Academic Year: Winter Semester 2021/22

6090: Security of Computer and Embedded Systems

Problem Sheet 8

Signatures and Public-key Infrastructures (PKIs)

In this problem sheet, we will

- deepen our knowledge of digital signatures, public-key infrastructures and the web of trust,
- analyze the hierarchic certification authority model and compare it to the distributed web of trust.

1. Signatures and Public-key Infrastructures

In this section, we will revise a couple of key concepts of the public-key infrastructures as well as the underlying cryptographic concepts. You might want to review the lecture and slides and read Hein [1] before working on these questions.

Exercise 1: ***Cryptographic Concepts***

1- A(n) ………….. function creates a message digest out of a message

□ a) encryption

□ b) decryption

□ c) hash

□ d) none of the mentioned

2- A digital signature needs a(n) ………….. system

□ a) symmetric key

□ b) asymmetric key

□ c) either a) or b)

□ d) neither a) nor b)

3- A difference between the PKI used by TLS for web browsers and the Web of Trust used by PGP is

□ a) PGP keys can be signed by any other user

□ b) PGP keys are certified in a hierarchical manner

□ c) PGP keys have no expiry date

□ d) PGP keys can use any type of public-key algorithms

4- Which of the following is true about Public-key Infrastructure?

□ a) PKI uses two-way symmetric key encryption with digital certificates, and Certificate Authority.

□ b) PKI uses private and public keys but does not use digital certificates.

□ c) PKI is a combination of digital certificates, public-key cryptography, and certificate authorities that provide enterprise wide security.

□ d) PKI uses only symmetric key encryption.

5- Digital signature provides

□ a) authentication

□ b) non-repudiation

□ c) both a) and b)

□ d) neither a) nor b)

6- PGP uses

□ a) a web of trust between the participants

□ b) a hierarchical trust model

□ c) public-key cryptography

□ d) different levels of trust

Exercise 2: *Signatures*

Consider the RSA crypto scheme with the following configuration:

- **Alice**'s public-key is (n_a, e_a) = (33, 7), her private key is d_a = 3
- **Bob**'s public-key is (n_b, e_b) = (65, 7), his private key is d_b = 7

Suppose that we encode the letters by their position in the alphabet (e.g., the letter "a" is represented by the number 1, spaces are not encoded).

1. Alice wants to send the signed message "meet at noon" to Bob.
   Encode the letters by their position in the alphabet (e.g., the letter "a" is represented by the number 1) and compute the signature for each character of the message (i.e., to simplify the problem, we do not hash the message).
2. How can Bob check, that the message is signed by Alice?
3. Assume Alice is using the same key-pair for signing documents and for encrypting messages. Explain if this is a problem or not.

3. Certification Authorities

In this section, we will have a closer look on public-key infrastructures (PKIs) in general and the X.509 Public-key Infrastructure in particular. You might want to review the lecture slides and read Hein [1] before working on these questions. For a deep dive of X.509 certificates, you might want to have a look at RFC3280 (https://www.ietf.org/rfc/rfc3280.txt).

Exercise 3: *Certificate Chains*

Figure 1 shows a chain of X.509 certification authorities (and users). The lines indicate the hierarchical relationship among the certificate authorities. Circles indicate certificate authorities and boxes indicate end users.

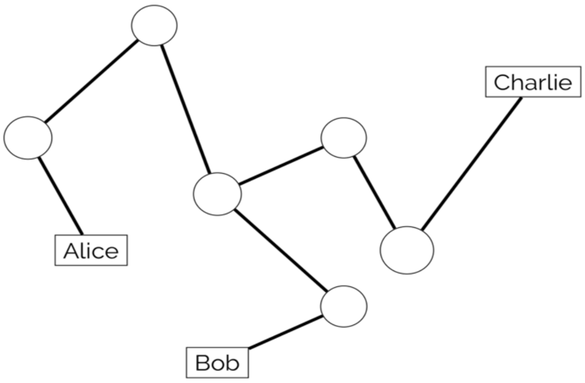


Figure 1: An example of a X.509 CA

1. Given the following information
   - When Alice wants to obtain the public-key of Bob, the chain path is $Q \ll S \gg, S \ll U \gg, U \ll R \gg, R \ll Bob \gg$
   - When Charlie wants to obtain the public-key of Bob, the chain path is $W \ll T \gg, T \ll U \gg, U \ll R \gg, R \ll Bob \gg$

   Write the names of the certification authorities in their appropriate circles.

