DI-FCT-UNL
Computer and Network Systems Security
Segurança de Sistemas e Redes de Computadores
2011/2012

Needham-Schroeder: Secure Key-Distribution Protocol with Public Key Methods

Topics

- Security and the Key-Distribution problem
  - Needham Schroeder Model with asymmetric cryptography
- Context for TP2

- *Needham, Roger & Schroeder, Michael (December 1978), "Using encryption for authentication in large networks of computers.", Communications of the ACM* ***21****(12): 993-999*
- *Lowe, Gavin (November 1995), "An attack on the Needham-Schroeder public ~~key authentication protocol.", Information Processing Letters~~* ~~***56****(3): 131-136*~~
- [Wikipedia >>>]

Needham-Schroeder with asymmetric cryptography

Components of the protocol and assumptions
PKC: has a pair (KprivPKC, KpubPKC)
PKC: resgistration service for (A, KpubA), (B, KpubB)
A, has a key pair, KprivA, KpubA
B, has a key pair, KprivB, KpubB
Nonces: Na, Nb

Initial assumption: A and B have and trust the pair (PKC, KpubPKC) (obtained previously)

A > PKC :	**A, B**
PKC > A :	**{ KpubB, B }KprivPKC**
A > B :	**{ Na, A } KpubB**
B > PKC:	**B, A**
PKC > B:	**{ KpubA, A} KprivPKC**
B > A:	**{Na+1, Nb}KpubA**
A > B:	**{ Nb+1 }KpubB**

Needham-Schroeder with asymmetric cryptography

Components of the protocol and assumptions
PKC: has a pair (KprivPKC, KpubPKC)
PKC: resgistration service for (A, KpubA), (B, KpubB)
A, has a key pair, KprivA, KpubA
B, has a key pair, KprivB, KpubB
Nonces: Na, Nb

Initial assumption: A and B have and trust the pair (PKC, KpubPKC) (obtained previously)

A > PKC :	**A, B**
PKC > A :	**{ KpubB, B }KprivPKC**
A > B :	**{ Na, A } KpubB**
B > PKC:	**B, A**
PKC > B:	**{ KpubA, A} KprivPKC**
B > A:	**{Na+1, Nb, Ks}KpubA**
A > B:	**{ Nb+1 }Ks**

Ks generation

Ataque MIM (Gavin Lowe, 95)

Protocol with M acting as a MIM
M: has a key pair (KprivM, KPubM)
M obtained { A, KpubM } KprivPKC
M wants to masquerade B when M speaks with A
(Authentiction attack)

A > M :	**A, B**	**M > PKC :**	**A, B**
		PKC > M :	**{ KpubB, B }KprivPKC**
M > A :	**{ KpubM, B }KprivPKC**		
A > M :	**{ Na, A } KpubM**	**M > B :**	**{ Na, A } KpubB**
B > M:	**B, A**	**M > PKC:**	**B, A**
		PKC > M:	**{ KpubA, A } KprivPKC**
M > B:	**{ KpubA, A } KprivPKC**		
B > M	**{ Na+1, Nb,Ks } KpubA**	**M > A:**	**{Na+1, Nb, Ks}KpubA**
		A > M:	**{Nb+1}Ks**
M > B:	**{Nb+1}Ks**		

Protocolo de Needham-Schroeder (fixed)

A > PKC :	A, B
PKC > A :	{ KpubB, B }KprivPKC
A > B :	{ Na, A } KpubB
B > PKC:	B, A
PKC > B:	{ KpubA, A} KprivPKC
B > A:	{Na+1, Nb, B} KpubA
A > B:	{ Nb+1 }KpubB

