What you may have understood of Dusko's talk yesterday if he hadn't been speaking so fast

Part I

Dusko Pavlovic

Kestrel Institute

Iliano Cervesato

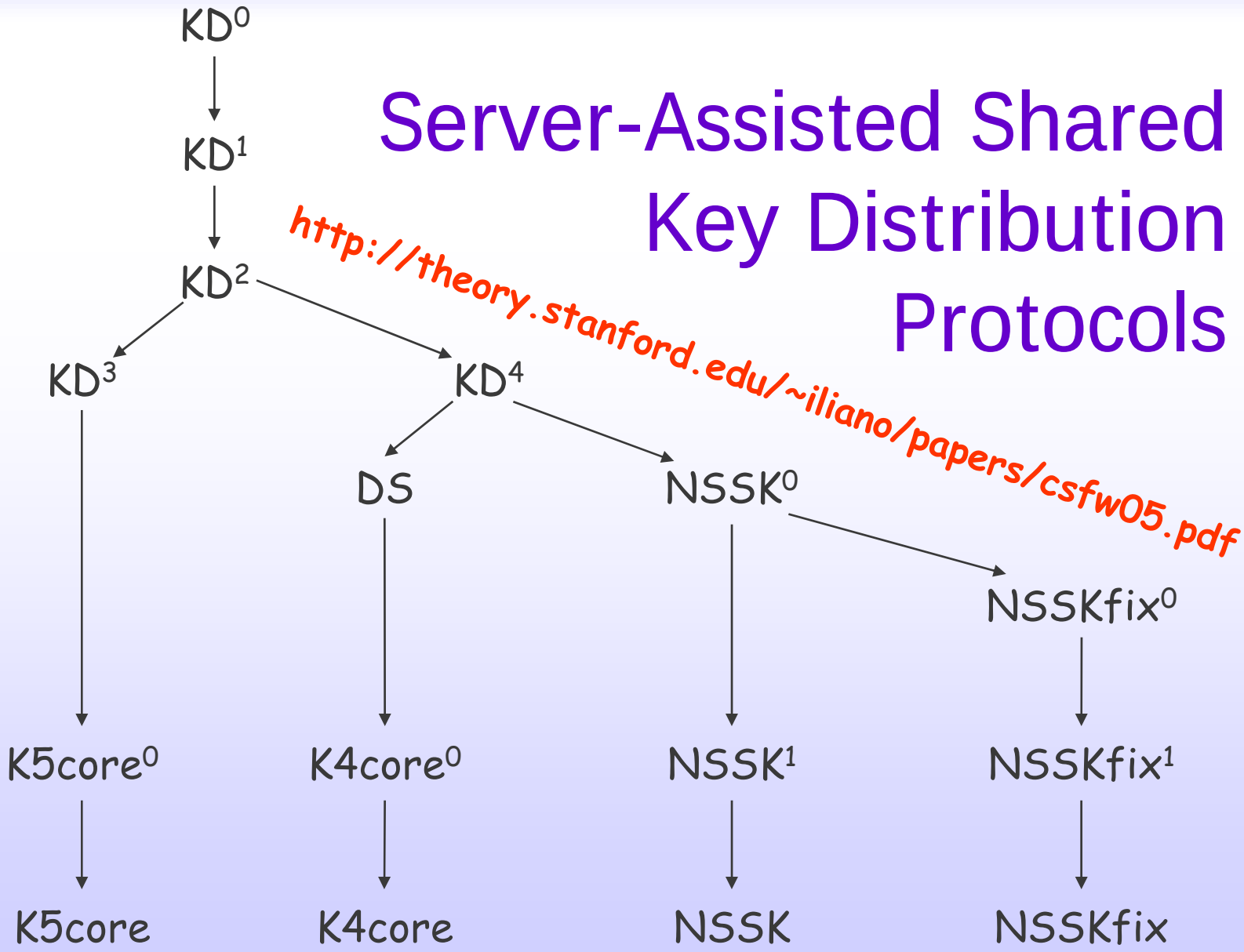
Tulane University

Contributions

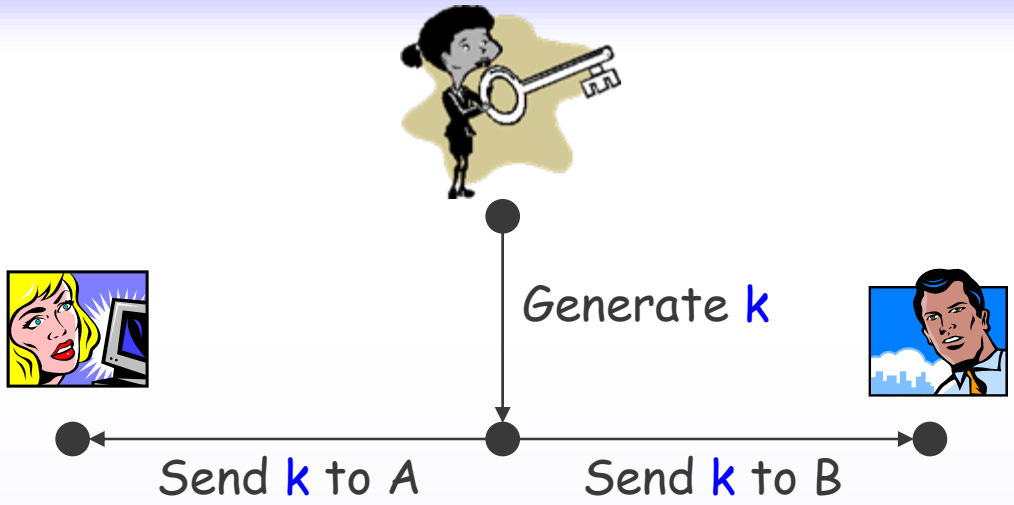
- Separate
  - Authentication reasoning
  - Secrecy reasoning
- Define a logic of pure authentication
  - Secrecy as assumptions
- Embed it in derivational framework
- Apply to shared-key server-assisted key distribution protocols
  - Taxonomy
  - Comparative study
  - Clear understanding of underlying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

