Key establishment (and other protocols)

Vašek Matyáš

PV079

Agenda

1. Introduction to key establishment protocols
2. Attacks
3. Time-variant parameters
4. An example of protocol review/design
5. Involvement of trusted third parties
6. Generation from biometrics
7. Conclusions

Protocol

- A multi-party algorithm, defined by a sequence of steps precisely specifying the actions required of two or more parties in order to achieve a specified objective [HoC]
- Security / cryptography protocols objectives
  - Confidentiality (secrecy), authentication of origin, entity authentication, integrity, key establishment, non-repudiation…

Key establishment protocols

- Shared secret becomes available to two or more parties, for subsequent cryptographic use
- **Key transport** – one party (securely) transfers a secret value to other(s)
- **Key agreement** – shared secret is derived by two (or more) parties based on data contributed by, or associated with, each of these, and (ideally) that no party can pre-determine the resulting value

Key establishment concepts

- **Key authentication** (**implicit**) – assurance to one party that no-one except the specific other party could have gained access to a given key
- **Key confirmation** – assurance to one party that another party actually has a given key
- **Explicit key authentication** – both above hold

- **Entity authentication** – assurance to one party of the identity of another party actively involved in a protocol

KE protocol characteristics

- Key freshness
- Key control
  - Can any party control or predict the key value?
- Efficiency
  - Number of message exchanges (passes)
  - Volume of data exchanged
  - Complexity of computation
  - Possibility of pre-computation
- Material pre-distribution (system setup, certificates…)
- Third party involvement
- Non-repudiation

Types of KE protocols

- Key transport based on symmetric techniques
- Key transport based on asymmetric techniques
- Key agreement based on symmetric techniques
- Key agreement based on asymmetric techniques

- Secret sharing
- Conference keying

Attacker can…

- Record messages
- Replay them later
  - Possibly in different order
  - Some repeatedly
  - Some not at all
- Modify a part of or whole message

Types of attacks on protocols

- Man-in-the-middle
- Replay
- Reflection
- Interleave
- Oracle (chosen-text)
- Forced delay
- …

Effects of key compromise

- **Perfect forward secrecy** – compromise of long-term secret keys doesn't compromise past session keys
- **Known-key attack resistance** – past session keys don't enable
  - Passive adversary to compromise future session keys
  - Active adversary to impersonate another party in the future

Knowledge of secret key – authentication

- For shared-key crypto based on
  - trust in the party the key is shared with
  - *Authentication ~ Ability to en-/de-crypt (or MAC…)*
- For public-key crypto based on
  - trust in the party possessing the private key and
  - trust in link between the public key and other data
  - *Authentication ~ Ability to sign or decrypt messages*

Entity authentication

- Unilateral / mutual
- Secret-based authentication
  - Weak
  - Challenge-response
  - Zero-knowledge

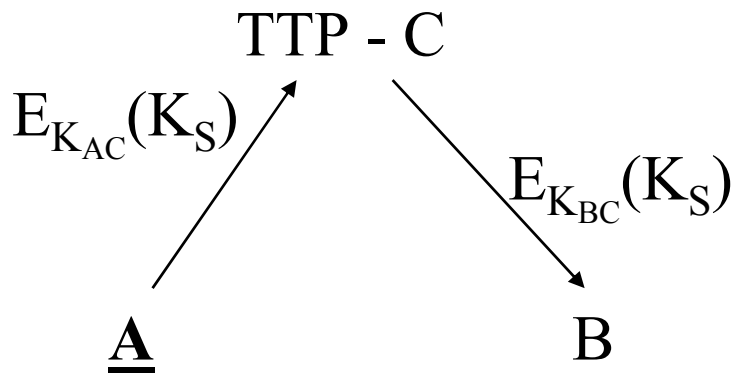
Use of session (short-term) keys

- To limit volume of ciphertext (under one key) for cryptanalytic attack
- To limit the window of exposure (time and data volume) in the event of key compromise
- To avoid storing large number of distinct keys by creating keys only when actually needed
- To create independence across sessions and/or applications

