Access Control and Privacy Policies (6)

Email: christian.urban at kcl.ac.uk
Office: S1.27 (1st floor Strand Building)
Slides: KEATS (also homework is there)

Access Control Logic

Formulas

$$
\begin{array}{lcl}
F & ::= & \text{true} \\
  & \mid & \text{false} \\
  & \mid & F \wedge F \\
  & \mid & F \vee F \\
  & \mid & F \Rightarrow F \\
  & \mid & p\,(t_1,\ldots,t_n) \\
  & \mid & P\ \text{says}\ F
\end{array}
$$

"saying predicate"

Judgements

$$\Gamma \vdash F$$

Judgements

$$\Gamma \vdash \boldsymbol{F}$$

Judgements

entails sign

$$\Gamma \vdash \boldsymbol{F}$$

a single formula

Gamma
stands for a collection of formulas
("assumptions")

Inference Rules

premisses

$$\frac{\Gamma \vdash \boldsymbol{F}_1 \quad \Gamma \vdash \boldsymbol{F}_2}{\Gamma \vdash \boldsymbol{F}_1 \wedge \boldsymbol{F}_2}$$

conclusion

Inference Rules

premisses

$$\frac{\Gamma \vdash \boldsymbol{F}_1 \quad \Gamma \vdash \boldsymbol{F}_2}{\Gamma \vdash \boldsymbol{F}_1 \wedge \boldsymbol{F}_2}$$

conclusion

$\boldsymbol{P}$ says $\boldsymbol{F} \vdash \boldsymbol{Q}$ says $\boldsymbol{F} \wedge \boldsymbol{P}$ says $\boldsymbol{G}$

Inference Rules

premisses

$$\frac{\Gamma \vdash \boldsymbol{F}_1 \qquad \Gamma \vdash \boldsymbol{F}_2}{\Gamma \vdash \boldsymbol{F}_1 \wedge \boldsymbol{F}_2}$$

conclusion

$$\underbrace{\boldsymbol{P} \text{ says } \boldsymbol{F}}_{\Gamma} \vdash \underbrace{\boldsymbol{Q} \text{ says } \boldsymbol{F}}_{\boldsymbol{F}_1} \wedge \underbrace{\boldsymbol{P} \text{ says } \boldsymbol{G}}_{\boldsymbol{F}_2}$$

Inference Rules

$$\frac{}{\Gamma, F \vdash F}$$

$$\frac{\Gamma \vdash F_1 \Rightarrow F_2 \quad \Gamma \vdash F_1}{\Gamma \vdash F_2} \qquad \frac{F_1, \Gamma \vdash F_2}{\Gamma \vdash F_1 \Rightarrow F_2}$$

$$\frac{\Gamma \vdash F}{\Gamma \vdash P \text{ says } F}$$

$$\frac{\Gamma \vdash P \text{ says } (F_1 \Rightarrow F_2) \quad \Gamma \vdash P \text{ says } F_1}{\Gamma \vdash P \text{ says } F_2}$$

Sending Messages

- Alice sends a message m

  Alice says m

Sending Messages

- Alice sends a message m

$$\text{Alice says } m$$

- Alice sends an encrypted message m with key K ($\{m\}_K \stackrel{\text{def}}{=} K \Rightarrow m$)

$$\text{Alice says } \{m\}_K$$

Sending Messages

- Alice sends a message m

$$\text{Alice says } m$$

- Alice sends an encrypted message m with key K $(\{m\}_K \stackrel{\text{def}}{=} K \Rightarrow m)$

