CS 591: Formal Methods in Security and Privacy

Approximate probabilistic relational Hoare Logic

Marco Gaboardi
gaboardi@bu.edu

Alley Stoughton
stough@bu.edu

Q&A

To increase interactivity, I will ask more question to each one of you.

It is not a test, you can always answer “pass!”

Recording

This is a reminder that we will record the class and we will post the link on Piazza.

This is also a reminder to myself to start recording!

From the previous classes

An example

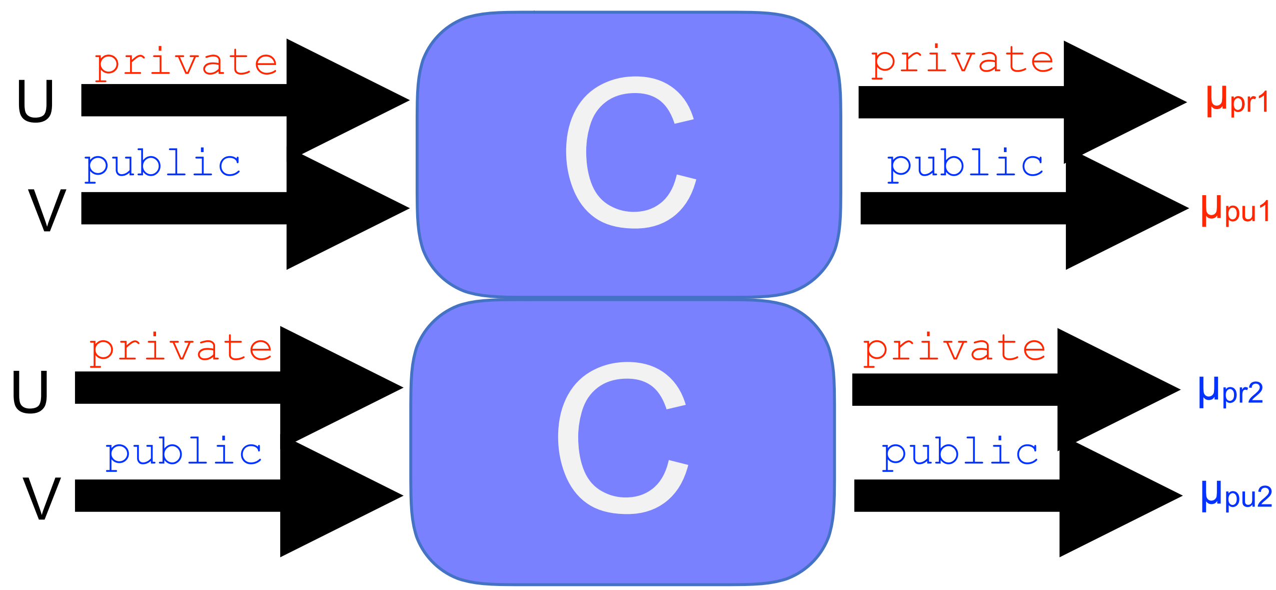
```
OneTimePad(m : private msg) : public msg
  key :=$ Uniform({0,1}^n);
   cipher := msg xor key;
   return cipher
```

