Colin Murray
UTEID: cdm2697
UTCS-username: tashar
Email: murray.colin43@gmail.com

CS361 Questions: Week 4

The questions marked with a dagger (†) require external research and may be more extensive and time consuming. You don't have to do them for the assignment but, but do them to increase your competency in the class.

Lecture 53

1. Why is it important for a digital signature to be non-reusable?

If signatures could be reused for any sort of message an attacker might eavesdrop and capture a valid signature, then proceed to use it to sign all of their malicious messages as if it came from the signer (who is presumed trustworthy).

2. Why is it the hash of the message typically signed, rather than the message itself?

The message may be tampered with by some attacker, but the attacker wouldn't be able to recomputed the hash for the tampered with message because the hash has been signed by the sender in a way that the attacker cannot reproduce. It's also probably more expensive to sign a very large message than simply sign its hash.

3. What assurance does R gain from the interchange on slide 4?

The message is: Unforgeable (only S can use K_s^{-1}), Authentic (anyone can verify using K_s), non-repudiation (only S can use K_s^{-1}), tamperproof (only R can remove outer layer of encryption), and not-reusable (signature is tightly bound to the message M).

Lecture 54

1. What is the importance of certificate authorities?

When attempting to establish some trust relation with an unknown vendor (or website) a CA that is trusted might work as a 3rd party to vouch for the legitimacy of this unknown party. Presumably it would be hard for the unknown party to receive the 3rd party's blessing if they were untrustworthy. In the context of network security, 3rd party CA's sign "certificates" that include individual web-pages public keys. This hopefully assures the authenticity and true identity of certain parties.

2. In the example on slide 5, why does X sign the hash of the first message with its private key?

X signs the hash with his own private key as a means of verifying to all those who possess X's public key that the hashed message contained is endorsed by X. If it can be legitimately decrypted using X's public key, then someone receiving Y's certificate can recomputed the hash

of the plaintext Y and K_Y and compare the result to the value of $\{\{h(\{Y, K_Y\})\}_{Kx}^{-1}\}_{Kx}$ = H({Y, Ky}).

3. Why is it necessary to have a hash of Y and K_y?

The hash guarantees message integrity. If an attacker intercepted the certificate and decided to put his own public key in place of K_y the message would hash to a different value, tipping off the recipient of the altered certificate that it was tampered with. It also prevents accidental corruption of the certificate to be noticed before Y's possibly corrupted public key is used for encryption.

4. What would happen if Z had a public key for X, but it was not trustworthy?

Z would be able to decrypt and verify the certificate of Y, but Z would not be able to impersonate Y or X or claim to have been signed by X. Since Z does not know X's private key, Z cannot forge a certificate under his own name as if it were signed by X.

Lecture 55

1. What happens at the root of a chain of trust?

At the root is a widely known and trusted entity. Ideally any individual who may be interested in verifying certificates should already know for certain the identity and public key of these root CAs. If a certificate chain originates at a trusted root, the entire chain is assumed to be as trustworthy as the root.

2. Why does an X.509 certificate include a "validity interval"?

It is important that once an entity gets a certificate they cannot use it forever. Suppose an entity were to obtain a certificate and immediately begin abusing their user's trust. The next time they hope to renew their certificate (if it hasn't already been revoke) they might have more trouble getting the CA to vouch for them again. This is also important for obsolete entities that have received valid certificates in the past but no longer use them. It is also important in the case that the private key associated with the public key in the signature is somehow leaked to some attacker, this attacker can only use this certificate as long as the validity interval is still good, at which point they're out of luck.

3. What would it mean if the hash and the received value did not match?

It would either indicate the certificate was tampered with or somehow corrupted in transmission. Either way the data within it shouldn't be trusted.

Lecture 56

1. What are some protocols previously discussed?

RSA, Diffie Hellman

2. What may happen if one step of a protocol is ignored?

The protocol should halt or terminate. Suppose A and B intend to perform a Diffie Hellman exchange but B never sends back g^b mod p. The process halts here, neither party can proceed without B's participation.

