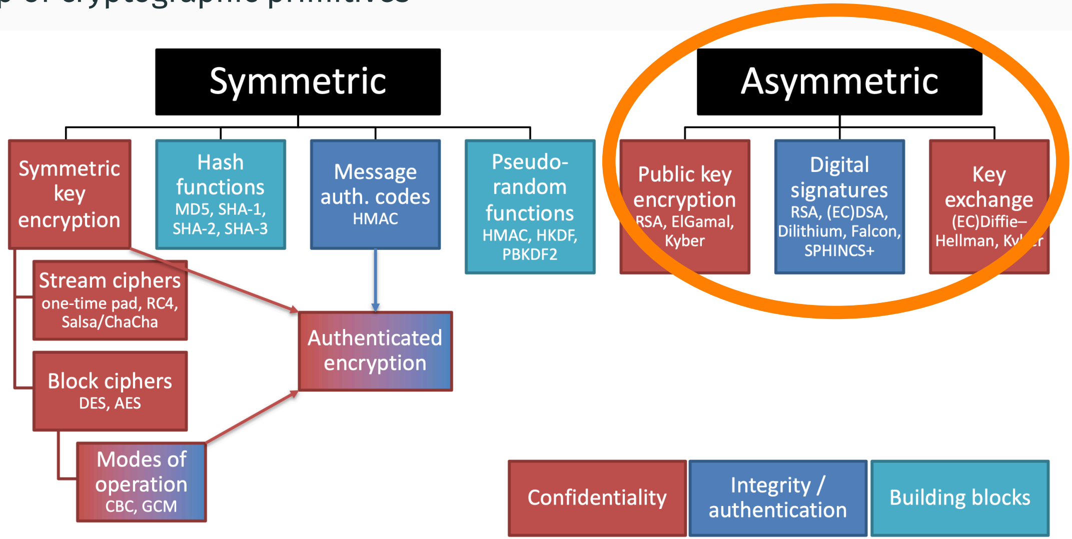
Topic 3.1
Public key cryptography – overview

Douglas Stebila

CO 487/687: Applied Cryptography

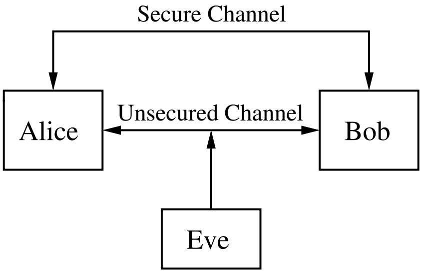
Fall 2024

Map of cryptographic primitives



Drawbacks with Symmetric-Key Cryptography

Symmetric-key cryptography:
Communicating parties a priori share some secret keying information.

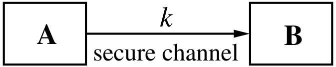


The shared secret keys can then be used to achieve confidentiality (e.g., using AES), or authentication (e.g., using HMAC), or both (e.g., using AES-GCM).

Key Establishment Problem

How do Alice and Bob establish the secret key k?

Method 1: Point-to-point key distribution.
(Alice selects the key and sends it to Bob over a secure channel)



The secure channel could be:

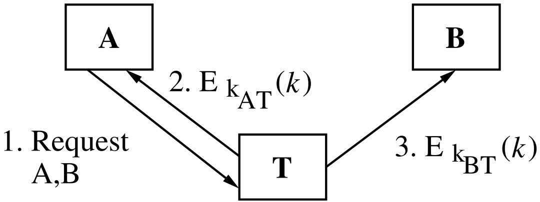
- A trusted courier.
- A face-to-face meeting.
- Installation of an authentication key in a SIM card.

This is generally not practical for large-scale applications.

Key Establishment Problem (2)

Method 2: Use a Trusted Third Party (TTP) T.

- Each user A shares a secret key k_{AT} with T for a symmetric-key encryption scheme E.
- To establish this key, A must visit T once.
- T serves as a key distribution centre (KDC)



1. A sends T a request for a key to share with B.
2. T selects a session key k, and encrypts it for A using k_{AT}.
3. T encrypts k for B using k_{BT}.

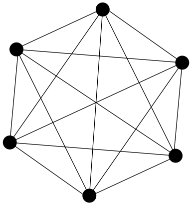
Key Establishment Problem (3)

Drawbacks of using a KDC:

1. The TTP must be unconditionally trusted.
2. The TTP it an attractive target.
3. The TTP must be on-line.
   - Potential bottleneck.
   - Critical reliability point.

Key Management Problem

- In a network of n users, each user has to share a different key with every other user.

