OLLSCOIL NA hÉIREANN
THE NATIONAL UNIVERSITY OF IRELAND, CORK

COLÁISTE NA hOLLSCOILE, CORCAIGH
UNIVERSITY COLLEGE, CORK

Summer Examination 2013

CS4614: Introductory Network Security

Professor I. Gent,
Professor B. O'Sullivan,
Dr. S.N. Foley

Answer *all* questions

1.5 Hours

A *Note to extern. This module is 5 ECTS credits. The exam paper is graded out of 80 marks with 20 marks for Continuous Assessment (per Marks and Standards).*

Questions are either: straightforward regurgitation of material; a reasonably familiar problem that requires application of knowledge, or intended to stretch the student with more challenging/unfamiliar problems. The intention is that a student who can regurgitate material can pass; a student who not only 'knows' the material but can apply it in straightforward ways can achieve a second class honours student. A first class honours fits the two previous categories and can apply the knowledge in more challenging ways to trickier and unfamiliar problems. $\overline{\mathbf{A}}$

1. a) Alice sends message M to Bob over a untrusted network. Assuming they share a secret, sketch how message secrecy, integrity and authentication should be provided. (*6 marks*)

A *If Alice and Bob share a strong cryptographic secret key K_{AB} then*

$$A \rightarrow B : E^{IV}_{CBC}(K_{AB}, M.h(M))$$

where $h()$ a one-way cryptographic hash function and $E^{IV}_{CBC}()$ is a block cipher configured for encryption in CBC mode with fresh initialization vector IV. **A**

b) Explain how *salt* defends against a password *pre-computation dictionary attack*. (*6 marks*)

A *Looking for explanation of dictionary attack using hash-table of dictionary (pass)words and how salt can significantly increase its size.* **A**

c) In the movie *Skyfall*, James Bond's Walther PPK handgun has a biometric reader designed to recognise his palm print, so that *only* he can fire it. Explain whether the designers of this authentication mechanism need to worry about the Birthday Paradox. (*6 marks*)

A *Intuitively, the Birthday Paradox tells us that the probability that k agents will all have distinct palmprints is less that 0.5 if $k > \sqrt{1/FAR}$, where FAR is the false accept rate for the biometric system. The designers do not have to worry about the birthday paradox since the handgun only has to recognize James Bond's palm and nobody else. It does not store biometrics for a group of agents, and therefore the birthday paradox does not apply. For the given gun, the probability of someone else being false recognized as James Bond is given by FAR.* **A**

d) Alice receives a document signed by Bob and a certificate for his public key. Sketch the operations carried out by Alice to confirm the document's authenticity. (*6 marks*)

A *Bob has public, private key pair (K_B, K_B^{-1}) Let $\{Bob, expiryDate, ...K_B\}_{sK_T}$ and $\{Doc\}_{sK_B}$ denote Bob's certificate and signed document, respectively. We assume that Alice knows of and trusts the signing CA K_T. The signed document corresponds to the document and the hash of the document encrypted using Bob's private key K_B^{-1}*

Alice checks the the signature on the document by re-computing the hash of the document and comparing it against the encrypted hash provided (decrypted using Bob's public key). If valid then Alice checks the validity of the certificate for K_B by confirming that its properly signed by a CA she trusts and that it has not expired. She may also consult a CRL or OCSP server (if specified in the certificate) to check whether the certificate is revoked. If the document signature and certificate is valid then Alice can believe that the document was signed by Bob. **A**

e) If Alice and Bob know each other's public keys (K_A and K_B, respectively) and K_{AB} is a session key, then explain which of the following provide a digital signature for message M.

$$A \rightarrow B : \{M\}_{K_{AB}}, \{\{h(M), K_{AB}, A, B, \ldots\}_{K_A^{-1}}\}_{K_B} \quad (1)$$

$$A \rightarrow B : \{M, h(M)\}_{K_{AB}}, \{\{K_{AB}, A, B, \ldots\}_{K_A^{-1}}\}_{K_B} \quad (2)$$

(*6 marks*)

A *In Protocol (1), Alice signs the hash of the message M and thus, its not possible for Bob to for example modify M and claim it came from Alice. In Protocol (2), only the shared key is signed, and thus Bob can modify the message, recompute the hash, and claim that $\{M', h(M')\}_{K_{AB}}$ came from Alice.* **A**

(*30 total marks*)

---

2. Alice A and Bob B share secret keys K_{AT} and K_{BT}, respectively, with trusted authentication server T. Alice wishes to communicate securely with Bob and initiates the following protocol.

$$\begin{aligned}
\text{Msg1}: &\quad A \to B &&: A \\
\text{Msg2}: &\quad B \to T &&: A, B \\
\text{Msg3}: &\quad T \to B &&: \{K_{AB}\}_{K_{AT}}, \{B\}_{K_{AT}}, \{K_{AB}, A\}_{K_{BT}} \\
\text{Msg4}: &\quad B \to A &&: \{K_{AB}\}_{K_{AT}}, \{B\}_{K_{AT}},
\end{aligned}$$

a) Describe how this protocol should be used to provide authenticated secure access to network resources. Highlight how it is different to a Kerberos-style protocol. (*15 marks*)

