Analysis and design of cryptographic protocols

Gianluca Dini
Dept. of Ingegneria dell'Informazione
University of Pisa
Email: gianluca.dini@unipi.it
Version: 05/05/2025

1

Preliminaries

ESTABLISHING A SESSION KEY

2

- You have a protocol and you'd like to see how it works and see the guarantees it provides.
  This method is not a silver bullet, but it is an educated way to reason over a protocol.
  This is the BAN Logic. Logic is based on 2 main concepts:
  1. Belief
  2. Action

Consider protocol in which A and B have to establish a key. The initial assumptions are called Beliefs (ex. A believes B's pubk is ...)
Then they do actions (ex. they send messages). A message has some contents, that can be encrypted (partially), signed etc. Given messages we increase our beliefs. As a consequence of actions, I increase my set of Beliefs. At the end of protocol, my beliefs should contain my objective.

NOTE: Logic either tells you (at the end of the proof) nothing, or it stops somewhere, and that gives you hint on bad things. So logic is useful to find problems, not prove correctness. Logic cannot prove protocol is wrong, but if you cannot prove it is correct be suspicious of it.
Logic works at specification level (so still room for bugs for implementation or design choices (PRBG)).

FORMALISM:

1. $P \equiv X$ — P believes X (X is a statement, P process, user etc.) to be true.
2. $P \triangleleft X$ — P sees X (abstraction of reading from file or receiving message).
3. $P \mid\sim X$ — P once said X (" " sending a message/writing file)
4. $P \Rightarrow X$ — P controls X / has jurisdiction over X (if P tells me X is true then I believe that. Used for assumptions especially for trusted 3rd parties)
5. $\#(X)$ — X is fresh
6. $P \overset{K}{\leftrightarrow} Q$ — K is a shared key between P and Q.

7. $P \overset{K}{\rightleftharpoons} Q$ — K is a shared secret between P and Q (to distinguish keys and pwds, for example)

8. $\overset{K}{\mapsto} P$ — K is P's publK

9. $\langle X \rangle_Y$ — X has been combined with Y (for ex: X xor Y, X and Y as input of hash)

10. $\{X\}_K$ — Encryption of X by means of K.

11. $\{X\}_{K^{-1}}$ — X has been signed by means of private key.

Using those formalisms, you can build statements:

- $A \equiv \#(Na)$: Alice believes Na is fresh.
- $A \equiv A \overset{K}{\leftrightarrow} B$: Alice believes she shares a secret key w/ Bob
- $T \equiv A \overset{K}{\leftrightarrow} B$: Trusted 3rd ply "
- $A \equiv T \Rightarrow A \overset{K}{\leftrightarrow} B$: Alice believes that T controls the shared key building and distr. process of shared key.
- $A \equiv T \Rightarrow \#(A \overset{K}{\leftrightarrow} B)$: Alice believes that T controls freshness of session key

This is a good way to explicitly state what you need.

SIMPLIFYING ASSUMPTIONS

NOTE: Logic divides time in PRESENT and PAST. We are interested to see how the protocol works in the present.

NOTE: beliefs assumed in the past executions of protocol may not be true in present.

- Logic has been extended to take into account time as an explicit variable, but then it becomes very complicated.

NOTE: Beliefs achieved in the present remain valid in the current execution of protocol.

Logic works on 3 postulates:

1. Message meaning rule:

$$\frac{P \equiv Q \overset{K}{\leftrightarrow} P,\ P \triangleleft \{X\}_K}{P \equiv Q \mid\sim X}$$

Suppose that P sees message X encrypted through K, and P believes Q and P share a secret, then P believes Q said X. We say nothing and have no info on whether Q said X in past executions.

2) $\dfrac{P \mid\equiv Q \overset{K}{\rightleftharpoons} P,\ P \triangleleft \langle X \rangle_K}{P \mid\equiv Q \mid\sim X}$

- If P sees message X combined with K, and P believes Q and P share a secret K, the P believes Q said X.

3) $\dfrac{P \mid\equiv \overset{K}{\longmapsto} Q,\ P \triangleleft \{X\}_{K^{-1}}}{P \mid\equiv Q \mid\sim X}$

- If P receives X signed with privK and knows K as pubK of Q, then believes Q said X (abstracting a lot of details like malleability and that ciphers are secure, like algorithms).

2. NONCE VERIFICATION RULE

if P believes X come from Q, and P believes X is fresh

$\dfrac{P \mid\equiv Q \mid\sim X,\ P \mid\equiv \#(X)}{P \mid\equiv Q \mid\equiv X}$

then P believes Q believes X (now, in this protocol execution).

3. JURISDICTION RULE

if P believes that Q believes X, P believes that Q controls X

$\dfrac{P \mid\equiv Q \mid\equiv X,\ P \mid\equiv Q \Rightarrow X}{P \mid\equiv X}$

then P believes X

How do we use this: start from what is called a real protocol:
Assume for instance in this protocol at message 4:

$$M4 \quad A \to B : \{A, K_{AB}\}_{K_B}$$

[INTERPRETATION: Bob receives info that K_{AB} is shared secret]

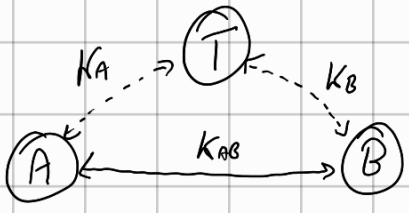
- What does this mean? We can use formalisms to specify meaning to move from real protocol to idealised protocol.

$$M4 \quad A \to B : \{A \xleftrightarrow{K_{AB}} B\}_{K_B}$$ IDEALISED PROTOCOL

Once you have the idealised protocol, you need to specify assumptions. So for instance we used K_C here that can be key shared with trusted third party, you should formalise it.

1. Take real protocol, idealise it, start from assumptions, apply postulates and see if you get your objectives.

NEEDHAM-SCHROEDER PROTOCOL (key distrib. protocol based on symmetric KE).
Assumption: Alice and Bob share a secret with a trusted third party. Objective: establish session key.



[A might be user, B server] [if T is not working / compromised, sys is not working / compromised].

REAL PROTOCOL:

M1 $A \to T : A, B, N_a$ Alice sends message to T saying "I'm alice, want to communicate with Bob, this is a Nonce.

M2 $T \to A : E_{K_A}(N_a, B, K_{ab}, E_{K_B}(K_{AB}, A))$ T responds with ciphertext of this.
↳ Bob's identification.

Alice decrypts and retains N_a, B, K_{ab}, and the encrypted value to forward to Bob.

M3 $A \rightarrow B: E_{K_B}(K_{AB}, A)$

M4 $B \rightarrow A: E_{K_{AB}}(N_B)$

M5 $A \rightarrow B: E_{K_{AB}}(N_B - 1)$

Bob sends A encrypted of another Nonce

With M4 Bob informs A that he holds session key, with M5 A does that too.

What now? State assumptions, define obtained protocol, define objective of protocol.

- Objective: at end of protocol both A and B should believe K_{AB} is session key.

OBJECTIVES

1. $A \equiv A \overset{K_{AB}}{\leftrightarrow} B$ Alice believes K_{AB} is session key to talk to Bob — KEY AUTHENTICATION
2. $B \equiv A \overset{K_{AB}}{\leftrightarrow} B$
3. $A \equiv B \equiv A \overset{K_{AB}}{\leftrightarrow} B$ Alice believes also that K_{AB} is in Bob's hands... — KEY CONFIRMATION OR DIRECT COMMUNICATION
4. $B \equiv A \equiv A \overset{K_{AB}}{\leftrightarrow} B$ ... and vice versa

EVERY ESTABLISHMENT PROTOCOL SHOULD GUARANTEE AT LEAST KEY AUTH.

Note: Nonce means something never used before (freshness). Obtained through random numbers, counters or timestamps. Timestamps require synchronization, counters are predictable.

