EP2500/EP3200 Networked Systems Security

Final Exam

Panagiotis Papadimitratos

Total: 320 points (plus 130 extra credit points). Pass (MSc students): 176, Pass (PhD students): 256

Closed books. Closed notes. No calculators.

Please work alone, no communication with classmates.

Please use your computer and webcam as per the instructions for zoom-based exams.

Please hand-write your answers on numbered paper sheets, each having clearly with your name.

Hint: Scan the exam and identify exercises that seem relatively easier.

Craft your strategy to avoid getting stuck for long with harder ones.

Choose wisely when to look at the extra credit questions.

January 13, 2021

Contents

Please see and use for all your answers the notation in Table 1. If you think you need any additional notation, please define and justify this explicitly and state any assumptions you make.

Consider an adversary that can never compromise the secret or private key(s) of the involved entities (unless explicitly stated otherwise), but she can intercept, forge, modify or erase any transmitted message.

For each answer you give, e.g., for each protocol you design, please give a brief (no more than 3-4 lines, unless absolutely necessary) explanation. When you present a protocol, please do not forget to present the message exchange and the operations (computations) on both (all) sides (involved entities).

Table 1: Notation of cryptographic primitives.

Symbol	Description
$A : \alpha$	Entity A takes action α
$x \leftarrow y$	The value y is assigned to variable x
RNG	A random number is generated
$x == y$	Check if x is equal to y
Pub_A	Public Key of A (known to all other hosts)
$Priv_A$	Private Key of A
$Cert_A$	Certificate of A containing Pub_A
K_A/K_{AB}	Symmetric secret generated by A / shared by A and B
$E_k(m)$	Encrypt m with the key k (symmetric or asymmetric)
$D_k(m)$	Decrypt m with the key k (symmetric or asymmetric)
$h = H(m)$	Hash of m, output of a one-way/hash function H with input m
$(x, y, \ldots) \rightarrow A$	Message sent to A containing x, y, $\ldots$
t^A_{clock}	The clock value of entity A
N_A	Nonce chosen by A

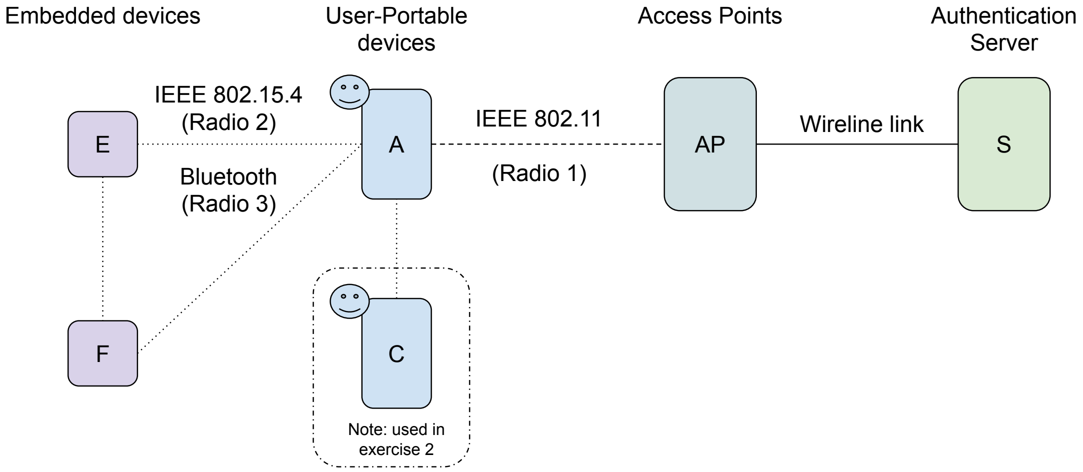

Figure 1: Exercise 1 and 2: IoT representation with user-portable devices (A) (and C for exercise 2), access points (AP), authentication servers (S), and embedded devices (E and F).