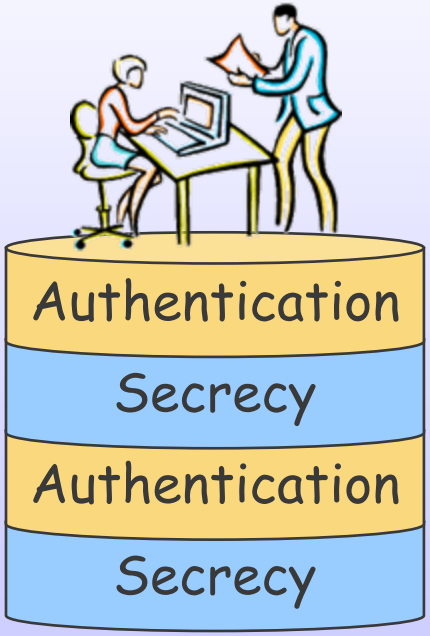
Server-Assisted Shared Key Distribution Protocols

http://theory.stanford.edu/~iliano/papers/csfw05.pdf



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

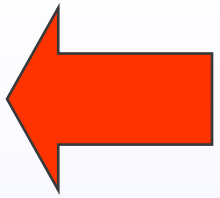
… BUT …

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
  - Logic of secrecy
    - Relies on authentication assumptions



- Benefits
  - Much simpler proofs
  - Modularity
  - Deeper understanding of
    - mechanisms
    - properties

Describing Protocol Runs

- Principal actions
  - $\langle m: A \rightarrow B \rangle_A$ – Send
  - $(X: Y \rightarrow Z)_A$ – Receive
  - $(m/p(x))_A$ – match
  - $(\nu\ n)_A$, $(\tau\ t)_A$ – new nonce, timestamp
- 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

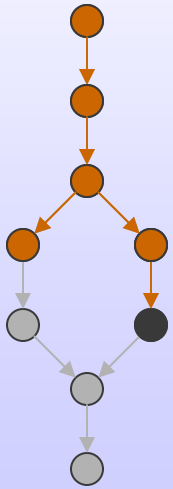
    $A: Obs_A \rightarrow \Phi$

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

Logical Assumptions

- Honesty
  - Principal does not deviate from role

honest S



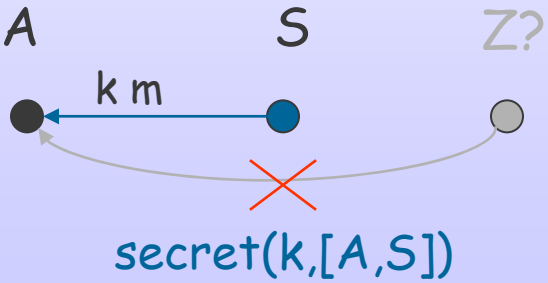
- Secrecy
  - Key uncompromised for given principals

secret(k, G) =

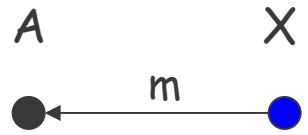
$\langle\langle k\ m\rangle\rangle_X \rightarrow X \in G$

&

$(x/k\ y)_X \rightarrow X \in G$



Axioms



- Basic truths about domain
  - Receive axiom

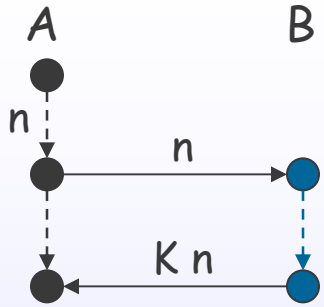
    Y: $((m))_A \rightarrow \langle\langle m\rangle\rangle_{X<} < ((m))_A$

  - Challenge-response axiom

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

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

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



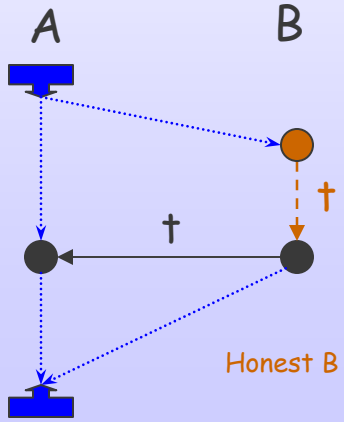
secret(K, [A,B])

  - Timestamp axiom

    A: honest B &

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

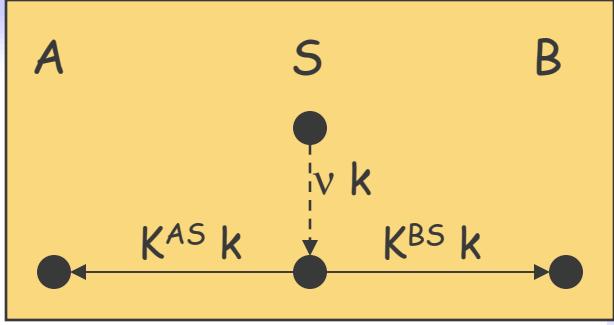
    $\rightarrow (\downarrow t)_A < (\tau\ t)_B < \langle\langle t\rangle\rangle_{B<} < ((t))_A < (\uparrow t)_A$