- ASSUMPTIONS
  - $A \equiv A \overset{K_A}{\leftrightarrow} T$ Alice believes K_A is a shared secret with TTP
  - $T \equiv A \overset{K_A}{\leftrightarrow} T$
  - $B \equiv B \overset{K_B}{\leftrightarrow} T$
  - $T \equiv B \overset{K_B}{\leftrightarrow} T$

  — KEYS

  If we inspect protocol, we see that TTP generates K_{AB}
  - $T \equiv A \overset{K_{AB}}{\leftrightarrow} B$
  - This is the section of assumptions regarding keys.
  - Section on freshness
    - $A \equiv \#(N_A)$
    - $B \equiv \#(N_B)$
    - $B \equiv \#(A \overset{K_{AB}}{\leftrightarrow} B)$ (1)

    — FRESHNESS

- $T \models \#(A \overset{K_{AB}}{\leftrightarrow} B)$ (2)

Section on trust relationship

- $A \models T \Rightarrow A \overset{K_{AB}}{\leftrightarrow} B$
- $B \models T \Rightarrow A \overset{K_{AB}}{\leftrightarrow} B$
- $A \models T \Rightarrow \#(A \overset{K_{AB}}{\leftrightarrow} B)$ (3)

Alice and Bob trust T on key generation, Alice believes T is an authority on K_{AB}.

IDEALIZATION

Message M1 is completely in the clear. It does not contribute to the idealized protocol. From security POV they are not relevant: we are analysing security of protocol

$M2,\ T \rightarrow A : \{N_a, A \overset{K_{AB}}{\leftrightarrow} B, \#(A \overset{K_{AB}}{\leftrightarrow} B), \{A \overset{K_{AB}}{\leftrightarrow} B\}_{K_b}\}_{K_a}$ — TTP is talking to Bob

Idea is explain meaning of M2 by means of formalism introduce. Good practice to write down sentence explaining. Every message should be self explanatory, without relying on context. T is telling A directly to use K_{AB} and B implicitly.

- So, message is encrypted by means of K_a. We can apply first postulate: Alice sees message encrypted by K_a, Alice believes that K_a is the shared key with TTP, so Alice believes that message comes from T.
- $A \models T \mid\sim (N_a, A \overset{K_{ab}}{\leftrightarrow} T, \#(A \overset{K_{AB}}{\leftrightarrow} B), \{A \overset{K_{AB}}{\leftrightarrow} B\}_{K_b})$
- N_a is fresh, everything else is concatenated with N_a, so everything is fresh. So, Alice believes that T at some point said X, Alice believes X is fresh, so Alice believes T is making statement X in current execution of protocol (no replay).

So:

- $A \models T \models (N_a, A \overset{K_{ab}}{\leftrightarrow} T, \#(A \overset{K_{AB}}{\leftrightarrow} B), \{A \overset{K_{AB}}{\leftrightarrow} B\}_{K_b})$
- $A \models T \models A \overset{K_{AB}}{\leftrightarrow} B$ [if I believe everything, I believe the components]

Since Alice considers T authority, then, Alice believes that actually K_{AB} is session key (third postulate):

- $A \equiv A \overset{K_{AB}}{\longleftrightarrow} B$ FIRST OBJECTIVE

Now, about M3:

M3 $A \to B : \{A \overset{K_{AB}}{\longleftrightarrow} B\}_{K_b}$ Message is coming from Alice but it is practically coming from T. Let's apply postulates:

1st postulate: Bob sees message Encrypt. by K_b, Bob believes K_b is shared with TTB, so Bob believes M3 comes from T.

- $B \equiv T \sim A \overset{K_{AB}}{\longleftrightarrow} B$

But there is nothing which proves this message is fresh! We stop. To move on we may add additional assumption. (1) Bob believes K_{AB} is fresh. So, by 2nd postulate given the new assumption:

- $B \equiv T \equiv A \overset{K_{AB}}{\longleftrightarrow} B$

And given 2nd assumption as trust: (third postulate)

- $B \equiv A \overset{K_{AB}}{\longleftrightarrow} B$ SECOND OBJECTIVE [To get K_b it we needed (1). This constitutes a vulnerability: it is always dangerous to assume something generated by someone else is fresh].

Move to M4:

M4 $B \to A : \{N_b, A \overset{K_{AB}}{\longleftrightarrow} B\}_{K_{AB}}$ Bob, by encrypting with K_{AB}, is proving to know K_{AB}.

Apply 1st postulate: 1st objective says that Alice now believes K_{AB} is session key, Alice can conclude M4 comes from Bob.

$A \equiv B \sim (N_b, A \overset{K_{AB}}{\longleftrightarrow} B)$

Proof steps: There is nothing telling Alice message is fresh: N_b has been generated by B and K_{AB} by T. No way to guarantee freshness.

But, K_{AB} generated by TTP, so we could put an specification (2) $T \equiv \#(A \overset{K_{AB}}{\longleftrightarrow} B)$ It is a reasonable assumption. And we put this in M2 too.

To conclude that K_{AB} is actually fresh, we need (3): $A \models T \Rightarrow \#(A \overset{K_{AB}}{\leftrightarrow} B)$.
Alice believes that T is an authority of freshness of K_{AB}.
So given assumption (2), T can state that K_{AB} is fresh.
Assumption (3) guarantees that:
$A \models \#(A \overset{K_{AB}}{\leftrightarrow} B)$

So putting together, we can postulate the two beliefs:
$A \models B \ni (N_b, A \overset{K_{AB}}{\leftrightarrow} B)$
$A \models \#(A \overset{K_{AB}}{\leftrightarrow} B)$
This guarantees that Alice believes K_{AB} is in the hands of Bob in the current execution of protocol.
$A \models B \models A \overset{K_{AB}}{\leftrightarrow} B$ THIRD OBJECTIVE

Now remains MS.
MS $A \to B: \{N_b, A \overset{K_{AB}}{\leftrightarrow} B\}_{K_{AS}}$
So, message is coming from Alice, message is fresh, so Bob:
$B \models A \models A \overset{K_{AB}}{\leftrightarrow} B$ FOURTH OBJECTIVE

So, assumptions added are three:
1. $T \models \#(A \overset{K_{AB}}{\leftrightarrow} B)$ VERY REASONABLE: T can be a server and has the power to generate good keys.

2. $A \models T \Rightarrow \#(A \overset{K_{AB}}{\leftrightarrow} B)$ Reasonable, A already believes T is a authority for generation, so for freshness too it is okay.

3. $B \models \#(A \overset{K_{AB}}{\leftrightarrow} B)$ @ We need to focus more on this. Strong assumption. It was discovered that this is a vulnerability.

So let us suppose adversary was able to obtain one session key K_{AB}' and record the messages M1 – M3 that led to establishment, in particular M3'. What adversary (Oscar's say C) could do is, impersonate Alice and transmit M3' to Bob. Under assumption @, Bob believes of the freshness of message and starts talking to C believing he is talking with A.
Adversary can do this at any time they want. So for adversary replaying M3 is sufficient. The because M3 does not contain freshness proof.

This vulnerability was solved by MIT in 1990s when the first distributed system under Athena Project was created. Solution was timestamp instead of nonces added in M3.
Timestamps are nice, but they assume clocks are synchronized. Which is difficult, on internet you cannot put an upper bound on delays. You can only synchronize up to a certain precision, and you need to keep in mind security. Using timestamps implies assuming secure time synchronization. If you use Nonces by means of random numbers it is easier. But with Bob you have no previous messages sent to Bob.
If you just put a timestamp in M3, we assume A, B, T have clocks synchronized. B can just check if timestamp is recent.
If I put a random number, that must be something that Bob has generated, for him to consider message fresh. But there are no previous messages.

VERY SIMPLE PROTOCOL ON TIME SYNCHRONIZATION

↳ Server with a very good clock. Alice wants to synch her clock

M1 $A \rightarrow S$: A, N_a

M2 $S \rightarrow A$: $\{N_a, T\}_{K_a}$ Server replies specifying the current time with a nonce. N_a implemented as counter. Whenever Alice implements protocol, she increments N_a (predictable).

If we analyze protocol by means of logic, K_a ensures message comes from TTP.

ASSUMPTIONS:

$A \models A \overset{K_a}{\leftrightarrow} S$

$S \models A \overset{K_a}{\leftrightarrow} S$

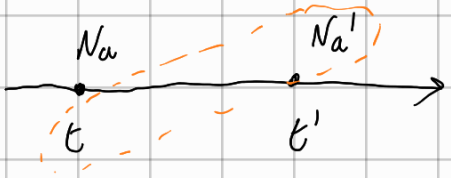
$A \not\vdash \#(N_a)$

$A \not\vdash S \Rightarrow (T)$

Since N_a is predictable, adversary could impersonate Alice:

$M \rightarrow S: A, N_a'$ (N_a' is a future value)

$S \rightarrow M: \{N_a', t\}$



Server hands N_a' with t.

So when Alice uses N_a':

$A \rightarrow S: A, N_a'$

adversary could reply with old message:

$S[M] \rightarrow A: \{N_a', t\}$

↑ MITM or whatever. So Alice sets up her own clock behind.

