Announcements

- Problem Set #5 due Wednesday
- Graded Exam 1 will be posted this afternoon
  Solutions posted as well
- Next exam is in two weeks
- In “TR Announcements” forum, list of topics we’ll fill in next semester
- FYI, Christof Paar’s lectures for his course are on YouTube

https://www.youtube.com/channel/UC1usFRN4LCMcfIV7UjHNuQg/videos

Few points on RSA

- RSA encryption is deterministic: Given plaintext x and public key $k_{pub} = (n, e)$, the ciphertext y is always the same
- AES addresses this using IV and block chaining
- Can address in RSA with padding (cf. Section 7.7 of text)
  - Essentially adds random bits and a fixed string to plaintext *before* encrypting
  - Adds perturbation so that same plaintext will encrypt to different values
  - Can also provide check on errors in transmission when decrypting
- RSA keys are large
  Next semester we'll see elliptic curve public key encryption that has same security as RSA with much smaller keys

Why hash functions?

- Message signed in RSA digital signature cannot be longer than Alice's modulus n
- Often want to sign *much* larger messages x (e.g. digital media)
- Solution is to sign the *hash* of x, $h(x)$
- Idea is that $h(x)$ is much smaller than x

From pg. 303 of text

Properties of Hash Functions

1. **Arbitrary message size** $h(x)$ can be applied to messages x of any size.
2. **Fixed output length** $h(x)$ produces a hash value z of fixed length.
3. **Efficiency** $h(x)$ is relatively easy to compute.
4. **Preimage resistance** For a given output z, it is impossible to find any input x such that $h(x) = z$, i.e, $h(x)$ is one-way.
5. **Second preimage resistance** Given x_1, and thus $h(x_1)$, it is computationally infeasible to find any x_2 such that $h(x_1) = h(x_2)$.
6. **Collision resistance** It is computationally infeasible to find any pairs $x_1 \neq x_2$ such that $h(x_1) = h(x_2)$.

Downsides to using RSA for key exchange

Assume Alice generates AES key, uses Bob's public credentials to sign

- Alice has to assume Bob has been careful about keeping private key private
- Bob has to assume Alice generated random AES key
- Bob has to assume Alice hasn't used same AES key elsewhere

Diffie-Hellman Key Exchange

A trusted source publishes p and α that Alice and Bob use
e.g. https://tools.ietf.org/html/rfc3526
Note [ ] represents the floor function

- Alice picks private a, sends Bob $A = \alpha^a \mod p$
- Bob picks private b, sends Alice $B = \alpha^b \mod p$
- Shared key is $k_{AB} = A^b \equiv B^a \mod p$