Using Encryption for Authentication in Computer Networks

Robert Grimm
New York University
Altogether Now: The Three Questions

- What is the problem?
- What is new or different?
- What are the contributions and limitations?
Needham/Schroeder

- Early exploration (’78) of how to use encryption to provide authentication
  - Diffie/Hellman published their paper on public key cryptography only two years earlier
- Basis for Kerberos network authentication protocol
  - Specifically, the symmetric key protocol
- “Our protocols should be regarded as examples”
  - Rightly so, the protocols have known attacks!
Getting Our Concepts Right

- Assumptions
  - Computers are *secure*
    - I.e., when user encrypts a message, neither the plaintext nor the key is leaked outside the application
  - But the network is *not*
    - Attackers can arbitrarily read, insert, delete, or modify messages on the network

- End-to-end encryption
  - Encryption must be performed by applications, not at the network level
    - E.g., key may not be known by the network interface
Authentication servers (certificate authorities)

- Trusted by all participating users
  - For symmetric-key crypto, user $\rightarrow$ key
  - For public-key crypto, user $\rightarrow$ public key

- Not limited to a single server
  - Group of collaborating servers
  - Forest of servers (certification authority model)
  - No server: web of trust in PGP
Nonces and timestamps
  - Ensure that messages are unique
    - Interactive protocols → random number
    - Offline protocols → timestamp
  - Prevent *replay attacks*

Tickets and certificates
  - Tickets establish a *session key* (shared secret)
  - Certificates attest a public key
Characteristic functions
- Now: Collision resistant hash functions
- Three properties
  - $h(M)$ is relatively easy to compute (and typically small)
  - Given $h(M)$, it is hard to calculate $M$
  - It is hard to find two $M_1$ and $M_2$ so that $h(M_1)=h(M_2)$
Let’s Mount an Attack
[Lowe 95]
The Public Key Protocol

- **A** → **AS**: A, B
- **AS** → **A**: $\{PKB, B\}^{SKAS}$
- **A** → **B**: $\{NA, A\}^{PKB}$
- **B** → **AS**: B, A
- **AS** → **B**: $\{PKA, A\}^{SKAS}$
- **B** → **A**: $\{NA, NB\}^{PKA}$
- **A** → **B**: $\{NB\}^{PKB}$
There Really Are Two Protocols

- A → AS: A, B
- AS → A: \{PKB, B\}_{SKAS}
- A → B: \{NA, A\}_{PKB}
- B → AS: B, A
- AS → B: \{PKA, A\}_{SKAS}
- B → A: \{NA, NB\}_{PKA}
- A → B: \{NB\}_{PKB}

Obtain public keys

Authenticate A and B

Let’s focus on the authentication protocol
The Man-in-the-Middle Attack

- $A \rightarrow I$: \{NA, A\}^{PKI}
- $I(A) \rightarrow B$: \{NA, A\}^{PKB}
- $B \rightarrow I(A)$: \{NA, NB\}^{PKA}
- $I \rightarrow A$: \{NA, NB\}^{PKA}
- $A \rightarrow I$: \{NB\}^{PKI}
- $I(A) \rightarrow B$: \{NB\}^{PKB}

- How can we prevent this attack?
Let’s Improve Our Notation
The Four Primitives

- Encrypt(PK, M) $\rightarrow$ CT
- Decrypt(SK, CT) $\rightarrow$ M
- Sign(SK, M) $\rightarrow$ $\sigma$
- Verify(PK, M, $\sigma$) $\rightarrow$ \{true, false\}
What Did We Learn Today?