Needham-Schroeder with asymmetric cryptography

1. A -> PKC : **A,B**

2. PKC -> A : **{ KBpub, B } KPKCpriv**

3. A -> B: **{ Na, A }KBpub**

---

4. B obtains (in a trust way) the KpubA from the PKC

---

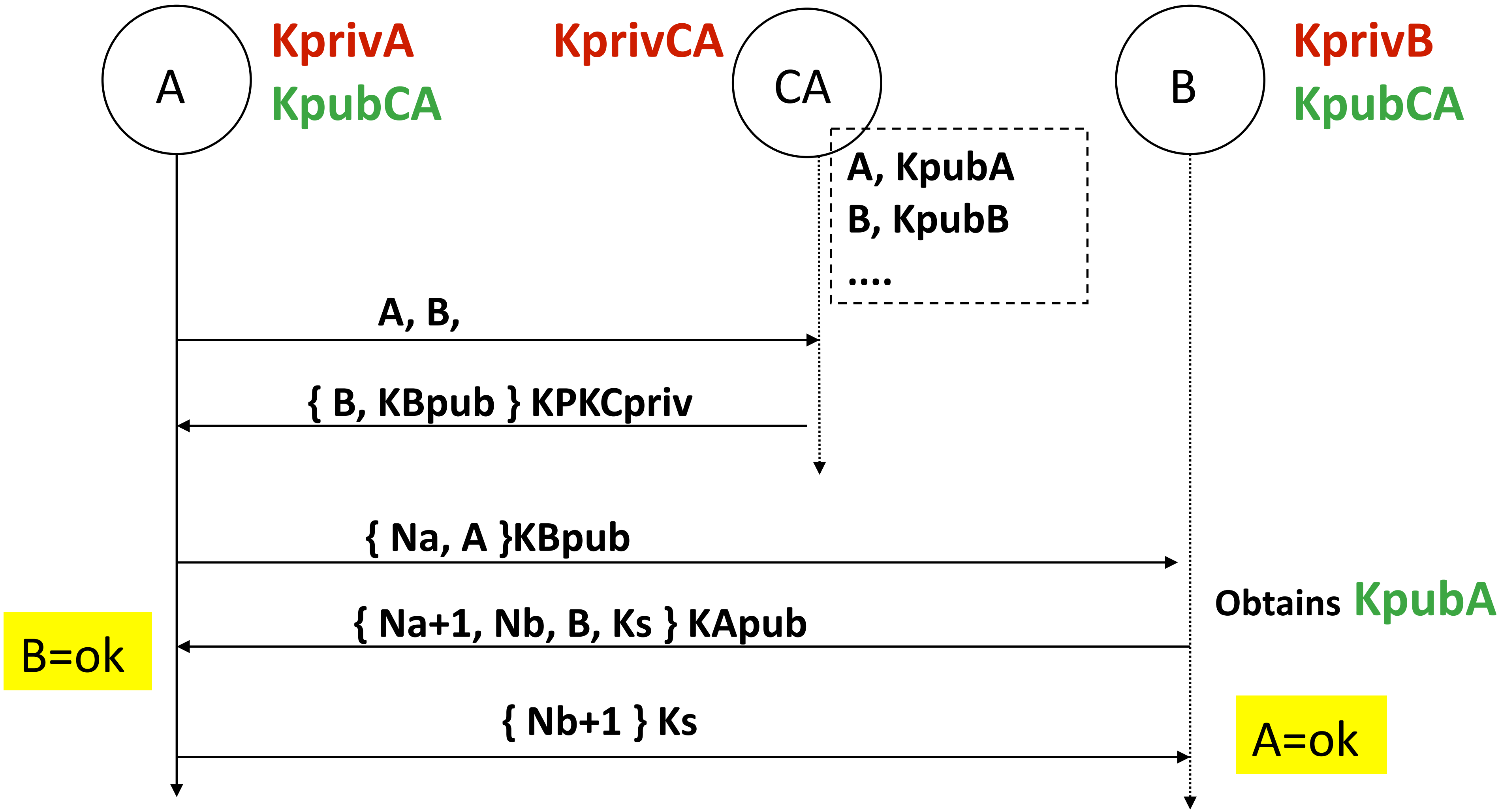
5. B -> A: **{ Na+1, Nb, B, Ks } KApub**

In this case, B generates the session key

6. A -> B: **{ Nb+1 } Ks**

A replies with the response to the nonce Nb, which authenticates A. Question: is B authenticated from the A viewpoint ?

Representation: timing diagram



Confidentiality (with symmetric crypto, session key Ks)

Optimization

A pode ter "a priori" {A, KApub} KPKCpriv obtido de KDC
Tanto A como B conhecem a priori KPKCpub

---

1. A -> PKC : A,B

2. PKC -> A : { B, KBpub } KPKCpriv

O PKC envia a chave pública de B cifrada com a sua chave secreta; A obtem a chave de B pois conhece a chave pública do PKC