1. Predictable stuff: pay attention
2. Logic has limit, because protocol by means of logic works smoothly, but there is an implementation vulnerability (N_a = counter).

We will analyze the Otway-Rees Protocol

- It involves a trusted third party (TTP)

PANKO CHEERS

A, B share a Key with T. → Protocol instance identifier (ex: protocol ex # 23). It is fresh for every exec

Outer procedure call — Identifiers — Nonce

- M1 $A \rightarrow B$: $M, A, B, E_{K_A}(N_A, M, A, B)$

We can assume M, N_A are random #, generated by Alice.

- M2 $B \rightarrow T$: $M, A, B, E_{K_A}(N_A, \overset{①}{M}, A, B), E_{K_B}(N_B, \overset{②}{M}, A, B)$ Bob adds this CT.

- TTP can decrypt both ciphertexts.

TTP checks that ① == ②, and also checks identifiers. TTP makes sure A and B believe to be in the same exec. of protocol.

- M3 $T \rightarrow B$: $M, E_{K_A}(N_A, K_{ab}), E_{K_B}(N_B, K_{ab})$
- M4 $B \rightarrow A$: $M, E_{K_A}(N_A, K_{ab})$

} Both can now decrypt and obtain session key

NOTE: Typically when I send a nonce, I expect to receive it back with some thing. $N_A \rightarrow E_K(N_A, X)$. Typical pattern to prove freshness.

1 of the most odd aspects is the fact that N_A is sent in encrypted form. Same for N_B. There is no reason to transmit it in ciphertext.

So, if N_A, N_B are for freshness, why do I also need M? Why N_A is not sufficient? And M is never encrypted, as

- In message M3 and M4 M is not present in the CT.

- Why? N_A and N_B are nonces, but they also serve the purpose of: N_A is a local name for M, local to A. N_B becomes a local name to B, local to B. Encryption is to link N_A to M and N_B to M. New names are linked by means of encryption.

Otway-Rees

- The protocol presents odd aspects
  - Na ed Nb are nonces, they are supposed to prove freshness. Then, why are they encrypted in messages M1 and M2?
  - Why do we need M in addition to Na and Nb?
  - Why does M disappear after M2?
  - Answer
    - Actually, Na and Nb are alternative names for M
    - Na is Alice's name for M
    - Nb is Bob's name for M
    - Na and Nb are a sort of "local" names

Let's try to analyse this protocol:

- ASSUMPTIONS
  - KEYS:

$A \models T \overset{K_A}{\longleftrightarrow} A$

$T \models T \overset{K_A}{\longleftrightarrow} A$

$B \models T \overset{K_B}{\longleftrightarrow} B$

$T \models T \overset{K_B}{\longleftrightarrow} B$

Session key is generated by TTP

$T \models A \overset{K_{AB}}{\longleftrightarrow} B$

  - FRESHNESS:

$A \models \#(N_a)$

$A \models \#(M)$ Nonce M is generated by Alice, so considered fresh too.

$B \models \#(N_b)$

- TRUST:

$A \models T \Rightarrow A \overset{K_{AB}}{\longleftrightarrow} B$

$B \models T \Rightarrow A \overset{K_{AB}}{\longleftrightarrow} B$

$A \models T \Rightarrow (B \sim M)$

$B \models T \Rightarrow (A \sim M)$

IDEALISED PROTOCOL

M1 $A \rightarrow B : \{N_a, M, A, B\}_{K_A}$ This message does not increase Bob's set of beliefs, it is encrypted with K_A. B simply forwards it to T.

M2 $B \rightarrow T : \{N_a, M, A, B\}_{K_A}, \{N_b, M, A, B\}_{K_B}$

When T receives M2, it applies first postulate to say that T believes info is coming from A.

$T \models A \sim (N_a, M, A, B)$ That's all, no freshness info for TTP here. N_a, M are fresh for A, not T

$T \models B \sim (N_B, M, A, B)$

TTP reasons this way: Alice is telling me she's in session M, Bob that he's in session M. Same session. But I cannot know if it is a current session or a past one. So TTP specifies this belief in M3.

First CT can be idealised this way:

M3 $T \to B : \{N_A, A \overset{K_{AB}}{\leftrightarrow} B, B \mid\sim M\}_{K_A}, \{N_B, A \overset{K_{AB}}{\leftrightarrow} B, A \mid\sim M\}_{K_B}$

TTP is saying K_{AB} is session key and other peer is in session M.

When Bob receives M3, Bob decrypts second field and concludes message is coming from TTP and is also fresh!

$B \mid\equiv T \mid\sim (N_B, A \overset{K_{AB}}{\leftrightarrow} B, A \mid\sim M)$

$B \mid\equiv \#(N_B)$

$B \mid\equiv T \mid\equiv (A \overset{K_{AB}}{\leftrightarrow} B, A \mid\sim M)$

Consider only ↑ this part plus the trust assumption 2, the:

$B \mid\equiv A \overset{K_{AB}}{\leftrightarrow} B$ which is one objective for key authentication

Let's add the following assumption:

- $B \mid\equiv T \Rightarrow (A \mid\sim M)$ B believes TTP correctly forwards messages coming from A.

So, B achieves additional belief that:

$B \mid\equiv A \mid\sim M$ reasonable (Weaker condition, true, because Bob does not know if this is actual key or not.

M4 $B \to A : \{N_A, A \overset{K_{AB}}{\leftrightarrow} B, B \mid\sim M\}_{K_A}$ Message logically from T

Given the 1st postulate, Alice concludes:

$A \mid\equiv T \mid\sim (N_A, A \overset{K_{AB}}{\leftrightarrow} B, B \mid\sim M)$

But given freshness of N_A, for 2nd postulate, A concludes:

$A \mid\equiv T \mid\equiv (N_A, A \overset{K_{AB}}{\leftrightarrow} B, B \mid\sim M)$

Given trust assumption 1, A can conclude by 3rd postulate:

$A \mid\equiv A \overset{K_{AB}}{\leftrightarrow} B$

And if we add the same trust assumption over relaying messages of B, that is:

$A \mid\equiv T \Rightarrow (B \mid\sim M)$

We can conclude that by 3rd postulate: $A \mid\equiv B \mid\sim M$, but since M is fresh for A, then: $A \mid\equiv B \mid\equiv M$

Principles:

- You should use nonces for freshness.
- Cryptography in M1 is used to link two names (M, N_A). Important to be aware of why you use cryptography.

Proposal was: what if, since N_A is just a name, I take N_A and send it in the clear?

So, M1 $A \to B: M, A, B, N_A, E_{K_{AT}}(M, A, B)$

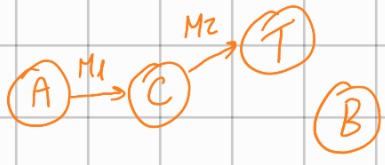
M2 $B \to T: M, A, B, N_A, E_{K_{AT}}(M, A, B), N_B, E_{K_{BT}}(M, A, B)$

Now M, N_A, N_B are completely unlinked and so M1, M3 and M2, M4 are unlinked.

This scheme is subject to MITM:

If A and C have performed a protocol execution in a different session M' and exchanged $E_{K_{AT}}(M', A, C)$, $E_{K_C}(M', A, C)$. In particular, C has $E_{K_{AT}}(M', A, C)$.

So C could, when Alice sends M1 to B, intercept M1.



So C sends now M2, but C checks T to believe we are in session M'.

M2 $B^{[C]} \to T$: $M', A, B^C, N_A, E_{K_{AT}}(M', A, B), N_B^{N_C}, E_{K_{BT}}(M', A, B) E_{K_C}(M', A, C)$

exploiting M' and $E_{K_{AT}}(M', A, C)$ from old sessions. TTP is completely unaware, both A and C should be in M'. So T replies:

M3 $T \to B^{[C]}$: $M', E_{K_{AT}}(N_A, K_{AB}), E_{K_{BT}}(N_B, K_{AB})^{E_{K_C}(N_C, K_{AB})}$

And C only needs to forward this quickly to Alice.

M4 $B^{[C]} \to A$: $M, E_{K_{AT}}(N_A, K_{AB})$ Alice believes to talk to B, while B never received anything.

- Problem is we unlinked the two pairs (M1, M3), (M2, M4) of messages. In previous protocols, N_A, N_B were local names for M and two participants had nonce to be in same session.

