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

$$F ::= \text{true}$$
$$| \quad \text{false}$$
$$| \quad F \wedge F$$
$$| \quad F \vee F$$
$$| \quad F \Rightarrow F$$
$$| \quad p\,(t_1,...,t_n)$$
$$| \quad P \text{ says } F \qquad \text{"saying predicate"}$$

Judgements

$$\Gamma \vdash F$$

Judgements

$$\Gamma \vdash F$$

Judgements

entails sign
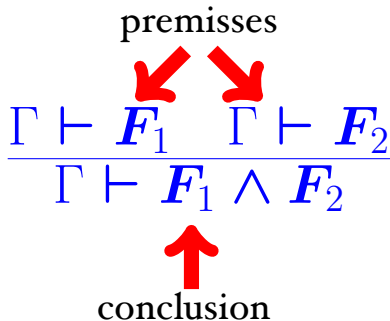
$$\Gamma \vdash F$$

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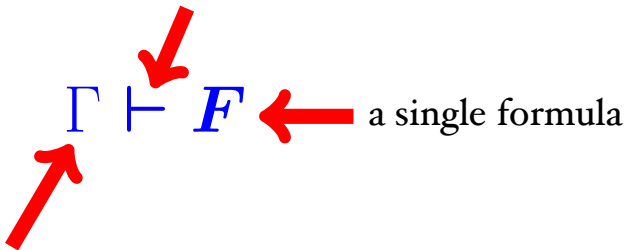
Gamma
stands for a collection of formulas
("assumptions")

Inference Rules

Inference Rules

premisses

$$\frac{\Gamma \vdash F_1 \quad \Gamma \vdash F_2}{\Gamma \vdash F_1 \wedge F_2}$$

conclusion

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

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

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

Inference Rules

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

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

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

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

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
  ($\{m\}_K \stackrel{\text{def}}{=} K \Rightarrow m$)
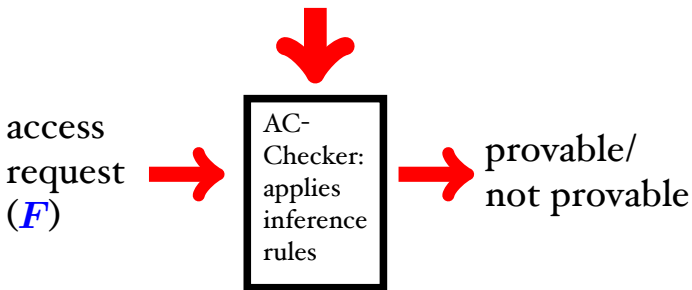
$$\text{Alice says } \{m\}_K$$

- 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

The Access Control Problem

Proofs

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

single position rules

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

Solving Sudokus

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

candidate rules

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

Solving Sudokus

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

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

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		3				
		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

The Sudoku grid:

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

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

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

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

Example Proof

We have (by axiom)

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

From (1) we get

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

From (3) and (2) we get

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

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

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

$$\Gamma \vdash del\_file$$

I want to prove

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

There is an inference rule

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

I want to prove

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

There is an inference rule

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

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

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

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

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

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

Done. Qed.

I want to prove

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

There is an inference rule

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

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

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

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

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

Program

How to prove $\Gamma \vdash F$?

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

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

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

$$\frac{\Gamma \vdash F_1}{\Gamma \vdash F_1 \lor F_2} \qquad \frac{\Gamma \vdash F_2}{\Gamma \vdash F_1 \lor F_2}$$

$$\frac{\Gamma \vdash F_1 \quad \Gamma \vdash F_2}{\Gamma \vdash F_1 \wedge F_2}$$

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 $\Gamma \vdash \text{Pred}$

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

3. So I am able to try to prove $\Gamma \vdash \text{Pred}$ with the additional assumption $\boldsymbol{F}_2$.

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

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.

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

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

- P is entitled to do F

  $P \text{ controls } F \stackrel{\text{def}}{=} (P \text{ says } F) \Rightarrow F$

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

Trusted Third Party

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

Message 1 $\quad A \rightarrow S : A, B$

Message 2 $\quad S \rightarrow A : \{K_{AB}\}_{K_{AS}}$ and $\{\{K_{AB}\}_{K_{BS}}\}_{K_{AS}}$

Message 3 $\quad A \rightarrow B : \{K_{AB}\}_{K_{BS}}$

Message 4 $\quad A \rightarrow B : \{m\}_{K_{AB}}$

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

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

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

Trusted Third Party

A sends S : Connect(A, B)

S says (Connect$(A, B) \Rightarrow$
$$\{K_{AB}\}_{K_{AS}} \wedge \{\{K_{AB}\}_{K_{BS}}\}_{K_{AS}})$$

S sends A : $\{K_{AB}\}_{K_{AS}} \wedge \{\{K_{AB}\}_{K_{BS}}\}_{K_{AS}}$

A sends B : $\{K_{AB}\}_{K_{BS}}$

A sends B : $\{m\}_{K_{AB}}$

Trusted Third Party

A sends S : $\text{Connect}(A, B)$

S says $(\text{Connect}(A, B) \Rightarrow$
$$\{K_{AB}\}_{K_{AS}} \wedge \{\{K_{AB}\}_{K_{BS}}\}_{K_{AS}})$$

S sends A : $\{K_{AB}\}_{K_{AS}} \wedge \{\{K_{AB}\}_{K_{BS}}\}_{K_{AS}}$

A sends B : $\{K_{AB}\}_{K_{BS}}$

A sends B : $\{m\}_{K_{AB}}$

$\Gamma \vdash B$ 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

Bob controls Permitted (File, read)
Bob says Permitted (File, read)
───────────────────────────────
Permitted (File, read)

Reading a File

$slev(\text{File}) < slev(\text{Bob}) \Rightarrow$
$\quad\quad\quad\quad$ Bob controls Permitted (File, read)
Bob says Permitted (File, read)
$slev(\text{File}) < slev(\text{Bob})$
—————————————————————————
$\quad\quad\quad$ Permitted (File, read)

Reading a File

$slev(\text{File}) < slev(\text{Bob}) \Rightarrow$
          Bob controls Permitted (File, read)
Bob says Permitted (File, read)
$slev(\text{File}) = P$
$slev(\text{Bob}) = S$
$slev(P) < slev(S)$
_____
          Permitted (File, read)

Substitution Rule

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

Substitution Rule

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

- $slev(\text{Bob}) = S$

- $slev(\text{File}) = P$

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

Reading a File

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

$\qquad\qquad$ Permitted (File, read)

Reading a File

$$slev(\text{File}) < slev(\text{Bob}) \Rightarrow$$
$$\text{Bob controls Permitted (File, read)}$$

Bob says Permitted (File, read)

$$slev(\text{File}) = P$$
$$slev(\text{Bob}) = TS$$
$$slev(P) < slev(S)$$
$$slev(S) < slev(TS)$$

Permitted (File, read)

Transitivity Rule

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

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

  $slev(P) < slev(TS)$

Reading Files

- Access policy for reading

$$\forall f.\ slev(f) < slev(\text{Bob}) \Rightarrow$$
$$\text{Bob controls Permitted } (f, \text{read})$$

Bob says Permitted (File, read)

$slev(\text{File}) = P$

$slev(\text{Bob}) = TS$

$slev(P) < slev(S)$

$slev(S) < slev(TS)$

───────────────────────────────

Permitted (File, read)

Reading Files

- Access policy for reading

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

$$\text{Permitted (File, read)}$$

Writing Files

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

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