3. A -> B: { Na, A }KBpub , { A, KApub } KPKCpriv

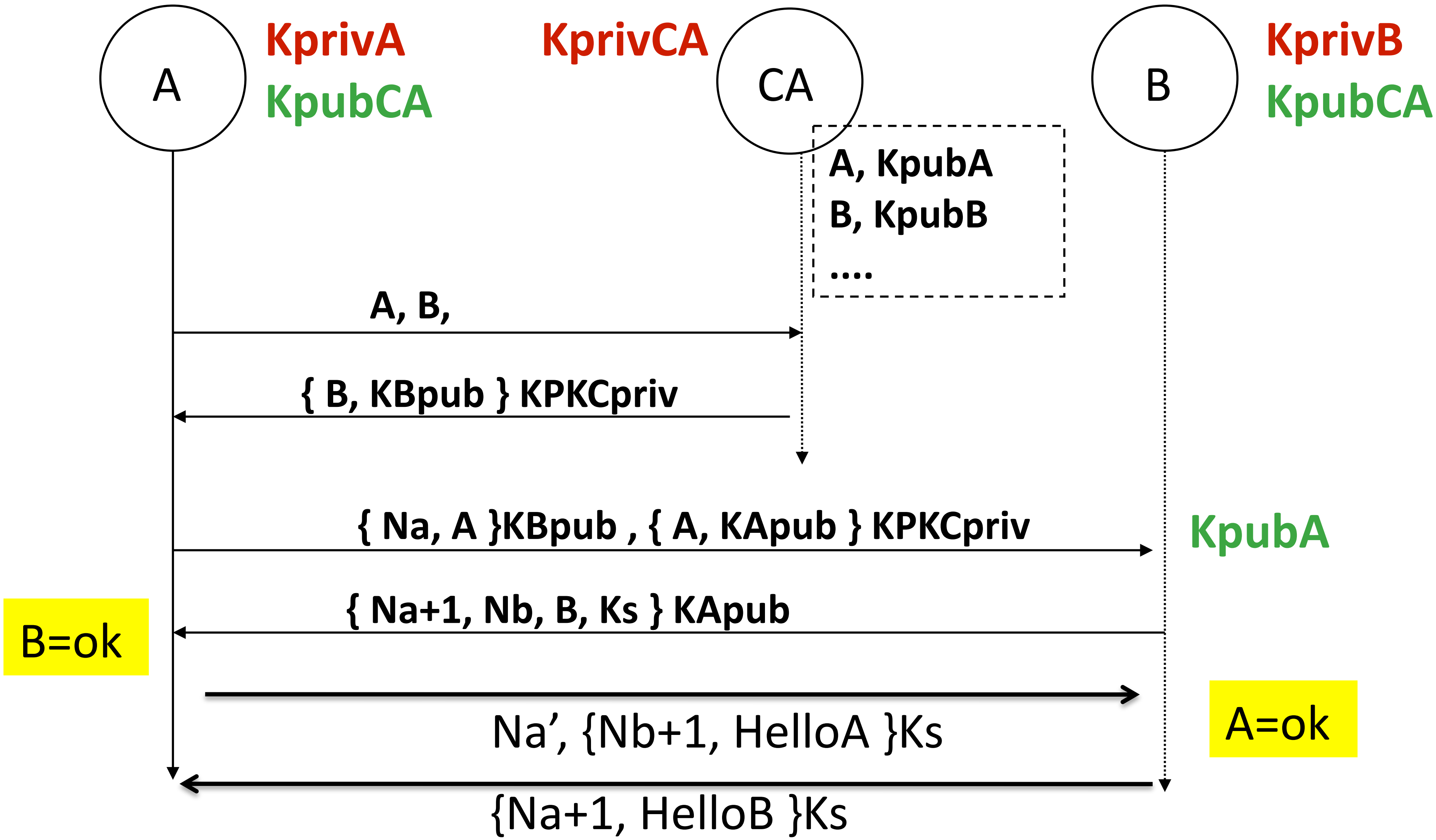
4. B -> A: { Na+1, Nb, B, Ks } KApub

B devolve a A o Na cifrado com a sua chave pública que corresponde a responder ao desafio enviado por A, envia-lhe também um desafio e uma chave secreta de sessão, tudo cifrado com a chave pública de A

