



Systems and Internet Infrastructure Security

Network and Security Research Center
Department of Computer Science and Engineering
Pennsylvania State University, University Park PA

CSE543 - Introduction to Computer and Network Security Module: Applied Cryptography

Professor Patrick McDaniel
Fall 2008

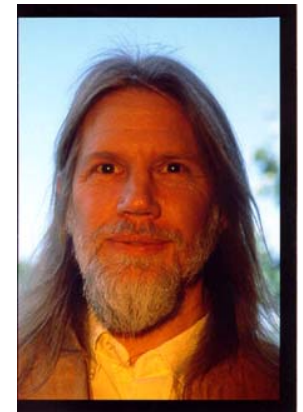
Key Distribution/Agreement



- **Key Distribution** is the process where we assign and transfer keys to a participant
 - ▶ Out of band (e.g., passwords, simple)
 - ▶ During authentication (e.g., Kerberos)
 - ▶ As part of communication (e.g., skip-encryption)
- **Key Agreement** is the process whereby two parties negotiate a key
 - ▶ 2 or more participants
- Typically, key distribution/agreement this occurs in conjunction with or after authentication.
 - ▶ However, many applications can pre-load keys

Diffie-Hellman Key

- The DH paper really started the modern age of cryptography, and indirectly the security community
 - ▶ Negotiate a secret over an insecure media
 - ▶ E.g., “in the clear” (seems impossible)
 - ▶ Idea: participants exchange intractable puzzles that can be solved easily with additional information.



- Mathematics are very deep
 - ▶ Working in multiplicative group G
 - ▶ Use the hardness of computing discrete logarithms in finite field to make secure

Diffie-Hellman Protocol

- For two participants p^1 and p^2
- Setup: We pick a prime number p and a base g ($< p$)
 - ▶ This information is public
 - ▶ E.g., $p=13$, $g=4$
- Step 1: Each principal picks a private value x ($< p-1$)
- Step 2: Each principal generates and communicates a new value

$$y = g^x \text{ mod } p$$

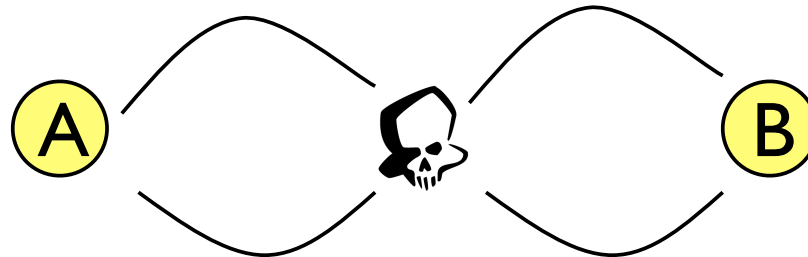
- Step 3: Each principal generates the secret shared key z

$$z = y^x \text{ mod } p$$

- *Perform a neighbor exchange.*

Attacks on Diffie-Hellman

- This is key agreement, not authentication.
 - ▶ You really don't know anything about who you have exchanged keys with
 - ▶ The man in the middle ...



- ▶ Alice and Bob think they are talking **directly** to each other, but Mallory is actually performing two separate exchanges
- You need to have an authenticated DH exchange
 - ▶ The parties sign the exchanges (more or less)
 - ▶ See Schneier for a intuitive description

Public Key Cryptography

- Public Key cryptography

- ▶ Each key pair consists of a public and private component: k^+ (public key), k^- (private key)

$$D(E(p, k^+), k^-) = p$$

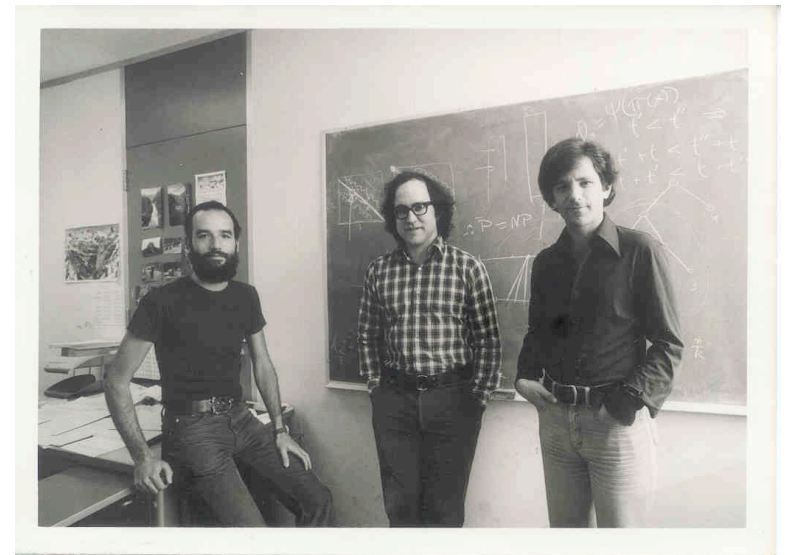
$$D(E(p, k^-), k^+) = p$$

- Public keys are distributed (typically) through public key certificates
 - ▶ Anyone can communicate secretly with you if they have your certificate
 - ▶ E.g., SSL-base web commerce