How to solve? Put names in M3 and M4: When TTP receives M2, it knows the two individuals.
Instead of $E_{K_A}(N_A, K_{AB})$ TTP can send $E_{K_A}(N_A, K_{AB}, A, C)$
Messages should be as much self explainable as possible.
Better to add additional field, we should not rely on the context.

- If we need to insert references to Alice and Bob in M3 and M4, then the protocol can ben modified as follows

M1. $A \rightarrow B: \quad A, B, N_a$
M2. $B \rightarrow T: \quad A, B, N_a, N_b$
M3. $T \rightarrow B: \quad E_{K_A}(N_a, A, B, K_{ab}), E_{K_B}(N_b, A, B, K_{ab})$
M4. $B \rightarrow A: \quad E_{K_A}(N_a, A, B, K_{ab})$

Another protocol, old version of SSL

M1 $A \to B: \{K_{ab}\}_{K_B}$ Message by means of which Alice sends Bob pre master secret.

M2 $B \to A: \{N_b\}_{K_{AB}}$

M3 $A \to B: \{C_a, \{N_b\}_{K_a^{-1}}\}_{K_{AB}}$ C_a: client certificate. $\{N_b\}$ signed w/ priv key

This protocol contains a vulnerability. Bob receives M3 so Bob can decrypt. Bob receives N_b has been digitally signed N_b. This implies Alice has seen N_b. To see N_b this implies Alice knows K_{AB}. This implies K_{AB} is session key to talk to Alice. So Bob believes K_{AB} is session key to talk to Alice.

1st statement: true; but the fact that she knows N_b does not imply that Alice actually believes K_{AB}.

Consider M1:

After receiving it:

$B \triangleleft K_{AB}$ Bob cannot make any assumption on who sent message: K_B is Bob's public key

After M2, Alice sees N_b, she knows that K_{AB} is shared key with Bob because she generated it. If K_{AB} is also fresh, Alice concludes:

$A \models B \models N_b$. IN CURRENT EXEC Bob has said N_b.

When Bob receives M3, Bob can decrypt but he cannot believe M3 is coming from Alice because of pub K. But verifying dig. signature by Alice tells Bob that:

$B \models A \models N_b$ Alice has seen N_b.

Our assumptions:

$A \models A \overset{K_{AB}}{\leftrightarrow} B$ $A \models \#(A \overset{K_{AB}}{\leftrightarrow} B)$

$A \models \overset{K_B}{\longmapsto} B$

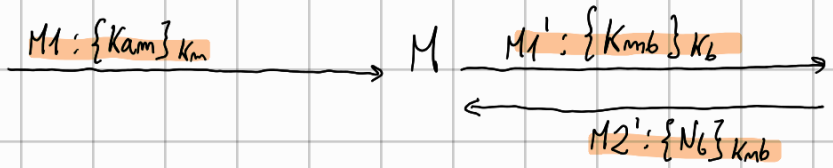
$B \models \#(N_b)$

We cannot prove in general, that K_{AB} is session key to talk to A.

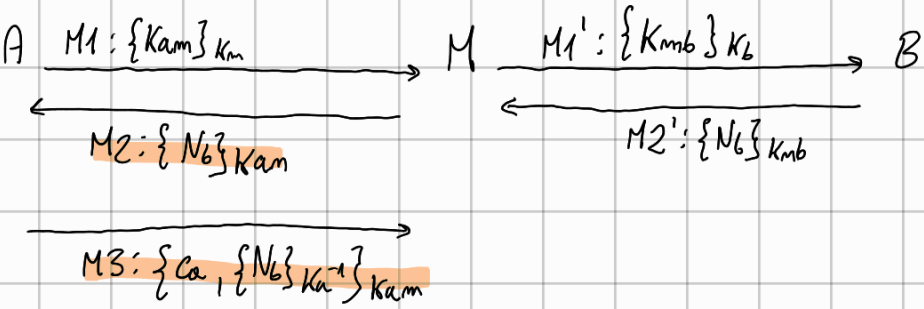
A MITM attack can be shown because of this:

M behaves as a server for A, as a client for B.

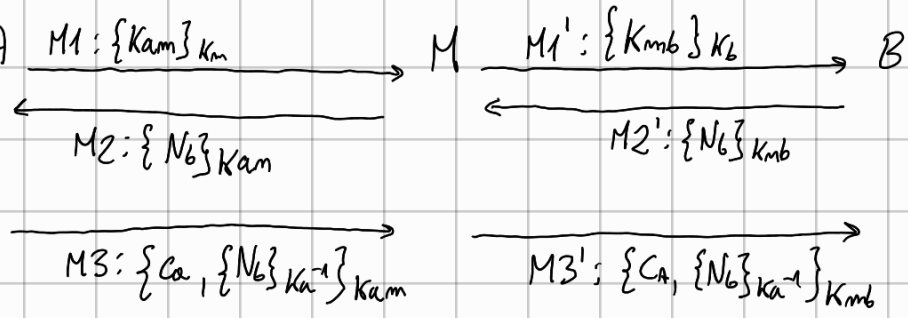
1. Alice transmits M1 to M. Alice wants to talk to M but M exploits this opportunity.



2. M sends M1 and B replies with M2'.
3. M can obtain N6 with Kmb and she replays it in communication with A:



4. Alice answers by performing signature, but M can take message and forward it to B encrypting it by means of Kmb



- Bob has no idea of the person that sent Kmb. Only info that Bob has is digital signature, that would lead him to believe he is talking to Alice, while he is talking to M.

If we need proof that Kab is coming from Alice, then we need to add the key in the digital signature. DON'T TAKE THINGS FOR GRANTED!

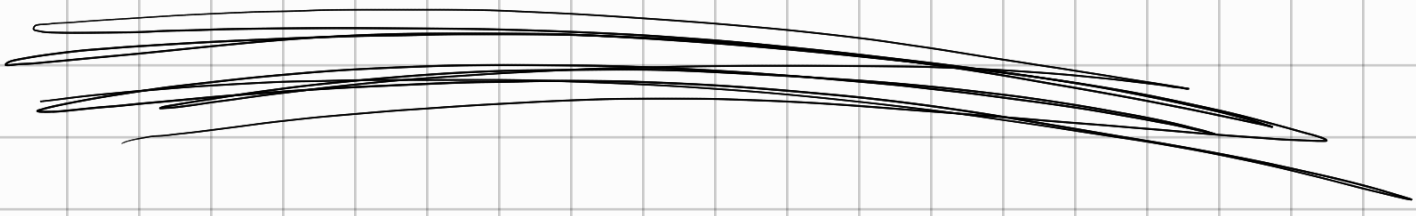
Modify M3:

$$\{C_A, \{N_b, K_{ab}\}_{K_a^{-1}}\}_{K_{ab}}$$ Here Bob has proof that Alice has sent Nb and Kab.

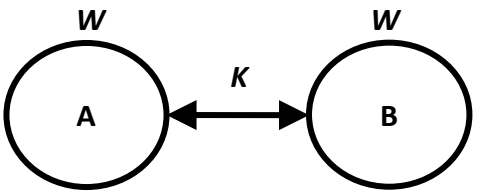
But what if adversary selects $K_{mb} = K_{am}$? This is still not enough! We would still have MITM.

Solution: public identifiers!

$\{C_A, \{N_b, K_{ab}, A, B\}_{K_{am}}\}_{K_{mb}}$ K_{ab} is the key that allows A and B to communicate.



Establishing a session key



- A and B *a-priori* share a **long term key *W* (key encryption key)**
- A and B wants to establish a **short term *K* (session key)**

- A session key is used for one communication session
- Session key is used for bulk encryption
- KEKs are used for runs of the key establishment protocols; in each run, the key encrypts a small amount of data

3

Establishing a session key

one-pass

$$M1 \quad A \to B: \quad E\left(W, t_A \,||\, "B,A" \,||\, K\right)$$

- t_A is a timestamp (a "fresh" quantity) that requires synchronized clocks

with challenge-response

$$M1 \quad A \square B: \quad n_B$$
$$M2 \quad A \to B: \quad E_W\left(W, n_B \,||\, "A,B" \,||\, K\right)$$

- n_B is a nonce (a "fresh" quantity) e.g., a counter or a random number

both parties contribute to the session key

$$M1 \quad A \square B: \quad n_B$$
$$M2 \quad A \to B: \quad E\left(W, K_A \,||\, n_B \,||\, n_A \,||\, "A,B"\right)$$
$$M3 \quad A \square B \quad E\left(W, K_B \,||\, n_A \,||\, n_B \,||\, "B,A"\right)$$

- n_A and n_B are nonces
- K_A and K_B are keying materiale
- $K = K_A \oplus K_B$

4

Security protocols are three-line programs that people still manage to get wrong.

