CS 591: Formal Methods in Security and Privacy
Probabilistic Noninterference

Marco Gaboardi
gaboardi@bu.edu

Alley Stoughton
stough@bu.edu
From the previous classes
An example

OneTimePad$(m : \text{private msg}) : \text{public msg}$
key := $\$ \text{Uniform}(\{0,1\}^n)$;
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, ... probabilistic expressions
Semantics of Commands

This is defined on the structure of commands:

\[ \{\text{abort}\}_m = \emptyset \]

\[ \{\text{skip}\}_m = \text{unit}(m) \]

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

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

\[ \{c;c'\}_m = \text{let } m' = \{c\}_m \text{ in } \{c'\}_m \]

\[ \{\text{if } e \text{ then } c_t \text{ else } c_f\}_m = \{c_t\}_m \text{ if } \{e\}_m = \text{true} \]

\[ \{\text{if } e \text{ then } c_t \text{ else } c_f\}_m = \{c_f\}_m \text{ if } \{e\}_m = \text{false} \]

\[ \{\text{while } e \text{ do } c\}_m = \sup_{n \in \text{Nat}} \mu_n \]

\[ \mu_n = \text{let } m' = \{(\text{while}^n e \text{ do } c)\}_m \text{ in } \{\text{if } e \text{ then abort}\}_m \]
Today: Probabilistic Noninterference
Revisiting the 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.
Revisiting the 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.

How do we formalize this?
Probabilistic Noninterference

A program $\text{prog}$ 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 $\mu_1$ and $\mu_2$ are low equivalent if and only if they coincide after projecting out all the private variables.

In symbols: $\mu_1 \sim_{\text{low}} \mu_2$
Example: Low equivalence on distributions

Consider memories with $x$ private and $y$ public. The distributions $\mu_1$ and $\mu_2$ defined as:

$\mu_1([x=2, y=0]) = \frac{2}{3}$, $\mu_1([x=3, y=1]) = \frac{1}{3}$

and

$\mu_2([x=1, y=0]) = \frac{1}{3}$, $\mu_2([x=5, y=0]) = \frac{1}{3}$, $\mu_2([x=4, y=1]) = \frac{1}{3}$

are low equivalent.
Noninterference as a Relational Property

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

\[
\{c\}_{m_1} = \mu_1 \text{ and } \{c\}_{m_2} = \mu_2 \implies \mu_1 \sim_{low} \mu_2
\]
Revisiting the example

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

"OneTimePad" function with parameters and return type defined.
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

\begin{Verbatim}
OneTimePad(m : private msg) : public msg
key := Uniform({0,1}^n);
cipher := msg xor key;
return cipher
\end{Verbatim}
Revisiting the example

\[
\text{OneTimePad}(m : \text{private msg}) : \text{public msg} \\
\text{key} := \$ \text{Uniform}({0,1}^n); \\
\text{cipher} := \text{msg xor key}; \\
\text{return cipher}
\]
Revisiting the example

\begin{align*}
\textbf{OneTimePad}(m : \text{private msg}) & : \text{public msg} \\
& \quad \text{key} := \$ \text{Uniform}(\{0,1\}^n) \\
& \quad \text{cipher} := m \oplus \text{key} \\
& \quad \text{return cipher}
\end{align*}
Revisiting the example

\[ \text{OneTimePad}(m : \text{private msg}) : \text{public msg} \]

\[
\begin{align*}
\text{key} & := \$ \text{Uniform}\{0,1\}^n; \\
\text{cipher} & := m \text{ xor} \text{ key}; \\
\text{return} & \text{ cipher}
\end{align*}
\]

\[ m_1 \quad m_2 \]

\[ m_1 \oplus k \]

Suppose we can now chose the key for \( m_2 \). What could we choose?
Revisiting the example

\[ \text{OneTimePad}(m : \text{private msg}) : \text{public msg} \]
\[
\begin{align*}
\text{key} & := \$ \text{Uniform}\{0,1\}^n; \\
\text{cipher} & := m \oplus \text{key}; \\
\text{return cipher}
\end{align*}
\]

