An Encapsulated Authentication Logic for Reasoning about Key Distribution Protocols

Catherine Meadows
NRL

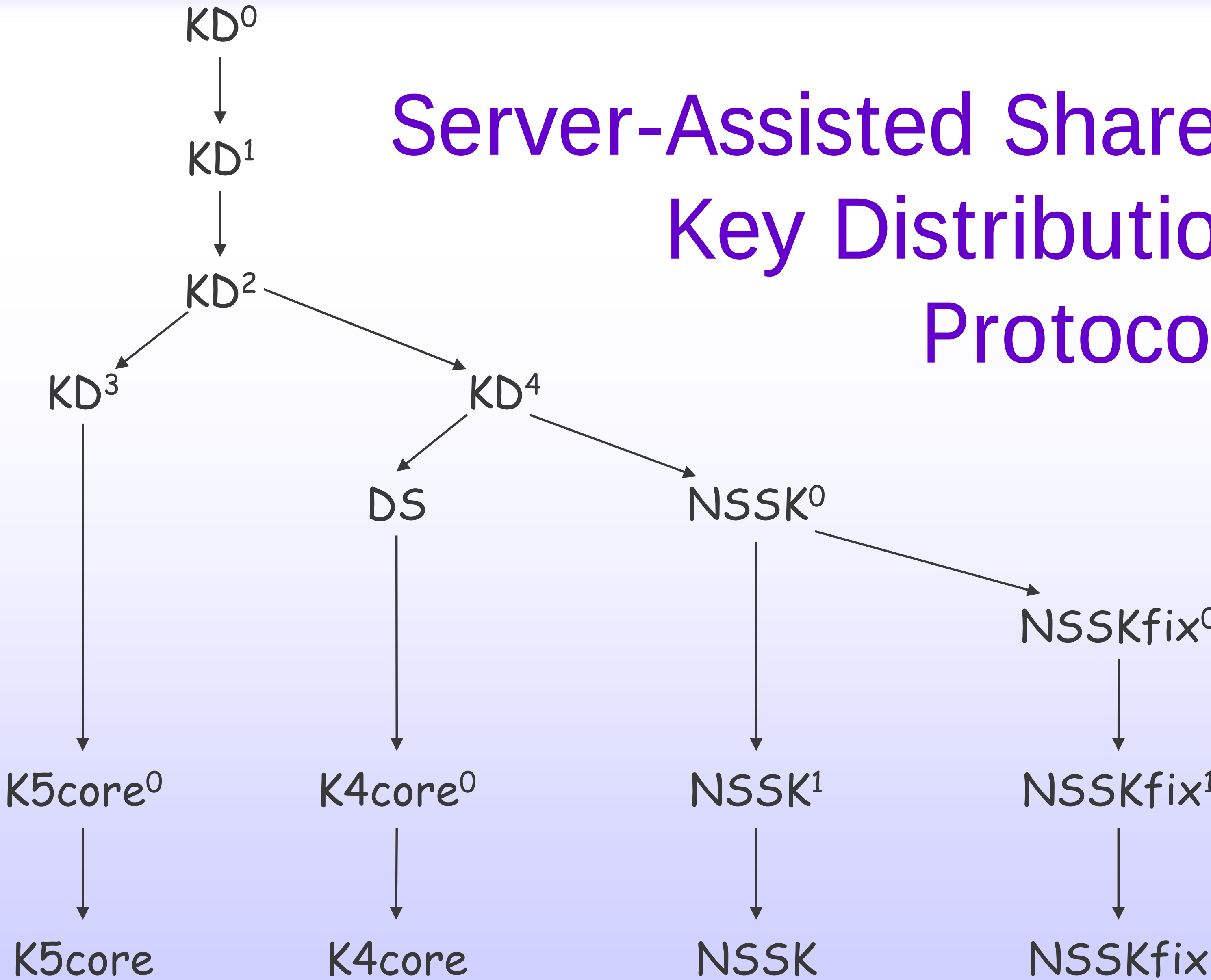
Dusko Pavlovic
Kestrel Institute

Iliano Cervesato
Tulane University

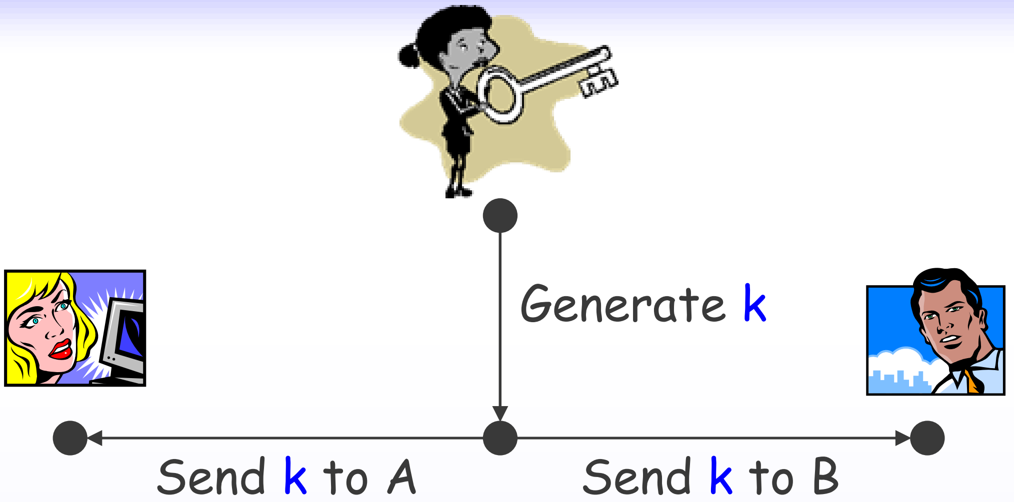
Contributions

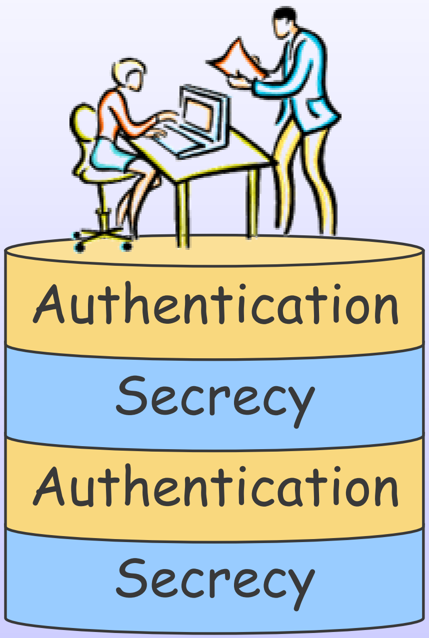
- Separate
  - Authentication reasoning
  - Secrecy reasoning
- Define a logic of pure authentication
  - Secrecy as assumptions
    - Proof obligations
- Embed it in derivational framework
- Apply to key distribution protocols
  - Taxonomy
  - Comparative study
  - Clear understanding of underlying mechanisms

Server-Assisted Shared Key Distribution Protocols



Key Distribution Protocols





- Secrecy depends on authentication
  - k secret only if sent over authenticated channels
- Authentication depends on secrecy
  - Cryptographic authentication relies on secrecy of long-term keys

Verifying KD Protocols

Historically single monolithic proofs

… BUT …

secrecy and authentication rely on very different proof methods

- **Authentication**
  - Completing partial order of actions
    - Get piping right
  - Local reasoning
  - Positive inference

- **Secrecy**
  - Secret goes only to intended recipients
    - Pipes do not leak
  - Global reasoning
  - Negative inference

Divide et Conquera

- Two coordinated logics
  - Logic of authentication
    - Relies on secrecy assumptions
      - Proof obligation in secrecy logic
  - Logic of secrecy
    - Relies on authentication assumptions
      - Proof obligation in auth. logic
- Benefits
  - Much simpler proofs
  - Modularity
    - Independent of notion of secrecy

Describing Protocol Runs

- Messages
  - k m - encryption
  - m,m' - pairing
- Principal actions
  - $\langle m: A \text{ -> } B \rangle_A$ - send
  - $(X: Y \text{ -> } Z)_A$ - receive
  - $(m/p(x))_A$ - match
  - $(\nu\ n)_A$, $(\tau\ t)_A$ - new nonce, timestamp

Abbrv.