1 Security protocols and key management

Exercise 1 Symmetric key security protocols (80 points + 30 points extra credit)

Consider a diverse Internet of Things (IoT) environment. Users carry portable devices, e.g., smartphones or tablets, with wireless networking capabilities, notably IEEE 802.11 (call this *Radio 1*), IEEE 802.15.4 (*Radio 2*), and BlueTooth Low Energy (*Radio 3*). Moreover, consider embedded devices with sensing and actuating capabilities operating with transceivers of type *Radio 2* or *Radio 3*; wireless access points (APs) operating with transceivers of type *Radio 1*; and an authentication server, S, at the back-end, connected via a wireline link (or subnetwork or the Internet) to the APs (see Fig. 1).

Each user device, A, embedded device, E, and access point, AP, shares a symmetric key with the server S, stored in the device at the time of bootstrapping (i.e., before it is handed to the user or deployed).

1. Let a user device, A, and an access point, AP, with no a priori association, get within communication range. The system needs a simple method for users to securely communicate over the wireless medium, based on a key shared by each A and each AP. It is important that each user device has its own key, distinct from the ones of the other user devices connected to the specific AP. Assume that A sends a *security association establishment request* message to the AP but until the association establishment is completed A is not granted network access, i.e., it is not allowed by AP to access any other entity over the Internet.

   Design a protocol that leverages the A-S and AP-S trust (security associations) and results in A and AP having the sought key.

Assume that A initiates the protocol, moreover, assume a single identifier for each device, e.g., a data link/medium access address and assume those known to S for all devices that are bootstrapped/associated with it. For simplicity, assume that there is no need that the cryptographic key itself be derived from contributions of more than one involved party. Finally, you can assume that A is only loosely synchronized with AP (and S) before establishing a security association with AP.

2. Explain concisely why your protocol achieves the following:
   - No unregistered device A (not bootstrapped/associated with S) can obtain a shared key with AP.
   - No attacker, M, overhearing the A-AP exchange (in both or either direction(s)), can modify at will the established (with the help of S) key by A and AP.
   - No attacker, M, as the aforementioned one, can replay messages from older executions of the protocol (possibly its own earlier protocol with AP) and mislead A.
   - More generally, no attacker, M, as the one in the previous question, can act as a rogue AP (not associated with S) and mislead legitimate devices A to establish connections with it.
3. Does your protocol provide protection against an attacker M that is connected on the same wireline network as AP and S? For the third sub-case above for M, consider the 'wireline' attacker trying to pose as an AP to S. Please give a brief justification and respond only with respect to your protocol.
4. If not already done in your first design, please add one more feature to your protocol: a confirmation message exchange (handshake) that allows A and AP to confirm that indeed the other party has exactly the obtained key.
5. Does your protocol provide protection against a compromised embedded device E? Please give a brief justification.
6. Can the same compromised embedded device E cause a Denial of Service DoS for the A-AP-S communication
7. Now consider an embedded device E; the user with device A comes within range of E (over *Radio 2* or *Radio 3)*, while she/it has access to the Internet over *Radio 1*. E provides specific measurements A is interested in. How can A and E establish a security association exactly in this setup? Recall that E does not have any connection to rest of the network, other than its own radio. Provide your protocol.
8. Assume E is queried by A and responds by sending messages containing each the ten most recent measurements (assume they are taken every second). Thus, it can transmit a message every 10 seconds. A needs to verify the authenticity of each message and the integrity of a sequence of such messages (over a period of one minute). Moreover, data must be kept confidential. Please design and present your protocol, stating your assumptions.
9. **Extra credit: 30 points** Next, assume A moves a bit, establishes an association (symmetric key) with another embedded device F (as it did with E, no need to repeat any details of the protocol).

E and F are neighbors, i.e., they can communicate directly. Although they are not supposed to in general (note: mostly, bidirectional portable device-to-embedded device communication). Now, A wishes to “configure” F and E so that they pass measurements from one to another when A is away (out of range) and store them; for fault tolerance and load balancing. A can then query either node and obtain the sought data.