Suppose we can now chose the key for \( m_2 \). What could we choose?
Properties of xor

\[ c \oplus (a \oplus c) = a \]
Properties of xor

\[ c \oplus (a \oplus c) = a \]

Example:

\[ 100 \oplus (101 \oplus 100) = 100 \oplus 001 = 101 \]
Revisiting the example

\begin{align*}
\text{OneTimePad}(m : \text{private msg}) & : \text{public msg} \\
\text{key} & := \$ \text{Uniform}\{0,1\}^n; \\
\text{cipher} & := m \text{ xor key}; \\
\text{return cipher}
\end{align*}

Applying the property above

\begin{align*}
m_1 \\ \downarrow \\ m_1 \oplus k
\end{align*}

\begin{align*}
m_2 \\ \downarrow \\ m_1 \oplus k
\end{align*}

Applying the property above
Revisiting the example

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

\[ \mu_1 \]

\[ \mu_2 \]
Coupling

$\mu_1$

$\mu_2$
Example of Our Coupling

\[ k_1 = 10 \oplus k_2 \oplus 00 \]
Example of Our Coupling

\[ k_1 = 10 \oplus k_2 \oplus 00 \]
Coupling formally

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

$$
\pi_1(\mu)(a) = \sum_b \mu(a, b) \quad \pi_2(\mu)(b) = \sum_a \mu(a, b)
$$
Probabilistic Relational Hoare Quadruples

\[ c_1 \sim c_2 : P \Rightarrow Q \]

Precondition

Program_1 \sim \text{Program}_2

Postcondition

Probabilistic Program

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: $\{c_1\}_{m_1} = \mu_1$ and $\{c_2\}_{m_2} = \mu_2$ implies $Q(\mu_1, \mu_2)$. 
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:

$\{c_1\}_{m_1} = \mu_1$ and $\{c_2\}_{m_2} = \mu_2$ 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
logical formula over ????
(i.e. relation over memories)
**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 $\mu$ are $\mu_1$ and $\mu_2$, respectively,
2) the support of $\mu$ is contained in $R$. That is, if $\mu(a, b) > 0$, then $(a, b) \in R$. 
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 $R \subseteq A \times B$, denoted $\mu_1 \bowtie R^* \bowtie \mu_2$ if and only if there exist a subdistribution $\mu \subseteq D(A \times B)$ such that:

1) if $\mu(a, b) > 0$, then $(a, b) \in Q$.

2) $\pi_1(\mu) = \mu_1$ and $\pi_2(\mu) = \mu_2$
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 \( R \subseteq A \times B \), denoted \( \mu_1 \ R^* \mu_2 \) if and only if there exist a subdistribution \( \mu \subseteq D(A \times B) \) such that:

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

Does it remind you something?
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:

\[
\{c_1\}_{m_1} = \mu_1 \quad \text{and} \quad \{c_2\}_{m_2} = \mu_2 \implies Q^*(\mu_1, \mu_2).
\]
Probabilistic Relational Hoare Logic

\[ \vdash \text{skip} \rightarrow \text{skip} : P \Rightarrow P \]
Probabilistic Relational Hoare Logic
Assignment

\[\vdash x_1 := e_1 \sim x_2 := e_2 : P [e_1 < 1>/ x_1 < 1>, e_2 < 2>/ x_2 < 2>] \Rightarrow P\]
Probabilistic Relational Hoare Logic Composition

\[ \vdash c_1 \sim c_2 : P \Rightarrow R \quad \vdash c_1' \sim c_2' : R \Rightarrow S \]

\[ \vdash c_1 ; c_1' \sim c_2 ; c_2' : P \Rightarrow S \]
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**