$\langle\langle m \rangle\rangle_A$

$((m))_A$

$\langle m \rangle_{A<}$

- Runs
  - Partial order of actions
    - Every receive has a send
    - Every match has succeeded
  - Observations
- Protocols
  - Set of parametric roles
    - Akin to observations

Authentication Logic

- First-Order logic with 3 predicates
  - a_A – action a_A has occurred
  - $a_A < b_B$ – a_A has occurred before b_B
  - $a_A = b_B$ – a_A and b_B are the same action

  Nothing else!
- Usage
  - Given A's observations, extend them with other principal's actions
    - Derive compatible runs

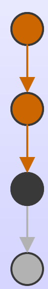
    $A: Obs_A \rightarrow \Phi$

    $A: \Psi \ \& \ Obs_A \rightarrow \Phi$
  - Iterated application of axioms

Logical Assumptions

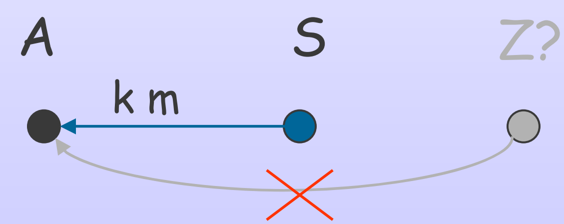
- Honesty
  - Principal does not deviate from role

honest S



- Secrecy
  - Key uncompromised for given principals

$$\mathit{secret}(k, G) = \langle\langle k\ m \rangle\rangle_{X<} \rightarrow X \in G \;\; \& \;\; (x/k\ y)_X \rightarrow X \in G$$

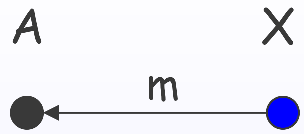


secret(k,[A,S])

Axioms

- Basic truths about domain
  - <u>Receive axiom</u>

    $Y: ((m))_A \rightarrow \langle\langle m \rangle\rangle_{X<} < ((m))_A$

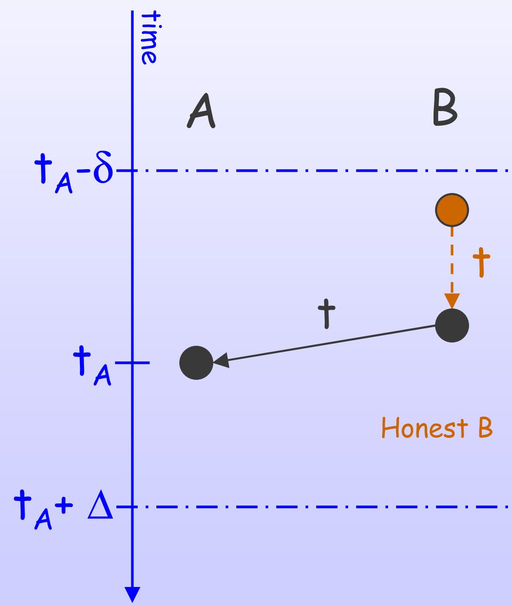


  - <u>Timestamp axiom</u>

    A: honest B &

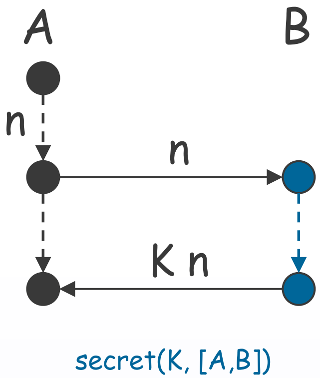
    $\langle\langle t \rangle\rangle_{B<} < ((t))_A$

    $\rightarrow (t\text{-}\delta)_A < (\tau\ t)_B < \langle\langle t \rangle\rangle_{B<} < ((t))_A < (t\text{-}\Delta)_A$



- Allow inferring new actions/ordering

Schemas and Instances



- Desired functionalities
  - Nonce-based Challenge-Response property

    A: Φ &

    $(\nu\ n)_A < \langle\langle C\ n \rangle\rangle_{A<} \quad < ((R\ n))_A$

    $\Rightarrow (\nu\ n)_A < \langle\langle C\ n \rangle\rangle_{A<} < ((C\ n))_B < \langle\langle R\ n \rangle\rangle_{B<} < ((R\ n))_A$

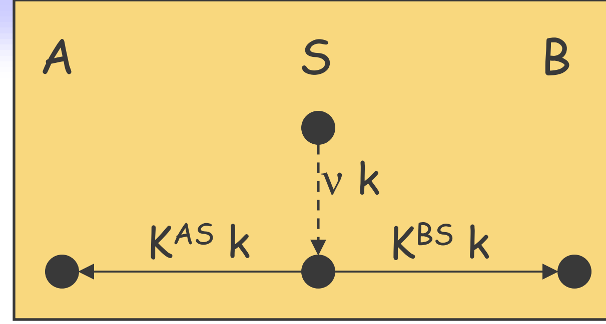
- Verified instances
  - Challenge in the clear/Response encrypted

    A: secret(K, [A,B]) &

    $(\nu\ n)_A < \langle\langle n \rangle\rangle_{A<} \quad < ((K\ n))_A$

    $\Rightarrow (\nu\ n)_A < \langle\langle n \rangle\rangle_{A<} < ((n))_B < \langle\langle K\ n \rangle\rangle_{B<} < ((K\ n))_A$

Abstract Key Distribution

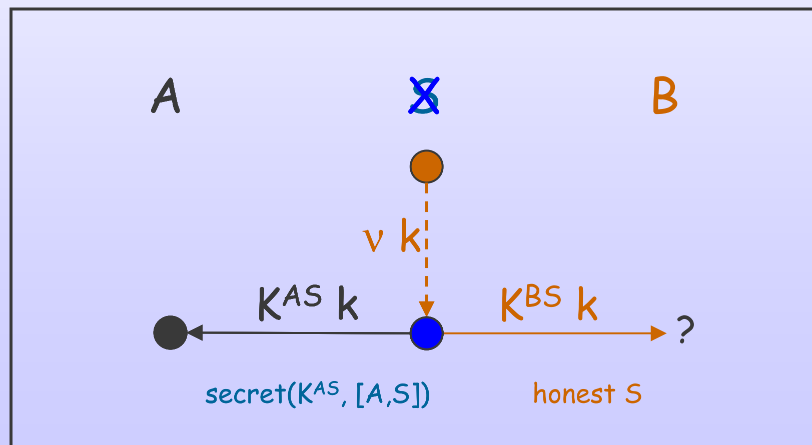


- S spontaneously
  - Generates k
  - Sends it to A, B
    - A, B hardwired
  - Encrypted with K^{AS}, K^{BS}
- A observes only $(K^{AS}\ k)$

A: secret(K^{AS}, [A,S]) & honest S & $(K^{AS}\ k)_A$

$\rightarrow (\nu\ k)_S < \left(\langle K^{AS}\ k\rangle_S <,\ \langle K^{BS}\ k\rangle_S <\right) < (K^{AS}\ k)_A$

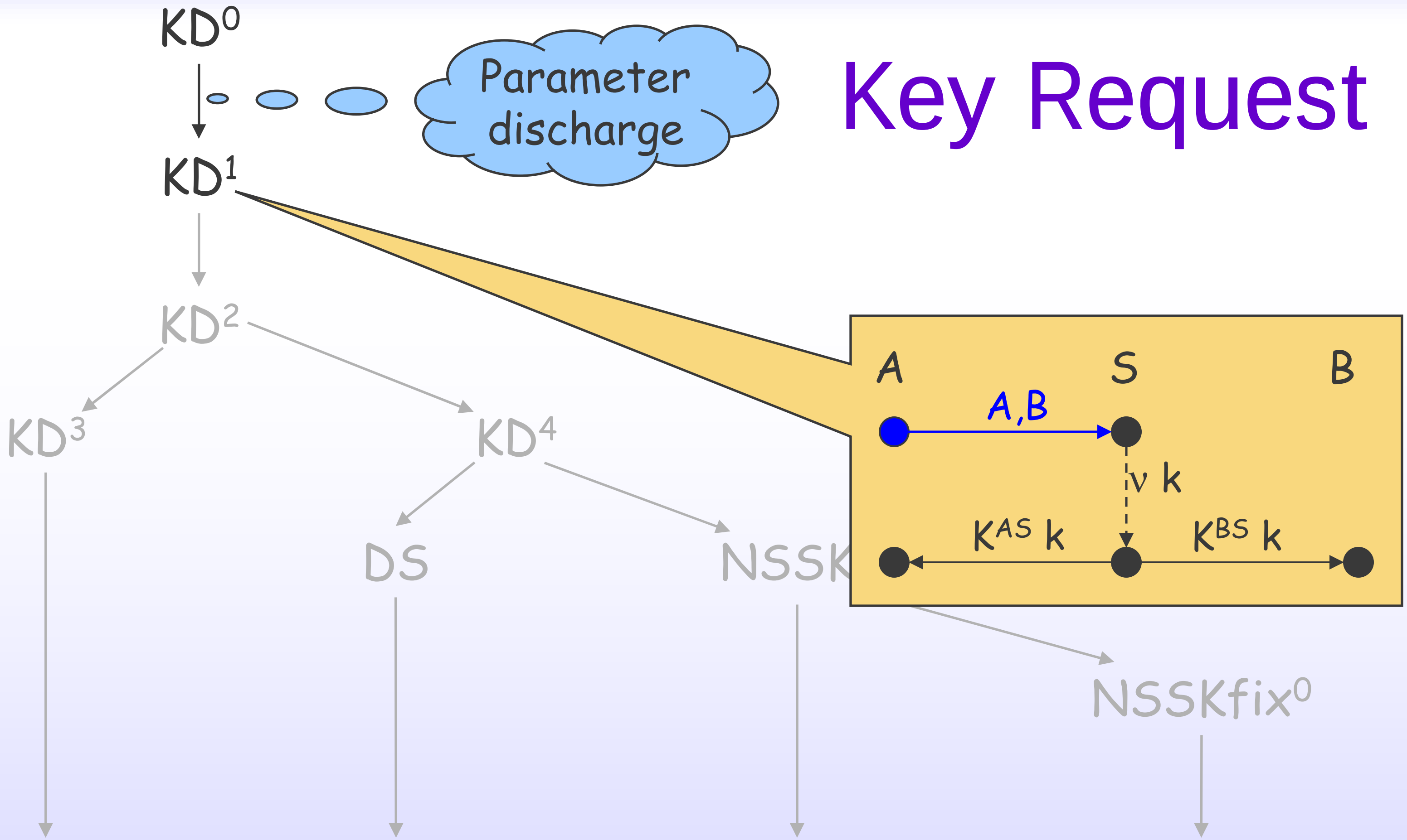
- A reconstructs run
  - Must assume
    - honest S
    - secret(K^{AS}, [A,S])
    - Not secret(K^{BS}, [B,S])
  - B's reception unknown
- Dual for B