$$\text{Alice says } \{m\}_K$$

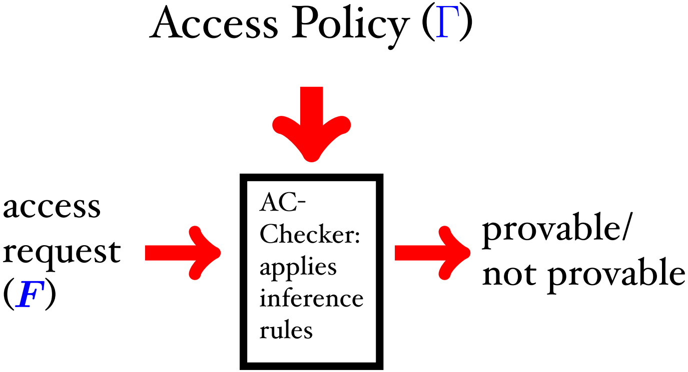
- Decryption of Alice's message

$$\frac{\Gamma \vdash \text{Alice says } \{m\}_K \quad \Gamma \vdash \text{Alice says } K}{\Gamma \vdash \text{Alice says } m}$$

Proofs

$$\Gamma \vdash \boldsymbol{F}$$

The Access Control Problem



Proofs

$$\frac{}{\vdash}\ \text{axiom}$$

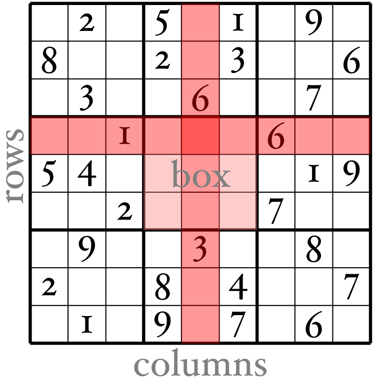
$$\frac{\vdash}{\vdash}$$

$$\frac{\vdash \qquad \vdash}{\vdash}$$

goal

start

Sudoku



1. **Row-Column:** each cell, must contain exactly one number
2. **Row-Number:** each row must contain each number exactly once
3. **Column-Number:** each column must contain each number exactly once
4. **Box-Number:** each box must contain each number exactly once

Solving Sudokus

			7				5	8
	5	6	2	1	8	7	9	3
						1		
							8	1
			3	7	6			
9	6							
		5						
		4		2	1	8	3	
8	7				3			

single position rules

$$\frac{\{1..9\} - \{4\} \text{ in one row}}{4 \text{ in empty position}}$$

Solving Sudokus

			7				5	8
	5	6	2	1	8	7	9	3
						1		
							8	1
			3	7	6			
9	6							
		5						
		4		2	1	8	3	
8	7				3			

single position rules

$$\frac{\{1..9\} - \{4\} \text{ in one row}}{4 \text{ in empty position}}$$

$$\frac{\{1..9\} - \{x\} \text{ in one column}}{x \text{ in empty position}}$$

$$\frac{\{1..9\} - \{x\} \text{ in one box}}{x \text{ in empty position}}$$

Solving Sudokus

			7			2	5	8
	5	6	2	1	8	7	9	3
						1	2	2
							8	1
			3	7	6			
9	6							
		5						
		4		2	1	8	3	
8	7				3			

candidate rules

$$\frac{X - \{x\} \text{ in one box} \quad X \subseteq \{1..9\}}{x \text{ candidate in empty positions}}$$

Solving Sudokus

			7			2	5	8
4	5	6	2	1	8	7	9	3
						1	2	2
							8	1
			3	7	6			
9	6							
		5						
		4		2	1	8	3	
8	7				3			

$$\frac{\{1..9\} - \{4\} \text{ in one row}}{4 \text{ in empty position}}$$

$$\frac{X - \{2\} \text{ in one box} \quad X \subseteq \{1..9\}}{2 \text{ candidate in empty positions}}$$

Solving Sudokus

			7			2	5	8
4	5	6	2	1	8	7	9	3
						1	2	2
							8	1
			3	7	6			
9	6							
		5						
		4		2	1	8	3	
8	7				3			

$$\frac{\{1..9\} - \{4\} \text{ in one row}}{4 \text{ in empty position}}$$

$$\frac{X - \{2\} \text{ in one box} \quad X \subseteq \{1..9\}}{2 \text{ candidate in empty positions}}$$

Solving Sudokus

			7				5	8
	5	6	2	1	8	7	9	3
						1		
							8	1
			3	7	6			
9	6							2
		5						
		4		2	1	8	3	
8	7				3			

$$\frac{\boldsymbol{X} - \{2\} \text{ in one box} \quad \boldsymbol{X} \subseteq \{1..9\}}{2 \text{ candidate}}$$

BTW

Are there sudokus that cannot be solved?