Establishing a session key

- Direct distribution

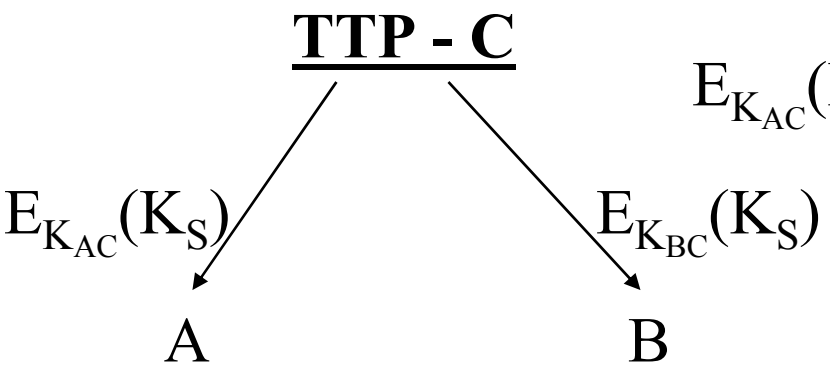
- Key transport center



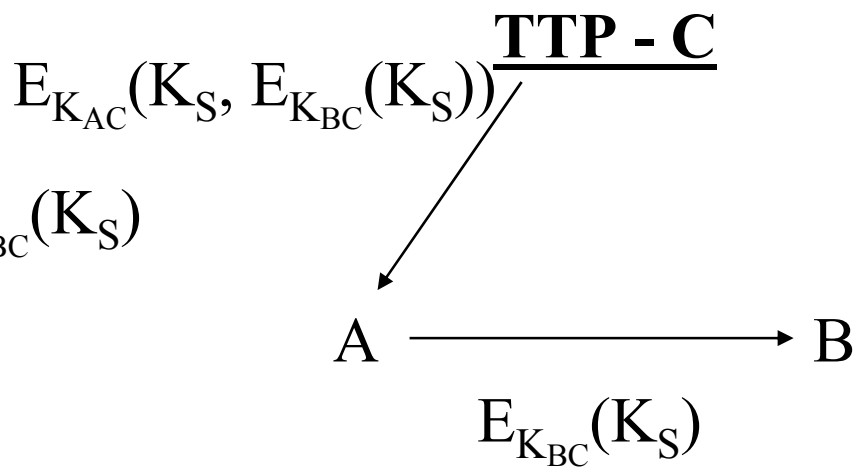
Establishing a session key, cont'd

Key distribution center

TTP-managed



Direct (pull/push)



Key transport successful?

- **Key authentication** (**implicit**) – assurance to one party that no-one except the specific other party could have gained access to a given key
- **Key confirmation** – assurance to one party that another party actually has a given key

- **Key receipt indication** – indication to one party that another party received the key

Zero-knowledge protocols

- Proof of knowledge – interactive proof with
  - **Completeness** – honest parties succeed with proof of acceptable probability for prover's claim
  - **Soundness** – dishonest prover cannot convince honest verifier without revealing the secret
- **Zero-knowledge** – when the communication between prover and verifier can be simulated without access to the secret knowledge

Zero-knowledge protocols

- A $\rightarrow$ B : witness
- A $\leftarrow$ B : challenge
- A $\rightarrow$ B : response

- **Zero-knowledge** – when the communication between prover and verifier can be simulated without access to the secret knowledge

Time-variant parameters (nonces)

- Random numbers (select from a uniform distribution), challenge-response
  - freshness
- Sequence numbers
  - Greater-by-one or only monotonic increase check
  - Counter maintenance, reset policy
- Timestamps
  - Acceptance window
  - Secure, synchronized & distributed time info (clocks)

