Calculi and Models for Security

Chiara Bodei, Massimo Bartoletti,
Pierpaolo Degano, Gian-Luigi Ferrari,
and Roberto Zunino

Dipartimento di Informatica,
Università di Pisa, Italy

Pisa, 17-28 Settembre 2007

Security

Too wide a concept

- systems, both H/W (critical systems, ...)
  and S/W (protection of resources, ...):
  language based security

  in the large
- communications, links and messages: protocols

  in the small

Not a black/white notion: trade off between cost (time, computational efforts, ...) and benefits (which ones?)

A formal approach

Long standing problem, (nets of) computers make it worse.

Tools and solutions

- ad hoc — firewall, anti-virus ...: pragmatics
- formal — models, analysers ...: from art to engineering

We advocate a formal approach to the construction and verification of secure systems and protocols, within a LINGUISTIC framework

Security — naïvely

No attacker (intruder, saboteur) interacting with the principals (parties) of a (distributed) system can alter their behaviour or exploit their resources. Far too strong a notion:

NON-INTERFERENCE

And preliminarily:

- who are principals and attackers? what can they do?
- how do they interact? what is a behaviour?
- how are systems and protocols specified?

Weaker properties

1. Secrecy (confidentiality): data, typically msg, only read by authorised people (receiver)
2. Auhentication: sender/receiver are the intended ones
3. Integrity: sensible data not fraudulently modified
4. Accountability (non repudability): one cannot deny the actions performed
5. Availability (antynom: denial of service): one can always access to the resources if authorised
6. Access control: authorised people only access critical resources

A different look

CIA — classical security notions

- 1 and 3: require protecting msg and communication links
- 2 and 4: offer protection to principals (active entities)
- 5 and 6: concer protecting resources (passive entities), e.g. servers, clients

The hostile environment

Principals and attackers seat in the eather.
The communication medium is unreliable: the attacker has full control over the network and can

- intercept
- manipulate
- redirect
- forge

msgs, with the only limitations of its computational power and possible trusted entities involved

Roughly and naïvely

Public network and TCP/IP

- a msg is split in packets to be sent in a row
- routing in multi-hops – two packets may follow different routes and arrive in unexpected order
- eveybody can stop a packet, inspect/change its contents

PROTOCOLS

- abstractly specify the sequence of actions and controls that implement msg exchange
- must be resistant to the hostile environment

Common false belief

Protecting the msg only suffices for

- avoiding eavsdropping
- ensuring CIA

Many centuries of deep studies for making the reader unable to deduce the contents of a msg

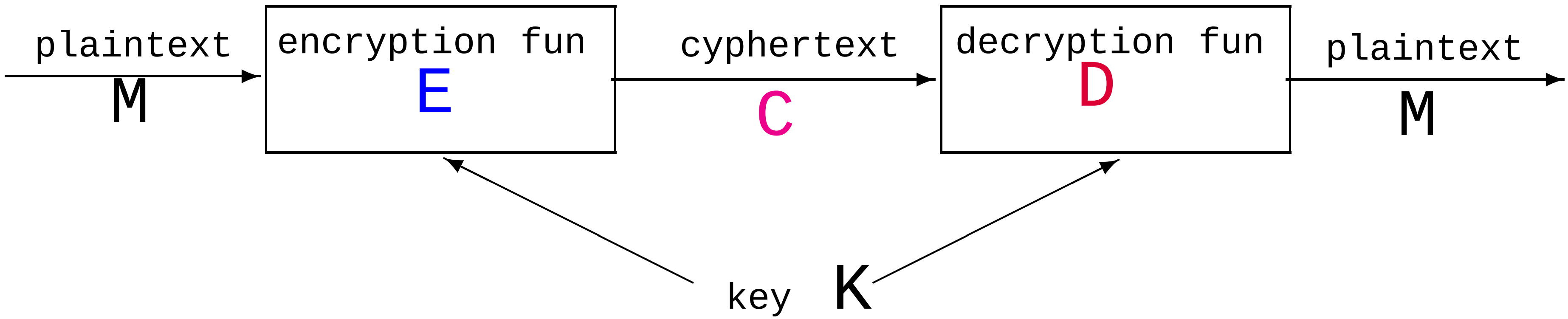
- steganography
- cryptography

(Very little on these fascinating techniques, basis for the so-called computational approach to security)