5. A -> B: { Nb+1 } Ks

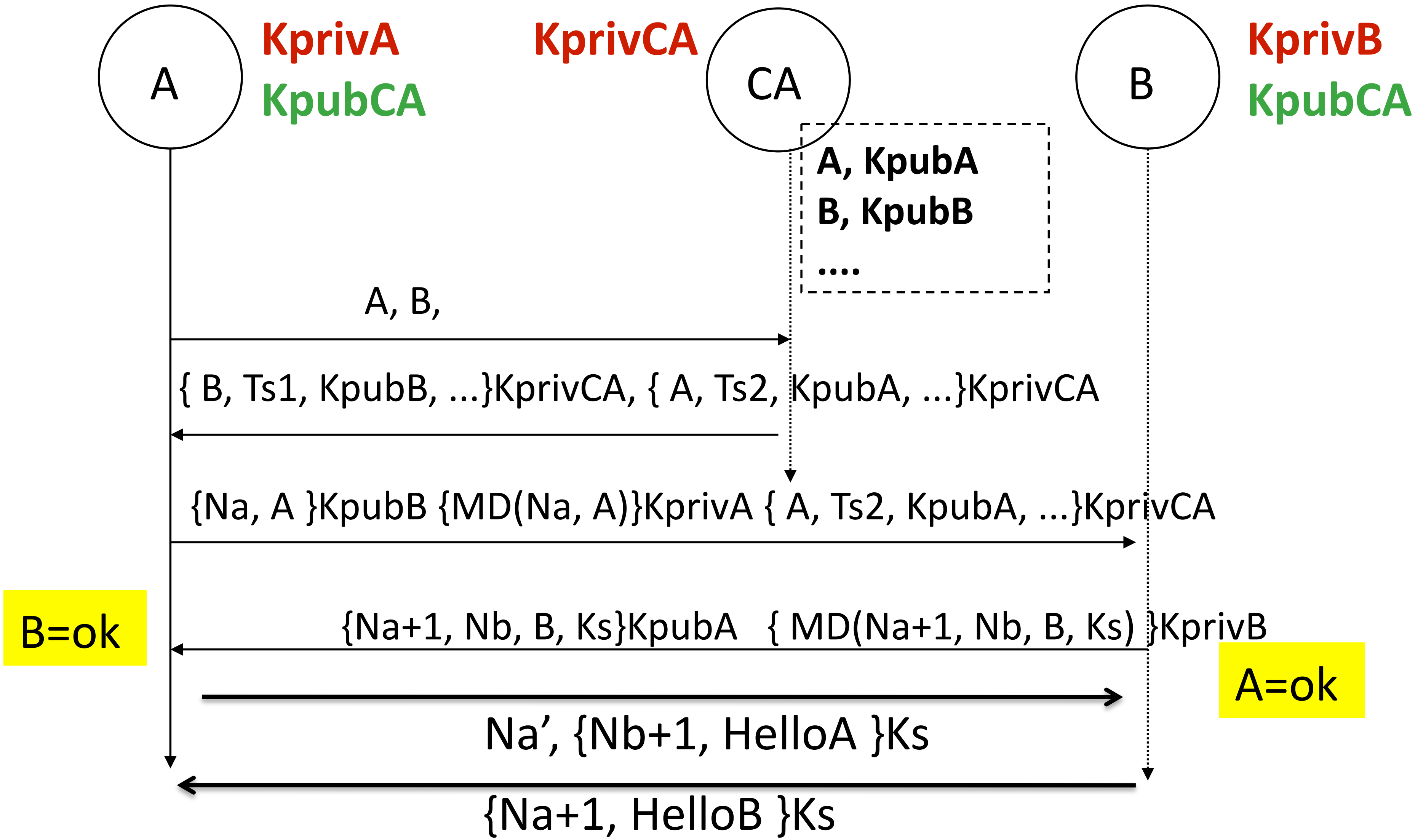
A responde ao desafio de B utilizando já a chave de sessão.

