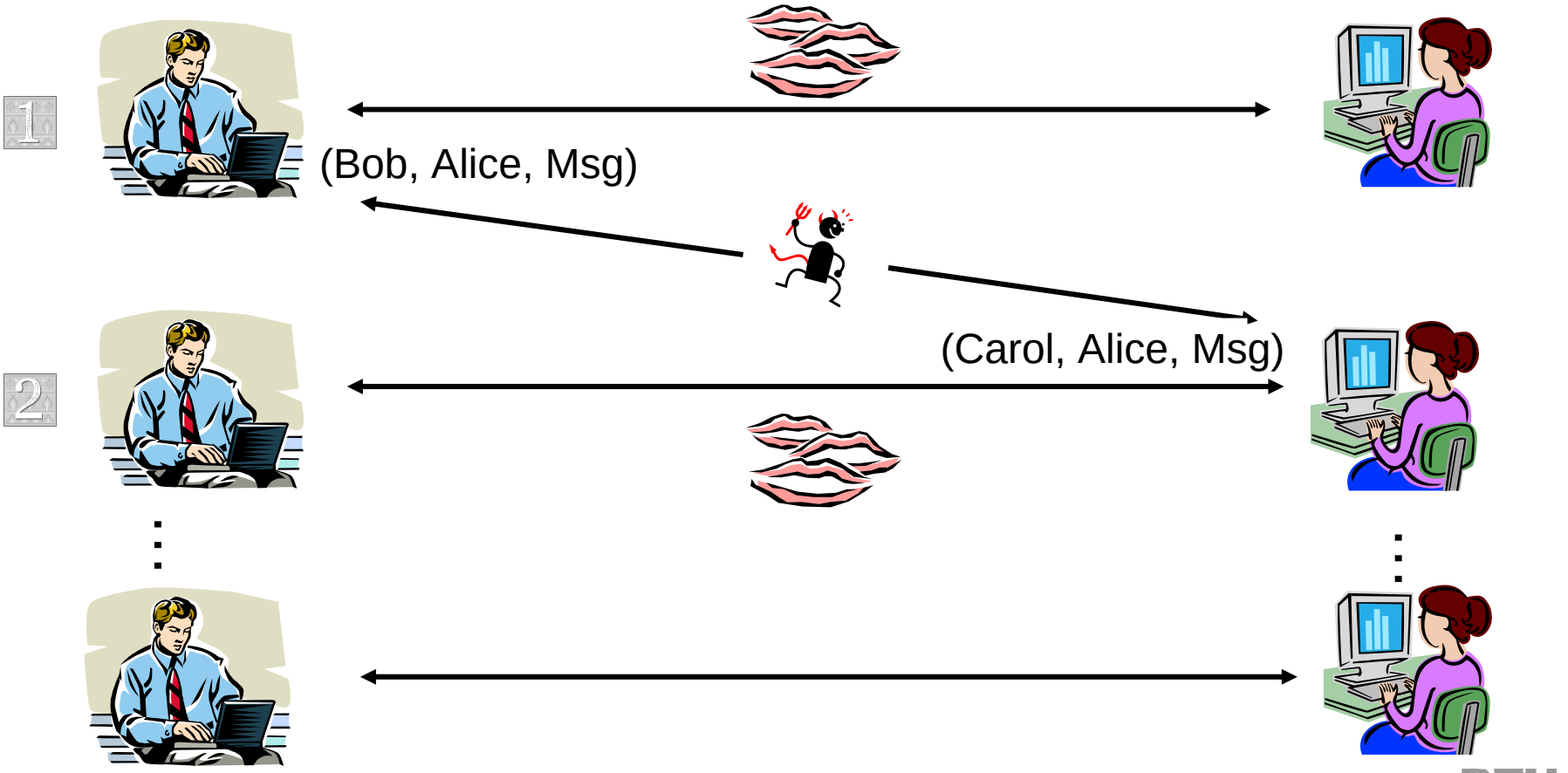
Detecting Replay Attacks by Freshness Annotations

