CS 599: Formal Methods in Security and Privacy

Probabilistic Noninterference

Marco Gaboardi
gaboardi@bu.edu

Alley Stoughton
stough@bu.edu

From the previous classes

An example

Suppose msg is the type of n bit messages, and xor is bitwise exclusive or.

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

$d_1, d_2, \ldots$ expressions for sub-distributions

Semantics of Commands

This is defined on the structure of commands:

$\{\texttt{abort}\}_m = \mathbf{0}$

$\{\texttt{skip}\}_m = \texttt{unit}(m)$

$\{\texttt{x:=e}\}_m = \texttt{unit}(m[x \leftarrow \{e\}_m])$

$\{\texttt{x:=\$ d}\}_m = \texttt{let } a=\{d\}_m \texttt{ in unit}(m[x \leftarrow a])$

$\{\texttt{c;c'}\}_m = \texttt{let } m'=\{c\}_m \texttt{ in } \{c'\}_{m'}$

$\{\texttt{if e then } c_t \texttt{ else } c_f\}_m = \{c_t\}_m$ If $\{e\}_m=\texttt{true}$

$\{\texttt{if e then } c_t \texttt{ else } c_f\}_m = \{c_f\}_m$ If $\{e\}_m=\texttt{false}$

$\{\texttt{while e do c}\}_m = \ldots$

Today:
Probabilistic Noninterference

Revisiting the example

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

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

How do we formalize this?

Probabilistic Noninterference

A program `prog` is probabilistically noninterferent if and only if, whenever we run it on two low equivalent memories m_1 and m_2, we have that the probabilistic distributions we get as outputs are the same on public outputs.

Low equivalence on distributions

Two distributions over memories μ_1 and μ_2 are low equivalent if and only if they coincide after projecting out all the private variables.

In symbols: $\mu_1 \sim_{low} \mu_2$

Example: Low equivalence on distributions

Consider memories with x private and y public. The distributions μ_1 and μ_2 defined as

μ_1`([x=2,y=0])=2/3,` μ_1`([x=3,y=1])=1/3`

and

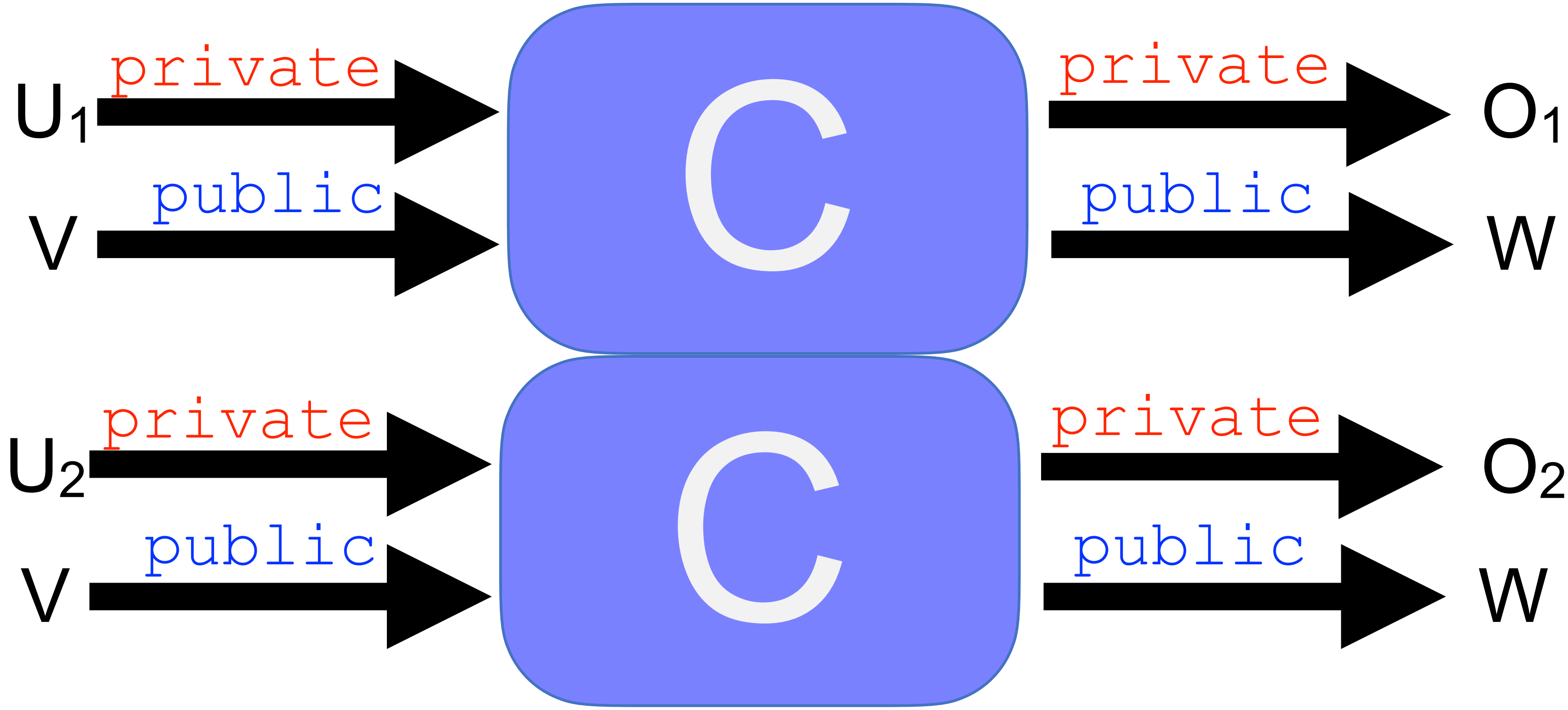
μ_2`([x=1,y=0])=1/3,`μ_2`([x=5,y=0])=1/3,`
μ_2`([x=4,y=1])=1/3`