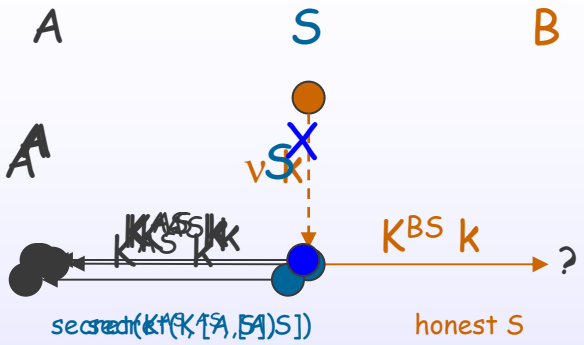
- Allow inferring new actions/ordering

KD^0

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

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

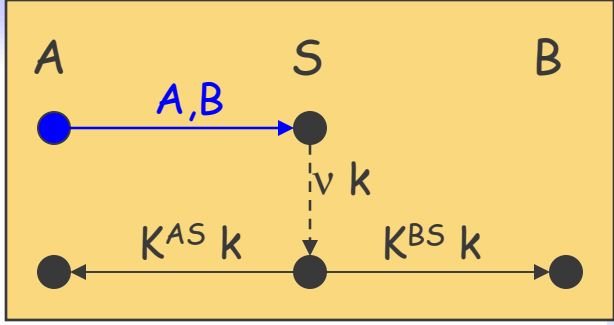
$$\rightarrow \quad \langle K^{AS}\ k\rangle_{S<} \;<\; ((K^{AS}\ k))_{AA}$$

$$\rightarrow (\nu\ k)_S \;<\; \begin{bmatrix} \langle K^{AS}\ k\rangle_{S<} \\ \langle K^{BS}\ k\rangle_{S<} \end{bmatrix} \;<\; (K^{AS}\ k)_A$$

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

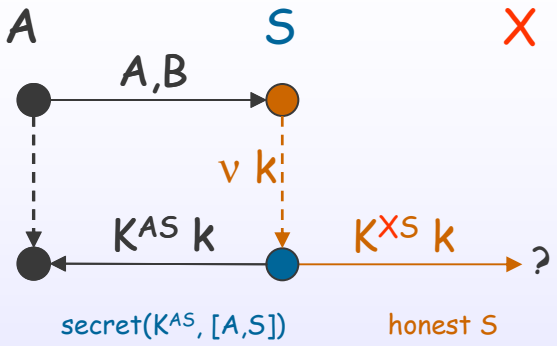
Key Request



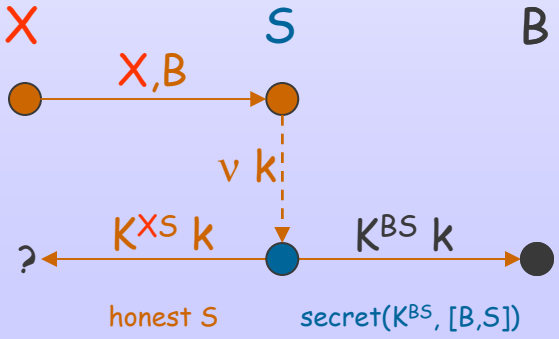
- *A issues request*

$$A: \quad \text{secret}(K^{AS}, [A,S]) \;\&\; \text{honest } S \;\&\; \langle A,B \rangle_A \prec (K^{AS}\ k)_A$$

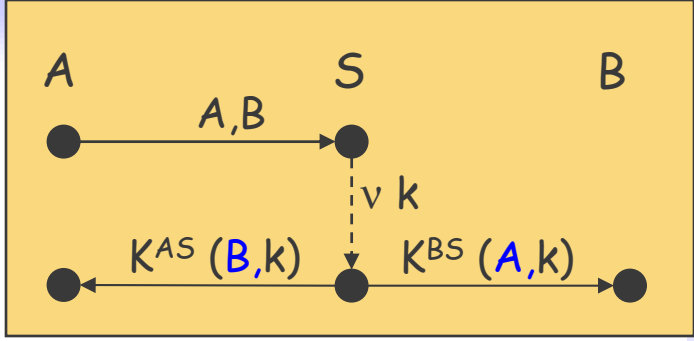
$$\rightarrow \left(A: \langle A,B \rangle_A \prec (K^{AS}\ k)_{AB} \;\;\&\;\; (A,X)_S \prec (\nu\ k)_S \prec \begin{bmatrix} \langle K^{AS}\ k \rangle_{S\prec} \\ \langle K^{XS}\ k \rangle_{S\prec} \end{bmatrix} \right) \prec (K^{AS}\ k)_A$$



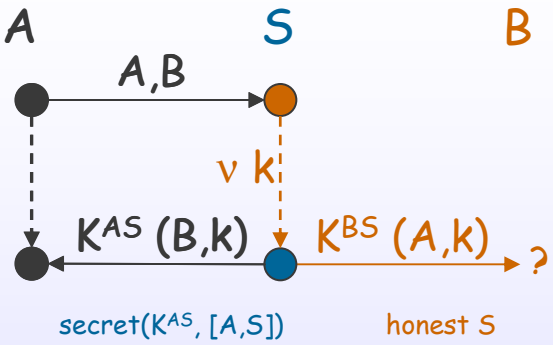
- *A may not be talking to B*
  - Even if S honest
  - Same holds for B