Roger M. Needham

May 25 BAN Logic 5

5

Design and verification of security protocols

THE BAN LOGIC – FORMALISM AND POSTULATES

May 25 BAN Logic 6

6

Main topics

- The BAN logic
- Design principles
- Case studies
  - Needham-Schroeder
  - Otway-Rees
  - SSL (an old version)
  - …

7

The BAN logic

- After its inventors: M. **B**urrows, M. **A**badi, R. **N**eedham
- Logic based on *belief* and *action*
- How to use the logic
  - The logic cannot prove that a protocol is wrong
  - However, if you cannot prove a protocol correct, then consider that protocol with great suspicion

8

Google Scholar – all versions

- M. Burrows, M. Abadi, R.M, Needham, A Logic of Authentication, *Symposium on Operating Systems Principles*, 1989
- M. Burrows, M. Abadi, R.M, Needham, A Logic of Authentication, *ACM Transactions on Computer Systems*, 1990

9

Formalism

$P \mid\equiv X$	**P believes X**. P behaves as if X were true
$P \lhd X$	**P sees X**. P has received/read a message/file containing X, either in the past or in the present execution of the protocol. P can read X and repeat it
$P \mid\sim X$	**P once said X**. P sent/wrote X in a message/file. P believed X when P sent/wrote it.
$P \Rightarrow X$	**P controls X**. P is an authority on X and we should trust P on this regard
$\#(X)$	**X is fresh**
$P \overset{K}{\leftrightarrow} Q$	**K is a shared key between P e Q**

10

Formalism

$P \overset{K}{\leftrightarrow} Q$ X is a shared secret between P e Q

$\overset{K}{\mapsto} P$ K is P's public key

$\langle X \rangle_Y$ X is a combined with Y

$\{X\}_K$ X has been encrypted with K

Formalism – Examples

$A \mid\equiv \#(N_a)$ A believes that N_a is fresh

$A \mid\equiv A \overset{K}{\leftrightarrow} B$ A believes K to be a shared key with B

$T \mid\equiv A \overset{K}{\leftrightarrow} B$ T believes that K is a shared key between A and B

$A \mid\equiv T \Rightarrow A \overset{K}{\leftrightarrow} B$ A believes T an authority on generating session keys

$A \mid\equiv T \Rightarrow \#\left(A \overset{K}{\leftrightarrow} B \right)$ A believes that T is competent in generating fresh session keys

Preliminaries

- BAN logic considers two epochs: the present and the past.
- The present begins with the start of the protocol.
- Beliefs achieved in the present are stable for all the protocol duration.
- Beliefs of the past may not hold in the present.
- **Assumption**: If P says X then P believes X.

13

Postulates: message meaning rule

$$\frac{P \mid\equiv Q \overset{K}{\leftrightarrow} P, P \triangleleft \{X\}_K}{P \mid\equiv Q \mid\sim X}$$

If *K* is a shared key between *P* and *Q*, and *P* sees a message encrypted by *K* containing *X* (and P did not send that message), then *P* believes that *X* was sent by *Q*

$$\frac{P \mid\equiv \overset{K}{\mapsto} Q, P \triangleleft \{X\}_{K^{-1}}}{P \mid\equiv Q \mid\sim X}$$

If *K* is *Q*'s public key, and *P* sees a message signed by con K^{-1} containing *X*, then *P* believes that *X* was sent by *Q*

$$\frac{P \mid\equiv Q \overset{Y}{\rightleftharpoons} P, P \triangleleft \langle X \rangle_Y}{P \mid\equiv Q \mid\sim X}$$

If Y is a shared secrete between P and *Q*, and *P* sees a message where Y is combined with X (and P did not send the message), then*P* believes that *X* was sent by *Q*

14

Postulates: nonce verification rule

$$\frac{P \mid\equiv \#(X), P \mid\equiv Q \mid\sim X}{P \mid\equiv Q \mid\equiv X}$$

- If *P* believes *Q* said *X* and P believes *X* is *fresh*, then *P* believes Q believes X (now, in this protocol execution)
- If *P* believes *X* was sent by *Q*, and *P* believes *X* is *fresh*, then *P* believes *Q* has sent *X* in this protocol execution instance

Postulates: jurisdiction rule

$$\frac{P \mid\equiv Q \mid\equiv X, P \mid\equiv Q \Rightarrow X}{P \mid\equiv X}$$

- If *P* believes *Q* believes *X* and *P* believes *Q* is an authority on *X*, then *P* believes *X too*
- If *P* believes *Q* says *X* and *P* trusts *Q* on *X*, then *P* believes *X* too

More postulates

$$\frac{P|\equiv X,\quad P|\equiv Y}{P|\equiv (X,Y)} \qquad \frac{P|\equiv (X,Y)}{P|\equiv X,\quad P|\equiv Y} \qquad \frac{P|\equiv Q|\equiv (X,Y)}{P|\equiv Q|\equiv X} \qquad \frac{P|\equiv Q|\sim (X,Y)}{P|\equiv Q|\sim X}$$

$$\frac{P|\equiv \#(X)}{P|\equiv \#(X,Y)}$$

$$\frac{P \lhd (X,Y)}{P \lhd X} \qquad \frac{P \lhd \langle X \rangle_Y}{P \lhd X}$$

$$\frac{P|\equiv \overset{K}{\leftrightarrow} P,\quad P \lhd \{X\}_K}{P \lhd X} \qquad \frac{P|\equiv \overset{K}{\mapsto} P,\quad P \lhd \{X\}_K}{P \lhd X} \qquad \frac{P|\equiv \overset{K}{\mapsto} Q,\quad P \lhd \{X\}_{K^{-1}}}{P \lhd X}$$

$$\frac{P|\equiv \overset{K}{R \leftrightarrow} R'}{P|\equiv \overset{K}{R' \leftrightarrow} R} \qquad \frac{P|\equiv Q|\equiv \overset{K}{R \leftrightarrow} R'}{P|\equiv Q|\equiv \overset{K}{R' \leftrightarrow} R} \qquad \frac{P|\equiv \overset{K}{R \rightleftharpoons} R'}{P|\equiv \overset{K}{R' \rightleftharpoons} R} \qquad \frac{P|\equiv Q|\equiv \overset{K}{R \rightleftharpoons} R'}{P|\equiv Q|\equiv \overset{K}{R' \rightleftharpoons} R}$$

Idealized protocol

In the ***real protocol***, each protocol step is represented as

$$A \to B : message$$

For example:

$$A \to B : \{A, K_{ab}\}_{K_{bs}}$$

This notations is ambiguous. Thus the protocol has to be ***idealized***

$$A \to B : \left\{ \overset{K_{ab}}{A \leftrightarrow B} \right\}_{K_{bs}}$$

The resulting specification is more clear and you can desume the formula

$$\overset{K_{ab}}{B \lhd A \leftrightarrow B}$$

Protocol analysis

- Protocol analysis consists in the following steps
  1. Derive the idealized protocol from the real one
  2. Determine assumptions
  3. Apply postulates to each protocol step and determine beliefs achieved by principals at the step
  4. Draw conclusions

19

Protocol analysis

[assumption] **S_1** [assertion 1]

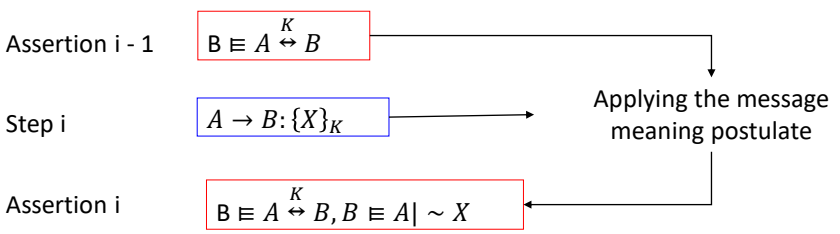
....

[assertion i - 1] **S_i** [assertion i]

...

