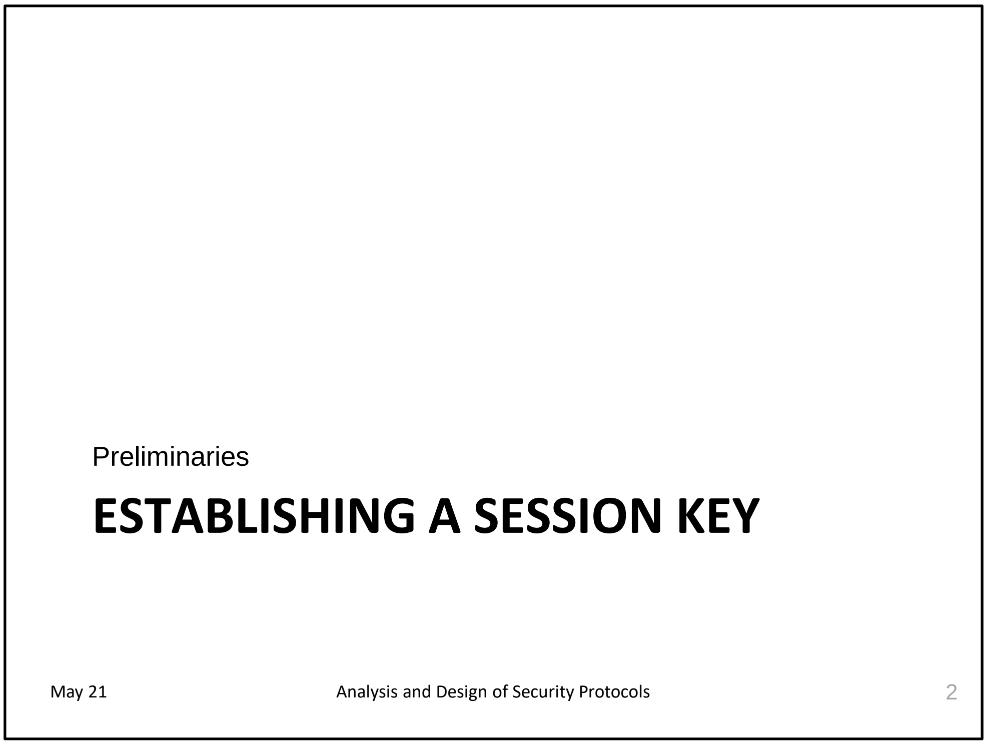
Analysis and design of cryptographic protocols

GIANLUCA DINI

Dept. of Ingegneria dell'Informazione
University of Pisa

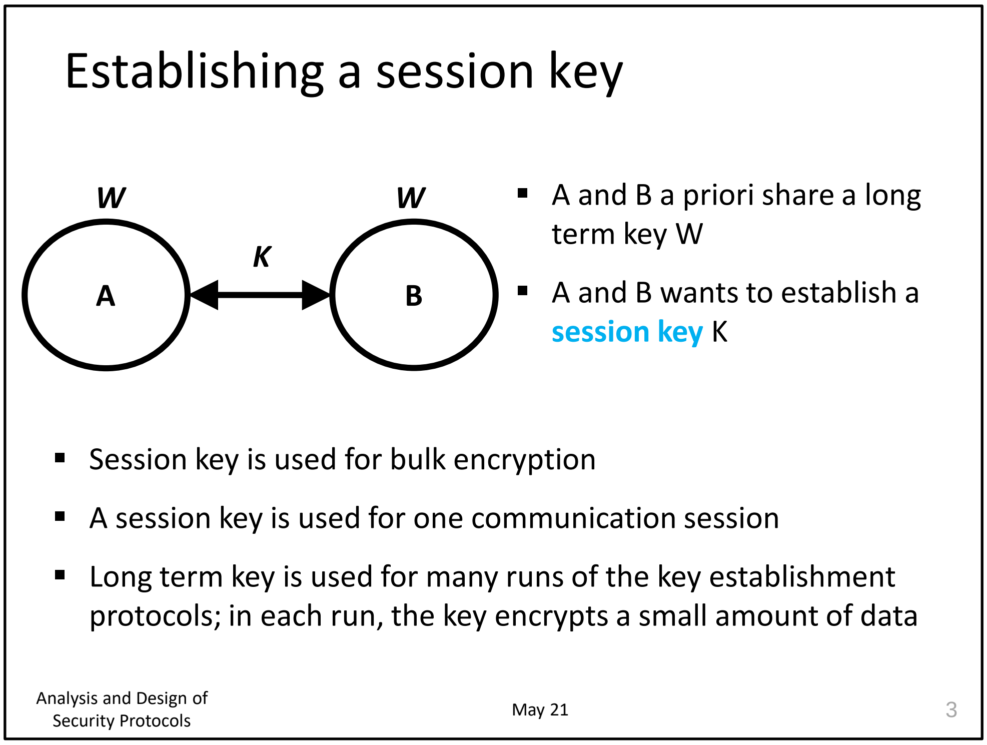
Email: gianluca.dini@unipi.it

Version: 2021-05-16

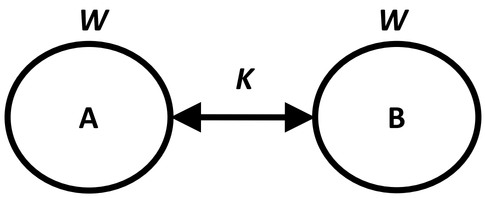


Preliminaries

ESTABLISHING A SESSION KEY



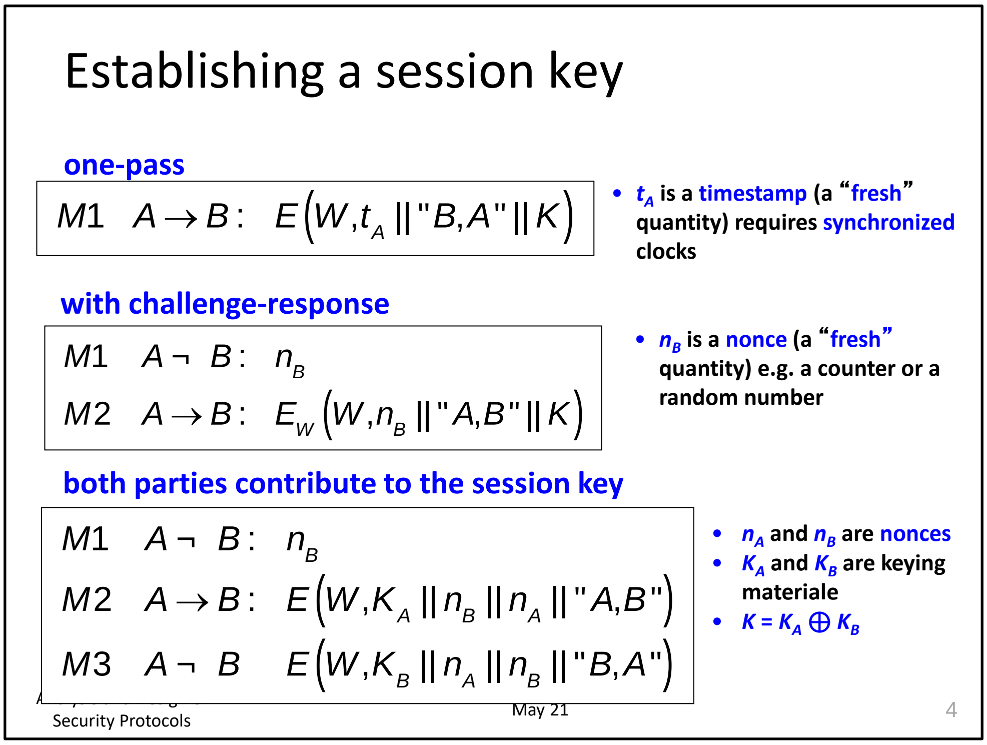
Establishing a session key



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

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

In order to reduce the amount of data available to a crypto-analyst, we strive for reducing the amount of data encrypted by means of the same key. For this reason we distinguish long term keys from *session keys*.
A session key is used once to encrypt a possible large amount of data. A long-term key is used to many times to encrypt small amounts of data.



Establishing a session key

one-pass

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

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

with challenge-response

$$M1 \quad A \neg \; B: \quad n_B$$
$$M2 \quad A \rightarrow B: \quad E_W\left(W, n_B \,||\, "A,B" \,||\, K\right)$$

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

both parties contribute to the session key

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

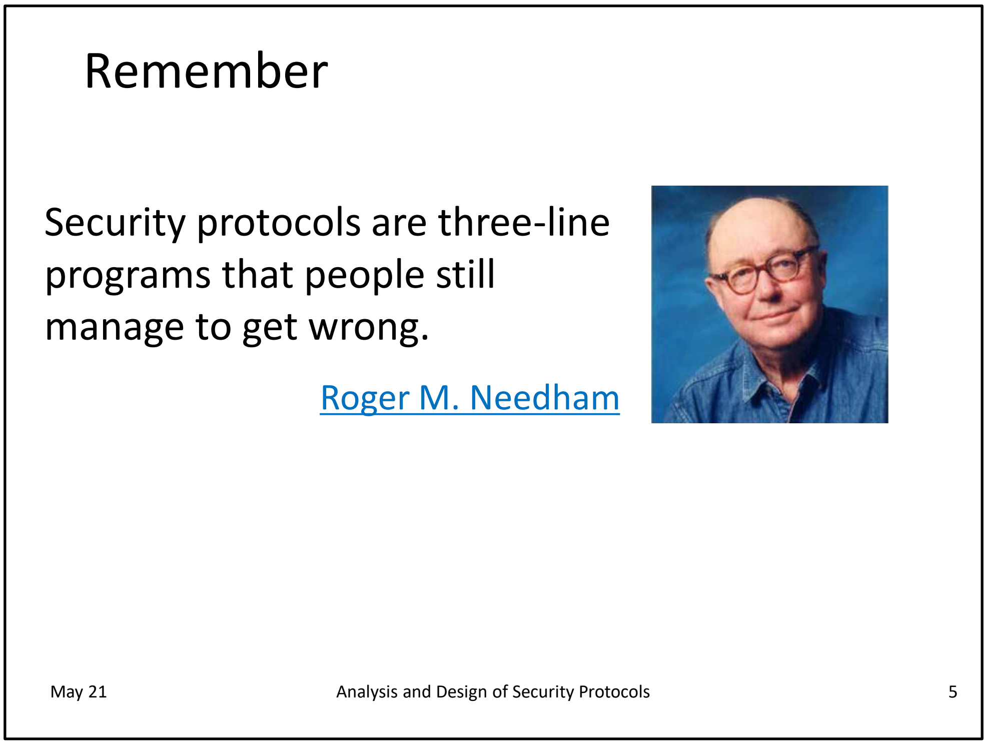
- ***n_A* and *n_B* are nonces**
- ***K_A* and *K_B* are keying materiale**
- ***$K = K_A \oplus K_B$***

Security Protocols

May 21

4

PROTOCOLLO ONE-PASS. Upon receiving message M1, Bobs believes that the message *is coming from* Alice because it is encrypted by means of the shared key W. Is that always true? Reasoning about shared keys (or digital signatures) we can make conjectures about the originator of a message. However, sender and originator may not always coincide. 1) Message M1 may reach Alice from Bob through an intermediary Carol: 2) Message M1 may not come from its originator if the message is a replay.
1. The timestamp t_A avoids replay but requires *synchronized clocks* (monotonic non-decreasing clocks). Without the timestamp, the message could be *replayed* and the adversary could impersonate Alice and force Bob to re-use key *K*. In practice all the traffic encrypted by K could be be replied.
This is itself a vulnerability. However, it becomes particularly dangerous if the adversary has obtained a session key K. In that case, the adversary would be able to impersonate Alice with respect to Bob and force the former to accept key K as session key. *The adversary could do this as many times as he wishes*. *It is always a good practice to assume that a session key may get compromised and that the adversary holds it as well as the sequence of messages that led to the establishment of that key.*
It follows that a freshness proofs allows Bob to believe that the message is "recent", it is not a replay, and thus is coming from the legitimate originator.
2. The string "B, A" means that the key is for B to communicate with A. This avoids, for example, that the message can be reused for different communications.
3. This protocol requires *secure clock synchronization* which is very hard in a distributed system. In particular, secure clock synchronization requires authenticated messages, that is, it requires a secure protocol, namely, exactly what we are trying to do.
PROTOCOL WITH CHALLENGE-RESPONSE. This protocol avoids clock synchronization but uses *nonces*, i.e., *quantities that are used just once*. It follows that a nonce must be generated fresh, i.e., it can be never used in two different instances of the protocol. A nonce can be implemented as a counter, a timestamp, or a random number. The former two are foreseeable, whereas the latter is not. In certain situations, this difference matters. Differently from a timestamp, the freshness of a nonce can be verified by the nonce generator only. In general a nonce is transmitted in the clear because it does not carry any confidential information.
PROTOCOL WHERE BOTH PARTIES CONTRIBUTE TO THE SESSION KEY. In the first two protocols, the session key is generated by one party. This assume that Bob has to trust Alice about key generation. This protocol allows each party to contribute with his own piece of key. Is it possible to reason in more formal way?

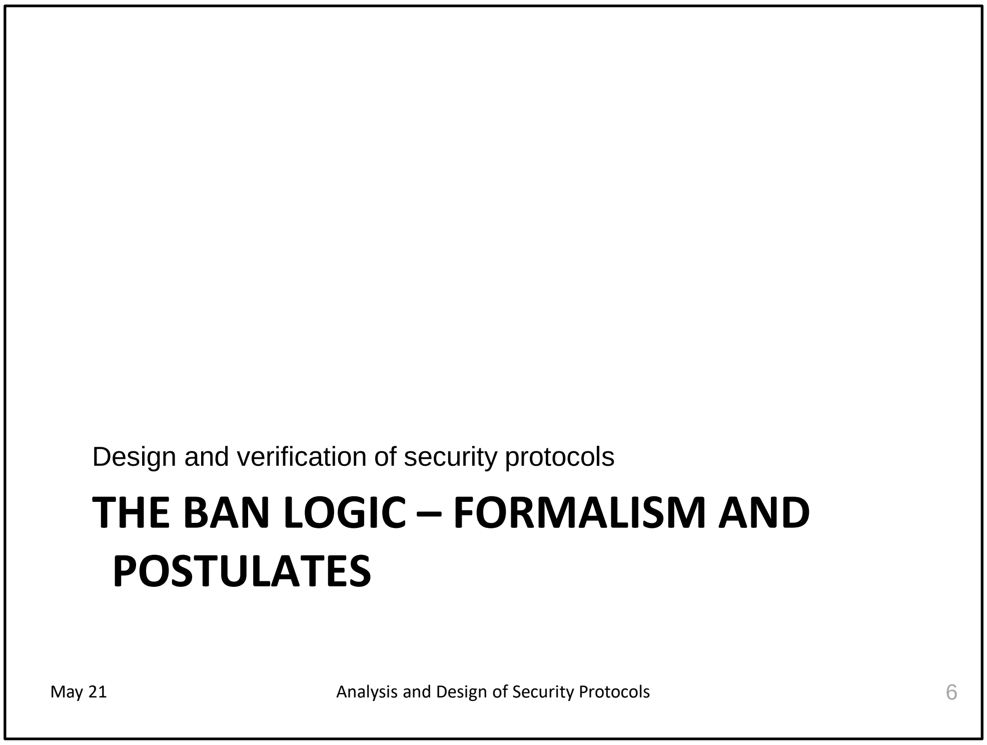


Remember

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

Roger M. Needham

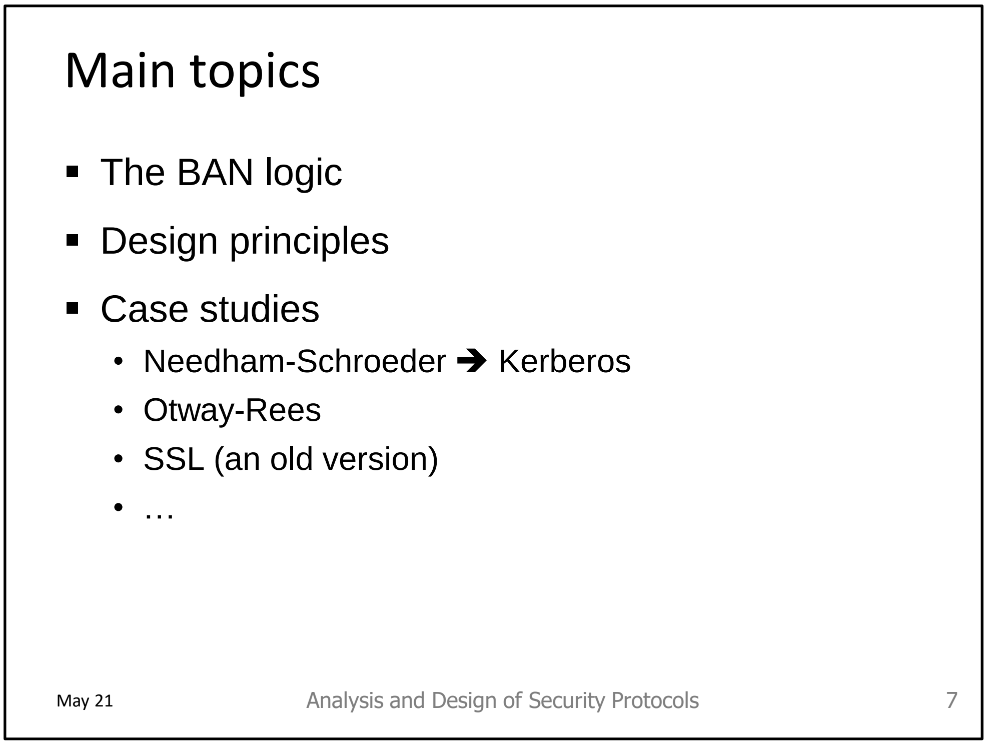
May 21 Analysis and Design of Security Protocols 5