Derivational Approach

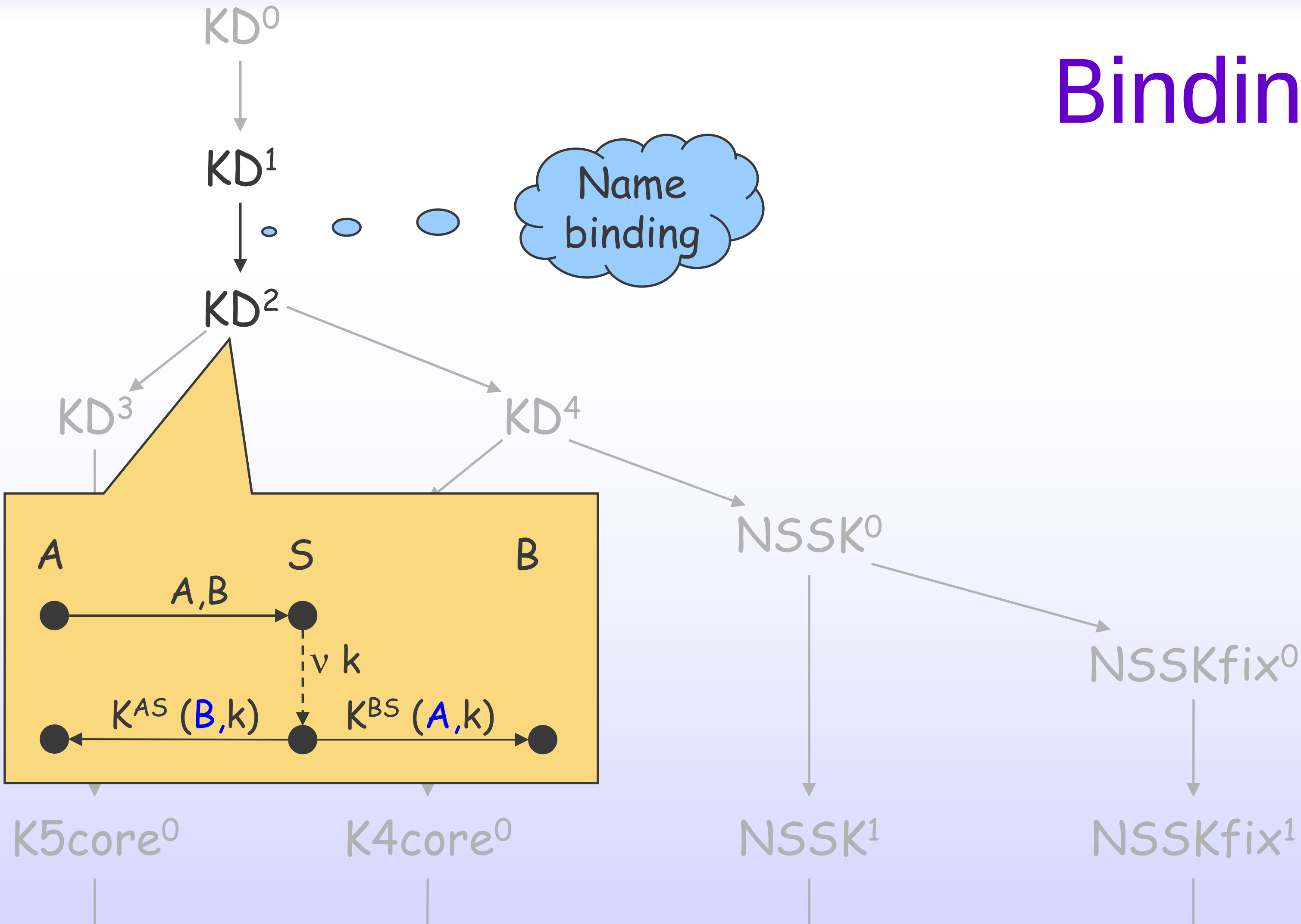
- Use rules, not just axioms
    - Operate on protocol and properties
  - Refinements
  - Transformations
- Advantages
  - Abstract general constructions
  - Reuse protocol fragments
  - Structured understanding of
    - Mechanism
    - Properties
    - Relations between protocols
  - Open-ended taxonomies

Key Request



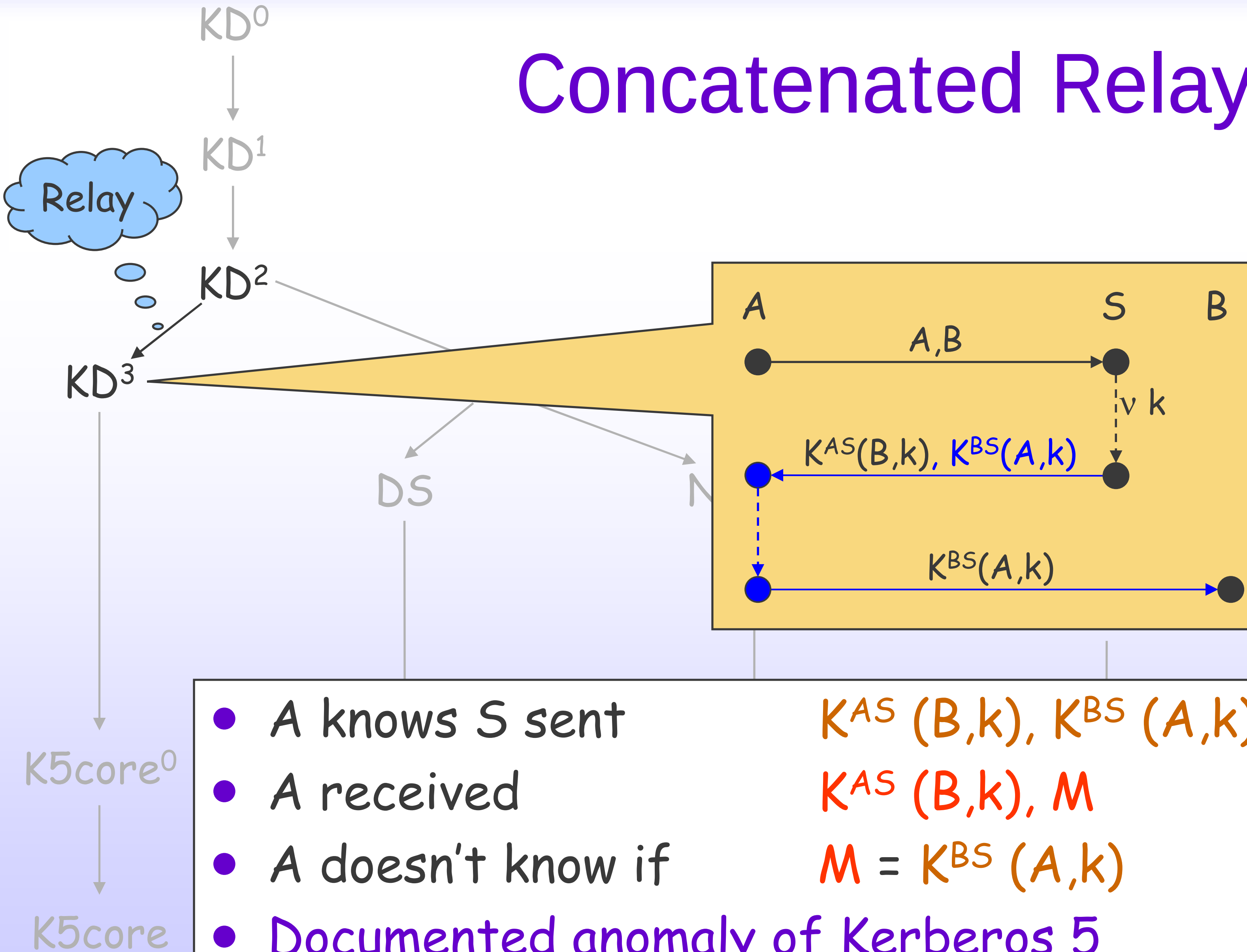
- A may not be talking to B
  - Even if S honest
- Same for B