[assertion n - 1] **S_n** [conclusions]

Example



20

Objectives of a protocol

Objectives depend on the context

- Typical objectives:

	$A \mid\equiv A \overset{K}{\leftrightarrow} B$	$B \mid\equiv A \overset{K}{\leftrightarrow} B$	(*key authentication*)
often	$A \mid\equiv B \mid\equiv A \overset{K}{\leftrightarrow} B$	$B \mid\equiv A \mid\equiv A \overset{K}{\leftrightarrow} B$	(*key confirmation*)
also	$A \mid\equiv \#\left(A \overset{K}{\leftrightarrow} B \right)$	$B \mid\equiv \#\left(A \overset{K}{\leftrightarrow} B \right)$	(*key freshness*)

- Interaction with a certification authority:

$$A \mid\equiv \overset{e_b}{\mapsto} B$$

BAN Logics

THE NEEDHAM-SCHROEDER PROTOCOL

Needham-Schroeder (1978)

Real protocol

$$
\begin{array}{lll}
M1 & A \rightarrow T: & A, B, N_a \\
M2 & T \rightarrow A: & E_{K_a}\left(N_a, B, K_{ab}, E_{K_b}(K_{ab}, A)\right) \\
M3 & A \rightarrow B: & E_{K_b}(K_{ab}, A) \\
M4 & B \rightarrow A: & E_{K_{ab}}(N_b) \\
M5 & A \rightarrow B: & E_{K_{ab}}(N_b - 1)
\end{array}
$$

Needham-Schroeder (1978)

***Implicit statement**, not explicitly derived from the real protocol*

- *The idealized protocol may contain implicit statements*

Idealized protocol

$$
\begin{array}{lll}
M2 & T \rightarrow A & \left\{ N_a, \left(A \overset{K_{ab}}{\leftrightarrow} B \right), \#\left(A \overset{K_{ab}}{\leftrightarrow} B \right), \left\{ A \overset{K_{ab}}{\leftrightarrow} B \right\}_{K_b} \right\}_{K_a} \\
M3 & A \rightarrow B & \left\{ A \overset{K_{ab}}{\leftrightarrow} B \right\}_{K_b} \\
M4 & B \rightarrow A & \left\{ N_b, A \overset{K_{ab}}{\leftrightarrow} B \right\}_{K_{ab}} \text{ from } B \\
M5 & A \rightarrow B & \left\{ N_b, A \overset{K_{ab}}{\leftrightarrow} B \right\}_{K_{ab}} \text{ from } A
\end{array}
$$

Needham-Schroeder (%)

$M2 \quad T \to A \quad \left\{ N_a, \left(A \overset{K_{ab}}{\leftrightarrow} B \right), \#\left(A \overset{K_{ab}}{\leftrightarrow} B \right), \left\{ A \overset{K_{ab}}{\leftrightarrow} B \right\}_{K_b} \right\}_{K_a}$

After receiving N_a, *T* said K_{ab} is "good" to talk to *Bob*

$M3 \quad A \to B \quad \left\{ A \overset{K_{ab}}{\leftrightarrow} B \right\}_{K_b}$

T said K_{ab} is good to talk to *Alice*

$M4 \quad B \to A \quad \left\{ N_b, A \overset{K_{ab}}{\leftrightarrow} B \right\}_{K_{ab}} \text{ from } B$

After receiving K_{ab}, *B* has said K_{ab} is good to talk to *A*

$M5 \quad A \to B \quad \left\{ N_b, A \overset{K_{ab}}{\leftrightarrow} B \right\}_{K_{ab}} \text{ from } A$

After receiving N_b, *A* has said K_{ab} is good to talk to *Bob*

Principle 1. We have to **specify the meaning of each message**; specification must depend on the message contents; it must be possible to write a sentence describing such a meaning

25

Needham-Schroeder

Assumptions

$A \mid\equiv A \overset{K_a}{\leftrightarrow} T$ $\quad$ $B \mid\equiv B \overset{K_b}{\leftrightarrow} T$

$T \mid\equiv A \overset{K_a}{\leftrightarrow} T$ $\quad$ $T \mid\equiv B \overset{K_b}{\leftrightarrow} T$

$T \mid\equiv A \overset{K_{ab}}{\leftrightarrow} B$

$A \mid\equiv \left(T \Rightarrow A \overset{K_{ab}}{\leftrightarrow} B \right)$ $\quad$ $B \mid\equiv \left(T \Rightarrow A \overset{K_{ab}}{\leftrightarrow} B \right)$

$A \mid\equiv \left(T \Rightarrow \#\left(A \overset{K_{ab}}{\leftrightarrow} B \right) \right)$

$A \mid\equiv \#(N_a)$ $\quad$ $B \mid\equiv \#(N_b)$

$T \mid\equiv \#\left(A \overset{K_{ab}}{\leftrightarrow} B \right)$ $\quad$ $B \mid\equiv \#\left(A \overset{K_{ab}}{\leftrightarrow} B \right)$

Objectives

$A \mid\equiv A \overset{K_{ab}}{\leftrightarrow} B$

$B \mid\equiv A \overset{K_{ab}}{\leftrightarrow} B$

$A \mid\equiv B \mid\equiv A \overset{K_{ab}}{\leftrightarrow} B$

$B \mid\equiv A \mid\equiv A \overset{K_{ab}}{\leftrightarrow} B$

Principle 2. Designer must know the **trust relationships** upon which the protocol is based. He/she must know why they are necessary. Such reasons must be made explicit.

26

Needham-Schroeder

After M2

message meaning e nonce verification

$$A \mid\equiv T \mid\equiv \left(A \overset{K_{ab}}{\leftrightarrow} B \right)$$

$$A \mid\equiv T \mid\equiv \#\left(A \overset{K_{ab}}{\leftrightarrow} B \right)$$

jurisdiction rule

$$A \mid\equiv \left(A \overset{K_{ab}}{\leftrightarrow} B \right)$$

$$A \mid\equiv \#\left(A \overset{K_{ab}}{\leftrightarrow} B \right)$$

After M3

message meaning

$$B \mid\equiv T \mid\sim A \overset{K_{ab}}{\leftrightarrow} B$$

nonce verification

$$B \mid\equiv T \mid\equiv A \overset{K_{ab}}{\leftrightarrow} B$$

jurisdiction rule

$$B \mid\equiv A \overset{K_{ab}}{\leftrightarrow} B$$

Needham-Schroeder

After M4

message meaning

$$A \mid\equiv B \mid\sim A \overset{K_{ab}}{\leftrightarrow} B$$

nonce verification

$$A \mid\equiv B \mid\equiv A \overset{K_{ab}}{\leftrightarrow} B$$

Principle 3. A key may have been used recently to encrypt a nonce but it may be old or compromised. The recent use of a key does not make it more secure

After M5

message meaning

$$B \mid\equiv A \mid\sim \left(N_b, A \overset{K_{ab}}{\leftrightarrow} B \right)$$

nonce verification

$$B \mid\equiv A \mid\equiv A \overset{K_{ab}}{\leftrightarrow} B$$

Replay attack against Needham-Schroeder

- As Bob blindly believes that any key he receives in M3 is fresh;
- If the adversary is able to obtain a session key K_{ab};
- If the adversary records the messages that lead to establish K_{ab}, in particular M3;
- Then, by replaying M3, the adversary is able to impersonate A w.r.t. B and establish K_{ab} at his/her will

A good design practice

It is always a *good design practice* to analyse the consequences from a situation in which

- a session key gets compromised and
- the adversary recorded the protocol run that led to that key establishment

BAN Logic

THE OTWAY-REES PROTOCOL

32

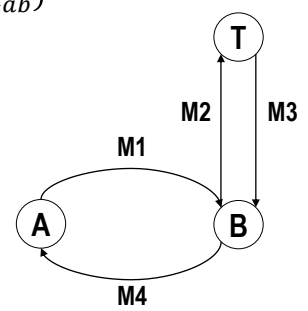
Otway-Rees – real protocol

M1. $A \rightarrow B$: $M, A, B, E_{K_A}(N_A, M, A, B)$

M2. $B \rightarrow T$: $M, A, B, E_{K_A}(N_A, M, A, B), E_{K_B}(N_B, M, A, B)$

M3. $T \rightarrow B$: $M, E_{K_A}(N_A, K_{ab}), E_{K_B}(N_B, K_{ab})$

M4. $B \rightarrow A$: $M, E_{K_A}(N_A, K_{ab})$



33

Otway-Rees

- The protocol presents odd aspects
  - Na ed Nb are nonces, they are supposed to prove freshness. Then, why are they encrypted in messages M1 and M2?
  - Why do we need M in addition to Na and Nb?
  - Why does M disappear after M2?
  - Answer
    - Actually, Na and Nb are alternative names for M
    - Na is Alice's name for M
    - Nb is Bob's name for M
    - Na and Nb are a sort of "local" names