Design and verification of security protocols

THE BAN LOGIC – FORMALISM AND POSTULATES

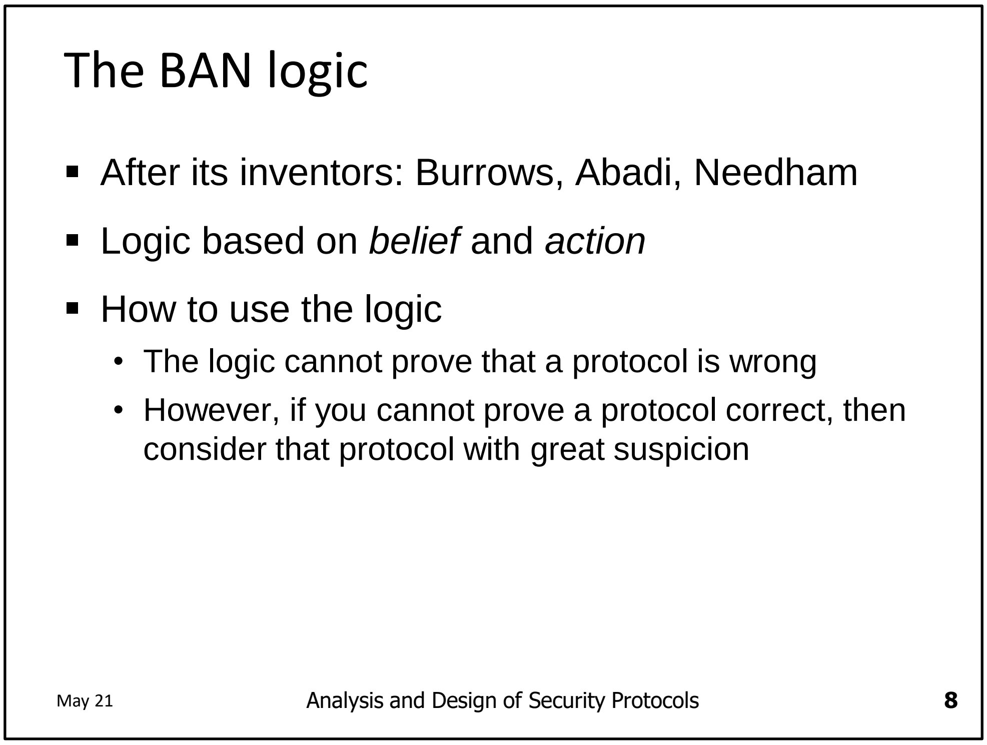
May 21 Analysis and Design of Security Protocols 6



Main topics

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

May 21 Analysis and Design of Security Protocols 7

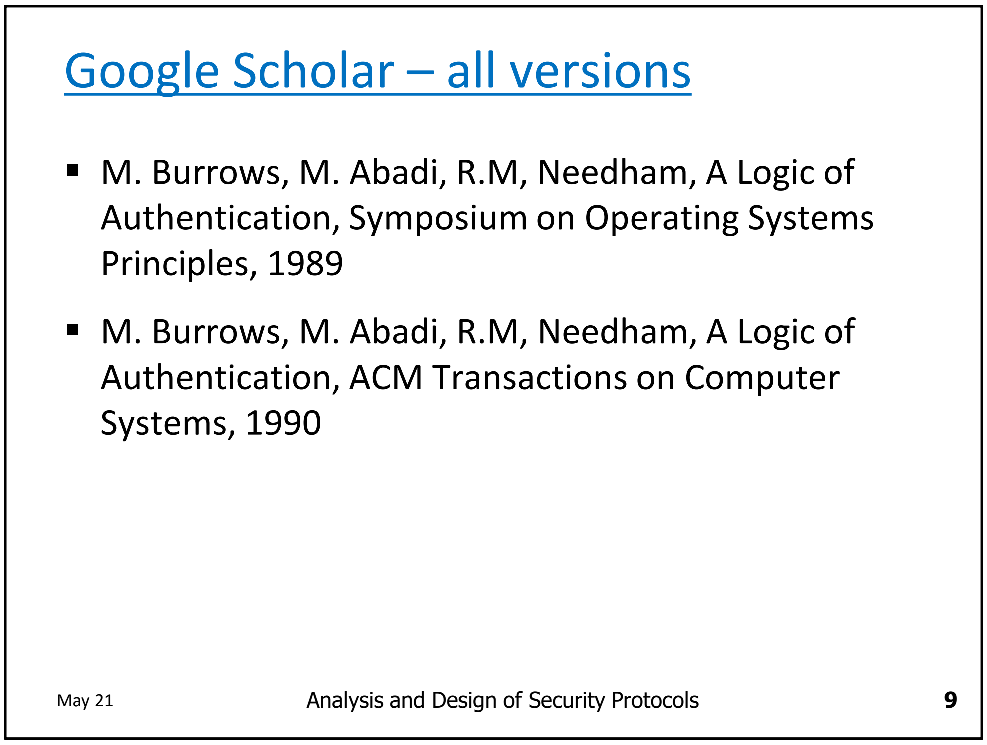


The BAN logic

- After its inventors: Burrows, Abadi, Needham
- 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

May 21 Analysis and Design of Security Protocols **8**

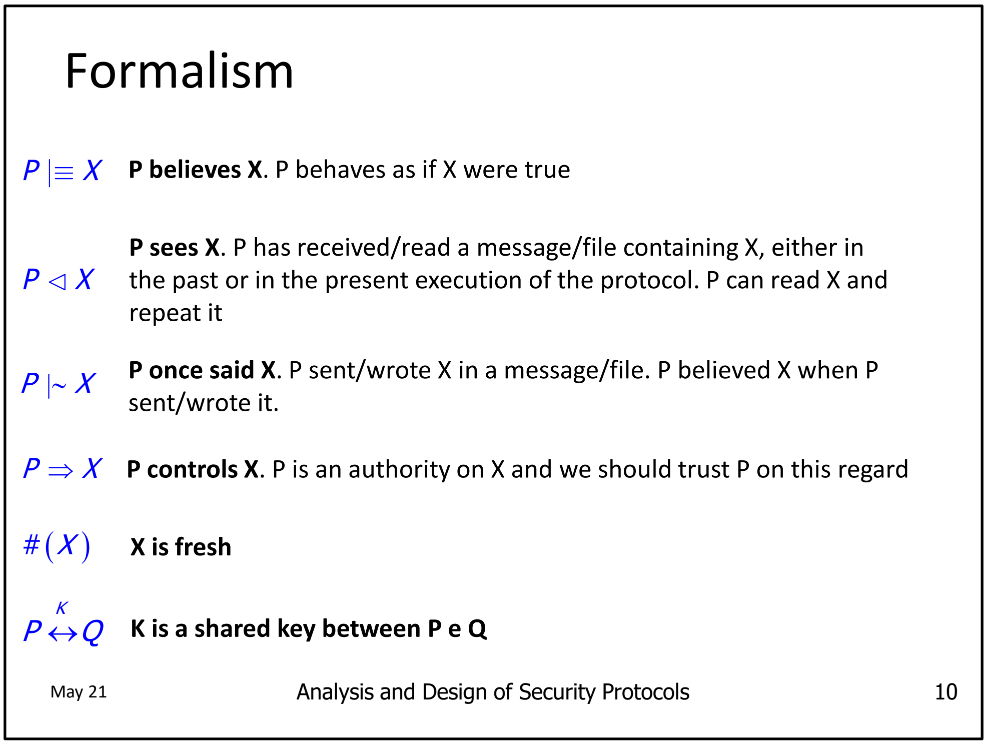
It’s a logic, a formal method, but it is intuitive. It allows you to formalize a protocol and then check whether the protocol satisfies certain properties.
It does not make miracles. You don’t give the logic the protocol and the then the logic tells you whether it is wrong and where it is bugged. Rather it is a tool to reason upon a security protocol. The rationale is: use the Logic to reason upon the protocol, namely to check whether the protocol satisfies certain properties. If you get to the end you cannot say anything. If you get stuck somewhere, then your protocol, and that point in particilaur, needs attention.
Finally, notice that here we work at specification level. Then you have plentiful of oppoirtunities to put mistakes in design and implementation/coding phase.



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

May 21 Analysis and Design of Security Protocols **9**

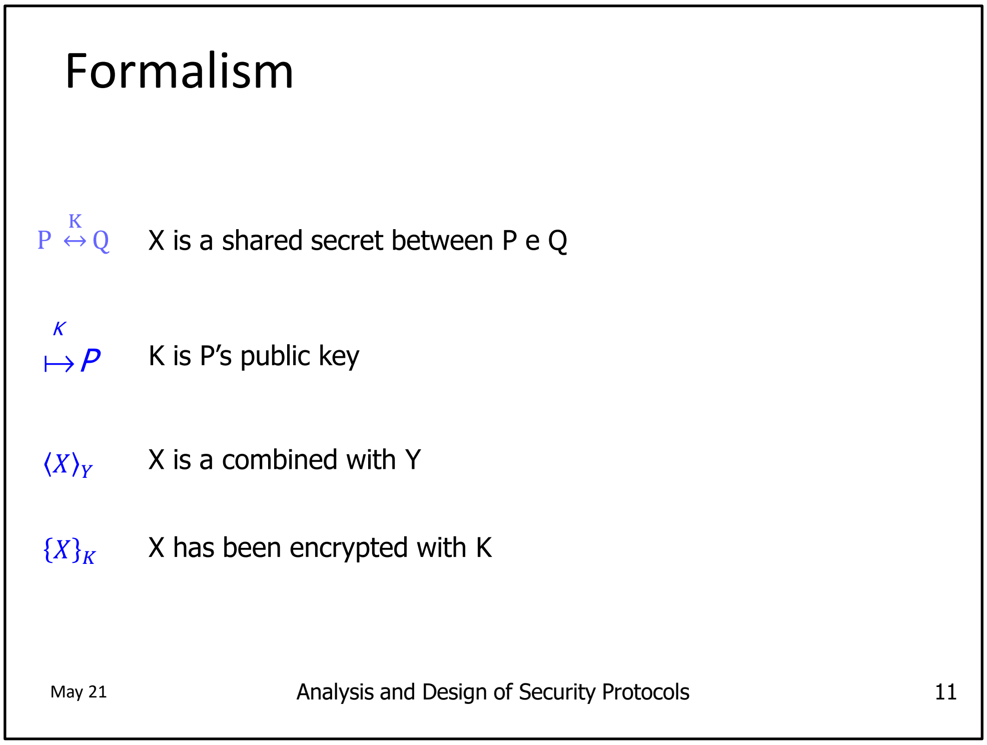


Formalism

$P \mid\equiv X$	**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**

May 21 Analysis and Design of Security Protocols 10

- **P believes X.** P behaves as if X were true
- **P sees X.** P has received a message containing X, either in the past or in the present execution of the protocol. P can read X and repeat it.
- **P once said X**. P sent a message containing X; P *believed* X when P sent X.
- **P controls X**. P is an authority on X and we should trust P on this regard.



Formalism

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

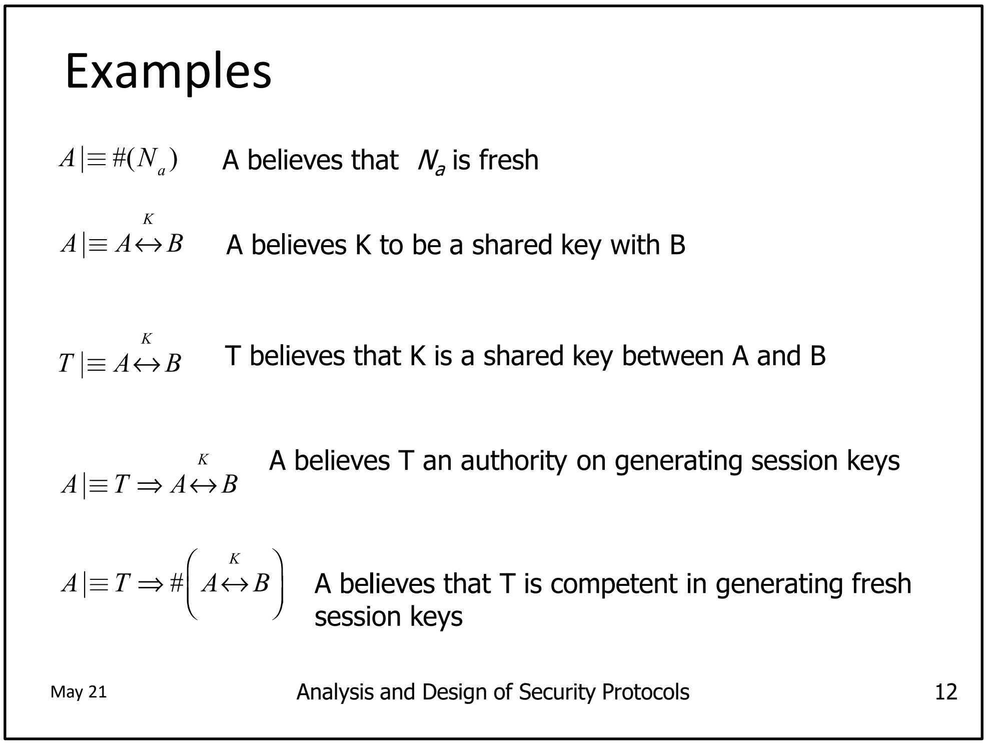
$\overset{K}{\mapsto} P$ K is P’s public key

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

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

May 21 Analysis and Design of Security Protocols 11

- **P believes X.** P behaves as if X were true
- **P sees X.** P has received a message containing X, either in the past or in the present execution of the protocol. P can read X and repeat it.
- **P once said X**. P sent a message containing X; P *believed* X when P sent X.
- **P controls X**. P is an authority on X and we should trust P on this regard.



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

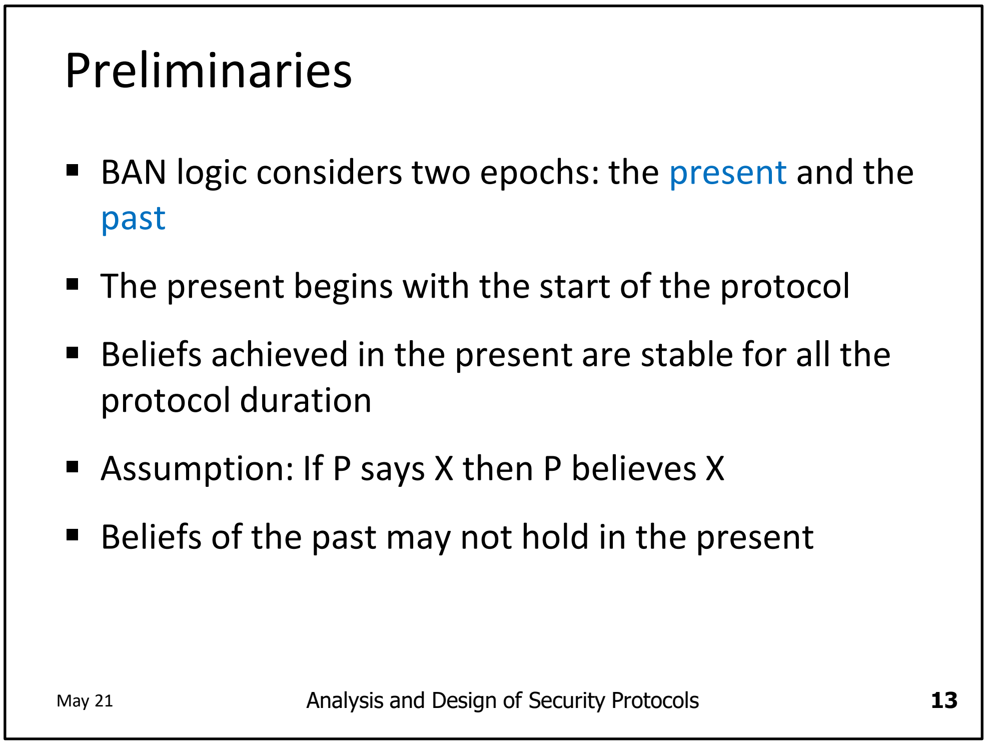
$T \mid\equiv A \overset{K}{\leftrightarrow} B$ T believes that K is a shared key between A and B

$A \mid\equiv T \Rightarrow A \overset{K}{\leftrightarrow} B$ 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

May 21 Analysis and Design of Security Protocols 12

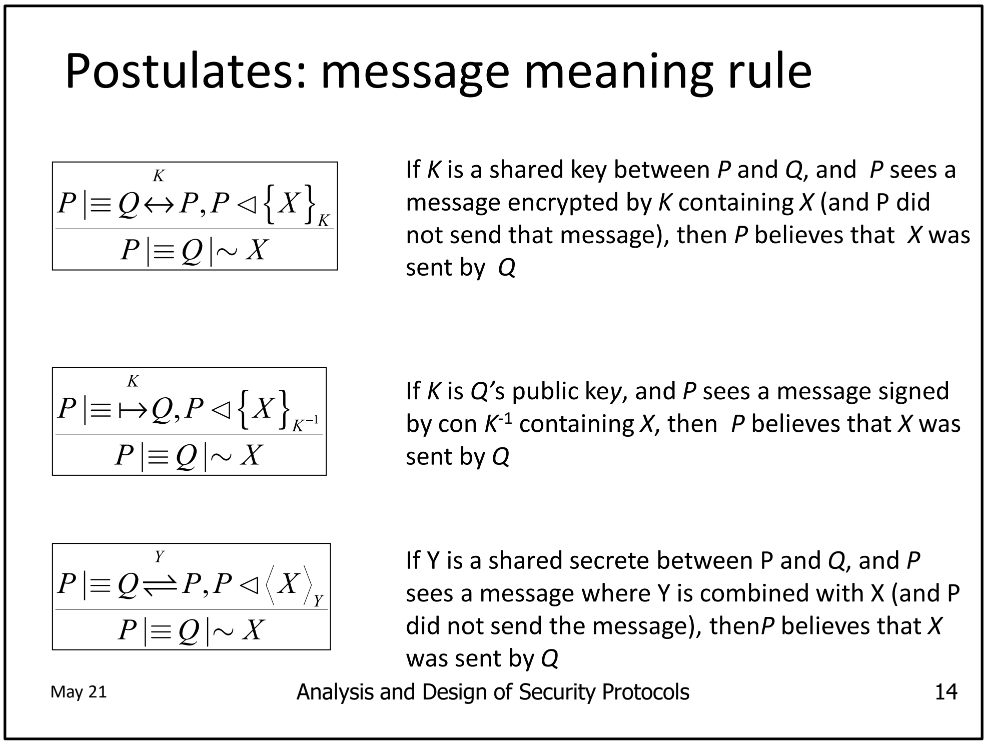
We can use the formalism to build «sentences».