Binding



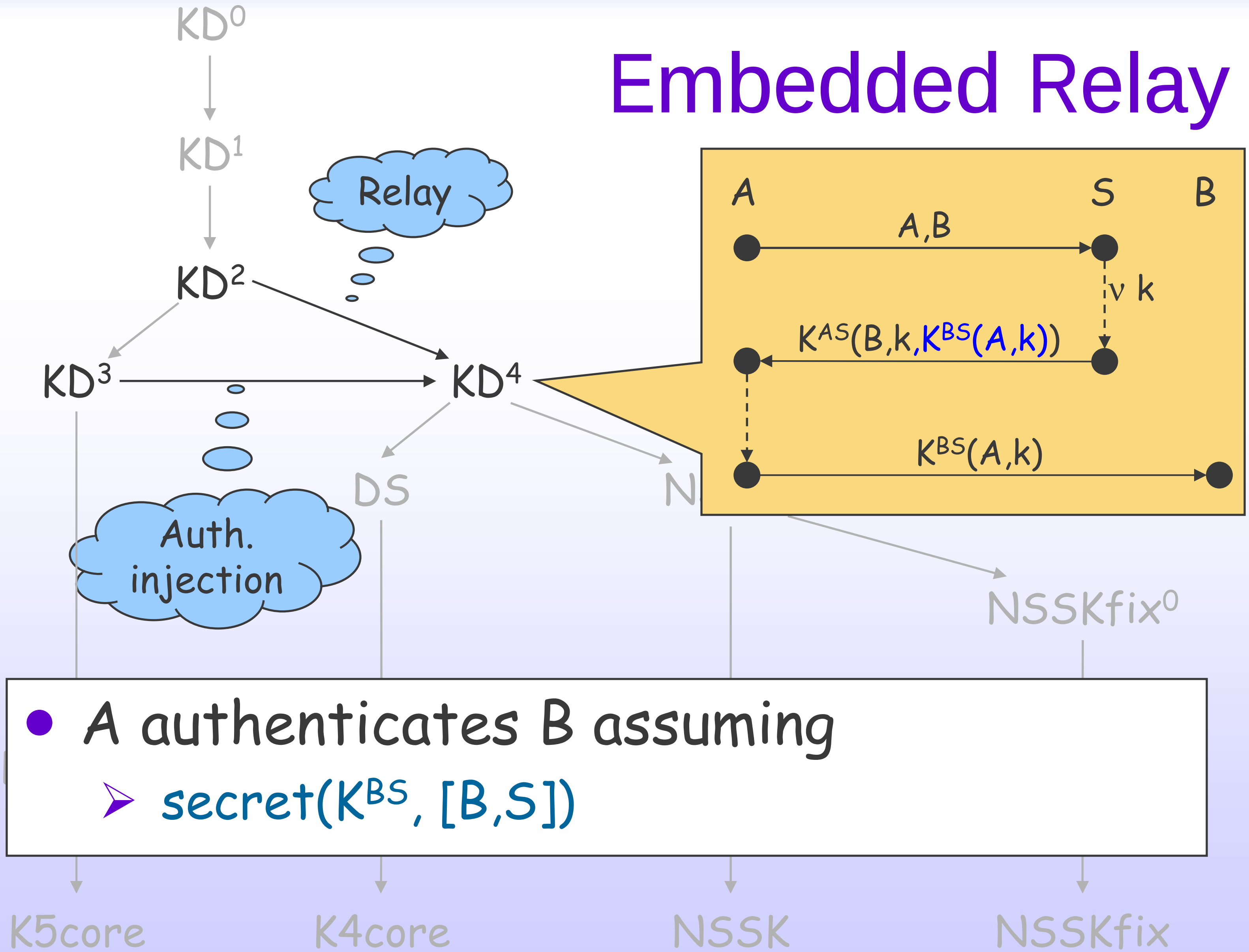
- A (B) authenticated to B (A)

Concatenated Relay



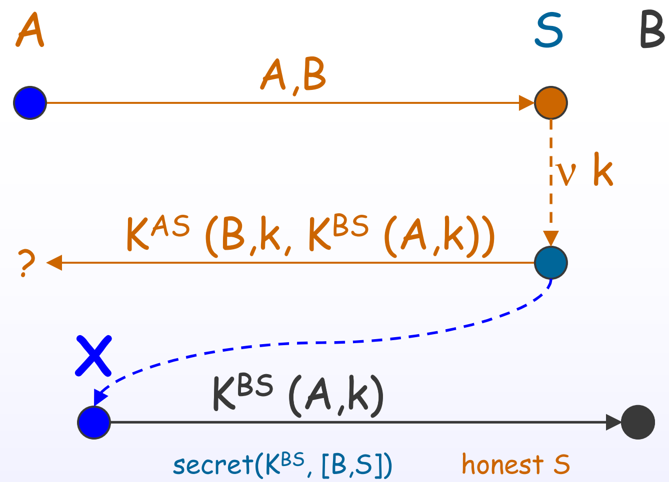
- A knows S sent K^{AS} (B,k), K^{BS} (A,k)
- A received K^{AS} (B,k), M
- A doesn't know if M = K^{BS} (A,k)
- Documented anomaly of Kerberos 5

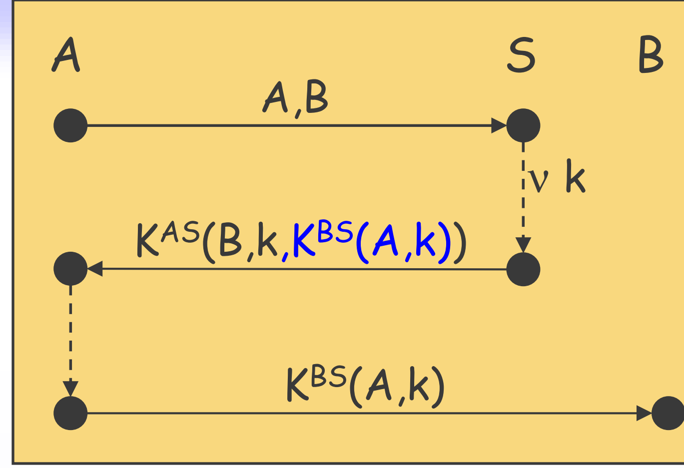
Embedded Relay



- A authenticates B assuming
  - ➢ secret(K^{BS}, [B,S])

B's Point of View

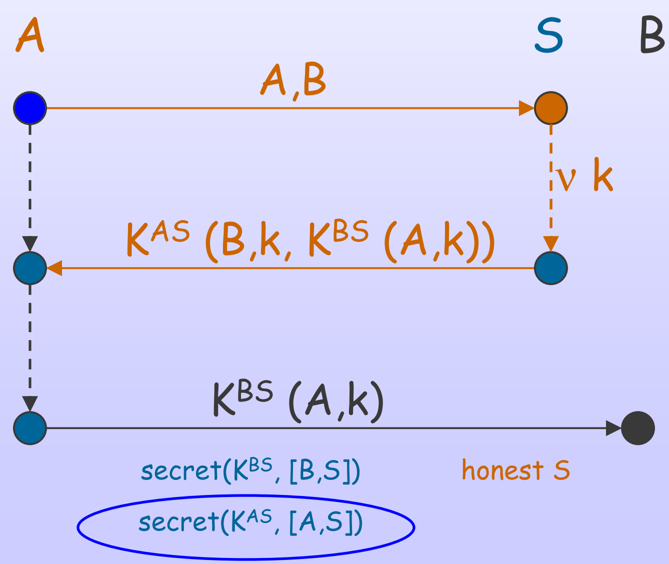




- With only
  - secret(K^{BS}, [B,S])

  knows S generated k
- With also
  - secret(K^{AS}, [A,S])