Describe a protocol, possibly with a single message that A can send to E and F, such that A sets up the necessary information to both of them and:

- E learns the identity (or medium access control address) of F and vice-versa
- E can authenticate messages F passes to it and vice-versa, preventing replays and detecting message losses (“gaps”).
- A can authenticate messages originating from E but stored by F (and the other way around)

Next, discuss if your protocol allows A to detect whether E skipped passing on messages to F for storing (or the other way around, the setting is symmetric). You can assume the periodic measurements as in question 8, with local and peer storage (e.g., F takes measurements and then stores them locally in the form of messages if A is not around to ask for them; also, it passes them to E, its peer). Can F make A believe that E has not passed all measurements in messages, although it does?

Exercise 2 Asymmetric/public key security protocols (50 points)

Consider the same setup as that in Exercise 1. The user-portable devices, A, are relatively more powerful than the embedded ones. Let's assume they have ample processing power to use regularly asymmetric key cryptography. Of course, this does not mean that they have unlimited processing power and resources; they remain portable devices.

It is important that each of the users be aware of the relative position of other users. The solution to achieve this is relatively simple: A transmits periodically "hello" messages; when another user C receives a "hello" message, it marks A as a neighbor, along with a timestamp. If n successive hello messages from A are missed (or no "hello" message is received within a period τ), A is purged from the neighbor table.

Assume that A and C are equipped each with a public-private key pair and a certificate provided by the same certificate authority, CA.

1. Augment the "hello" protocol so that these messages can be authenticated by any other user device in range, also allowing it to discard any replays. Design and present the protocol, justifying briefly, with a clear statement of your assumptions, why the objectives are achieved.
2. How would your design change if A and C can be safely assumed to be tightly synchronized?
3. Describe the communication and computation overhead of the secured "hello" protocol you designed. Assume the useful information is 32 bytes per "hello" message, and that, for simplicity, a digital signature is 32 bytes long, and a certificate 128 bytes long. Without forgoing the use of public key cryptorgraphy, propose at least one way to reduce communication overhead and at least one way to reduce computation overhead on the receiver side. Justify with a simple quantification. Try to not degrade security - if you do not manage to, please explain.
4. Assume that A wishes to keep track which of its neighbors, C, just added A as a neighbor. Once C receives a "hello" from A it responds with an ACK. Present an extension that ensures the authenticity, integrity, and freshness of those ACK.
5. What if A, once it has such an ACK, wishes to share measurements with C? Assume that A needs to push a large volume of messages collected from embedded devices (note: please do not be concerned with the origin authenticity of those messages). How can C authenticate **efficiently** the large set of measurements, arranged in a large data file, that A passes on? Now, feel free to use both symmetric and asymmetric key cryptography to propose an efficient solution that keeps the measurements secret from any eavesdropper. *Without* assuming an *a priori* existence of symmetric shared keys, briefly explain how your solution achieves the objectives and why it is efficient.

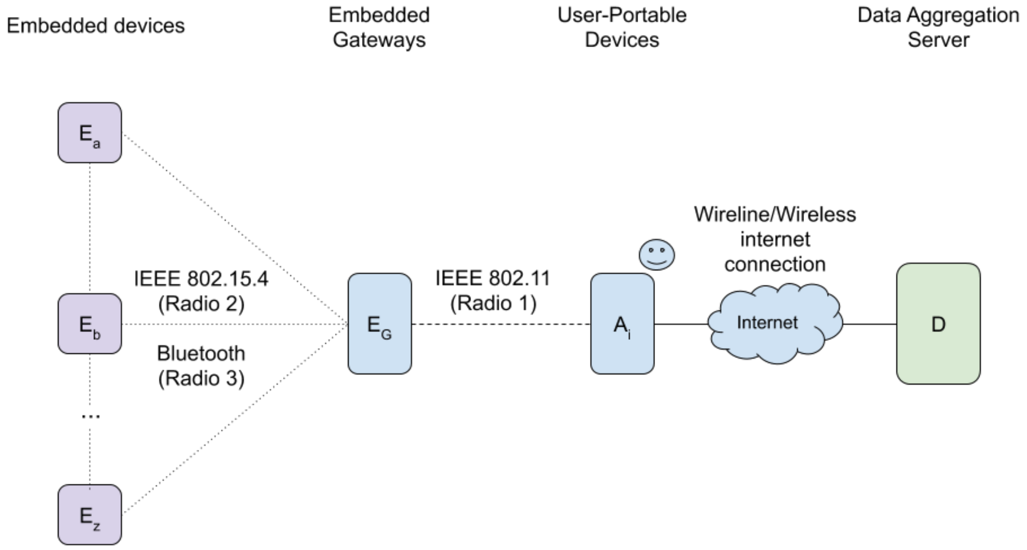

Figure 2: IoT representation with user-portable devices A_i, embedded devices E_i, Embedded gateways E_G, Data Aggregation servers D.