BTW

Are there sudokus that cannot be solved?

1	2	3	4	5	6	7	8	
								2
								3
								4
								5
								6
								7
								8
								9

Sometimes no rules apply at all....unsolvable sudoku.

Example Proof

$$\frac{?}{P \text{ says } F_1 \wedge Q \text{ says } F_2 \vdash Q \text{ says } F_2 \wedge P \text{ says } F_1}$$

Example Proof

We have (by axiom)

(1) $P \text{ says } F_1 \wedge Q \text{ says } F_2 \vdash P \text{ says } F_1 \wedge Q \text{ says } F_2$

From (1) we get

(2) $P \text{ says } F_1 \wedge Q \text{ says } F_2 \vdash P \text{ says } F_1$
(3) $P \text{ says } F_1 \wedge Q \text{ says } F_2 \vdash Q \text{ says } F_2$

From (3) and (2) we get

$$P \text{ says } F_1 \wedge Q \text{ says } F_2 \vdash Q \text{ says } F_2 \wedge P \text{ says } F_1$$

Done.

Other Direction

We want to prove

$$P \text{ says } F_1 \wedge Q \text{ says } F_2 \vdash Q \text{ says } F_2 \wedge P \text{ says } F_1$$

We better be able to prove:

(1) $P \text{ says } F_1 \wedge Q \text{ says } F_2 \vdash Q \text{ says } F_2$

(2) $P \text{ says } F_1 \wedge Q \text{ says } F_2 \vdash P \text{ says } F_1$

For (1): If we can prove

$$P \text{ says } F_1 \wedge Q \text{ says } F_2 \vdash Q \text{ says } F_2 \wedge P \text{ says } F_1$$

then (1) is fine. Similarly for (2).

I want to prove

$$\Gamma \vdash \text{del\_file}$$

I want to prove

$$\Gamma \vdash \text{del\_file}$$

There is an inference rule

$$\frac{\Gamma \vdash \boldsymbol{F}}{\Gamma \vdash \boldsymbol{P}\ \text{says}\ \boldsymbol{F}}$$

I want to prove

$$\Gamma \vdash \text{del\_file}$$

There is an inference rule

$$\frac{\Gamma \vdash \boldsymbol{F}}{\Gamma \vdash \boldsymbol{P} \text{ says } \boldsymbol{F}}$$

So I can derive $\Gamma \vdash \text{Alice says del\_file}$.

I want to prove

$$\Gamma \vdash \text{del\_file}$$

There is an inference rule

$$\frac{\Gamma \vdash \boldsymbol{F}}{\Gamma \vdash \boldsymbol{P} \text{ says } \boldsymbol{F}}$$

So I can derive $\Gamma \vdash$ Alice says del_file.

Γ contains already Alice says del_file.
So I can use the rule

$$\frac{}{\Gamma, \boldsymbol{F} \vdash \boldsymbol{F}}$$

Done. Qed.

I want to prove

$$\Gamma \vdash \text{del\_file}$$

There is an inference rule

$$\frac{\Gamma \vdash \boldsymbol{F}}{\Gamma \vdash \boldsymbol{P} \text{ says } \boldsymbol{F}}$$

So I can derive $\Gamma \vdash$ Alice says del_file.