are low equivalent.

Noninterference as a Relational Property

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

$\{c\}_{m1}=\mu_1$ and $\{c\}_{m2}=\mu_2$ implies $\mu_1 \sim_{low} \mu_2$



Revisiting the example

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

How can we prove that this is noninterferent?

Revisiting the example

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

$$m_1 \longrightarrow m_1 \oplus k \qquad\qquad m_2 \longrightarrow m_2 \oplus (m_1 \oplus k \oplus m_2)$$

We will show it is sound to pick, with some restrictions, a *function* of k as the key for m_2. What could we choose so that the cipher texts are equal?

Properties of bitwise xor

$$c \oplus (a \oplus c) = a$$

Example:

$$100 \oplus (101 \oplus 100) =$$

$$100 \oplus 001 = 101$$

Revisiting the example

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

$m_1 \longrightarrow m_1 \oplus k$

$m_2 \longrightarrow m_1 \oplus k$

Applying the property above

Noninterference as a Relational Property

c is noninterferent if and only if for every $m_1 \sim_{low} m_2$: $\{c\}_{m1}=\mu_1$ and $\{c\}_{m2}=\mu_2$ implies $\mu_1 \sim_{low} \mu_2$

We will express and prove probabilistic noninterference using Probabilistic Relational Hoare Logic

Probabilistic Relational Hoare Quadruples

Precondition

Program$_1$ ~ Program$_2$

Postcondition

Precondition

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

Probabilistic Program

Probabilistic Program

Postcondition

Validity of Probabilistic 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: $\{c_1\}_{m1} = \mu_1$ and $\{c_2\}_{m2} = \mu_2$ implies $Q(\mu_1, \mu_2)$.

Is this correct?!?

Relational Assertions

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

logical formula
over pair of memories
(i.e., relation over memories)

logical formula
over ????

We need to lift Q to be a relation on distributions, and we do this using the notion of a *coupling* between distributions

Coupling Example 1

00	0.25
01	0.25
10	0.25
11	0.25

$\mathtt{k}_1$

relation

$$\mathtt{k}_1 = \mathtt{10} \oplus \mathtt{k}_2 \oplus \mathtt{00}$$

$\mathtt{k}_2$

00	0.25
01	0.25
10	0.25
11	0.25

$\mathtt{k}_2$

$\mathtt{k}_1$

	00	01	10	11
00			0.25	
01				0.25
10	0.25			
11		0.25		

Coupling Example 2

00	0.25
01	0.25
10	0.25
11	0.25

k_1

relation

$k_1 = k_2$ or $k_1 = k_2 \oplus 11$

k_2

00	0.00
01	0.50
10	0.00
11	0.50

k_2

k_1

	00	01	10	11
00				0.25
01		0.25		
10		0.25		
11				0.25

Coupling formally

Given two sub-distributions $\mu_1 \in D(A)$, and $\mu_2 \in D(B)$, a coupling between them is a joint sub-distribution $\mu \in D(AxB)$ whose *marginal sub-distributions* are μ_1 and μ_2, respectively.

$$\pi_1(\mu)(a) = \sum_b \mu(a,b) \qquad \pi_2(\mu)(b) = \sum_a \mu(a,b)$$

$$\pi_1(\mu) = \mu_1 \qquad \pi_2(\mu) = \mu_2$$

R-Coupling

Given two sub-distributions $\mu_1 \in D(A)$, and $\mu_2 \in D(B)$, an R-coupling between them, for $R \subseteq A \times B$, is a joint sub-distribution $\mu \in D(A \times B)$ such that:

1) the marginal sub-distributions of μ are μ_1 and μ_2, respectively,
2) the support of μ is contained in R. That is, if μ`(a,b)>0, then (a,b)`$\in$R.