2. Indicate the minimum certificates that must be maintained in the directory for each CA along the paths used in 1). Which public keys must be maintained by the users Alice, Bob and Charlie?

3. Suppose the certificate of the CA that issues Alice's certificate gets stolen.
   a. What actions should be taken by Alice's CA?
   b. What actions should be taken by the other CAs?
   c. Assume an attacker now generated a new certificate for Alice. How can Charlie detect this when validating Alice's-public key?

Exercise 4: *Intermediate CAs*

Figure 2 shows n tree-structured, X.509 PKI: we have one global CA (the root CA r) as the root of the tree. The root CA does not issue certificates to end-entities itself. Rather, it delegates this task to intermediate CAs ($i_{x,y}$). The idea is that the root CA signs the certificate of the next intermediate CAs down in the tree, which in turn sign the next intermediate CAs down in the tree, etc. Finally, some intermediate CA will sign certificates of end entities (e_i).

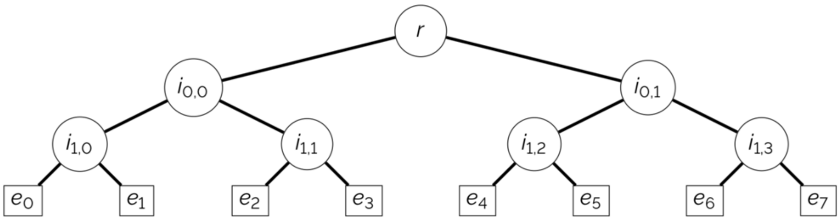


Figure 2: An example of X.509 CA hierarchy

1. What purpose could intermediate CAs possibly serve? Give at least one example.
2. There is an implicit assumption made in this form of PKI that is often called "transitive trust". Explain: When Bob wishes to verify the certificate, which entities must he ***trust*** to issue correct certificates in the first step of the verification? What about next steps?

Exercise 5: *Self-signed Certificates*

Certificates, e.g., as part of an X.509 PKI, can also sign itself (i.e., it is signed by the same entity whose identity it certifies). The result is a so called *self-signed* certificate.

Let us assume the web server of an online-only (i.e., the bank does not have any branches) bank X is using a self-signed certificate.

1. Why is company X using a self-signed certificate for their web server? Name at least one advantage.
2. What privacy and authenticity guarantees does the self-signed certificate provide?
3. Would the situation be different, if the bank would have regular branches that you need to visit for opening an account? What would you, as a customer, need to do?

4. Web of Trust

In this section, we will have a closer look on the trust model of PGP. We assume that Alice, Bob, Carl, Dave, Fred, Garry, and Elena are using PGP.

Table 1 shows their (public) keyrings. Again, you might want to review the lecture slides and read Hein [1] before working on these questions.

Exercise 6: ***Public Keyrings***

Table 1 shows a set of keyrings as used by Alice, Bob, Carl, Dave, Fred, Garry, and Elena. Draw a direct graph that visualizes this set of keyrings. Represent the participants as nodes and use the following instructions for constructing the edges of the graph:

$X \longrightarrow Y$	X has certified the key of Y
$X \underset{Z}{\dashrightarrow\bullet} Y$	X trusts Y with the trust level Z
$X \xrightarrow[W]{} Y$	The Key of Y has for X the legitimacy W

where $W \in \{Unkown, Marginal, Complete\}$ is the Key Legitimacy and the $Z \in \{Unkown, Untrusted, Marginal, Complete, Ultimate\}$ is the Trust Level. Recall that the

A key k has the legitimacy "complete" for a user X, if and only if one of the following conditions holds:

- k is signed by the user X
- k is signed by a user Y and X trusts the user Y completely (*Trust in Signer* is "complete")
- k is signed by two different users Y and Y', and X trusts both Y and Y' marginally (*Trust in Signer* is "marginal")

Note that Table 1 does not contain the values of legitimacy and the trust in signer. This should be deduced by analysing the relations between different entities.

