Diffe-Hellman, PKI, IPSec

Gaurav Naik, Gus Anderson
Outline

● Diffe-Hellman Key Exchange

● PKI Conceptual Implementation
  ● Goals
  ● RSA Review
  ● Example

● IPSec
Diffe Hellman Background

- First public-key type scheme proposed
- By Diffie & Hellman in 1976 along with the exposition of public key concepts
- Is a practical method for public exchange of a secret key
- Used in a number of commercial products
- Its beauty is in its simplicity
Diffe-Hellman Key Exchange

- A public-key distribution scheme
  - cannot be used to exchange an arbitrary message
  - rather it can establish a common key
  - known only to the two participants

- Value of key depends on the participants (and their private and public key information)

- Security relies on the difficulty of computing discrete logarithms (similar to factoring) – hard
Diffe-Hellman Key Exchange

- All users agree on global parameters:
  - $q$ is a large prime integer
  - $a$ being a primitive root mod $q$
    - $a$ is an integer less than $q$, and $\forall n \in (1, q - 1)$, there is a power $k$ of $a$ such that $n = a^k \mod q$.

- Each user (eg. A) generates their key
  - Chooses a secret key (number): $x_A < q$
  - Compute their public key: $y_A = a^{x_A} \mod q$

- Each user makes public that key $y_A$
Diffe-Hellman Key Exchange

- shared session key for users A & B is $K_{AB}$:
  
  $K_{AB} = a^{x_A \cdot x_B} \mod q$
  
  = $y_A^{x_B} \mod q$ (which B can compute)
  
  = $y_B^{x_A} \mod q$ (which A can compute)

- $K_{AB}$ is used as session key in private-key encryption scheme between Alice and Bob

- If Alice and Bob subsequently communicate, they will have the same key as before, unless they choose new public-keys

- Attacker needs an $x$, must solve discrete log
Diffe-Hellman Key Exchange

- Users Alice & Bob who wish to swap keys:
  - Agree on prime \( q=353 \) and \( a=3 \)

- Select random secret keys:
  - A chooses \( x_A=97 \), B chooses \( x_B=233 \)

- Compute respective public keys:
  - \( y_A=3^{97} \mod 353 = 40 \) (Alice)
  - \( y_B=3^{233} \mod 353 = 248 \) (Bob)

- Compute shared session key as:
  - \( K_{AB}= y_B^{x_A} \mod 353 = 248^{97} = 160 \) (Alice)
  - \( K_{AB}= y_A^{x_B} \mod 353 = 40^{233} = 160 \) (Bob)
Diffe-Hellman Key Exchange

- Users could create random private/public DH keys each time they communicate.

- Users could create a known private/public DH key and publish in a directory, then consulted and used to securely communicate with them.

- Both of these are vulnerable to a man-in-the-Middle Attack.

- Authentication of the keys is needed!
Outline

- Diffe-Hellman Key Exchange

- **PKI Conceptual Implementation**
  - Goals
  - RSA Review
  - Example

- IPSec
PKI Goals

- **Authentication**
  - Assurance of who sent the message

- **Confidentiality**
  - No unauthorized party has gained access to the message

- **Authenticity / Data Integrity**
  - What is received is what was sent

- **Non repudiation**
  - The sending party cannot deny that the message was sent by them
RSA again ...

- Bob chooses large prime numbers $p$ and $q$, $n=pq$
  - $e$ s.t. $\gcd(e, (p-1)(q-1))=1$
  - $d$ s.t. $ed = 1 \mod (p-1)(q-1)$

- Bob’s public key is $(n, e)$
- Bob’s private key is $d$

- Encryption function for sending a message to Bob
  - $E(M) = C = M^e \mod n$

- Bob’s Decryption function
  - $D(C) = M = C^d \mod n$

- Bob’s signature function
  - $S(M) = S = M^d \mod n$

- Signature Verification function
  - $V(S) = M = S^e \mod n$
Let’s work out the rest …

- **Authentication**
  - Assurance of who sent the message

- **Confidentiality**
  - No unauthorized party has gained access to the message

- **Authenticity / Data Integrity**
  - What is received is what was sent

- **Non repudiation**
  - The sending party cannot deny that the message was sent by them
Outline

