CS 5430:

Example of Credentials-Based Authorization

Fred B. Schneider

Samuel B Eckert Professor of Computer Science

Department of Computer Science
Cornell University
Ithaca, New York 14853
U.S.A.

Cornell CIS
Computer Science

CAL

Language:

C ::= F (F *a formula of First-order Predicate Logic*)

| P **says** C

| P′ **speaksfor** P

| P′ **speaks** x:C **for** P

| C ∧ C′

| C ∨ C′

| C ⇒ C′

N.b. ¬C: (C ⇒ false)

Models for CAL

ω(P) is the set of beliefs principal P has.

- P **says** C iff $C \in \omega(P)$
- P' **speaksfor** P iff $\omega(P') \subseteq \omega(P)$

ω(P) called the **worldview** of P

CAL Inference Rules: **says**

$$\frac{C}{P \ \textbf{says}\ C} \qquad \frac{P\ \textbf{says}\ C}{P\ \textbf{says}\ (\ P\ \textbf{says}\ C)} \qquad \frac{P\ \textbf{says}\ (\ P\ \textbf{says}\ C)}{P\ \text{says}\ C}$$

$$\frac{P\ \textbf{says}\ (C \Rightarrow C')}{(P\ \textbf{says}\ C\ ) \Rightarrow\ (\ P\ \textbf{says}\ C')}$$

CAL Inference Rules: speaksfor

$$\frac{P \textbf{ says } (P' \textbf{speaksfor } P)}{P' \textbf{speaksfor } P} \text{ hand-off}$$

$$\frac{P' \textbf{speaksfor } P}{(P' \textbf{says } C) \Rightarrow (P \textbf{ says } C)}$$

$$\frac{P \textbf{ speaksfor } P', P' \textbf{speaksfor } P''}{P \; \boldsymbol{speaksfor} \; P''}$$

Unrestricted Delegation

$$\frac{P' \textbf{ says } C, \quad \dfrac{P' \textbf{ speaksfor } P}{(P' \textbf{ says } C) \Rightarrow (P \textbf{ says } C)}}{P \textbf{ says } C}$$

- **Warning**: P inherits beliefs from any principal that was delegated to.
- P trusting P' means
  - P adopts all beliefs of P'
  - P also adopts beliefs of any principal P' trusts (transitive).

Why Delegate?

Transitivity of delegation allows clients to be ignorant of the implementation details of services the clients invoke.

- Transitive delegations are made by implementation of service to lower-level services.
- Transitive delegations are hidden from clients.

Restricted Delegation

$$\frac{P' \,\mathbf{speaks}\; x\!: C \;\mathbf{for}\; P}{(P' \,\mathbf{says}\; C[x \coloneqq \tau]) \Rightarrow (P \;\mathbf{says}\; C[x \coloneqq \tau])}$$

Example:

CS **says** $Major(Alice)$

CS **says** $\neg Major(Alice)$

CU **says** (CS **speaksfor** CU) 😢

CU **says** (CS **speaks** $x\!: Major(x)$ **for** CU) 🙂

… CU does not inherit $\neg Major(x)$ from CS

Compound Principals

- Every principal P has a worldview $\omega(P)$.
- Compound principals combine worldviews from multiple principals to obtain a worldview for the compound principal.
- Example:
  - $P \wedge Q$: $\omega(\ P \wedge Q\ )$: $\omega(P) \cap \omega(Q)$

Useful Compound Principals

- Subprincipals of P: $P.x$
- Groups $G = \{G_1, G_2, \ldots G_n\}$

Subprincipals

For any term η:

$$\frac{}{P \mathbf{\ speaksfor\ } P.\eta}$$

$$\frac{\eta = \eta'}{P.\eta \mathbf{\ speaksfor\ } P.\eta'}$$

Use of Subprincipals

- Any belief of P is attributed to $P.x$ for any x.
  - **Hack**: Employ $P.\epsilon$ for beliefs by P that should not be attributed to other sub-principals of P.
- If L *implements* H then H is a subprincipal of L.
  - **Example**: HW implements OS, so HW.OS is the principal that corresponds to the operating system.

Implements: CAL Analysis

L implements H, so H is a subprincipal of L.