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
- Assumption: If P says X then P believes X
- Beliefs of the past may not hold in the present

May 21 Analysis and Design of Security Protocols **13**



Postulates: message meaning rule

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

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*

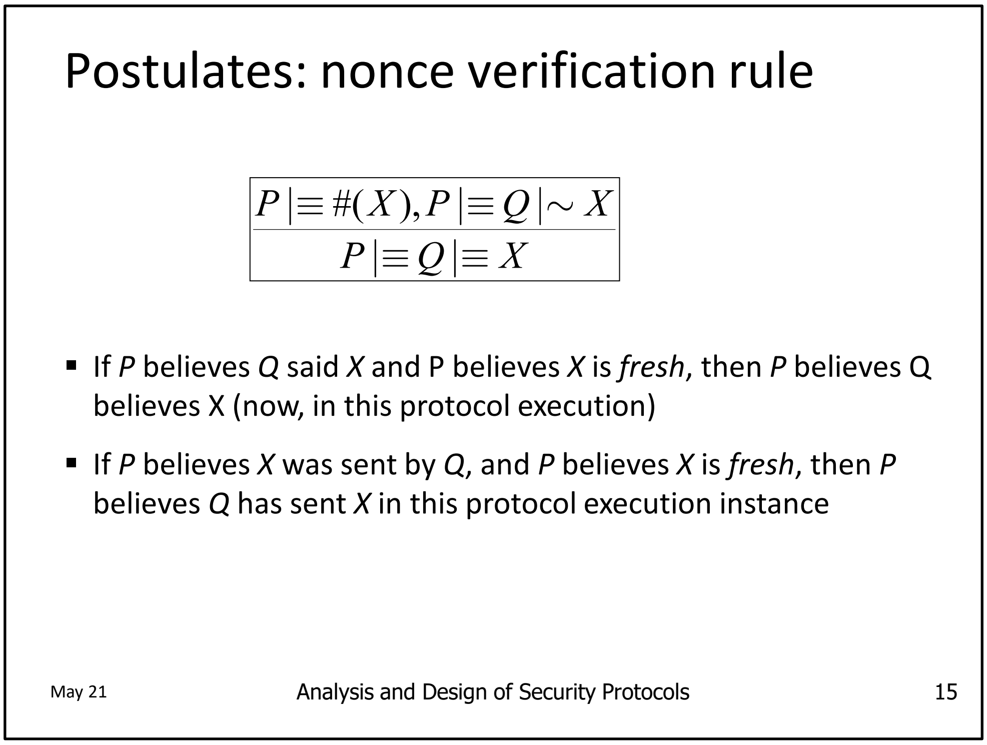
$$\boxed{\frac{P \mid\equiv \overset{K}{\mapsto} Q, P \triangleleft \{X\}_{K^{-1}}}{P \mid\equiv Q \mid\sim X}}$$

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*

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

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*

May 21 Analysis and Design of Security Protocols 14

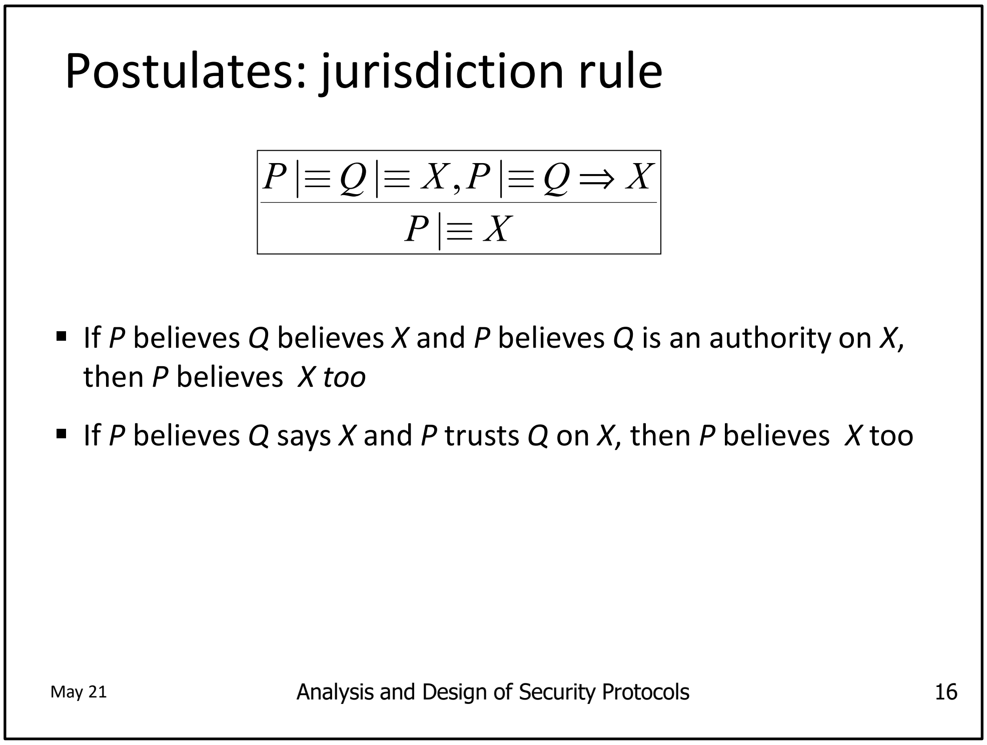


Postulates: nonce verification rule

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

- 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

May 21 Analysis and Design of Security Protocols 15

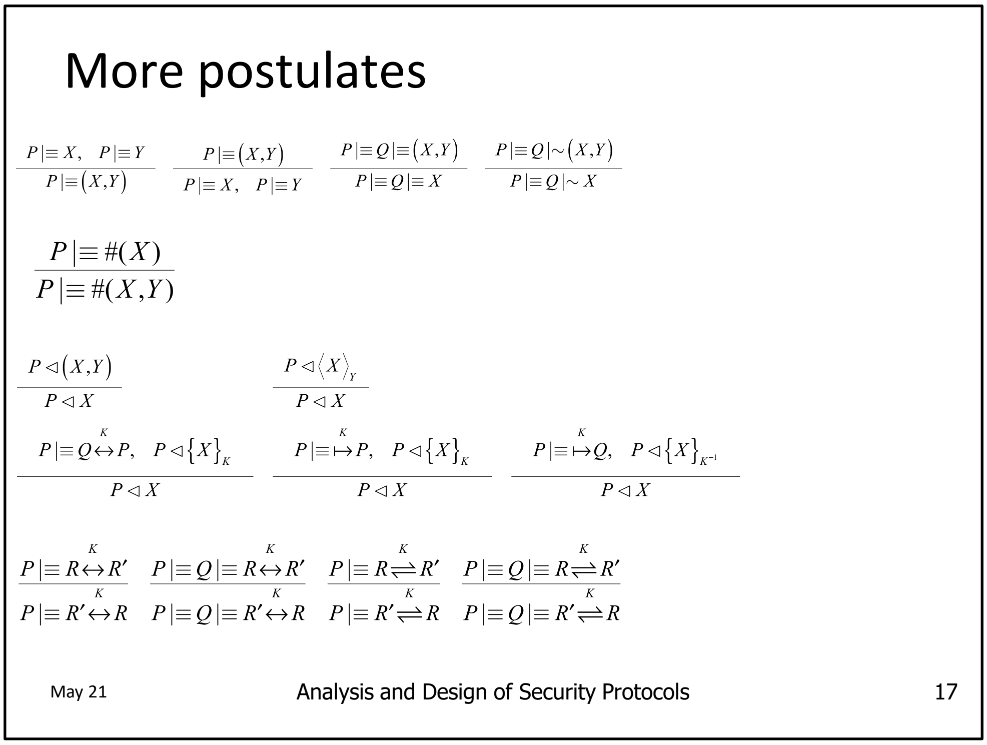


Postulates: jurisdiction rule

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

- 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

May 21 Analysis and Design of Security Protocols 16



More postulates

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

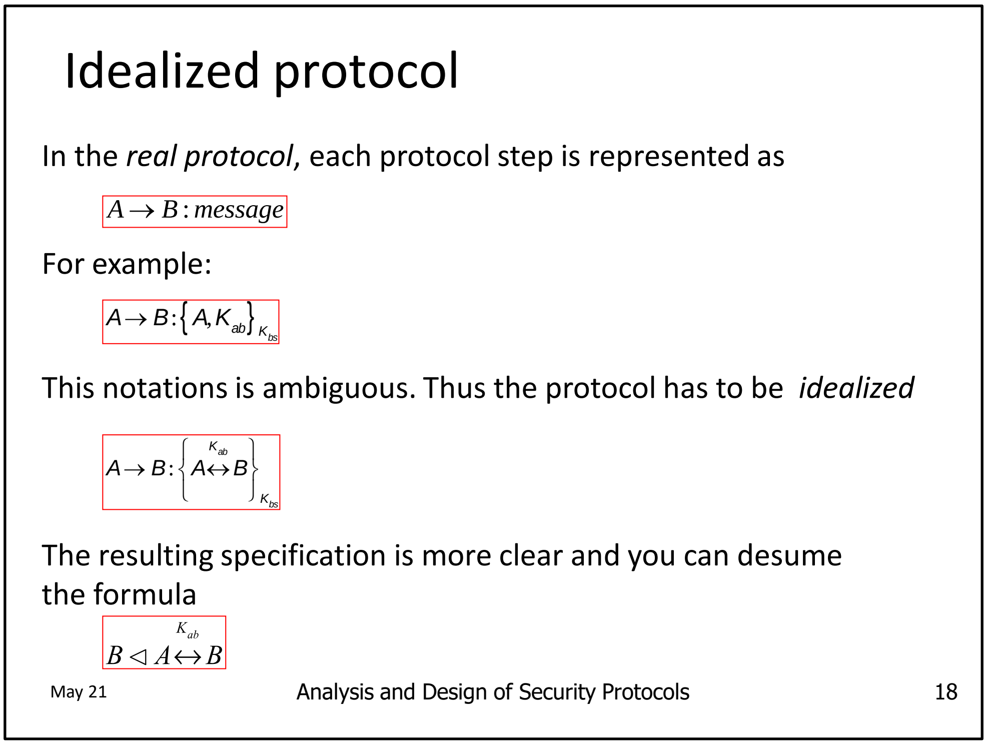
$$\frac{P \mid\equiv \#(X)}{P \mid\equiv \#(X,Y)}$$

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

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

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

May 21 Analysis and Design of Security Protocols 17



Idealized protocol

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

$$\boxed{A \to B : message}$$

For example:

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

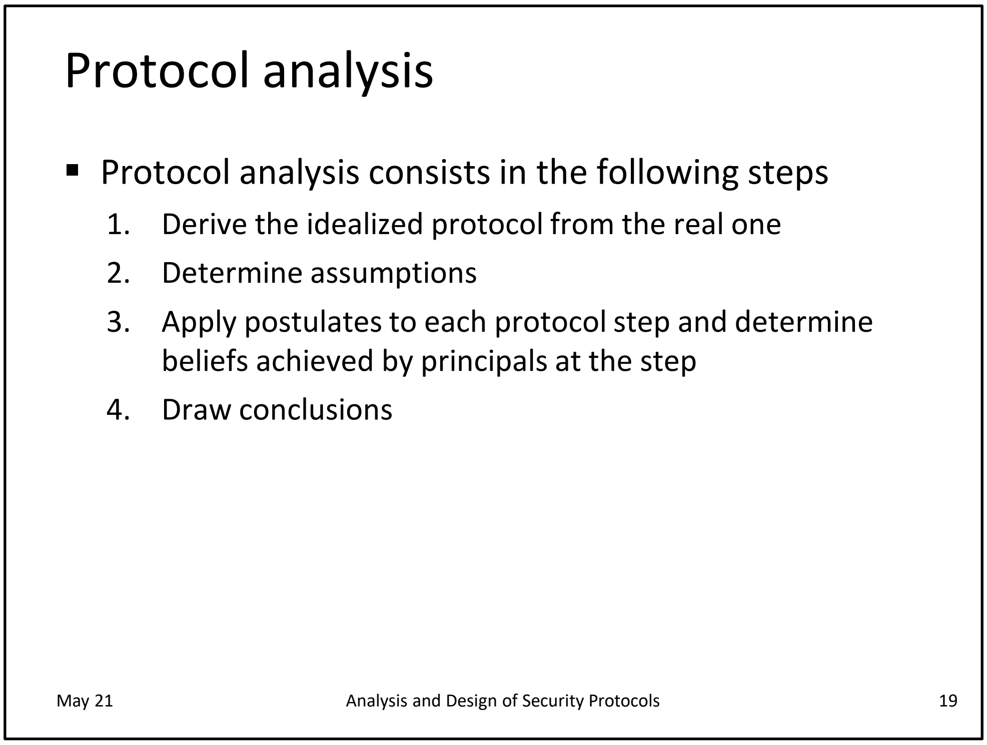
This notations is ambiguous. Thus the protocol has to be *idealized*

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

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

$$\boxed{B \triangleleft A \overset{K_{ab}}{\leftrightarrow} B}$$

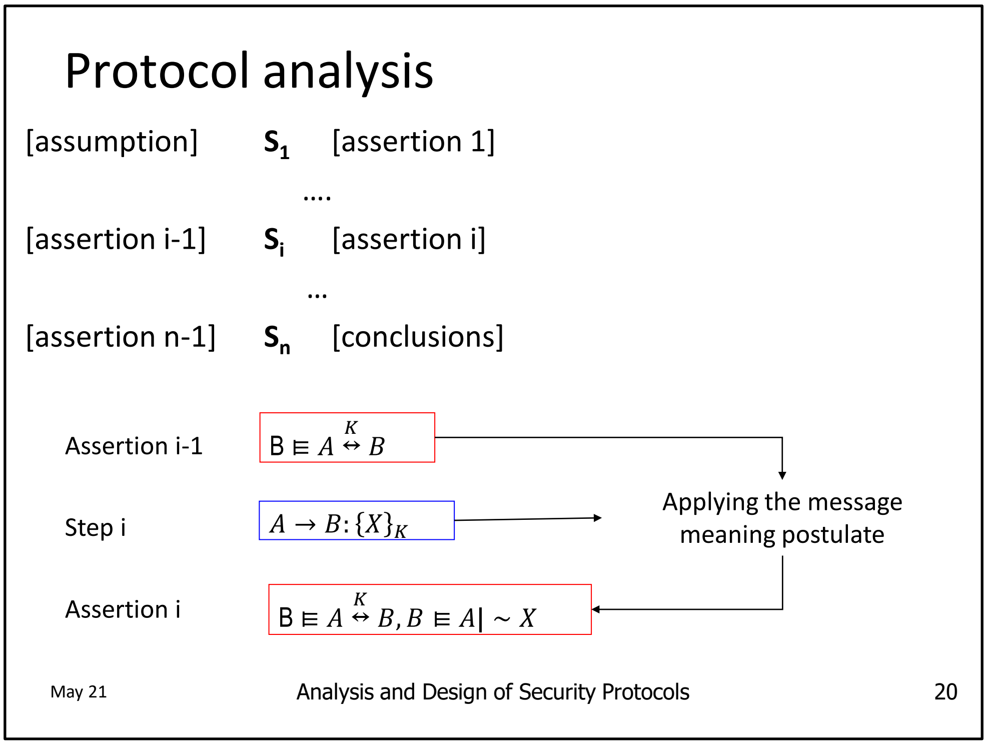
May 21 Analysis and Design of Security Protocols 18



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

May 21 Analysis and Design of Security Protocols 19



Protocol analysis

[assumption]	$\mathbf{S_1}$	[assertion 1]
	....	
[assertion i-1]	$\mathbf{S_i}$	[assertion i]
	...	
[assertion n-1]	$\mathbf{S_n}$	[conclusions]

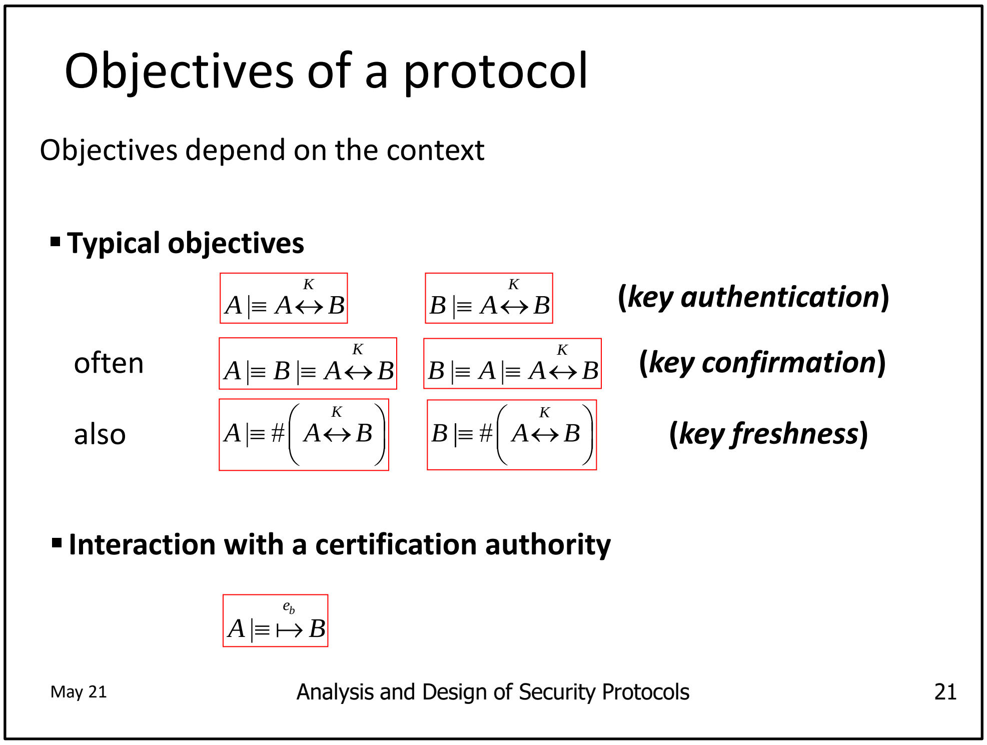
Assertion i-1 $\boxed{B \mid\equiv A \overset{K}{\leftrightarrow} B}$