Key transport – symmetric techniques

- $A \rightarrow B : E_K(r_A , TVP^* , A^* , B^*)$

- $A \leftarrow B : n_B$
- $A \rightarrow B : E_K(r_A , n_B , A^* , B^*)$

Shamir's no-key protocol

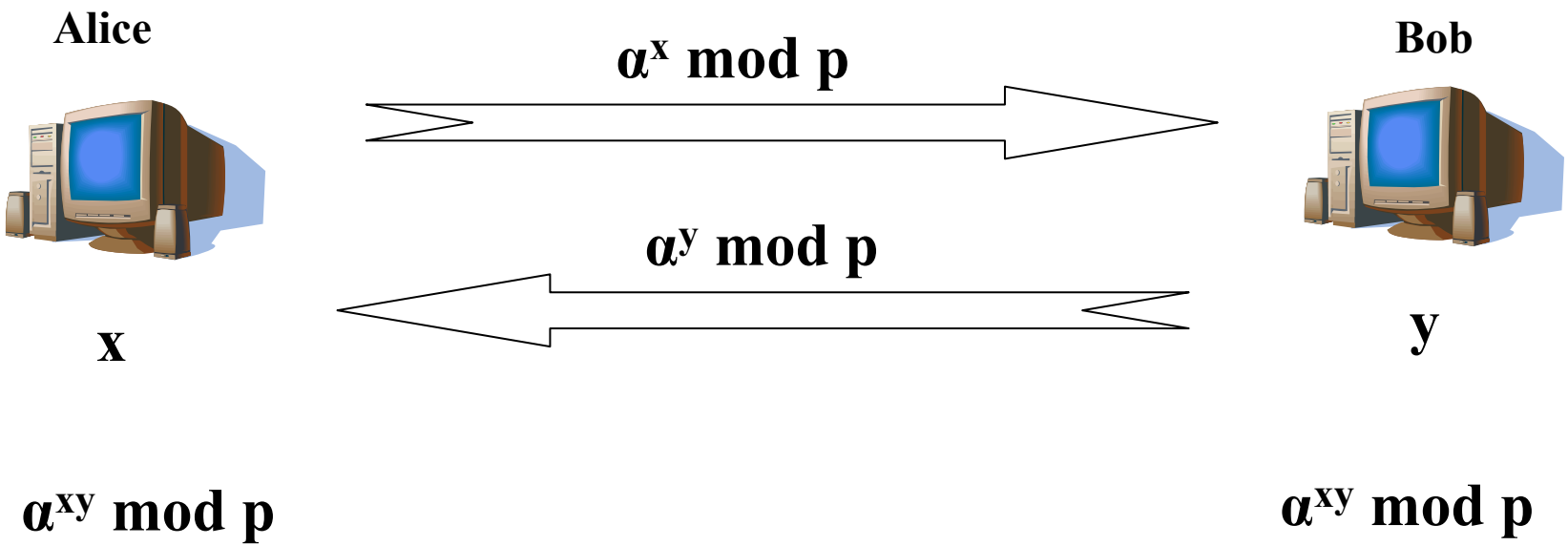
- $A \rightarrow B : E_{K_A}(X)$
- $A \leftarrow B : E_{K_B}(E_{K_A}(X))$
- $A \rightarrow B : E_{K_B}(X)$

- Use of a commutative cipher (not Vernam's ☺)

