Name: Jessica Lucci
EID: jml3624
CS Login: jlucci
Email: jessicalucci14@gmail.com

Lecture 53

1. If a signature was reusable, it could be taken from the intended document by a malicious party and put (or reused) on a non-authorized document.
2. The hash of a message is typically signed instead of the message itself, as public key encryption is expensive (and the message may be arbitrarily long).
3. R gains both confidentiality (only R can decode the outer message) and integrity (only S's public key can decode the inner message) assurances from the exchange. R also gains all assurances associated with a signature - the message is unforgeable, authentic, tamperproof, non repudiable and non reusable.

Lecture 54

1. Certificate authorities act as sort of "third party authorizers", vouching for specific companies/certificates for people who have never encountered said companies/certificates before.
2. X signs the hash of the message with his private key to act as a certificate authority for Y's identity - by using his private key anyone with access to his public key can decode the message. It is in this manner that X acts as a certificate authority.
3. There must be a hash of Y and K_Y so that K_Y can be identified as Y's public key.
4. If Z had a public key for X that wasn't trustworthy, Z wouldn't be able to verify X's signature. This means that Z wouldn't be able to trust Y's certificate (since X signed it).

Lecture 55

1. A chain of trust is rooted at some sort of unimpeachable authority.
2. X.509 certificates include a validity interval, as sometimes certificates may only be authorized for certain time periods. For example, company X may have to renew its certificate yearly to make sure it is still complying with given security standards.
3. If the hash and the received value didn't match, this would mean that the received value was modified from its original, intended value.

Lecture 56

1. Symmetric authentication and key agreement are some protocols previously discussed.
2. If one step of the protocol is ignored the information may become compromised, or inaccessible to the intended parties.

3. The ciphers must be able to commute to accomplish the task on slide 4, because the person trying to get the message on the inside wouldn't be able to decipher the outer key (unlock the outer box) otherwise.
4. An attacker could extract M by computing:
    a. $Ka = ((M \oplus Ka) \oplus Kb) \oplus (((M \oplus Ka) \oplus Kb) \oplus Ka))$ = (Step 2 $\oplus$ Step 3)
    b. $M = ((M \oplus Ka)) \oplus Ka$ = (Step 2 $\oplus$ Step 3) $\oplus$ Step 1
5. An attacker could extract K_a by computing:
    a. $K_a = ((M \oplus Ka) \oplus Kb) \oplus (((M \oplus Ka) \oplus Kb) \oplus Ka))$ = (Step 2 $\oplus$ Step 3)
6. An attacker could extract K_b by computing:
    a. $K_b = ((M \oplus Ka)) \oplus ((M \oplus Ka) \oplus Kb)$ = (Step 1 $\oplus$ Step 2)
7. Cryptographic protocols are difficult to design because they must account for all types of malicious attacks (on integrity, confidentiality, etc). Cryptographic protocols are easy to get wrong for the same reason they're so difficult to design. One small oversight could break even the most clever protocols.

Lecture 57

1. Almost every interaction on the internet occurs via some sort of protocol. From HTTP to IP, there's a massive protocol suite that defines how the internet works.
2. Cryptographic protocols are important in making sure all those interactions that occur on the internet are secure.
3. The protocol in slide 6 assumes that there is a public key infrastructure in place, and both A and B have reliable public keys.
4. The goals of protocol in slide 6 are that each party has access to the passed key and both parties are authenticated to each other.
5. The goals of the protocol on slide 6 are satisfied, as both parties can access the passed key, and can authenticate which party sent the key.
6. The protocol is flawed, because C can essentially trick either A or B into giving them the key - assume C has intercepted $\{\{K\}K_{a-1}\}K_b$, and call this K^1. Then, C starts a new initiation of the protocol with B - $C \rightarrow B$: $\{\{K^1\}K_{c-1}\}K_b$, to which B responds with $B \rightarrow C$: $\{\{K^1\}K_{b-1}\}K_c$. By doing this, B has unlocked the inner key for C, since $\{\{K^1\}K_{b-1}\}K_c = \{\{\{K\}K_{a-1}\}K_b\}K_{b-1}\}K_c$. C uses it's private key to unlock the outer message, leaving $\{\{\{K\}K_{a-1}\}K_b\}K_{b-1}\}$, then B's private and public keys cancel out leaving $\{\{K\}K_{a-1}\}$, and finally, C uses A's public key to unlock the last message, exposing the original key.

Lecture 58

1. If a protocol includes unnecessary steps or messages, the protocol could be exposing unnecessary security risks and/or causing the protocol to run inefficiently.
2. If a protocol encrypts items that could be sent in the clear, the protocol is expending unnecessary resources, and is running inefficiently.