Binding



- S includes names

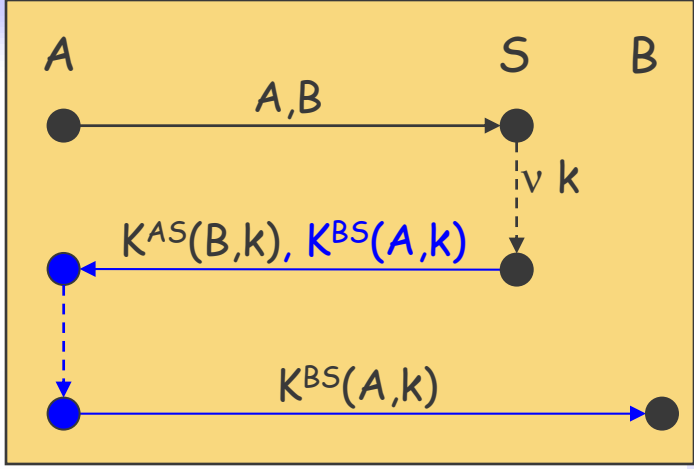


- A (B) authenticated to B (A)

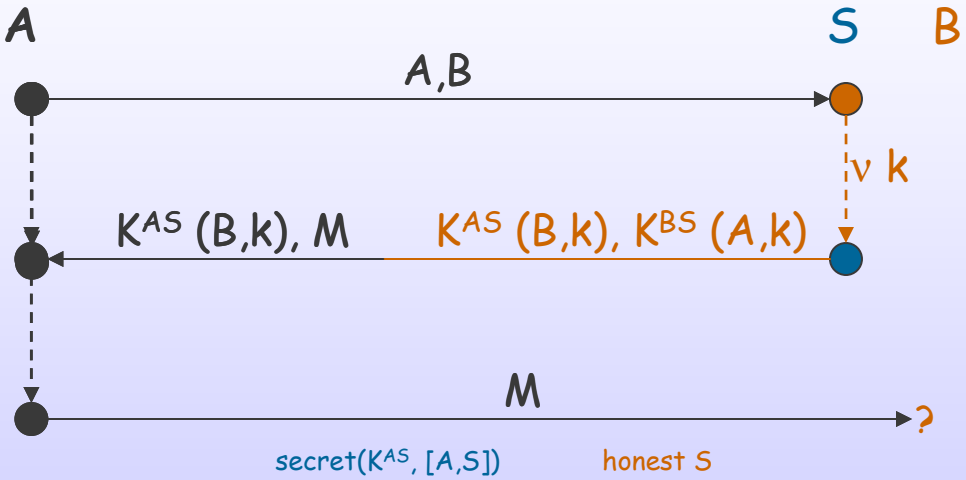
Similar for B

- Not how typical KD protocols are set up
  - S sends to 1 party only

Concatenated Relay



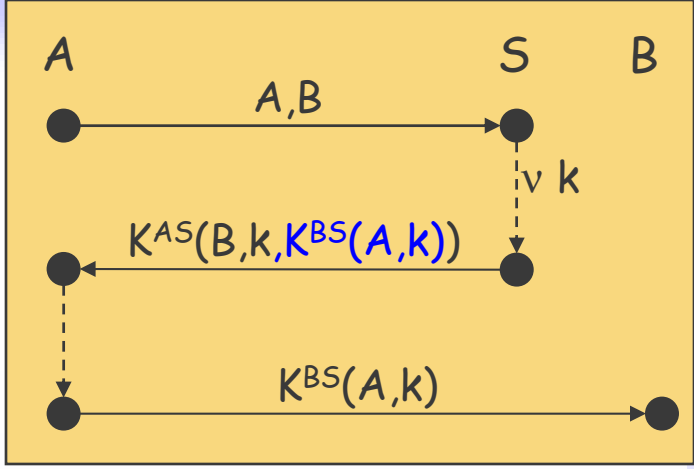
- S sends all to A
  - A forwards to B
- Seedling of Kerberos 5

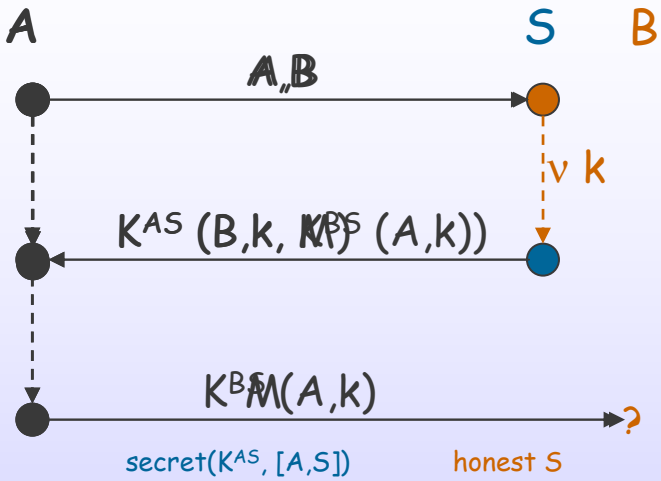


- A knows S sent
  $K^{AS}(B,k), K^{BS}(A,k)$
- A received
  $K^{AS}(B,k), M$
- A doesn't know if
  $M = K^{BS}(A,k)$
- Documented anomaly of Kerberos 5

