Cryptography Roadmap

Cryptology

Cryptography

Symmetric Ciphers

Asymmetric Ciphers

Utilities (Hash Functions, MACs, etc.)

Apps/Protocols (TLS, ToR, etc.)

Cryptanalysis (Attacks)
Sections of this Lecture

• Digital Signatures
• Cryptographic Hash Functions
• Passwords
• WannaCry
DIGITAL SIGNATURES
Digital Signing

Plain Text → Asymmetric Cipher → Cipher Text

Cipher Text → Asymmetric Cipher → Plain Text

Private Key

Public Key
RSA

Plain Text $\rightarrow$ (plain text)$^d \mod n$ $\rightarrow$ Digital Signature

Digital Signature $\rightarrow$ (cipher text)$^e \mod n$ $\rightarrow$ Plain Text

Private Key $\{d, n\}$

Public key $\{e, n\}$
How do I digitally sign a large file?

- Naive solution: split the message in parts and sign each one of them

(1) Signature has the length of the message
(2) Computational overhead (several RSA computations)
(3) Security limitations
Cryptographic Hash Functions

**Ideally**: If message 1 and message 2 differ by one bit, then A and B differ in 50% of their bits.
Digital Signing

Verifying Digital Signatures

- Document (Arbitrary Size + signature)
- Message Signature

- Cryptographic Hash Function

- Document Hash Key

- Public-Key Cryptography (RSA)
- Public Key

- Message Signature

- Document Hash Key
CRYPTOGRAPHIC HASH FUNCTIONS
Merkle-Damgård Construction (SHA1)

Message (arbitrary length)

\[ M_1, M_2, M_3, \ldots, M_N \text{ (blocks)} \]

(512 bits)

Compression Function

160 bits

Digest (160 bits)
High-level Properties

• Complicated **one-way** functions

• One-way
  – Hard to compute the message by having just the hash value (or *digest*)
  – No cryptographic keys
  – Should not be confused with invertible functions (1-1)

• Collision
  – Find a message that cryptographically hashes to a given digest $H$
Basic Requirements

\[ x = ? \]

\[ h(x) \] preimage resistance

\[ h(x_1) = h(x_2) \] second preimage resistance

\[ h(x_1) = h(x_2) \] collision resistance
Basic Requirements

• Preimage Resistance ("One way")
  – If you know just the digest it should be computationally hard to find a message M with the same digest

• Second Pre-image Resistance (input resistance)
  – Given a message M1, it should be computationally hard to find a second message, M2, with the same digest

• Collision resistance
  – It should be computationally hard to find any two messages, M1 and M2, with the same digest
Lifetimes of cryptographic hash functions

More: http://valerieaurora.org/hash.html

SHA256 is considered currently safe
PASSWORDS
Other Uses - Passwords

• Services
  – Store cryptographic hashes of passwords
  – Passwords in plaintext are deleted

• Authentication
  – Services compute and check cryptographic hashes and not plaintext passwords

• Encrypting passwords is a bad idea
  – Attacker can leak the key

• Passwords are *salted*
  – Identical plaintext passwords produce different hash keys
Attacking Passwords

• Brute force
• Dictionary attacks
• Rainbow tables
  – Salt can make this extremely hard
• GPUs
WANNACRY
1. Encrypt file with EncK (per-file encryption)
2. Encrypt EncK with Sub-PuK and store it to WannaCry Header (per-host encryption)
3. Encrypt Sub-PrK with PuK and send it to attacker (attacker has a different decryption key per host)

Read more: WannaKey, https://github.com/aguinet/wannakey
Resources

• This lecture was built using material that can be found at
  – Chapter 9, 11, Handbook of Applied Cryptography, http://cacr.uwaterloo.ca/hac/
  – Chapter 6, 10, Serious Cryptography, https://nostarch.com/seriouscrypto