3. Why must the ciphers commute in order to accomplish the task in slide 4?

Commutative ciphers have no sense of order through which any of the encryptions are applied. No matter how many other secret keys are applied on top of a single one, that single key can be removed without removing the layers on top of it. This in effect creates a situation analogous to the lock box thought experiment.

4. Describe how an attacker can extract M from the protocol in slide 6.

The attacker must simply xor message (1) with message (2) and xor that with message (3).
(M xor Ka) xor ((M xor Ka) xor Kb) = Kb -> Kb xor ((M xor Ka) xor Kb) xor Ka) = M

5. Describe how an attacker can extract K_a from the protocol in slide 6.

The attacker must xor message (2) with message (3)
((M xor Ka) xor Kb) xor (((M xor Ka) xor Kb) xor Ka) = Ka

6. Describe how an attacker can extract K_b from the protocol in slide 6.

The attacker must xor message (1) with message (2)
(M xor Ka) xor ((M xor Ka) xor Kb) = Kb

7. Why are cryptographic protocols difficult to design and easy to get wrong?

In a real world sense the lockbox analogy works well when describing the xor padding scheme used above. The lockbox method is secure, but the seemingly well implemented version using xor is not secure due to the canceling nature of xor. Designing a secure protocol can be extremely difficult due to the vast number or possible ways these things can be abused or go wrong.

Lecture 57

1. Explain the importance of protocols in the context of the internet.

In a vast distributed network like the internet it is important to have some agreed upon mechanisms in place to rout information from one location to another (or many others). Without standard protocols information wouldn't get anywhere.

2. Explain the importance of cryptographic protocols in the context of the internet.

The internet is an insecure, entirely public place. Eaves droppers could be anywhere and everywhere. Without good cryptographic protocols that combat eavesdroppers or even active network attackers (those that control all network packet contents being sent to you) there could be no effective means to securely communicate sensitive information across the internet (no online banking, no password protection, no private email, ect.).

3. What are the assumptions of the protocol in slide 6?

A and B are assumed to know each other's public key's already.

4. What are the goals of the protocol in slide 6?

The goals are Unicity (only A and B know the secret K), Authenticity (A and B know they're both who they say they are) and confidentiality (K is inaccessible to an eavesdropper)

5. Are the goals of the protocol in slide 6 satisfied? Explain.

Not necessarily, anyone can claim to be A or B and provide their own public key. There must be some means to verify the authenticity of a public key. There is also a potential flaw as mentioned below that could break the confidentiality of the message.

6. How is the protocol in slide 6 flawed?

Say an attacker were to eavesdrop on the first message $\{\{K\}_{Ka^{-1}}\}_{Kb} = K_1$ and use this as the shared key by establishing a new run of the protocol with B. B will receive the message:
C -> B : $\{\{K_1\}_{Ke^{-1}}\}_{Kb} = \{\{\{\{K\}_{Ka^{-1}}\}_{Kb}\}_{Ke^{-1}}\}_{Kb}$
B will respond with this message:
B -> C : $\{\{K_1\}_{Kb^{-1}}\}_{Ke} = \{\{\{\{K\}_{Ka^{-1}}\}_{Kb}\}_{Kb^{-1}}\}_{Ke} = \{\{K\}_{Ka^{-1}}\}_{Ke}$
E can now decrypt B's response using their private key and A's public key in order to steal the shared secret K being used between A and B.

Lecture 58

1. Why is it important to know if a protocol includes unnecessary steps or messages?

One reason is that protocols are intended to be as efficient as possible. Another might be that every message in a protocol should be necessary for the other party to receive in order for the protocol to proceed. If optional steps were allowed, one party might move forward with the protocol without receiving an optional message (perhaps due to a dropped packet) with potential consequences.

2. Why is it important to know if a protocol encrypts items that could be sent in the clear?

Encryption is an expensive operation to perform unnecessarily. Some items are also required to be sent in the clear, like the beginning messages in an SSL handshake, otherwise one party might not have sufficient information to begin decryption.