Obfuscation or Secrecy

- steganography = covert writing
  (invisible ink, interleaving of letters, little graphical alterations, ...)
  msg covert by a secret algorithm
- cryptography = hidden writing
  (Caesar's, symmetric, asymmetric, ...)
  msg hidden by a public algorithm and a secret key
  Much stronger mathematical properties

Symmetric cryptography



C incomprehensible and indistinguishable from another cypertext C' (i.e. cannot tell whether they come from the same msg M of from two different msgs)

Sender/receiver agreement

Principals should agree on:

- Encryption and decryption functions E, D — public encryption schema
- Key K — secret
  If K is unknown, E, D useless:

  perfect encryption assumption

Some notation

Encryption:

$C = E(K, M)$ — cryptotext

(M can be a list $M_1, ..., M_n$)

$= \{M\}_K$ — once fixed E, D.

Abstractly a term of an algebra, not a bit string

Decryption:

`decrypt` C `as` $\{x\}_K$ — an explicit operation

only succeeds if $C = \{M\}_K$ and binds x to M
(perfect encryption: no leakage of portions of M)

1-time pad

Select a key

$$K = p_1, p_2, ..., p_n$$

as long as the elements of the msg

$$M = m_1, m_2, ..., m_n$$

Then $\{M\}_K = p_1 \oplus m_1, p_2 \oplus m_2, ..., p_n \oplus m_n$
and decryption just the same.

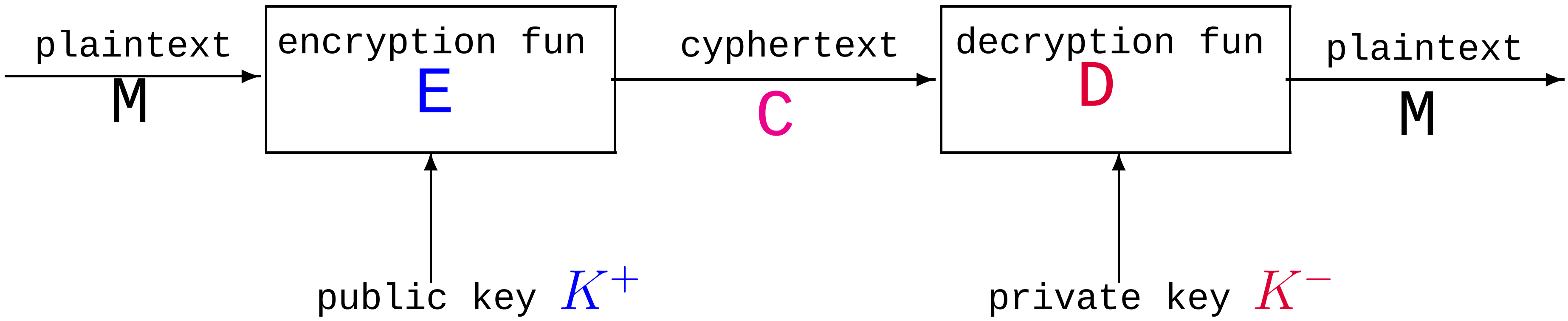
Very long keys, used only *once*, both partners agree on plenty of keys: expensive

Perfect encryption

We assume it (and relax it later) because

- hard to break – often 256 bits suffice: short and good as session keys
- most properties are unaffected by having it or not
- a compromised key may be a disaster (fw secrecy)
- who's responsible for a key?
- keys need to be often changed — secret exchange!!