  knows A knows k
  - A may not be honest

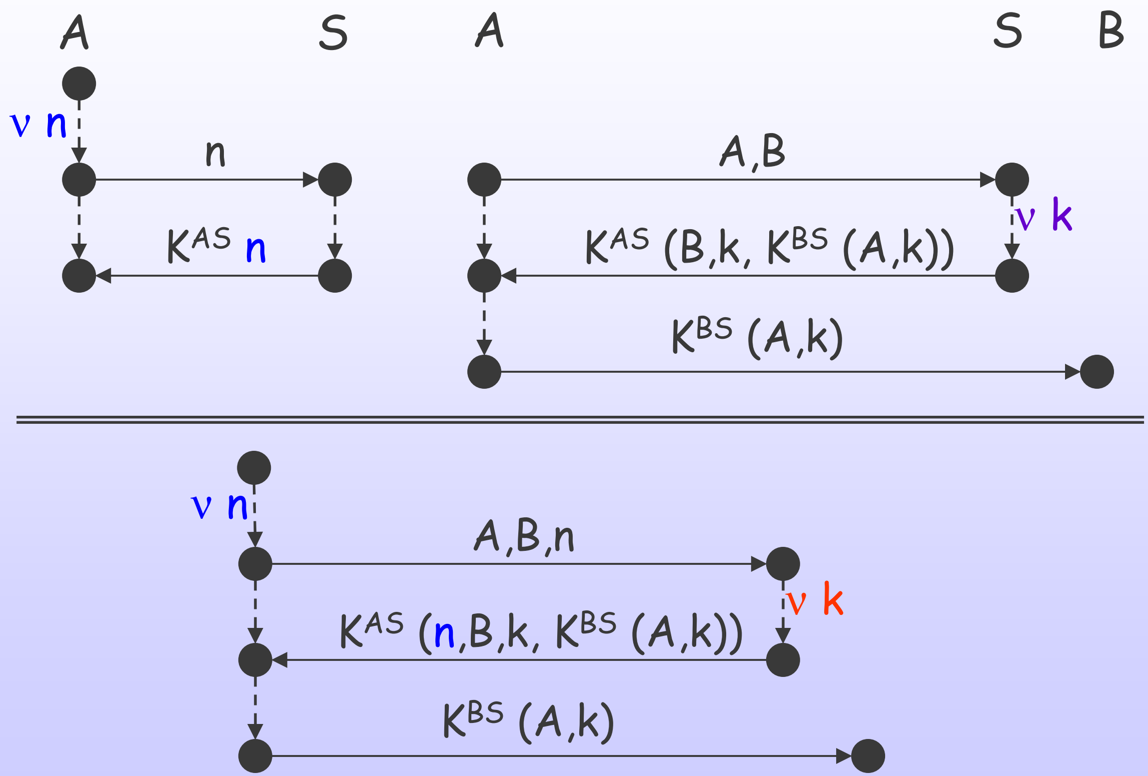


Additional Properties

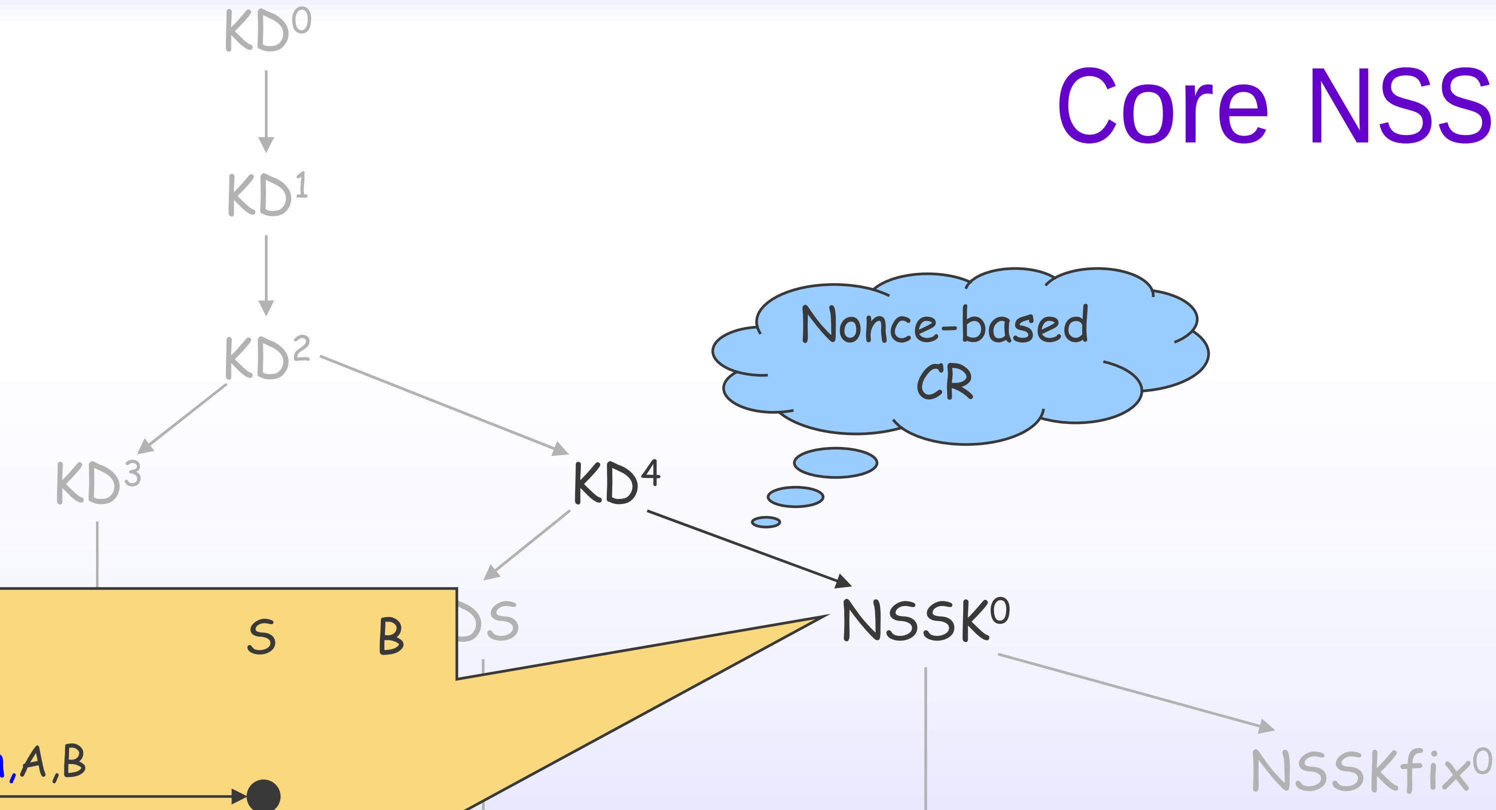
- Recency
  - $(\nu\ k)_S$ bracketed by events controlled by A/B
    - Otherwise, intruder can infer k and attack protocol
    - Even if S is honest
  - Not satisfied so far
- Key confirmation
  - A/B knows that B/A has k
    - Essential for using k
  - Only B in KD^4 (under assumption)