Γ contains already Alice says del_file.
So I can use the rule

$$\overline{\Gamma, \boldsymbol{F} \vdash \boldsymbol{F}}$$

What is wrong with this? **Done. Qed.**

Program

How to prove $\Gamma \vdash \boldsymbol{F}$?

$$\overline{\Gamma, \boldsymbol{F} \vdash \boldsymbol{F}}$$

$$\frac{\boldsymbol{F}_1, \Gamma \vdash \boldsymbol{F}_2}{\Gamma \vdash \boldsymbol{F}_1 \Rightarrow \boldsymbol{F}_2}$$

$$\frac{\Gamma \vdash \boldsymbol{F}}{\Gamma \vdash \boldsymbol{P}\ \text{says}\ \boldsymbol{F}}$$

$$\frac{\Gamma \vdash \boldsymbol{F}_1}{\Gamma \vdash \boldsymbol{F}_1 \vee \boldsymbol{F}_2} \qquad \frac{\Gamma \vdash \boldsymbol{F}_2}{\Gamma \vdash \boldsymbol{F}_1 \vee \boldsymbol{F}_2}$$

$$\frac{\Gamma \vdash \boldsymbol{F}_1 \quad \Gamma \vdash \boldsymbol{F}_2}{\Gamma \vdash \boldsymbol{F}_1 \wedge \boldsymbol{F}_2}$$

Program: `prove2`

I want to prove $\Gamma \vdash \text{Pred}$

Program: `prove2`

I want to prove $\Gamma \vdash \text{Pred}$

1. I found that Γ contains the assumption $\boldsymbol{F}_1 \Rightarrow \boldsymbol{F}_2$

Program: prove2

I want to prove $\Gamma \vdash \text{Pred}$

1. I found that Γ contains the assumption $\boldsymbol{F}_1 \Rightarrow \boldsymbol{F}_2$
2. If I can prove $\Gamma \vdash \boldsymbol{F}_1$,

Program: prove2

I want to prove $\Gamma \vdash \text{Pred}$

1. I found that Γ contains the assumption $\boldsymbol{F}_1 \Rightarrow \boldsymbol{F}_2$
2. If I can prove $\Gamma \vdash \boldsymbol{F}_1$, then I can prove

$$\Gamma \vdash \boldsymbol{F}_2$$

$$\frac{\Gamma \vdash \boldsymbol{F}_1 \Rightarrow \boldsymbol{F}_2 \quad \Gamma \vdash \boldsymbol{F}_1}{\Gamma \vdash \boldsymbol{F}_2}$$

Program: `prove2`

I want to prove $\Gamma \vdash \mathrm{Pred}$

1. I found that Γ contains the assumption $\boldsymbol{F}_1 \Rightarrow \boldsymbol{F}_2$
2. If I can prove $\Gamma \vdash \boldsymbol{F}_1$, then I can prove
$$\Gamma \vdash \boldsymbol{F}_2$$
3. So I am able to try to prove $\Gamma \vdash \mathrm{Pred}$ with the additional assumption $\boldsymbol{F}_2$.

$$\boldsymbol{F}_2, \Gamma \vdash \mathrm{Pred}$$

Recall the following scenario:

- If Admin says that file should be deleted, then this file must be deleted.
- Admin trusts Bob to decide whether file should be deleted.
- Bob wants to delete file.

$$\Gamma = \begin{array}{l} (\text{Admin says del\_file}) \Rightarrow \text{del\_file}, \\ (\text{Admin says } ((\text{Bob says del\_file}) \Rightarrow \text{del\_file})), \\ \text{Bob says del\_file} \end{array}$$

$$\Gamma \vdash \text{del\_file}$$

- P is entitled to do F
  $P \text{ controls } F \overset{\text{def}}{=} (P \text{ says } F) \Rightarrow F$

$$\frac{\Gamma \vdash P \text{ controls } F \quad \Gamma \vdash P \text{ says } F}{\Gamma \vdash F}$$

Trusted Third Party

Simple protocol for establishing a secure connection via a mutually trusted 3rd party (server):

Message 1	$A \rightarrow S : A, B$
Message 2	$S \rightarrow A : \{K_{AB}\}_{K_{AS}}$ and $\{\{K_{AB}\}_{K_{BS}}\}_{K_{AS}}$
Message 3	$A \rightarrow B : \{K_{AB}\}_{K_{BS}}$
Message 4	$A \rightarrow B : \{m\}_{K_{AB}}$

Sending Rule

$$\frac{\Gamma \vdash P \;\text{says}\; F \quad \Gamma \vdash P \;\text{sends}\; Q : F}{\Gamma \vdash Q \;\text{says}\; F}$$