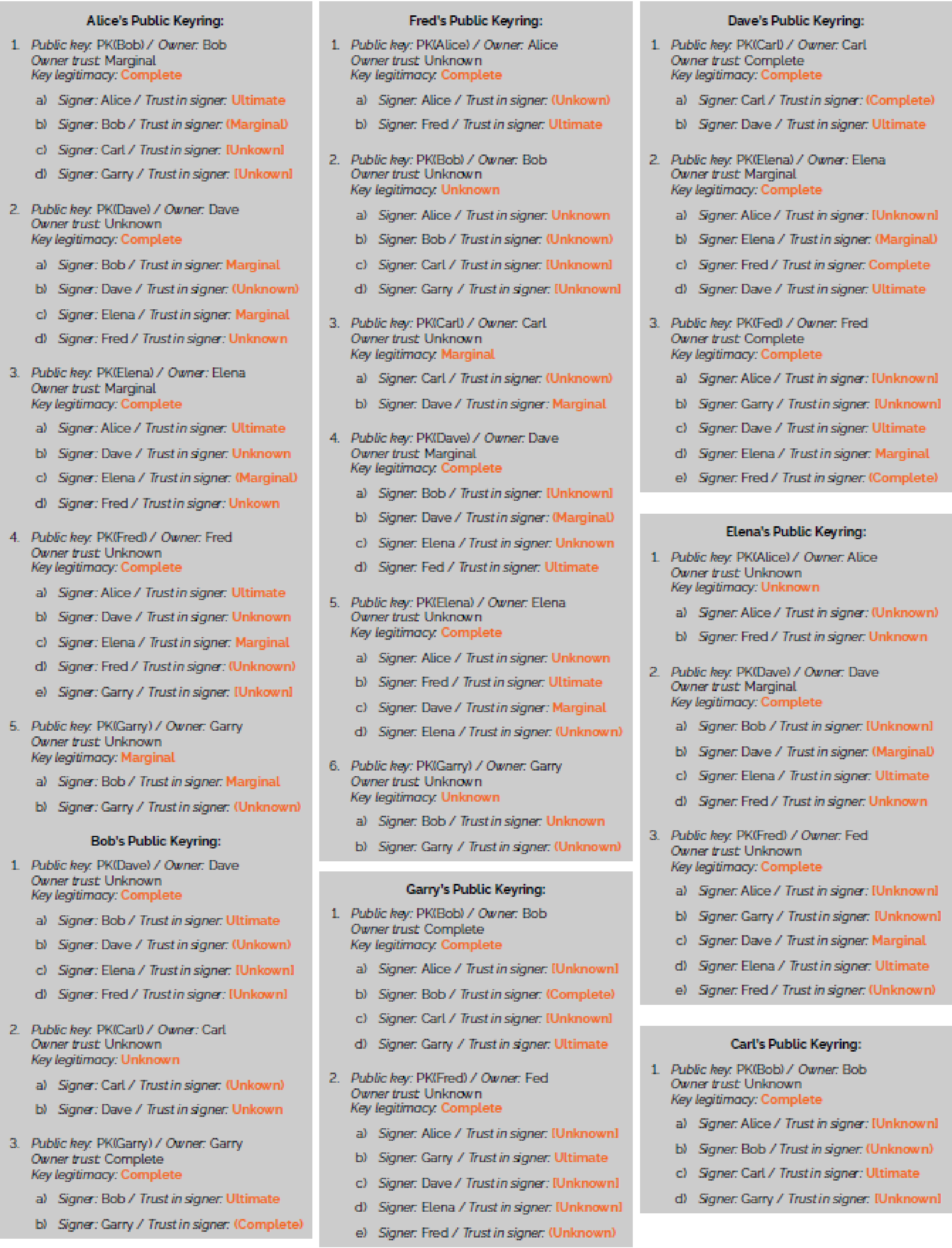
To make the graph easier to understand,

- draw only the key legitimacy for Alice and
- omit edges where $W = Unkown$ or $Z = Unkown$.

