Exercise 4

Exercises Peer-to-Peer-Systems and Security (SS2012)

Dr. Heiko Niedermayer
Lehrstuhl für Netzarchitekturen und Netzdienste
Technische Universität München

Thursday 28.6 2012
Hand-in: Thursday 5.7. 2012 in lecture or per mail
Exercise: Monday 9.7. 2012

Task 1 Authentication

In this task we specify a cryptographic protocol which is meant to be used for mutual authentication.

a) Specify on what information and when in the protocol do the entities A,B, and S detect a successful authentication run?
b) The protocol is insecure. Find an attack. (The strength of the attacker is that it can read, send, fake, and drop messages in the network, yet it cannot break cryptography. This is a common security model in network security.)

Hint: Use information from previous runs to attack the protocol. Sig_X stands for encryption with private key.

Protocol:

Prerequisites: S is TTP and for each participant X S knows the corresponding public key PK_X. All participants know the public key PK_S of S.
Let kab = hash(Na,Nb).
Protocol:

```
A -> S: A, Enc_PK_S(Na, B)
S -> A: PK_B, Enc_PK_A(Sig_S(Na, A))
A -> B: Enc_PK_B(Na, A, B, Sig_S(Na, A))
B -> A: Nb, {Na}kab
A -> B: {Nb}kab
```

Solution:

B recognizes A in the 5th message due to the signature from the server in message 3 and the knowledge of nb and kab.
A recognizes B in the 4th message due to the knowledge of Na and kab.
S does not detect success of the protocol run. It does not recognize A, but only provides information that only A can read, its signature of Na and A encrypted with A's public key.

Attack:

- The attacker C_A uses a previous communication of A and C_A to attack A and B.

A communicates with C (later to become C_A).

```
1. A -> S: A, Enc_PK_S(Na,C)
2. S -> A: PK_C,Enc_PK_A(Sig_S(Na,A))
3. A -> C: Enc_PK_C(Na,A,C,Sig_S(Na,A))
4. C -> A: Nc,{Na}kab
5. A -> C: {Nc}kab
```

Now C attacks A and B by impersonating A. Thus, we will call C now C_A. It reuses the old nonce Na and signature from the server that it received in the 3rd message. C_A knows the public key PK_B of B (if not, it could ask S and receives it in protocol step 2).

```
3. C_A -> B: Enc_PK_B(Na,A,B,Sig_S(Na,A))
4. B -> C_A: Nb,{Na}kab
5. C_A -> B: {Nb}kab
```

=> B accepts C_A as A and C_A knows kab.

Task 2 Some questions

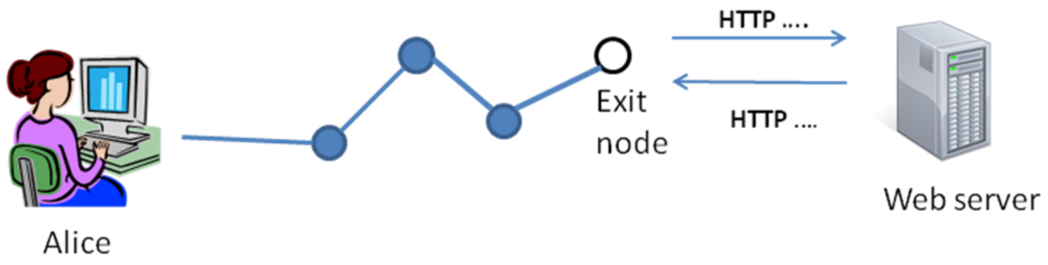
Answer the questions with knowledge from the lecture.

a) Cryptographic identities seem to make authentication a lot easier. Let assume, we use cryptographic identities. Do we still need a Certificate Authority? If yes, for what? If no, why not?

b) Why is trust important for key distribution?

c) Why does Zfone or SSH in the Baby Duck model not simply use a conventional authentication protocol like in SSL to authenticate the communication partners? What problem do they try to resolve?

The following graph sketches the situation for 2d):



d) Alice is using an anonymity system (like Tor) to access a web server. Lets assume the communication within the anonymity system along the dark thick lines is all-encrypted and highly secure and anonymous. To exit the anonymity network towards the normal Internet, exit nodes are used in such systems. The exit node (white node) operates as proxy that executes the HTTP requests to the web server for Alice.
Question: How can this so-called exit node attack Alice's private data or break her anonymity if she is not careful? (Hint: consider what the exit node can read)

Solution:

a)

Yes, since anyone can create a cryptographic identity (public key). It is not clear who (true identity) this entity is or what kind of access rights it has.