Sending Rule

$$\frac{\Gamma \vdash \boldsymbol{P} \text{ says } \boldsymbol{F} \quad \Gamma \vdash \boldsymbol{P} \text{ sends } \boldsymbol{Q} : \boldsymbol{F}}{\Gamma \vdash \boldsymbol{Q} \text{ says } \boldsymbol{F}}$$

$$\boldsymbol{P} \text{ sends } \boldsymbol{Q} : \boldsymbol{F} \overset{\text{def}}{=} (\boldsymbol{P} \text{ says } \boldsymbol{F}) \Rightarrow (\boldsymbol{Q} \text{ says } \boldsymbol{F})$$

Trusted Third Party

$$
\begin{aligned}
&A \text{ sends } S : \text{Connect}(A, B)\\
&S \text{ says } (\text{Connect}(A, B) \Rightarrow\\
&\qquad\qquad \{K_{AB}\}_{K_{AS}} \wedge \{\{K_{AB}\}_{K_{BS}}\}_{K_{AS}})\\
&S \text{ sends } A : \{K_{AB}\}_{K_{AS}} \wedge \{\{K_{AB}\}_{K_{BS}}\}_{K_{AS}}\\
&A \text{ sends } B : \{K_{AB}\}_{K_{BS}}\\
&A \text{ sends } B : \{m\}_{K_{AB}}
\end{aligned}
$$

Trusted Third Party

$$
\begin{array}{l}
A \text{ sends } S : \text{Connect}(A, B)\\
S \text{ says } (\text{Connect}(A, B) \Rightarrow\\
\qquad\qquad \{K_{AB}\}_{K_{AS}} \wedge \{\{K_{AB}\}_{K_{BS}}\}_{K_{AS}})\\
S \text{ sends } A : \{K_{AB}\}_{K_{AS}} \wedge \{\{K_{AB}\}_{K_{BS}}\}_{K_{AS}}\\
A \text{ sends } B : \{K_{AB}\}_{K_{BS}}\\
A \text{ sends } B : \{m\}_{K_{AB}}
\end{array}
$$

$\Gamma \vdash B \text{ says } m$?

Public/Private Keys

- Bob has a private and public key: $K_{Bob}^{pub}, K_{Bob}^{priv}$

$$\frac{\Gamma \vdash \text{Alice says } \{m\}_{K_{Bob}^{pub}} \quad \Gamma \vdash K_{Bob}^{priv}}{\Gamma \vdash \text{Alice says } m}$$

Public/Private Keys

- Bob has a private and public key: K_{Bob}^{pub}, K_{Bob}^{priv}

$$\frac{\Gamma \vdash \text{Alice says } \{m\}_{K_{Bob}^{pub}} \quad \Gamma \vdash K_{Bob}^{priv}}{\Gamma \vdash \text{Alice says } m}$$

- this is **not** a derived rule!

Security Levels

- Top secret (TS)
- Secret (S)
- Public (P)

$$slev(P) < slev(S) < slev(TS)$$

Security Levels

- Top secret (TS)
- Secret (S)
- Public (P)

$$slev(P) < slev(S) < slev(TS)$$

- Bob has a clearance for "secret"
- Bob can read documents that are public or sectret, but not top secret

Reading a File

$$\frac{\text{Bob controls Permitted (File, read)} \qquad \text{Bob says Permitted (File, read)}}{\text{Permitted (File, read)}}$$

Reading a File

$$\frac{\begin{array}{l} \textit{slev}(\text{File}) < \textit{slev}(\text{Bob}) \Rightarrow \\ \qquad\qquad \text{Bob controls Permitted (File, read)} \\ \text{Bob says Permitted (File, read)} \\ \textit{slev}(\text{File}) < \textit{slev}(\text{Bob}) \end{array}}{\text{Permitted (File, read)}}$$