Optimization in the tioming diagram



Confidentiality with symmetric crypto and session key Ks

Authentication warranties



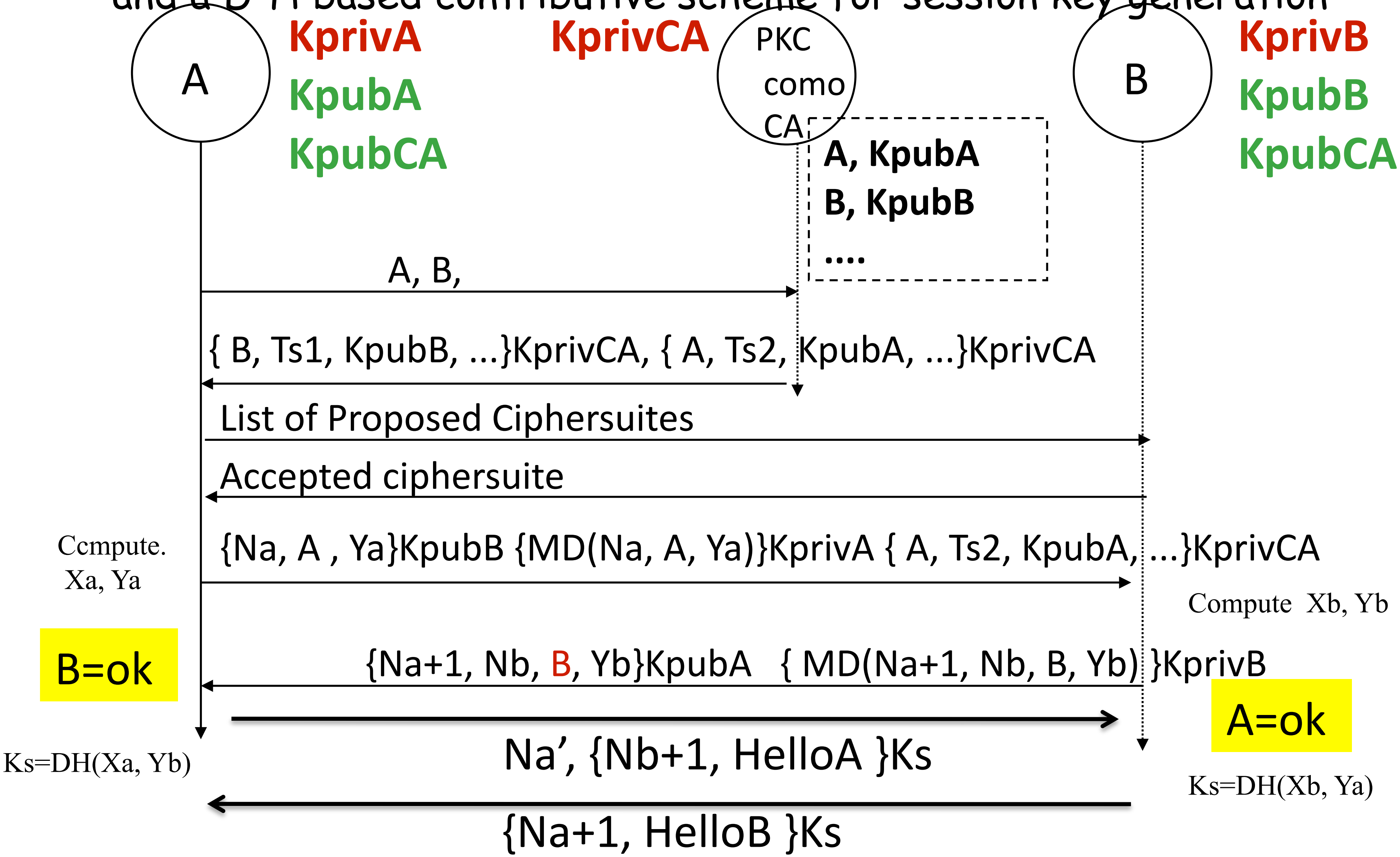
Confidentiality with symmetric crypto and session key Ks

Aspectos a ter em conta (contexto TP2)

- Implementação e completude do protocolo:
  - Utilização de assinaturas de chave pública de acordo com as boas práticas e métodos normalziados
    - Assinaturas cobrem sínteses dos conteúdos de mensagens a assinar, bem como utilziação adequada de métodos de padding
    - As assinaturas podem usar de forma flexíveis (parametrizações de ciphersuites)
    - Analisar e compreender exemplos e exercícios sugeridos utilizando assinaturas digitais de chave pública e posíveis variantes (RSA, ElGammal, DSA e /ou ECC-DSA, etc)
    - Implementar sempre garantias de non-replaying ou message-freshness para protecção adicional

N-S with public key and Diffie Hellman exchange

Variant: N-S- cripto assimétrica, ciphersuites negotiation
and a D-H based contributive scheme for session key generation



