CS 591: Formal Methods in Security and Privacy
Approximate probabilistic relational Hoare Logic

Marco Gaboardi
gaboardi@bu.edu

Alley Stoughton
stough@bu.edu
To increase interactivity, I will ask more questions to each one of you.

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

Remember that the third assignment was due yesterday. If you are still working on it, no problem, but please do let us know.
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.
c is probabilistically noninterferent if and only if for every $m_1 \sim_{\text{low}} m_2$:

$\{c\}_{m_1} = \mu_1$ and $\{c\}_{m_2} = \mu_2$ implies $\mu_1 \sim_{\text{low}} \mu_2$
Probabilistic Relational Hoare Quadruples

Precondition
Program₁ \sim Program₂

Postcondition

\begin{align*}
\text{Precondition} & \quad \text{(a logical formula)} \\
\text{Program}_1 \sim \text{Program}_2 \\
\text{Postcondition} & \quad \text{(a logical formula)}
\end{align*}

\begin{align*}
c_1 \sim c_2 : P & \Rightarrow Q
\end{align*}
**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 $(a,b) \in R$. 
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 $Q^*(\mu_1, \mu_2)$. 
Consequences of Coupling

Given the following pRHL judgment

\[ \vdash c_1 \sim c_2 : \text{True} \Rightarrow Q \]

We have that:

if \( Q \Rightarrow (R\langle 1 \rangle \iff S\langle 2 \rangle) \), then \( \Pr[c_1 : R] = \Pr[c_2 : S] \)

if \( Q \Rightarrow (R\langle 1 \rangle \Rightarrow S\langle 2 \rangle) \), then \( \Pr[c_1 : R] \leq \Pr[c_2 : S] \)
A more realistic example

StreamCipher(m : private msg[n]) : public msg[n]
  pkey := $ PRG(Uniform({0,1}^k));
  cipher := msg xor pkey;
  return cipher
Properties of PRG

We would like the PRG to increase the number of random bits but also to guarantee the result to be (almost) random.

We can express this as:

\[
\text{PRG: } \{0,1\}^k \rightarrow \{0,1\}^n \text{ for } n>k
\]

\[
\text{PRG(U}\text{Uniform(}\{0,1\}^k) \approx \text{Uniform(}\{0,1\}^n)
\]

How can we measure the similarity between the result of PRG and the uniform distribution?
Statistical distance

We say that two distributions $\mu_1, \mu_2 \in D(A)$, are at statistical distance $\delta$ if and only if:

$$\Delta(\mu_1, \mu_2) = \max_{E \subseteq A} | \mu_1(E) - \mu_2(E) | = \delta$$

For discrete distributions the statistical distance can also be characterized as:

$$\Delta(\mu_1, \mu_2) = \frac{1}{2} \sum_{a \in A} | \mu_1(a) - \mu_2(a) |$$
Properties of PRG

We would like the PRG to increase the number of random bits but also to guarantee the result to be (almost) random.

We can express this as:

\[
\text{PRG: } \{0,1\}^k \rightarrow \{0,1\}^n \text{ for } n>k
\]

\[
\Delta(\text{PRG}(\text{Uniform}(\{0,1\}^k),\text{Uniform}(\{0,1\}^n)) \leq 2^{-n}
\]

In fact this is a too strong requirement - usually we require that every polynomial time adversary cannot distinguish the two distributions in statistical distance.
How can we prove this secure?

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

StreamCipher(m : private msg[n])
  : public msg[n]
  pkey := $ PRG(Uniform({0,1}^k));
  cipher := msg xor pkey;
  return cipher
How can we prove this secure?

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

StreamCipher(m : private msg[n])
  : public msg[n]
  pkey := $ PRG(Uniform({0,1}^k));
  cipher := msg xor pkey;
  return cipher
How can we prove this secure?

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

\begin{verbatim}
  : public msg[n]
  key := $\text{Uniform}([0,1]^n)$;
  cipher := msg xor key;
  return cipher
\end{verbatim}

StreamCipher: $\text{StreamCipher}(m : \text{private msg}[n])$

\begin{verbatim}
  : public msg[n]
  pkey := $\text{PRG(Uniform}([0,1]^k))$;
  cipher := msg xor pkey;
  return cipher
\end{verbatim}
How can we prove this secure?

**OneTimePad**

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

**StreamCipher**

\[ \text{StreamCipher}(m : \text{private } msg[n]) : \text{public } msg[n] \]
\[ \text{pkey} := \$ \text{PRG}(\text{Uniform}(\{0,1\}^k)); \]
\[ \text{cipher} := \text{msg} \oplus \text{pkey}; \]
\[ \text{return cipher} \]
How can we prove this secure?

