Key Establishment Protocols

Prof. Bart Preneel
COSIC – KU Leuven - Belgium
Firstname.Lastname(at)esat.kuleuven.be
http://homes.esat.kuleuven.be/~preneel
December 2014

1

Goals

- Understand properties of protocols for key establishment and entity authentication
- Understand flaws in protocols
- Analyze new protocols

2

Key management

- generation
- registration/certification
- **establishment (this chapter)**
- installation
- usage
- storage/archiving
- escrow
- destruction/revocation

most expensive and most complex aspect of practical cryptography

3

Outline

- definitions & properties
- key transport with symmetric cryptography
- key transport with asymmetric cryptography
- key agreement with asymmetric cryptography
- analysis of protocols

Based on chapter 12 of Handbook of Applied Cryptography

4

Definitions

- A (cryptographic) protocol is 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.
- Key establishment is a process or protocol whereby a shared secret becomes available to two or more parties.
  - key transport
  - key agreement
  - static (always same key): pre-distribution
  - dynamic
  - with or without a trusted third party

5

Use of session keys

- Session keys are (typically temporary) keys, that are distributed with a key establishment protocol (ephemeral secret).
- Motivation:
  - limit available ciphertext for 1 key
  - limit exposure in the event of a key compromise
  - avoid long-term storage of a large number of distinct keys (in a network with many nodes)
  - create independence across communication sessions or applications

6

Definitions: authentication

Important!

- entity authentication: one is corroborated of the identity of another party, and of the fact that this party is alive (active) during the protocol
- data origin authentication: one is corroborated of the source of data
- (implicit) key authentication: one party is assured that no other party aside from a specifically identified second party has the possibility to determine the secret key
- key confirmation: one party is assured that a second (possibly unidentified) party has possession of a particular secret key
- explicit key authentication: one is convinced that another identified party possesses a given secret key (= implicit key authentication + key confirmation)

note: a connection-less view of the world!! (vs. connection-oriented)

7

Classification of simple protocols

- (entity) authentication (or identification)
- key establishment
- authenticated key establishment is a key establishment protocol that offers (implicit) key authentication.

8

Timestamps and nonces

time stamp

- detect repetition (within a given time window)
- detect forced delay
- limit privileges in time

- approach: information of the local clock is cryptographically protected and sent to the other parties.
  - notation: t_X

nonce = value that is used only once (no more than once).

- approach: nonce is sent to the other party; this value is then cryptographically integrated into the answer
- two types:
  - serial number n_X
  - random number r_X

9

Protocol properties

1. which authentication (entity, key confirmation, key authentication)
2. unilateral or mutual authentication
3. guaranteed 'freshness' of the key
4. key control
5. efficiency: number of messages, number of bytes transmitted, computations
6. conditions for third party (on-line, off-line)
7. type of certificates
8. proof of key exchange (non-repudiation)

10

The opponent (1)

Assumptions:

- the cryptographic algorithms (encryption, signature, MAC) are considered to be unbreakable
- (encryption = envelope, also providing data origin authentication!?)

Capabilities

- active or passive network access
- outsider or insider (permanent/temporary)
- goals
  - obtain session key
  - impersonation
  - mislead parties about the parties they are communicating with

11

The opponent (2)

special problems

- leakage of long term key material compromises previous session keys (lack of historical secrecy or no (perfect) forward secrecy)
- leakage of a session key compromises future session keys or allows for future impersonation (vulnerable to known key attack)

These definitions are confused very often

12

Key transport based on symmetric cryptography

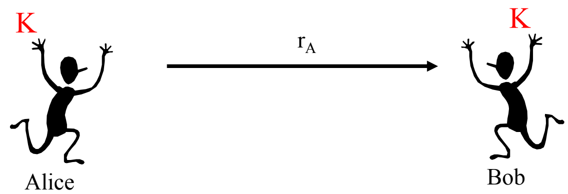
- point to point: key transport with encryption or with a MAC
- with third party (server): Kerberos

- encryption (block cipher)
- MAC (Message Authentication Code)

- (perfect) forward secrecy hard – need to update the key with a one-way function after every transaction

