Tutorial Week 11

Questions

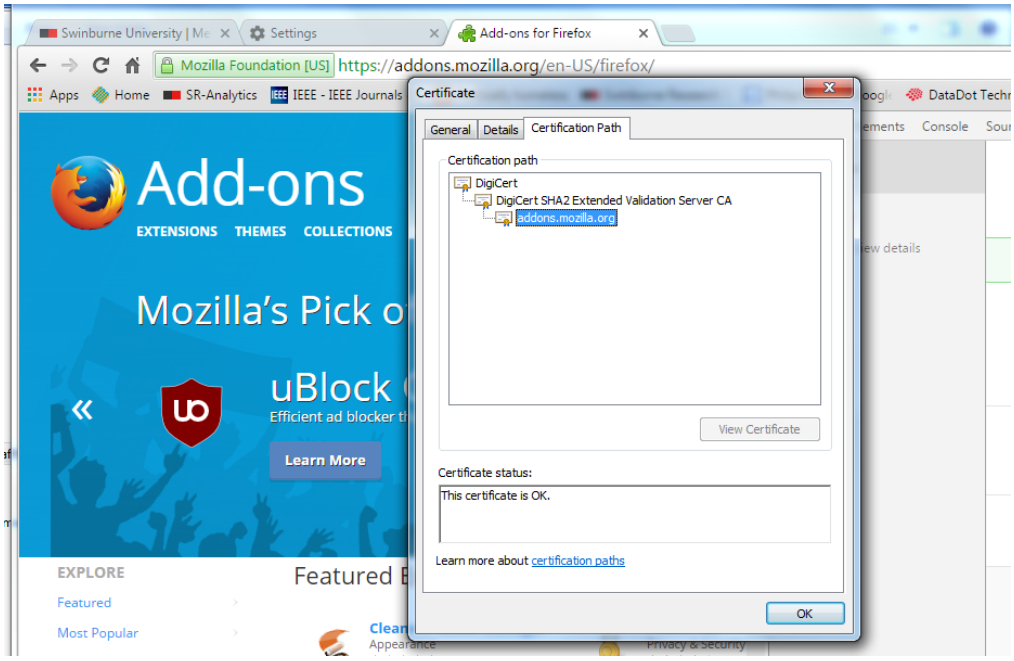
1. When purchasing goods via a website, why does the SSL/TLS protocol require the website to provide a digital certificate but not the person doing the purchasing?

   There are two reasons:

   a. SSL uses public key cryptography to exchange data for the session key, so one party must supply a public key and the other party must generate the session key. SSL insists the server provide the public key via a digital certificate.

   b. The second reason is that the risk associated with the transaction is on the part of the purchaser. The owner of the website can validate the purchase information (usually a credit card) without using a digital certificate. However, the purchaser has no guarantee that the website is not a spoofed site. A digital certificate provides some guarantee as to the identity of the website.

2. The following is a digital certificate for addons.mozilla.org. The certificate has been verified by clicking the lock icon. What steps will have taken place in verifying the certificate?



   1. Dates and certificate purpose will be checked to ensure the certificate is in date and that it is authorised to be used as a web server certificate.
   2. The signature of addons.mozilla.org will be verified. This is a multi-step process:
      a. The server certificate is signed by DigiCert SHA2. The certificate for this CA needs to be retrieved either from the cache of intermediate certificates stored in the browser or retrieved from the CA itself. The certificate contains the CA's public key which is used to verify the signature of the addons.mozilla.org server certificate
      b. Because the DigiCert SHA2 certificate is not a root certificate, it must be verified as well. Dates and purpose are checked and the signature is also checked. For its signature to be checked the certificate of the CA which signed its certificate is needed.

c. The DigiCert SHA2 certificate is signed by a root certificate (DigiCert) stored in the browser. Its public key is used to verify the signature of the intermediate certificate.

d. Optionally all certificates might be checked to see if they have been revoked by using the OCSP protocol.