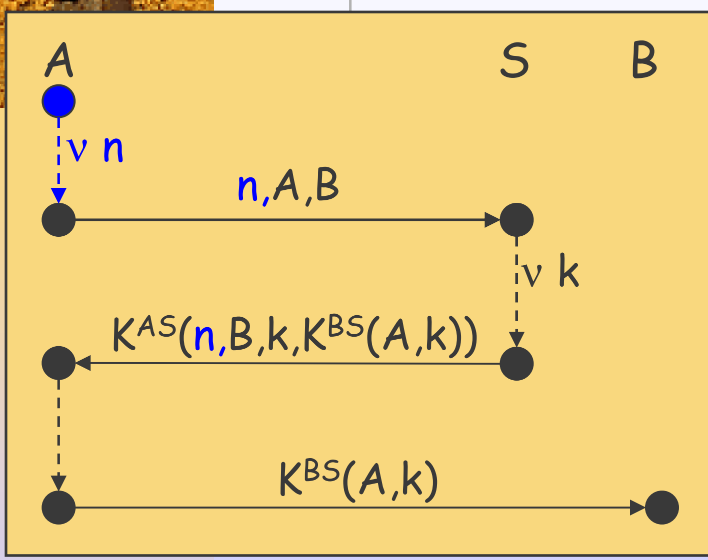
Recency with Nonces

- Use challenge-response as bracket



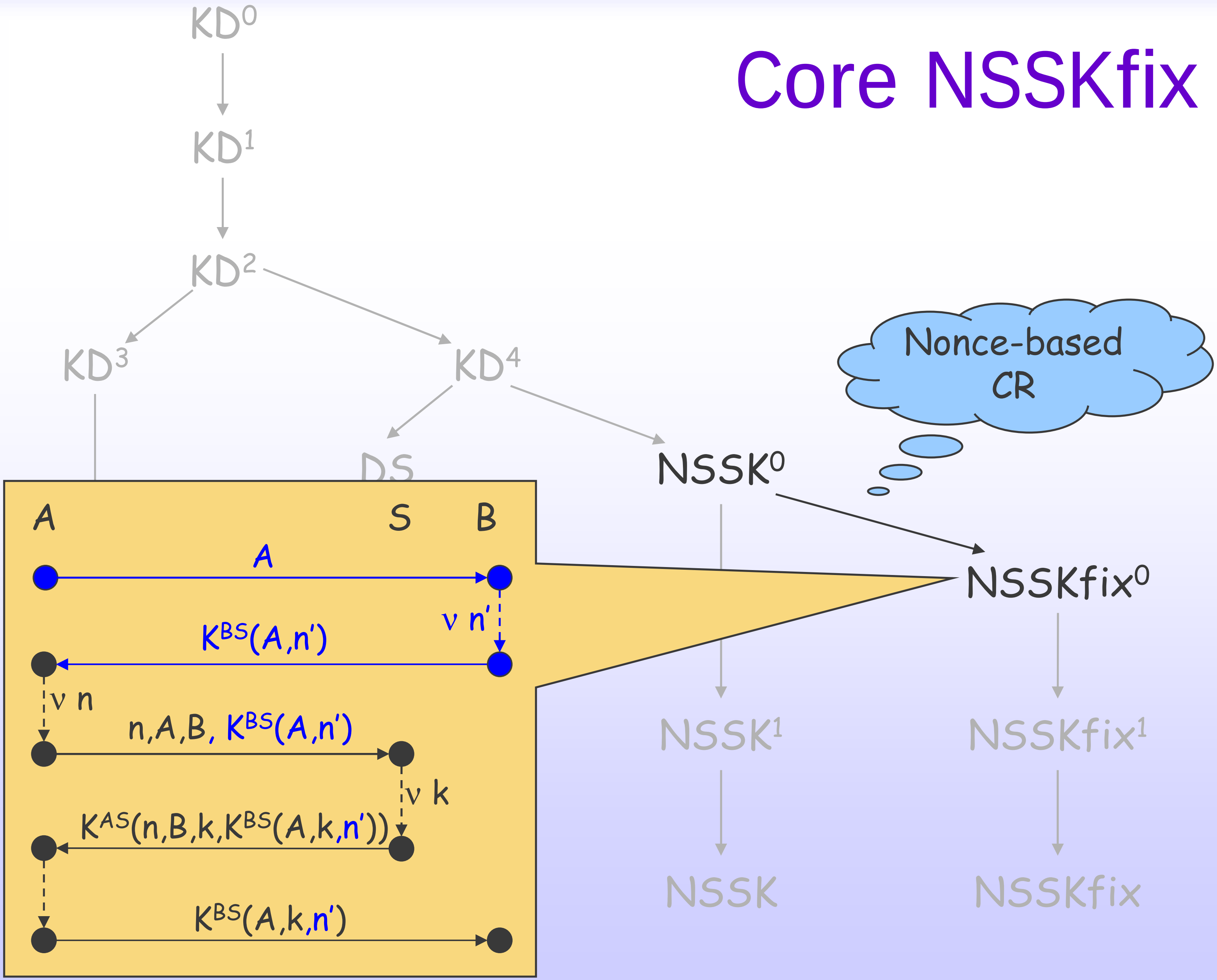
Core NSSK



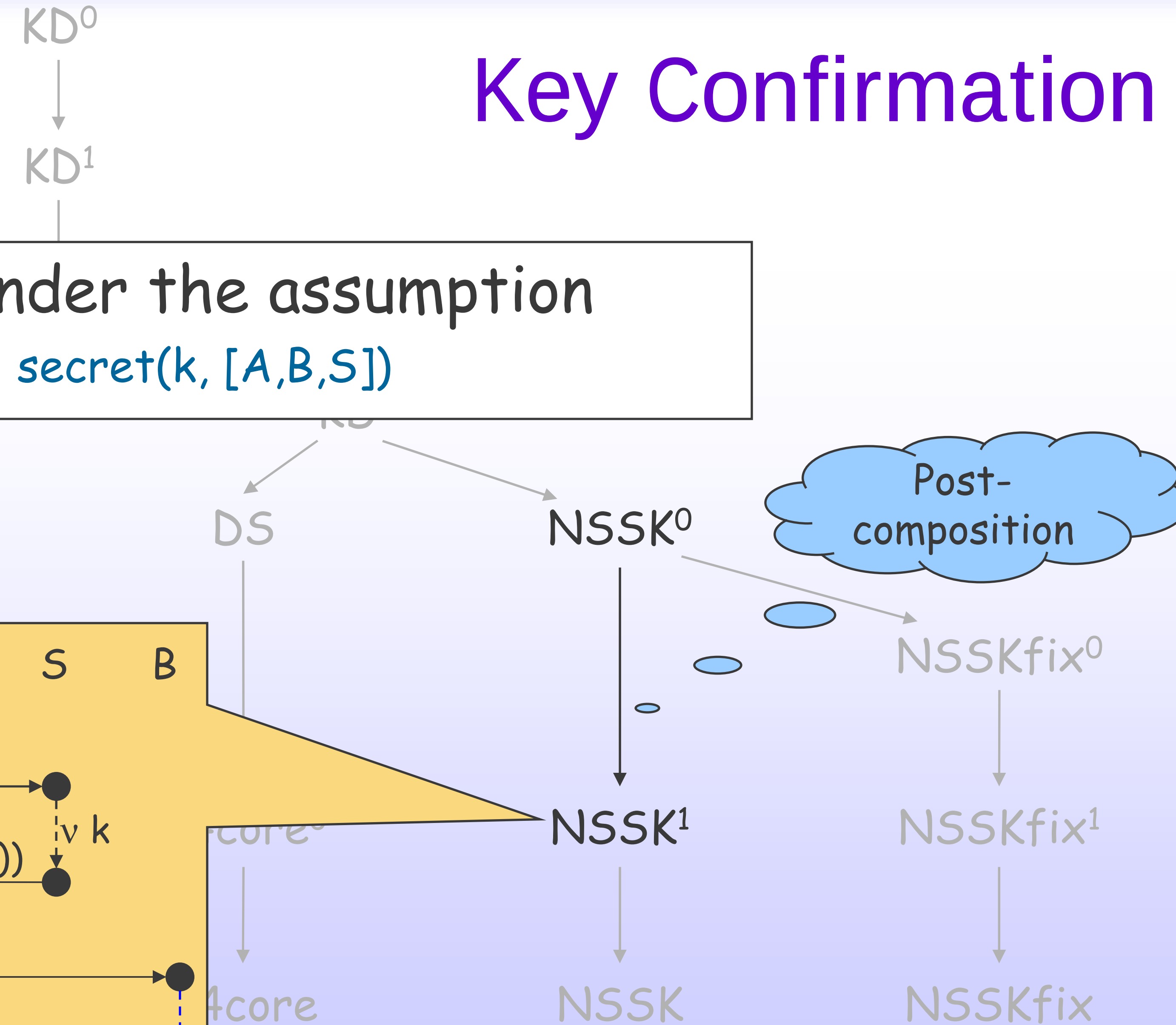


- Ensures recency of k to A
- A can reconstruct run up to B's action
- No such guarantees for B
  - Denning-Sacco attack

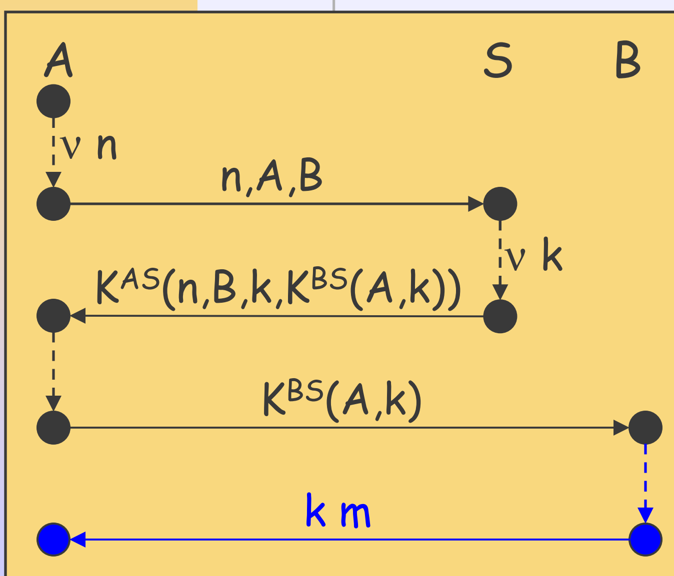
Core NSSKfix



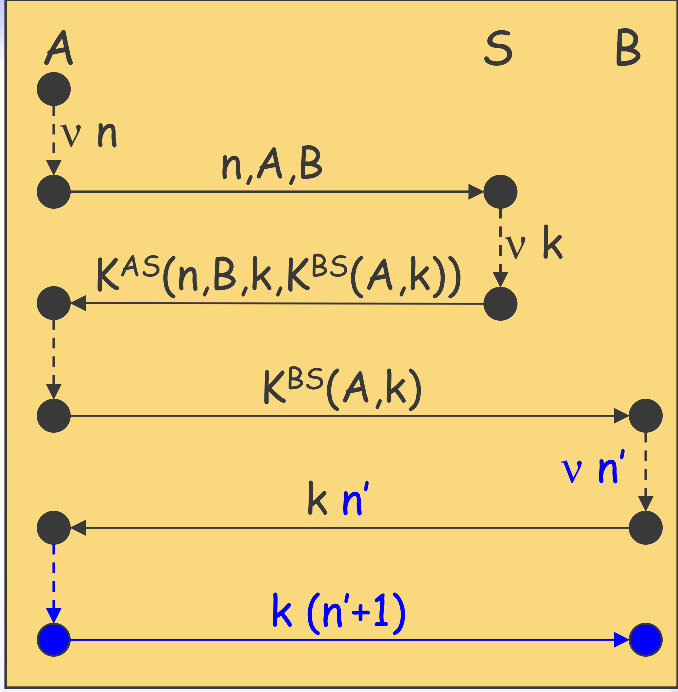
Key Confirmation



- Under the assumption
  - secret(k, [A,B,S])



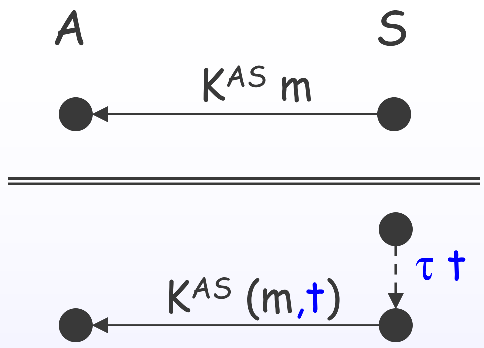
NSSK does more!