- L **says** $(H$ **says** $C)$
- L **speaksfor** H

$$L \textbf{ says } (H \textbf{ says } C), \ \frac{L \textbf{ speaksfor } H}{\big(L \textbf{ says } (H \textbf{ says } C)\big) \Rightarrow (H \textbf{ says } (H \textbf{ says } C)}$$

Implements: CAL Analysis

L implements H, so H is a subprincipal of L.

- L **says** $(H$ **says** $C)$
- L **speaksfor** H

$$\frac{L \textbf{ says } (H \textbf{ says } C), \quad \dfrac{L \textbf{ speaksfor } H}{\big(L \textbf{ says } (H \textbf{ says } C)\big) \Rightarrow (H \textbf{ says } (H \textbf{ says } C)}}{\dfrac{H \textbf{ says } (\, H \textbf{ says } C)}{H \textbf{ says } C}}$$

Group Principals

A **group** is defined by a finite enumeration of its member principals. $G = \{ P_1 , P_2 , \ldots P_N \}$

- Conjunctive Groups

$$\frac{P_i \textbf{ says } C, \textbf{for every } P_i \in G}{P_G \textbf{ says } C}$$

$$\frac{P_G \textbf{ says } C}{P \textbf{ says } C} \qquad \frac{}{P_G \textbf{ speaksfor } P} \text{ for } P \in G$$

Group Principals

- Disjunctive Groups. Hold beliefs that any member principal holds plus deductive closure!

$$\frac{P \textbf{ says } C}{P_G \textbf{ says } C} \qquad \frac{}{P \textbf{ speaksfor } P_G} \text{ for } P \in G$$

$$\frac{P_G \textbf{ says } C\,, \quad P_G \textbf{ says } (C \Rightarrow C')}{P_G \textbf{ says } C'}$$

Credentials Can Convey Beliefs

k_S-**sign**(C): K_S **says** C

- Public keys are principals.
- K_S **speaksfor** S if principal S is the only agent with access to private key k_S.

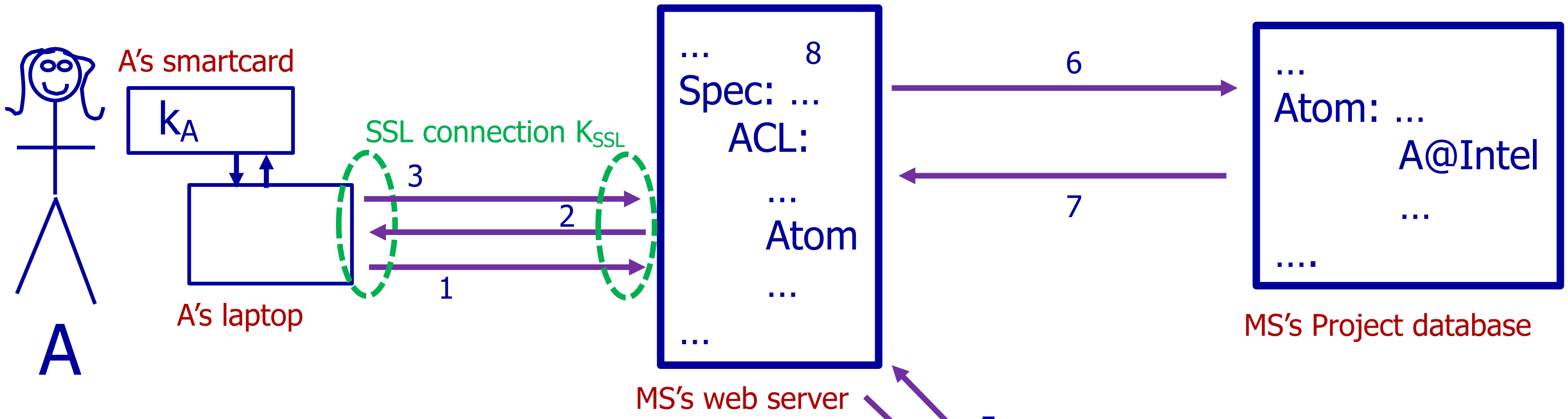
A principal S can be a hash of the running code and data that was read.