Asymmetric cryptography



A pair of keys $\langle K^+, K^- \rangle$:

K^+ — public

K^- — private and secret

Asymmmetric public key crypto

Ex. RSA — based on Fermat's little theorem:
$n^{p-1} - 1 \equiv 0 \; mod \; p$ (for p prime, $n \neq 0$)

Diffie-Hellman algorithm for establishing the key:

$$g^a; \; (g^a)^b = g^{a \times b} = (g^b)^a$$

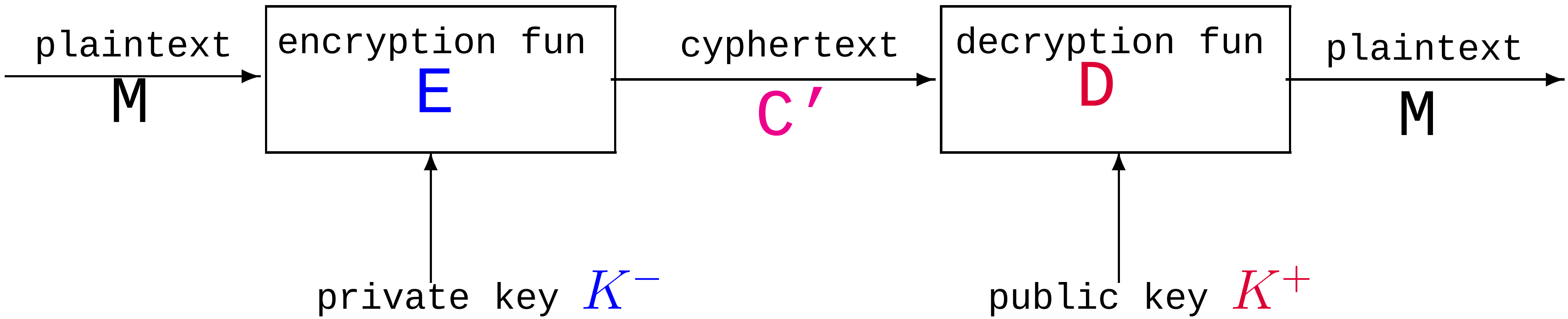
how to recover a, b given $g^{a \times b}$: factorization is hard!

Ex. Elliptic curves — more efficient

Asymmmetric long term keys

- no key distribution! K^+ posted in a trusted place
- a trusted certification authority emits a certificate for a principal: its keys, etc
- not very efficient w.r.t. symmetric cryptography — elliptic not so bad
- good as long term keys for exchanging short term (session) symmetric keys
- good for ...

Signatures



The cyphertext C' now is readable by anyone, as K^+ is public, but only the owner of K^- can encrypt it!

(other, more efficient signature schemata exist)

Protecting msg is not enough

The usage of the cyphertext is of paramount importance!! Need to express it

A bit more formal: (key-exchange) protocol narrations

The A lice and B ob notation (plus I ntruder and trusted S erver):

