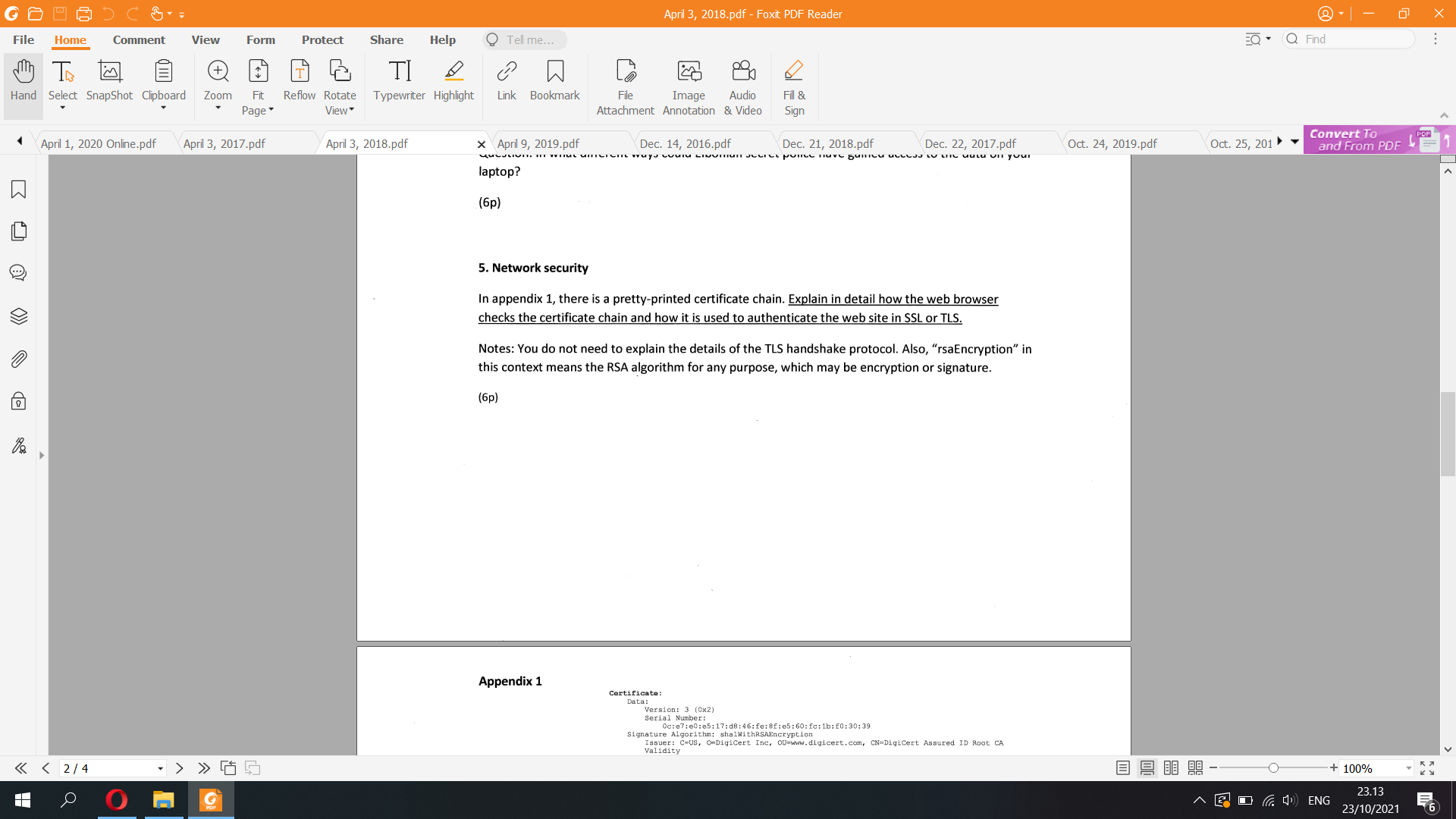


a) 128 bits mean there are 2^128 different keys to be chosen from. On average, brute-force cracking takes around 50% of all variations => Eve need on average 2^127 guesses to guess the secret key

b) Since the data sent over the insecure network doesn't have a nonce or time stamp, the data is subject to replay attack. Eve can simply save the cipher text of HMACk(command) for on and off HMAC cipher text as Alice had previously sent her code. Eve can replay them to the light switch and thus, Eve can control the light switch by replaying the data,

c) I can add a timestamp that prevents delaying of messages, which receiver does not accept messages older than a certain amount of time to prevent replaying. However, there are several flaws, such as Eve can just replay this message back to Alice, or just delay it forever, rendering the light switch unusable because there is no HMAC code sent to the light switch!   






Answer:

In this process of authentication, the browser receives the certificate chain from the server as a process of trust chain.

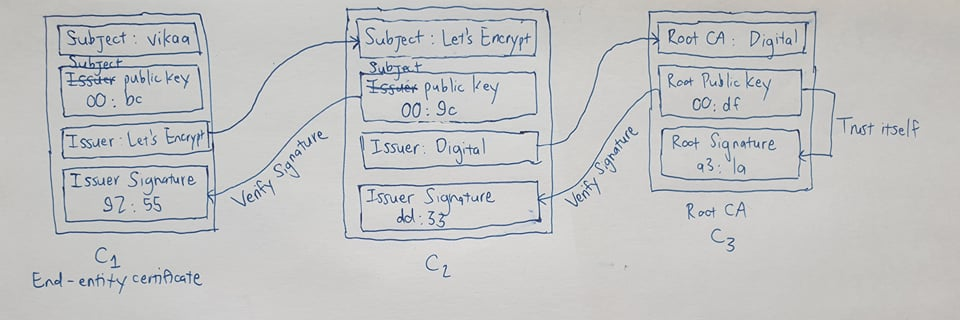
1. **The TLS is responsible for public key distribution** so that the browser can verify the signature on each certificate using the subject public key of the certificate above.After verification, **the last certificate (which contains the host \*.vikaa.fi) will bind together the host name \*.vikaa.fi and public key of the server**

Assume that the browser’s list of self-signed certificates for trusted root CAs contain the root Digital Signature Trust Co., the authentication works as follows

2. In the TLS handshake, the **browser** tells the server that it **recognizes the root CA** Digital Signature Trust Co.

3. The **browser receives a certificate chain from the server** as given in the exercise. I name the first certificate C1, second one as C2 and third one as C3

4. **Browser checks the validity of the certificate chain backwards from the end-entity-certificate C1 (\*.vikaa.fi) towards the root CA (Digital Signature Trust Co.)**:



A. The browser checks that there must be **exactly one end-entity certificate at the bottom** of the chain, which is vikaa.fi. The other certificates in the chain must be CA certificates, which are in turn Let’s Encrypt and Digital Signature Trust Co.