Reading a File

$$\frac{\begin{array}{l} slev(\text{File}) < slev(\text{Bob}) \Rightarrow \\ \qquad\qquad \text{Bob controls Permitted (File, read)} \\ \text{Bob says Permitted (File, read)} \\ slev(\text{File}) = P \\ slev(\text{Bob}) = S \\ slev(P) < slev(S) \end{array}}{\text{Permitted (File, read)}}$$

Substitution Rule

$$\frac{\Gamma \vdash \boldsymbol{slev}(\boldsymbol{P}) = \boldsymbol{l}_1 \quad \Gamma \vdash \boldsymbol{slev}(\boldsymbol{Q}) = \boldsymbol{l}_2 \quad \Gamma \vdash \boldsymbol{l}_1 < \boldsymbol{l}_2}{\Gamma \vdash \boldsymbol{slev}(\boldsymbol{P}) < \boldsymbol{slev}(\boldsymbol{Q})}$$

Substitution Rule

$$\frac{\Gamma \vdash slev(P) = l_1 \quad \Gamma \vdash slev(Q) = l_2 \quad \Gamma \vdash l_1 < l_2}{\Gamma \vdash slev(P) < slev(Q)}$$

- $slev(\mathrm{Bob}) = S$
- $slev(\mathrm{File}) = P$
- $slev(P) < slev(S)$

Reading a File

$$\frac{\begin{array}{l} slev(\text{File}) < slev(\text{Bob}) \Rightarrow \\ \qquad\qquad \text{Bob controls Permitted (File, read)} \\ \text{Bob says Permitted (File, read)} \\ slev(\text{File}) = P \\ slev(\text{Bob}) = TS \\ ? \end{array}}{\text{Permitted (File, read)}}$$

Reading a File

$$\frac{\begin{array}{l} slev(\text{File}) < slev(\text{Bob}) \Rightarrow \\ \qquad\qquad \text{Bob controls Permitted (File, read)} \\ \text{Bob says Permitted (File, read)} \\ slev(\text{File}) = P \\ slev(\text{Bob}) = TS \\ slev(P) < slev(S) \\ slev(S) < slev(TS) \end{array}}{\text{Permitted (File, read)}}$$

Transitivity Rule

$$\frac{\Gamma \vdash l_1 < l_2 \qquad \Gamma \vdash l_2 < l_3}{\Gamma \vdash l_1 < l_3}$$

- $slev(P) < slev(S)$
- $slev(S) < slev(TS)$

  $slev(P) < slev(TS)$

Reading Files

- Access policy for reading

$$
\frac{\begin{array}{l}\forall f.\ slev(f) < slev(\text{Bob}) \Rightarrow \\ \qquad\qquad \text{Bob controls Permitted}\ (f, \text{read}) \\ \text{Bob says Permitted (File, read)} \\ slev(\text{File}) = P \\ slev(\text{Bob}) = TS \\ slev(P) < slev(S) \\ slev(S) < slev(TS)\end{array}}{\text{Permitted (File, read)}}
$$

Reading Files

- Access policy for reading

$$\forall f.\ slev(f) \leq slev(\text{Bob}) \Rightarrow$$
$$\text{Bob controls Permitted } (f, \text{read})$$
$$\text{Bob says Permitted (File, read)}$$
$$slev(\text{File}) = TS$$
$$slev(\text{Bob}) = TS$$
$$slev(P) < slev(S)$$
$$slev(S) < slev(TS)$$
$$\overline{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}$$
$$\text{Permitted (File, read)}$$

Writing Files

- Access policy for <u>writing</u>

$$\frac{\begin{array}{l}\forall f.\ \mathit{slev}(\text{Bob}) \leq \mathit{slev}(f) \Rightarrow \\ \qquad\qquad\qquad \text{Bob controls Permitted } (f, \text{write}) \\ \text{Bob says Permitted (File, write)} \\ \mathit{slev}(\text{File}) = \mathit{TS} \\ \mathit{slev}(\text{Bob}) = \mathit{S} \\ \mathit{slev}(\mathit{P}) < \mathit{slev}(\mathit{S}) \\ \mathit{slev}(\mathit{S}) < \mathit{slev}(\mathit{TS})\end{array}}{\text{Permitted (File, write)}}$$