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

Probabilistic Noninterference as a Relational Property

c is probabilistically 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 := msg 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 := msg xor key;
   return cipher
```

m_1 m_2

$m_1 \oplus k$

Revisiting the example

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

m_1 $\quad$ m_2

$\downarrow$

$m_1 \oplus k$

Suppose we can now chose the key for m_2. What could we choose?

Revisiting the example

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

m_1 → $m_1 \oplus k$

m_2 → $m_2 \oplus (m_1 \oplus k \oplus m_2)$

Suppose we can now chose the key for m_2. What could we choose?

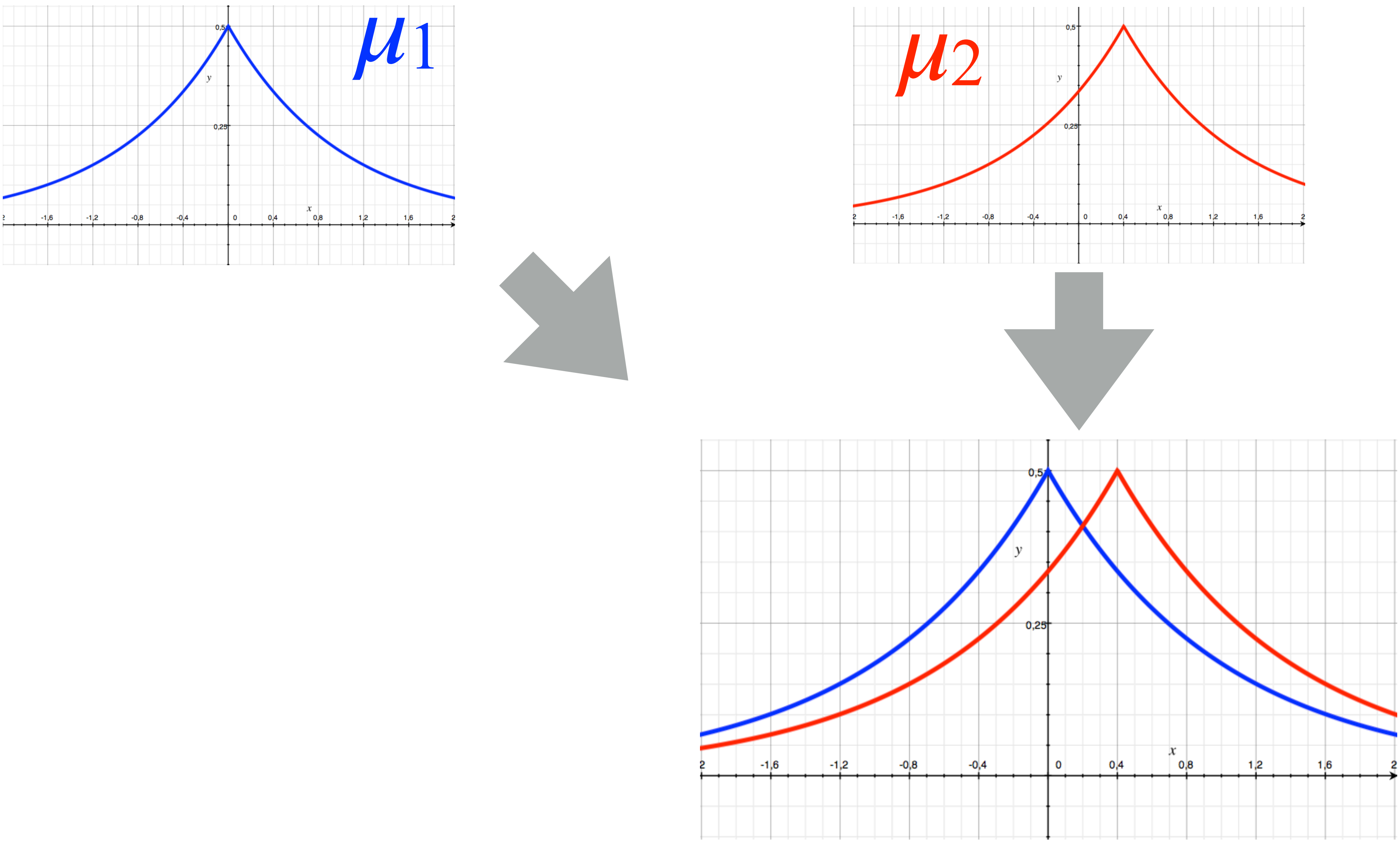
Revisiting the example

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

$m_1 \rightarrow m_1 \oplus k$

$m_2 \rightarrow m_1 \oplus k$

Coupling



Example of Our Coupling

00	0.25
01	0.25
10	0.25
11	0.25

$$k = \underbrace{10}_{m_1} \oplus k \oplus \underbrace{00}_{m_2}$$

00	0.25
01	0.25
10	0.25
11	0.25

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

Probabilistic Relational Hoare Quadruples

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

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

Precondition
(a logical formula)

Probabilistic
Program

Probabilistic
Program

Postcondition
(???)

R-Coupling

Given two 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 distribution $\mu \in D(A \times B)$ such that:

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

Example of Our Coupling

00	0.25
01	0.25
10	0.25
11	0.25

$$R(k, \underbrace{10}_{m_1} \oplus k \oplus \underbrace{00}_{m_2})$$

00	0.25
01	0.25
10	0.25
11	0.25

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

Relational lifting of a predicate

We say that two subdistributions $\mu_1 \subseteq D(A)$ and $\mu_2 \subseteq D(B)$ are in the relational lifting of the relation $R \subseteq AxB$, denoted μ_1 R^* μ_2 if and only if there exist an R-coupling between them.

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

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

To say that this is valid we need to show that for every m_1, m_2 such that $P(m_1, m_2)$ we need to show $P^*(\texttt{unit}(m_1), \texttt{unit}(m_2))$.

Probabilistic Relational Hoare Logic

Composition

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

How about random assignment?

Today:
Rand rule
+
approximate probabilistic noninterference

Probabilistic Relational Hoare Logic

Random Assignment

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

$$\vdash \texttt{x}_1 \texttt{ :=\$ d}_1 \sim \texttt{x}_2 \texttt{ :=\$ d}_2 : \texttt{??}$$

We would like to have:

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

What is the problem with this rule?

Restricted Probabilistic Expressions

We consider a restricted set of expressions denoting probability distributions.

$$d ::= f(d_1, \ldots, d_k)$$

Where `f` is a distribution declaration

Some expression examples similar to the previous

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

Restricted Probabilistic Expressions

We consider a restricted set of expressions denoting probability distributions.

$$d ::= f(d_1, \ldots, d_k)$$

Where `f` is a distribution declaration

Some expression examples similar to the previous

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

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

A sufficient condition for R-Coupling

Given two distributions $\mu_1 \in D(A)$, and $\mu_2 \in D(B)$, and a relation $R \subseteq A \times B$, if there is a mapping $h: A \to B$ such that:

1) h is a bijective map between elements in $supp(\mu_1)$ and $supp(\mu_2)$,
2) for every $a \in supp(\mu_1)$, $(a,h(a)) \in R$
3) $Pr_{x\sim\mu1}[x=a] = Pr_{x\sim\mu2}[x=h(a)]$

Then, there is an R-coupling between μ_1 and μ_2.
We write $h \lhd (\mu_1,\mu_2)$ in this case.

Probabilistic Relational Hoare Logic

Random Assignment

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

Back to our example

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

Back to our example

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

Back to our example

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

m_1 m_2

Back to our example

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

m_1 $\qquad\qquad$ m_2

$\downarrow$

$m_1 \oplus k$

Back to our example

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

$$m_1 \longrightarrow m_1 \oplus k$$

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

Back to our example

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

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

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

Is this a good map?

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

Back to our example

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

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

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

Is this a good map?

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

What is the relation?

Back to our example

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

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

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

Is this a good map?

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

What is the relation?

$$m\langle 1\rangle \oplus k\langle 1\rangle = m\langle 2\rangle \oplus k\langle 2\rangle$$

Back to our example

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

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

Is this a good map?

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

1) it is bijective between elements in the support of {d_1} and {d_2}
2) for every $k \in$ supp({d_1}), $m\langle 1\rangle \oplus k = m\langle 2\rangle \oplus (m\langle 1\rangle \oplus k \oplus m\langle 2\rangle)$
3) $Pr_{x\sim\{d1\}}[x=v] = Pr_{x\sim\{d2\}}[x=v]$

Back to our example

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

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

Is this a good map?

$$h(k) = (m<1> \oplus k \oplus m<2>)$$

1) it is bijective between elements in the support of $\{d_1\}$ and $\{d_2\}$
2) for every $k \in supp(\{d_1\})$, $m<1> \oplus k = m<2> \oplus (m<1> \oplus k \oplus m<2>)$
3) $Pr_{x\sim\{d1\}}[x=v] = Pr_{x\sim\{d2\}}[x=v]$

It is a good map!

Back to our example

$$h(k) = (m\langle 1\rangle \oplus k \oplus m\langle 2\rangle) \lhd (\{d_1\}, \{d_2\})$$

$$P = \forall k, k \in \{0,1\}^n$$

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

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

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

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

Back to our example

$h(k) = (m\langle 1\rangle \oplus k \oplus m\langle 2\rangle) \lhd (\{d_1\},\{d_2\})$

$P = \forall k, k \in \{0,1\}^n$

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

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

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

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

Using the assignment rule, we can conclude.

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?