Step i $\boxed{A \rightarrow B: \{X\}_K}$

Applying the message meaning postulate

Assertion i $\boxed{B \mid\equiv A \overset{K}{\leftrightarrow} B, B \mid\equiv A \mid\sim X}$

May 21 Analysis and Design of Security Protocols 20



Objectives of a protocol

Objectives depend on the context

- **Typical objectives**

$\boxed{A \mid\equiv A \overset{K}{\leftrightarrow} B}$ $\boxed{B \mid\equiv A \overset{K}{\leftrightarrow} B}$ ***(key authentication)***

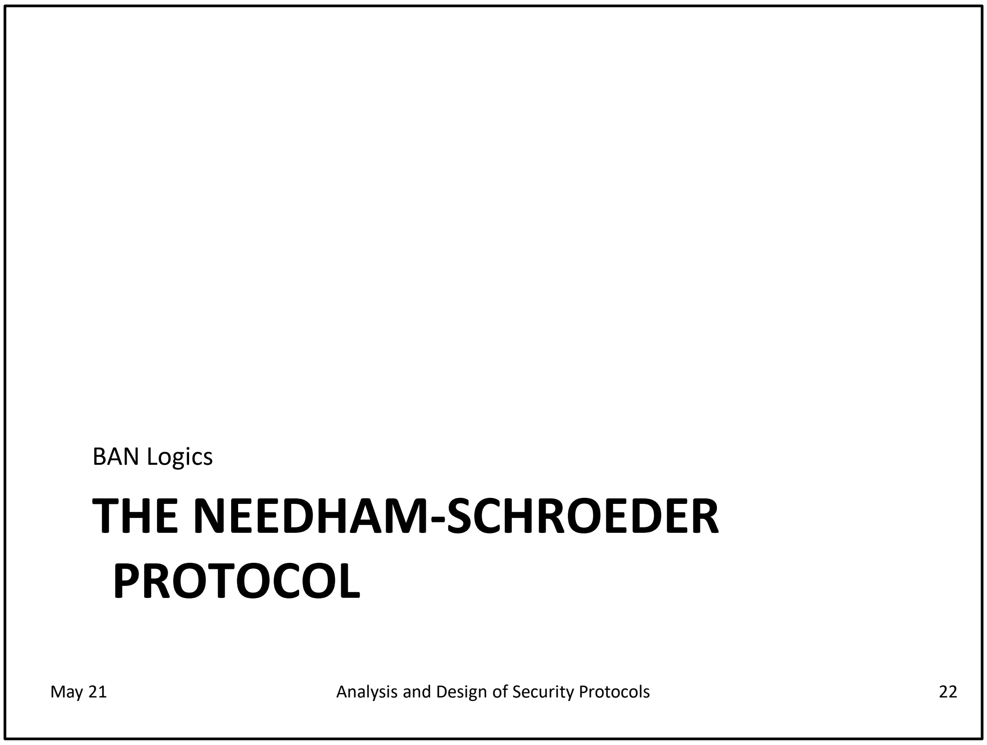
often $\boxed{A \mid\equiv B \mid\equiv A \overset{K}{\leftrightarrow} B}$ $\boxed{B \mid\equiv A \mid\equiv A \overset{K}{\leftrightarrow} B}$ ***(key confirmation)***

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

- **Interaction with a certification authority**

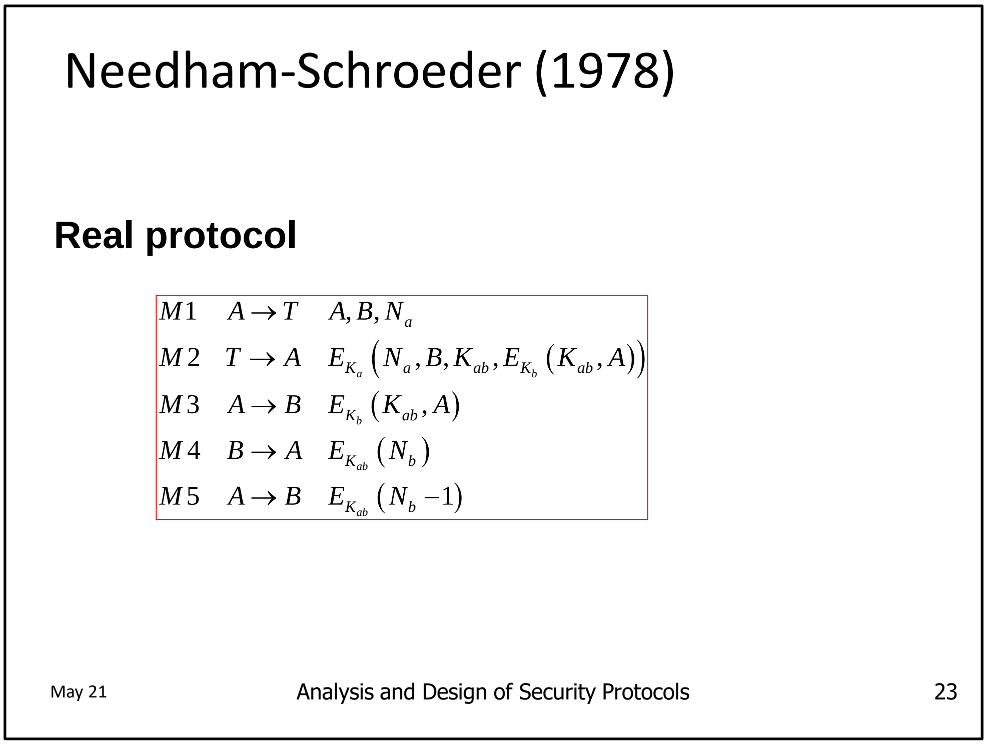
$\boxed{A \mid\equiv \overset{e_b}{\mapsto} B}$

May 21 Analysis and Design of Security Protocols 21



BAN Logics

THE NEEDHAM-SCHROEDER PROTOCOL



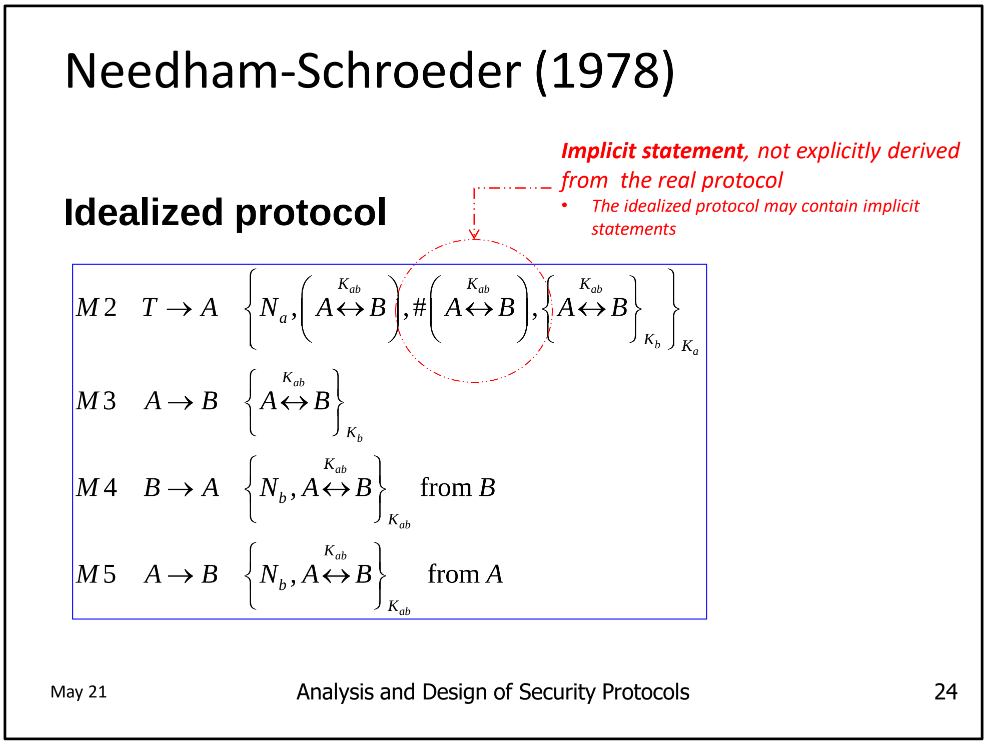
Needham-Schroeder (1978)

Real protocol

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

May 21 Analysis and Design of Security Protocols 23

This protocol was invented in 1978.
After a few modifications, it became the basis for Kerberos and Active Directory.



Needham-Schroeder (1978)

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

- *The idealized protocol may contain implicit statements*

Idealized protocol

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

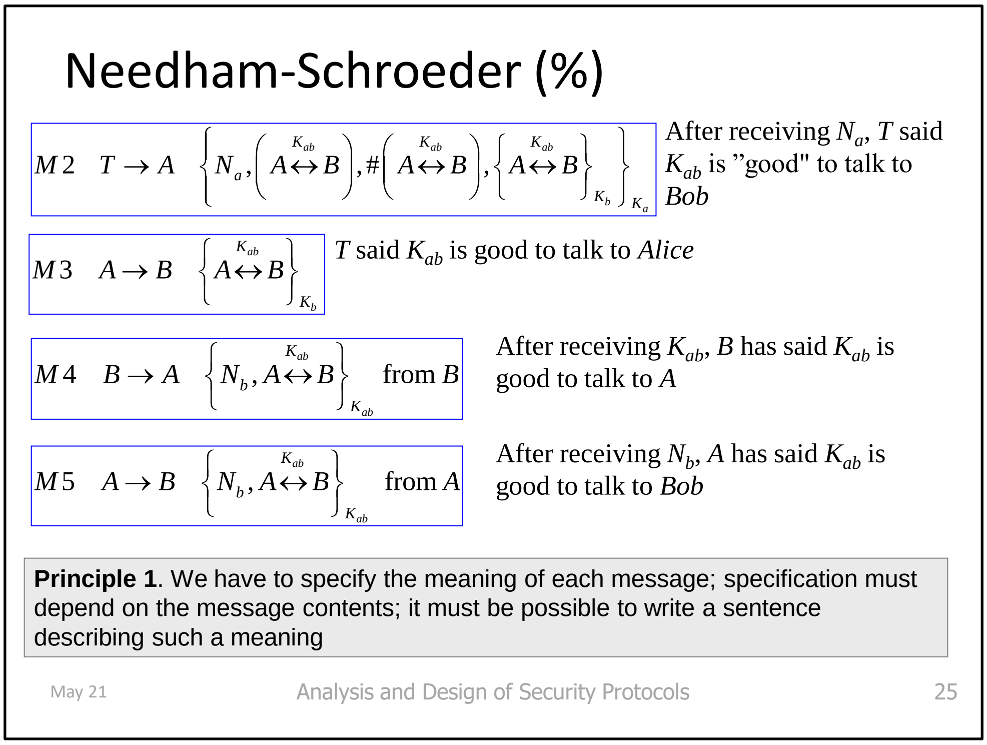
$$M3 \quad A \to B \quad \left\{ A \overset{K_{ab}}{\leftrightarrow} B \right\}_{K_b}$$

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

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

May 21 Analysis and Design of Security Protocols 24

Notice that M2 contains a statement about kab freshness. This statement is necessary because, otherwise, Alice cannot derive the freshness of message M4. Such a message does not contain any proof of freshness but the fact that it has been encrypted by means of kab which is assumed to be created fresh (by assumption). This fact will be evident during the analysis. *It is worthwhile to notice that the idealized protocol may contains implicit statement in addition to those that are explicitly present in the real protocol.*



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*

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

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

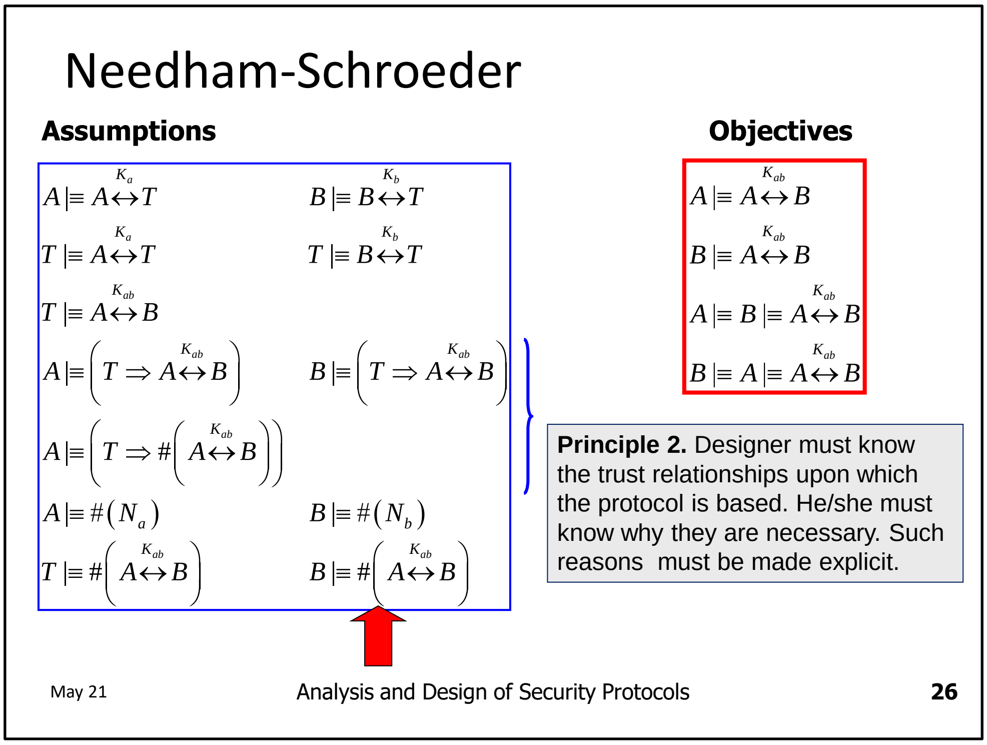
$M5 \quad A \to B \quad \left\{N_b, A \overset{K_{ab}}{\leftrightarrow} B\right\}_{K_{ab}}$ from A

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

May 21 Analysis and Design of Security Protocols 25

It is a good analysis practice to write down a sentence that explains the meaning (semantics) of a message.
It is a good design practice to design protocol whose messages are self-contained, that is, their messages do not depend on the context.



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

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

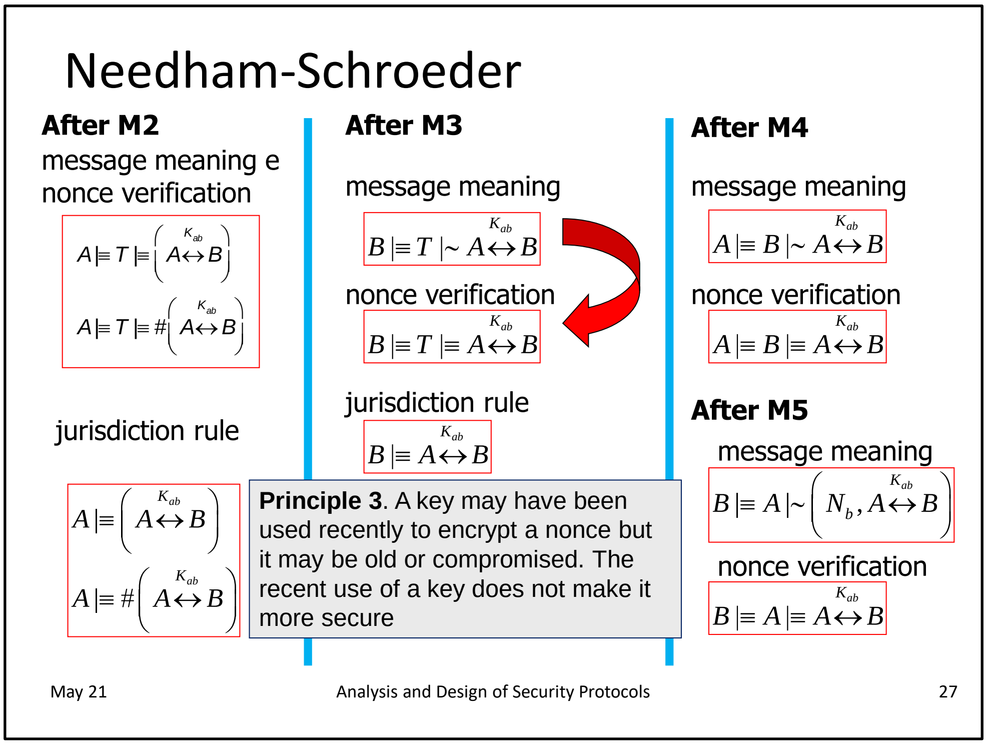
$B \mid\equiv A \overset{K_{ab}}{\leftrightarrow} B$

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

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

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.