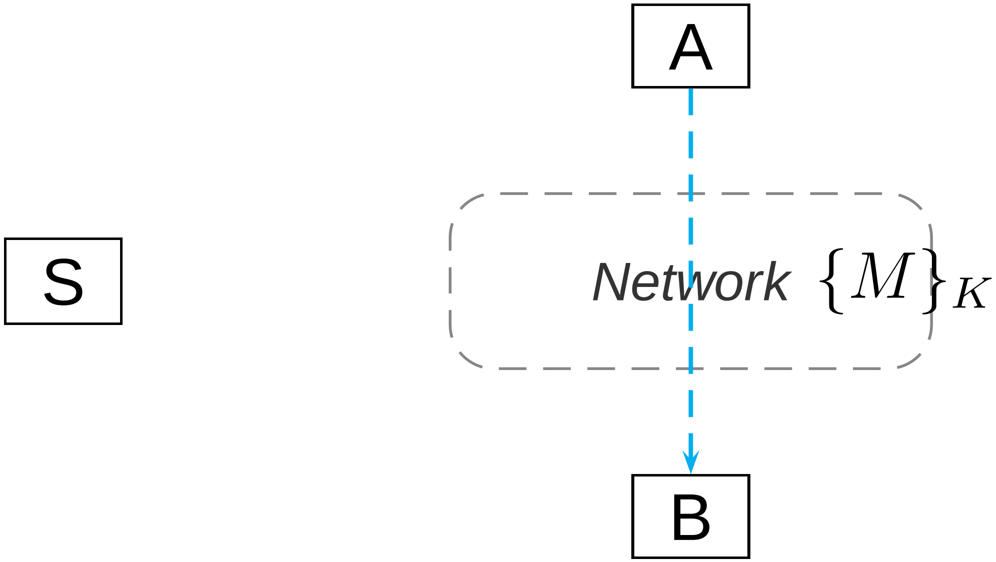
$$A \to S : \ A, B, \{K\}_{K_A}$$

A — sender
S — receiver
$A, B, \{K\}_{K_A}$ — 3-fields msg, K_A shared by A and S

A simple protocol

A key exchange inspired protocol by the Wide Mouthed Frog (K is the short term key, K_A and K_B the long term keys):

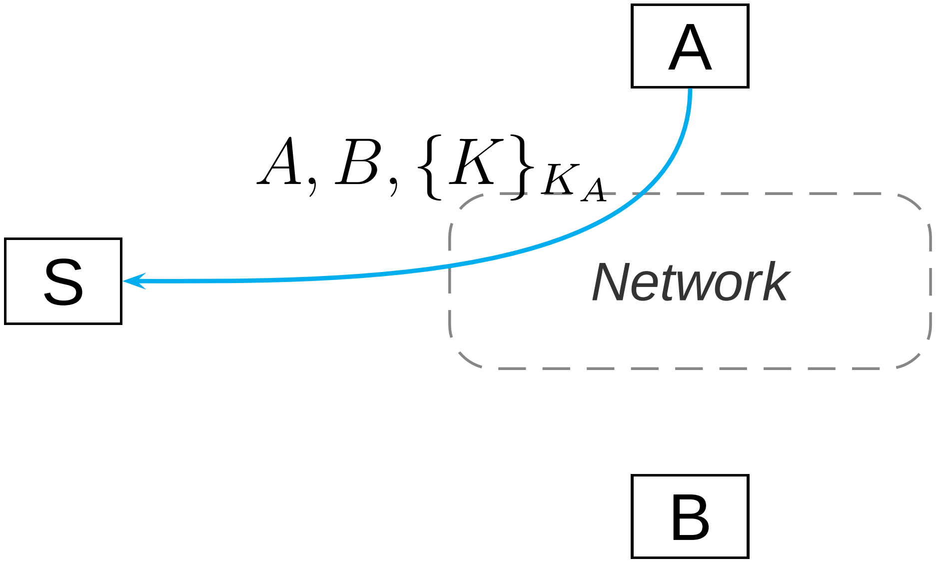
1. $A \rightarrow S: \quad A, B, \{K\}_{K_A}$
2. $S \rightarrow B: \quad A, \{K\}_{K_B}$
3. $A \rightarrow B: \quad \{M\}_K$



A simple protocol

A key exchange inspired protocol by the Wide Mouthed Frog (K is the short term key, K_A and K_B the long term keys):