RSA (Rivest, Shamir, Adelman)

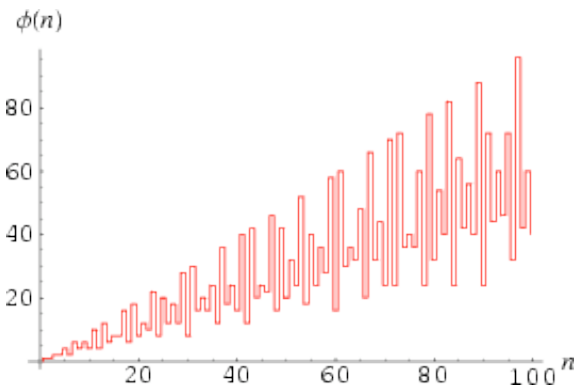
- A dominant public key algorithm
 - ▶ The algorithm itself is conceptually simple
 - ▶ Why it is secure is very deep (number theory)
 - ▶ Use properties of exponentiation modulo a product of large primes

"A method for obtaining Digital Signatures and Public Key Cryptosystems",
Communications of the ACM,
Feb., 1978 21(2) pages
120-126.



RSA Key Generation

- Pick two large primes p and q
- Calculate $n = pq$
- Pick e such that it is relatively prime to $\phi(n) = (q-1)(p-1)$
 - “Euler’s Totient Function”
- $d \sim e^{-1} \pmod{\phi(n)}$ or
 $de \pmod{\phi(n)} = 1$



1. $p=3, q=11$
2. $n = 3*11 = 33$
3. $\phi(n) = (2*10) = 20$
4. $e = 7 \mid \text{GCD}(20,7) = 1$
5. “Euclid’s Algorithm”
 $d = 7^{-1} \pmod{20}$
 $d \mid d7 \pmod{20} = 1$
 $d = 3$

RSA Encryption/Decryption



- Public key k^+ is $\{e,n\}$ and private key k^- is $\{d,n\}$
- Encryption and Decryption
 - $E(k^+,P) : \text{ciphertext} = \text{plaintext}^e \bmod n$
 - $D(k^-,C) : \text{plaintext} = \text{ciphertext}^d \bmod n$
- Example
 - ▶ Public key (7,33), Private Key (3,33)
 - ▶ Data “4” (encoding of actual data)
 - ▶ $E(\{7,33\},4) = 4^7 \bmod 33 = 16384 \bmod 33 = 16$
 - ▶ $D(\{3,33\},16) = 16^3 \bmod 33 = 4096 \bmod 33 = 4$

Encryption using private key

- Encryption and Decryption

$$E(k^-, P) : \text{ciphertext} = \text{plaintext}^d \bmod n$$

$$D(k^+, C) : \text{plaintext} = \text{ciphertext}^e \bmod n$$

- E.g.,

- ▶ $E(\{3,45\},4) = 4^3 \bmod 33 = 64 \bmod 33 = 31$

- ▶ $D(\{7,45\},19) = 31^7 \bmod 33 = 27,5|2,6|4,1|1 \bmod 33 = 4$

- Q: Why encrypt with private key?

Digital Signatures

- Models physical signatures in digital world
 - ▶ Association between private key and document
 - ▶ ... and indirectly identity and document.
 - ▶ Asserts that document is authentic and non-reputable
- To sign a document
 - ▶ Given document d , private key k^-
 - ▶ Signature $S(d) = E(k^-, h(d))$
- Validation
 - ▶ Given document d , signature $S(d)$, public key k^+
 - ▶ Validate $D(k^+, S(d)) = H(d)$



A Protocol Story

- Needham-Schroeder Public Key Protocol
 - ▶ Defined in 1978
- Assumed Correct
 - ▶ Many years without a flaw being discovered
- Proven Correct
 - ▶ BAN Logic
- So, It's Correct, Right?



