CS 591: Formal Methods in Security and Privacy

Probabilistic computations

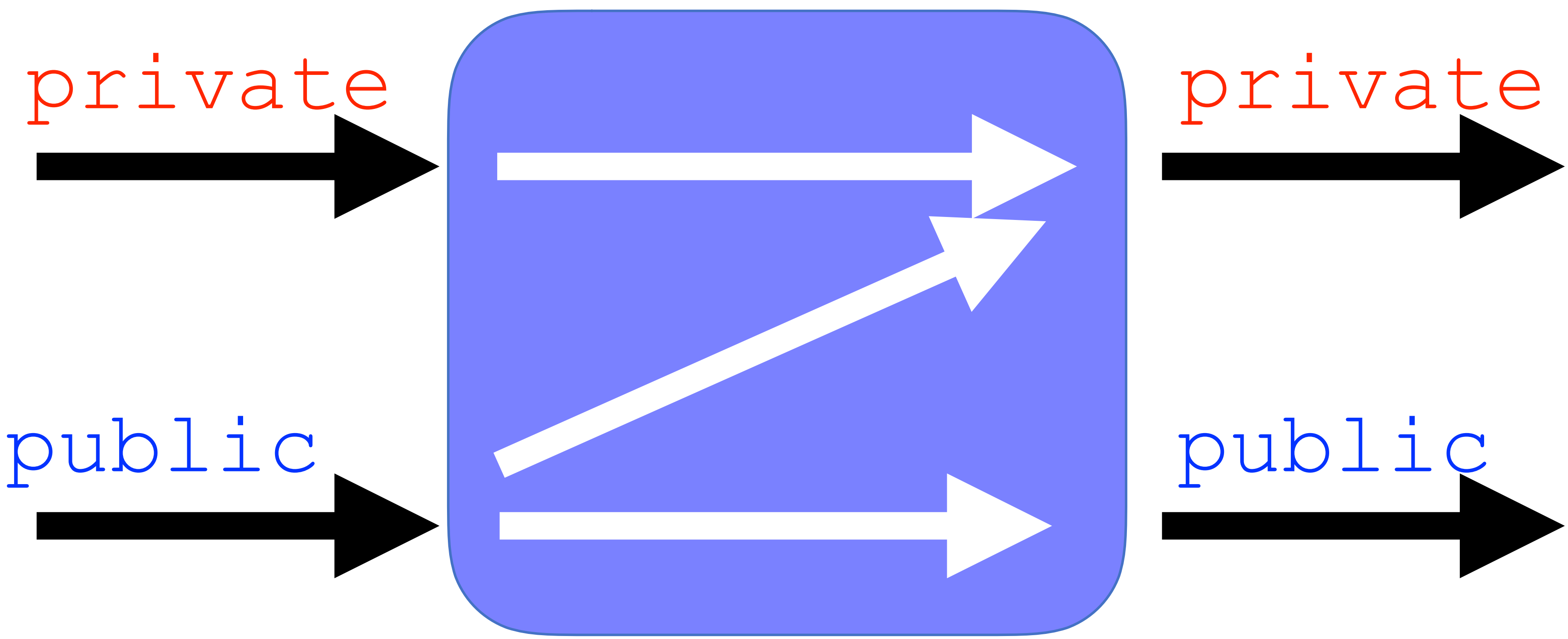
Marco Gaboardi
gaboardi@bu.edu

Alley Stoughton
stough@bu.edu

From the previous classes

Information Flow Control

We want to guarantee that confidential inputs do not flow to nonconfidential outputs.