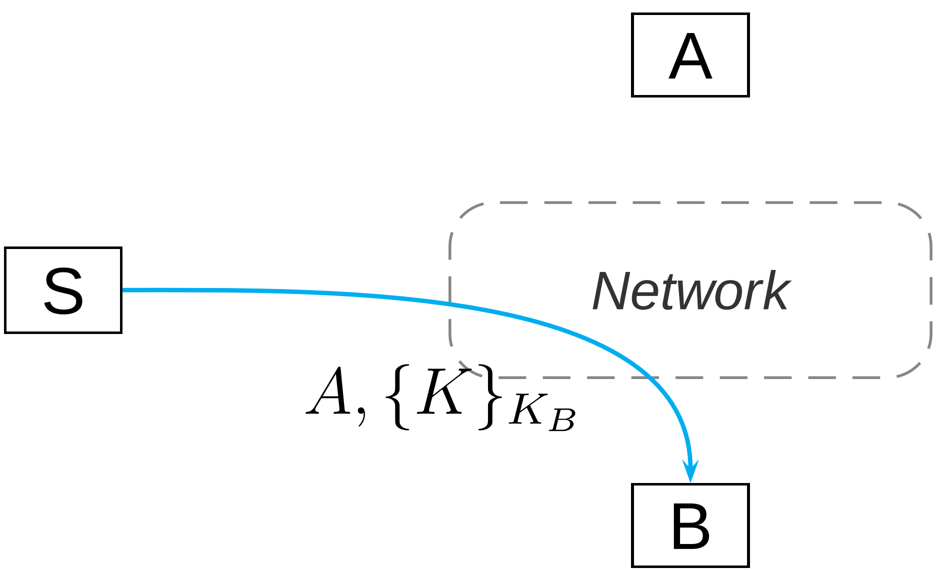
1. $A \longrightarrow S: \quad A, B, \{K\}_{K_A}$
2. $S \longrightarrow B: \quad A, \{K\}_{K_B}$
3. $A \longrightarrow B: \quad \{M\}_K$



A simple protocol

A key exchange inspired protocol by the Wide Mouthed Frog (K is the short term key, K_A and K_B the long term keys):

1. $A \rightarrow S: \quad A, B, \{K\}_{K_A}$
2. $S \rightarrow B: \quad A, \{K\}_{K_B}$
3. $A \rightarrow B: \quad \{M\}_K$



First type of attack: secrecy

By $I(X)$ we mean "I pretends to be X."

$$
\begin{array}{lll}
1'. & A \rightarrow I(S): & A, B, \{K\}_{K_A} \\
1''. & I(A) \rightarrow S: & A, I, \{K\}_{K_A} \\
2. & S \rightarrow I: & A, \{K\}_{K_I} \\
3. & A \rightarrow I: & \{M\}_K
\end{array}
$$

I exploits the server to get the key.

Second type: authentication attack

$$
\begin{array}{lll}
1'. & A \longrightarrow I(S): & A, C, \{K\}_{K_A} \\
1''. & I(A) \longrightarrow S: & A, B, \{K\}_{K_A} \\
2. & S \longrightarrow B: & A, \{K\}_{K_B} \\
3'. & A \longrightarrow I(B): & \{M\}_K \\
3''. & I(A) \longrightarrow B: & \{M\}_K
\end{array}
$$

A talks to B not to C as intended.

Third type: reply attack

$$
\begin{array}{lll}
1. & A \rightarrow S : & A, B, \{K\}_{K_A} \\
2. & S \rightarrow I(B) : & A, \{K\}_{K_B} \text{ —} I \text{ stores the ticket} \\
2'. & I(S) \rightarrow B : & A, \{K\}_{K_B} \\
3. & A \rightarrow I(B) : & \{M\}_K \\
3'. & I(A) \rightarrow B : & \{M\}_K \\
2''. & I(S) \rightarrow B : & A, \{K\}_{K_B} \\
3''. & I(A) \rightarrow B : & \{M\}_K
\end{array}
$$

B receives twice the same M, maybe paying twice the bill. The second ticket is NOT fresh.

Freshness

To ensure freshness of msgs, a single fresh component of an encryption sufficies.