Relational lifting of a predicate

We say that two sub-distributions $\mu_1 \in D(A)$ and $\mu_2 \in D(B)$ are in the relational lifting of the relation $R \subseteq A x B$, denoted μ_1 `R*` μ_2, if and only if there exists a sub-distribution $\mu \in D(AxB)$ such that:

1) if μ`(a,b)>0, then (a,b)`$\in$`R`.
2) $\pi_1(\mu) = \mu_1$ and $\pi_2(\mu) = \mu_2$

I.e., there is an R-coupling for μ_1 `and` μ_2

Consequences of Lifting

Suppose E and F are predicates on memories. If we know

μ_1 (E<1> <=> F<2>)* μ_2,

then we can conclude that

$$\Pr_{\mu_1}[E] = \Pr_{\mu_2}[F]$$

Consequences of Lifting

Suppose E and F are predicates on memories. If we know

μ_1 `(E<1> => F<2>)*` μ_2,

`then we can conclude that`

$$\Pr_{\mu_1}[E] \le \Pr_{\mu_2}[F]$$

Validity of Probabilistic 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: $\{c_1\}_{m1} = \mu_1$ and $\{c_2\}_{m2} = \mu_2$ implies $Q^*(\mu_1, \mu_2)$.

Probabilistic Relational Hoare Logic

Skip

$$\frac{}{\vdash \texttt{skip} \sim \texttt{skip} : P \Rightarrow P}$$

Probabilistic Relational Hoare Logic

Assignment

$$\vdash x_1 := e_1 \sim x_2 := e_2 : P[e_1\langle 1\rangle / x_1\langle 1\rangle, e_2\langle 2\rangle / x_2\langle 2\rangle] \Rightarrow P$$

Probabilistic Relational Hoare Logic
Composition

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

Probabilistic 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.

Probabilistic Relational Hoare Logic

If-then-else

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

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

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

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

Probabilistic Relational Hoare Logic

While

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

$$\frac{\vdash c_1 \sim c_2 : e_1\langle 1\rangle \wedge P \Rightarrow P}{\vdash \begin{array}{c}\mathtt{while}\ e_1\ \mathtt{do}\ c_1 \\ \sim \\ \mathtt{while}\ e_2\ \mathtt{do}\ c_2\end{array} : P \Rightarrow P \wedge \neg e_1\langle 1\rangle}$$

Probabilistic Relational Hoare Logic

If-then-else - left

$$\vdash c_1 \sim c_2 : e\langle 1\rangle \wedge P \Rightarrow Q$$

$$\vdash c_1' \sim c_2 : \neg e\langle 1\rangle \wedge P \Rightarrow Q$$

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

Probabilistic Relational Hoare Logic

If-then-else - right

$$\vdash c_1 \sim c_2 : e\langle 2\rangle \wedge P \Rightarrow Q$$

$$\vdash c_1 \sim c_2' : \neg e\langle 2\rangle \wedge P \Rightarrow Q$$

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

Probabilistic Relational Hoare Logic

Assignment - left

$$\frac{}{\vdash x:=e \sim skip: P[e<1>/x<1>] \Rightarrow P}$$

How about the random assignment?

Probabilistic Relational Hoare Logic
Random Assignment

$$\overline{\qquad\qquad\qquad\qquad\qquad\qquad}$$

$$\vdash \mathtt{x}_1 \;\mathtt{:=\$}\; \mathtt{d}_1 \sim \mathtt{x}_2 \;\mathtt{:=\$}\; \mathtt{d}_2 : \mathtt{??} \Rightarrow Q$$

We would like to have:

$$\frac{\begin{array}{l}\texttt{for all } m_1,\ m_2,\ P(m_1,m_2) \Rightarrow \\ \texttt{let } a=\{d_1\}_{m1} \texttt{ in unit}(m_1[x_1 \leftarrow a]) \\ Q^* \\ \texttt{let } a=\{d_2\}_{m2} \texttt{ in unit}(m_2[x_2 \leftarrow a])\end{array}}{\vdash x_1 := \$\ d_1 \sim x_2 := \$\ d_2 : P \Rightarrow Q}$$

What is the problem with this rule?