Exercise 3 Secure & private crowd-sensing (60 points + 40 points extra credit)

Consider a similar setup as that in Exercise 1 and Exercise 2 as illustrated in Fig. 2. The user-portable devices, A, and a subset of the embedded devices that are equally well equipped as As, can use regularly public key cryptography. We term the special-purpose embedded devices gateways, E_G. A and E_G are registered each with one Certification Authority, CA. In general, E_G are static and A mobile. Let us also assume there are CA_1, CA_2, and CA_3, each serving nodes in one domain respectively. Finally, there is a *data aggregation server*, D; users carrying As and owners of E_G agree, at will, to contribute data collected from embedded devices E_a, E_b, ..., E_z, and possibly A and E_G devices.

1. Let A be registered with CA_1 and E_G be registered with CA_2. Assume that, as in Exercise 2, A initiates a protocol to obtain data by E_G. How can A and E_G achieve mutual authentication based on public key primitives? Please do not re-write full protocols based on Exercise 2, but rather explain what additional steps, prior and/or during the protocols, would be needed.
2. Can an E_G instantiate a policy that allows differentiating the level of access (e.g., no access, low priority, high priority) for A_a, A_b, ..., A_z? Moreover, how can D differentiate which type of device it obtains the data from? Please discuss briefly.

3. Consider again the case of A obtaining data from E_G, authenticated, with their integrity and confidentiality preserved, over *Radio 1*. Now consider that the data are measurements that originate from a set of embedded devices, E_a, E_b, ..., E_z, that E_G directly communicated securely with, over *Radio 2* or *3*, using symmetric key primitives. Can A corroborate that a measurement originates from a precise embedded E_a? In other words, can the measurement origin authenticity and integrity be verified?

4. If not, what would you change in the design of the system to make this possible? Sketch your solution and explain. If you wish or you need to, you can assume E_G is trustworthy.

5. What would be the approach to address the requirement for the previous question if E_G cannot be trusted? Is your approach already addressing this case? Please explain or propose an alternative solution.

6. Now consider that, indeed, a misbehaving E_G is detected to not comply with the system specification, after an attacker compromised it. The system operator installs a new $E_{G-trusted}$. How can nodes A, with possibly no prior interaction with this system, be informed and ignore E_G? What if the attacker extracted the private key of E_G? Please feel free to leverage CA_2, the trusted third party E_G (and $E_{G-trusted}$) is registered with.

7. **Extra credit: 40 points**

   (a) Reconsider the role of A and E_G: what is the challenge, when D cannot authenticate each measurement but it relies on A or E_G?

   (b) Inversely, what is the drawback of a solution that ensures that each measurement, from E_a, E_b, ..., E_z, is directly authenticated by both A first and then D?

   (c) What if CA_1 provides an anonymized (or pseudonumized) certificate that omits A's identity? Can D identify A as the sender of any of the $data_i$, where $i = 1, \ldots, 5$? Can it link $data_2$ and, for example, $data_5$ to A? Please explain briefly why.

   (d) What if CA_1 provides A with five anonymized (or pseudonumized) certificates, $PNYM_1$, $PNYM_2$, ..., $PNYM_5$, to A? Can D identify A as the sender of any $data_i$, where $i = 1, \ldots, 5$? Can it link $data_2$ and, for example, $data_5$ to A? Please explain briefly why.