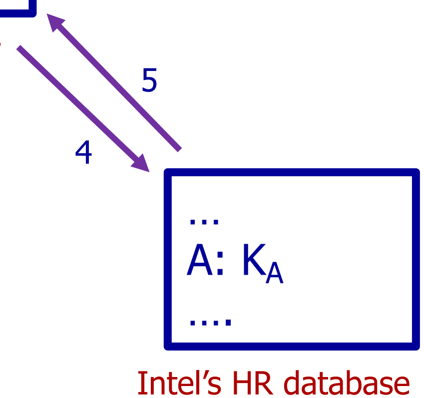
Access to a Joint Project

- A works for Intel and is known as A@Intel.
  - Public key K_A; private key k_A
  - Laptop
  - Member of Atom group
- MS has web page Spec
  - ACL allows access to Spec for members of Atom
  - CAL models as: Atom **speaksfor** Spec
    - Therefore: Atom **says** (access Spec) $\vdash$ Spec **says** (access Spec)

Suppose A requests access a Spec web page…

Application:
Accessing a Joint Project





1. read page: Spec
2. challenge: r
3. k_A-**sign**(r, A)

   SSL connection K_{SSL} (items 1–3)

4. A?
5. k_{intel}-⟨ K_A , A@Intel ⟩
6. A@Intel in Atom?
7. k_{MS}-⟨ A@Intel, Atom ⟩
8. MS web server authorizes access by Atom: Atom ∈ Spec.ACL

CAL Model for Spec Access

1. K_{SSL} **says** (A@Intel **says** (read page: Spec))
2. K_{SSL} **says** r
3. K_{SSL} **says** (K_A **says** (r,A))

   K_{SSL} **speaksfor** K_A since K_A is a subprincipal of K_{SSL}

   Conclude: K_A **says** (r,A)

5. K_{intel} **says** K_A **speaksfor** A@Intel

   K_{intel} **speaksfor** *@Intel, so: K_{intel} **speaksfor** A@Intel

   Conclude: K_A **speaksfor** A@Intel

7. K_{MS} **says** (A@Intel **speaksfor** Atom)

   MS **speaksfor** Atom since Atom is a subprincipal of MS

   K_{MS} **speaksfor** MS defn of K_{MS}

   Conclude: A@Intel **speaksfor** Atom

CAL Model for Spec Access

1. K_{SSL} **says** (A@Intel **says** (read page: Spec))
2. K_{SSL} **says** r
3. K_{SSL} **says** (K_A **says** (r,A))

   K_{SSL} **speaksfor** K_A since K_A is a subprincipal of K_{SSL}

   Conclude: K_A **says** (r,A)

5. K_{intel} **says** K_A **speaksfor** A@Intel

   K_{intel} **speaksfor** *@Intel, so: K_{intel} **speaksfor** A@Intel

   Conclude: K_A **speaksfor** A@Intel

7. K_{MS} **says** (A@Intel **speaksfor** Atom)

   MS **speaksfor** Atom since Atom is a subprincipal of MS

   K_{MS} **speaksfor** MS defn of K_{MS}

   Conclude: A@Intel **speaksfor** Atom

A@Intel **says** (read page: Spec)

CAL Model for Spec Access

1. K_{SSL} **says** (A@Intel **says** (read page: Spec))
2. K_{SSL} **says** r
3. K_{SSL} **says** (K_A **says** (r,A))

   K_{SSL} **speaksfor** K_A since K_A is a subprincipal of K_{SSL}

   Conclude: K_A **says** (r,A)

5. K_{intel} **says** K_A **speaksfor** A@Intel

   K_{intel} **speaksfor** *@Intel, so: K_{intel} **speaksfor** A@Intel

   Conclude: K_A **speaksfor** A@Intel

7. K_{MS} **says** (A@Intel **speaksfor** Atom)

   MS **speaksfor** Atom since Atom is a subprincipal of MS

   K_{MS} **speaksfor** MS defn of K_{MS}

   Conclude: A@Intel **speaksfor** Atom

A@Intel **says** (read page: Spec)

A@Intel **speaksfor** Atom

Access Authorization

A@Intel **says** (read page: Spec)

A@Intel **speaksfor** Atom

Atom **speaksfor** Spec   due to Atom $\in$ Spec.ACL

$\vdash$

Spec **says** (read page: Spec)