13

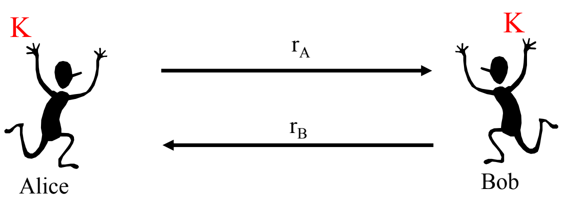
Point to point key derivation with a MAC



- session key = $MAC_K(r_A)$
- implicit key authentication
- no protection against reuse

14

Point to point key derivation with a MAC

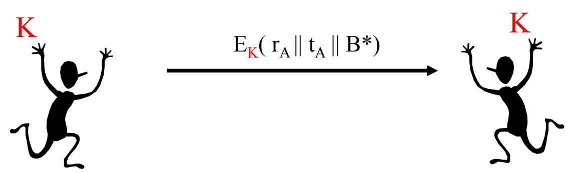


- session key = $MAC_K(r_A \| r_B)$
- implicit key authentication
- protection against reuse

|| denotes concatenation of strings

15

Point to point key derivation with a block cipher and time stamp

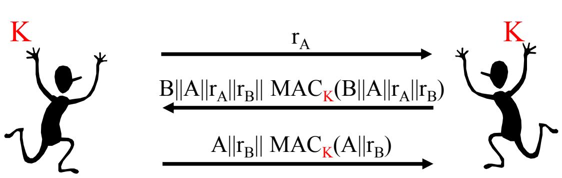


- session key = r_A
- t_A detects delay or repetition within a window
- B prevents reuse on A

the * in B* means that this field is optional

16

Point to point key derivation with a MAC: AKEP2



- session key = $prf_{K'}(r_B)$. Here $K' \neq K$, but K' may be derived from K
- mutual authentication with implicit key authentication
- key confirmation possibly by using the session key to encrypt a known message
- variant with key transport

17

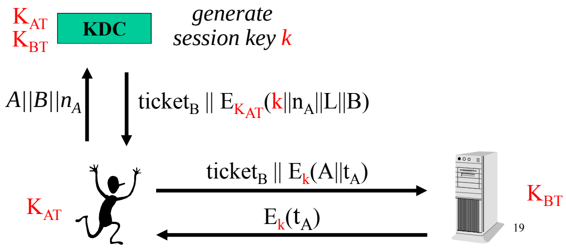
Using a third party

- Trusted Third Party (TTP) assists with key establishment; can also assist with entity/data origin authentication
- symmetric:
  - Key Distribution Center (KDC): generates and distributes session key
  - Key Translation Center (KTC): translates session key
- asymmetric:
  - Certification Authority (CA)

18

Symmetric key distribution with 3rd party (KDC Key Distribution Center) - Kerberos

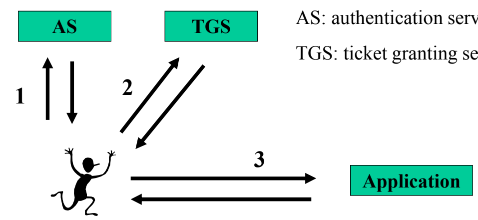
- Alice/Bob shares a long term secret with KDC: K_{AT}/K_{BT}
- Alice/Bob/KDC have synchronized clocks
- $\text{ticket}_B = E_{K_{BT}}(k \| A \| L)$
- L life time of a ticket – limits validity of a key



19

Kerberos/Single Sign On (SSO)

- Alice's long term key K_{AT} is derived from a password P
- Alice stores $E_{K_{AT}}(k\|n_A\|L\|B)$ on her disk for the period L (1 day)
- To avoid one password entry per application: use intermediate server (ticket granting server)



AS: authentication server

TGS: ticket granting server

20

Kerberos/Single Sign On (SSO)

- Kerberos (MIT, project Athena 1987)
  - RFC 1510 (1993) replaced by RFC 4120 (2005)
  - included from Windows 2000 onwards as default entity authentication method (extensions defined in RFC 3244 ``Microsoft Windows 2000 Kerberos Change Password and SetPassword Protocols.")
  - included in MAC OS X
- alternatives (no market success): Kryptoknight (IBM) and Sesame (Siemens/Bull/ICL)
- limitations of Kerberos:
  - still uses passwords: guessing attacks
  - requires modification to application; no authorisation
  - in pre-2005 versions: no authenticated encryption (separate operations)