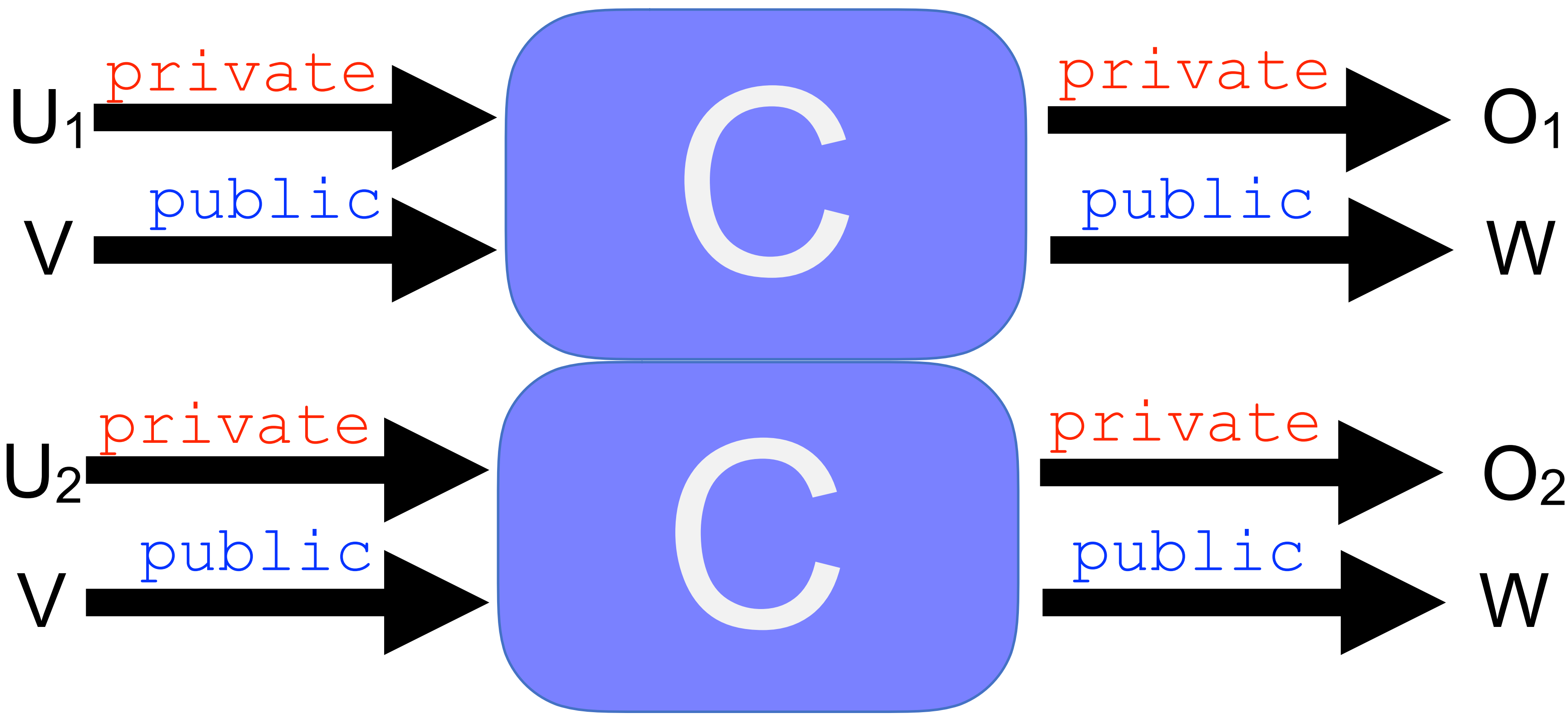


Noninterference as a Relational Property

In symbols, c is noninterferent if and only if for every $m_1 \sim_{low} m_2$:

1) $\{c\}_{m1} = \perp$ iff $\{c\}_{m2} = \perp$

2) $\{c\}_{m1} = m_1'$ and $\{c\}_{m2} = m_2'$ implies $m_1' \sim_{low} m_2'$



Relational Hoare Quadruples

Precondition
(a logical formula)

Precondition
$\text{Program}_1 \sim \text{Program}_2$
Postcondition

$$c_1 \sim c_2 : P \Rightarrow Q$$

Program Program Postcondition
(a logical formula)

Relational Assertions

$$c_1 \sim c_2 : P \Rightarrow Q$$

Need to talk about variables
of the two memories

$$c_1 \sim c_2 : x\langle 1 \rangle \leq x\langle 2 \rangle \Rightarrow x\langle 1 \rangle \geq x\langle 2 \rangle$$

Tags describing which
memory we are referring to.

Validity of Hoare quadruple

We say that the quadruple $c_1 \sim c_2 : P \Rightarrow Q$ is valid if and only if for every pair of memories m_1, m_2 such that $P(m_1, m_2)$ we have:

1) $\{c_1\}_{m1} = \perp$ iff $\{c_2\}_{m2} = \perp$
2) $\{c_1\}_{m1} = m_1'$ and $\{c_2\}_{m2} = m_2'$ implies $Q(m_1', m_2')$.

How do we check this?

Which rules do we need to prove this?

```
x:private
y:public

y:=x;y:=5
```

$: \;\; =_{low} \Rightarrow \;\; =_{low}$

Rules of Relational Hoare Logic

Assignment

$$\overline{\vdash \texttt{x}_1\texttt{:=e}_1 \sim \texttt{x}_2\texttt{:=e}_2 : \texttt{P[e}_1\texttt{<1>/x}_1\texttt{<1>,e}_2\texttt{<2>/x}_2\texttt{<2>]} \Rightarrow \texttt{P}}$$

Rules of Relational Hoare Logic

Consequence

$$\frac{P \Rightarrow S \qquad \vdash c_1 \sim c_2 : S \Rightarrow R \qquad R \Rightarrow Q}{\vdash c_1 \sim c_2 : P \Rightarrow Q}$$

We can weaken P, i.e. replace it by something that is implied by P. In this case S.