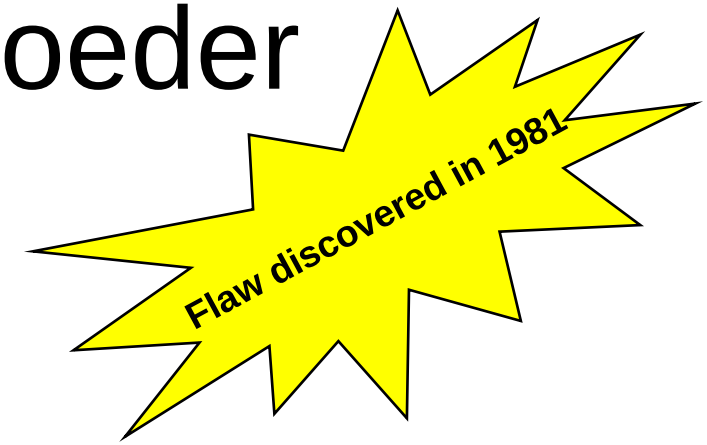
Han Gao[1],Pierpaolo Degano[2],
Chiara Bodei[2], Hanne Riis Nielson[1]

Informatics and Mathematics Modelling, Technical University of Denmark[1]
Department of Informatics, Pisa University[2]

Replay Attacks in Protocols



Needham-Schroeder



- Invented in 1978

1. $A \rightarrow S: \quad A, B, N_a$
2. $S \rightarrow A: \quad \{N_a, B, K, \{K, A\}_{K_b}\}_{K_a}$
3. $A \rightarrow B: \quad \{A, K\}_{K_b}$
4. $B \rightarrow A: \quad \{N_b\}_K$
5. $A \rightarrow B: \quad \{N_b - 1\}_K$
6. $A \rightarrow B: \quad \{Msg\}_K$

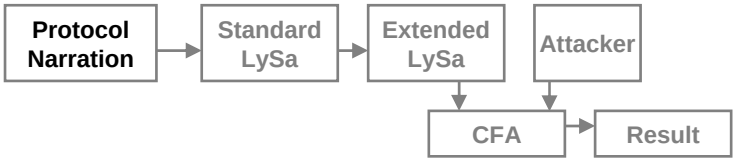
Key distribution steps:

The key should be known to both A and B

Authentication steps:

A and B make sure that they both know the key

Message exchange step



DTU

Needham-Schroeder

- The Denning-Sacco Attack

An old session key K' is leaked

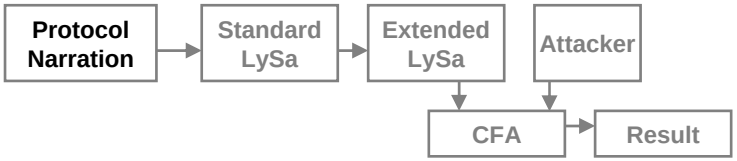
1. $A \rightarrow S : \quad A, B, N_a$
2. $S \rightarrow A : \quad \{N_a, B, K, \{K, A\}_{K_b}\}_{K_a}$
3. $A \rightarrow B : \quad \{A, K\}_{K_b}$
4. $B \rightarrow A : \quad \{N_b\}_K$
5. $A \rightarrow B : \quad \{N_b - 1\}_K$
6. $A \rightarrow B : \quad \{Msg\}_K$

1. $\ldots$
2. $\ldots$
3. $M(A) \rightarrow B : \quad \{A, K'\}_{K_b}$
4. $B \rightarrow M(A) : \quad \{N_b\}_{K'}$
5. $M(A) \rightarrow B : \quad \{N_b - 1\}_{K'}$
6. $M(A) \rightarrow B : \quad \{Msg\}_{K'}$

