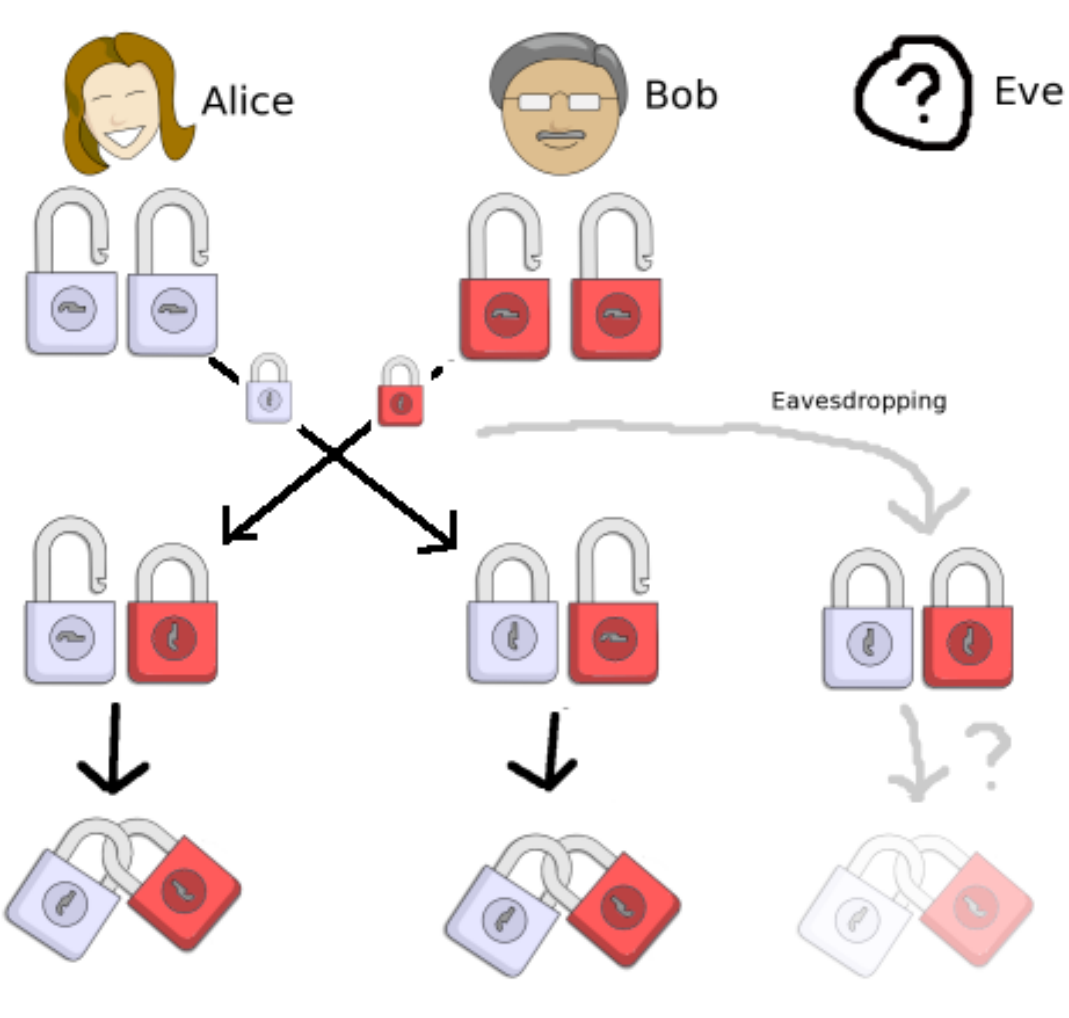
Diffie-Hellman key exchange



Diffie-Hellman key exchange

1. Alice and Bob agree on the public parameters of p prime and $g \in \mathbb{Z}_p^*$ generator.
2. Alice choose a random a number for her private parameter while Bob choose the random b as his private parameter.
3. Alice and Bob respectively calculates