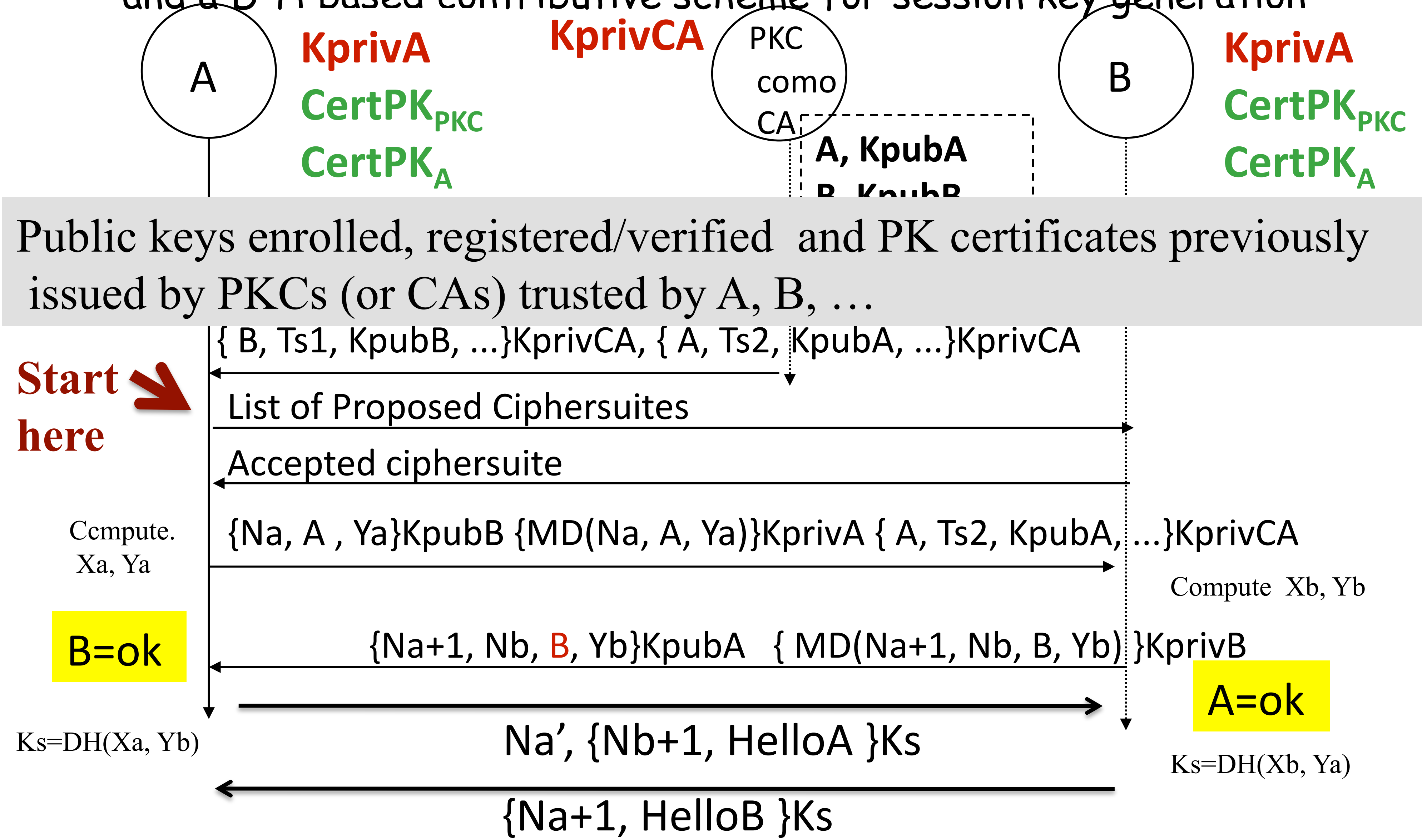
Aspectos a ter em conta

- Implementação prática:
  - Parâmetros públicos de D-H: garantia de confidencialidade ? Justifica-se ?
    - Problemas de eficiência e custo computacional
      - » ((M^N1 mod X) ^ N2 mod Y), ou de complexidade equivalente a M^(N1*N2)
  - Mas autenticação dos parâmetros protegidos nas assinaturas digitais é um aspecto ESSENCIAL
    - Recordar problema de acordos anónimos D-H e ataque à autenticação com homem-ao-meio

- Outro aspecto:
  - Usando certificados de Chave Pública (previamente obtidos de CAs ou PKIs actuando como TCBs), o protocolo não precisa de ser feito desde o início