A is convinced that K is fresh

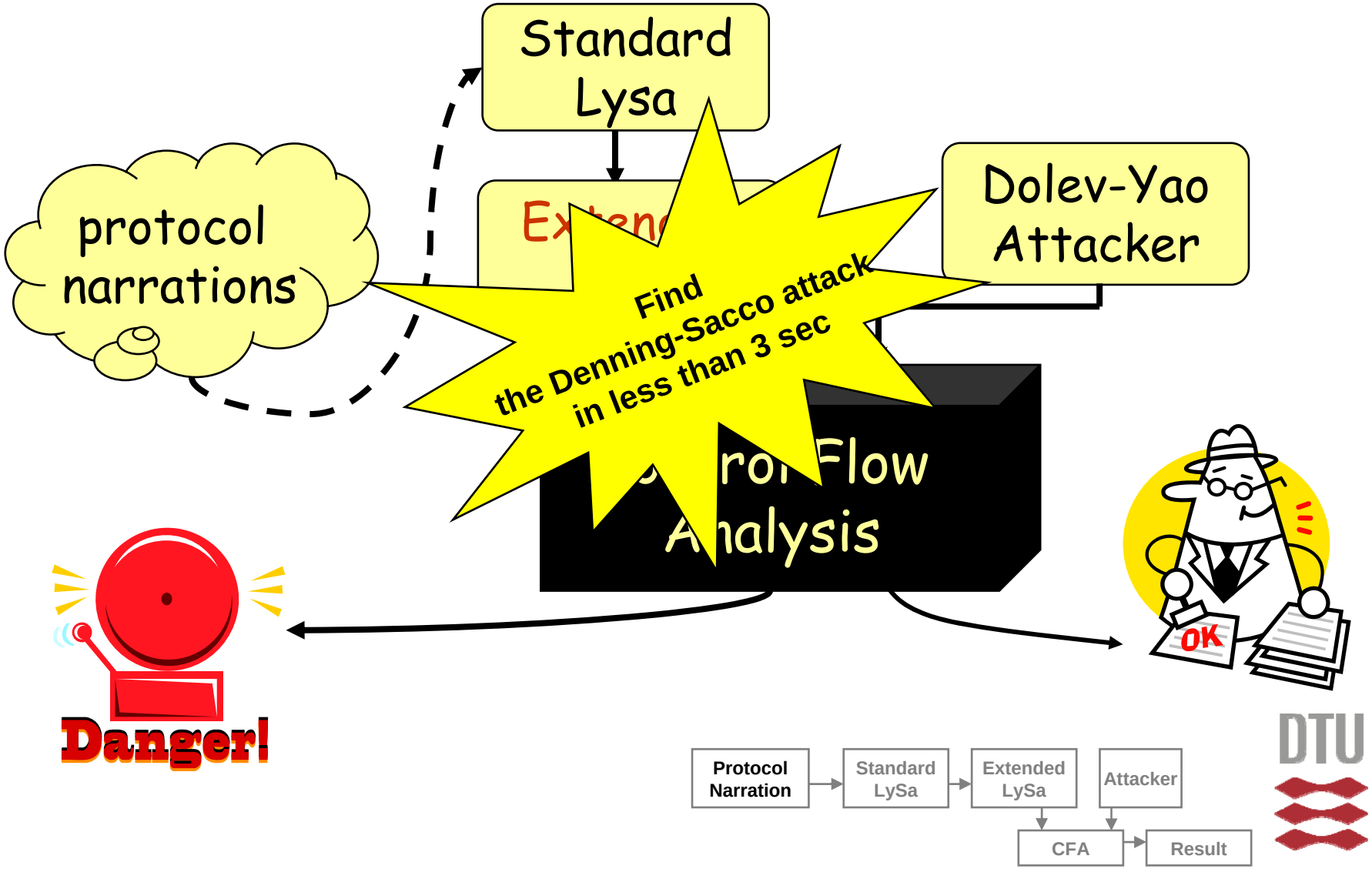
No such guarantee for B

B believes he is talking to A!