$$g^a \pmod{p} \quad \text{and} \quad g^b \pmod{p}$$

   and sends it to the other.
4. Alice and Bob calculates the common key as

$$g^{ab} \equiv (g^b)^a \equiv (g^a)^b \pmod{p}.$$

Notes on DH

- Can be done with as many participant as we want the key exchange will be not less secure. The common key will be

$$g^{\prod_{i=1}^{n} a_i} \pmod{p}$$

- As usage the key exchange with a symmetric cipher has somewhat the same functionality as an asymmetric encryption (RSA, ECC) with different parameters but in usually in practice they are used together.
  - In case of DH the $(g^a\ (p),\ g,\ p)$ quartet can be used as public key.
  - In case of asymmetric encryption a common key can be calculated by concatenating two random values sent to each other via this encryption.

Notes on DH

Definition Perfect Froward Secrecy The preciously encrypted messages can not be decrypted based on later compromised keys.

DHE (Diffie-Hellman in ephemeral mode): The key exchange messages signed (by RSA or DSA), but the keys for authentication and encryption should be different and the later ones should be not stored and regenerated together with the common key.

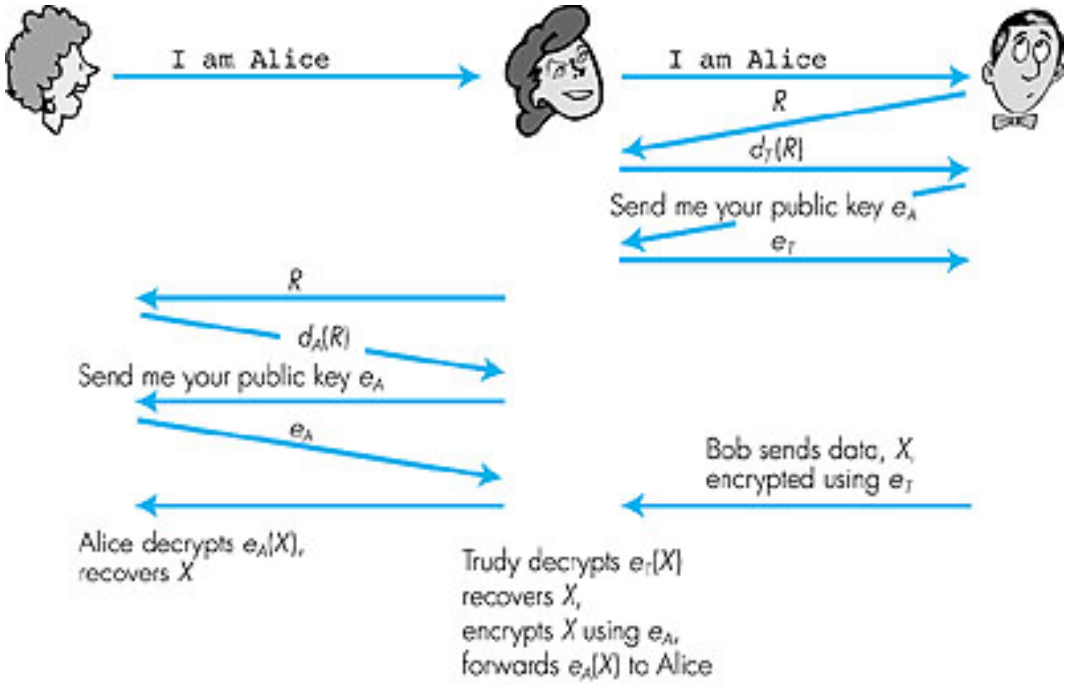
RSA standard is free but the DH is not.

Notes on DH

In theory the RSA and DH has the same security with the same size of keys, but in real life the DH (discrete logarithm) gives better results against cryptanalysis (except logjam), but DH over elliptic curves gives even better results.