**Consequence**

\[
\begin{align*}
P &\Rightarrow S \\
\vdash \neg c_1 \sim c_2 &: S \Rightarrow R \\
R &\Rightarrow Q \\
\hline
\vdash \neg c_1 \sim c_2 &: P \Rightarrow Q
\end{align*}
\]
Probabilistic Relational Hoare Logic

If-then-else

\( P \Rightarrow (e_1 < 1> \iff e_2 < 2>) \)

\( \vdash c_1 \sim c_2 : e_1 < 1> \land P \Rightarrow Q \)

\( \vdash c_1' \sim c_2' : \neg e_1 < 1> \land P \Rightarrow Q \)

\[
\begin{array}{c}
\text{if } e_1 \text{ then } c_1 \text{ else } c_1' \\
\text{if } e_2 \text{ then } c_2 \text{ else } c_2'
\end{array}
\]

\( \vdash \sim : P \Rightarrow Q \)
Probabilistic Relational Hoare Logic

While

\[ P \Rightarrow (e_1^{<1>} \Leftrightarrow e_2^{<2>}) \]

\[ \vdash c_1 \sim c_2 : e_1^{<1>} \land P \Rightarrow P \]

\[ \vdash \text{while } e_1 \text{ do } c_1 \]

\[ \vdash \sim : P \Rightarrow P \land \neg e_1^{<1>} \]

\[ \vdash \text{while } e_2 \text{ do } c_2 \]
Probabilistic Relational Hoare Logic

If-then-else - left

\[ \vdash c_1 \sim c_2 : e < 1 > \land P \Rightarrow Q \]

\[ \vdash c_1' \sim c_2 : \neg e < 1 > \land P \Rightarrow Q \]

\[ \vdash \sim c_2 \quad : P \Rightarrow Q \]
Probabilistic Relational Hoare Logic

If-then-else - right

\[ \vdash c_1 \sim c_2 : e < 2 > \land P \Rightarrow Q \]

\[ \vdash c_1 \sim c_2' : \neg e < 2 > \land P \Rightarrow Q \]

\[ \vdash \sim \text{if } e \text{ then } c_2 \text{ else } c_2' : P \Rightarrow Q \]
Probabilistic Relational Hoare Logic
Assignment - left

\[ \vdash x := e \sim \text{skip}: P[e^{<1>}/x^{<1>}] \Rightarrow P \]
How about the random assignment?
Probabilistic Relational Hoare Logic
Random Assignment

\[ \vdash x_1 := d_1 \sim x_2 := d_2 : ?? \]
We would like to have:

\[ P(m_1, m_2) \]

\[ \Rightarrow \]

\[ \text{let } a = \{d_1\}_{m_1} \text{ in unit}(m_1[x_1 \leftarrow a]) \]

\[ Q^* \]

\[ \text{let } a = \{d_2\}_{m_2} \text{ in unit}(m_2[x_2 \leftarrow a]) \]

\[ \vdash x_1 ::= \$ \ d_1 \sim x_2 ::= \$ \ d_2 : \ P \Rightarrow Q \]

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

\texttt{uniform(\{0,1\}^{128})} \quad \texttt{bernoulli(.5)} \quad \texttt{laplace(0,1)}
Restricted Probabilistic Expressions

We consider a restricted set of expressions denoting probability distributions.

\[ d::= f(d_1,...,d_k) \]

Where \( f \) is a distribution declaration.

Some expression examples similar to the previous

\[ \text{uniform}\{0,1\}^{128}, \text{bernoulli}(0.5), \text{laplace}(0,1) \]

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 $h: A \rightarrow B$ such that:

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

Then, there is an $R$-coupling between $\mu_1$ and $\mu_2$. We write $h \leadsto (\mu_1, \mu_2)$ in this case.
Provable Relational Hoare Logic
Random Assignment

\[ h \triangleleft (\{d_1\}, \{d_2\}) \]
\[ P = \forall v, v \in \text{supp}(\{d_1\}) \]
\[ \Rightarrow Q[v/x_1<1>, h(v)/x_2<2>] \]

\[ \vdash x_1 := \$ d_1 \sim x_2 := \$ d_2 : P \Rightarrow Q \]
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