DTU

Whole Picture

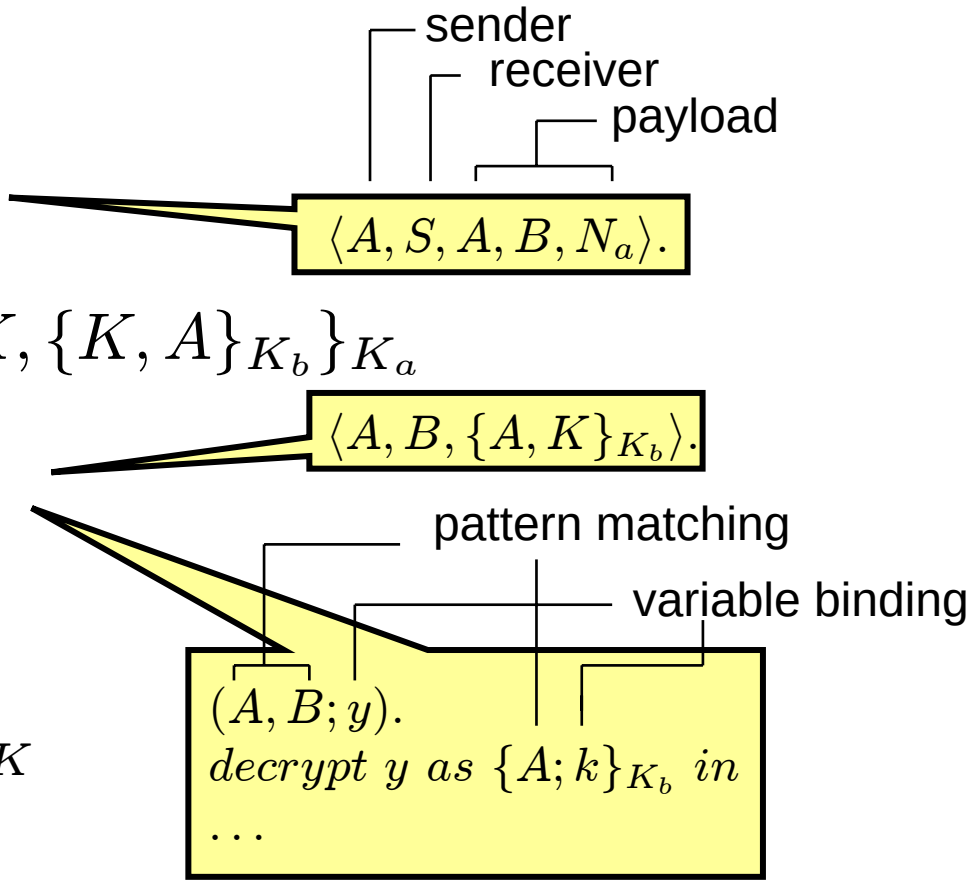


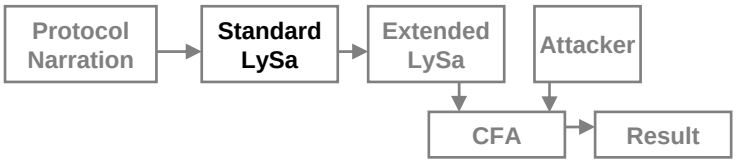
LySa Calculus

One global channel

1. $A \rightarrow S : \quad A, B, N_a$
2. $S \rightarrow A : \quad \{N_a, B, K, \{K, A\}_{K_b}\}_{K_a}$
3. $A \rightarrow B : \quad \{A, K\}_{K_b}$
4. $B \rightarrow A : \quad \{N_b\}_K$
5. $A \rightarrow B : \quad \{N_b - 1\}_K$
6. $A \rightarrow B : \quad \{Msg\}_K$

$$P = P_A \mid P_B \mid P_S$$





DTU

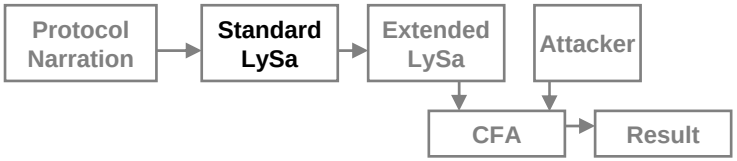
Session Identifiers

protocol run 1

$\mathcal{T}([\![\langle A, S, N_a \rangle.]\!]_1)$ ⟷ $\mathcal{T}([\![(A, S; x).]\!]_1)$

protocol run 2

$\mathcal{T}([\![\langle A, S, N_a \rangle.]\!]_2)$ ⟷ $\mathcal{T}([\![(A, S; x).]\!]_2)$



DTU

Extended LySa Calculus

Stops when reaching n or x

	Standard		Extended
Terms	E	$\mathcal{F}$ →	$\mathcal{E}$
Processes	P	$\mathcal{T}$ →	$\mathcal{P}$

Stops when reaching 0 or !

$$\mathcal{F}([\{N\}_K]_s) = \{[N]_s\}_{[K]_s}$$

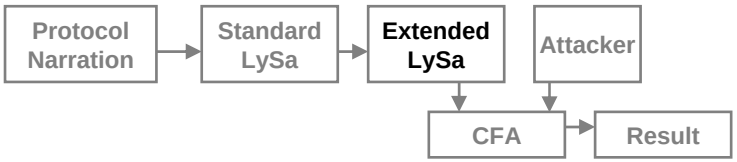
$$\mathcal{T}([\langle N\rangle.0 \mid !((;x).0)]_s) =$$
$$\mathcal{T}([\langle N\rangle.0]_s) \mid \mathcal{T}([!((;x).0)]_s) =$$
$$\langle [N_s]\rangle.0 \mid [!((;x).0)]_s$$