As performance goes the public encryption of RSA is cheaper while the private encryption usually more expensive than DH with the same key sizes while we get even better results using EC.

DH attacks – Men-in-the-Middle



The DH messages could be authenticated by signatures or the protocol can be fortified (STS).

DH attacks – Men-in-the-Middle

STS (Station-to-Station) protocol where asymmetric keys for signatures and common parameters are pre-shared and

1. Alice generates random value a and send g^a to Bob.
2. Bob generates his random value b, computes g^b and calculates $k = g^{ab}$.
3. Bob calculates the signature for $g^b||g^a$, encrypt the signature with k and sends g^b and the signature to Alice.
4. Alice calculates k, decrypts and verifies the signature.
5. Alice calculates $g^a||g^b$, sings it, encrypts it with k and sends it to Bob.
6. Bob decrypt the received message with k and verifies it

DH attack – Logjam

To solve the discrete logarithm problem the number field sieve (most efficient algorithm) preforms 4 computational steps from which the first three (most expensive) steps only depend on the order of the group.
These steps can be pre-calculated for the handful of groups that are used in practice (good primes according to different standards).
The Logjam attack was performed pre-calculating the mentioned steps for 512 bit primes and then the individual logarithms could be solved in a few minutes each.
The authors of the logjam attack estimated that the cost of the pre-calculation would be in the order of $100 for 1024 bit primes (which is said to be done by NSA).

DH attacks

- Solving the discrete logarithm which is hard if the $p-1$ is not smooth.
- Rewinding old messages of key exchange to disturb or hijack the communication. Preventative
  - the messages can contain nounce
  - the messages can contain timestamp or sequence number
  - signature and/or HMAC
  - the common key not dropped until the new one verified
  - the acknowledged message can contain the hash of all the messages involved in the key agreement.

Protocols that are based on DH

- OTR, Socialist Millionaire, Millionaire,
- Dining Cryptographers
- Secure, verifiable voting system
  `http://compalg.inf.elte.hu/m̃erai/pub/vote_refereed.pdf`
- ElGamal

ElGamal encryption system

The ElGamal cipher is an asymmetric key encryption algorithm for public key cryptography which is based on DH.

The algorithm depends on the fact that is the discrete logarithm problem is hard.

The algorithm consist three components:

- key generator
- encryption algorithm
- decryption algorithm

ElGama – Key generation

1. Alice generates a cyclic group G of order q with generator g.
2. Alice chooses an x from G.
3. Alice computes $h = g^x$
4. Alice publishes (h, G, q, g) as her public key, while the x is her private key.

ElGamal – Encryption

1. Bob chooses a random y from the group and then calculates $c_1 = g^y$.
2. Bob calculates the shared secret $s = h^y = g^{xy}$.
3. Bob maps his message m onto an element m' of G group.
4. Bob calculates $c_2 = m's$.
5. Bob sends the ciphertext $(c_1, c_2) = (g^y, m'h^y) = (g^y, m'g^{xy})$ to Alice.

The y is an ephemeral key thus new y is used for each message.

ElGamal – Decrypting

1. Alice calculates the shared secret by $s = c_1^x$.
2. Alice computes $m' = c_2 s^{-1}$ which she then converts back into te plaintext message m.

$$c_2 s^{-1} = m' h^y (g^{xy})^{-1} = m' g^{xy} g^{-xy}$$

ElGamal – Hardness assumptions

Decisional Diffie-Hellman assumption (DDH): For any g^a and g^b the g^{ab} looks like a random element. (Stronger than the discrete logarithm because there can be groups where not hard to show such pairs but the DL remains hard.)

Computational Diffie-Hellman assumption: For a chosen generator and random a, b it is computationally intractable to compute gab from the g^a, g^b pair. (Stronger than DL but weaker than DDH).

If the DDH true in a group then the ElGamal semantic security while if the only the CDH holds then the encryption function is one-way.