- Diffe-Hellman Key Exchange

- PKI Conceptual Implementation
  - Goals
  - RSA Review
  - Example

- IPSec
IPSec

- Implementation of what we just worked through
IPSec: IP Security [RFC2401-12]

- Transport security at the IP (Internet Protocol) layer
- Goal: secure traffic between any two IP systems
  - Any device with an IP address: hosts, gateways, mobile devices, IP-enabled microwaves, …
- Security services for IP packets
  - encryption and authentication
- **SA (“Security Association”)** creation & management
- Application independent: security for the “Internet infrastructure”
Network Layers
Virtual Private Networks (VPN)
IPSec Processing Basics

- Two IP devices A and B want to communicate securely under the protection of IPSec

- First a Security Association (SA) between A and B is established
  - SA: a set of parameters, algs, & shared keys agreed between A and B, and locally stored by each party

- Then, A and B secure the IP traffic by applying ENC and MAC on each IP packet they exchange

- Omitted: many details, system issues, implementation, complexities, controversies, etc
IPSec Encapsulation Mechanisms

Plain IP packet

Encapsulated Security Payload (ESP)

ESP MAC-only

ESP-Tunnel Mode
ESP Format [RFC 2403]

- IP Header
- SPI
- Replay Prevention Sequence Number
- Initial Vector
- Payload
- Padding
- Pad Length
- Protocol
- MAC Value
- MAC
- Encrypted (padded) Payload
IPSec’s Crypto Algorithms

- Negotiable

- Default (for interoperability and common use)
  - Encryption: 3DES (moving to AES)
  - Integrity: HMAC (SHA1, MD5)

- Some crypto highlights:
  - HMAC developed for use in IPSec
    - the prepend key story: \( \text{MAC}_K(M) = \text{MD5}(K \mid M) \)
  - encrypt-then-authenticate (the “right order”) [Bellovin’96, K’01, CK’01]
IKE: Internet Key Exchange

- Creates SAs for use by IPSec
  - Negotiates security parameters for the SA
    - type of key exchange, credentials, crypto algorithms, crypto strength, traffic to protect, etc
  - Key Exchange: share keys between parties

- Manages SAs: create, refresh, maintain, delete
  - IKEv2 (2003): IKE specifies it all
IKE: Internet Key Exchange

- When A wants to talk to B under protection of IPSec, and they do not have an established SA:
  - A invokes IKE to signal B its request for an SA
  - IKE is run between A and B: result is a shared SA (services to be applied and fresh shared keys)
  - Negotiated parameters stored locally at A and B (SAD, SA Database)
  - SPI (sec. parameters index): pointer to SA included in the IPSec header of each packet

- Architectural separation: IKE writes to SAD, IPSec reads from SAD (full picture more involved: e.g. SPD)
The Cryptography of IKE

- Driving cryptographic requirements
  - Authenticated key exchange: public and symmetric keys
  - Perfect forward secrecy (PFS): exposure of long term keys does not compromise security of past sessions
    - Diffie-Hellman (optional for fast re-key functionality)
  - Identity protection: hiding parties identities from passive and/or active attackers
    - Logical identities (e.g. cert’s) vs. IP/physical addresses
IKEv1 [RFC2409]

- Several authenticated DH protocols supported. Differ in mode of authentication:
  - Long-term pre-shared (symmetric) key
  - Public-key encryption
  - Digital Signature
  - Re-key (with optional DH)

- With and without identity protection

- Modes designed to share as many elements as possible (e.g., auth’d info, nonces, key derivation)
IKEv1

- Many cryptographic elements taken from SKEME [K’95] and OAKLEY [Orman’98]
  - Uniform set of authentication modes
  - Key derivation
  - Authentication based on public-key encryption
  - But SKEME did not provide signature-based auth’n

- Signature mode specifically developed for IKE (the SIGMA protocol)
  - Replacement for Photuris’ signature-based DH which used an (insecure) variant of the STS protocol
IKEv2 (RFC to appear)

- Simplification of SA management spec

- Simplification of Key Exchange
  - Got rid of many of the authentication options: e.g., the PK Encryption and “aggressive” modes
  - Single signature mode: kept SIGMA design

- Added password-based authentication
  - Asymmetric setting [HK’99]

- Streamlined key derivation spec

- Added optional Denial-of-Service defense [Karn]