B. **Issuer of each certificate must match the subject of the CA certificate above it** (Issuer of vikaa is Let’s Encrypt matches C2’s subject and issuer of Let’s Encrypt is Digital Signature Trust Co. matches C3’s subject)

C. The browser **verifies the issuer signature of each certificate with the subject public key of the certificate above** from the issuer’s CA certificate (Let’s Encrypt public key verifies the issuer signature of vikaa and Digital Signature Trust Co. public key verifies the issuer signature of Let’s Encrypt)

D. **Browser checks for certificate revocation from the OCSP server or CRL** if any certificate specifies these.

E. Browser checks that the **certificates are in a CT log**.

F. There may be **constraints in the certificates**, which must also be checked such as name constraint

The browser **goes upwards until it finds a trusted CA**. At first it doesn't have vikaa.fi in its trusted CA list, so it goes upward one certificate from C3 to C2. The browser’s certificate list still doesn't have Let’s Encrypt, so it goes upwards from C2 to C3. Now, a trusted root CA is found which is Digital Signature Trust Co. The browser stops searching here.

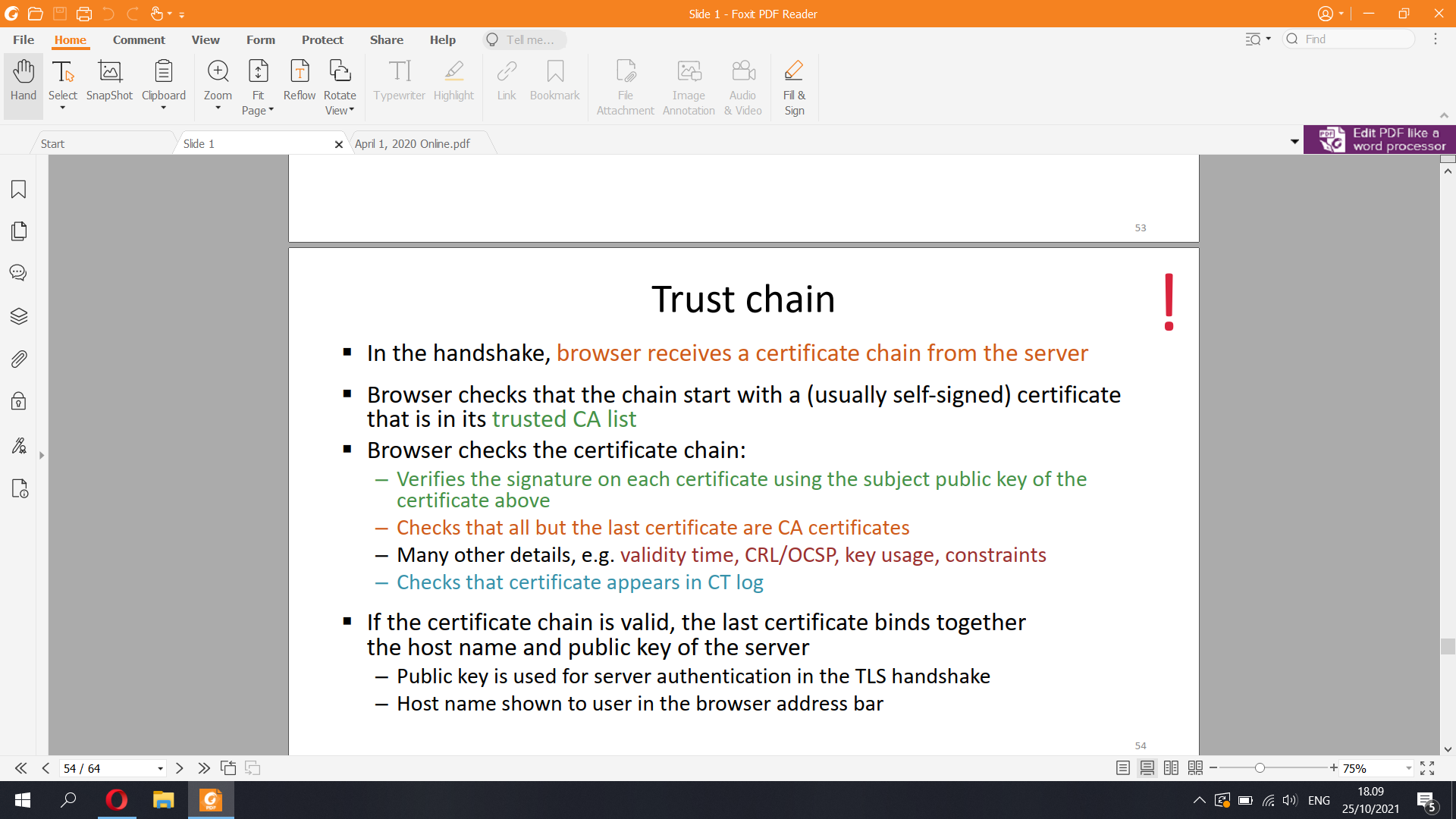
5. Browser **checks that the C3 certificate of Digital Signature has been issued for the right purpose**: the **end-entity certificate of vikaa.fi must specify TLS server authentication**.