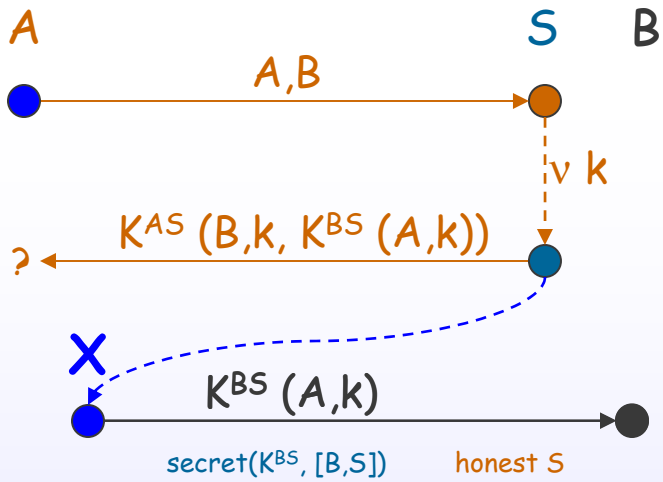
Embedded Relay

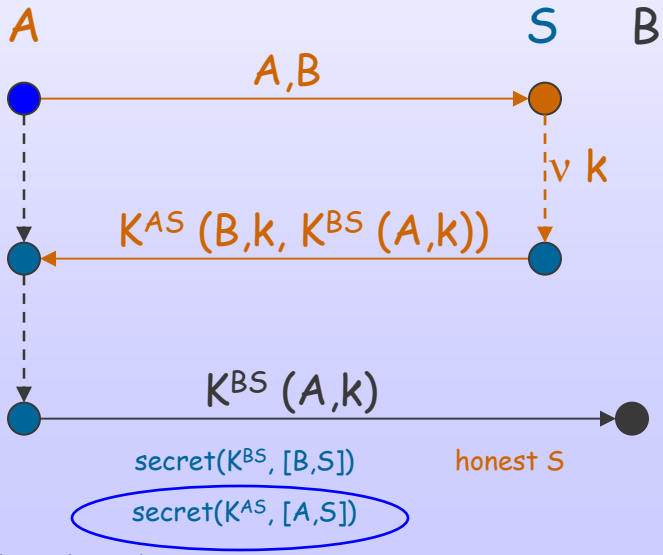
- Encrypted "ticket"
- Basis of
  - NSSK
  - Denning Sacco
  - Kerberos 4

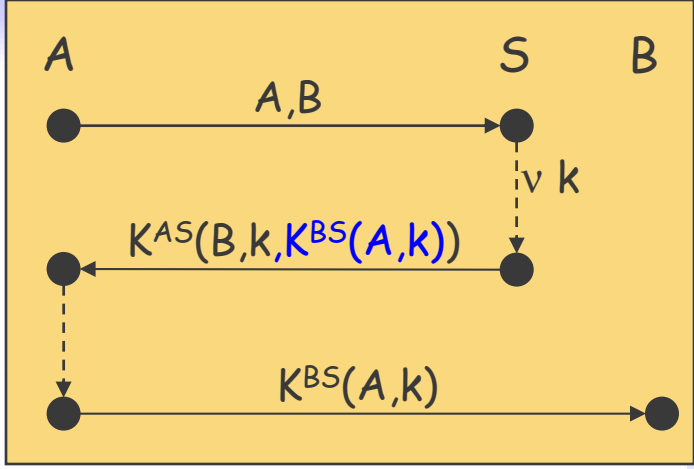




B's Point of View







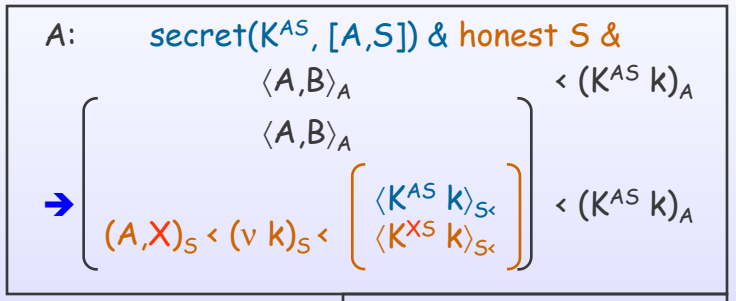
- With only
  - secret(K^{BS}, [B,S])

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

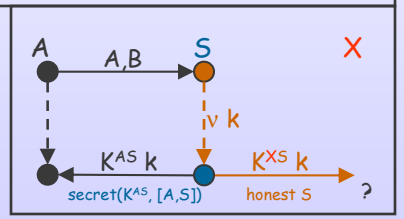
  knows A knows k
  - A may not be honest

