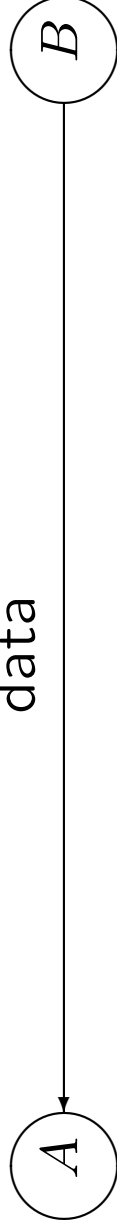
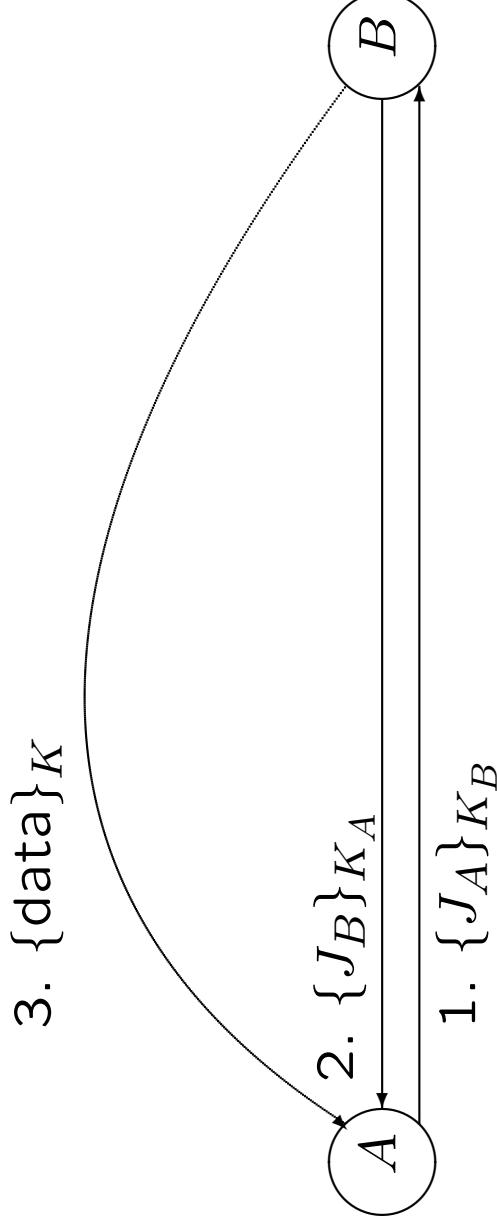


Two Concepts of Authenticity

A specification



A proposed “secure” implementation



- K_A and K_B are the public keys of A and B .
- J_A and J_B are two fresh random numbers.
- $\{J_A\}_{K_B}$ and $\{J_B\}_{K_A}$ are their public-key encryptions.
- H is a one-way hash function.
- $K = H(J_A, J_B)$ is a shared key.
- $\{\text{data}\}_K$ is the shared-key encryption of data.

An informal analysis

If A follows the protocol then she is assured that the shared key [...] is not known to anyone except B (though A does not have the assurance that B knows the key). And analogously for B .

(Hugo Krawczyk)

Two interpretations

An “authenticated” message M from B to A may be used in at least two distinct ways:

- A may hold B responsible for M .

E.g., if A is a file server, B is a client, and M is a request to delete B 's file f , then A deletes f .

- A may give credit for M to B .

E.g., when A is running a contest and receives M , A may give credit for this entry to B

These two uses are sharply different.

Some protocols are fine for one use but not the other.

Responsibility and credit

Responsibility and credit may belong to high-level entities (network services, people, groups, companies).

They can be transferred.

They may be attributed by a variety of parties (on-line participants, off-line judges).

They may not coincide with origin.

They may not coincide with knowledge (of a key, of a message).

Example: signing a public key

A creates a short-term key pair (K, K^{-1}) ,
sends the public key K to B,
signs K with the long-term secret key K_A^{-1} ,
then uses K^{-1} for signing further messages:

Message 1 $A \rightarrow B: A, B, \{K, A, B, T\}_{K_A^{-1}}$
Message 2 $A \rightarrow B: A, B, \{\{M\}_{K^{-1}}\}_{K_B}$
⋮

A may take responsibility for messages signed with K^{-1} .

A might even want credit for those messages.

An attack?

Message 1 $A \rightarrow B : A, B, \{K, A, B, T\}_{K_A^{-1}}$
(intercepted by C)

Message 1' $C \rightarrow B : C, B, \{K, C, B, T\}_{K_C^{-1}}$

Message 2 $A \rightarrow B : A, B, \{\{M\}_{K^{-1}}\}_{K_B}$
(intercepted by C)

Message 2' $C \rightarrow B : C, B, \{\{M\}_{K^{-1}}\}_{K_B}$

So credit to A or C is not justified.

Responsibility for A or C is reasonable.

A stronger protocol

To establish credit, A may sign its own name with K^{-1} ,
for example as in:

Message 1 $A \rightarrow B: A, B, \{K, A, B, T\}_{K_A^{-1}}$

Message 2 $A \rightarrow B: A, B, \{\{A, M\}_{K^{-1}}\}_{K_B}$

(A may not actually see or have K^{-1} .)

Example: encrypting a session key

A transmits a session key K to B , encrypting K under B 's public key K_B , and including the name A along with K :

Message 1 $A \rightarrow B : \{A, B, K\}_{K_B}$

Message 2 $A \rightarrow B : \{M\}_K$

Message 3 $B \rightarrow A : \{M'\}_K$

This protocol is adequate for applications that require responsibility of B for M' .

It may also be adequate for applications that require credit to A for M .

Example: making a key from shares

Recall:

Message 1 $A \rightarrow B : \{J_A\}_{K_B}$

Message 2 $B \rightarrow A : \{J_B\}_{K_A}$

$K = H(J_A, J_B)$

Example: making a key from shares (cont.)

An attack?

Message 1 $A \rightarrow B : \{J_A\}_{K_B}$

Message 1' $B \rightarrow C : \{J_A\}_{K_C}$

Message 2 $C \rightarrow A : \{J_C\}_{K_A}$

$$K = H(J_A, J_C)$$

C believes that K is shared with A .

A believes that K is shared with B .

B does not get K .

Another informal analysis

B should not get credit for messages under *K*.

B may be held responsible for messages under *K*.

A stronger protocol

It is prudent to use the names of principals, as in:

Message 1 $A \rightarrow B : \{J_A\}_{K_B}$

Message 2 $B \rightarrow A : \{J_B\}_{K_A}$

$K = H(J_A, J_B)$

:

$B \rightarrow A : \{A, B, M\}_K$

or:

Message 1 $A \rightarrow B : \{J_A\}_{K_B}$

Message 2 $B \rightarrow A : \{J_B\}_{K_A}$

$K = H(A, B, J_A, J_B)$

In sum

Authentication may yield responsibility, credit, or both.

Responsibility is essential.

- It is the basis of access control.
- It is compatible with delegation, with rules such as:

If *A* says that *C* speaks for *A*,
then *C* does speak for *A*.

Authentication protocols may not need to be concerned with credit.

- It can be left for higher-level communication.
- But establishing credit may contribute to robustness.