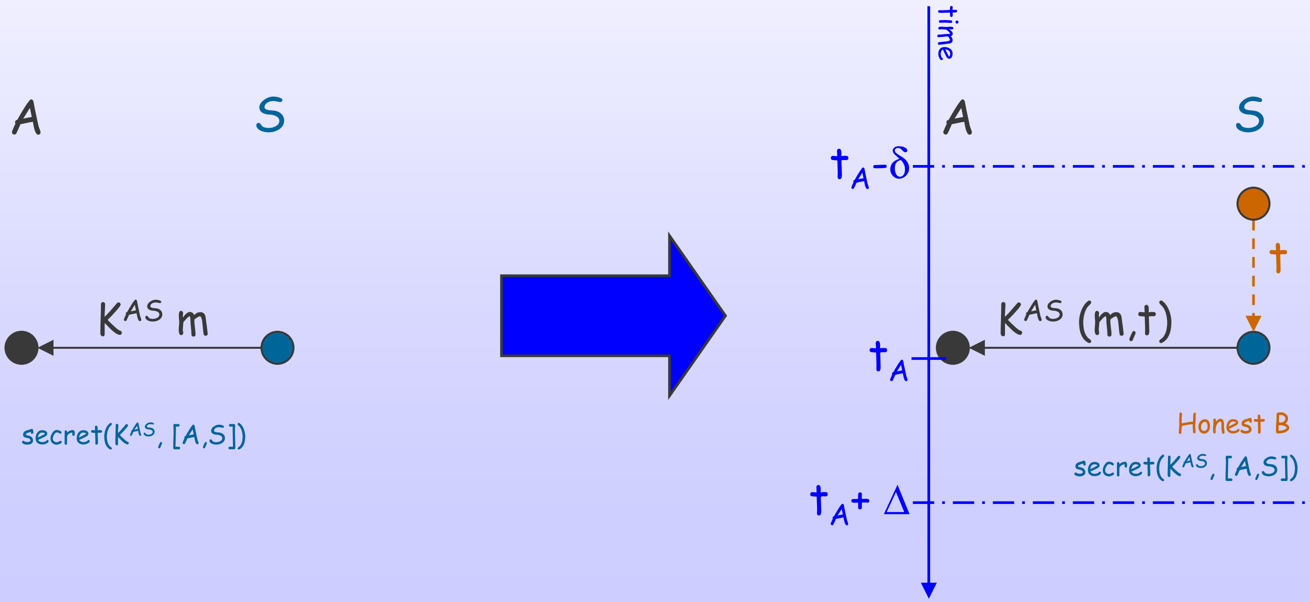


- B concludes with CR
  - k **not** confirmed to A
    - Unless tagging
  - B already knows A has k
- Exchange typical of repeated authentication
  - B repeatedly request service from A
    - … but A is initiator!
- Similarly for NSSK-fix