Lecture 59

1. Why might it be difficult to answer what constitutes an attack on a cryptographic protocol?

There may be many different goals from simply being disruptive to attempting to steal information behind an attack. The attacker may also utilize a variety of methods to mount an attack, even methods that are unknown at the time the protocol is designed. As such it can be very difficult to quantify all the methods of attack for any given protocol.

2. Describe potential dangers of a replay attack.

If any sort of secret establishing or authentication messages could be replayed on a later session it could be used to impersonate another user or compromise another user's session.

3. Are there attacks where an attacker gains no secret information? Explain.

The attacker's goal may be to simply inflict consequences on communicating parties, perhaps by disrupting or halting their communication. Availability as discussed in previous modules can be very valuable.

4. What restrictions are imposed on the attacker?

Very few, if any at all. A good protocol should strive to maintain resilience against even the most resourceful and determined attackers. Since attackers like this could indeed exists, any protocol should be designed assuming the attacker has very few restrictions.

5. Why is it important that protocols are asynchronous?

In a distributed system there is no way of communicating that a protocol is about to be initiated with all parties involved. The sender just sends the initial message of the protocol and hopes for a response. There are no guarantees any message will get to its destination so any party involved cannot hope to assume it knows any more information than the messages received and messages sent so far. The medium of distributed networks like the internet are simply far too unreliable for good synchronization.

Lecture 60

1. Would the Needham-Schroeder protocol work without nonces?

Without nonces there would be the potential of replay attacks from earlier sessions.

2. For each step of the NS protocol, answer the two questions on slide 5.
    1. A is communicating to the key server S that A wants to communicate with B. The receiver S should have no concern whether the message came from A or not, there is no way to tell for sure. This is not a problem because of the next step.
    2. S creates a session key K_{ab} for A to communicate with B and includes another copy of this key in a message encrypted with B's secret key specifically for A to send to B (as well as telling B of A's identity in a encrypted message that A could not have forged or tamper with). S also affirms to

A that B is indeed the subject A indented to communicate with and encrypts the entire thing (including A's nonce) with A's secret key so only A can unpack the message. Upon receiving this from S, A knows which secret key to use with B, that S was who they said they were (both by being able to decrypt S's message which only S could have encrypted, and the inclusion of A's nonce which also prevents replay of S's message (so it must have been S responding to the earlier message)). A also has a means to communicate the secret session key with B without any further computation on A's part.

3. A sends the message from S that S specifically made for B that will prove to B A's identity (as endorsed by S) as well as communicating the secret key B will use for the later session. Upon receiving this B knows the packet came from S (only S knows K_{bs}) and B can verify that A is indeed the one who S permitted to communicate with B (A's identity being included in S's message so that B can verify it against who sent the message to him).
4. B uses K_{ab} to send a nonce to A in order to prove that A can decrypt it and return the Nonce altered in a predictable way. Upon receiving the message, A assumes it's from B and tries decrypting the message using K_{ab}. A does not know what the nonce should look like but will send back the nonce in the next message anyway.
5. A sends the nonce back in a predictably slightly altered way (so the message isn't identical to the one sent by B in step 4). B upon receiving this message can be certain by decrypting it that A shares the same secret key K_{ab} and can now trust A knowing A is who they claim to be and S must have given A permission to communicate with B.

Lecture 61

1. As in slide 5, if A's key were later changed, after having K_{as} compromised, how could A still be impersonated?

No, the attacker only knows the old K_{as} and as such would no longer be able to decrypt S's response in step 2.

2. Is it fair to ask the question of a key being broken?

It depends on the strength of the encryption, though it should be considered in the design of any protocol. It may be worthwhile to design a protocol in a way such that if a key is broken the damage potential is mitigated, or if the cryptographic algorithm being used is powerful enough perhaps the risk of the key being broken is a lesser concern. Kerberos is a good example of utilizing a long-term key as little as possible to protect it and switching to short term keys as soon as possible which are infeasible to break since they only last a couple minutes (and if they are broken, little damage could be done before they expire).