If it is sufficient to recognize a user again or to fight a man-in-the-middle attack between IDs, then a CA is not necessary.

b)

Since the knowledge of a key represents a way to identify entities and secure communication, one trusts that the other legitimate communication partner handles his keys in a responsible way.
If external entities like TTP or CA are used, one needs to trust their operation, that they send and forward the correct information. A CA or TTP is also used to connect key and identity for an entity. This can only happen if the entity can trust into the operation and accurateness of the CA or TTP. If this is not the case, one has to assume that maybe the key of the wrong person is sent or that confidentiality is not ensured.

c)

They assume that there is no global trustworthy TTP or CA, at least not one that they can use in all cases or for all application. Since security is almost impossible for the first contact without external help of CA or TTP, these protocols try to resolve the corresponding problems with other methods.

d)

The exit node will see all communication with the web server in cleartext if HTTP is used. Personal data posted to the server or even unencrypted passwords (not uncommon in forums) would be leaked.

If HTTPS is used, the exit node would not see the traffic and, thus, this problem would be resolved, yet using HTTPS depends on the website, not primarily on the user. The exit node could also try to stage a man-in-the-middle attack on the protocols, which would succeed if either the users accepts the wrong certificate or the exit node also controls a CA that is considered trusted in the root store of the browser (list of trusted CAs).

Task 3 Eclipse Attack on Chord

Assume that you want to attack a node in Chord and eclipse it from the rest of the network. You have as many resources as you like, but significantly less than 50 % of all nodes.

a) What do you have to do to be able to intercept all of his outgoing messages to other nodes? (Eclipse the outgoing links)
b) What do you have to do to prevent packets towards the node reach the node? (Eclipse ingoing links)

Solution:

a) Position Sybil nodes in 2^i distance
b) Position as predecessor

Task 4 Eclipse Attack on Kademlia

Assume that you want to attack a node in Kademlia and eclipse it from the rest of the network. This is a bit harder than in Chord and will most likely be less perfect. You have as many resources as you like, but significantly less than 50 % of all nodes.

a) What do you have to do to be able to intercept his outgoing messages to other nodes? (Eclipse the outgoing links)
b) What do you have to do to prevent packets towards the node reach the node? (Eclipse ingoing links)

Solution:

a)

In Kademlia, you make it into the bucket of a node, if you talk with it. Thus, as attacker you have to do lookups and other requests to be added. Here, you contact the victim with many Sybil nodes from various ID ranges to take over the buckets of the victim. Your nodes have to be long-lived as only unresponsive nodes are replaced by newer nodes.

b)

The same strategy as in a), yet you do this with other nodes. Here, you present a lot of Sybil nodes from the ID range of the victim, so that the other nodes will store your nodes into their buckets.

Task 5 Bootstrap Tree and Social Network Graph

In the lecture we discussed defences against Sybil and Eclipse attack based on the bootstrap graph and based on social network graphs. Brief answers are sufficient.

a) Why is it not possible to *only* use the bootstrap graph to route to a certain ID (node with certain ID)?
b) Why is it not possible to *only* use the social network graph to route to a certain ID (node with certain ID)?
c) How could you use either the bootstrap graph or the social network graph in your normal DHT routing to defend against routing attacks?

Solution:

a)

because the bootstrap graph does not tell you where an ID is, because it is not ordered or built according to these Ids. So, you would have to search in all subtrees to find a node with an ID. To find the closted node to an ID, you would have to look for all nodes and then select the best node.

b)

The same argument as in a) also holds for the social network graph.

c)

Bootstrap graph: Do iterative routing and e.g. every second node that is asked for better nodes to the target is not a closer node according to the routing, but a node from the bootstrap graph. The nodes selected in the bootstrap graph are randomly selected from different subtrees, parent nodes are also considered to be more trusted and selected from time to time.

Social network graph: Only add non-suspicious nodes in your routing table as preventive measure. If you do iterative routing you can apply a similar strategy as in the bootstrap graph case, ask the next node according to the routing, then ask a neighbour node in the social graph, then the best according to the routing, then again a node from the social graph... for better next hops to the target.