We can strengthen Q, i.e. replace it by something that implies Q. In this case R.

Rules of Relational Hoare Logic

Composition

$$\frac{\vdash \mathtt{c}_1 \sim \mathtt{c}_2 : P \Rightarrow R \qquad \vdash \mathtt{c}_1' \sim \mathtt{c}_2' : R \Rightarrow S}{\vdash \mathtt{c}_1 ; \mathtt{c}_1' \sim \mathtt{c}_2 ; \mathtt{c}_2' : P \Rightarrow S}$$

Which rules do we need to prove this?

```
x:private
y:public

if y mod 3 = 0 then
  x:=1
else
  x:=0
```

$: \;=_{\mathtt{low}} \Rightarrow \;=_{\mathtt{low}}$

Rules of Relational Hoare Logic
If-then-else

$$P \Rightarrow (e_1\langle 1\rangle \Leftrightarrow e_2\langle 2\rangle)$$

$$\vdash c_1 \sim c_2 : e_1\langle 1\rangle \land P \Rightarrow Q$$

$$\vdash c_1' \sim c_2' : \neg e_1\langle 1\rangle \land P \Rightarrow Q$$

$$\vdash \begin{array}{c} \texttt{if } e_1 \texttt{ then } c_1 \texttt{ else } c_1' \\ \sim \\ \texttt{if } e_2 \texttt{ then } c_2 \texttt{ else } c_2' \end{array} : P \Rightarrow Q$$

Which rules do we need to prove this?

```
s1:public
s2:private
r:private
i:public
n:public

proc Compare (s1:list[n] bool,s2:list[n] bool)
i:=0;
r:=0;
while i<n do
 if not(s1[i]=s2[i]) then
     r:=1
 i:=i+1

: n>0 /\ =low ⇒ =low
```

Rules of Relational Hoare Logic
While

$$\frac{P \Rightarrow (e_1\langle 1\rangle \Leftrightarrow e_2\langle 2\rangle) \qquad \vdash c_1 \sim c_2 : e_1\langle 1\rangle \wedge P \Rightarrow P}{\vdash \begin{array}{c}\texttt{while } e_1 \texttt{ do } c_1 \\ \sim \\ \texttt{while } e_2 \texttt{ do } c_2\end{array} : P \Rightarrow P \wedge \neg e_1\langle 1\rangle}$$

Invariant

Rules of Relational Hoare Logic
If-then-else - left

$$\frac{\vdash c_1 \sim c_2 : e\langle 1\rangle \wedge P \Rightarrow Q \qquad \vdash c_1' \sim c_2 : \neg e\langle 1\rangle \wedge P \Rightarrow Q}{\vdash \begin{array}{c}\texttt{if } e \texttt{ then } c_1 \texttt{ else } c_1' \\ \sim \\ c_2\end{array} : P \Rightarrow Q}$$

Rules of Relational Hoare Logic

If-then-else - right

$$\frac{\vdash c_1 \sim c_2 : e\langle 2\rangle \wedge P \Rightarrow Q \qquad \vdash c_1 \sim c_2' : \neg e\langle 2\rangle \wedge P \Rightarrow Q}{\vdash \begin{array}{c} c_1 \\ \sim \\ \texttt{if } e \texttt{ then } c_2 \texttt{ else } c_2' \end{array} : P \Rightarrow Q}$$

Rules of Relational Hoare Logic

Assignment - left

$$\frac{}{\vdash x:=e \sim \mathtt{skip}: \quad P[e\langle 1\rangle/x\langle 1\rangle] \Rightarrow P}$$

Soundness

If we can derive $\vdash c_1 \sim c_2 : P \Rightarrow Q$ through the rules of the logic, then the quadruple $c_1 \sim c_2 : P \Rightarrow Q$ is valid.

Relative Completeness

If a quadruple $c_1 \sim c_2 : P \Rightarrow Q$ is valid, and we have an oracle to derive all the true statements of the form $P \Rightarrow S$ and of the form $R \Rightarrow Q$, then we can derive $\vdash c_1 \sim c_2 : P \Rightarrow Q$ through the rules of the logic.

Soundness and completeness with respect to Hoare Logic

$$\vdash_{\mathtt{RHL}}\ \mathtt{c_1 \sim c_2 : P \Rightarrow Q}$$

iff

$$\vdash_{\mathtt{HL}}\ \mathtt{c_1 ; c_2 : P \Rightarrow Q}$$

Under the assumption that we can partition the memory adequately, and that we have termination.