2 Secure Routing

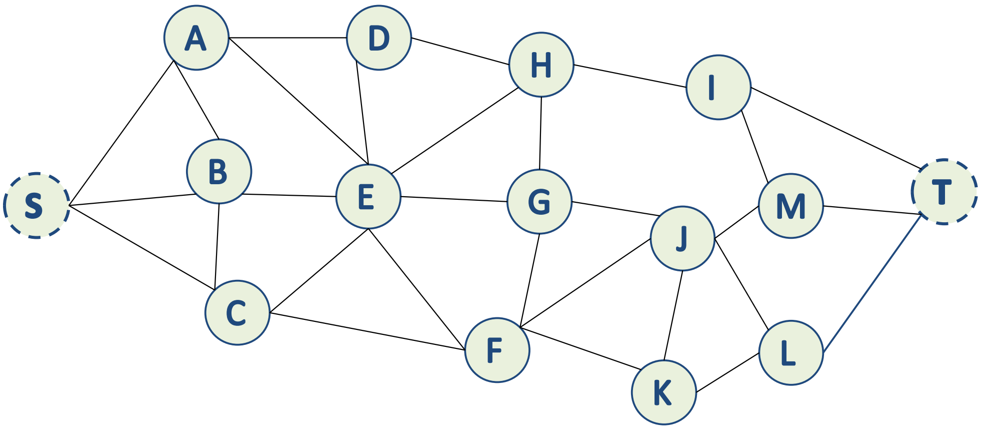

Figure 3: Wireless multi-hop network topology. For Exercise 5, S is the source node and T the destination node.

Consider the wireless multi-hop network in Fig. 3. Each line represents a wireless link, in other words signifying that the two nodes incident on a link are neighbors (i.e., communicate directly across the wireless medium, or, simply put, they are within range). If no link exists, then there is no direct connection. For example, F and H are out of range. Communication is locally a broadcast: e.g., a packet sent by A goes simultaneously over (A, S), (A, B), (A, D), and (A, E). Nonetheless, the local, node-to-node transmission can be a uni-cast, e.g., a message specifically sent from A to B as the sole intended receiver; given the wireless channel, such a message can be received by the other neighbors, S, D, E.

Exercise 4 Secure Link-State Routing (50 points + 30 points extra credit)

Consider first a secure link state routing protocol run by all the nodes. Each node be equipped with a public/private key pair and a certificate; all certificates are provided by the same CA. Nodes/routers can be considered loosely synchronized. Adjacent routers can be assumed to share symmetric keys. Please recall that routers broadcast *Link State Advertisements (LSAs)*, communicating links to their neighbors, to all other routers.

1. Please describe how a malicious router, M, can attempt to (a) introduce inexistent links connecting it to B and F and the inexistent Z, Y, and W routers, and (b) add two fictitious links connecting B and J and C and K.
2. Discuss how likely it is for M to succeed when there is adjacent router authentication; e.g., $G \rightarrow H : m, MAC_{K_{GH}}(m)$, that is, for any message m G passes to a neighbor H, there is an authenticator, e.g., a Message Authentication Code (MAC), using the shared key K_{GH}.
3. Augment the protocol by using public key cryptography; how could you prevent the attacks (a) and (b)? Explain the new LSA format and the operations at the sending and the receiving nodes and explain why attacks are thwarted (or not).
4. What if a link changes or breaks? Could M replay an older LSA and mislead the network the link remains intact? Explain why, based on earlier assumptions; or augment your protocol to thwart such an attack. Can M remove an operational link connecting two other routers, e.g., D and H?
5. Is it useful to maintain node-node, i.e., router-to-neighboring router, symmetric key based authentication when LSAs are protected based on public key cryptography? How many certificates does a node, S, need to have in order to be able to obtain a complete view of the topology.
6. **Extra credit, 30 points**

   (i) How can your protocol detect an attacker that transmits LSAs at an excessive rate? Please make any necessary assumptions. What if a benign router, J, receives excessive LSAs by M and forwards them? Could J be flagged as adversarial by its neighbors?

   (ii) What if the topology were highly dynamic? Could benign, correct routers be flagged as transmitting LSAs, honestly reflecting the latest network topology, at an excessive rate? If yes, how would you augment the protocol to avoid this?