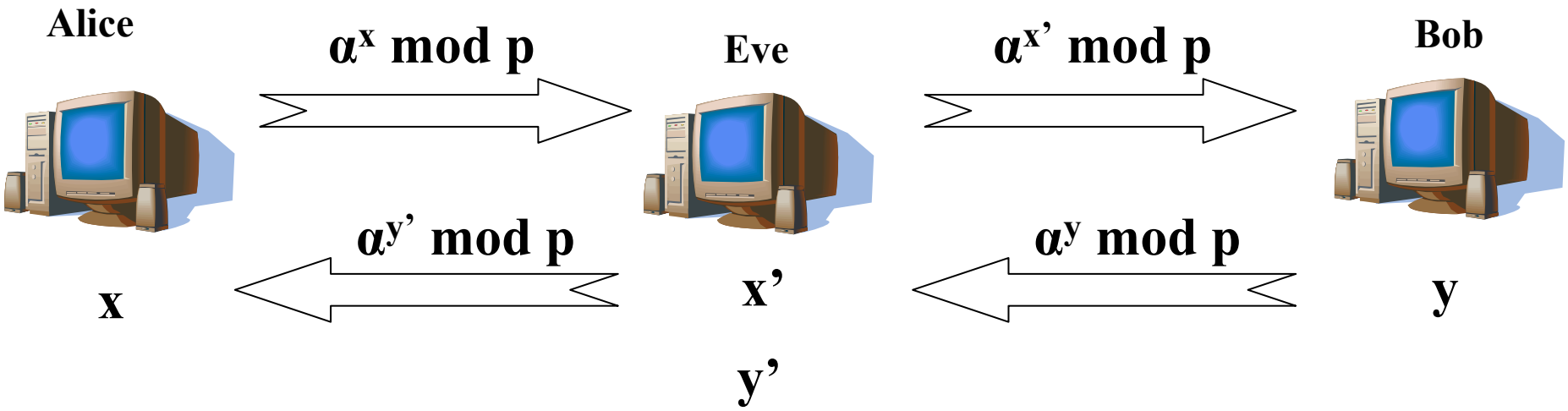
Fiat-Shamir identification protocol

- A trusted center T selects and publishes RSA-like modulus $n = p \cdot q$, keeps p and q secret
- A selects secret **s** (coprime with n, $1 \leq s \leq n-1$), computes $v = s^2 \bmod n$. This v is the public key of A.
- One round of A's authentication to B has three steps:
  - $A \rightarrow B: x = r^2 \bmod n$
  - $A \leftarrow B: e, e \in \{0, 1\}$
  - $A \rightarrow B: y = r \cdot s^e \bmod n$
- These steps are iterated t-times, then the probability of A successfully cheating is 2^{-t}.

Diffie-Hellman protocol



Man-in-the-middle attack



COMSET protocol

- $A \leftarrow B : r_B , P_A(r_B , r_A , K_B)$
- $A \rightarrow B : r_A$

- Unilateral authentication of A to B
- Key transfer from B to A

- Role of r_B is to convince A of B's knowledge of the encrypted message

R-COMSET

- $A \rightarrow B : r_{1A}, P_B(A, K_A, r_{1A}, r_{2A}, TVP_A)$ (1)
- $A \leftarrow B : r_{1B}, r_{2A}, P_A(B, K_B, r_{1B}, r_{2B}, TVP_B)$ (2)
- $A \rightarrow B : r_{2B}$ (3)

- Mutual authentication of A and B
- Confidential exchange of two key parts
- Final key to be calculated as a one-way function of the two keys (XOR prone to attacks – Burmester'94)

R-COMSET interleaving attack

- $A \rightarrow B : r_{1A}, P_B(A, K_A, r_{1A}, r_{2A}, TVP_A)$ (1)
- $C \leftarrow B : r_{1B}, r_{2A}, P_A(B, K_B, r_{1B}, r_{2B}, TVP_B)$ (2)

- $C \rightarrow A : r_{1B}, P_A(B, K_B, r_{1B}, r_{2B}, TVP_B)$ (i)
- $C \rightarrow A : r_{1C}, r_{2A}, P_A(B, K_C, r_{1C}, r_{2C}, TVP_C)$ (2')

- $C \leftarrow A : r'_{1A}, r_{2B}, P_B(A, K'_A, r'_{1A}, r'_{2A}, TVP'_A)$ (ii)

- $C \rightarrow B : r_{2B}$ (3')
- "Communication problem" (iii)

RRC (Revised R-COMSET)

- $A \rightarrow B : r_{1A}, P_B(A^*, K_A, r_{1A}, r_{2A}, TVP^*_A)$ (1)
- $A \leftarrow B : P_A(B^*, K_B, r_{2A}, r_B, TVP^*_B)$ (2)
- $A \rightarrow B : r_B$ (3)

- A does not behave like an oracle
- Can consider (3) to be a one-way function of r_B

Helsinki protocol

- $A \rightarrow B : P_B(A, K_A , r_A , TVP^*_A)$ (1)
- $A \leftarrow B : P_A(B^*, K_B , r_A , r_B , TVP^*_B)$ (2)
- $A \rightarrow B : r_B$ (3)

- r_{1A} eliminated – redundancy/integrity check by A
  - yet recall the original role "to convince about knowledge of the encrypted message"
- Effectively a modification of Needham-Schroeder public-key protocol

Quarter of a century…

- R. Needham & M. Schroeder – "Using encryption for authentication in large networks of computers", Comm. ACM, vol. 21, no. 12, pp. 993-999, 1978.
- Introduced both public- and shared-key protocols
- Predicted use of hybrid cryptosystems
- Raised the issue of subtle problems in protocols and argued for their analysis/verification