May 21 Analysis and Design of Security Protocols **26**



Needham-Schroeder

After M2

message meaning e nonce verification

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

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

jurisdiction rule

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

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

After M3

message meaning

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

nonce verification

$B \mid\equiv T \mid\equiv A \xleftrightarrow{K_{ab}} B$

jurisdiction rule

$B \mid\equiv A \xleftrightarrow{K_{ab}} B$

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 M4

message meaning

$A \mid\equiv B \mid\sim A \xleftrightarrow{K_{ab}} B$

nonce verification

$A \mid\equiv B \mid\equiv A \xleftrightarrow{K_{ab}} B$

After M5

message meaning

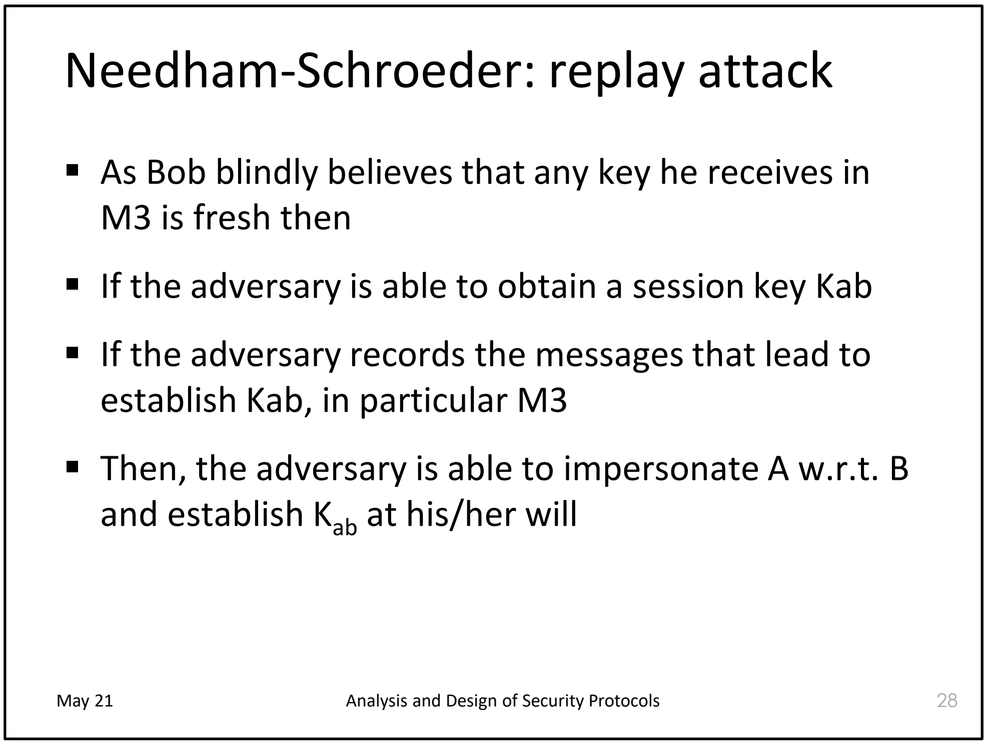
$B \mid\equiv A \mid\sim \left(N_b, A \xleftrightarrow{K_{ab}} B\right)$

nonce verification

$B \mid\equiv A \mid\equiv A \xleftrightarrow{K_{ab}} B$

May 21 Analysis and Design of Security Protocols 27

Message M4. In order to apply nonce verification to M4, Alice needs a fresh information. The only fresh information in M4 for Alice is that M4 has been encrypted by means of Kab which is proven fresh thanks to the additional field in the message idealized version of message M2. By assumption, T believes that Kab is fresh and for this reason T may put the statement fresh(Kab) in message M2.

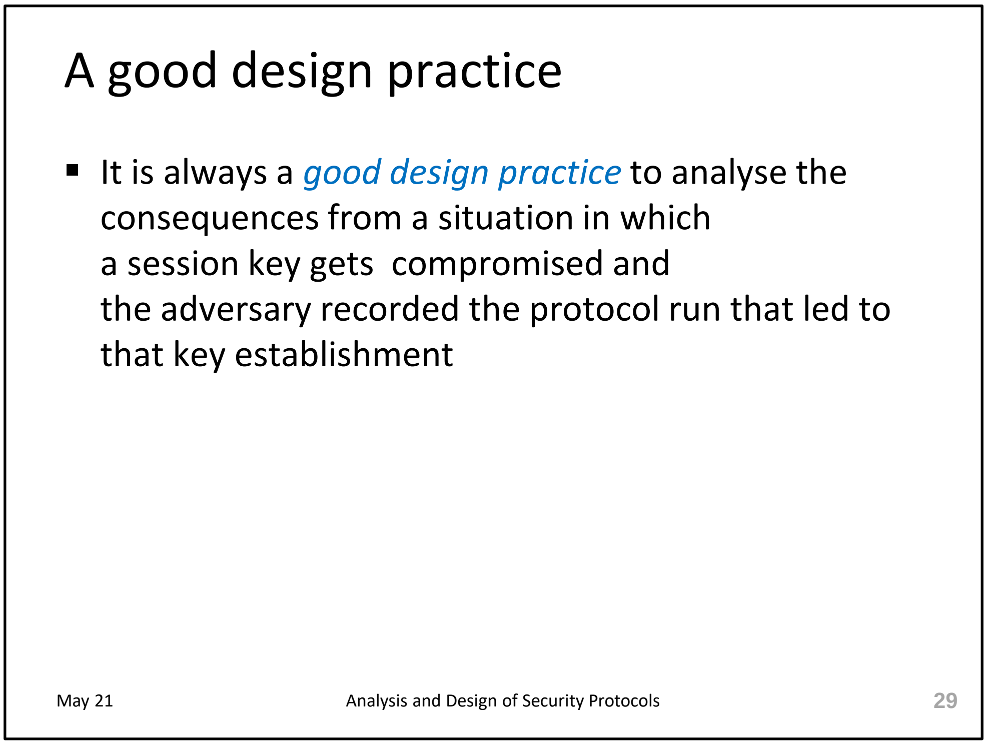


Needham-Schroeder: replay attack

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

May 21 Analysis and Design of Security Protocols 28

VULNERABILITY of the PROTOCOL – REPLAY ATTACK. The replay attack exploits the assumption thay B believes Kab fresh. Let us suppose that an adversary, somehow, gets to know a session key Kab and, contextually, manages to record the protocol instance that led to the establishment of such a key. In particular, let us assume that the adversary has recorded the related message M3. It follows that the adversary gets able to impersonate A w.r.t. B whenever (s)he likes by replaying message M3 and performing the subsequent protocol steps using the compromised key Kab. So doing, the adversary can (re)use Kab as session key. A possible countermeasure consists in using timestamps instead of nonces. This amendment has been done in the variation of the NSP that was implemented in Kerberos. However, timestamps require secure clock synchronization which, in turn, requires that authentication has been already solved.

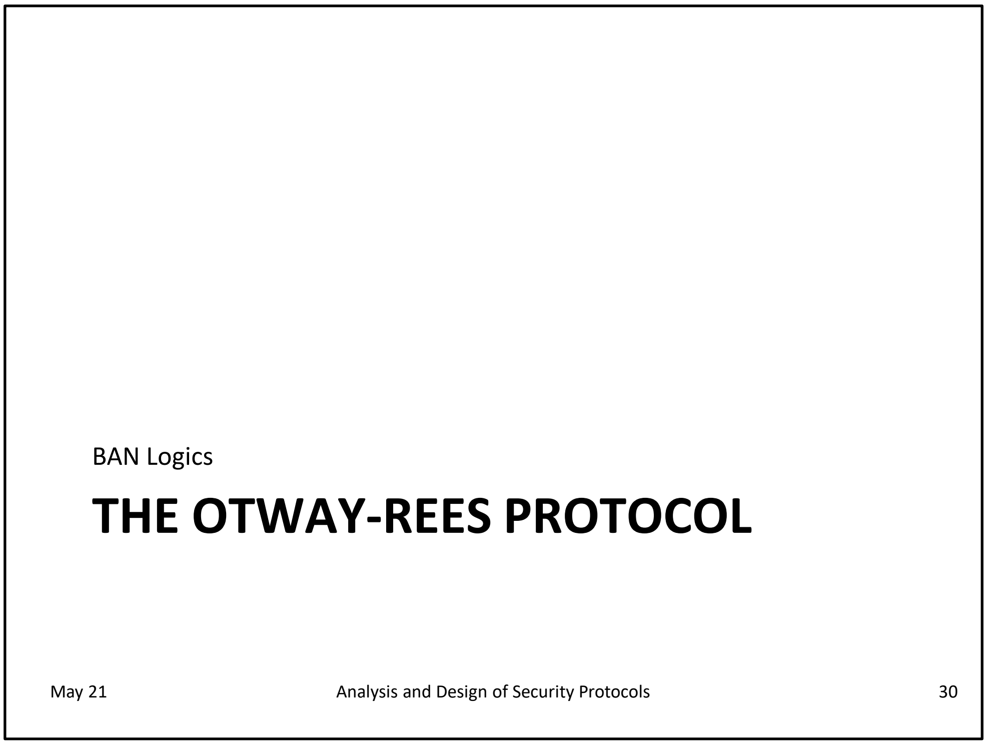


A good design practice

- 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

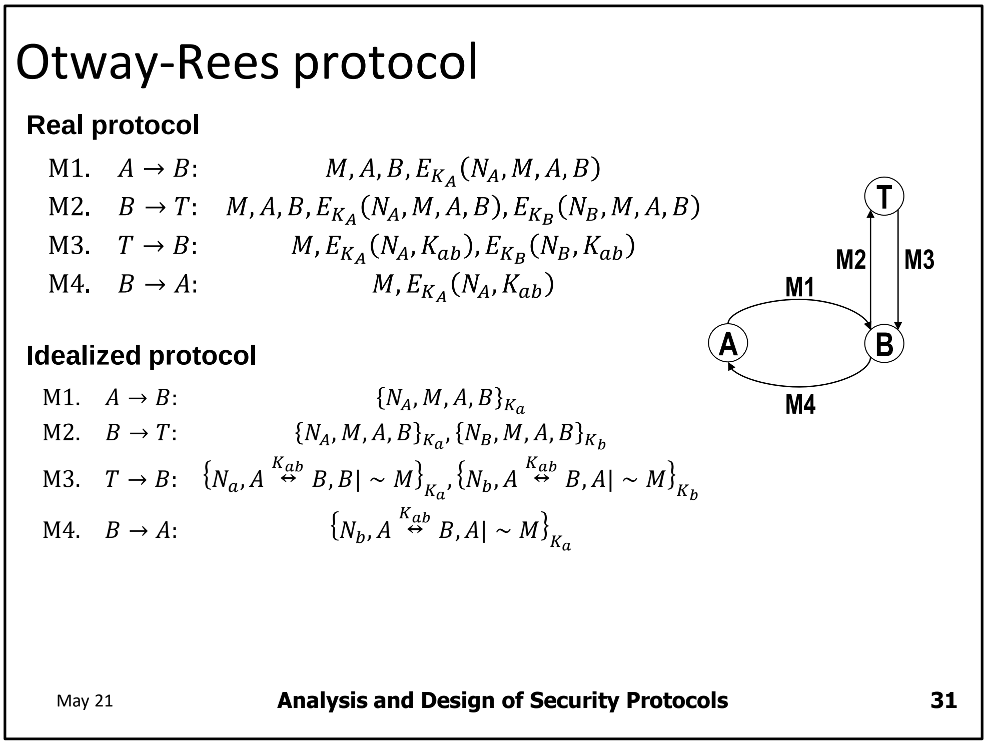
May 21 Analysis and Design of Security Protocols 29

In this case, session encrypted by means of the compromised session key gets completely lost. This is inevitable. However, we should have no further side effects.



BAN Logics

THE OTWAY-REES PROTOCOL



Otway-Rees protocol

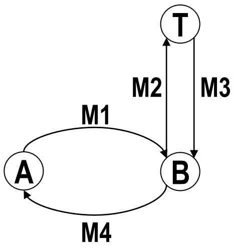
Real protocol

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

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

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

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



Idealized protocol

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

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

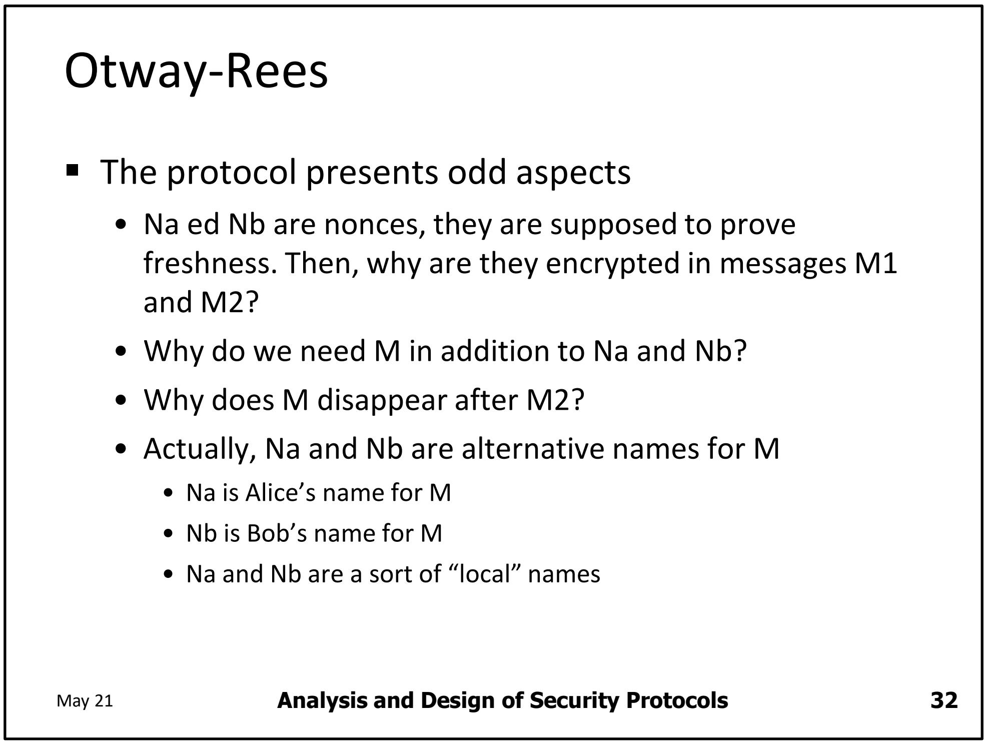
M3. $T \to B$: $\left\{N_a, A \overset{K_{ab}}{\leftrightarrow} B, B \mid\sim M\right\}_{K_a}, \left\{N_b, A \overset{K_{ab}}{\leftrightarrow} B, A \mid\sim M\right\}_{K_b}$

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

Centralized model with Trusted-Third Party (T). Each principal shares a long-term secret key with T. The protocol allows principals A and B to establish a session key Kab. The protocol is structured as two nested RPCs.
In the protocol, M represents the protocol instance identifier. It is a fresh quantity generated by A (the initiator).
The protocol shows odd issues:

1. Why is M necessary in addition to Na and Nb? After all Na and Nb guarantee freshness.
2. Why is M going to disappear after message M2?
3. Why are Na and Nb transmitted in their encrypted format in M1 and M2? All in all, they are nonces, they don’t carry any secret information, encrypting them in M1 and M2 would seem unnecessary at a first sight.

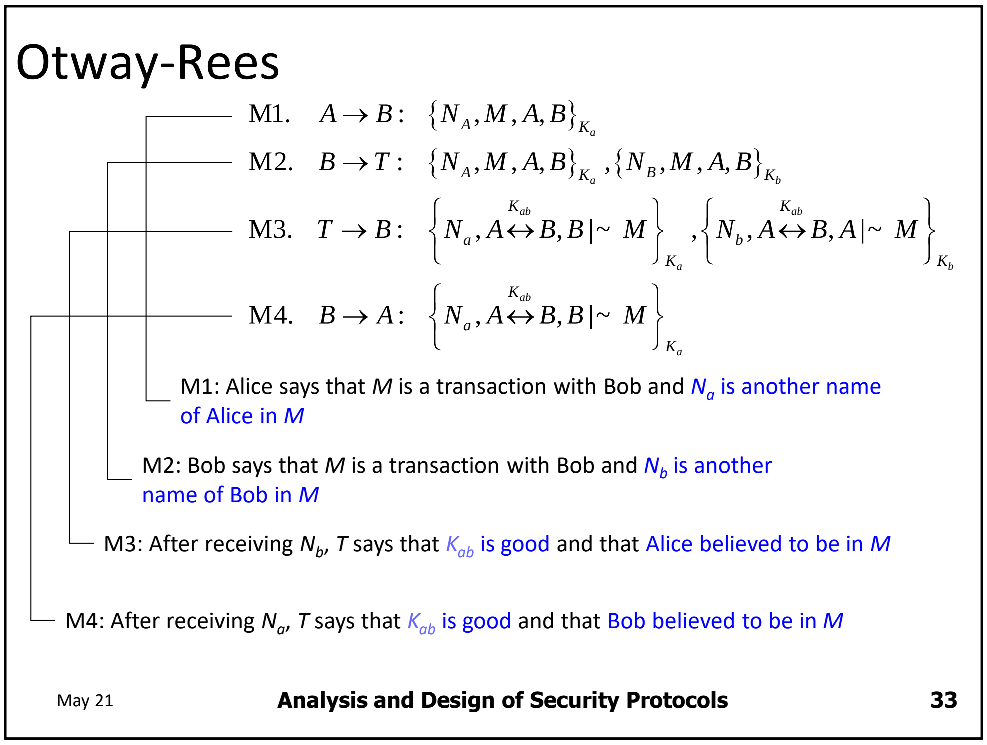
The answer is that Na and Nb are used as alternative names for M. More precisely, Na is another name of Alice in (protocol instance) M whereas Nb is another name of Bob in (protocol instance) M. Na and Nb are local names whereas M is a global name. Encryption in M1 and M2 is to link M to Na and Nb, respectively



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?
  - 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