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

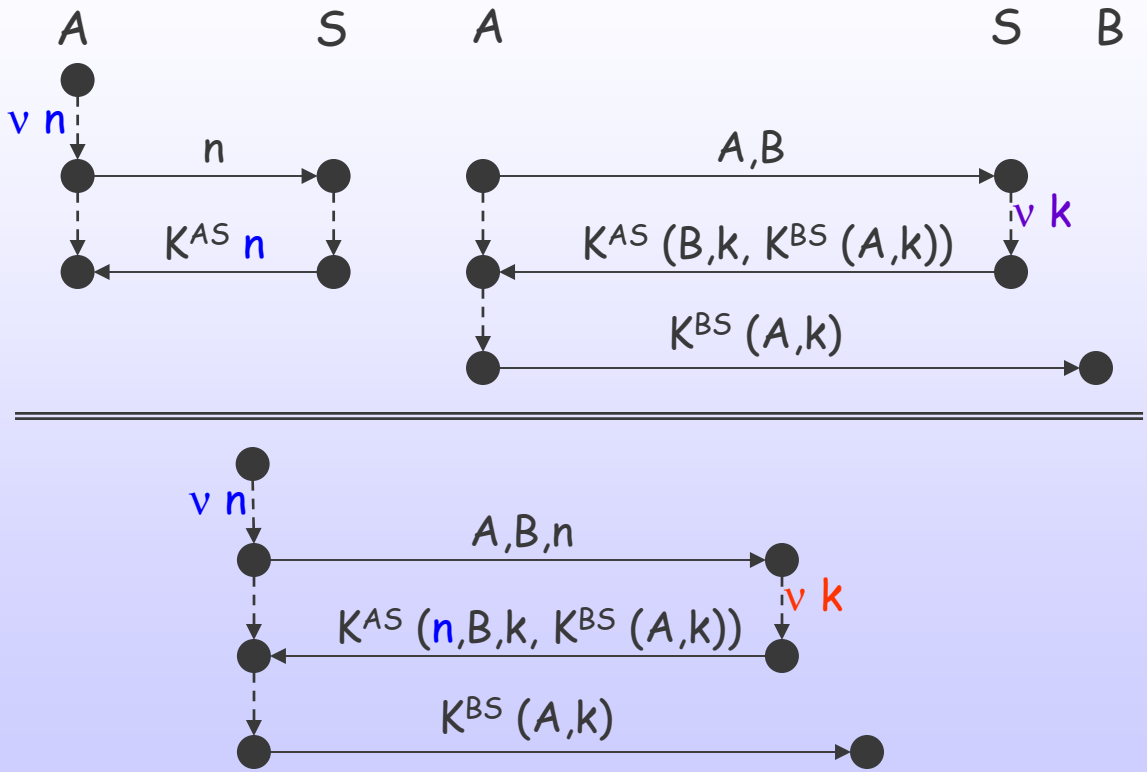


- Key confirmation
  - A/B knows that B/A has k
    - Essential for using k
  - Only B in KD^4 (under assumption)

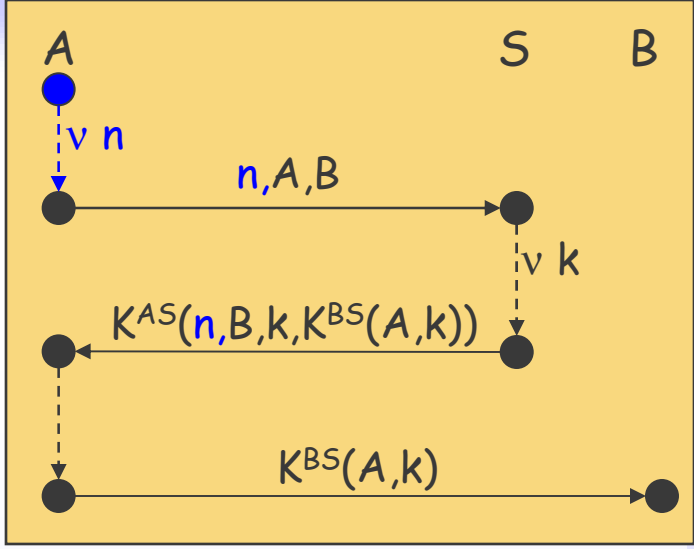


Recency with Nonces

- Use challenge-response as bracket



Core NSSK



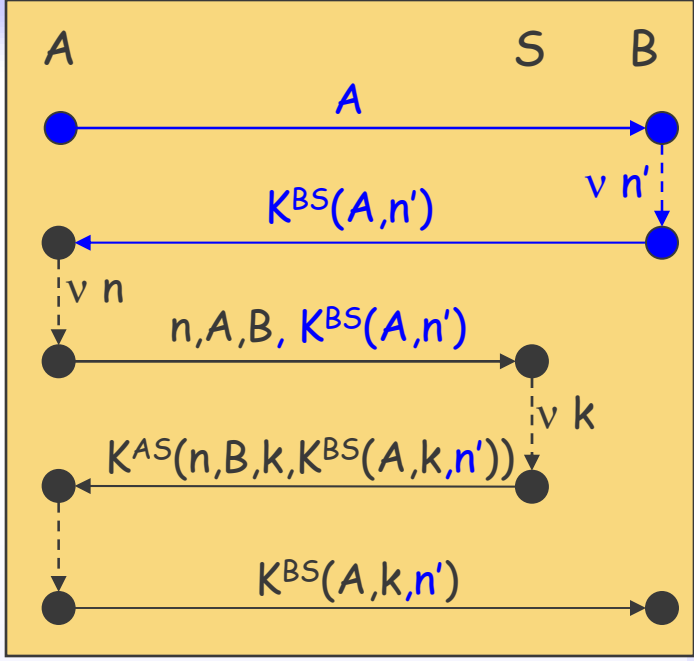
- $(\nu\ k)_S$ bounded by $(\nu\ n)_A$
- Ensures recency of k to A

A: secret(K^{AS}, [A,S]) & honest S &
$(\nu\ n)_A < \langle A,B,n\rangle_A < (K^{AS}\ (n,B,k,M))_A < (M)_A$

➔ $(\nu\ n)_A < \langle A,B,n\rangle_A <$
$(A,B,n)_S < (\nu\ k)_S < \langle K^{AS}\ (n,B,k,\ K^{BS}\ (A,k))\ \rangle_S <$
$(K^{AS}\ (n,B,k,\ K^{AS}\ (A,k)))_A < (K^{AS}\ (A,k))_A$