A *Looking for straightforward description of how a protocol establishes secure and authentic key between Alice and Bob. This is not unlike relationships in a Kerberos style protocol, however the description needs to explain the slightly different setup between initiator, respondent and authentication service.*

In Kerberos, the assumption is that a connection exists between the authentication server and the initiator/Alice, while it is not necessary to have a direct connection between the the authentication server and the respondent/Bob. In the above protocol, the implicit assumption is that the connection is between the respondent and the authentication server while a connection between the initiator and authentication server is not necessary. In Kerberos the initiator/Alice directly contacts the authentication server for a ticket for the service/respondent/Bob. In the above protocol, the respondent Bob must contact the authentication server for a ticket for the initiator/Alice. For example, in a conventional workstation setup under Kerberos, the workstation contacts the Kerberos server. The above protocol could be used, for example, when a smart card (initiator) does not have a connection to an authentication server, but the smart card reader (respondent) does have a connection. **A**

b) Describe an attack on the protocol whereby a malicious user Mike can trick Alice into believing that she is initiating a secure connection with Bob (but it is actually Mike). (*10 marks*)

A *Mike eavesdrops on past run α of the protocol between Alice and Bob and extracts attribute $\{B\}_{K_{AT}}$ from*

$$Msg\alpha 3: \quad T \to B \quad : \{K_{AB}\}_{K_{AT}}, \{B\}_{K_{AT}}, \{K_{AB}, A\}_{K_{BT}}$$

Mike, a participant, shares a key K_{MT} with T and initiates another run β with T:

$$\begin{aligned}
Msg\beta 2: &\quad M \to T &&: A, M \\
Msg\beta 3: &\quad T \to M &&: \{K_{AM}\}_{K_{AT}}, \{M\}_{K_{AT}}, \{K_{AM}, A\}_{K_{MT}}
\end{aligned}$$

Mike keeps a copy of $\{K_{AM}\}_{K_{AT}}$ and obtains key K_{AM} by decrypting $\{K_{AM}, A\}_{K_{MT}}$.

Mike sets himself up on the network, pretending to be Bob and waits for Alice to connect in run δ.

$$Msg\delta 1: \quad A \to B[M] \quad : A$$

Mike responds to Alice with the attributes collected from runs α and β:

$$Msg\delta 4: \quad B[M] \to A \quad : \{K_{AM}\}_{K_{AT}}, \{B\}_{K_{AT}},$$

on receipt of message $\delta 4$, Alice believes she is sharing the key K_{AM} with Bob. **A**

(*25 total marks*)

Question 3 overleaf.

3. UCC lecturer Alice securely submits exam results to a network-based Exams-Office service using the Java code fragment below. Results `rslts` are sent over a socket-based connection (encapsulated as `DataOutputStream out`). Alice's Java KeyStore `keystore` stores her public DSA key, alias `"alicePK"`.

```
Random rangen = new Random(0);
byte[] keySession = new byte[2];
rangen.nextBytes(keySession);
SecretKeyFactory desF = SecretKeyFactory.getInstance ("DES");
KeySpec ks = new DESKeySpec(keySession);
SecretKey key = desF.generateSecret(ks)
Cipher cipher = Cipher.getInstance("DES/ECB/PKCS5Padding");
cipher.init(Cipher.ENCRYPT_MODE,key);
byte[] cBytes= cipher.doFinal(rslts)
out.write(cBytes);

String alice= "alicePassword".toCharArray();
PrivateKey priv = (PrivateKey) keystore.getKey("alicePK",alice);
Signature signature = Signature.getInstance ("DSA");
signature.initSign(priv);
byte[] sig = signature.sign(keySession);
out.write(sig);
```

a) Identify and explain the security vulnerabilities in this implementation. (*15 marks*)

A *Code implements the protocol* $A \rightarrow B : \{msg\}_K, \{K\}_{sK_A}$

Protocol problems: Secrecy is not provided by the protocol since it does not protect the secrecy of K*: any principal can read the key in* $\{K\}_{sK_a}$*, since it is only signed. Once in possession of the key* K*, the principal can decrypt* $\{M\}_K$ *and read the message* M*. Integrity is not provided since, given that the key* K *can be determined by any principal, then any principal (Eve) can change the message to* M' *and re-encrypt with* K *and* B *cannot detect the modification. Authentication is not provided since, given that the key* K *can be determined by any principal, then any principal (Eve) who sees a past exchange can generate a message and pretend that it came from* A*.*

Code Problems: Should not have a password to the keystore hard coded within a program. `java.util.Random` *is not a secure random number generator: an attacker can predict in advance future random numbers/session keys. The same value is used to seed the random number generator and thus every key will be the same. The effective session key size is only 16 bits, which is vulnerable to a brute force attack. The message is encrypted using DES-ECB mode, which is vulnerable to cut-paste attack and traffic analysis.* **A**

b) It has been suggested that it would be better to use Java SSL to secure the connection between Alice and Bob. Outline how Java SSL should be used in this case and include an explanation of how the use of public key certificates in the protocol can help Bob to discover Alice's public key. (*10 marks*)

A *Looking for a general discussion of Java SSL and in particular the configuration of the key-manager and trust manager for the above problem. This should also include discussion of the certificates, certificate-chains and the trusted third party. I dont expect students to generate complete syntax/working code with all the details. But what I am looking for is an explanation of the components and how they t together.* **A**

(*25 total marks*)