3. How might you address these flaws if you were the protocol designer?

The weakest link in the protocol should be strengthened, in this case a stronger means of encryption should be used to compensate for the weakness of vulnerable keys. It might also be a good idea for S to include a time-stamp inside message 3 in Needham-Schroeder to help prevent replays after K_{ab} is broken. B should notice the decrypted $\{K_{ab}, A, T\}_{Kbs}$ includes a timestamp perhaps several weeks old and may reject it.

Lecture 62

1. What guarantees does Otway-Rees seem to provide to A and B?

A and B are authenticated by S and receive a shared secret key from S that can be decrypted by each individually (without any other party being able to learn K_{ab}). It also assures against replay attacks from earlier sessions using nonces and prevents A tampering with any messages from S to B and B tampering with any messages from S to A.

2. Are there guarantees that Needham-Schroeder provides that Otway-Rees does not or vice versa?

Needham-Schroeder provides protection from replay attacks within a single key establishment protocol. It also assures that A and B both know they ended up with the same shared keys (steps 4 and 5 in N-S).

3. How could you fix the flawed protocol from slide 4?

B responds to A simply to verify that they now are sharing a public key. A nonce could be used to prevent the vulnerability like so:
A -> B : $\{\{K, N_a\}_{Ka}^{-1}\}_{Kb}$
B -> A : $\{N_a\}_K$
B acknowledges to A that the secret key K was received. A can confirm by decrypting B's response and comparing the nonce with the one sent.

Lecture 63

1. Why is the verification of protocols important?

There can be many very subtle flaws in protocols when they are designed. Some protocols are used for years before a flaw is realized. It is very important that the design of a protocol achieves the security goals the design intends to guaranteed without leaving vulnerabilities and without relying on unsafe assumptions.

2. What is a belief logic?

They are used to reason about what information a subject may be able to infer based on the messages they see in a protocol. They use a formal system for reasoning beliefs by applying logical operations and rules of inference.

3. A protocol is a program; where do you think beliefs come in?

Beliefs could come into play if an attacker intercepts enough information to form a belief that may or may not be useful in mounting an attack. It also examining the messages can point out what a participating subject may believe and if this belief is sufficient for a strong protocol.

Lecture 64

1. What is a modal logic?

It is a type of formal logic that uses operators and predicates to express “modalities” or statements.

2. Explain the intuition behind the message meaning inference rule.

Given A believes that A and B share a key, and A receives a message encrypted with that key, A is entitled to believe that that message must have come from B. Essentially this suggests that if A receives any message encrypted with a shared key, and A knows it shares that key with B, A can assume B sent the message.

3. Explain the intuition behind the nonce verification inference rule.

If A believes that X is fresh and A believes B once said X, then A believes B believes X. This is to say that if A receives a message from B that is “fresh”, that A has never encountered before, A can assume that B believes whatever was sent in the message. This is opposed to B not believing the message (or that he sent it), say if X was a replay from an earlier exchange.

4. Explain the intuition behind the jurisdiction inference rule.

If A believes B is an expert over X and A believes B believes X, then A must believe X. This is simply suggesting that if A sees that B knows everything about X and truly believes X to be the case, A will feel safe in also believing X and take B’s word for it.

5. What is idealization and why is it needed?

Idealization attempts to turn messages sent in a protocol into logical belief inferences. Each message is decomposed into its essential purpose of communicating to the receiver a certain belief and/or causing the receiver believe a certain thing.

Lecture 65

1. Why do you think plaintext is omitted in a BAN idealization?

A plaintext message could come from anything and anywhere, it’s difficult to draw any beliefs from something that could easily be forged.

2. Some idealized steps seem to refer to beliefs that will happen later in the protocol. Why would that be?

There are assumptions and likewise beliefs that can be formed early on in the protocol from messages received that will not come into play until later in the protocol’s actual step by step procedure.

3. One benefit of a BAN proof is that it exposes assumptions. Explain that.

It can be revealing to formally expose the set of assumptions required when looking into a protocol. Assumptions can be a dangerous point of weakness so it is important to formally list out every assumption being made at every step of the process and make sure that these assumptions aren't problematic.