May 21 **Analysis and Design of Security Protocols** 32



Otway-Rees

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

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

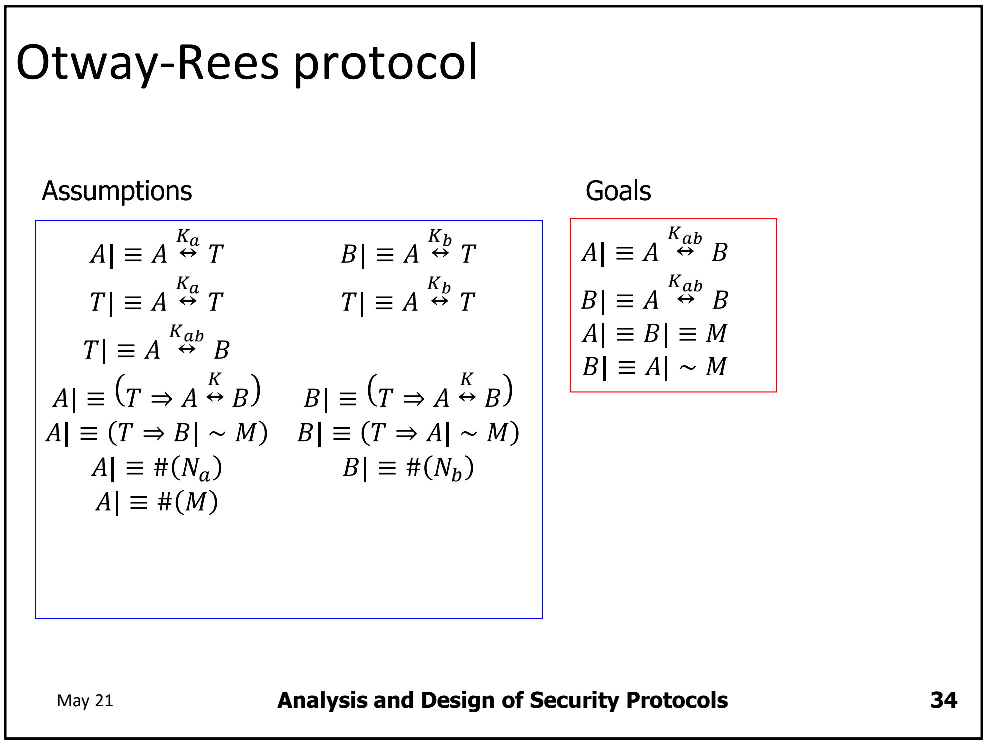
M1: Alice says that *M* is a transaction with Bob and N_a is another name of Alice in *M*

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*

May 21 **Analysis and Design of Security Protocols** **33**



Otway-Rees protocol

Assumptions

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

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

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

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

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

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

$A \mid\equiv \#(M)$

Goals

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

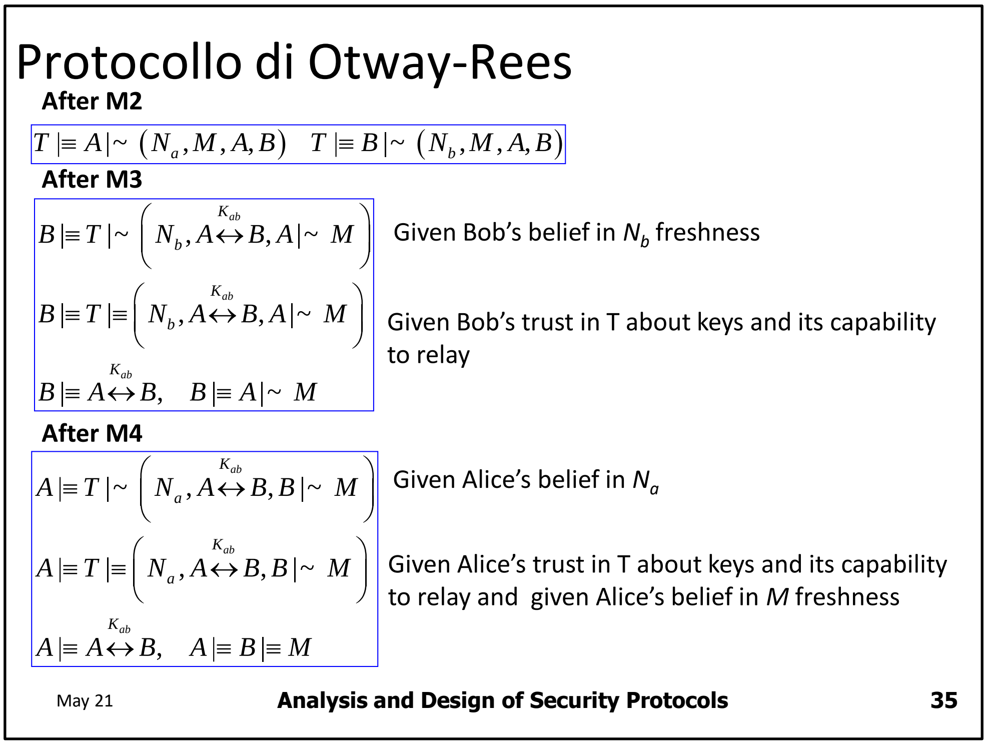
$B \mid\equiv A \overset{K_{ab}}{\leftrightarrow} B$

$A \mid\equiv B \mid\equiv M$

$B \mid\equiv A \mid\sim M$

May 21 $\quad$ **Analysis and Design of Security Protocols** $\quad$ **34**

ASSUMPTIONS
About keys and secrets. Trivial
About freshness. Trivial
About trust.
1) T is an authority on Kab: trivial
2) A and B trust T to correctly relay what the other peer said (not necessarily in the same instance of the protocol).



Protocollo di Otway-Rees

After M2

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

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

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

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

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

After M4

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

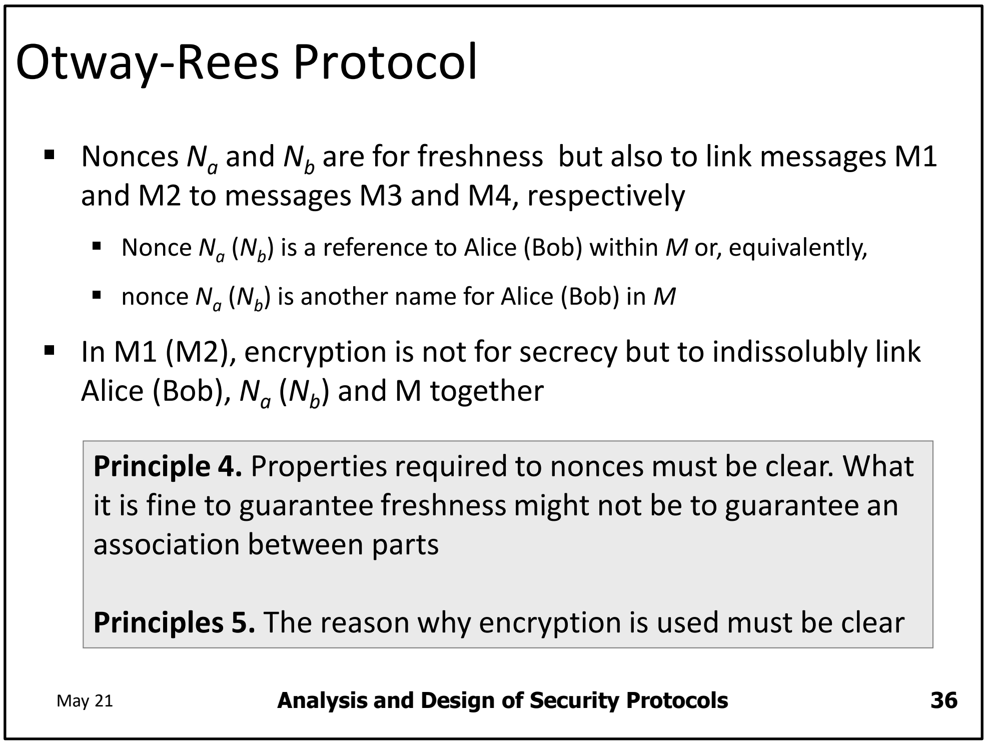
Given Alice’s belief in N_a

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

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

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

May 21 **Analysis and Design of Security Protocols** **35**



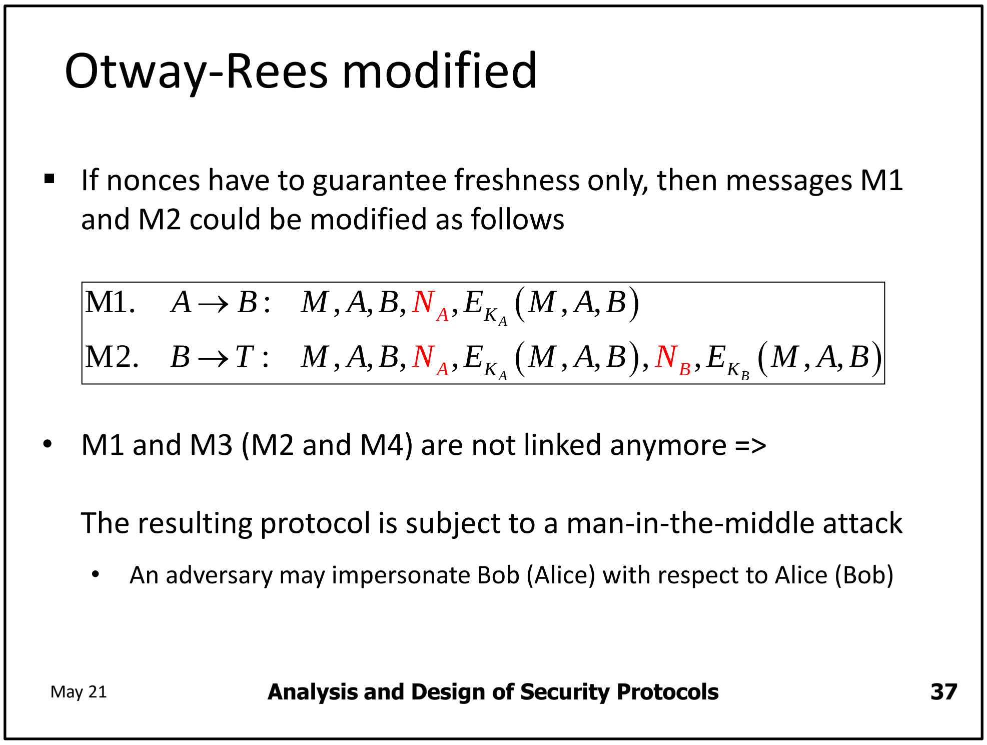
Otway-Rees Protocol

- 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

May 21 **Analysis and Design of Security Protocols** **36**



Otway-Rees modified

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

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

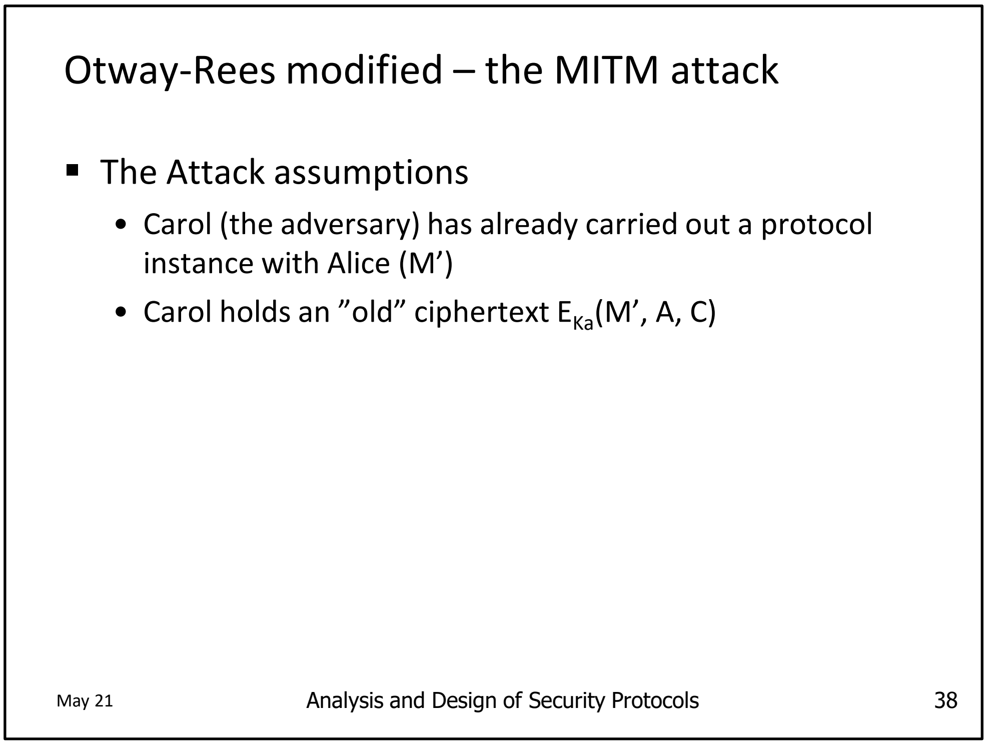
- M1 and M3 (M2 and M4) are not linked anymore =>

  The resulting protocol is subject to a man-in-the-middle attack

  - An adversary may impersonate Bob (Alice) with respect to Alice (Bob)

May 21 **Analysis and Design of Security Protocols** **37**

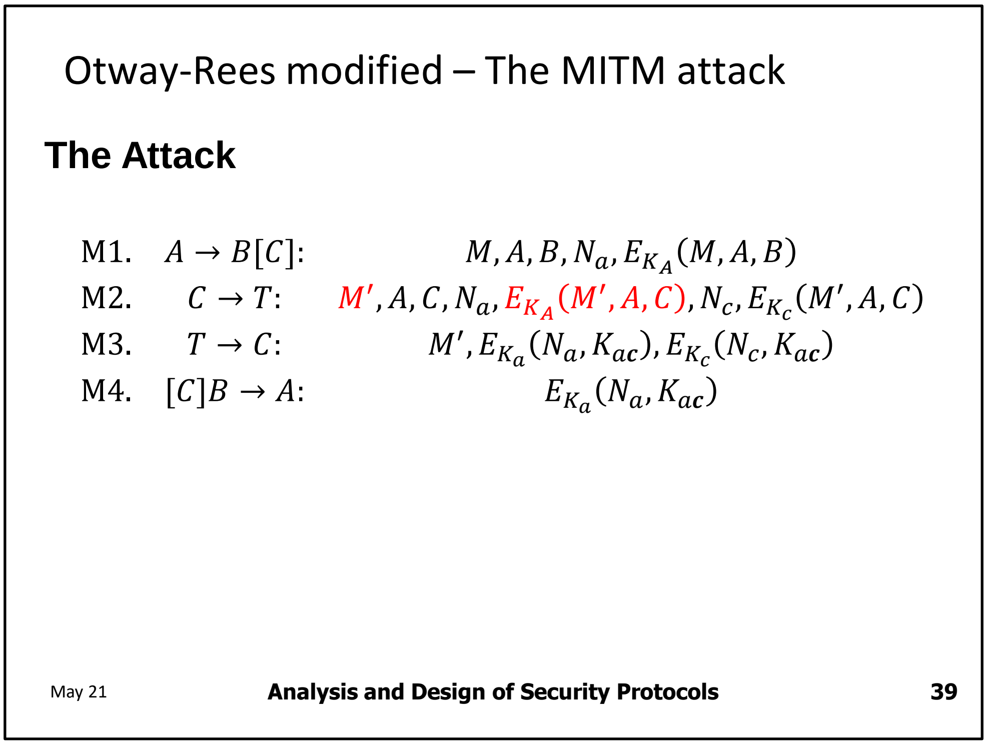
This attack requires that C performed an instance of the protocol with A in the past.
Notice that messages M3 and M4 are not linked to message M1 and M2 anymore.



Otway-Rees modified – the MITM attack

- The 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)

May 21 Analysis and Design of Security Protocols 38



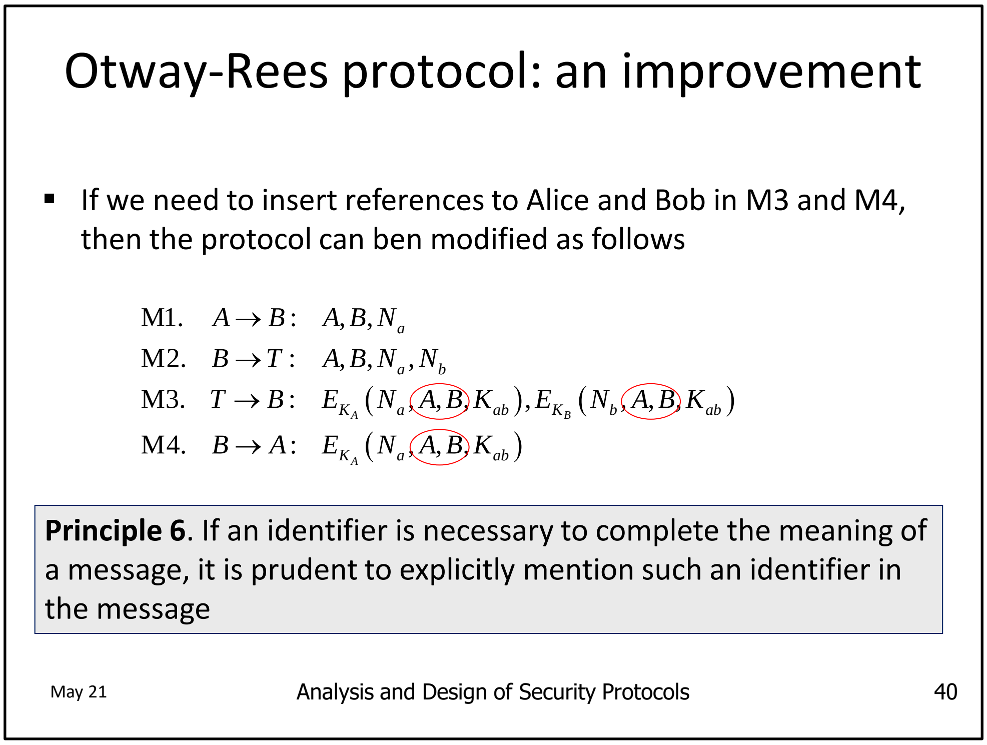
Otway-Rees modified – The MITM attack

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.	$[C]B \rightarrow A$:	$E_{K_a}(N_a, K_{ac})$

May 21 **Analysis and Design of Security Protocols** **39**

A sends M1 to B but C intercepts it. In M2, C replays the old cyphertext to T. T generates a session key K_{ac} for A and C and returns it to C in M3. C forwards A the portion of M3 encrypted by K_A pretending to be B. Upon receiving this message, A believes to be able to communicate with B by means of key K_{ac}. So doing, C is able to impersonate B w.r.t. to A anytime (s)he likes.