Lecture 59

1. It is difficult to define what constitutes an attack on cryptographic protocol, as some things that may jeopardize one protocol may not be malignant to another, and there's a vast amount of different types of attacks.
2. A replay attack could potentially confuse the parties participating in the protocol/exchange, causing them to lose information, expose information, etc.
3. There attacks in which the attacker gains no secret information, but rather just disrupts or stops an exchange.
4. It is assumed that an attacker cannot interject arbitrary messages.
5. If a protocol were synchronous, an attacker would easily be able to determine the steps and in turn, what step is currently executing within the protocol.

Lecture 60

1. The Needham-Schroeder protocol would not work without nonces, because the principals would have no way of determining how "fresh" a message is.
2. Steps:
    a. $A \rightarrow S : A,B, N_a$
        i. A is requesting a key from S to talk to B with, and is also sending nonce N_a for their communication session.
        ii. S knows that A wants to talk to B, and that N_a is the nonce to use for this session.
    b. $S \rightarrow A : \{N_a,B,K_{ab}, \{K_{ab}, A\}K_{bs}\}K_{as}$
        i. S is sending new public key K_{ab} to be used for A to B communications, the nonce to indicate this message is "fresh", and a message containing A's identity and the newly generated public key encrypted with B's server key (K_{bs}).
        ii. A now knows that it has a fresh, secure key to use to communicate with B, as well as a secure message to send to B that includes A's identity, and the key K_{ab} to be used for A to B communications.
    c. $A \rightarrow B : \{K_{ab},A\}K_{bs}$
        i. A sends a message that only B can decode that includes A's identity and the key K_{ab} to be used for A to B communications.
        ii. B knows that it has received a message from A that includes the key K_{ab} for A to B communications.
    d. $B \rightarrow A : \{N_b\}K_{ab}$
        i. B sends a message encrypted with K_{ab} to A with a new nonce, N_b in order to acknowledge it has received A's message.
        ii. A knows that B has received the key, and that it can use it.
    e. $A \rightarrow B : \{N_b - 1\}K_{ab}$

i. A sends to be a message encrypted with K_{ab} to B with B's nonce N_b - 1.
ii. B now knows that A also has access and use of K_{ab}.

Lecture 61

1. An impersonator could send a message to S using the compromised key K_{as} to establish secure communication with any other party.
2. It's fair to ask the question of the key being broken if the encryption of said key is very weak. Almost all cryptographic protocols assume the keys being used are secure, so unless the protocol expects the key to be of low reliability, a broken key would not be a flaw in the protocol.
3. In order to address these flaws, I would put in an additional step in the protocol that makes A identify itself to B, and B to A without the intermediate of S.

Lecture 62

1. Otway-Rees protocol seems to assure A and B of each other's identities, safe communication between A and B, and that all messages are happening within session M.
2. Otway-Rees provides the guarantee that all messages are occurring within session *M*, while Needham-Schroeder provides the guarantee that both A and B can use K_{ab}.
3. You could fix the protocol by preventing encryption and decryption steps from being next to each other - for example, between encrypt and decrypt, an extra public key could be added in.

Lecture 63

1. If protocols aren't verified, possible holes within that protocol may exist that jeopardize the security of the parties using said protocol.
2. Belief logic is a formal system that allows reasoning about what principles should be able to infer from the messages they can see.
3. Beliefs are the boolean logic of programs - if X then Y.

Lecture 64

1. Modal logic is a type of logic that essentially extends propositional/predicate logic to include modality operators (usually, sometimes, etc.).
2. The intuition behind the message meaning inference rule is that if A and B are sharing a key that is unique to them, then if either of them receives a message encoded with that key, then they can safely assume it came from the opposite party (since no one else would be using that key).
3. The intuition behind the nonce verification inference rule is that if a principal sends a verifiably fresh message (or "says" a message) then the sender of that message believes their message is valid.

4. The intuition behind the jurisdiction inference rule is that if a principal has jurisdiction, or ownership, of some object, and they believe its' true, then everyone else should believe that object is true as well.
5. Idealization is the process of translating protocol steps into logical inferences, and is needed because omits unnecessary or extraneous steps of the protocol (parts of the message that don't contribute to the beliefs of the recipient).

Lecture 65

1. Plaintext is omitted from BAN idealization, as plaintext doesn't contribute to the beliefs of any recipient.
2. Some idealized steps seem to refer to beliefs that occur later in the protocol, because those beliefs are *assumed* to be true at those points in the protocol.
3. BAN proofs expose assumptions, meaning that they expose things that are believed to be true within the protocol, but actually have not been proven to be true.