**OneTimePad**

(m : private msg[n])

: public msg[n]

key := $ Uniform({0,1}^n);

cipher := msg xor key;

return cipher

**StreamCipher**

(m : private msg[n])

: public msg[n]

pkey := $ PRG(Uniform({0,1}^k));

cipher := msg xor pkey;

return cipher

\[ \Delta( m \oplus k ), \quad m \oplus p_k \leq \delta \]
Today:
approximate probabilistic
noninterference
How to reason formally about this formally?
Approximate Probabilistic Relational Hoare Logic

\[ \vdash_{\delta} c_1 \sim c_2 : P \Rightarrow Q \]

- Indistinguishability parameter
- Precondition (a logical formula)
- Postcondition (a logical formula)

Probabilistic Program

Probabilistic Program

Probabilistic Program
How can we define validity?
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 $Q^*(\mu_1, \mu_2)$. 
Given two distributions $\mu_1 \in D(A)$, and $\mu_2 \in D(B)$, we have an $R-\delta$-coupling between them, for $R \subseteq A \times B$ and $0 \leq \delta \leq 1$, if there are two joint distributions $\mu_L, \mu_R \in D(A \times B)$ such that:

1) $\pi_1(\mu_L) = \mu_1$ and $\pi_2(\mu_R) = \mu_2$, 
2) the support of $\mu_L$ and $\mu_R$ is contained in $R$. That is, if $\mu_L(a,b) > 0$, then $(a,b) \in R$, and if $\mu_R(a,b) > 0$, then $(a,b) \in R$. 
3) $\Delta(\mu_L, \mu_R) \leq \delta$
Approximate 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 $\delta$-lifting of the relation $R \subseteq A \times B$, denoted $\mu_1 R_\delta^* \mu_2$ if and only if there exist an $R$-coupling between them.
Validity of approximate Probabilistic Hoare judgments

We say that the quadruple $\vdash_\delta c_1 \sim c_2 : P \Rightarrow Q$ 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 $Q_{\delta^*}(\mu_1, \mu_2)$. 
Example of R-\(\delta\)-Coupling

\[
\begin{align*}
\mu_1 & \\
\text{OO} & 0.25 & \text{OO} & 0.20 \\
\text{O1} & 0.25 & \text{O1} & 0.25 \\
\text{1O} & 0.25 & \text{1O} & 0.25 \\
\text{11} & 0.25 & \text{11} & 0.30 \\
\end{align*}
\]
Example of R-δ-Coupling

\[ R(a, b) = \{ a=b \} \]
Example of R-$\delta$-Coupling

$$R(a, b) = \{ a=b \}$$

<table>
<thead>
<tr>
<th></th>
<th>$\mu_1$</th>
<th>$\mu_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>OO</td>
<td>0.25</td>
<td>0.20</td>
</tr>
<tr>
<td>O1</td>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td>10</td>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td>11</td>
<td>0.25</td>
<td>0.30</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>$\mu_L$</th>
<th>$\mu_R$</th>
</tr>
</thead>
<tbody>
<tr>
<td>OO</td>
<td>0.25</td>
<td>0.20</td>
</tr>
<tr>
<td>O1</td>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td>10</td>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td>11</td>
<td>0.25</td>
<td>0.30</td>
</tr>
</tbody>
</table>
Example of R-δ-Coupling

\[ R(a, b) = \{ a = b \} \]

\[ \Delta (\mu_L, \mu_R) = 0.05 \]
Example of R-δ-Coupling

<table>
<thead>
<tr>
<th></th>
<th>(\mu_1)</th>
<th></th>
<th>(\mu_2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>OO</td>
<td>0.2</td>
<td>OO</td>
<td>0</td>
</tr>
<tr>
<td>O1</td>
<td>0.25</td>
<td>O1</td>
<td>0.40</td>
</tr>
<tr>
<td>1O</td>
<td>0.25</td>
<td>1O</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0.3</td>
<td>11</td>
<td>0.6</td>
</tr>
</tbody>
</table>
Example of R-δ-Coupling

\[ R(a, b) = \{ a \leq b \} \]
Example of R-δ-Coupling

\[ R(a, b) = \{ a \leq b \} \]
Example of R-δ-Coupling

\[ R(a, b) = \{ a \leq b \} \]

\[ \Delta (\mu_L, \mu_R) = 0.05 \]
A more realistic example

StreamCipher (m : private msg[n]) : public msg[n]
  pkey := PRG(Uniform({0,1}^k));
  cipher := msg xor pkey;
  return cipher
### Example of R-δ-Coupling