\textbf{OneTimePad}(m : private msg) : public msg

\hspace{1cm} key := \$ \text{Uniform}\left(\{0,1\}^n\right);

\hspace{1cm} cipher := msg \text{ xor} key;

\hspace{1cm} return cipher

\[ \begin{align*}
    m_1 \\
    m_2
\end{align*} \]
Back to our example

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

\[ m_1 \oplus k \]

\[ m_1 \]

\[ m_2 \]
Back to our example

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

\[ m_1 \oplus k \]
\[ m_2 \oplus (m_1 \oplus k \oplus m_2) \]
OneTimePad(m : private msg) : public msg
    key := Uniform({0,1}^n);
    cipher := msg xor key;
    return cipher

\[ h(k) = (m<1> \oplus k \oplus m<2>) \]
Back to our example

\[ \text{OneTimePad}(m : \text{private msg}) : \text{public msg} \]

\[
\begin{align*}
\text{key} & := \text{Uniform}\{\{0,1\}^n\}; \\
\text{cipher} & := \text{msg xor key}; \\
\text{return cipher}
\end{align*}
\]

\[ d_1 = \text{Uniform}\{\{0,1\}^n\} \quad d_2 = \text{Uniform}\{\{0,1\}^n\} \]

Is this a good map?

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

What is the relation?
Back to our example

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

d_1=Uniform(\{0,1\}^n)  \quad d_2=Uniform(\{0,1\}^n)

Is this a good map?

\[ h(k) = (m^{<1>} \oplus k \oplus m^{<2>}) \]

What is the relation?

\[ m^{<1>} \oplus k^{<1>} = m^{<2>} \oplus k^{<2>} \]
Back to our example

\[ h(k) = (m^{<1>} \oplus k \oplus m^{<2>}) \]

Is this a good map?

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

\[ d_1 = \text{Uniform}(\{0,1\}^n) \quad \text{d}_2 = \text{Uniform}(\{0,1\}^n) \]

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 \( k \in \text{supp}\{d_1\} \), \( m<1> \oplus k = m<2> \oplus (m<1> \oplus k \oplus m<2>) \)
3) \( \Pr_{x \sim \{d_1\}}[x=v] = \Pr_{x \sim \{d_2\}}[x=v] \)

It is a good map!
Back to our example

\[ h(k) = (m^{<1>} \oplus k \oplus m^{<2>}) \leftarrow (\{d_1\}, \{d_2\}) \]

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

\[ \Rightarrow m^{<1>} \oplus k^{<1>} = m^{<2>} \oplus k^{<2>} \]

\[ [v/k^{<1>}, h(v)/k^{<2>}] = m^{<1>} \oplus k = m^{<2>} \oplus (m^{<1>} \oplus k \oplus m^{<2>}) \]

\[ \vdash k_1 \leftarrow \text{Uniform} (\{0,1\}^n) \sim k_2 \leftarrow \text{Uniform} (\{0,1\}^n) : \\
\text{True} \Rightarrow m^{<1>} \oplus k_1^{<1>} = m^{<2>} \oplus k_2^{<2>} \]
Back to our example

\[ h(k) = (m1 \oplus k \oplus m2) \iff (\{d1\}, \{d2\}) \]

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

\[ \Rightarrow m1 \oplus k1 < 1 > = m2 \oplus k2 < 2 > \]

\[ [v / k1 < 1 >, h(v) / k2 < 2 >] = m1 \oplus k = m2 \oplus (m1 \oplus k \oplus m2) \]

\[ \vdash k1 : = \text{Uniform} (\{0, 1\}^n) \sim k2 : = \text{Uniform} (\{0, 1\}^n) : \]

\[ \text{True} \Rightarrow m1 \oplus k1 < 1 > = m2 \oplus k2 < 2 > \]

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?