1. $A \to S : \quad A, B, N_a$ — $\langle A, S, A, B, N_a\rangle.$
2. $S \to A : \quad \{N_a, B, K, \{K, A\}_{K_b}\}_{K_a}$
3. $A \to B : \quad \{A, K\}_{K_b}$ — $\langle A, B, \{A, K\}_{K_b}\rangle.$
4. $B \to A : \quad \{N_b\}_K$ — $(A, B; y).$ *decrypt* y *as* $\{A; k\}_{K_b}$ *in* $\cdots$
5. $A \to B : \quad \{N_b - 1\}_K$
6. $A \to B : \quad \{Msg\}_K$

$$P = P_A \mid P_B \mid P_S$$

$$\mathcal{P} = [!P]_0$$

Unfold once in each semantics step



DTU

Freshness Property

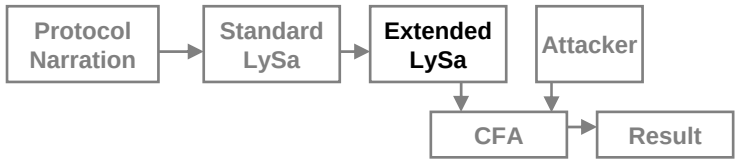
Equality with sessin IDs ingnored

Extract the session ID

$$\frac{\mathcal{E}_0 \approx \mathcal{E}'_0 \ \wedge \ \mathcal{E}_1 \approx \mathcal{E}'_1 \ \wedge \ \mathcal{R}((\mathcal{I}(\mathcal{E}_0), \mathcal{I}(\mathcal{E}'_0)), (\mathcal{I}(\mathcal{E}_1), \mathcal{I}(\mathcal{E}'_1)))}{decrypt\ [\{\mathcal{E}_1, \mathcal{E}_2\}_{\mathcal{E}_0}]_s\ as\ \{\mathcal{E}'_1; x_2\}_{\mathcal{E}'_0}\ in\ \mathcal{P} \xrightarrow{\{\mathcal{E}_1, \mathcal{E}_2\}_{\mathcal{E}_0}}_{\mathcal{R}} \mathcal{P}[\mathcal{E}_2/x_2]}$$

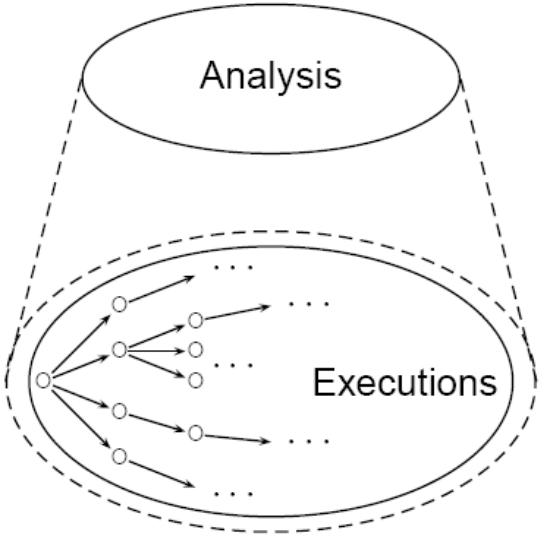
$$decrypt\ \{[N_a]_1, [N_b]_1\}_{[K]_1}\ as\ \{[N_a]_1; x\}_{[K]_1}\ in\ 0$$

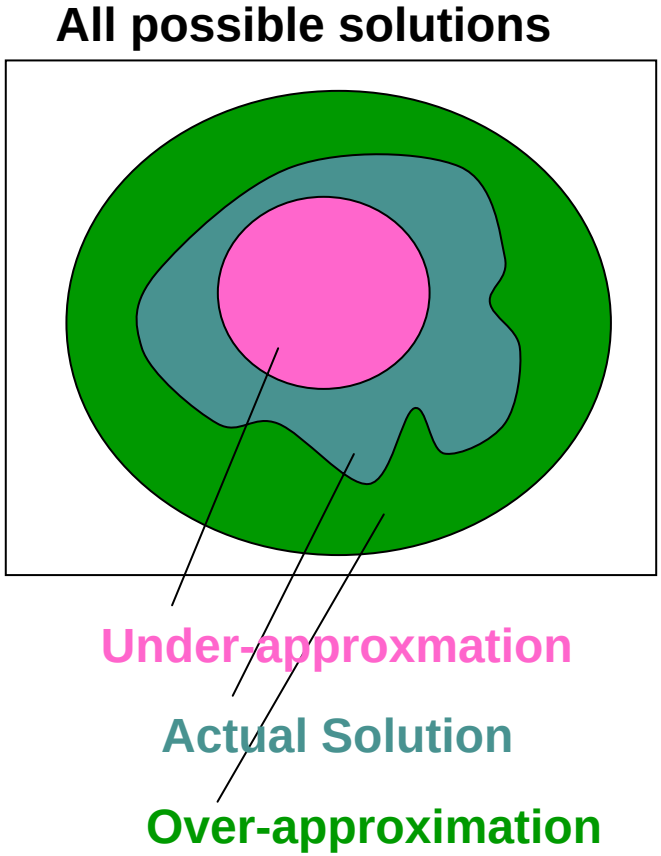
$$decrypt\ \{[N_a]_2, [N_b]_2\}_{[K]_2}\ as\ \{[N_a]_1; x\}_{[K]_1}\ in\ 0$$

