# Key Agreement and Distribution

**Abstract**

*This section presents the basic principles of symmetric key establishment, that is the procedures to agree or distribute symmetric keys (the shared secret) between communicating entities. About symmetric key agreement, the Diffie-Hellman and Station-to-Station protocols are presented. About symmetric key distribution, the Shamir-Massey-Omura, Blom, Wide-Mouthed Frog, and Needham-Schroeder protocols are presented.*

*Stefano Bregni*

1

---

## Outline

- **Concepts of key agreement and distribution**
- Symmetric key agreement
    - Diffie-Hellman key exchange protocol
    - Station-to-Station protocol
- Symmetric key distribution
    - Shamir-Massey-Omura 3-Pass Protocol
    - Blom Scheme for key pre-distribution
    - Wide-Mouthed Frog protocol
    - Needham-Schroeder protocol

## Motivation and Rationale

- Cryptography is secure, practically unbreakable
  - Kerckhoffs: security is based on *key secrecy* (symmetric or private key)
- *Symmetric-key* cryptography
  - how to distribute secret symmetric keys to users securely?
- *Public-key* cryptography
  - no keys to distribute
  - public keys are *posted*, not distributed to single users
  - private keys cannot be derived from public keys (without trapdoor)
- Public-key cryptography seems to be a panacea for secure communication
  - to distribute encrypted symmetric keys to users
  - to send encrypted messages (heavy computational load)
  - but how to trust public keys are posted by legitimate entities?
  
  authentication and certificates

---

## Key Agreement and Distribution

- *Symmetric-key establishment*
  procedure to establish a *shared secret* (one common symmetric key or two different ones) between communicating entities
  - *symmetric-key agreement*
    – with authentication
    – without authentication
    
    **the key is computed by users in cooperation**
  - *symmetric-key distribution*
    – with authentication
    – without authentication
    
    **the key is determined in advance by an entity and transmitted to others**
    – public-key cryptography is a common solution to distribute secret symmetric keys (e.g., RSA to distribute DES keys to send messages)

## Outline

- Concepts of key agreement and distribution
- **Symmetric key agreement**
    - Diffie-Hellman key exchange protocol
    - Station-to-Station protocol
- Symmetric key distribution
    - Shamir-Massey-Omura 3-Pass Protocol
    - Blom Scheme for key pre-distribution
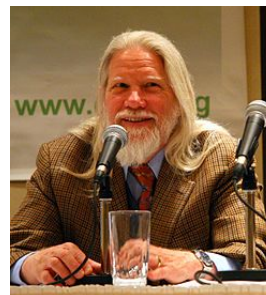    - Wide-Mouthed Frog protocol
    - Needham-Schroeder protocol

---

Key Agreement (without Authentication)
## Diffie-Hellman Key Exchange Protocol

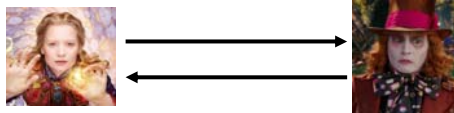*First published in 1976*

- Whitfield Diffie

- Martin Hellmann

3

# Diffie-Hellman Key Exchange Protocol

- Alice chooses a suitable prime $p$ (DLP difficult in $Z_p{}^*$) and a primitive root $\alpha$ (mod $p$). Numbers $p$ and $\alpha$ are public

- Alice chooses a secret random $1 \leq x \leq p\text{-}2$ (secret)
  Bob chooses a secret random $1 \leq y \leq p\text{-}2$ (secret)

- Alice ➜ Bob:  $\alpha^x$ (mod $p$)
  Alice ⬅ Bob:  $\alpha^y$ (mod $p$)

- Alice computes $K \equiv K_{AB} \equiv (\alpha^y)^x$ (mod $p$)
  Bob computes $K \equiv K_{BA} \equiv (\alpha^x)^y$ (mod $p$)



Key Agreement and Distribution

*Stefano Bregni*

7

---

# Man-in-the-Middle Attack to Diffie-Hellman

- Alice chooses a suitable prime $p$ (DLP difficult in $\mathbb{Z}_p{}^*$) and a primitive root $\alpha$ (mod $p$). Numbers $p$ and $\alpha$ are public.

- Alice chooses a random $1 \leq x \leq p\text{-}2$ (secret).
  Bob chooses a random $1 \leq y \leq p\text{-}2$ (secret).