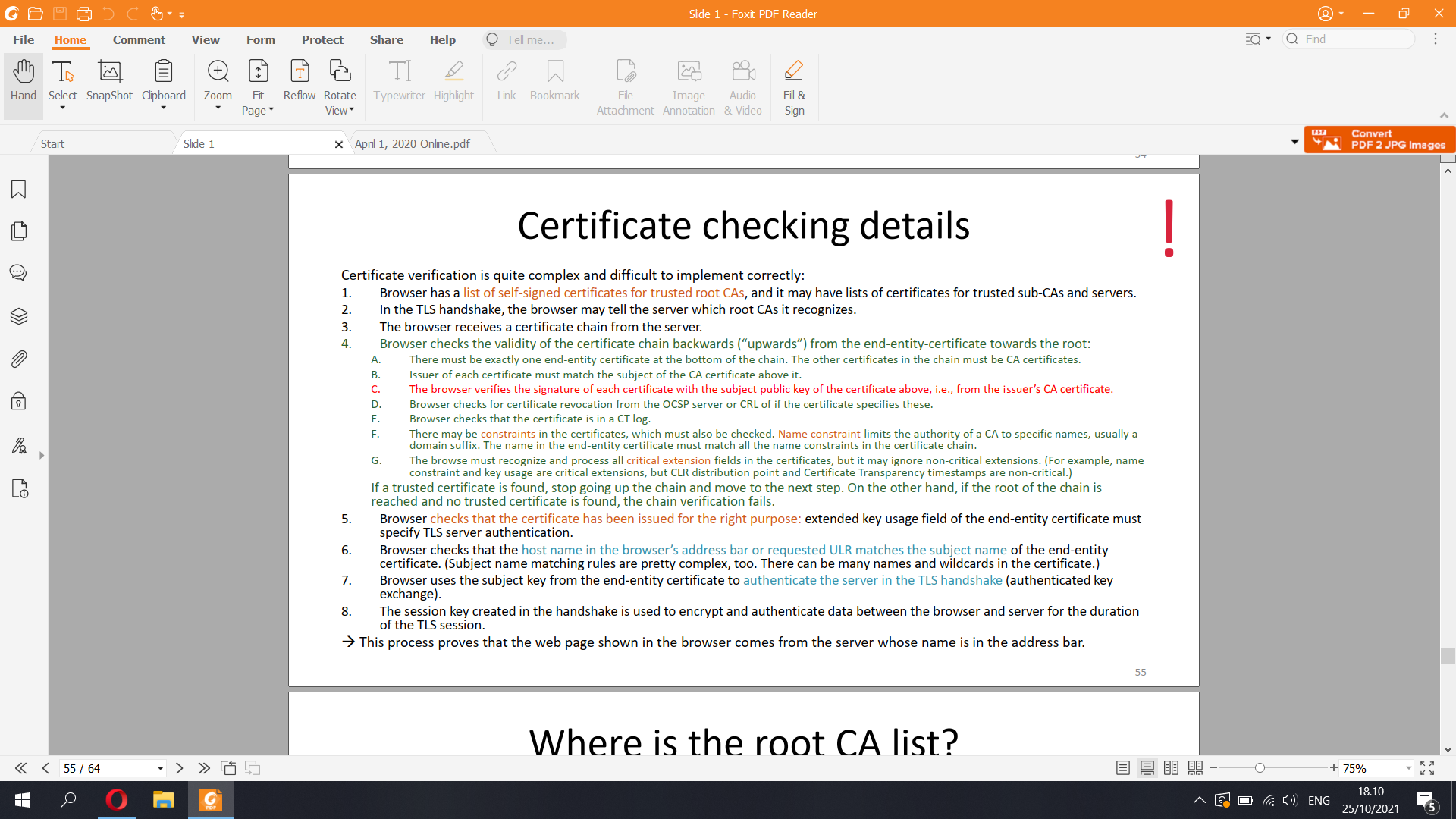
6. Browser checks that the **host name in the browser’s address bar or requested URL matches the subject name of the end-entity certificate** (<https://infosec1.vikaa.fi> host name in this URL matches the subject name of end-entity-certificate vikaa)

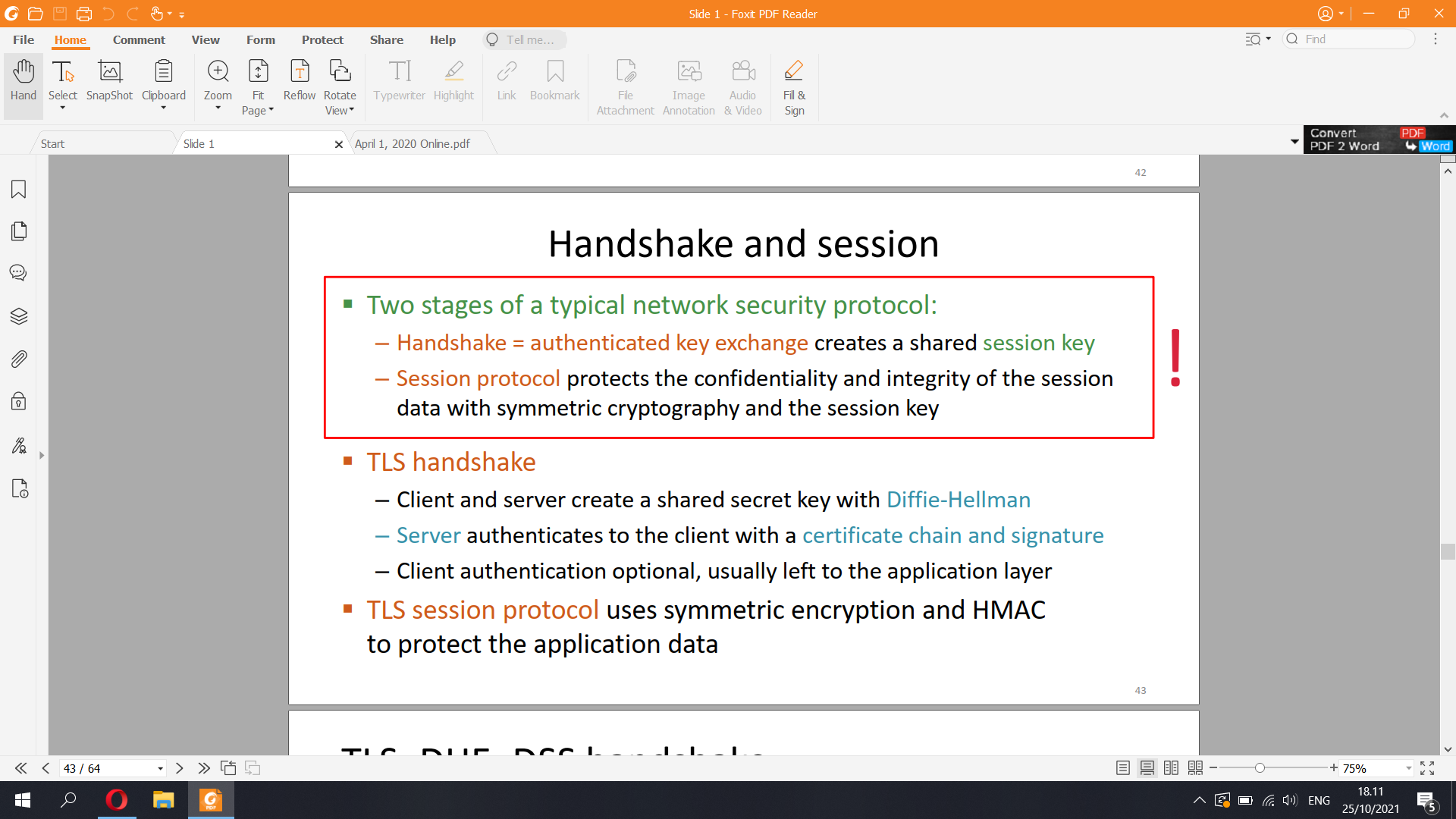
7. **Browser uses the** **subject key from the end-entity certificate to authenticate the server in the TLS handshake (authenticated key exchange).** (00:bc:.....)