Otway-Rees protocol: an improvement

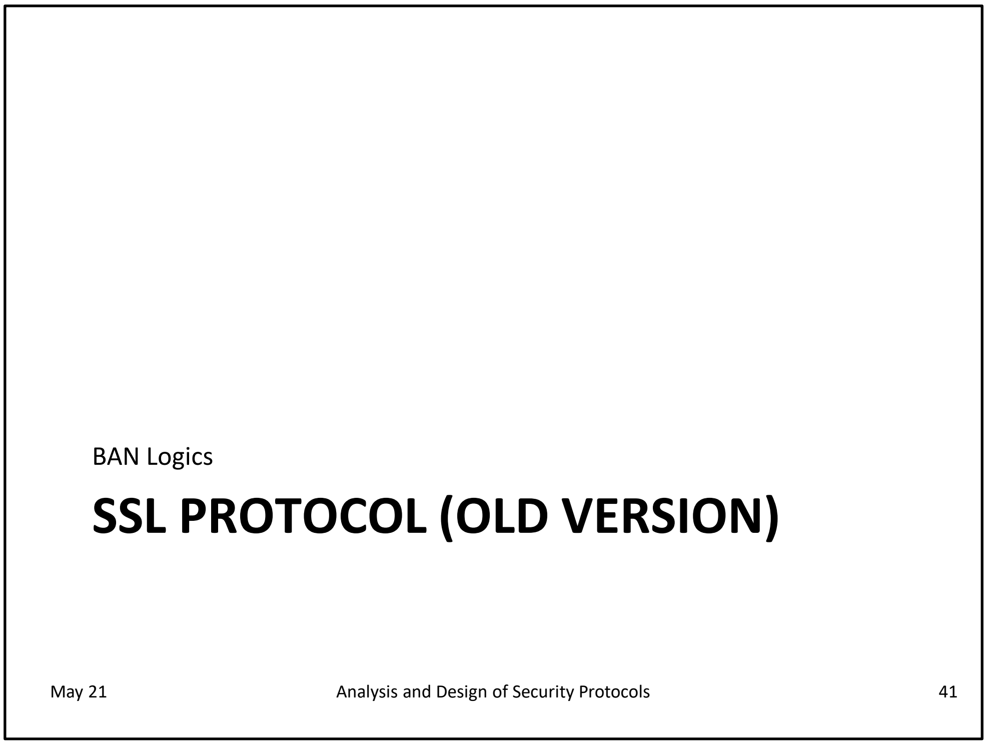
- If we need to insert references to Alice and Bob in M3 and M4, then the protocol can ben modified as follows

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

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

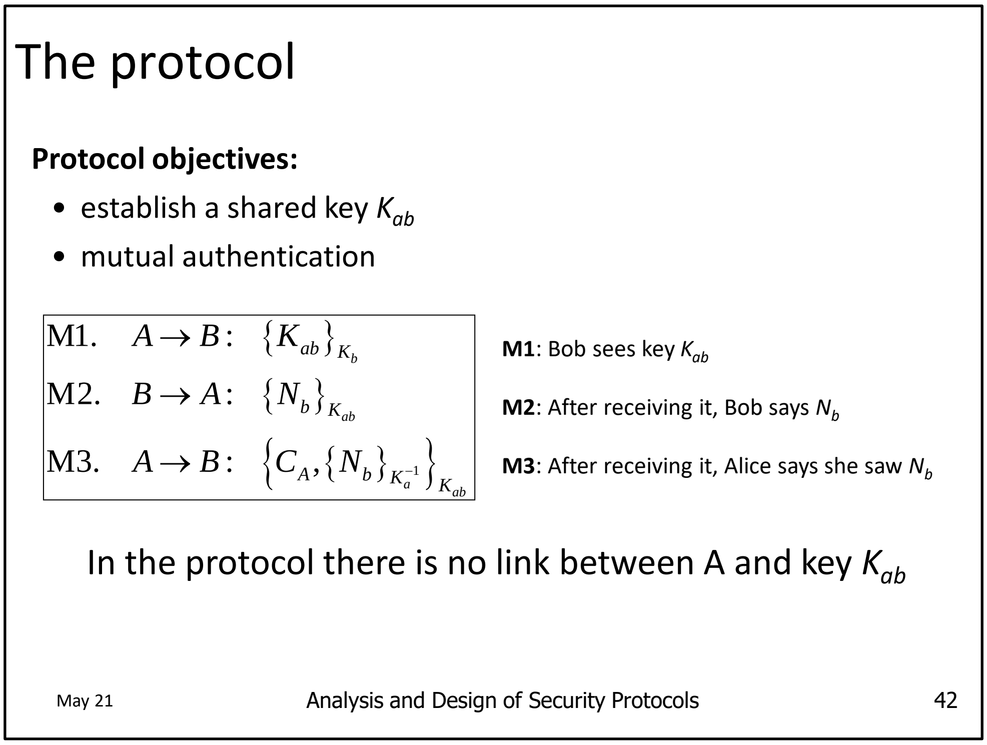
May 21 Analysis and Design of Security Protocols 40

In the new version of the protocol, the previous attack is not feasible anymore because T would put «A, C» in message M3 so now C cannot forward it A in M4. On the other hand, C cannot violate the integrity of the message.



BAN Logics

SSL PROTOCOL (OLD VERSION)



The protocol

Protocol objectives:

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

M1. $A \rightarrow B: \{K_{ab}\}_{K_b}$	**M1**: Bob sees key K_{ab}
M2. $B \rightarrow A: \{N_b\}_{K_{ab}}$	**M2**: After receiving it, Bob says N_b
M3. $A \rightarrow B: \{C_A, \{N_b\}_{K_a^{-1}}\}_{K_{ab}}$	**M3**: After receiving it, Alice says she saw N_b

In the protocol there is no link between A and key K_{ab}

May 21 Analysis and Design of Security Protocols 42

INFORMAL 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: i.e., Alice has Kab, decrypts M2, obtains Nb and thus signs Nb. 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. See the next attack.

<u>Analysis</u>

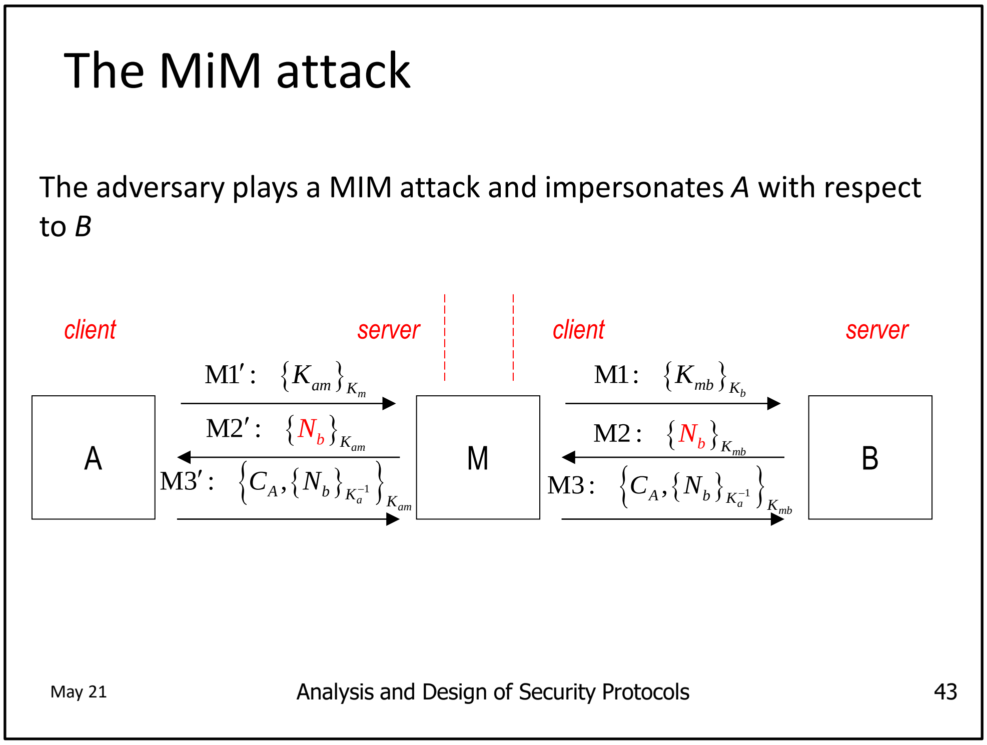
<u>Assumptions</u>

- Alice believes that Kab is shared with Bob
- Alice believes that Kab is fresh
- Alice believes that Kb is for communicating with Bob

<u>Idealized protocol</u>

- After M1, B sees Kab
- After M2, A sees Nb, A believes that
    1. B said Nb and K_{ab};
    2. B believes Nb because Kab is fresh for Alice;
    3. B believes Kab because Kab is fresh for Alice.
- After M3, B believes A believes Nb.

<u>However there is no proof for B to believe that Alice has seen Kab.</u>

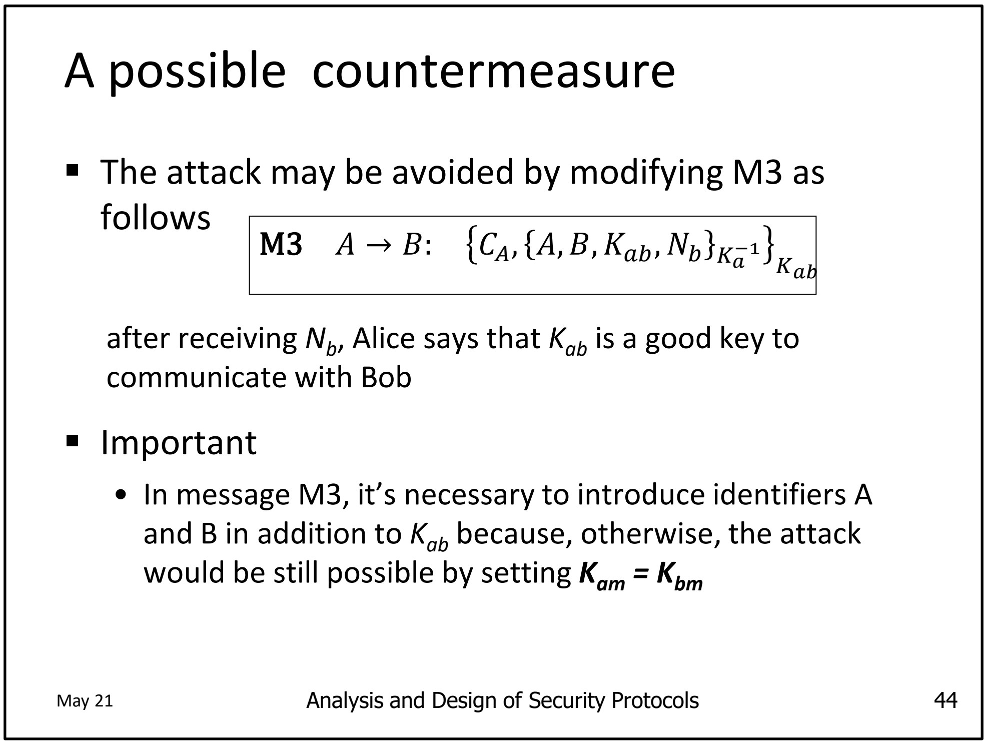


The MiM attack

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

May 21 Analysis and Design of Security Protocols 43

The adversary M starts an SSL session with Alice (M1’-M3’) and Bob (M1-M3), playing the role of server and client, respectively. The crucial step is message M2. M forwards A the same nonce N_b it received from B. Such a nonce is digitally signed by A (message M3’). M forwards B such a digital signature in M3. It follows that M impersonates A w.r.t. to B.



A possible countermeasure

- The attack may be avoided by modifying M3 as follows

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

  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 $\boldsymbol{K_{am} = K_{bm}}$

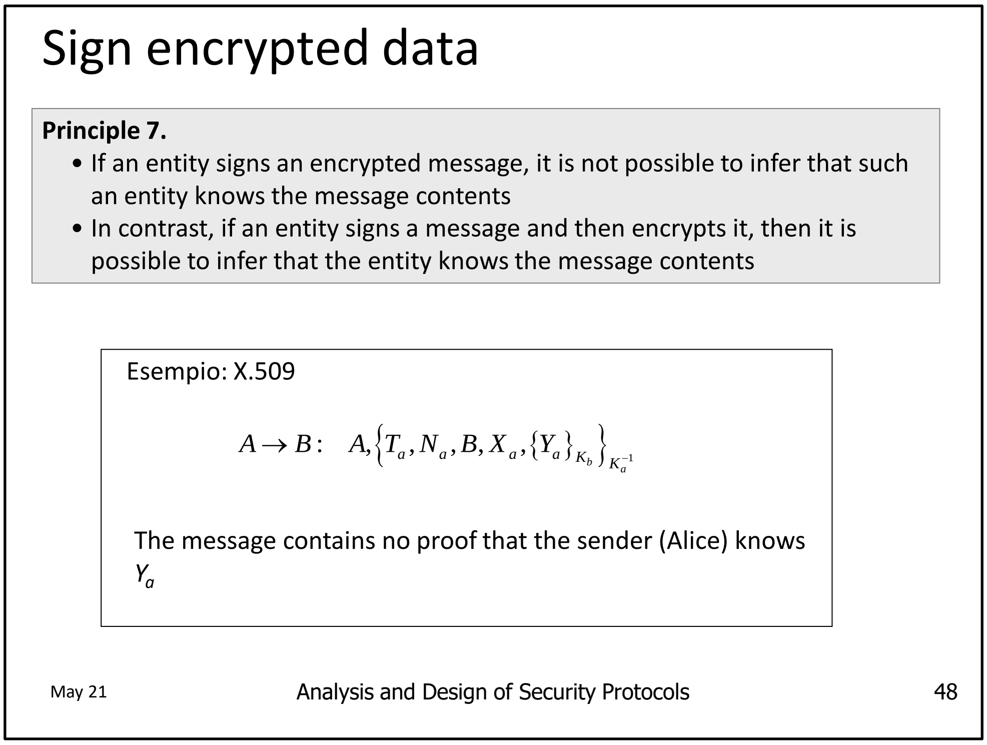
May 21 Analysis and Design of Security Protocols 44

By digitally signing Kab in M3, B has the proof that actually A saw Kab. However, this is not sufficient. Actually, the digital signature in M3 must also involve identifiers A and B as well. Kab only is not sufficient because an adversary could choose Kam = Kmb = K. By doing so, M3 = M3' = $\{C_A, \{K, Nb\}_{privKA}\}_{Kab}$.
The previous attack now fails because M3' becomes: M3' = $\{C_A, \{A, M, K, Nb\}_{privKA}\}_{Kab}$. Mallet cannot forge it.

BAN Logics

OTHER ISSUES

May 21 Analysis and Design of Security Protocols 47



Sign encrypted data

Principle 7.

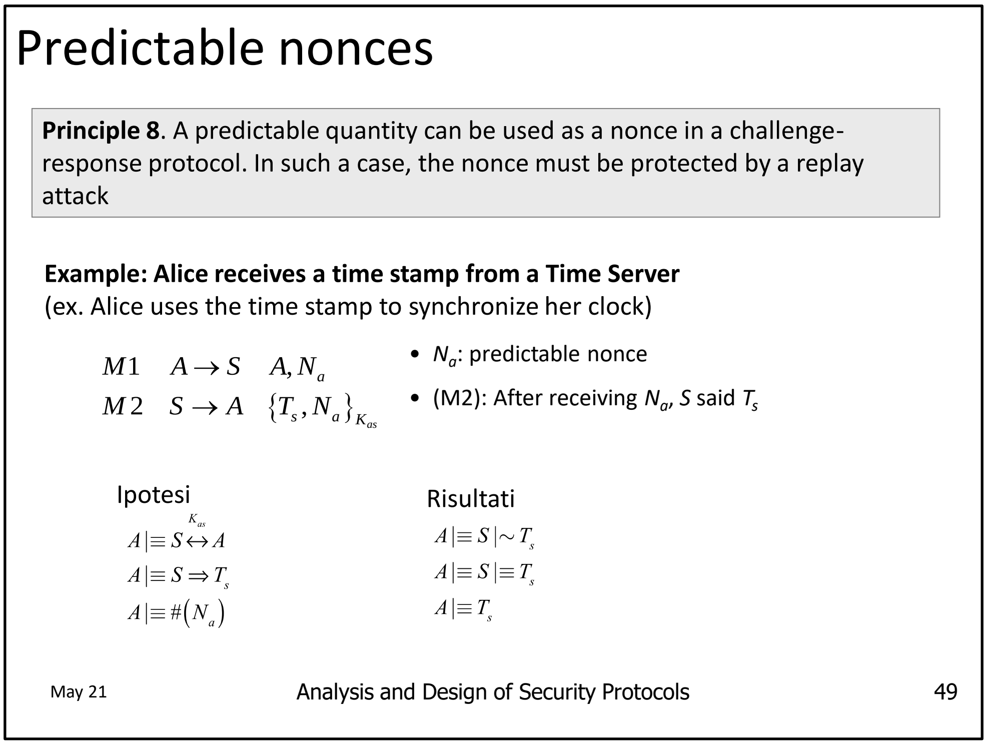
- If an entity signs an encrypted message, it is not possible to infer that such an entity knows the message contents
- In contrast, if an entity signs a message and then encrypts it, then it is possible to infer that the entity knows the message contents

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*

May 21 Analysis and Design of Security Protocols 48



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 time stamp from a Time Server
(ex. Alice uses the time stamp to synchronize her clock)

$$
\begin{array}{lll}
M1 & A \rightarrow S & A, N_a \\
M2 & S \rightarrow A & \{T_s, N_a\}_{K_{as}}
\end{array}
$$

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

Ipotesi

$A \mid\equiv S \overset{K_{as}}{\leftrightarrow} A$

$A \mid\equiv S \Rightarrow T_s$

$A \mid\equiv \#(N_a)$

Risultati

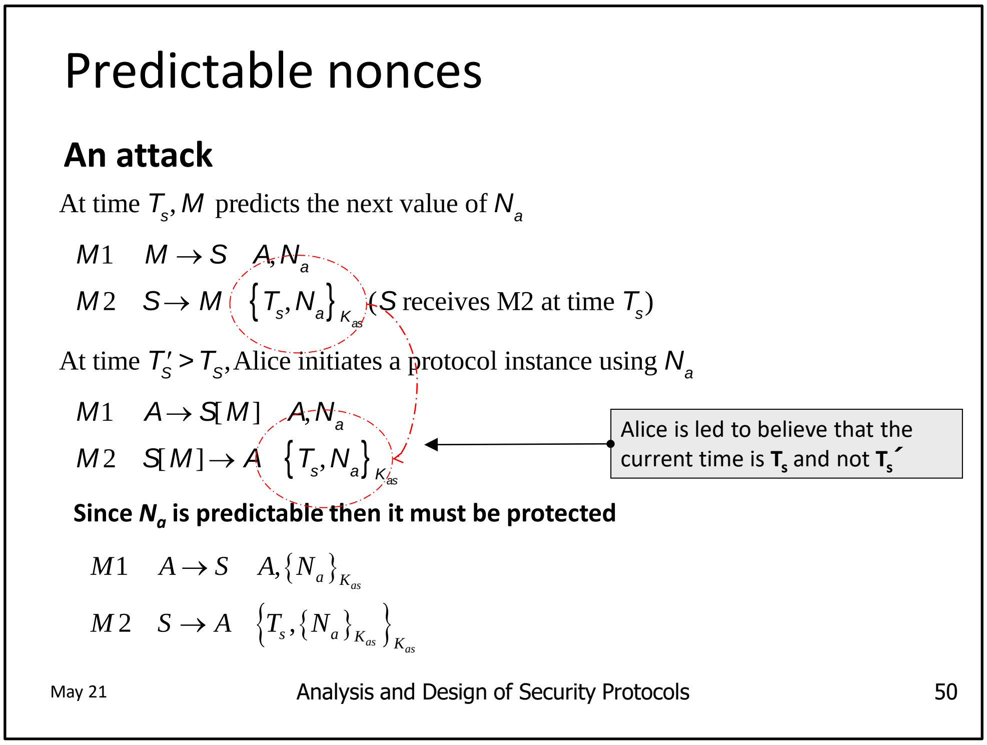
$A \mid\equiv S \mid\sim T_s$

$A \mid\equiv S \mid\equiv T_s$

$A \mid\equiv T_s$

May 21 Analysis and Design of Security Protocols 49

By definition, a nonce is a quantity that has never been used in a previous instance of the protocol. A nonce can be a timestamp, a counter or a random number. Timestamp and counters are predictable. Predictability may be exploited in replay attacks.
Consider the clock synchronization protocol. As Na is predictable, a replay attack aimed at turn back the clock is possible.



