CRYPTOGRAPHY LECTURE, 2022

Computer Science and Algorithmics, PWr

Mirosław Kutyłowski

Key agreement

Session key

- if communication has to be encrypted, we need a secret key K
- a symmetric scheme may be used, once the communicating parties share the key K
- how to establish a shared key?

Shared key schemes

- Alice and Bob hold a long term shared key L
- current session key derived from L
- it should be unpredictable

A typical scheme

i. Alice chooses a random nonce r_A and sends $\mathrm{Enc}_L(r_A, \mathrm{Alice})$ to Bob

ii. Bob decrypts the ciphertext, chooses a random nonce r_B and returns the following ciphertext to Bob $\mathrm{Enc}_L(r_B, r_A, \mathrm{Bob})$ to Bob

iii. Alice decrypts and checks whether r_A is contained in the plaintext

iv. both parties compute the session key $K := F(r_A, r_B)$, where F is a special one-way function (hash function)

Properties

- it suffices that randomness on one side is strong
- dishonest Bob has more influence on the final session key than Alice
- input for breaking L is very limited if the encryption scheme is carefully chosen
  - for any key S: $\mathrm{Enc}_L(x)$ should be a valid ciphertext with key S, that is: $\mathrm{Dec}_S(\mathrm{Enc}_L(x))$ will not fail

Mobile networks

session keys established for a base station and a “subscriber” (SIM card)

- i based on long term shared keys shared by the SIM card and a Telecom
- ii the long term shared key must not be sent to the base station (run by a different company)
- iii Telecom cannot manage all requests in real time (latency, . . .)

Solution:

i. Telecom sends to the serving network a list of entries: (c_i, K_i), where $K_i = F(L, c_i)$

ii. upon connection: the base station sends the challenge c_i, the SIM card computes $K_i := F(L, c_i)$

iii. master session key is K_i, used for PRNG and stream ciphers authentication is based of knowledge of the shared key K_i

WIFI

protocols based on a shared secret, WEP (serious weaknesses), WPA (strong but . . . also some concerns)

WPA (WPA-PSK - WPA with preshared key)

– PMK Pairwise Master Key is preshared

– PTK (Pairwise Transient Key) derived as a session key

namely: PTK=$f(\mathrm{PMK}, \mathrm{ANonce}, \mathrm{SNonce})$

– f can be chosen

– PTK splitted into TK (Temporal Key), KCK (Key Confirmation Key), KEK (Key Encryption Key)

– for WPA2 also GPK (Group Temporal Key) transported to the “supplicant” (user) (used by Access Point for broadcast)

How to establish a shared key?

Bluetooth concept:

- establish a shared key in a secure environment during the 1st pairing of devices
- the only protection: codes on both devices:
  - avoiding confusion
  - explicit consent of the user

Similar concepts:

- plaintext communication over power supply in a house
- plaintext optical communication

Ideal key exchange

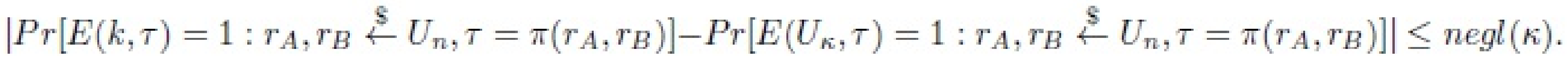
a protocol executed between Alice and Bob such that

- **passive adversary:** Mallet can hear each message transmitted
- **randomness:** Alice and Bob use some randomness (say r_A and r_B)
- **correctness:** (with very high probability) Alice and Bob derive the same session key
- **security:**

$$|Pr[E(k,\tau)=1 : r_A, r_B \overset{\$}{\leftarrow} U_n, \tau = \pi(r_A, r_B)] - Pr[E(U_\kappa,\tau)=1 : r_A, r_B \overset{\$}{\leftarrow} U_n, \tau = \pi(r_A, r_B)]| \leq negl(\kappa).$$

meaning: anything that Mallet can derive from the transcript of communication τ and the correct key, he can compute for τ and the random key