Otway-Rees – idealized protocol

M1. $A \rightarrow B$: $\{N_A, M, A, B\}_{K_a}$

M2. $B \rightarrow T$: $\{N_A, M, A, B\}_{K_a}, \{N_B, M, A, B\}_{K_b}$

M3. $T \rightarrow B$: $\{N_a, A \overset{K_{ab}}{\leftrightarrow} B, B| \sim M\}_{K_a}, \{N_b, A \overset{K_{ab}}{\leftrightarrow} B, A| \sim M\}_{K_b}$

M4. $B \rightarrow A$: $\{N_b, A \overset{K_{ab}}{\leftrightarrow} B, A| \sim M\}_{K_a}$

Otway-Rees

M1. $A \to B: \{N_A, M, A, B\}_{K_a}$

M2. $B \to T: \{N_A, M, A, B\}_{K_a}, \{N_B, M, A, B\}_{K_b}$

M3. $T \to B: \left\{N_a, A \overset{K_{ab}}{\leftrightarrow} B, B \mid\sim M\right\}_{K_a}, \left\{N_b, A \overset{K_{ab}}{\leftrightarrow} B, A \mid\sim M\right\}_{K_b}$

M4. $B \to A: \left\{N_a, A \overset{K_{ab}}{\leftrightarrow} B, B \mid\sim M\right\}_{K_a}$

M1: Alice says that *M* is a transaction with Bob and N_a is another name of Alice in *M*

M2: Bob says that *M* is a transaction with Bob and N_b is another name of Bob in *M*

M3: After receiving N_b, *T* says that K_{ab} is good and that Alice believed to be in *M*

M4: After receiving N_a, *T* says that K_{ab} is good and that Bob believed to be in *M*

Otway-Rees protocol

Assumptions

$A| \equiv A \overset{K_a}{\leftrightarrow} T$ $\quad$ $B| \equiv A \overset{K_b}{\leftrightarrow} T$

$T| \equiv A \overset{K_a}{\leftrightarrow} T$ $\quad$ $T| \equiv A \overset{K_b}{\leftrightarrow} T$

$T| \equiv A \overset{K_{ab}}{\leftrightarrow} B$

$A| \equiv (T \Rightarrow A \overset{K_{ab}}{\leftrightarrow} B)$ $\quad$ $B| \equiv (T \Rightarrow A \overset{K_{ab}}{\leftrightarrow} B)$

$A| \equiv (T \Rightarrow B| \sim M)$ $\quad$ $B| \equiv (T \Rightarrow A| \sim M)$

$A| \equiv \#(N_a)$ $\quad$ $B| \equiv \#(N_b)$

$A| \equiv \#(M)$

Goals

$A| \equiv A \overset{K_{ab}}{\leftrightarrow} B$

$B| \equiv A \overset{K_{ab}}{\leftrightarrow} B$

$A| \equiv B| \equiv M$

$B| \equiv A| \sim M$

Protocollo di Otway-Rees

After M2

$$T \mid\equiv A \mid\sim (N_a, M, A, B) \quad T \mid\equiv B \mid\sim (N_b, M, A, B)$$

After M3

$$B \mid\equiv T \mid\sim \left(N_b, A \overset{K_{ab}}{\leftrightarrow} B, A \mid\sim M\right)$$

Given Bob's belief in N_b freshness

$$B \mid\equiv T \mid\equiv \left(N_b, A \overset{K_{ab}}{\leftrightarrow} B, A \mid\sim M\right)$$

$$B \mid\equiv A \overset{K_{ab}}{\leftrightarrow} B, \quad B \mid\equiv A \mid\sim M$$

Given Bob's trust in T about keys and its capability to relay

After M4

$$A \mid\equiv T \mid\sim \left(N_a, A \overset{K_{ab}}{\leftrightarrow} B, B \mid\sim M\right)$$

Given Alice's belief in N_a

$$A \mid\equiv T \mid\equiv \left(N_a, A \overset{K_{ab}}{\leftrightarrow} B, B \mid\sim M\right)$$

$$A \mid\equiv A \overset{K_{ab}}{\leftrightarrow} B, \quad A \mid\equiv B \mid\equiv M$$

Given Alice's trust in T about keys and its capability to relay and given Alice's belief in *M* freshness

Otway-Rees Protocol

- Nonces N_a and N_b are for freshness but also to link messages M1 and M2 to messages M3 and M4, respectively
  - Nonce N_a (N_b) is a reference to Alice (Bob) within M or, equivalently,
  - nonce N_a (N_b) is another name for Alice (Bob) in M
- In M1 (M2), encryption is not for secrecy but to indissolubly link Alice (Bob), N_a (N_b) and M together

Principle 4. Properties required to nonces must be clear. What it is fine to guarantee freshness might not be to guarantee an association between parts.

Principles 5. The reason why encryption is used must be clear

Otway-Rees modified [→]

- If nonces have to guarantee freshness only, then messages M1 and M2 could be modified as follows

$$\begin{array}{ll} \text{M1.} & A \rightarrow B: \quad M, A, B, N_A, E_{K_A}(M, A, B) \\ \text{M2.} & B \rightarrow T: \quad M, A, B, N_A, E_{K_A}(M, A, B), N_B, E_{K_B}(M, A, B) \end{array}$$

  - M1 and M3 (M2 and M4) are not linked anymore ➔
  - The resulting protocol is subject to a man-in-the-middle attack ➔
  - An adversary may impersonate Bob (Alice) with respect to Alice (Bob)

Otway-Rees modified [→]

- THE MIM ATTACK: ASSUMPTIONS
- Carol (the adversary) has already carried out a protocol instance with Alice (M')
- Carol holds an "old" ciphertext $E_{Ka}(M', A, C)$

Otway-Rees modified [→]

- MIM ATTACK: THE ATTACK

M1. $A \rightarrow B[C]$: $M, A, B, N_a, E_{K_A}(M, A, B)$
M2. $C \rightarrow T$: $M', A, C, N_a, E_{K_A}(M', A, C), N_c, E_{K_c}(M', A, C)$
M3. $T \rightarrow C$: $M', E_{K_a}(N_a, K_{ac}), E_{K_c}(N_c, K_{ac})$
M4. $[C]B \rightarrow A$: $E_{K_a}(N_a, K_{ac})$

- Message M3 (A's part) is underspecified: it contains no (indissoluble) link to B

May 25 BAN Logic 42

42

Otway-Rees protocol: an improvement

- If we need to insert references to Alice and Bob in M3 and M4, then the protocol can ben modified as follows

M1. $A \rightarrow B$: A, B, N_a
M2. $B \rightarrow T$: A, B, N_a, N_b
M3. $T \rightarrow B$: $E_{K_A}(N_a, A, B, K_{ab}), E_{K_B}(N_b, A, B, K_{ab})$
M4. $B \rightarrow A$: $E_{K_A}(N_a, A, B, K_{ab})$

Principle 6. If an identifier is necessary to complete the meaning of a message, it is prudent to explicitly mention such an identifier in the message

May 25 BAN Logic 43

43

BAN Logics

SSL PROTOCOL – AN OLD VERSION

The protocol

Protocol objectives

- establish a shared key K_{ab}
- mutual authentication

Assumptions

- A: client; B: server
- K_b is Bob's public key
- K_a is Alice's public key

$$\begin{aligned}
&\text{M1.} \quad A \rightarrow B: \quad \{K_{ab}\}_{K_b} \\
&\text{M2.} \quad B \rightarrow A: \quad \{N_b\}_{K_{ab}} \\
&\text{M3.} \quad A \rightarrow B: \quad \left\{C_A, \{N_b\}_{K_a^{-1}}\right\}_{K_{ab}}
\end{aligned}$$

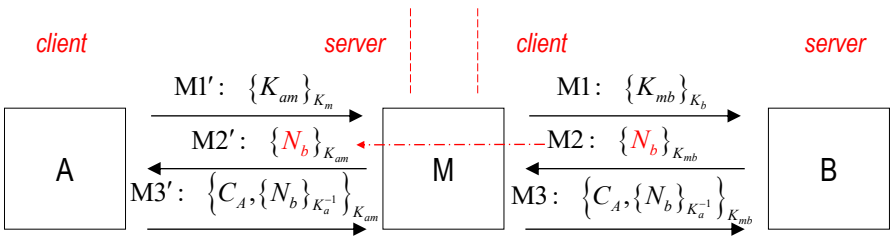
Infiormal (wrong) reasoning

- In M3 Bob sees Nb signed by Alice, so Bob gets convinced that Nb comes from Alice.
- As Bob transmitted Nb in M2 encrypted by Kab, then Bob thinks that Alice holds Kab
- This informal reasoning is wrong!
  - The reason is that M3 only proves that Alice saw Nb.
  - The protocol provides no evidence that Alice has seen Kab as well.
  - Consequences may be serious/dangerous: MIM attack.