8. The **session key created in the handshake is used to encrypt and authenticate data between the browser and server for the duration of the TLS session.** From C1 to C3

**→ This process proves that the web page vikaa.fi shown in the browser comes from the server whose name is in the address bar (vikaa.fi server)**







- X.509 certificate fields

Mandatory fields:

▪ Version

▪ Serial number — together with Issuer, uniquely identifiers the

certificate

▪ Signature algorithm — for the signature on this certificate; usually

sha1RSA; includes any parameters

▪ Issuer — name (e.g. CN = Microsoft Corp Enterprise CA 2)

▪ Valid from — usually the time when issued

▪ Valid to — expiry time

▪ Subject — distinguished name of the subject

▪ Public key — public key of the subject

Common extension fields:

▪ Key usage — bit field indicating usages for the subject key (digitalSignature,

nonRepudiation, keyEncipherment, dataEncipherment, keyAgreement, keyCertSign, cRLSign,

encipherOnly, decipherOnly)

▪ Subject alternative name — email address, DNS name, IP address, etc.

▪ Issuer alternative name

▪ Basic constraints — (1) is the subject a CA or an end entity, (2) maximum length of

delegation to sub-CAs after the subject

▪ Name constraints — limit the authority of the CA

▪ Certificate policies — list of OIDs to indicate policies for the certificate

▪ Policy constraints — certificate policies

▪ Extended key usage — list of OIDs for new usages, e.g. server authentication, client

authentication, code signing, email protection, EFS key, etc.

▪ CRL distribution point — where to get the CRL for this certificate, and who issues CRLs

▪ Authority info access — where to find information about the CA and its policies

An X.509 certificate consists of a number of fields. The Subject field is the one of most relevance to this tutorial. It gives the DName of the client to which the certificate belongs. A DName is a unique name given to an X.500 directory object. It consists of a number of attribute-value pairs called Relative Distinguished Names (RDNs). Some of the most common RDNs and their explanations are as follows:

CN: CommonName

OU: OrganizationalUnit

O: Organization

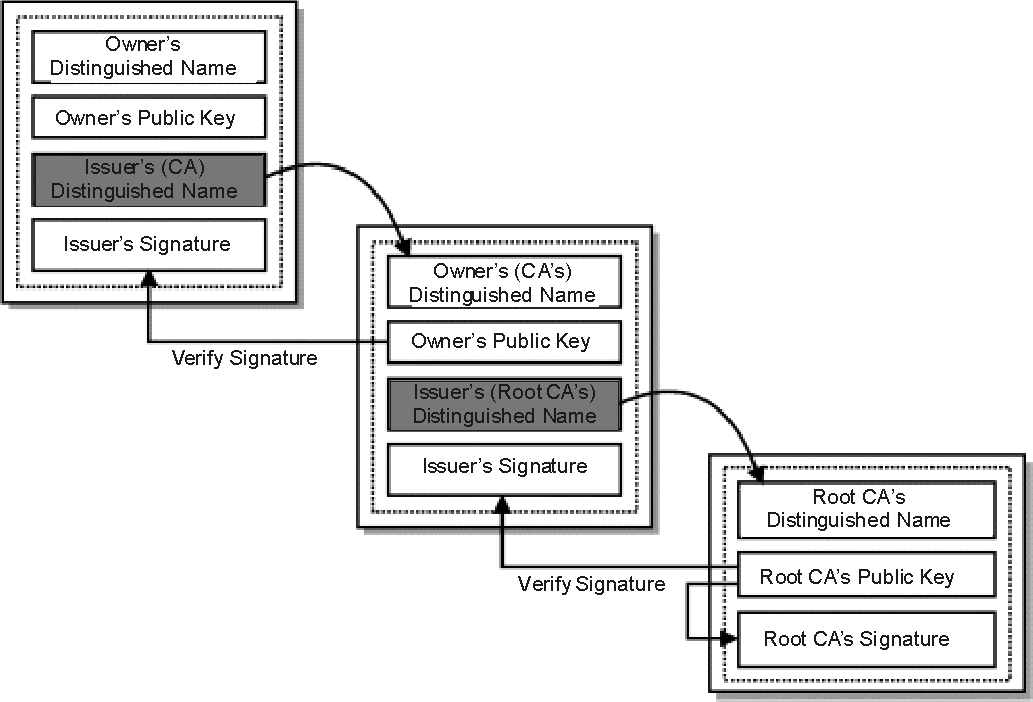
L: Locality

S: StateOrProvinceName

C: CountryName

The issuer is the CA that issued the cert (to the subject), and the root is the CA that is end point of all the trust in the hierarchy. The typical relationship is root--->issuer--->subject. The subject of the certificate is the entity its public key is associated with (i.e. the "owner" of the certificate).

<https://security.stackexchange.com/questions/56389/ssl-certificate-framework-101-how-does-the-browser-actually-verify-the-validity>



"public keys encrypt, private keys decrypt" is correct for data/message ENCRYPTION. For digital signatures, it is the reverse. With a digital signature, you are trying to prove that the document signed by you came from you. To do that, you need to use something that only YOU have: your private key.

A digital signature in its simplest description is a hash (SHA1, MD5, etc.) of the data (file, message, etc.) that is subsequently encrypted with the signer's private key. Since that is something only the signer has (or should have) that is where the trust comes from. EVERYONE has (or should have) access to the signer's public key.

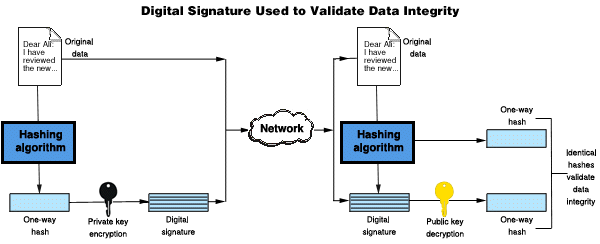
So, to validate a digital signature, the recipient

Calculates a hash of the same data (file, message, etc.),

Decrypts the digital signature using the sender's PUBLIC key, and

Compares the 2 hash values.