Needham-Schroeder Public Key

- Does It Still Look OK?
- Message a.1: $A \rightarrow B : A, B, \{N_A, A\}_{PK_B}$
 - ▶ A initiates protocol with fresh value for B
- Message a.2: $B \rightarrow A : B, A, \{N_A, N_B\}_{PK_A}$
 - ▶ B demonstrates knowledge of N_A and challenges A
- Message a.3: $A \rightarrow B : A, B, \{N_B\}_{PK_B}$
 - ▶ A demonstrates knowledge of N_B
- A and B are the only ones who can read N_A and N_B

Nonce

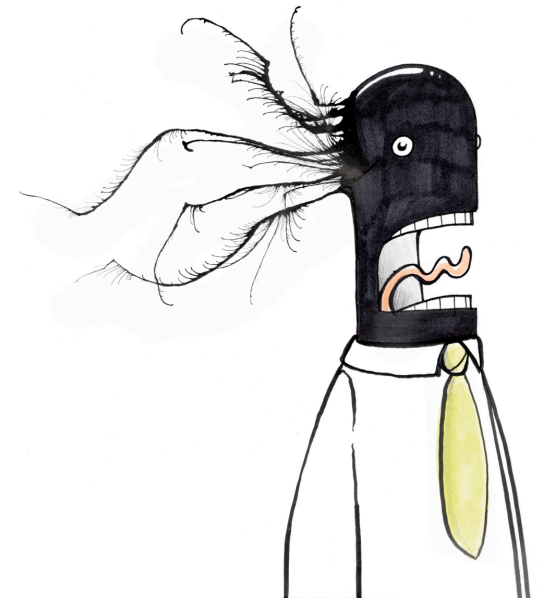


Gavin Lowe Attack

- An active intruder X participates...
- Message a.1: $A \rightarrow X : A, X, \{N_A, A\}_{PKX}$
- Message b.1: $X(A) \rightarrow B : A, B, \{N_A, A\}_{PKB}$
 - ▶ X as A initiates protocol with fresh value for B
- Message b.2: $B \rightarrow X(A) : B, A, \{N_A, N_B\}_{PKA}$
- Message a.2: $X \rightarrow A : X, A, \{N_A, N_B\}_{PKA}$
 - ▶ X asks A to demonstrate knowledge of N_B
- Message a.3: $A \rightarrow X : A, X, \{N_B\}_{PKX}$
 - ▶ A tells X N_B ; thanks A !
- Message b.3: $X(A) \rightarrow B : A, B, \{N_B\}_{PKB}$
 - ▶ X completes the protocol as A

What Happened?

- X can get A to act as an “oracle” for nonces
 - ▶ Hey A, what’s the N_B in this message from any B?
- A assumes that any message encrypted for it is legit
 - ▶ Bad idea
- X can enable multiple protocol executions to be interleaved
 - ▶ Should be part of the threat model?



The Fix

- It's Trivial (find it)
- Message a.1: $A \rightarrow B : A, B, \{N_A, A\}_{PK_B}$
 - ▶ A initiates protocol with fresh value for B
- Message a.2: $B \rightarrow A : B, A, \{N_A, N_B, \mathbf{B}\}_{PK_A}$
 - ▶ B demonstrates knowledge of N_A and challenges A
- Message a.3: $A \rightarrow B : A, B, \{N_B\}_{PK_B}$
 - ▶ A demonstrates knowledge of N_B

Impact on Protocol

- Protocol Analysis Took a Black Eye
 - ▶ BAN Logic Is Insufficient
 - ▶ BAN Logic Is Misleading
- Protocol Analysis Became a Hot Topic
 - ▶ Lowe's FDR
 - ▶ Meadow's NRL Analyzer
 - ▶ Millen's Interrogator
 - ▶ Rubin's Non-monotonic protocols
 - ▶
- In the end, could find known flaws, but...
 - ▶ attacker model is too complex

Dolev-Yao Result

- Strong attacker model
 - ▶ Attacker intercepts every message
 - ▶ Attacker can cause one of a set of operators to be applied at any time
 - Operators for modifying, generating any kind of message
 - ▶ Attacker can apply any operator except other's decryption
- Theoretical Results
 - ▶ Polynomial Time for One Session
 - ▶ Undecidable for Multiple Sessions
 - ▶ *Moral: Analysis is Difficult Because Attacker Can Exploit Interactions of Multiple Sessions*
- End Result: Manual Induction and Expert Analysis are

Review: secret vs. public

- Secret key cryptography
 - Symmetric keys, where A single key (k) is used is used for E and D
 - $D(E(p, k), k) = p$
- All (intended) receivers have access to key
- Note: Management of keys determines who has access to encrypted data
 - E.g., password encrypted email
- Also known as symmetric key cryptography

- Public key cryptography

Each key pair consists of a public and private component:

k^+ (public key), k^- (private key)

$D(E(p, k^+), k^-) = p$

$D(E(p, k^-), k^+) = p$

- Public keys are distributed (typically) through public key certificates
 - Anyone can communicate secretly with you if they have your certificate
 - E.g., SSL-based web commerce

The symmetric/asymmetric key tradeoff

- Symmetric (shared) key systems
 - ▶ Efficient (Many MB/sec throughput)
 - ▶ Difficult key management
 - Kerberos
 - Key agreement protocols
- Asymmetric (public) key systems
 - ▶ Slow algorithms (so far ...)
 - ▶ Easy (easier) key management
 - PKI - public key infrastructures
 - Webs of trust (PGP)