`http://lambda.cs.yale.edu/cs422/doc/needham.pdf`

Needham-Schroeder public-key protocol

- $A \rightarrow B : P_B(K_A , A)$ (1)
- $A \leftarrow B : P_A(K_A , K_B)$ (2)
- $A \rightarrow B : P_B(K_B)$ (3)

- A's private key compromise affects both K_A , K_B and therefore also the final session key, unlike the protocols studied before
- To detect replay, session keys (or at least images) have to be kept

ISO/IEC 11770 (1999)

- Information technology – Security techniques – Key Management
- Part 1: Key management framework
- Part 2: Mechanisms using symmetric techniques
- Part 3: Mechanisms using asymmetric techniques

ISO/IEC 11770-3

- Secret key agreement (7 mechanisms)
- Secret key transport (6 mechanisms)
- Public key transport
  - Without a TTP (2 mechanisms)
  - Using a CA (1 mechanism)

Involvement of trusted parties

- For system setup and/or any protocol run
  - Off-line, on-line, in-line
- Key transport and/or generation
- Trust to keep secrets vs. trust to certify data
- Assumptions of following the course of action prescribed by the protocol, not knowingly collaborating with attackers, etc.

Identity-based systems

- Users don't have explicit public keys
- Yet in all approaches at some stage a trusted third party involvement is required to provide a link between users' identities (or other public information) and their private keys

The Global Trust Register

www.cl.cam.ac.uk/Research/Security/Trust-Register

Global Trust Register

- Paper-based Register (off-line top-level CA)
- Keys and other info (URL, address, phone...)
- Keys verified and rated D➢ C➢ B➢ A (highest)
- Reliable, convenient, free press privilege

- Top-level X.509 CAs (and secure websites)
- Important PGP keys
- EDI and Entrust/Solo(X.509) keys

Global Trust Register – lessons

- Importance of revocation – critical
- High-level certificates stable
- Problems (e.g., user interface) with browser and e-mail client software
- Split of confidentiality and authentication keys
- CA operations expensive

Biometrics and cryptographic material

- Biometrics – automated methods of identity verification or identification based on measurable biological characteristics
- Biometrics almost never match at 100%!!!
- Threshold-based decision introduces the errors of *false acceptance* and *rejection*
  - Zero-effort or active bypassing?
  - User group size vs. accuracy
    - Verification vs. identification?

Key-generation attempts

- User provides her/his biometric sample and her/his key can be generated from this sample
- Attractive benefits
  - Key (re-)generated "on the fly"
  - Key is used only with its owner present
  - Can be used and then destroyed

Simplistic approach

- Find an invariant part of a biometric characteristic that with a very high probability
  - will be same for the right user being measured
  - will be different for an imposter
- Add a secret value (biometric data is not secret) and process these two values (e.g., hash function)
- This approach (with some twists) has been suggested in dozens of papers. The issue is that the secret value and not the biometric data is the critical basis of the key.

Fruitful approaches

- D. Wheeler – Error correction
  - Deriving faultless data ("keys") from faulty data (analogue measurements, e.g. biometric ones)
  - Error-correction bits involved in the protocol, both parties must possess results of a measurement
  - Both sender and receiver have same bit-string after the protocol execution
- F. Monrose et al. – Thresholding
  - Spoken passphrase – secrecy of the passphrase and its speech pattern (similar work for keystroke dynamics)
  - Based on secret sharing (*t* of *n* shares) operations with this secret (key) are enabled

Aspects of real generation

- Major problems
  - Biometrics are not secret!!!
    - Can/should secret be added?
    - How do we protect, store, and use that secret?
    - What are the chances of exhaustive search?
  - Key-space
    - Limited by measurable characteristics
    - Probability of different values?
- Minor problems
  - Compromised key – key change?
  - Organ damaged – key loss?
  - Implementation issues, e.g. dependence on the reader

The building blocks

- Secure primitives necessary, yet not sufficient
- Playing it safe – precise specification of
  - what shall and shall not be done
  - before, during and after the protocol run
  - with restrictions on use of a given protocol
- Assumptions of critical importance!
- Protocol analysis tools useful, yet not foolproof and also not designing protocols