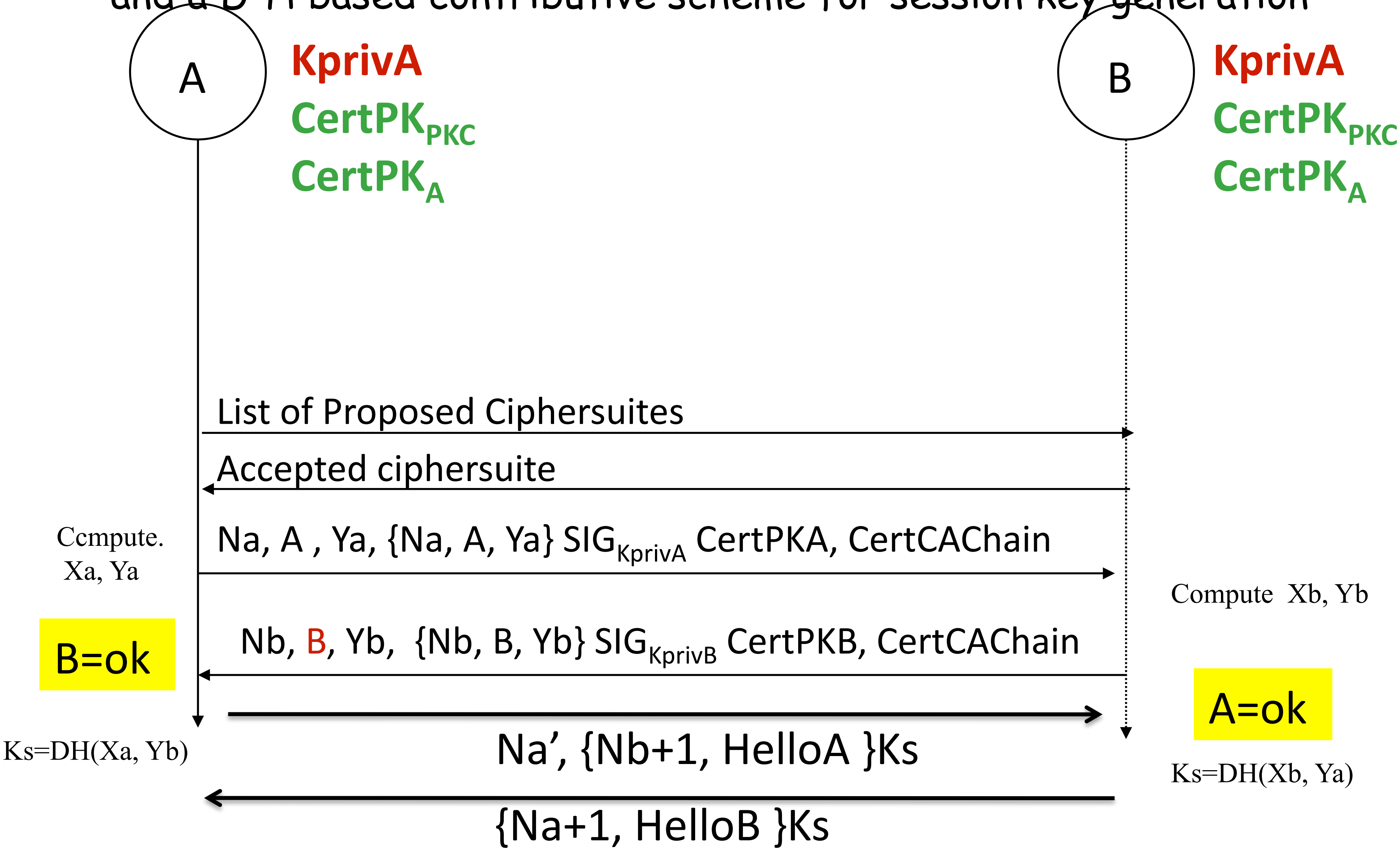
N-S with public key and Diffie Hellman exchange

Variant: N-S- cripto assimétrica, ciphersuites negotiation and a D-H based contributive scheme for session key generation



N-S with public key and Diffie Hellman exchange

Variant: N-S- cripto assimétrica, ciphersuites negotiation
and a D-H based contributive scheme for session key generation



Topics

- Security and the Key-Distribution problem
  - Needham Schroeder Model with asymmetric cryptography
- Context for TP2

- *Needham, Roger & Schroeder, Michael (December 1978), "Using encryption for authentication in large networks of computers.", Communications of the ACM* ***21****(12): 993-999*
- *Lowe, Gavin (November 1995), "An attack on the Needham-Schroeder public* ~~*key authentication protocol.", Information Processing Letters* ***56****(3): 131-136*~~
- [Wikipedia >>>]

Context for TP2
(Start to think !)

Context for TP2 (1)