Restricted Probabilistic Expressions

We consider a restricted set of expressions denoting probability distributions.

```
d::= f(d1,…,dk)
```

Where `f` is a distribution declaration

Some expression examples similar to the previous

`uniform({0,1}`^{128}`)`   `bernoulli(.5)`   `laplace(0,1)`

Notice that we don't need a memory anymore to interpret them

Isomorphisms on Sub-distributions

Given two sub-distributions $\mu_1 \in D(A)$ and $\mu_2 \in D(B)$, we say that a mapping $h: A \longrightarrow B$ is an *isomorphism* between μ_1 and μ_2 ($h \triangleleft (\mu_1, \mu_2)$) if and only iff:

1) h is a bijective map between elements in supp(μ_1) and supp(μ_2),
2) for all $a \in A$, $\mu_1(a) = \mu_2(h(a))$

Probabilistic Relational Hoare Logic
Random Assignment

$$
\frac{\begin{array}{l} P = h \triangleleft (d_1, d_2) \land \\ \quad \forall v, \\ \quad v \in \mathtt{supp}(d_1) \\ \quad \Rightarrow \\ \quad Q[v/x_1\langle 1\rangle, h(v)/x_2\langle 2\rangle] \end{array}}{\vdash x_1 := \$\ d_1 \sim x_2 := \$\ d_2 : P \Rightarrow Q}
$$

we let h’s definition refer to program variables tagged with <1> or <2>

Back to our example

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

m<1> → m<1>⊕k

m<2> → m<2>⊕(m<1>⊕k⊕m<2>)

Back to our example

d_1=Uniform($\{0,1\}^n$)

d_2=Uniform($\{0,1\}^n$)

$$h(k) = (m\langle 1\rangle \oplus k \oplus m\langle 2\rangle)$$

Is this an isomorphism from d_1 to d_2?

1) Is it bijective between elements in the support of d_1 and d_2?
2) Is it true that for all $v \in \{0,1\}^n$, $d_1(v) = d_2(h(v))$?

Yes, it's an isomorphism!

Back to our example

$$h(k)=m\langle 1\rangle \oplus k \oplus m\langle 2\rangle,\quad d_1 = d_2 = \mathrm{Uniform}(\{0,1\}^n)$$

$$P = h \lhd (d_1, d_2) \land$$

$$\forall v, v \in \mathrm{support}(d_1) \Rightarrow$$

$$m\langle 1\rangle \oplus k_1\langle 1\rangle = m\langle 2\rangle \oplus k_2\langle 2\rangle [v/k_1\langle 1\rangle, h(v)/k_2\langle 2\rangle]$$

$$\vdash k_1 :=\$ \mathrm{Uniform}(\{0,1\}^n) \sim k_2 :=\$ \mathrm{Uniform}(\{0,1\}^n) :$$

$$P \Rightarrow m\langle 1\rangle \oplus k_1\langle 1\rangle = m\langle 2\rangle \oplus k_2\langle 2\rangle$$

Back to our example

$$\frac{\begin{array}{l} h(k)=m<1>\oplus k\oplus m<2>,\ d_1 = d_2 = \mathrm{Uniform}(\{0,1\}^n) \\ P = h \lhd (d_1,d_2) \land \\ \quad \forall v, v \in \mathrm{support}(d_1) \Rightarrow \\ \quad m<1>\oplus v=m<2>\oplus (m<1>\oplus v\oplus m<2>) \end{array}}{\begin{array}{l} \vdash k_1 :=\$\mathrm{Uniform}(\{0,1\}^n) \sim k_2 :=\$\mathrm{Uniform}(\{0,1\}^n) : \\ \quad P \Rightarrow m<1>\oplus k_1<1>=m<2>\oplus k_2<2> \end{array}}$$

Back to our example

$P = h \lhd (d_1, d_2) \wedge \forall v, v \in support(d_1) \Rightarrow m\langle 1\rangle \oplus v = m\langle 2\rangle \oplus (m\langle 1\rangle \oplus v \oplus m\langle 2\rangle)$

$$\text{True} \Rightarrow P$$

$$\vdash k_1 := \$Uniform(\{0,1\}^n) \sim k_2 := \$Uniform(\{0,1\}^n) : P \Rightarrow m\langle 1\rangle \oplus k_1\langle 1\rangle = m\langle 2\rangle \oplus k_2\langle 2\rangle$$

$$\vdash k_1 := \$Uniform(\{0,1\}^n) \sim k_2 := \$Uniform(\{0,1\}^n) : \text{True} \Rightarrow m\langle 1\rangle \oplus k_1\langle 1\rangle = m\langle 2\rangle \oplus k_2\langle 2\rangle$$

By Consequence

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.

Completeness?