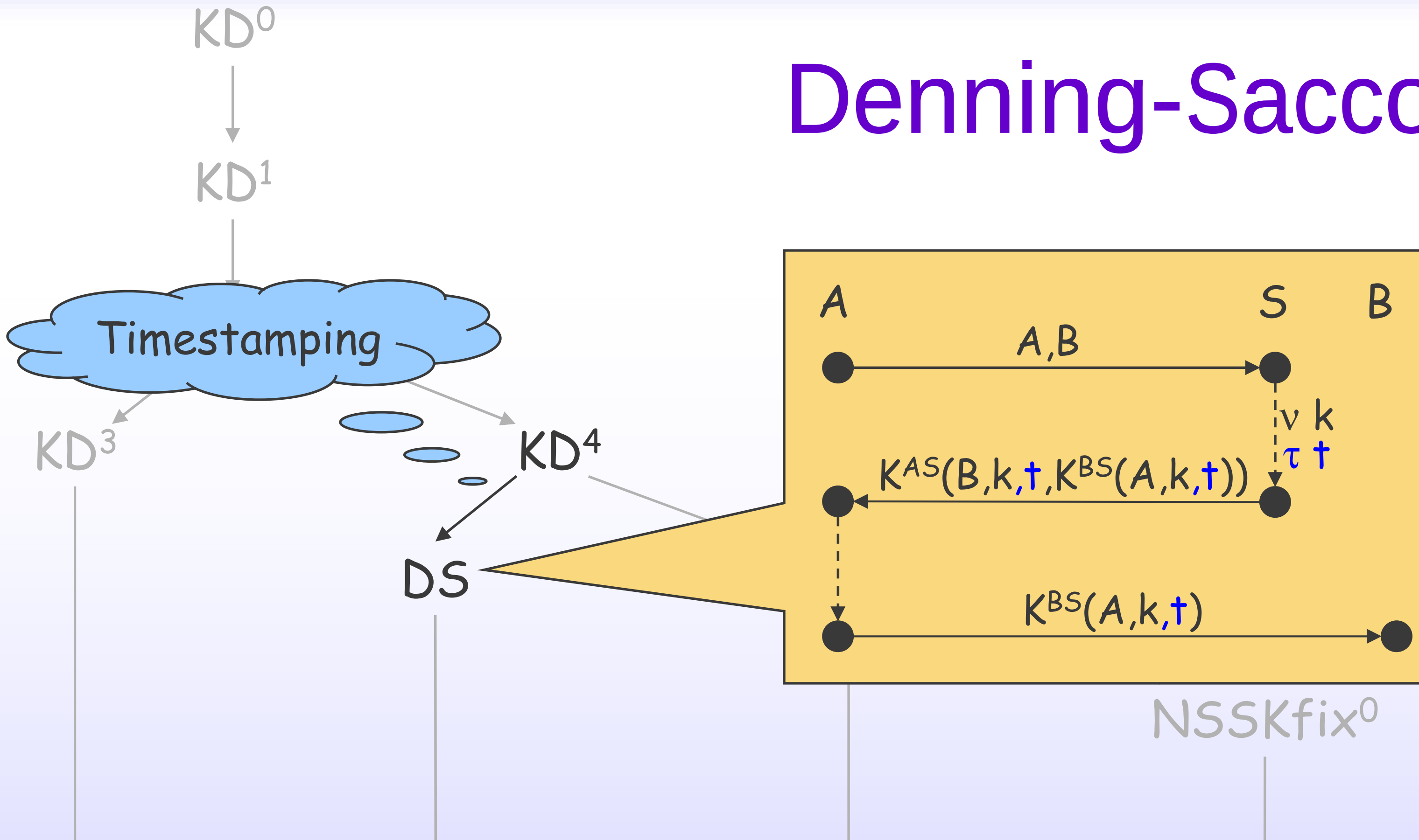
Recency with Timestamps

- Timestamp as bracketing device
  - Requires loosely synchronized clocks

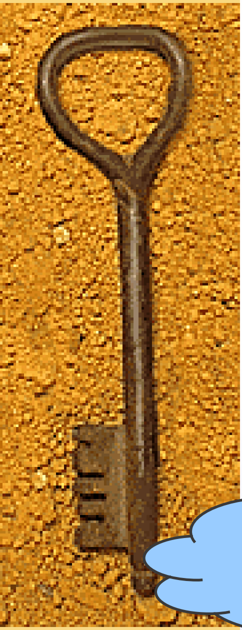




Denning-Sacco

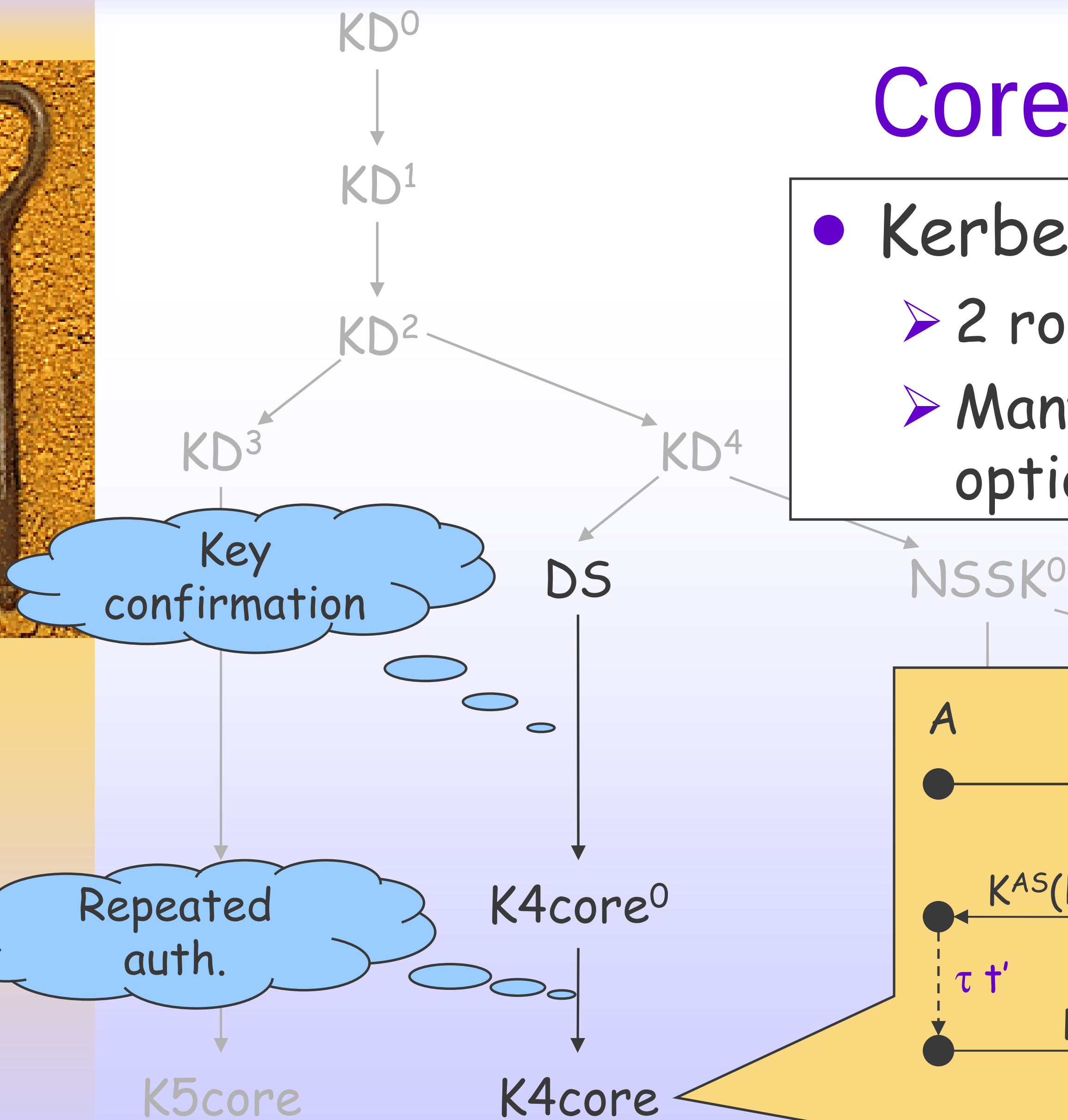


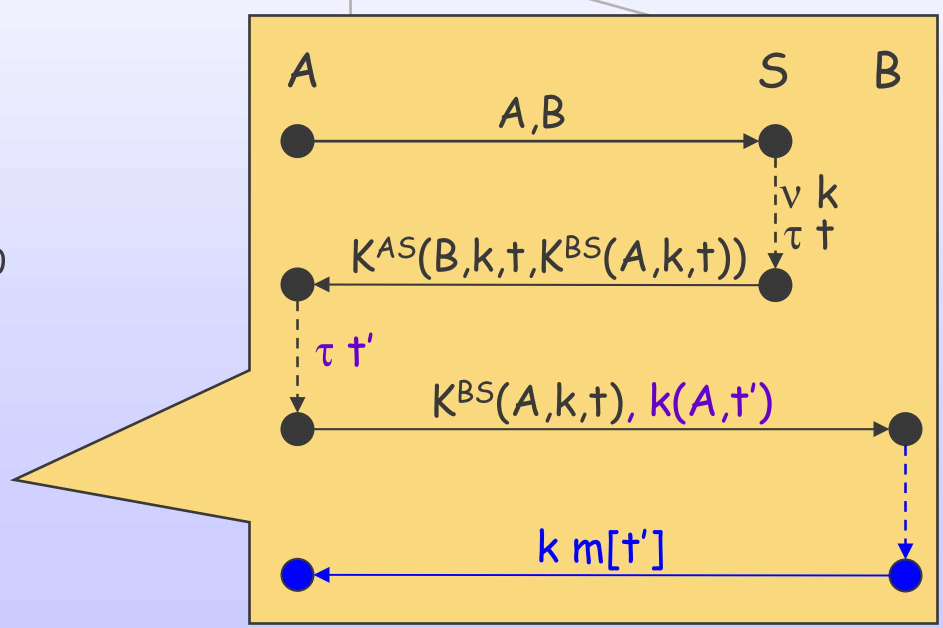
- Guarantee recency to both A and B
- Same assurance as core NSSK-fix
  - Only 3 messages



Core Kerberos 4

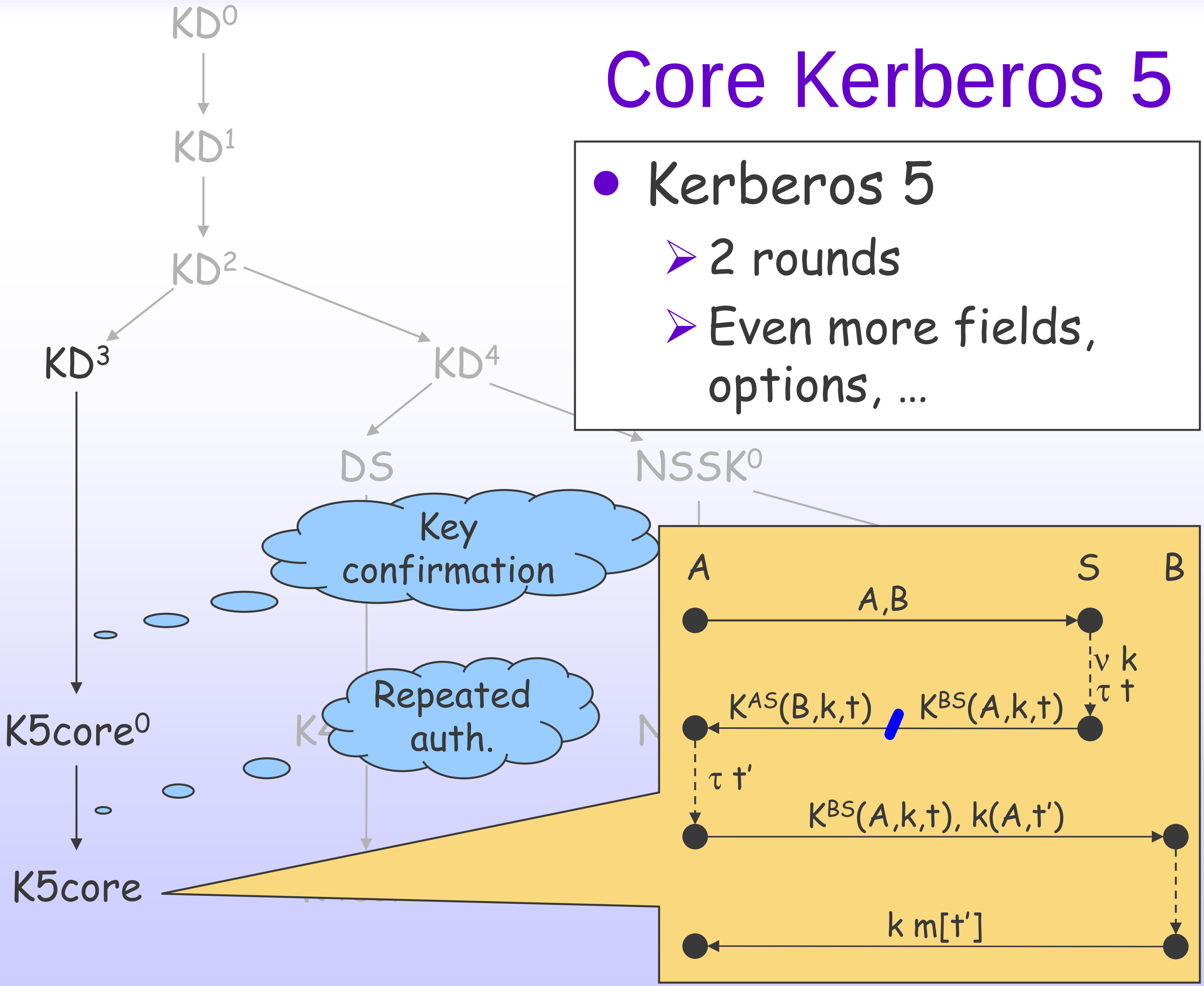
- Kerberos 4
  - 2 rounds
  - Many more fields, options, …





Core Kerberos 5

- Kerberos 5
  - 2 rounds
  - Even more fields, options, …



Current ~~Future~~ Work

Define Secrecy Logic

- Authentication as assumptions
- Modular model of secrecy
  - Dolev-Yao
  - Information-theoretic
  - Computational
- Apply to examples
  - Diffie-Hellman hierarchy
  - Full Kerberos 5
  - PKINIT

Future

- Implement within Kestrel's PDA