Discussion:



$$|Pr[E(k,\tau)=1 : r_A, r_B \overset{\$}{\leftarrow} U_n, \tau = \pi(r_A, r_B)] - Pr[E(U_\kappa,\tau)=1 : r_A, r_B \overset{\$}{\leftarrow} U_n, \tau = \pi(r_A, r_B)]| \leq negl(\kappa).$$

Diffie-Hellman key exchange in a group G

a seminal protocol for modern public-key cryptography,

i. Alice
   a. chooses r_A at random
   b. computes $c_A := g^{r_A}$ in G
   c. sends c_A to Bob
   d. receives c_B from Bob
   e. calculates $K := c_B^{r_A}$ in G

ii. Bob
   a. chooses r_B at random
   b. computes $c_B := g^{r_B}$ in G
   c. sends c_B to Alice
   d. receives c_A from Alice
   e. calculates $K := c_A^{r_B}$

Note that $K_{\text{Bob}} = c_A^{r_B} = g^{r_A r_B} = g^{r_B r_A} = c_B^{r_A} = K_{\text{Alice}}$

Computational Diffie-Hellman Problem

given g, g^a, g^b compute g^{ab} in the group G

remark: it is easy to compute $g^a \cdot g^b = g^{a+b}$

CDH assumption (we believe that it holds for certain groups)

given $g, A(=g^a), B(=g^b)$ chosen at random in G,

it is infeasible to compute $C = g^{a \cdot b}$

Example group:

Z_p with multiplication modulo p, where p is a large prime number

Decisional Diffie-Hellman Problem

a game where a challenger

i. chooses g^c, g^d at random

ii. chooses bit b at random

iii. if $b = 0$, then $R := g^{c \cdot d}$, else R is chosen at random

iv. presents the values (g, g^c, g^d, R)

the adversary responds a bit b' and wins if $b = b'$.

DDH Assumption

advantage of the adversary in the above game is negligible

meaning: it is infeasible to distinguish a random key from the real key, so the commpunication transcript is useless for the adversary

Man-in-the-middle attack (MitM)

- Alice and Bob think that they are executing DH protocol …
- … but in fact:

  Alice in talking with Mallet, Bob is talking with Mallet
- Alice establishes a session key K (with Mallet)
- Bob establishes a session key K' (with Mallet)
- during the session:
  - → when Alice sends $C = \mathrm{Enc}_K(M)$ then Mallet forwards to Bob the ciphertext $\mathrm{Enc}_{K'}(\mathrm{Dec}_K(C))$
  - → similarly from Bob to Alice
  - → Mallet learns every single message in plaintext!

DH is useless alone

Static Diffie-Hellman protocol

- Alice holds a long term public key $X_A = g^{x_A}$ and the private key x_A
- Bob knows that X_A is the key of Alice

Protocol

i. Bob chooses r_B at random, computes $c_B := g^{r_B}$ and sends c_B to Alice,

ii. Alice computes $K := c_B^{x_A}$

iii. Bob computes $K := X_A^{r_B}$

Properties:

- MitM does not work
- it is necessary to know X_A and trust it

Shamir's Key Transport Protocol

1. the sender chooses a key m

2. the sender chooses d_1 at random, calculates $e_1 := m^{d_1}$ and sends e_1 to the receiver

3. the receiver chooses d_2 at random, calculates $e_2 := e_1^{d_2}$, and sends e_2 back to the sender

4. the sender computes $e_3 := e_2^{1/d_1}$ and sends it to the receiver

   (note that $e_3 = ((m^{d_1})^{d_2})^{1/d_1} = m^{d_2}$)

5. the receiver calculates $m' := e_3^{1/d_2}$

Key agreement with public key encryption

A session key K chosen by one party and then encrypted with the public key of the receiver (nobody can decrypt except for the receiver)

… reduces the number of messages compared with Shamir

… better cryptographic properties than static DH

(static DH creates an oracle for computing A^{x_A})