References

[1] B. Hein. PKI Trust Models: Whom Do You Trust? Technical report, The SANS Institute, 2013. Available online at: https://www.sans.org/reading-room/whitepapers/vpns/pki-trust-models-trust-36112

Table 1: Public Keyrings for Alice, Bob, Carl, Dave, Fred, Garry, and Elena



Alice's Public Keyring:

1. *Public key:* PK(Bob) / *Owner:* Bob
   Owner trust: Marginal
   Key legitimacy: Complete
   a) *Signer:* Alice / *Trust in signer:* Ultimate
   b) *Signer:* Bob / *Trust in signer:* (Marginal)
   c) *Signer:* Carl / *Trust in signer:* [Unkown]
   d) *Signer:* Garry / *Trust in signer:* [Unkown]
2. *Public key:* PK(Dave) / *Owner:* Dave
   Owner trust: Unknown
   Key legitimacy: Complete
   a) *Signer:* Bob / *Trust in signer:* Marginal
   b) *Signer:* Dave / *Trust in signer:* (Unknown)
   c) *Signer:* Elena / *Trust in signer:* Marginal
   d) *Signer:* Fred / *Trust in signer:* Unknown
3. *Public key:* PK(Elena) / *Owner:* Elena
   Owner trust: Marginal
   Key legitimacy: Complete
   a) *Signer:* Alice / *Trust in signer:* Ultimate
   b) *Signer:* Dave / *Trust in signer:* Unknown
   c) *Signer:* Elena / *Trust in signer:* (Marginal)
   d) *Signer:* Fred / *Trust in signer:* Unkown
4. *Public key:* PK(Fred) / *Owner:* Fred
   Owner trust: Unknown
   Key legitimacy: Complete
   a) *Signer:* Alice / *Trust in signer:* Ultimate
   b) *Signer:* Dave / *Trust in signer:* Unknown
   c) *Signer:* Elena / *Trust in signer:* Marginal
   d) *Signer:* Fred / *Trust in signer:* (Unknown)
   e) *Signer:* Garry / *Trust in signer:* [Unkown]
5. *Public key:* PK(Garry) / *Owner:* Garry
   Owner trust: Unknown
   Key legitimacy: Marginal
   a) *Signer:* Bob / *Trust in signer:* Marginal
   b) *Signer:* Garry / *Trust in signer:* (Unknown)

Bob's Public Keyring:

1. *Public key:* PK(Dave) / *Owner:* Dave
   Owner trust: Unknown
   Key legitimacy: Complete
   a) *Signer:* Bob / *Trust in signer:* Ultimate
   b) *Signer:* Dave / *Trust in signer:* (Unkown)
   c) *Signer:* Elena / *Trust in signer:* [Unkown]
   d) *Signer:* Fred / *Trust in signer:* [Unkown]
2. *Public key:* PK(Carl) / *Owner:* Carl
   Owner trust: Unknown
   Key legitimacy: Unknown
   a) *Signer:* Carl / *Trust in signer:* (Unkown)
   b) *Signer:* Dave / *Trust in signer:* Unkown
3. *Public key:* PK(Garry) / *Owner:* Garry
   Owner trust: Complete
   Key legitimacy: Complete
   a) *Signer:* Bob / *Trust in signer:* Ultimate
   b) *Signer:* Garry / *Trust in signer:* (Complete)

Fred's Public Keyring:

1. *Public key:* PK(Alice) / *Owner:* Alice
   Owner trust: Unknown
   Key legitimacy: Complete
   a) *Signer:* Alice / *Trust in signer:* (Unkown)
   b) *Signer:* Fred / *Trust in signer:* Ultimate
2. *Public key:* PK(Bob) / *Owner:* Bob
   Owner trust: Unknown
   Key legitimacy: Unknown
   a) *Signer:* Alice / *Trust in signer:* Unknown
   b) *Signer:* Bob / *Trust in signer:* (Unknown)
   c) *Signer:* Carl / *Trust in signer:* [Unknown]
   d) *Signer:* Garry / *Trust in signer:* [Unknown]
3. *Public key:* PK(Carl) / *Owner:* Carl
   Owner trust: Unknown
   Key legitimacy: Marginal
   a) *Signer:* Carl / *Trust in signer:* (Unknown)
   b) *Signer:* Dave / *Trust in signer:* Marginal
4. *Public key:* PK(Dave) / *Owner:* Dave
   Owner trust: Marginal
   Key legitimacy: Complete
   a) *Signer:* Bob / *Trust in signer:* [Unknown]
   b) *Signer:* Dave / *Trust in signer:* (Marginal)
   c) *Signer:* Elena / *Trust in signer:* Unknown
   d) *Signer:* Fed / *Trust in signer:* Ultimate
5. *Public key:* PK(Elena) / *Owner:* Elena
   Owner trust: Unknown
   Key legitimacy: Complete
   a) *Signer:* Alice / *Trust in signer:* Unknown
   b) *Signer:* Fred / *Trust in signer:* Ultimate
   c) *Signer:* Dave / *Trust in signer:* Marginal
   d) *Signer:* Elena / *Trust in signer:* (Unknown)
6. *Public key:* PK(Garry) / *Owner:* Garry
   Owner trust: Unknown
   Key legitimacy: Unknown
   a) *Signer:* Bob / *Trust in signer:* Unknown
   b) *Signer:* Garry / *Trust in signer:* (Unknown)