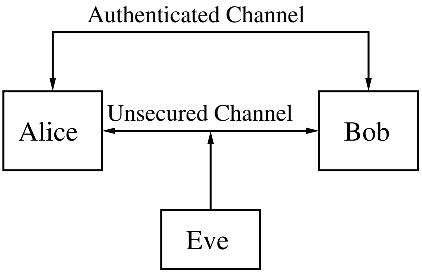


- Each user thus has to store $n-1$ different secret keys.
- The total number of secret keys is $\binom{n}{2} \approx n^2/2$.

Non-repudiation can’t be achieved using symmetric techniques

- Non-repudiation: Preventing an entity from denying previous actions or commitments.
  - Denying being the source of a message.
- Strictly speaking, symmetric-key techniques cannot be used to achieve non-repudiation.
  - Why?
- However, symmetric-key techniques can be used to achieve *some* degree of non-repudiation, but typically requires the services of an on-line TTP (e.g., use a MAC algorithm where each user shares a secret key with the TTP).

Public-Key Cryptography



Public-key cryptography: Communicating parties a priori share some authenticated (but non-secret) information.

Invented by Ralph Merkle, Whitfield Diffie, Martin Hellman in 1975.

Ralph Merkle (1974)

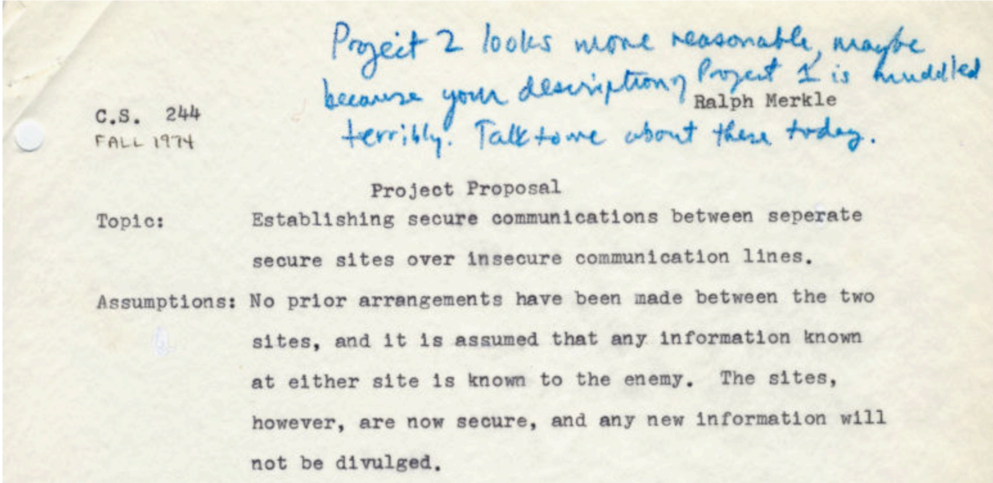
Excerpts from Merkle's CS 244 project proposal
(Computer Security, UC Berkeley, Fall 1974)

"Secure communications are made possible because of knowledge, known to both people, which is not known to anyone else. Usually, both people know this knowledge because they were able to hold a private conversation with each other before they began to send encrypted messages over an unsecure channel."

"It might seem intuitively obvious that if two people have never had opportunity to prearrange an encryption method, then they will be unable to communicate securely over an insecure channel. While this might seem intuitively obvious, I believe it is false. I believe that it is possible for two people to communicate securely without having made any prior arrangements that are not completely public."

https://www.ralphmerkle.com/1974/

Ralph Merkle (1974)



C.S. 244
FALL 1974

Project 2 looks more reasonable, maybe because your description of Project 1 is muddled terribly. Talk to me about them today.

Ralph Merkle

Project Proposal

Topic: Establishing secure communications between seperate secure sites over insecure communication lines.

Assumptions: No prior arrangements have been made between the two sites, and it is assumed that any information known at either site is known to the enemy. The sites, however, are now secure, and any new information will not be divulged.

https://www.ralphmerkle.com/1974/

Merkle Puzzles

Goal: Alice and Bob establish a secret session key by communicating over an authenticated (but non-secret) channel.

1. Alice creates N puzzles P_i, $\quad 1 \leq i \leq N$ (e.g., $N = 10^9$).
   Each puzzle takes t hours to solve (e.g., $t = 5$).
   The solution to P_i reveals a 128-bit session key sk_i and a randomly-selected 128-bit serial number n_i (which Alice selected and stored).
2. Alice sends $P_1, P_2, \ldots, P_N$ to Bob.
3. Bob selects j at random from $[1, N]$ and solves puzzle P_j to obtain sk_j and n_j.
4. Bob sends n_j to Alice.
5. The secret session key is sk_j.