21

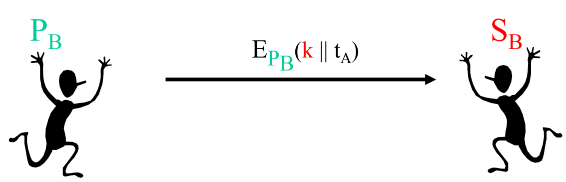
Key transport based on asymmetric cryptography

- without digital signatures
  - time stamp
  - nonce: Needham-Schroeder
- with digital signature
  - time stamp: 3 variants

- point to point, but protecting the authenticity of public keys
- requires CA (Certification Authority) in large systems

22

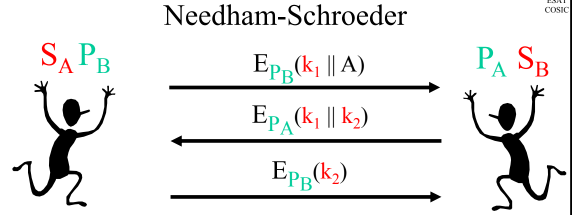
No digital signature; with time stamp



- only implicit key authentication
- 1-pass, suited for e-mail
- t_A prevents replay

23

No digital signature; with time stamp (2) Needham-Schroeder

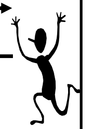


- session key = $\text{hash}(k_1 \| k_2)$

24

Triangle attack on Needham-Schroeder

S_A P_E $E_{P_E}(k_1 \| A)$ → P_A P_B S_E P_A S_B



$E_{P_B}(k_1 \| A)$ →

← $E_{P_A}(k_1 \| k_2)$

← $E_{P_A}(k_1 \| k_2)$

$E_{P_E}(k_2)$ →

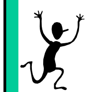
$E_{P_B}(k_2)$ →

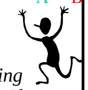
- connection-less: Alice and Bob are not misled about 'connections' (as there get the answers from the right persons)
- Alice is misled as she believes k_1 and k_2 are secrets shared with Eve

25

Key transport using RSA: X.509

S_A P_B *generate k* P_A S_B





$t_A^* \| E_{P_B}(A \| k) \| Sig_{S_A}(B \| t_A^* \| E_{P_B}(A \| k))$ →

decrypt using SKB and verify using PKA

Mutual: B can return a similar message including part of the first message

Problem (compared to D-H/STS): lack of **forward secrecy**

If the long term key *SKB* of Bob leaks, all past session keys can be recovered!

Key agreement with asymmetric cryptography

- Diffie–Hellman & variants
- Station to Station

- all calculations are done modulo a large (safe) prime p with generator α

27

Diffie-Hellman

Alice		Bob
generate x, *compute* α^x	α^x →	*generate y*, *compute* α^y
	← α^y	
compute $k=(\alpha^y)^x$		*compute* $k=(\alpha^x)^y$

- how does Alice know that she shares this secret key ***k*** with Bob?
- answer: Alice has no idea at all about who the other person is! The same holds for Bob
- no authentication or key confirmation

28

Diffie-Hellman variants

a = x, b = y fixed; α^a and α^b public.

- mutual implicit key authentication
- disadvantage: session key constant

only b = y fixed; α^b public (≃ ElGamal encryption)

- only 1 party has implicit key authentication

29

Station to Station protocol (STS)

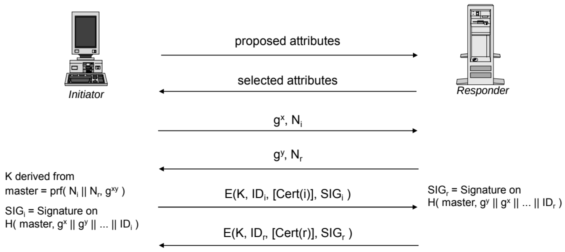
- The entity authentication problem can be fixed by adding digital signatures
- This protocol plays a very important role on the Internet (under different names)