Possible projects

In Easycrypt

- Look at how to guarantee trace-based noninterference.
- Look at how to guarantee side-channel free noninterference.
- Look at the relations between self-composition and relational logic.

Not related to Easycrypt

- Look at type systems for non-interference.
- Look at other methods for relational reasoning
- Look at declassification

Today:
Probabilistic Language

An example

```
OneTimePad(m : private msg) : public msg
   key :=$ Uniform({0,1}^n);
    cipher := msg xor key;
    return cipher
```

Learning a ciphertext does not change any a priori knowledge about the likelihood of messages.

Probabilistic While (PWhile)

```
c::= abort
   | skip
   | x:= e
   | x:=$ d
   | c;c
   | if e then c else c
   | while e do c
```

$d_1, d_2, \ldots$ probabilistic expressions

Probabilistic Expressions

We extend the language with expression describing probability distributions.

$$\boxed{\texttt{d::= f}(e_1,\ldots,e_n,d_1,\ldots,d_k)}$$

Where `f` is a distribution declaration

Some expression examples

`uniform(`$\{0,1\}^n$`)`   `gaussian(`k,σ`)`   `laplace(k,b)`

Semantics of Probabilistic Expressions

We would like to define it on the structure:

$$\{f(e_1,\ldots,e_n,d_1,\ldots,d_k)\}_m = \{f\}(\{e_1\}_m,\ldots,\{e_n\}_m,\{d_1\}_m,\ldots,\{d_k\}_m)$$

but is the result just a value?

Probabilistic Subdistributions

A discrete subdistribution over a set A is a function
$\mu : A \longrightarrow [0, 1]$
such that the mass of μ,

$$|\mu| = \sum_{a \in A} \mu(a)$$

verifies $|\mu| \leq 1$.

The support of a discrete subdistribution μ,
$supp(\mu) = \{a \in A \mid \mu(a) > 0\}$
is necessarily countable, i.e. finite or countably infinite.

We will denote the set of sub-distributions over A by D(A), and say that μ is of type D(A) denoted μ:D(A) if $\mu \in D(A)$.

Probabilistic Subdistributions

We call a subdistribution with mass exactly 1, a distribution.

We define the probability of an event $E \subseteq A$ with respect to the subdistribution $\mu : D(A)$ as

$$\mathbb{P}_\mu[E] = \sum_{a \in E} \mu(a)$$

Probabilistic Subdistributions

Let's consider $\mu \in D(A)$, and $E \subseteq A$, we have the following properties

$$\mathbb{P}_\mu[\varnothing] = 0$$

$$\mathbb{P}_\mu[A] \leq 1$$

$$0 \leq \mathbb{P}_\mu[E] \leq 1$$

$E \subseteq F \subseteq A$ implies $\mathbb{P}_\mu[E] \leq \mathbb{P}_\mu[F]$

$E \subseteq A$ and $F \subseteq A$ implies $\mathbb{P}_\mu[E \cup F] \leq \mathbb{P}_\mu[E] + \mathbb{P}_\mu[F] - \mathbb{P}_\mu[E \cap F]$

We will denote by **O** the subdistribution μ defined as constant 0.

Operations over Probabilistic Subdistributions

Let's consider an arbitrary a∈A, we will often use the distribution unit(a) defined as:

$$\mathbb{P}_{\text{unit}(a)}[\{b\}] = \begin{cases} 1 \text{ if } a=b \\ 0 \text{ otherwise} \end{cases}$$

We can think about unit as a function of type unit:A $\rightarrow$ D(A)

Operations over Probabilistic Subdistributions

Let's consider a distribution $\mu \in D(A)$, and a function $M : A \to D(B)$ then we can define their composition by means of an expression let a =μ in M a defined as:

$$\mathbb{P}_{\text{let a =μ in M a}}[E] = \sum_{a \in \text{supp}(\mu)} \mathbb{P}_{\mu}[\{a\}] \cdot \mathbb{P}_{(Ma)}[E]$$

Semantics of Probabilistic Expressions - revisited

We would like to define it on the structure:

$$\{f(e_1,\ldots,e_n,d_1,\ldots,d_k)\}_m = \{f\}(\{e_1\}_m,\ldots,\{e_n\}_m,\{d_1\}_m,\ldots,\{d_k\}_m)$$

With input a memory m and output a subdistribution $\mu \in D(A)$ over the corresponding type A. E.g.

$$\{uniform(\{0,1\}^n)\}_m \in D(\{0,1\}^n)$$

$$\{gaussian(k,\sigma)\}_m \in D(Real)$$