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

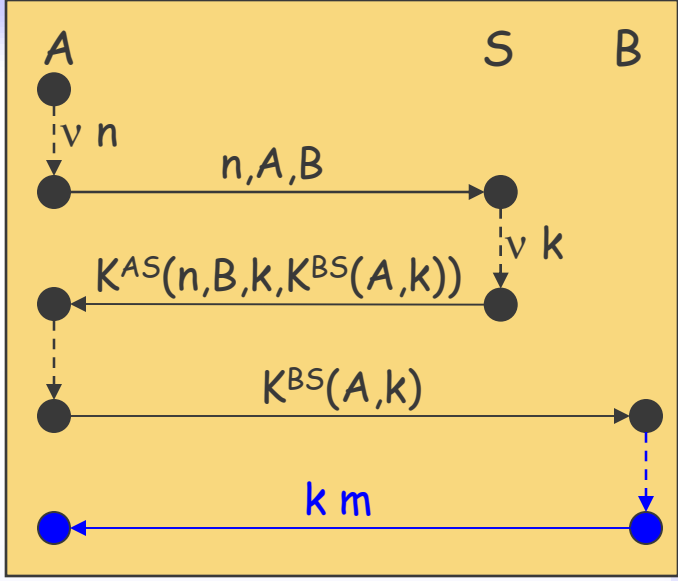
Core NSSK-fix

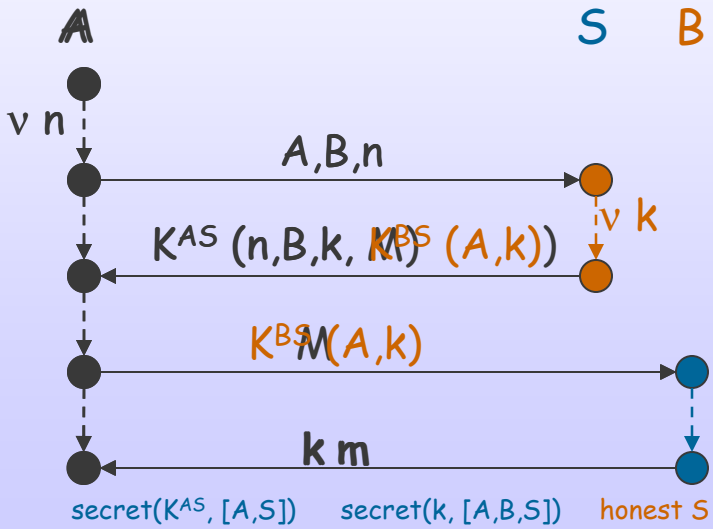


- Same device for B
  - Complications
    - Keeping *A* as initiator
    - Sending n′ to *A*
- Achieves recency for B too
  - Complicated

Key Confirmation

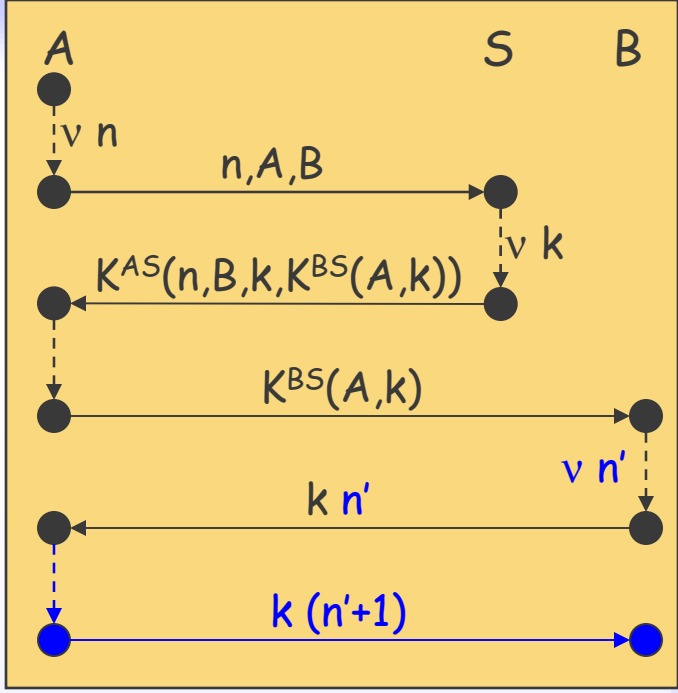
- B knows A has k
- B tells A he has k by sending agreed message





Extends to NSSK-fix

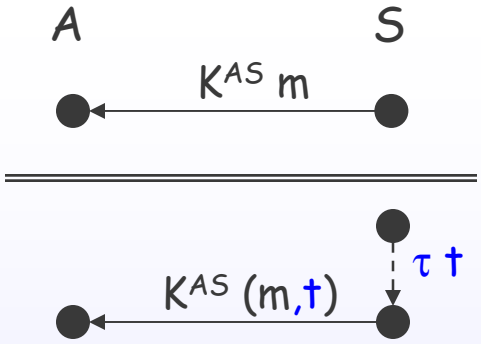
NSSK does more!