Alice		Bob
choose x	α^x →	*choose y*
$k=(\alpha^y)^x$	← α^y	$k=(\alpha^x)^y$
	$E_k(SigA(\alpha^x \| \alpha^y))$ →	
√*SigB*	← $E_k(SigB(\alpha^y \| \alpha^x))$	√*SigA*

30

IKE - Main Mode with Digital Signatures

ESAT COSIC



H is equal to prf or the hash function tied to the signature algorithm (all inputs are concatenated)

31

STS properties

ESAT COSIC

- mutual explicit key authentication
- mutual entity authentication
- mutual key confirmation
- anonymity (unless certificates are exchanged in the beginning)
- (perfect) forward secrecy
- no problem if k leaks

32

Protocol analysis

ESAT COSIC

- in order to analyze a protocol, or in order to prove its security, one needs the following information:
  - protocol specification (messages AND actions)
  - goals
  - assumptions and initial state

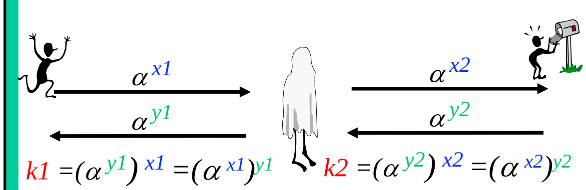
1. Ad hoc: study attack strategies
   - person-in-the-middle
   - reflection attack
   - 'interleaving' attack
2. Information-theoretic
3. Complexity theoretic: universal composability
4. Formal methods, logics,…

33

Person-in-the middle attack on Diffie-Hellman

ESAT COSIC

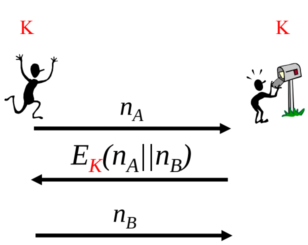
- Eve shares a key *k1* with Alice and a key *k2* with Bob
- Requires *active* attack



34

A simple protocol

ESAT COSIC

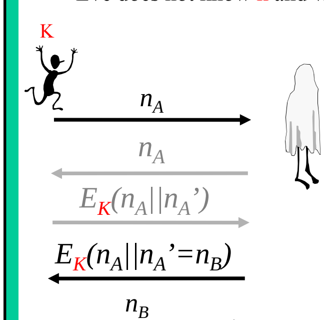


35

Reflection attack

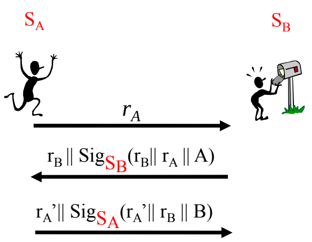
ESAT COSIC

- Eve does not know k and wants to impersonate Bob



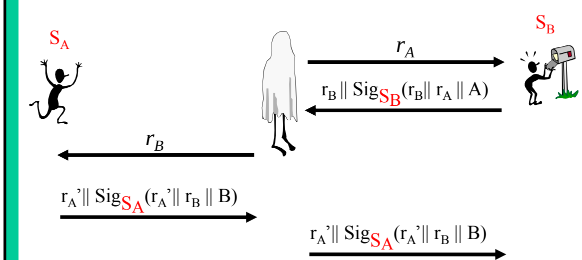
36

Another simple protocol



37

Interleaving attack



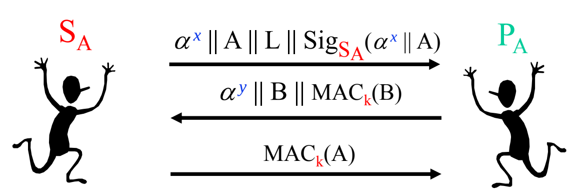
38

Conclusions

- Properties of protocols are subtle
- Many standardized protocols exist
  - ISO/IEC, IETF
- Difficulty: which properties are needed for a specific application
- Rule #1 of protocol design: **Don't**
  - not even by simplifying existing protocols

39

Exercise



- session key $k = hash(k' \| A \| B)$ with $k' = \alpha^{xy}$
- L = life time of session key in minutes

40