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´**

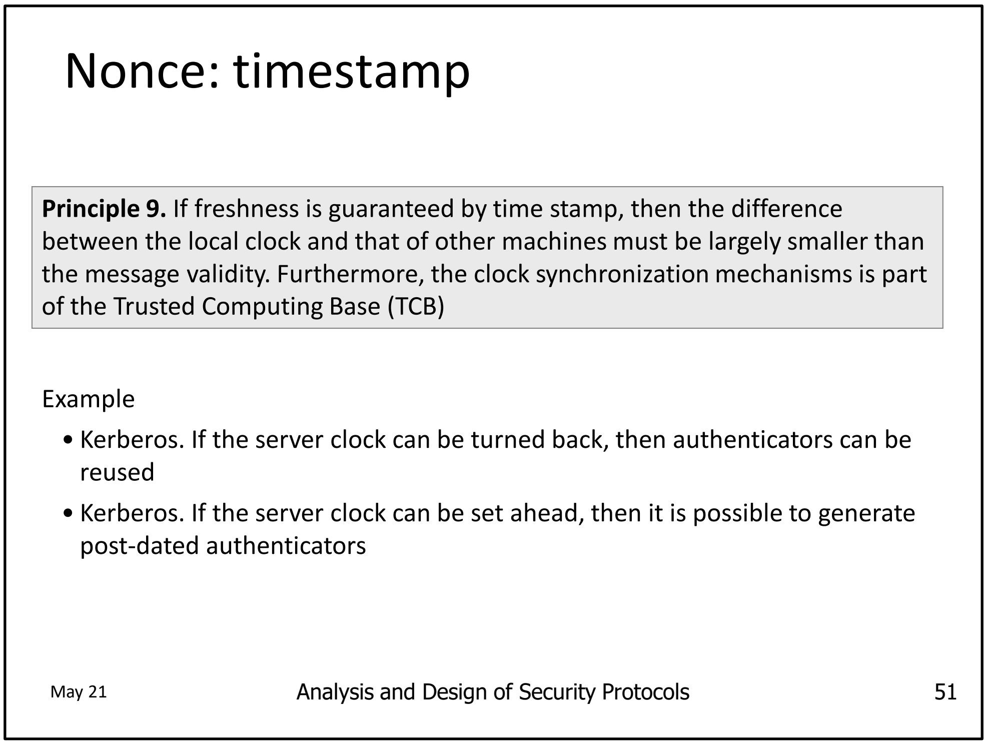
Since N_a is predictable then it must be protected

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

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

May 21 Analysis and Design of Security Protocols 50

At a given time T_s, Adversary M predicts a future value for Na and sends it to S pretending to be A. S returns M the Na timestamped by Ta.
Later, a Ts’ > Ts, A performs the protocol to synchronize its clock using Na. M replies A by sending the message M it received at time Ts. It follows that A turns back its own clock at Ts.
A possible countermeasure consists in encrypting Na. Such an encryption «randomizes» Na so making the resul unpredictable.



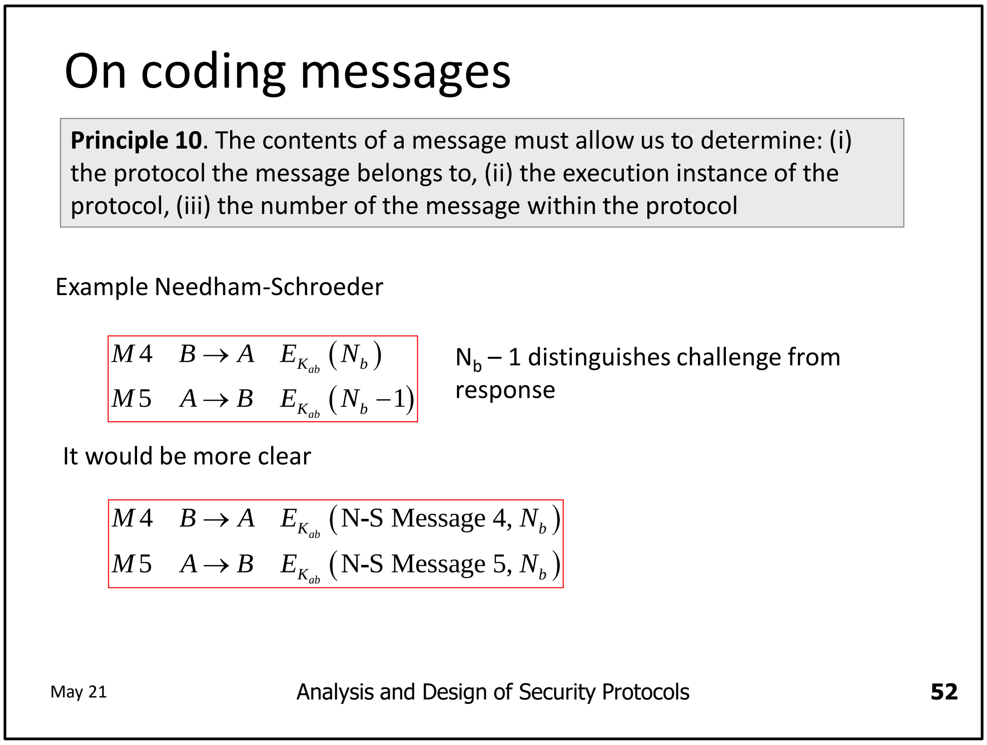
Nonce: timestamp

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)

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

May 21 Analysis and Design of Security Protocols 51



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

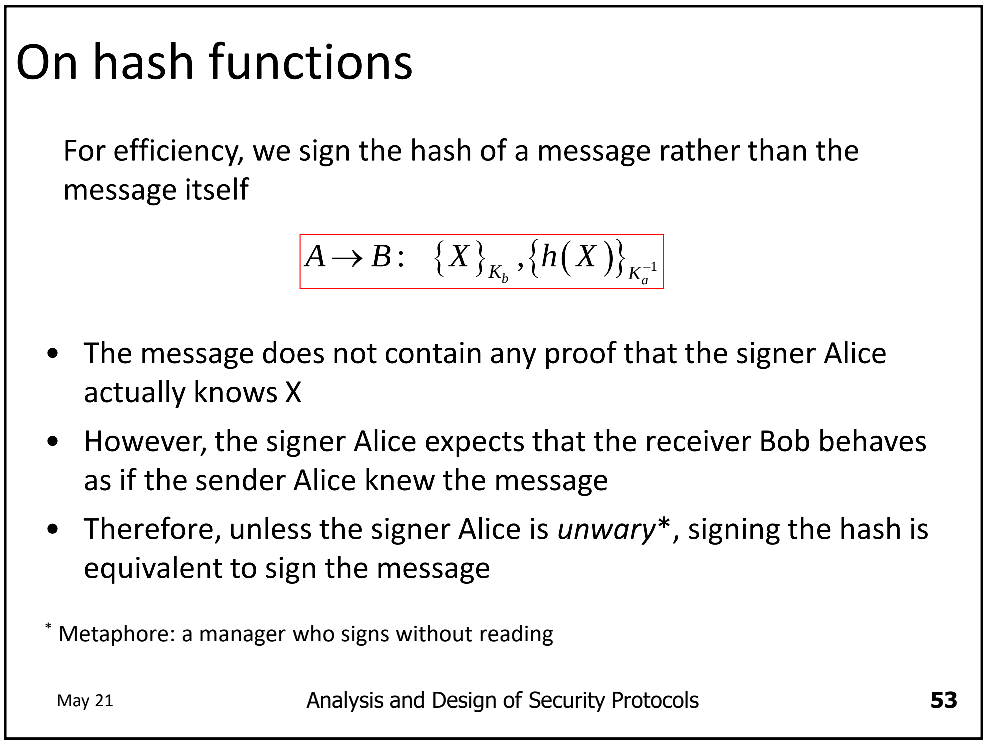
$$\boxed{\begin{array}{lll} M4 & B \rightarrow A & E_{K_{ab}}\left(N_b\right) \\ M5 & A \rightarrow B & E_{K_{ab}}\left(N_b - 1\right) \end{array}}$$

N_b – 1 distinguishes challenge from response

It would be more clear

$$\boxed{\begin{array}{lll} M4 & B \rightarrow A & E_{K_{ab}}\left(\text{N-S Message 4, } N_b\right) \\ M5 & A \rightarrow B & E_{K_{ab}}\left(\text{N-S Message 5, } N_b\right) \end{array}}$$

May 21 Analysis and Design of Security Protocols **52**



On hash functions

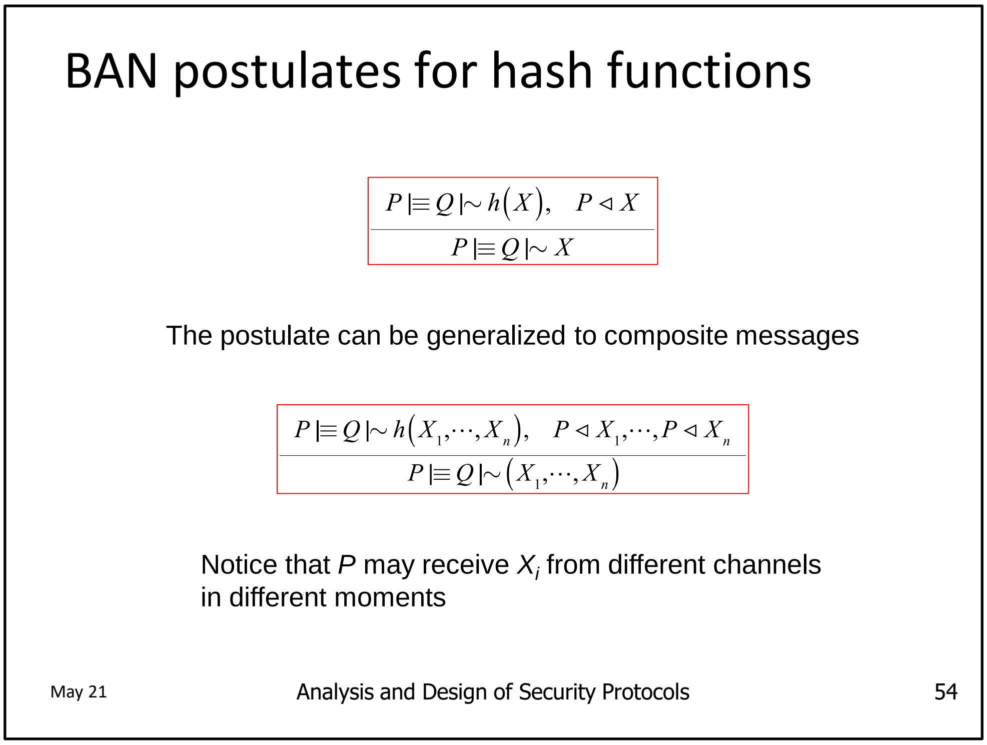
For efficiency, we sign the hash of a message rather than the message itself

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

- 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

May 21 Analysis and Design of Security Protocols **53**



BAN postulates for hash functions

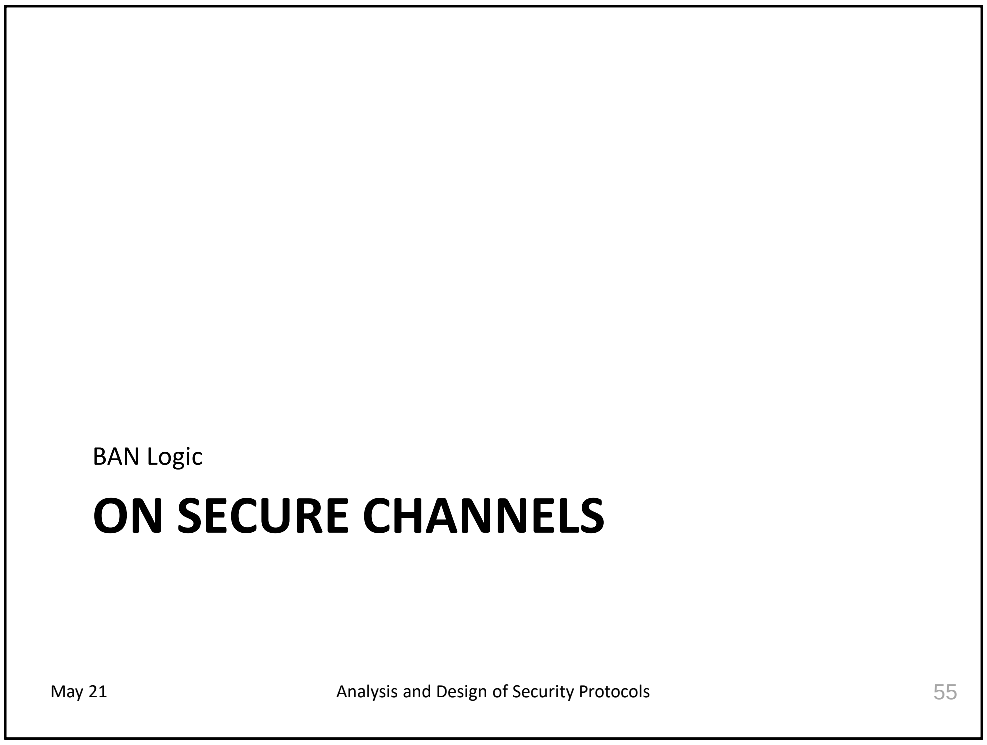
$$\frac{P \mid\equiv Q \mid\sim h(X), \quad P \triangleleft X}{P \mid\equiv Q \mid\sim X}$$

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

May 21 Analysis and Design of Security Protocols 54

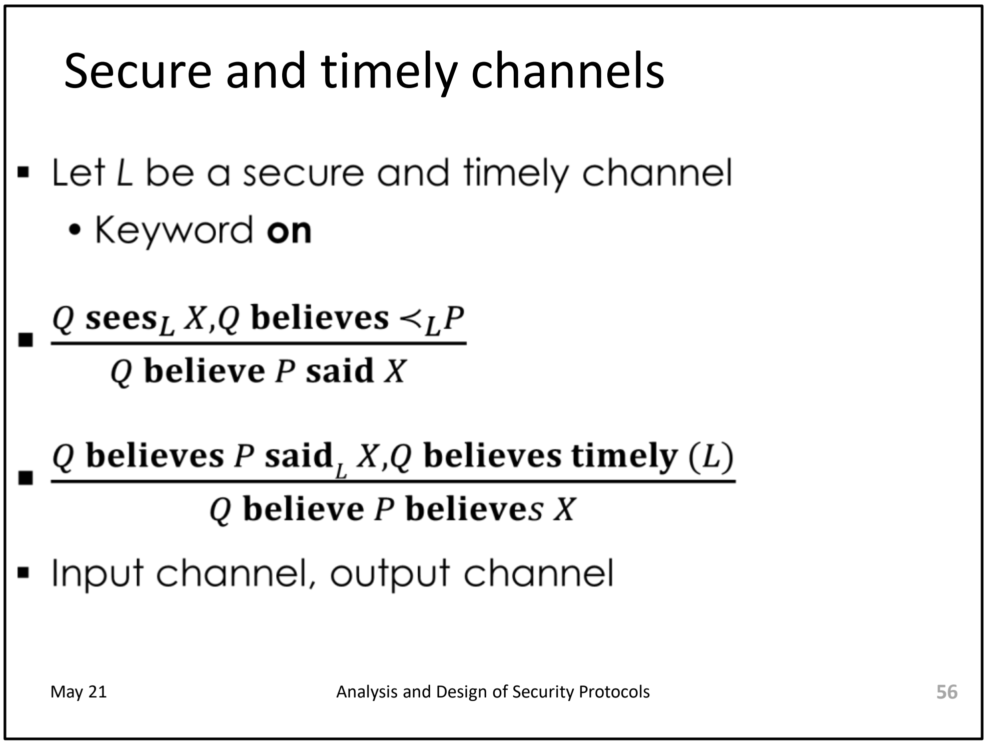


BAN Logic

ON SECURE CHANNELS

May 21 Analysis and Design of Security Protocols 55

Esempio di una smart card



Secure and timely channels

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

May 21 Analysis and Design of Security Protocols 56