<table>
<thead>
<tr>
<th></th>
<th>( \mu_1 )</th>
<th></th>
<th>( \mu_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>OO</td>
<td>0.25</td>
<td>OO</td>
<td>0</td>
</tr>
<tr>
<td>O1</td>
<td>0.25</td>
<td>O1</td>
<td>0</td>
</tr>
<tr>
<td>1O</td>
<td>0.25</td>
<td>1O</td>
<td>0.5</td>
</tr>
<tr>
<td>11</td>
<td>0.25</td>
<td>11</td>
<td>0.5</td>
</tr>
</tbody>
</table>
Example of R-δ-Coupling

\[ R(a, b) = \{ a = b \} \]
Example of R-δ-Coupling

\[ R(a, b) = \{ a = b \} \]

\[ \mu_1 \]

<table>
<thead>
<tr>
<th></th>
<th>OO</th>
<th>O1</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>OO</td>
<td>0.25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O1</td>
<td>0.25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0.25</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ \mu_2 \]

<table>
<thead>
<tr>
<th></th>
<th>OO</th>
<th>O1</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>OO</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ \mu_L \]

<table>
<thead>
<tr>
<th></th>
<th>OO</th>
<th>O1</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>OO</td>
<td>0.25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O1</td>
<td>0.25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0.25</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ \mu_R \]

<table>
<thead>
<tr>
<th></th>
<th>OO</th>
<th>O1</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>OO</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example of R-δ-Coupling

\[ R(a, b) = \{ a=b \} \]

\[ \Delta (\mu_L, \mu_R) = 0.5 \]
Probabilistic Relational Hoare Logic

Skip

\[ \vdash_0 \text{skip} \leadsto \text{skip}: P \Rightarrow P \]
\[ \vdash_0 x_1 := e_1 \sim x_2 := e_2 : \\
\text{P}[e_1<1>/x_1<1>, e_2<2>/x_2<2>] \Rightarrow \text{P} \]
Probabilistic Relational Hoare Logic
Composition

\[\vdash_0 c_1 \sim c_2 : P \Rightarrow R \quad \vdash_0 c_1' \sim c_2' : R \Rightarrow S\]

\[\vdash_0 c_1 ; c_1' \sim c_2 ; c_2' : P \Rightarrow S\]
Probabilistic Relational Hoare Logic Composition

\[ \vdash \delta_1 \overline{c_1 \sim c_2} : P \Rightarrow R \quad \vdash \delta_2 \overline{c_1' \sim c_2'} : R \Rightarrow S \]

\[ \vdash \delta_1 + \delta_2 \overline{c_1 ; c_1' \sim c_2 ; c_2'} : P \Rightarrow S \]
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 $R$.

We can **relax** $\delta_1$, i.e. replace it by something larger, in this case $\delta_2$. 

\[
\begin{align*}
P &\Rightarrow S \\
\vdash \delta_1 C_1 \sim C_2 : S &\Rightarrow R \\
R &\Rightarrow Q \\
\delta_1 &\leq \delta_2
\end{align*}
\]

\[
\begin{align*}
\vdash \delta_2 C_1 \sim C_2 : P &\Rightarrow Q
\end{align*}
\]
Probabilistic Relational Hoare Logic

If-then-else

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

\[ \vdash_\delta c_1 \sim c_2 : e_1<1> \land P \Rightarrow Q \]

\[ \vdash_\delta c_1' \sim c_2' : \neg e_1<1> \land P \Rightarrow Q \]

if \( e_1 \) then \( c_1 \) else \( c_1' \)

\[ \vdash_\delta \sim : P \Rightarrow Q \]

if \( e_2 \) then \( c_2 \) else \( c_2' \)
Probabilistic Relational Hoare Logic

If-then-else - left

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

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

\[ \vdash \delta \cdot \sim c_2 : P \Rightarrow Q \]
Probabilistic Relational Hoare Logic
A specific rule for PRG

\[ \vdash 2^{-n} \ x_1 := \$ \ \text{Uniform}({0,1}^n) \sim \ x_2 := \$ \ \text{PRG}(\text{Uniform}({0,1}^k)) \]

: True \Rightarrow x_1 <1> = x_2 <2>
How can we prove this secure?

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

\[
\text{StreamCipher}(m : \text{private msg}[n]) \\
\quad : \text{public msg}[n] \\
\quad \text{pkey} := \$ \text{PRG(Uniform}([0,1]^k)); \\
\quad \text{cipher} := m \text{ xor pkey}; \\
\quad \text{return cipher}
\]

We can apply the PRG rule, the composition rule, and the assignment rule and prove:

\[\vdash 2^{-n} \text{OneTimePad} \sim \text{StreamCipher} \hspace{1cm} : \text{m}^{<1>} = \text{m}^{<2>} \Rightarrow \text{c}^{<1>} = \text{c}^{<2>}\]