- tickets come with expiring date
- a random number, used only once, called nonce is created by principals and shared as a secret between them

Typing attacks

From Otway-Rees protocol:

$A \to B : \;\; C, \{n, m, A, B\}_K$ — n, m nonces

msg intercepted by I; and while A expects from B

$B \to A : \;\; C, \{n, K_{new}\}_K$

receives instead from I

$I(B) \to A : \;\; C, \{n, m, A, B\}_K$

which is a big trouble if the length of K_{new} equals that of m, A, B — mismatch in decrypting, easy with bitstrings

Other kinds of attacks

- various manipulations on certificates
- confusion of principals in parallel sessions
- man-in-the-middle
- (many cryptographic attacks on bitstrings – neglected here)

Man-in-the-middle

Il Lupo usa Cappuccetto Rosso per farsi aprire dalla Nonna e la Nonna (mangiata) per mangiarsi anche Cappuccetto, ma il Cacciatore ...

Lose at most one out of two simultaneous chess matches against two international masters

The Needham-Schröder protocol has a flaw, discovered 17 years later, by specifying it in CSP and mechanically analysing it [Loewe 1995]

Need of deep formalization and formal reasoning!!

Needham-Schröder shared key

1. $A \to S: \quad A, B, n_A$ — the nonce is fresh!
2. $S \to A: \quad \{n_A, B, K, \{K, A\}_{K_B}\}_{K_A}$
3. $A \to B: \quad \{K, A\}_{K_B}$
4. $B \to A: \quad \{n_B\}_K$ — n_B is a shared secret between A and B
5. $A \to B: \quad \{n_B + 1\}_K$

The shared secret authenticates A and B each other.

(The original version with public key later on.)

The attack

$$
\begin{array}{lll}
1. & A \to S: & A, B, n_A \\
2. & S \to A: & \{n_A, B, K, \{K, A\}_{K_B}\}_{K_A} \\
3'. & A \to I(B): & \{K, A\}_{K_B} \\
3''. & I(A) \to B: & \{K_{old}, A\}_{K_B} \text{ — } \{K_{old}, A\}_{K_B} \text{ is an old ticket} \\
4'. & B \to I(A): & \{n_B\}_{K_{old}} \\
5'. & I(A) \to B: & \{n_B + 1\}_{K_{old}}
\end{array}
$$

B is fooled to accept K_{old} old as fresh, even if I does not understand $\{K_{old}, A\}_{K_B}$.
Additionally, if K_{old} is compromised (e.g. by off-line cryptoanalysis) I gets information from B.

Needham-Schröder public key

1. $A \rightarrow S: \quad A, B$
2. $S \rightarrow A: \quad \{B, K_B^+\}_{K_S^-}$
3. $A \rightarrow B: \quad \{A, n_A\}_{K_B^+}$
4. $B \rightarrow S: \quad B, A$
5. $S \rightarrow B: \quad \{A, K_A^+\}_{K_S^-}$
6. $B \rightarrow A: \quad \{n_A, n_B\}_{K_A^+}$ — repair $B \rightarrow A : \{B, n_A, n_B\}_{K_A^+}$
7. $A \rightarrow B: \quad \{n_B\}_{K_B^+}$

A runs in parallel

with I

$\vdots$

3. $A \rightarrow B: \quad \{A, n_A\}_{K_B^+}$

$\vdots$

$\vdots$

6. $I \rightarrow A: \quad \{n_A, n_B\}_{K_A^+}$

7. $A \rightarrow I: \quad \{n_B\}_{K_B^+}$

$\vdots$

with B & I

$\vdots$

$\cdots$

3. $A \rightarrow I(B): \quad \{A, n_A\}_{K_B^+}$

3′. $I(A) \rightarrow B: \quad \{A, n_A\}_{K_B^+}$

$\vdots$

6. $B \rightarrow I(A): \quad \{n_A, n_B\}_{K_A^+}$

$\vdots$

$\vdots$

7. $I(A) \rightarrow B: \quad \{n_B\}_{K_B^+}$

I exploits A to decrypt $\{n_A, n_B\}_{K_A^+}$, so B thinks I is A