If they match, the signature is considered valid. If they don't match, it either means that a different key was used to sign it, or that the data has been altered (either intentionally or unintentionally).



//////

Certificate:

Data:

**Version**: 3 (0x2)

**Serial Number**:

03:3b:9e:d5:40:cd:f9:cb:2e:30:d7:42:5d:f1:48:9a:42:8b

**Signature Algorithm**: sha256WithRSAEncryption

**Issuer**: C = US, O = Let's Encrypt, CN = Let's Encrypt Authority X3

**Validity**

Not Before: Feb 25 14:30:29 2020 GMT

Not After : May 25 14:30:29 2020 GMT

**Subject**: CN = \*.vikaa.fi

**Subject Public Key Info**:

Public Key Algorithm: rsaEncryption

RSA Public-Key: (2048 bit)

Modulus:

00:bc:e9:a3:6b:c7:c7:b3:5b:59:28:e1:a0:02:6e:

81:f1:6b:d3:db:ff:7a:ba:a8:ef:a5:c1:26:7a:ea:

8f:27:af:d9:b0:e2:4a:55:7c:eb:1a:c5:18:1a:f2:

f2:b8:eb:09:ee:85:a2:21:61:d4:19:95:13:e9:6a:

2e:b7:b8:04:c8:b8:4e:90:4d:51:22:b2:25:7c:79:

92:4d:67:a5:ad:a1:68:ef:8c:89:a0:37:08:85:6f:

ed:ec:6f:a3:c7:42:1f:59:66:62:be:cc:fa:64:36:

51:68:fd:86:73:d3:d0:32:1b:6c:61:02:44:1d:ee:

7e:ea:5d:aa:8f:5e:3e:00:71:6b:55:42:62:67:aa:

f5:27:6d:70:26:fb:15:00:ad:ba:50:5c:b1:b8:30:

be:ab:92:71:5f:43:b8:3f:4c:88:a6:7b:69:16:a0:

6f:4f:e5:56:0a:c4:ab:cf:37:75:f7:ef:c8:30:65:

d9:78:b7:f8:3a:52:e4:7a:ee:6c:b2:f5:de:99:24:

d9:25:97:eb:c4:c9:84:3f:c7:8f:65:0c:60:5c:24:

f5:13:a4:27:89:31:49:0e:64:e1:9b:aa:bf:0b:21:

83:7c:34:cc:34:2a:18:6f:e2:0e:26:0c:61:e4:6d:

c3:4f:ea:43:85:52:2f:e7:28:4c:3f:e6:36:be:33:

3a:b5

Exponent: 65537 (0x10001)

X509v3 extensions:

X509v3 Key Usage: critical

Digital Signature, Key Encipherment

X509v3 Extended Key Usage:

TLS Web Server Authentication, TLS Web Client Authentication

X509v3 Basic Constraints: critical

CA:FALSE

X509v3 Subject Key Identifier:

97:FD:B1:4C:88:52:0D:68:1C:24:F4:6C:BE:80:FE:AA:B2:40:B9:EE

X509v3 Authority Key Identifier:

keyid:A8:4A:6A:63:04:7D:DD:BA:E6:D1:39:B7:A6:45:65:EF:F3:A8:EC:A1

Authority Information Access:

OCSP - URI:http://ocsp.int-x3.letsencrypt.org

CA Issuers - URI:http://cert.int-x3.letsencrypt.org/

X509v3 Subject Alternative Name:

DNS:\*.vikaa.fi

X509v3 Certificate Policies:

Policy: 2.23.140.1.2.1

Policy: 1.3.6.1.4.1.44947.1.1.1

CPS: http://cps.letsencrypt.org

CT Precertificate SCTs:

Signed Certificate Timestamp:

Version : v1 (0x0)

Log ID : E7:12:F2:B0:37:7E:1A:62:FB:8E:C9:0C:61:84:F1:EA:

7B:37:CB:56:1D:11:26:5B:F3:E0:F3:4B:F2:41:54:6E

Timestamp : Feb 25 15:30:29.248 2020 GMT

Extensions: none

Signature : ecdsa-with-SHA256

30:45:02:21:00:92:69:11:79:81:C3:45:71:23:2D:C0:

EE:B1:32:C9:E3:55:37:E9:BD:6F:8F:1D:A6:96:AC:F9:

A5:AB:3A:E0:D7:02:20:10:02:3B:B6:3F:89:BF:93:93:

09:26:0B:00:35:AC:E3:F0:E6:4F:39:75:C0:1B:C4:78:

46:E1:5E:67:72:04:85

Signed Certificate Timestamp:

Version : v1 (0x0)

Log ID : B2:1E:05:CC:8B:A2:CD:8A:20:4E:87:66:F9:2B:B9:8A:

25:20:67:6B:DA:FA:70:E7:B2:49:53:2D:EF:8B:90:5E

Timestamp : Feb 25 15:30:29.239 2020 GMT

Extensions: none

Signature : ecdsa-with-SHA256

30:46:02:21:00:9F:EB:33:9A:68:E9:59:85:11:38:70:

A7:FC:2E:10:D6:9F:7C:9C:E8:A7:D8:3A:E6:4A:A6:2C:

DA:48:59:F9:71:02:21:00:B2:44:E4:72:3B:87:B3:43:

0F:34:21:6F:11:85:5F:FE:EF:6F:2A:DD:29:0D:06:D1:

7B:CF:E5:5F:B4:3D:F4:6B

**Signature Algorithm**: sha256WithRSAEncryption

92:55:cc:04:44:ad:8d:44:b7:1b:56:08:a4:80:23:ea:15:3d:

71:51:c6:c6:37:64:cf:d8:31:ad:a6:8e:b2:52:5d:75:ad:f2:

21:4b:86:e3:63:a2:84:5d:8e:0e:70:31:ed:a4:4e:ed:51:03:

1b:9e:e0:60:aa:4a:7c:2e:04:72:94:14:fd:4b:ad:a4:c3:33:

5d:da:77:ba:33:cf:b8:65:48:bd:44:0f:e5:cc:e0:2e:9b:5e:

34:ef:15:a2:fc:ee:3f:10:fb:9d:7f:03:34:3f:6d:29:82:fe:

f8:7f:41:4b:8b:51:44:f3:bf:51:bc:30:96:a9:53:33:05:72:

5

86:e2:26:34:a5:4f:e7:b8:78:d7:0d:78:99:3b:64:16:da:90:

2c:67:9d:b1:de:9f:37:36:08:77:20:89:f9:54:fd:4b:e2:d9:

e0:ad:df:65:42:54:b5:ad:23:ff:ac:31:e4:49:bb:fd:ff:db:

11:cb:ae:13:37:e1:f7:97:5f:95:7c:67:48:80:3b:8e:a0:73:

ae:54:ed:56:00:b9:d3:44:0f:21:81:f8:bd:0a:81:4e:08:4b:

05:39:32:5c:76:12:ea:cc:05:3c:9b:ca:e7:44:fe:69:f1:7e:

62:55:2e:94:13:fc:c1:d4:00:c1:5b:20:3e:ad:77:68:90:8c:

C3:8c:68:46

Certificate:

Data:

Version: 3 (0x2)

Serial Number:

0a:01:41:42:00:00:01:53:85:73:6a:0b:85:ec:a7:08

Signature Algorithm: sha256WithRSAEncryption

**Issuer**: O = Digital Signature Trust Co., CN = DST Root CA X3

Validity

Not Before: Mar 17 16:40:46 2016 GMT

Not After : Mar 17 16:40:46 2021 GMT

Subject: C = US, O = Let's Encrypt, CN = Let's Encrypt Authority X3

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

RSA Public-Key: (2048 bit)

Modulus:

00:9c:d3:0c:f0:5a:e5:2e:47:b7:72:5d:37:83:b3:

68:63:30:ea:d7:35:26:19:25:e1:bd:be:35:f1:70:

92:2f:b7:b8:4b:41:05:ab:a9:9e:35:08:58:ec:b1:

2a:c4:68:87:0b:a3:e3:75:e4:e6:f3:a7:62:71:ba:

79:81:60:1f:d7:91:9a:9f:f3:d0:78:67:71:c8:69:

0e:95:91:cf:fe:e6:99:e9:60:3c:48:cc:7e:ca:4d:

77:12:24:9d:47:1b:5a:eb:b9:ec:1e:37:00:1c:9c:

ac:7b:a7:05:ea:ce:4a:eb:bd:41:e5:36:98:b9:cb:

fd:6d:3c:96:68:df:23:2a:42:90:0c:86:74:67:c8:

7f:a5:9a:b8:52:61:14:13:3f:65:e9:82:87:cb:db:

fa:0e:56:f6:86:89:f3:85:3f:97:86:af:b0:dc:1a:

ef:6b:0d:95:16:7d:c4:2b:a0:65:b2:99:04:36:75:

80:6b:ac:4a:f3:1b:90:49:78:2f:a2:96:4f:2a:20:

25:29:04:c6:74:c0:d0:31:cd:8f:31:38:95:16:ba:

a8:33:b8:43:f1:b1:1f:c3:30:7f:a2:79:31:13:3d:

2d:36:f8:e3:fc:f2:33:6a:b9:39:31:c5:af:c4:8d:

0d:1d:64:16:33:aa:fa:84:29:b6:d4:0b:c0:d8:7d:

c3:93

Exponent: 65537 (0x10001)

X509v3 extensions:

X509v3 Basic Constraints: critical

CA:TRUE, pathlen:0

X509v3 Key Usage: critical

Digital Signature, Certificate Sign, CRL Sign

Authority Information Access:

OCSP - URI:http://isrg.trustid.ocsp.identrust.com

CA Issuers - URI:http://apps.identrust.com/roots/dstrootcax3.p7c

X509v3 Authority Key Identifier:

keyid:C4:A7:B1:A4:7B:2C:71:FA:DB:E1:4B:90:75:FF:C4:15:60:85:89:10

X509v3 Certificate Policies:

Policy: 2.23.140.1.2.1

Policy: 1.3.6.1.4.1.44947.1.1.1

CPS: http://cps.root-x1.letsencrypt.org

X509v3 CRL Distribution Points:

Full Name:

URI:http://crl.identrust.com/DSTROOTCAX3CRL.crl

X509v3 Subject Key Identifier:

A8:4A:6A:63:04:7D:DD:BA:E6:D1:39:B7:A6:45:65:EF:F3:A8:EC:A1

Signature Algorithm: sha256WithRSAEncryption

dd:33:d7:11:f3:63:58:38:dd:18:15:fb:09:55:be:76:56:b9:

70:48:a5:69:47:27:7b:c2:24:08:92:f1:5a:1f:4a:12:29:37:

24:74:51:1c:62:68:b8:cd:95:70:67:e5:f7:a4:bc:4e:28:51:

cd:9b:e8:ae:87:9d:ea:d8:ba:5a:a1:01:9a:dc:f0:dd:6a:1d:

6a:d8:3e:57:23:9e:a6:1e:04:62:9a:ff:d7:05:ca:b7:1f:3f:

c0:0a:48:bc:94:b0:b6:65:62:e0:c1:54:e5:a3:2a:ad:20:c4:

e9:e6:bb:dc:c8:f6:b5:c3:32:a3:98:cc:77:a8:e6:79:65:07:

2b:cb:28:fe:3a:16:52:81:ce:52:0c:2e:5f:83:e8:d5:06:33:

fb:77:6c:ce:40:ea:32:9e:1f:92:5c:41:c1:74:6c:5b:5d:0a:

5f:33:cc:4d:9f:ac:38:f0:2f:7b:2c:62:9d:d9:a3:91:6f:25:

1b:2f:90:b1:19:46:3d:f6:7e:1b:a6:7a:87:b9:a3:7a:6d:18:

fa:25:a5:91:87:15:e0:f2:16:2f:58:b0:06:2f:2c:68:26:c6:

4b:98:cd:da:9f:0c:f9:7f:90:ed:43:4a:12:44:4e:6f:73:7a:

28:ea:a4:aa:6e:7b:4c:7d:87:dd:e0:c9:02:44:a7:87:af:c3:

34:5b:b4:42

Certificate:

Data:

Version: 3 (0x2)

Serial Number:

44:af:b0:80:d6:a3:27:ba:89:30:39:86:2e:f8:40:6b

Signature Algorithm: sha1WithRSAEncryption

**Issuer**: O = Digital Signature Trust Co., CN = DST Root CA X3

Validity

Not Before: Sep 30 21:12:19 2000 GMT

Not After : Sep 30 14:01:15 2021 GMT

Subject: O = Digital Signature Trust Co., CN = DST Root CA X3

6

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

RSA Public-Key: (2048 bit)

Modulus:

00:df:af:e9:97:50:08:83:57:b4:cc:62:65:f6:90:

82:ec:c7:d3:2c:6b:30:ca:5b:ec:d9:c3:7d:c7:40:

c1:18:14:8b:e0:e8:33:76:49:2a:e3:3f:21:49:93:

ac:4e:0e:af:3e:48:cb:65:ee:fc:d3:21:0f:65:d2:

2a:d9:32:8f:8c:e5:f7:77:b0:12:7b:b5:95:c0:89:

a3:a9:ba:ed:73:2e:7a:0c:06:32:83:a2:7e:8a:14:

30:cd:11:a0:e1:2a:38:b9:79:0a:31:fd:50:bd:80:

65:df:b7:51:63:83:c8:e2:88:61:ea:4b:61:81:ec:

52:6b:b9:a2:e2:4b:1a:28:9f:48:a3:9e:0c:da:09:

8e:3e:17:2e:1e:dd:20:df:5b:c6:2a:8a:ab:2e:bd:

70:ad:c5:0b:1a:25:90:74:72:c5:7b:6a:ab:34:d6:

30:89:ff:e5:68:13:7b:54:0b:c8:d6:ae:ec:5a:9c:

92:1e:3d:64:b3:8c:c6:df:bf:c9:41:70:ec:16:72:

d5:26:ec:38:55:39:43:d0:fc:fd:18:5c:40:f1:97:

eb:d5:9a:9b:8d:1d:ba:da:25:b9:c6:d8:df:c1:15:

02:3a:ab:da:6e:f1:3e:2e:f5:5c:08:9c:3c:d6:83:

69:e4:10:9b:19:2a:b6:29:57:e3:e5:3d:9b:9f:f0:

02:5d

Exponent: 65537 (0x10001)

X509v3 extensions:

X509v3 Basic Constraints: critical

CA:TRUE

X509v3 Key Usage: critical

Certificate Sign, CRL Sign

X509v3 Subject Key Identifier:

C4:A7:B1:A4:7B:2C:71:FA:DB:E1:4B:90:75:FF:C4:15:60:85:89:10

Signature Algorithm: sha1WithRSAEncryption

a3:1a:2c:9b:17:00:5c:a9:1e:ee:28:66:37:3a:bf:83:c7:3f:

4b:c3:09:a0:95:20:5d:e3:d9:59:44:d2:3e:0d:3e:bd:8a:4b:

a0:74:1f:ce:10:82:9c:74:1a:1d:7e:98:1a:dd:cb:13:4b:b3:

20:44:e4:91:e9:cc:fc:7d:a5:db:6a:e5:fe:e6:fd:e0:4e:dd:

b7:00:3a:b5:70:49:af:f2:e5:eb:02:f1:d1:02:8b:19:cb:94:

3a:5e:48:c4:18:1e:58:19:5f:1e:02:5a:f0:0c:f1:b1:ad:a9:

dc:59:86:8b:6e:e9:91:f5:86:ca:fa:b9:66:33:aa:59:5b:ce:

e2:a7:16:73:47:cb:2b:cc:99:b0:37:48:cf:e3:56:4b:f5:cf:

0f:0c:72:32:87:c6:f0:44:bb:53:72:6d:43:f5:26:48:9a:52:

67:b7:58:ab:fe:67:76:71:78:db:0d:a2:56:14:13:39:24:31:

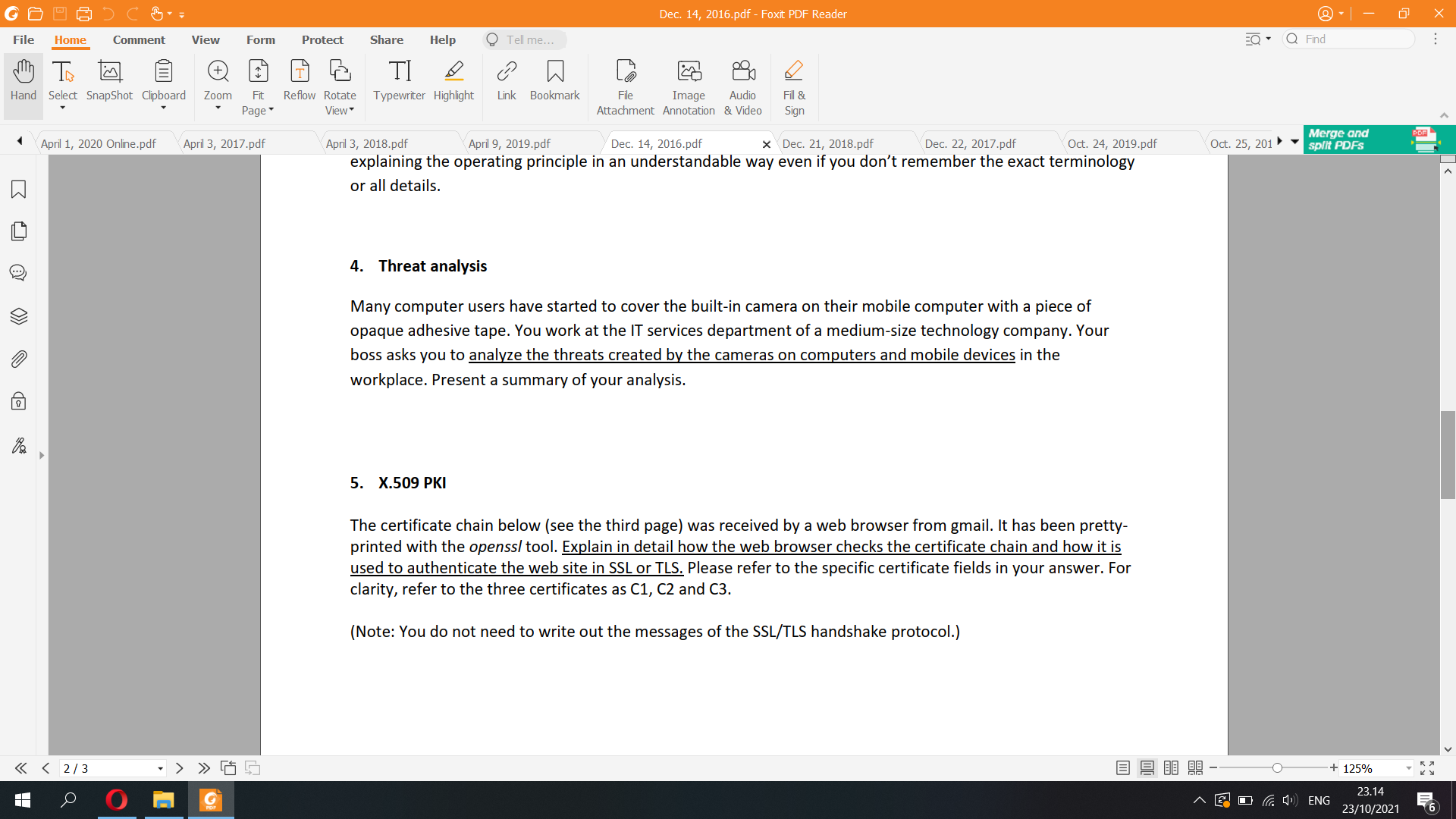
85:a2:a8:02:5a:30:47:e1:dd:50:07:bc:02:09:90:00:eb:64:

63:60:9b:16:bc:88:c9:12:e6:d2:7d:91:8b:f9:3d:32:8d:65:

b4:e9:7c:b1:57:76:ea:c5:b6:28:39:bf:15:65:1c:c8:f6:77:

96:6a:0a:8d:77:0b:d8:91:0b:04:8e:07:db:29:b6:0a:ee:9d:

82:35:35:10



Certificate C1:

Data:

Version: 3 (0x2)

Serial Number: 5034357460863282341

(0x45dda16fff17eca5)

Signature Algorithm: sha256WithRSAEncryption

Issuer: C=US, O=Google Inc, CN=Google

Internet Authority G2

Validity

Not Before: Oct 7 11:10:51 2015 GMT

Not After : Jan 5 00:00:00 2016 GMT

Subject: C=US, ST=California, L=Mountain

View, O=Google Inc, CN=mail.google.com

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

Public-Key: (2048 bit)

Modulus:

00:96:db:37:d0:56:cf:f9:1d:76:74:eb:f3:b1:ed:

…many more bytes…

01:db

Exponent: 65537 (0x10001)

X509v3 extensions:

X509v3 Extended Key Usage:

TLS Web Server Authentication, TLS

Web Client Authentication

X509v3 Subject Alternative Name:

DNS:mail.google.com,

DNS:inbox.google.com

Authority Information Access:

CA Issuers -

URI:http://pki.google.com/GIAG2.crt

OCSP -

URI:http://clients1.google.com/ocsp

X509v3 Subject Key Identifier:

37:DB:18:BA:07:20:3C:DA:A6:B1:9F:C2:5C:4C:6C:85:7C:B2

:6B:E0

X509v3 Basic Constraints: critical

CA:FALSE

X509v3 Authority Key Identifier:

keyid:4A:DD:06:16:1B:BC:F6:68:B5:76:F5:81:B6:BB:62:1A

:BA:5A:81:2F

X509v3 Certificate Policies:

Policy: 1.3.6.1.4.1.11129.2.5.1

Policy: 2.23.140.1.2.2

X509v3 CRL Distribution Points:

Full Name:

URI:http://pki.google.com/GIAG2.crl

Signature Algorithm: sha256WithRSAEncryption

64:be:a0:00:54:57:c3:32:0f:c0:3e:63:19:e4:b4:96:56:8b

:

ea:66:98:96:38:47:f5:85:cd:cf:da:25:19:a7:ba:5b:

…many more bytes…

8c:e8:ad:b9:21:67:ed:85:45:8a:a1:94:5d:04

Certificate C2:

Data:

Version: 3 (0x2)

Serial Number: 146051 (0x23a83)

Signature Algorithm: sha256WithRSAEncryption

Issuer: C=US, O=GeoTrust Inc., CN=GeoTrust

Global CA

Validity

Not Before: Apr 5 15:15:56 2013 GMT

Not After : Dec 31 23:59:59 2016 GMT

Subject: C=US, O=Google Inc, CN=Google

Internet Authority G2

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

Public-Key: (2048 bit)

Modulus:

00:9c:2a:04:77:5c:d8:50:91:3a:06:a3:82:e0:d8:

…many more bytes…

72:69

Exponent: 65537 (0x10001)

X509v3 extensions:

X509v3 Authority Key Identifier:

keyid:C0:7A:98:68:8D:89:FB:AB:05:64:0C:11:7D:AA:7D:65

:B8:CA:CC:4E

X509v3 Subject Key Identifier:

4A:DD:06:16:1B:BC:F6:68:B5:76:F5:81:B6:BB:62:1A:BA:5A

:81:2F

X509v3 Key Usage: critical

Certificate Sign, CRL Sign

Authority Information Access:

OCSP - URI:http://g.symcd.com

X509v3 Basic Constraints: critical

CA:TRUE, pathlen:0

X509v3 CRL Distribution Points:

Full Name:

URI:http://g.symcb.com/crls/gtglobal.crl

X509v3 Certificate Policies:

Policy: 1.3.6.1.4.1.11129.2.5.1

Signature Algorithm: sha256WithRSAEncryption

aa:fa:a9:20:cd:6a:67:83:ed:5e:d4:7e:de:1d:c4:7f:

…many more bytes…

7e:c8:35:d8

Certificate C3:

Data:

Version: 3 (0x2)

Serial Number: 1227750 (0x12bbe6)

Signature Algorithm: sha1WithRSAEncryption

Issuer: C=US, O=Equifax, OU=Equifax Secure

Certificate Authority

Validity

Not Before: May 21 04:00:00 2002 GMT

Not After : Aug 21 04:00:00 2018 GMT

Subject: C=US, O=GeoTrust Inc., CN=GeoTrust

Global CA

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

Public-Key: (2048 bit)

Modulus:

00:da:cc:18:63:30:fd:f4:17:23:1a:56:7e:5b:df:

…many more bytes…

e4:f9

Exponent: 65537 (0x10001)

X509v3 extensions:

X509v3 Authority Key Identifier:

keyid:48:E6:68:F9:2B:D2:B2:95:D7:47:D8:23:20:10:4F:33

:98:90:9F:D4

X509v3 Subject Key Identifier:

C0:7A:98:68:8D:89:FB:AB:05:64:0C:11:7D:AA:7D:65:B8:CA

:CC:4E

X509v3 Basic Constraints: critical

CA:TRUE

X509v3 Key Usage: critical

Certificate Sign, CRL Sign

X509v3 CRL Distribution Points:

Full Name:

URI:http://crl.geotrust.com/crls/secureca.crl

X509v3 Certificate Policies:

Policy: X509v3 Any Policy

CPS:

https://www.geotrust.com/resources/repository

Signature Algorithm: sha1WithRSAEncryption

76:e1:12:6e:4e:4b:16:12:86:30:06:b2:81:08:cf:f0:

…many more bytes…

3f:12