- Alice ➜ Bob: $\alpha^x$ (mod $p$)        Alice ⬅ Bob: $\alpha^y$ (mod $p$)

- Eva chooses a random $1 \leq z \leq p\text{-}2$ and intercepts $\alpha^x$, $\alpha^y$

- Alice ⬅ Eva: $\alpha^z$ (mod $p$) (Alice thinks it's Bob's $\alpha^y$)
  Eva ➜ Bob: $\alpha^z$ (mod $p$) (Bob thinks it's Alice's $\alpha^x$)

- Eva computes $K_{EA} \equiv (\alpha^x)^z$ (mod $p$) and $K_{EB} \equiv (\alpha^y)^z$ (mod $p$)
  Alice and Bob compute $K_{AE} \equiv (\alpha^z)^x$ (mod $p$) and $K_{BE} \equiv (\alpha^z)^y$ (mod $p$)



Key Agreement and Distribution

*Stefano Bregni*

8

4

**Authenticated Key Agreement**
# Station-to-Station Protocol

- Basically a Diffie-Hellmann protocol enhanced with the added feature of authentication by digital signatures

1. They choose a large prime $p$ and a primitive root $\alpha$.
2. Alice chooses a random $x$ and Bob chooses a random $y$.
3. Alice computes $\alpha^x \pmod{p}$, and Bob computes $\alpha^y \pmod{p}$.
4. Alice sends $\alpha^x$ to Bob.
5. Bob computes $K \equiv (\alpha^x)^y \pmod{p}$.
6. Bob sends $\alpha^y$ and $E_K(sig_B(\alpha^y, \alpha^x))$ to Alice.
7. Alice computes $K \equiv (\alpha^y)^x \pmod{p}$.
8. Alice decrypts $E_K(sig_B(\alpha^y, \alpha^x))$ to obtain $sig_B(\alpha^y, \alpha^x)$.
9. Alice asks Trent to verify that $ver_B$ is Bob's verification algorithm.
10. Alice uses $ver_B$ to verify Bob's signature.
11. Alice sends $E_K(sig_A(\alpha^x, \alpha^y))$ to Bob.
12. Bob decrypts, asks Trent to verify that $ver_A$ is Alice's verification algorithm, and then uses $ver_A$ to verify Alice's signature.

**Key Agreement and Distribution**
*9*

*Stefano Bregni*

---

# Outline

- Concepts of key agreement and distribution
- Symmetric key agreement
  - Diffie-Hellman key exchange protocol
  - Station-to-Station protocol

- **Symmetric key distribution**
  - Shamir-Massey-Omura 3-Pass Protocol
  - Blom Scheme for key pre-distribution
  - Wide-Mouthed Frog protocol
  - Needham-Schroeder protocol

**Key Agreement and Distribution**
*10*

*Stefano Bregni*

## Distribution of Symmetric Keys in Advance

- Requires a secure channel to start and for any update
- The longer a symmetric key is used, the easier is crypto-analysis
- Public-key cryptography is a common solution to distribute secret symmetric keys
  - e.g., RSA to distribute DES keys to send messages
- A *Trusted Authority* (TA) can generate and distribute a symmetric key $K_{ij} = K_{ji}$ for each couple of users $i$, $j$
  - total number of keys: $n(n\text{-}1)/2$

**Key Agreement and Distribution**

*Stefano Bregni*

*11*

---

## Shamir-Massey-Omura 3-Pass Protocol

- Jim K. Omura

*First published in 1980-82*

- Adi Shamir

- James Massey



**Key Agreement and Distribution**

*Stefano Bregni*

*12*

## Shamir-Massey-Omura 3-Pass Protocol

- To transfer a secret key $K$ from Alice to Bob via a public unsecure channel
- Alice chooses a suitable prime $p$ (DLP difficult in $Z_p{}^*$) large enough to represent $K$. The prime $p$ is posted public.
- Alice chooses a secret random $a$ with MCD($a,p$-1)=1 and computes $a^{-1}$ (mod ($p$-1) )
  Bob chooses a secret random $b$ with MCD($b,p$-1)=1 and computes $b^{-1}$ (mod ($p$-1) )
- Alice ➔ Bob:    $K_1 \equiv K^a$ (mod $p$)
  Alice ← Bob:    $K_2 \equiv K_1{}^b \equiv K^{ab}$ (mod $p$)
  Alice ➔ Bob:    $K_3 \equiv K_2{}^{a^{-1}} \equiv K^{aba^{-1}}$ (mod $p$)
- Bob computes   $K \equiv K_3{}^{b^{-1}} \equiv K^{aba^{-1}b^{-1}}$ (mod $p$)
- Vulnerable to the Man-in-the-Middle attack