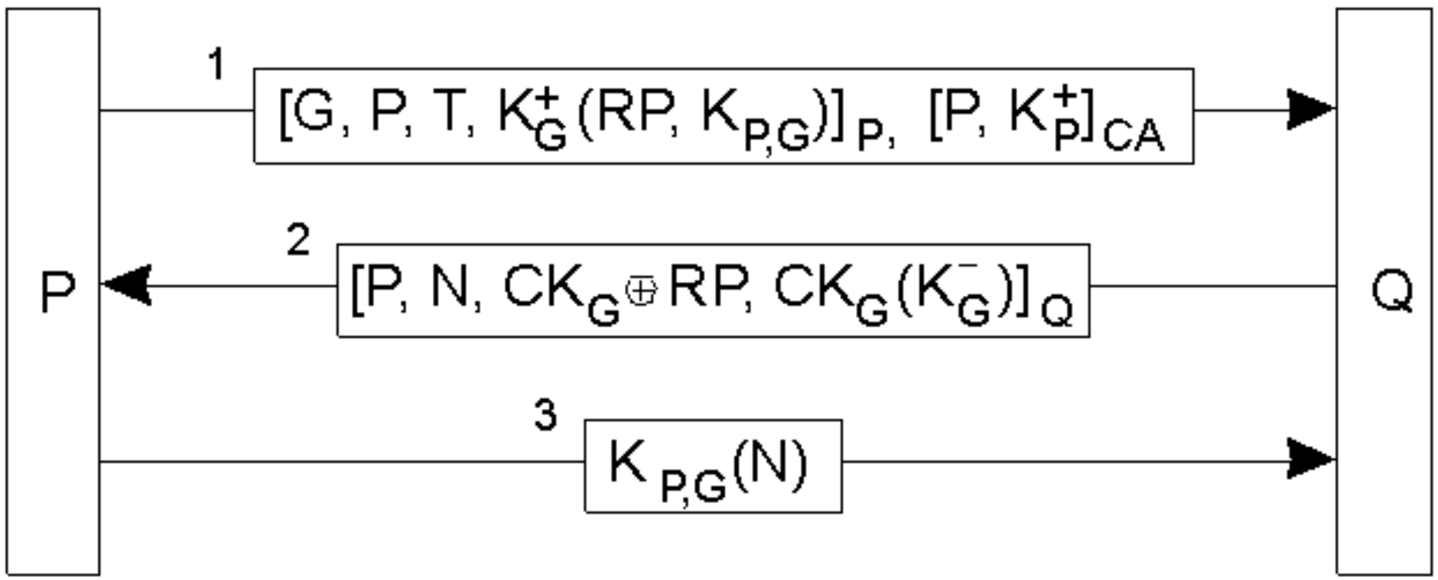
- You must design an authentication and GROUP-CHAT Key -Distribution protocol, using now a PKC and Asymmetric Cryptography Methods
- Needham-Schroeder Variant with Public-Key Cryptography combined with a Diffie-Hellman Agreement for Key Chat-Session generation
  - Inspired by the previous protocol
- Authentication of participants using digital signatures using asymmetric methods (RSA, DSA or ECC) with public keys securely registered and distributed by a PKC
  - In your new solution compared with TP1, you will change the KDC TBC entity by a PKC TCB entity
- Session keys generated by a Diffie-Hellman Key Agreement

Context for TP2 (2)

- Implicit PFS and PBS
- What about contributive key-session generation ?
  - Implicit to the D-H agreement ? In which conditions ?
  - Is it extensible to a group contributive scheme ?
- Issue: it iis expectable that it can be slow !!
  - Practical/experimental evaluation: how does it cost to join the CHAT ?
    - Time vs. Complexity Issues vs. Massive Joins /Leaves and Group Dimension)
    - Evaluation of "Time to Join" and "Key -Establishment" metrics
    - Churning effect / Group Membership Changes

Context for TP2 (3)

- Comparison with a Group-Admission Protocol, as stated by Reiter, 1996
- Principle:
  - K+ means a Group Public Key
  - K- means a Group Private Key



New Group Session Key Generation

Context for TP2 (4)

- Critical analysis of the Reiter Protocol, to be applied to the context of TP2, in a comparative discussion with the N-S and D-H combined scheme.
  - Performance, complexity, scalability, fast -rekeying and security warranties:
    - Authentication, Confidentiality, Integrity, Non-Replaying and better conditions for DoS
  - Group-Certification Management (secure management of KG+, KG-): how to warrant PFS and PBS ?
  - How to warrant a key-session contributive generation scheme ?

Context for TP2 (5): PKINIT Kerberos based solution

- Variant: PKINIT Kerberos
- Implement N-S: imagine PKC is the AS
- Participant and AS mutual authenticated.
  - Good ! No Password-Vulnerability
- Session Key generated as a TGT in a Kerberos Ticket from AS (signed by AS)
- Participant will send the TGT to the TGS Server (*)
- Who is the TGS Server ?
  - Your current KDC
  - The rest is the same
- What about Rekeying ?
  - Start from (*) is possible for fast rekeying. Think how it can be done, based on the KERBEROS assumptions

Context for TP2 (6): using SSL and JSSE

All this can be implemented (flexible parameterized)
In SSL (TLS) with Mutual Authenticated Sessions (SSL Handshake) with SSL Sockets (JSSE Support)
… Also possible with Secure RMI/SSL Sockets

- Session Key generated as a TGT in a Kerberos Ticket from AS (AS as a SSL Server)
- Participant will send the TGT to the TGS Server (*)

- Who is the TGS Server ?
  - Your current KDC
  - The rest is the same

- What about Rekeying ?
  - Start from (*) is possible for fast rekeying. Think how it can be done, based on the KERBEROS assumptions