The University of Queensland
School of Information Technology and Electrical Engineering

Semester 2, 2017

COMS3000/7003 – Tutorial 11, Answers

Q1) Convert the decimal numbers 10, 33, and 63 to both binary and hexadecimal representations (provide all six answers).

Answer:

10 1010 A
33 100001 21
63 111111 3F

Q2) What is the result of bit-wise AND between the binary representations of the decimal numbers 173 and 255? This is trivial if you understand binary, but write the working down if you need.

Answer: 173

173 10101101
255 11111111
AND 10101101 → 173

Q3) What is the DECIMAL result of bit-wise AND between the binary representations of the hexadecimal numbers 1A and FF? Again, write the working down if you need.

Answer: 26 ($1A_{16}$)

1A 0001 1010
FF 1111 1111
AND 0001 1010 = "0x1A" (hexadecimal 1A, i.e. base 16: $1A_{16}$) = 1*16 + 10 = 26

Q4) What is the result a netmask (bit-wise AND) of 255.255.255.0 on an IPv4 address of 203.49.27.105.

Answer: 203.49.27.0 (i.e. the network address of the 203.49.27.0/24 subnet)

Q5) In last week's tutorial (Q10) we said even with our 4-bit inputs, we must use 6-bit outputs because the possible outputs are modulo 33, hence 0-32, and 32 requires 6 bits to represent it. However with only 4-bit plaintext inputs, there will only ever be 16 possible ciphertext outputs provided out of the 64 possible 6-bit numbers.

a) What are the 16 possible ciphertexts?

Answer:

m = 0000 → c = 000000 (0)
m = 0001 → c = 000001 (1)
m = 0010 → c = 011101 (29)
m = 0011 → c = 001001 (9)
m = 0100 → c = 010000 (16)
m = 0101 → c = 011101 (14)
m = 0110 → c = 011101 (30)
m = 0111 → c = 011101 (28)
m = 1000 → c = 011101 (2)
m = 1001 → c = 011101 (15)
m = 1010 → c = 011101 (10)
m = 1011 → c = 011101 (11)
m = 1100 → c = 011101 (12)
m = 1101 → c = 011101 (7)
m = 1110 → c = 011101 (20)
m = 1111 → c = 011101 (27)

b) Do we really need to provide for 6-bit ciphertext outputs or will 5-bit outputs suffice in this particular case?

Answer: 5-bits is sufficient in this case (only!) if we are only ever encrypting 4-bit nibbles.

Q6) Explain how RSA can be used for digital signatures.

Answer:
A digital signature is computed by 'encrypting' a message with the private key. To verify the signature, the message is 'decrypted' with the public key.

Everybody can verify the signature, since everybody has the public key. Only the owner of the private key can sign the message.

In RSA, encryption and decryption are completely interchangeable and symmetric operations. That's why RSA can be used for encryption as well as digital signatures. This is not the case for all public key systems.

Q7) What is the main purpose of a public key certificate?

Answer:
A public key certificate links an identity to its public key, i.e. it provides authenticity for public keys.

Q8) What is the actual value of the "Version" field inside an X.509 version 2 certificate?

Answer: 1

(or 00000001, it is a ASN.1 1-byte integer with value 1)

The original (so called "version 1") has 0; version 2 has the value 1; and version 3 has the value 2.

Q9) Research the "TLS Handshake Protocol" (for TLS version 1.2 from an authoritative source) and explain:

a) What is the data contained within a ChangeCipherSpec message?

Answer:
From Dierks & Rescorla, The Transport Layer Security (TLS) Protocol Version 1.2, (RFC 5246) 2008:
The message consists of a single byte of value 1.

b) What is the data contained with the client's Finished message?

Answer:
From Dierks & Rescorla, The Transport Layer Security (TLS) Protocol Version 1.2, (RFC 5246) 2008:
A SHA256 HMAC of master_secret and "client finished" concatenated with the hash of all the handshake messages so far.

c) What is the data contained with the server's Finished message?

Answer:
From Dierks & Rescorla, The Transport Layer Security (TLS) Protocol Version 1.2, (RFC 5246) 2008:
A SHA256 HMAC of master_secret and "server finished" concatenated with the hash of all the handshake messages so far.

d) Why are the ChangeCipherSpec and Finished messages always sent together?

Answer:
From Dierks & Rescorla, The Transport Layer Security (TLS) Protocol Version 1.2, (RFC 5246) 2008:

Change Cipher Spec Protocol
The change cipher spec protocol exists to signal transitions in ciphering strategies. The protocol consists of a single message, which is encrypted and compressed **under the current (not the pending) connection state.** The message consists of a single byte of value 1.
...

Finished
When this message will be sent:
A Finished message is **always sent immediately after a change cipher spec message** to verify that the key exchange and authentication processes were successful. It is essential that a change cipher spec message be received between the other handshake messages and the Finished message.

Meaning of this message:
The Finished message is the first one protected with the just negotiated algorithms, keys, and secrets. Recipients of Finished messages **MUST verify that the contents are correct.** Once a side has sent its Finished message and received and validated the Finished message from its peer, it may begin to send and receive application data over the connection.
...

handshake_messages
All of the data from all messages in this handshake up to, but not including, this message.

...

Note: ChangeCipherSpec messages, alerts, and any other record types are not handshake messages and are not included in the hash computations.