CS 458 Information Security

# Homework 4: Key Management and Identity

## 1. Trusted Third Party

(a) A trusted third party solves what problem in key management?
Number of keys grows quadratically with the number of parties.

(b) Assuming the third party is indeed trustworthy (i.e. not malicious, e.g. the auth center in a corporate network), what attacks can compromise the key exchange?

Eavesdropping then replaying
Man-in-the-middle

(c) **Bonus**: In the Needham-Schroeder protocol, what are "$r_1$" and "$r_2$" called? What attack do they prevent?

Nonces. Replay attacks.

## 2. Digital Signatures

(a) Bob received a 128-bit AES key and the message "from Alice: use this key to send me your credit card number", both encrypted with his public key. Should he do what the message says? Assume Bob does want to send Alice his credit card number.

No. There's no way to confirm the message actually came from Alice. Anyone could have encrypted such a message with Bob's public key.
* The wording of this problem is lengthy and awkward to avoid stating "Alice sent the message".

(b) We discussed "meet-in-the-middle" attack last week. We'll talk about "man-in-the-middle" attack this week. They are completely different concepts, despite their similar names. Explain what each attack is.
**Meet-in-the-middle**: this was introduced to explain why 2DES was insecure; in general, a meet-in-the-middle attack occurs when we can split the exhaustive search for keys into two independent, smaller searches (trading space for time).

**Man-in-the-middle**: this is describing an active attacker that does more than eavesdropping, and actually tampers with the traffic. A (wo)man-in-the-middle intercepts messages and replaces them with fraudulent ones of his/her choice.

(c) Digital signatures bind identity to the message.

(d) How are digital signatures different from MACs?

MACs verify the integrity of the message (i.e. it has not been tempered with). When used correctly, MACs together with ciphers provide authenticated encryption (both confidentiality and integrity). Still, you can't be sure who you're talking to unless the message comes with a digital signature (you could be receiving untampered messages from an attacker).

### 3. Web of Trust

(a) When you visit a website whose URL starts with "https://", what protocol are you using?

SSL/TLS. TLS is the new SSL.

(b) When you click on the lock in the address bar on a secure website, it shows you a certificate. What does a certificate do?

It confirms you're actually talking to Google and not a malicious site pretending to be Google.

(c) When you go to your bank's website and your browser prompts "this website cannot be trusted", what just happened?

Your browser is trying to connect with SSL/TLS but the certificate cannot confirm the site you're visiting is your bank. Possible causes include (*not* exhaustive):

a. The certificate expired
b. The domain name you're trying to visit doesn't match what's on the certificate
c. You're talking to a man in the middle
d. The certificate is self-signed ("Hi, my name is bank; nice to meet you.")
e. There's no certificate at all

(d) **Bonus**: in practice, who decides which root certificates are trustworthy?

Your browser. Each browser ships with a list of root certificates they trust, and the chain of trust starts there.

(e) **Bonus**: can you create a certificate for google.com?

Sadly, yes. Managing the web of trust is a very delicate issue with many non-technical aspects.