An eavesdropper has to solve $N/2 =$ 500,000,000 puzzles on average to determine the puzzle index j (and thus sk_j).

Merkle Puzzles (2)

Example:

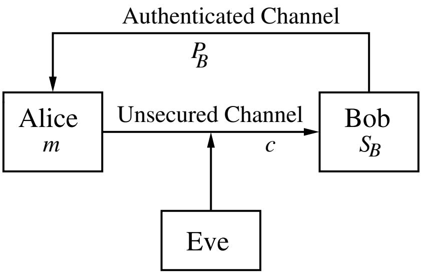
$P_i = \text{AES-CBC}_{k_i}(sk_i, n_i, n_i)$, where $k_i = (r_i \| 0^{88})$ and r_i is a randomly selected 40-bit string.

P_i can be solved in 2^{40} steps by exhaustive key search.

Key Pair Generation for Public-Key Crypto

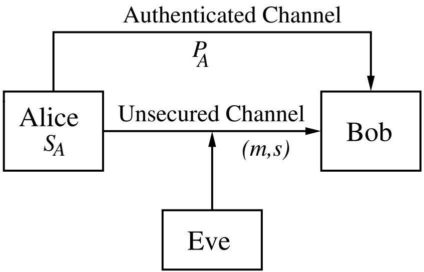
- Each entity A does the following:
    1. Generate a key pair (P_A, S_A).
    2. S_A is A's secret key.
    3. P_A is A's public key.
- Security requirement: It should be infeasible for an adversary to recover S_A from P_A.
- Example: $S_A = (p, q)$ where p are q are randomly-selected prime numbers; $P_A = p \cdot q$.

Public-Key Encryption



- To encrypt a secret message m for Bob, Alice does:
    1. Obtain an authentic copy of Bob's public key P_B.
    2. Compute $c = E(P_B, m)$; E is the encryption function.
    3. Send c to Bob.
- To decrypt c, Bob does:
    1. Compute $m = D(S_B, c)$; D is the decryption function.

Digital Signatures



- To sign a message m, Alice does:
  1. Compute $s = \text{Sign}(S_A, m)$.
  2. Send m and s to Bob.
- To verify Alice's signature s on m, Bob does:
  1. Obtain an authentic copy of Alice's public key P_A.
  2. Accept if $\text{Verify}(P_A, m, s) =$ "Accept".

Digital Signatures (2)

- Suppose that Alice generates a signed message (m, s).
- Then anyone who has an authentic copy of Alice's public key P_A can verify the authenticity of the signed message.
  - This authentication property cannot be achieved with a symmetric-key MAC scheme.
- Digital signatures are widely used to sign software updates which are then broadcast to computers around the world.

Public-Key Versus Symmetric-Key

Advantages of public-key cryptography:

- No requirement for a secured channel.
- Each user has only 1 key pair, which simplifies key management.
- A signed message can be verified by anyone.
- Facilitates the provision of non-repudiation services (with digital signatures).

Disadvantages of public-key cryptography:

- Public-key schemes are slower than their symmetric-key counterparts.

Hybrid Schemes: combining public-key and symmetric-key

In practice, symmetric-key and public-key schemes are used together. Here is an example:

To send a message m with confidentiality and authenticity, Alice does:

1. Select a secret key k for a symmetric-key encryption scheme such as AES.
2. Obtain an authentic copy of Bob's public key P_B.
3. Send $c_1 \leftarrow \text{PKE.Enc}(P_B, k)$, $c_2 \leftarrow \text{AES.Enc}_k(m)$, and $s \leftarrow \text{Sign}(S_A, H(c_1 \| c_2))$.

To recover m and verify its authenticity, Bob does:

1. Obtain an authentic copy of Alice's public key P_A.
2. Check that $\text{Verify}(P_A, H(c_1 \| c_2), s) = \text{Accept}$.
3. Decrypt c_1: $k \leftarrow \text{PKE.Dec}(S_B, c_1)$.
4. Decrypt c_2: $m \leftarrow \text{AES.Dec}_k(c_2)$.

4+8+7 things to remember from CO 487

Public key versus symmetric

- Symmetric key cryptography needs Alice and Bob to have the *same secret key* on both sides, which would have to be distributed in advance via a confidential channel.
- Public key cryptography only needs Alice and Bob to have *distributed their public keys* in advance over an authenticated (but not necessarily confidential) channel.
- But public key cryptography is usually *slower* than symmetric key cryptography.
- Combine both in “hybrid encryption” to get the best of both worlds.