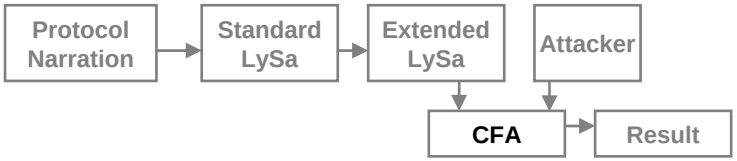


Static Analysis

- Approximation
  - Over-Approximation
- Algorithms
  - Control Flow Analysis



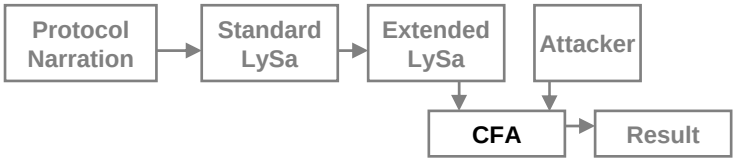




Static Analysis

- Analysis of Terms $\rho \models \mathcal{E} : \vartheta$
  - Determine the possible values that each term may evaluate to
- Analysis of Processes $\rho, \kappa \models_{\mathsf{RM}} \mathcal{P} : \psi$
  - Collect the values that may flow on the network
  - Error component

$$\frac{\text{analysis}(\mathcal{T}([P]_0)) \mid \text{analysis}(\mathcal{T}([P]_1))}{\text{analysis}(\mathcal{P})}$$



DTU

The Attacker

- Capabilities
  - Eavesdrop
  - Alter
  - Insider or outsider or both
  - Obtain old session keys

Analysis of Needham-Schroeder

1. $A \rightarrow S : \quad A, B, N_a$
2. $S \rightarrow A : \quad \{N_a, B, K, \{K, A\}_{K_b}\}_{K_a}$
3. $A \rightarrow B : \quad \{A, K\}_{K_b}$
4. $B \rightarrow A : \quad \{N_b\}_K$
5. $A \rightarrow B : \quad \{N_b - 1\}_K$
6. $A \rightarrow B : \quad \{Msg\}_K$

$\langle A, B, \{A, K\}_{K_b} \rangle.$

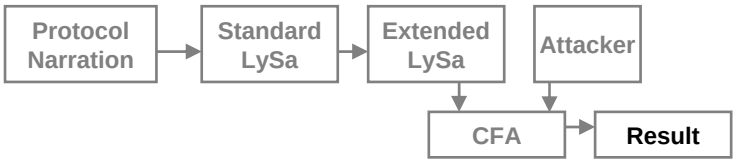
$(A, B; y).$
$decrypt\ y\ as\ \{A; k\}_{K_b}\ in$
$\ldots$

$$P = P_A \mid P_B \mid P_S$$
$$\mathcal{P} = [!P]_0$$

$$\frac{\text{analysis}(\mathcal{T}([P]_0)) \mid \text{analysis}(\mathcal{T}([P]_1))}{\text{analysis}(\mathcal{P})}$$

Session 0 $\mathcal{T}([\langle A, B, \{A, K\}_{K_b} \rangle]_0) \longrightarrow \mathcal{T}\ ([(A, B, y).\ decrypt\ y\ as\ \{A; k\}_{K_b}\ in]_0)$

Session 1 $\mathcal{T}([\langle A, B, \{A, K\}_{K_b} \rangle]_1) \longrightarrow \mathcal{T}\ ([(A, B, y).\ decrypt\ y\ as\ \{A; k\}_{K_b}\ in]_1)$



Conclusion

- Simply process calculus with cryptographic primitives for modelling security protocols
- Automatic algorithm for providing security assurances for protocols
  - Semantics correct and sound
- Implementation has been used to validate a number of protocols

DTU

Thank You!