Garry's Public Keyring:

1. *Public key:* PK(Bob) / *Owner:* Bob
   Owner trust: Complete
   Key legitimacy: Complete
   a) *Signer:* Alice / *Trust in signer:* [Unknown]
   b) *Signer:* Bob / *Trust in signer:* (Complete)
   c) *Signer:* Carl / *Trust in signer:* [Unknown]
   d) *Signer:* Garry / *Trust in signer:* Ultimate
2. *Public key:* PK(Fred) / *Owner:* Fed
   Owner trust: Unknown
   Key legitimacy: Complete
   a) *Signer:* Alice / *Trust in signer:* [Unknown]
   b) *Signer:* Garry / *Trust in signer:* Ultimate
   c) *Signer:* Dave / *Trust in signer:* [Unknown]
   d) *Signer:* Elena / *Trust in signer:* [Unknown]
   e) *Signer:* Fred / *Trust in signer:* (Unknown)

Dave's Public Keyring:

1. *Public key:* PK(Carl) / *Owner:* Carl
   Owner trust: Complete
   Key legitimacy: Complete
   a) *Signer:* Carl / *Trust in signer:* (Complete)
   b) *Signer:* Dave / *Trust in signer:* Ultimate
2. *Public key:* PK(Elena) / *Owner:* Elena
   Owner trust: Marginal
   Key legitimacy: Complete
   a) *Signer:* Alice / *Trust in signer:* [Unknown]
   b) *Signer:* Elena / *Trust in signer:* (Marginal)
   c) *Signer:* Fred / *Trust in signer:* Complete
   d) *Signer:* Dave / *Trust in signer:* Ultimate
3. *Public key:* PK(Fed) / *Owner:* Fred
   Owner trust: Complete
   Key legitimacy: Complete
   a) *Signer:* Alice / *Trust in signer:* [Unknown]
   b) *Signer:* Garry / *Trust in signer:* [Unknown]
   c) *Signer:* Dave / *Trust in signer:* Ultimate
   d) *Signer:* Elena / *Trust in signer:* Marginal
   e) *Signer:* Fred / *Trust in signer:* (Complete)

Elena's Public Keyring:

1. *Public key:* PK(Alice) / *Owner:* Alice
   Owner trust: Unknown
   Key legitimacy: Unknown
   a) *Signer:* Alice / *Trust in signer:* (Unknown)
   b) *Signer:* Fred / *Trust in signer:* Unknown
2. *Public key:* PK(Dave) / *Owner:* Dave
   Owner trust: Marginal
   Key legitimacy: Complete
   a) *Signer:* Bob / *Trust in signer:* [Unknown]
   b) *Signer:* Dave / *Trust in signer:* (Marginal)
   c) *Signer:* Elena / *Trust in signer:* Ultimate
   d) *Signer:* Fred / *Trust in signer:* Unknown
3. *Public key:* PK(Fred) / *Owner:* Fed
   Owner trust: Unknown
   Key legitimacy: Complete
   a) *Signer:* Alice / *Trust in signer:* [Unknown]
   b) *Signer:* Garry / *Trust in signer:* [Unknown]
   c) *Signer:* Dave / *Trust in signer:* Marginal
   d) *Signer:* Elena / *Trust in signer:* Ultimate
   e) *Signer:* Fred / *Trust in signer:* (Unknown)

Carl's Public Keyring:

1. *Public key:* PK(Bob) / *Owner:* Bob
   Owner trust: Unknown
   Key legitimacy: Complete
   a) *Signer:* Alice / *Trust in signer:* [Unknown]
   b) *Signer:* Bob / *Trust in signer:* (Unknown)
   c) *Signer:* Carl / *Trust in signer:* Ultimate
   d) *Signer:* Garry / *Trust in signer:* [Unknown]