47

The MiM attack

The adversary plays a MIM attack and impersonates *A* with respect to *B*



48

The protocol

Protocol objectives

- establish a shared key K_{ab}
- mutual authentication

Assumptions

- A: client; B: server
- K_b is Bob's public key
- Ka is Alice's public key

M1. $A \rightarrow B: \quad \{K_{ab}\}_{K_b}$

M2. $B \rightarrow A: \quad \{N_b\}_{K_{ab}}$

M3. $A \rightarrow B: \quad \left\{C_A, \{N_b\}_{K_a^{-1}}\right\}_{K_{ab}}$

M1: Bob sees key K_{ab}

M2: After receiving K_{ab}, Bob says N_b

M3: After receiving N_b, Alice says she saw N_b

May 25 BAN Logic 49

49

Analysis of the protocol

- Assumptions
  - Alice believes that Kab is shared with Bob
  - Alice believes that Kab is fresh
  - Alice believes that Kb is Bob's public key
  - Bob believes that Ka is Alices's public key

May 25 BAN Logic 50

50

Analysis of the protocol

- Idealized protocol
  1. After M1, B sees Kab
  2. After M2, A sees Nb, A believes that
     1. B said Nb and Kab;
     2. B believes Nb and Kab because Kab is fresh for Alice;
  3. After M3, B believes A believes N_b.
- However there is no proof for B to believe that Alice has seen Kab.

A possible countermeasure

- The attack may be avoided by modifying M3 as follows

$$\text{M3 } A \rightarrow B: \quad \left\{C_A, \{A, B, K_{ab}, N_b\}_{K_a^{-1}}\right\}_{K_{ab}}$$

after receiving N_b, Alice says that K_{ab} is a good key to communicate with Bob

- Important
  - In message M3, it's necessary to introduce identifiers A and B in addition to K_{ab} because, otherwise, the attack would be still possible by setting $K_{am} = K_{bm}$

BAN Logics

OTHER ISSUES

55

Sign encrypted data

Principle 7.

- If an entity signs an encrypted message, it is not possible to infer that such an entity knows the message contents
- In contrast, if an entity signs a message and then encrypts it, then it is possible to infer that the entity knows the message contents K_a^{-1}

if you need to do both, first sign then encrypt

Esempio: X.509

$$A \rightarrow B: \quad A, \left\{ T_a, N_a, B, X_a, \{Y_a\}_{K_b} \right\}_{K_a^{-1}}$$

The message contains no proof that the sender (Alice) knows Y_a

56

Predictable nonces

Principle 8. A predictable quantity can be used as a nonce in a challenge-response protocol. In such a case, the nonce must be protected by a replay attack

EXAMPLE: ALICE RECEIVES A TIMESTAMP FROM A TIME SERVER
(ex. Alice uses the timestamp to synchronize her clock)

$M1 \quad A \rightarrow S \quad A, N_a$
$M2 \quad S \rightarrow A \quad \{T_s, N_a\}_{K_{as}}$

- N_a: predictable nonce
- (M2): After receiving N_a, S said T_s

Assumptions	Goal	Analysis
$A \mid\equiv S \overset{K_{as}}{\leftrightarrow} A$	$A \mid\equiv T_s$	$A \mid\equiv S \mid\sim T_s$
$A \mid\equiv S \Rightarrow T_s$		$A \mid\equiv S \mid\equiv T_s$
$A \mid\equiv \#(N_a)$		$A \mid\equiv T_s$

Predictable nonces

An attack

At time T_s, M predicts the next value of N_a

$M1 \quad M \rightarrow S \quad A, N_a$
$M2 \quad S \rightarrow M \quad \{T_s, N_a\}_{K_{as}}$ (S receives M2 at time T_s)

At time $T_S' > T_S$, Alice initiates a protocol instance using N_a

$M1 \quad A \rightarrow S[M] \quad A, N_a$
$M2 \quad S[M] \rightarrow A \quad \{T_s, N_a\}_{K_{as}}$

Alice is led to believe that the current time is T_s and not T_s'

Since N_a is predictable then it must be protected

$M1 \quad A \rightarrow S \quad A, \{N_a\}_{K_{as}}$
$M2 \quad S \rightarrow A \quad \{T_s, \{N_a\}_{K_{as}}\}_{K_{as}}$

Nonce: timestamp

Principle 9. If freshness is guaranteed by time stamp, then the difference between the local clock and that of other machines must be largely smaller than the message validity. Furthermore, the clock synchronization mechanisms is part of the Trusted Computing Base (TCB)

You have to assume that clock synchron. mechanism is working well, and it is secure

Example

- Kerberos. If the server clock can be turned back, then authenticators can be reused
- Kerberos. If the server clock can be set ahead, then it is possible to generate post-dated authenticators

On coding messages

Principle 10. The contents of a message must allow us to determine: (i) the protocol the message belongs to, (ii) the execution instance of the protocol, (iii) the number of the message within the protocol

Example Needham-Schroeder

$$M4 \quad B \to A \quad E_{K_{ab}}(N_b)$$
$$M5 \quad A \to B \quad E_{K_{ab}}(N_b - 1)$$

$N_b - 1$ distinguishes challenge from response

It would be more clear

$$M4 \quad B \to A \quad E_{K_{ab}}(\text{N-S Message 4}, N_b)$$
$$M5 \quad A \to B \quad E_{K_{ab}}(\text{N-S Message 5}, N_b)$$

It would just be cleaner to express protocol and message number!

The less you reason on clt the better.

On hash functions

For efficiency, we sign the hash of a message rather than the message itself

$$A \to B: \quad \{X\}_{K_b}, \{h(X)\}_{K_a^{-1}}$$

- The message does not contain any proof that the signer Alice actually knows X
- However, the signer Alice expects that the receiver Bob behaves as if the sender Alice knew the message
- Therefore, unless the signer Alice is *unwary**, signing the hash is equivalent to sign the message

* Metaphore: a manager who signs without reading

BAN postulates for hash functions

Nice thing of logic: you can extend it if you need

$$\frac{P \mid\equiv Q \mid\sim h(X), \quad P \triangleleft X}{P \mid\equiv Q \mid\sim X}$$

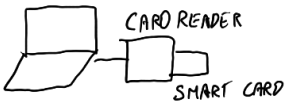
Plaintext

ASSUMPTION THAT WE USE

The postulate can be generalized to composite messages

$$\frac{P \mid\equiv Q \mid\sim h(X_1,\cdots,X_n), \quad P \triangleleft X_1,\cdots,P \triangleleft X_n}{P \mid\equiv Q \mid\sim (X_1,\cdots,X_n)}$$

Notice that *P* may receive X_i from different channels in different moments



If PC receives data how can you reason on this data? You have an extension of the logic for secure channels from HW pov

So assumption is HW channel is secure and timely (no reasonable delays)

BAN Logic

ON SECURE CHANNELS

63

Secure and timely channels

- Let L be a secure and timely channel
- Extended message-meaning rule

P sees X **on** L, P **believes** L **comes from** Q
P **believes** Q **said** X **on** L

- If P sees a message X on channel L, which is a secure channel from P, then Q believes that once P said X

You can safely assume message comes from the other party.

64

Secure and timely channels

- Let L be a secure and timely channel
- Extended nonce-verification rule

$$\frac{\text{P believes Q said X on L, P believes L is timely}}{\text{P believes Q believes X}}$$

  - If P believes that once Q uttered a message X on channel L, and that this channel is timely, then P believes Q believes X.

65

Secure and timely channels

- Let *L* be a secure and timely channel
  - Keyword **on**
- $\dfrac{Q\ \mathbf{sees}_L\ X, Q\ \mathbf{believes} \prec_L P}{Q\ \mathbf{believe}\ P\ \mathbf{said}\ X}$
- $\dfrac{Q\ \mathbf{believes}\ P\ \mathbf{said}_L\ X, Q\ \mathbf{believes\ timely}\ (L)}{Q\ \mathbf{believe}\ P\ \mathbf{believes}\ X}$
- Input channel, output channel

66

References

- Martin Abadi, Michael Burrows, Charles Kaufman, Butler Lampson, Authentication and delegation with smart-cards, Science of Computer Programming 21 (1993) 93–113.

67

68