   (iii) Please discuss the trade-off of your suggestion(s) in (ii).

Exercise 5 Secure Reactive Route Discovery (80 points + 30 points extra credit)

With reference to Fig. 3 again, consider a route discovery initiated by S, using the *Secure Routing Protocol (SRP)*: it sends a RREQ, looking for a route to T. Recall that each intermediate node, A,...,M, rebroadcasts each fresh RREQ once. Otherwise, it ignores a previously heard RREQ. Each route discovery is identified by a sequence number, Q_{SEQ}, and a random Q_{ID}.

Assume that S and T already share a symmetric key and with this one they can calculate a Message Authentication Code (MAC). You can assume the availability of public-private keys and certificates.

1. Recall that each intermediate node adds its identity to the RREQ they re-broadcast. Please describe the RREQ propagation, e.g., over A, D and H, etc, or any other path you prefer. Recall, however, that each node 'knows' (has a security association with) at most its neighbors and the destination(s) it needs to communicate with. In this case, S knows T and can discover A, B, and C.

2. Briefly discuss secure neighbor discovery, taking place asynchronously and in a sense proactively, before a route discovery is initated.

3. Consider a RREP crafted by T with the following fields:

   - Q_{SEQ}
   - $\{T, M, J, G, B, S\}$
   - $MAC_{K_{S,T}}(\{T, M, J, G, B, S\}, Q_{SEQ})$

   Is it a valid RREP? If not, please provide a valid RREP in response to the RREQ you described in part 1, above, for this exercise. Please explain how this or any RREP reaches back the source of the corresponding RREQ, i.e., S in our network, and how it is validated by S.

4. Given the RREQ propagation and RREP you described it thus far, can an adversary, E, prevent legitimate RREQs by S from being processed by D, C, ..., G, the adversary's neighbors? Explain why not, or why yes, and if yes please fix the problem.

5. Now, set aside the assumption that S and T already share a symmetric key. Instead, assume that they each has a public-private key pair and a certificate provided by the same *certification authority*. How can you modify the SRP route discovery using public key cryptography? In particular, can you have a protocol that allows S and T to establish a shared key *simultaneously* with the route discovery? State your notation, assumptions and describe your protocol. Explain briefly why the same security is achieved now as that based on the pre-established symmetric key holds. How can either of the two confirm that the established symmetric key is successfully obtained by the other party? (Hint: this will differ, depending on whether you used a transport or agreement approach)

6. For either of the above variants of SRP, consider now B as an adversary that attempt to "hide" itself from the discovered route. For example, can B mislead S and T that they are connected by a route $\{S, E, G, J, M, T\}$? If successful, what is the effect of such an attack, why does B perpetrate it?

7. What if each pair of nodes runs periodically a secure neighbour discovery protocol and use such information as a precursor to any route discovery? That is, as you discussed in part 2 above, a protocol that ensures them that they are neighbours or equivalently that their communication link is up? E.g., S knows that (S, A), (S, B) and (S, C) are up (or equivalently that A, B and C are neighbours. You can assume that the network topology changes slowly enough so that none of these links goes down before the end of the subsequent route discovery. If you answered yes to the previous question 6, is the attack now stopped? If not, please explain.

8. Next, consider B and M being adversaries that collude, i.e., work together. Can they manipulate the route discovery so that they mislead S and T that they are connected by a route $\{S, B, M, T\}$? If yes, please outline their attack, what they need to know/have and how to evade the controls of SRP. What is the effect of such an attack, why do B and M perpetrate it? If they cannot, explain how the protocol stops such an attempt.

9. **Extra credit, 30 points** What is the message (RREQ plus RREP messages) complexity of the route discovery? Can the protocol discover multiple routes or only one? If multiple routes can be discovered, are they disjoint? If so, which protocol, the secure link state routing protocol in the

previous exercise or the SRP, is more likely to provide S with all available disjoint paths? How would the answers change if each intermediate node retransmitted (locally broadcasted) two copies of the same RREQ? What if mobility allows nodes to encounter nodes, establish keys or learn their certificates and public keys - can this be used to augment SRP and improve security? Sketch how.