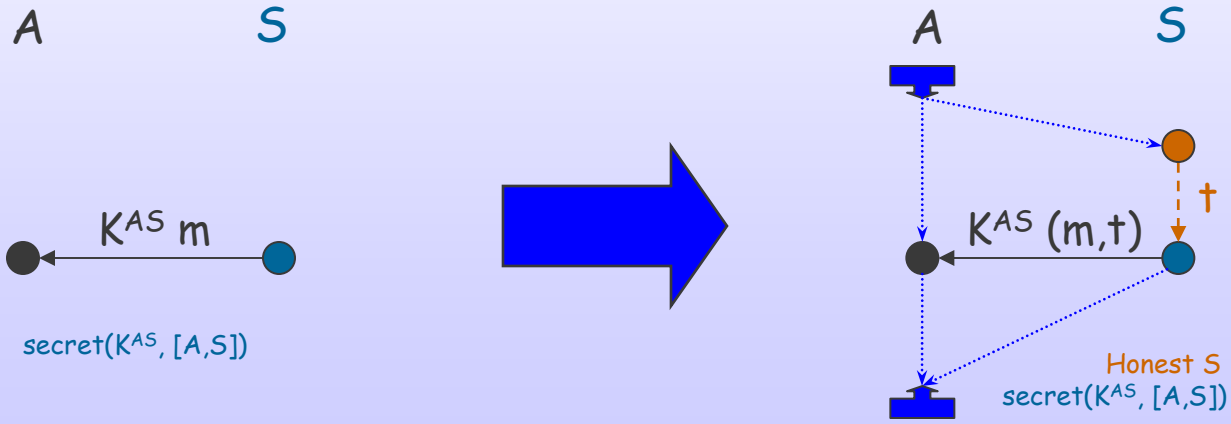


- B concludes with CR
  - k **not** confirmed to *A*
    - Unless tagging
  - B already knows *A* has k
- Exchange typical of repeated authentication
  - B repeatedly request service from *A*
    - … but *A* is initiator!
- Similarly for NSSK-fix

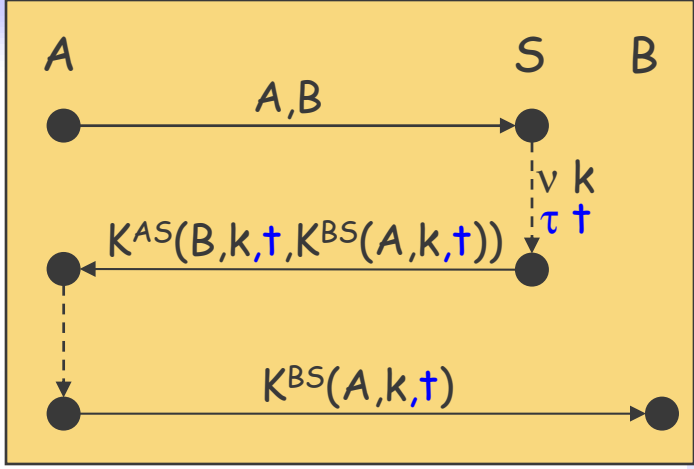
Recency with Timestamps

- Timestamp as bracketing device
  - Requires loosely synchronized clocks



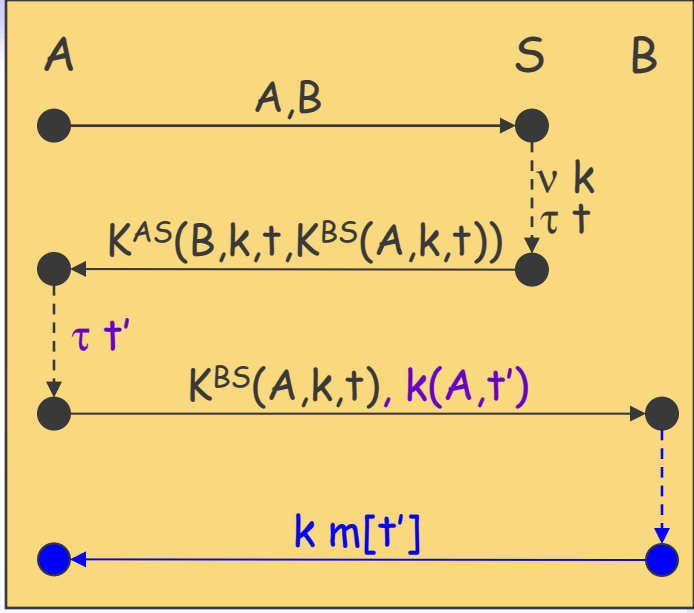


Denning-Sacco



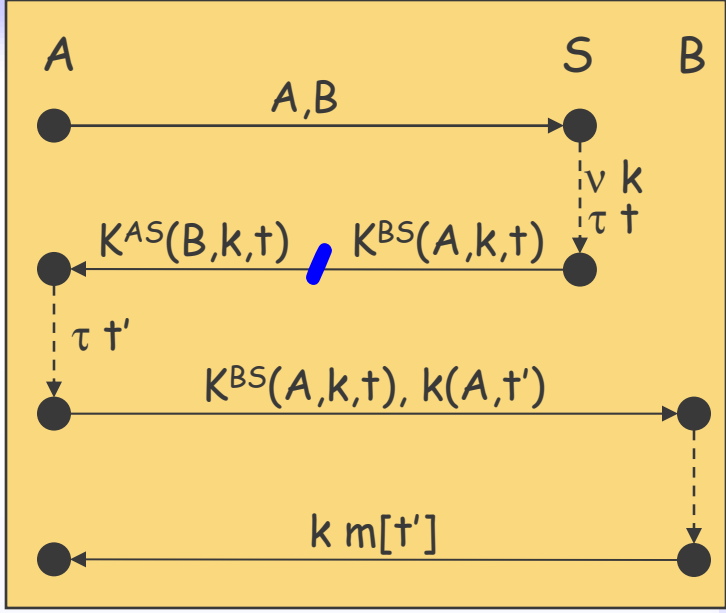
- Use timestamp for recency
  - Guarantee recency to both A and B
  - Same assurance as core NSSK-fix
    - Only 3 messages

Core Kerberos 4



- = NSSK
  - + key confirmation
  - + repeated authentication
    - Timestamp ensures recency of each new request
    - A is client and initiator
- Kerberos 4
  - ➢ 2 rounds of core Kerberos
  - ➢ Many more fields, options, …

Kerberos 5



- Same development but…
  - Start from concatenated variant of Denning Sacco

~~Future~~ Current 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