3. At the successful completion of all these steps the addons.mozilla.org certificate is verified

3. Alice wishes to send a message to Bob. In order to guarantee the validity of identity, who will require a digital certificate in the following situations?

a) The message is to be encrypted.

Bob's public will be used for this, so Bob will need a certificate.

b) The message is to be signed.

Allice's private key will be used for this and her public key will be used to verify the signature. Consequently Alice will require a certificate

4. Test whether the following numbers are prime to a confidence level of 0.75.

9, 11,

9

Choose $a = 3$

Then $a^{((p-1)/2)} \bmod p = 3^{((9-1)/2)} \bmod 9 = 81 \bmod 9 = 0$, so 9 is not prime.

11

Choose $a = 2$

Then $a^{((p-1)/2)} \bmod p = 2^{((11-1)/2)} \bmod 11 = 2^5 \bmod 11 = 32 \bmod 11 = 10$ (= *p-1)*

So probability 11 is prime is more than 0.5

Choose $a = 3$

Then $a^{((p-1)/2)} \bmod p = 3^{((11-1)/2)} \bmod 11 = 3^5 \bmod 11 = 243 \bmod 11 = 1$

So probability 11 is prime is more than 0.75

5. Test whether 5 is prime to a confidence level of 0.9375

Choose $a = 2$

Then $a^{((p-1)/2)} \bmod p = 2^{((5-1)/2)} \bmod 5 = 2^2 \bmod 5 = 4 \bmod 5 = 4$ (= *p-1)*

So probability 5 is prime is more than 0.5

Choose $a = 3$

Then $a^{((p-1)/2)} \bmod p = 3^{((5-1)/2)} \bmod 5 = 3^2 \bmod 5 = 9 \bmod 5 = 4$ *(= p – 1)*

So probability 5 is prime is more than 0.75

Choose $a = 4$

Then $a^{((p-1)/2)} \bmod p = 4^{((5-1)/2)} \bmod 5 = 4^2 \bmod 5 = 16 \bmod 5 = 1$

So probability 5 is prime is more than 0.875

6. Consider the following simplified block encryption scheme:

   Plaintext is encrypted a byte at a time using the following steps:

   Step 1. The plain text is expanded to 12 bits by duplicating the first and last two bits (ie, abcdefgh becomes aabbcdefgghh

   Step 2. A 12 bit sub key is XORed with the expanded text from step 1

   Step 3. The bit sequence from step 2 is split into two 6 bit sequences and fed into the following two S-BOXes

   Step 4. The output of the S-BOXes is concatenated and fed through a permutation process that reverses the bit sequence order

   What is the output for a plaintext input of 1001 0100 and a 12 bit sub-key of 1001 0011 1010?

S1		**Middle four bits**															
		0000	**0001**	**0010**	**0011**	**0100**	**0101**	**0110**	**0111**	**1000**	**1001**	**1010**	**1011**	**1100**	**1101**	**1110**	**1111**
Outer bits	**00**	00	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110
	01	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000
	10	10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000
	11	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101

S2		**Middle four bits**															
		0000	**0001**	**0010**	**0011**	**0100**	**0101**	**0110**	**0111**	**1000**	**1001**	**1010**	**1011**	**1100**	**1101**	**1110**	**1111**
Outer bits	**00**	00	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110
	01	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000
	10	10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000
	11	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101

   Step 1. The plain text is expanded from 8 to 12 bits by duplicating the first and last two bits (ie, abcdefgh becomes aabbcdefgghh

   1001 0100 becomes 1100 0101 0000

   Step 2. A 12 bit sub key is XORed with the expanded text from step 1

   1100 0101 0000 XORed with sub-key 1001 0011 1010 becomes 0101 0110 1010

   Step 3. The bit sequence from step 2 is split into two 6 bit sequences and fed into the two S-BOXes

   First 6 bit sequence is 010101 which is mapped to 0000

   Second 6 bit sequence is 101010 which is mapped to 1010

   Step 4. The output of the S-BOXes is concatenated and fed through a permutation process that reverses the bit sequence order

   Output of S-BOXes is 0000 1010 which after permutation becomes 0101 0000

Tutorial Week 11