**Key Agreement and Distribution**                                                              *Stefano Bregni*
*13*

---

## Generalized Shamir-Massey-Omura Protocol

- The Shamir-Massey-Omura protocol can be generalized to any encryption function with commutative property
  - for example, to RSA:
    $RSA_{K_B}[ RSA_{K_A}(P) ] \equiv RSA_{K_A}[ RSA_{K_B}(P) ]$
- Alice ➔ Bob:    $RSA_{K_A}(P)$
- Alice ← Bob:    $RSA_{K_B}[ RSA_{K_A}(P) ]$
- Alice ➔ Bob:    $RSA_{K_B}(P) = RSA_{K_A-1}[ RSA_{K_B}[ RSA_{K_A}(P) ] ]$
- Bob computes   $P = RSA_{K_B-1}[ RSA_{K_B}(P) ]$
- There is no need to publish and certify keys $K_A$ and $K_B$

**Key Agreement and Distribution**                                                              *Stefano Bregni*
*14*

# Blom Scheme for Key Pre-Distribution

1. Each user $U$ in the network is assigned a distinct public number $r_U$ (mod $p$).

2. Trent chooses three secret random numbers $a$, $b$, and $c$ mod $p$.

3. For each user $U$, Trent calculates the numbers

$$a_U \equiv a + br_U \pmod{p} \quad b_U \equiv b + cr_U \pmod{p}$$

and sends them via his secure channel to $U$.

4. Each user $U$ forms the linear polynomial

$$g_U(x) = a_U + b_U x.$$

5. If Alice (A) wants to communicate with Bob (B), then Alice computes $K_{AB} = g_A(r_B)$, while Bob computes $K_{BA} = g_B(r_A)$.

6. It can be shown that $K_{AB} = K_{BA}$ (Exercise 2). Alice and Bob communicate via a symmetric encryption system, for example, DES, using the key (or a key derived from) $K_{AB}$.

---

# Authenticated Key Distribution

- How to trust the Trusted Authority?
- *Replay Attack* by a Man-in-the-Middle
  - Oscar intercepts a message and repeats it later, attempting to impersonate someone or urge some response
- To counter Replay Attacks:
  - authentication of senders and secrecy of *Key Encryption Keys*
    - who provides the KEKs shared with the TA?
  - ensure that messages are current
    - *sequence number (*entities must track all numbers to detect re-usage)
    - *timestamp (*entities must be time-synchronized)
    - *nonce (*random number used once in a challenge-and-response scheme)
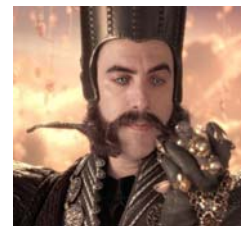
# Wide-Mouthed Frog Protocol

- One of the simplest key-distribution protocols based on symmetric key encryption involving a Trusted Authority (Trent)
- Based on using timestamps
- Alice chooses a Session Key $K_{AB}$ to communicate with Bob and requests Trent to transfer it to Bob securely

  1. Alice → Trent: $E_{K_{AT}}[t_A \| ID_B \| K_{AB}]$.
  2. Trent → Bob: $E_{K_{BT}}[t_T \| ID_A \| K_{AB}]$.

- The two timestamps ensure that messages are fresh
- The timestamp update by Trent implies a potential threat

---

# Man-in-the-Middle Attack to Wide-Mouthed Frog

- Mallory induces Trent, Alice and Bob to extend the validity of an old key, even indefinitely

  1. After seeing one exchange of the protocol, Mallory pretends to be Bob wanting to share a key with Alice. Mallory sends Trent the replay $E_{K_{BT}}[t_T \| ID_A \| K_{AB}]$.

  2. Trent sends $E_{K_{AT}}\left[t'_T \| ID_B \| K_{AB}\right]$ to Alice, with a new timestamp $t'_T$. Alice thinks this is a valid message since it came from Trent and was encrypted using Trent's key. The key $K_{AB}$ will now be valid for a period of time after $t'_T$.

  3. Mallory then pretends to be Alice and gets $E_{K_{BT}}\left[t''_T \| ID_A \| K_{AB}\right]$. The key $K_{AB}$ will now be valid for a period of time after $t''_T > t'_T$.

  4. Mallory continues alternately playing Trent against Bob and then Trent against Alice.

**Authenticated Key Distribution**
# Needham-Schroeder Protocol

- Alice and Bob obtain a Session Key $K_{AB}$ from Trent to communicate with each other
- Based on using nonces
- Two versions exist
  - public key with Trent (NSPK)
  - **symmetric key with Trent (NSSK)**

1. Alice → Trent: $ID_A \| ID_B \| r_1$

2. Trent → Alice: $E_{K_{AT}}[K_S \| ID_B \| r_1 \| E_{K_{BT}}[K_S \| ID_A]]$

3. Alice → Bob: $E_{K_{BT}}[K_S \| ID_A]$

4. Bob → Alice: $E_{K_S}[r_2]$

5. Alice → Bob: $E_{K_S}[r_2 - 1]$