

CS 511: Formal Methods in Security and Privacy

Probabilistic Noninterference

Marco Gaboardi
gaboardi@bu.edu

Alley Stoughton
stough@bu.edu

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
```

$d_1, d_2, \dots$ probabilistic expressions

Semantics of Commands

This is defined on the structure of commands:

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

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

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

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

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

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

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

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.

Revisiting the example

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

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
```

Revisiting the example

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

m_1

m_2

Revisiting the example

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

m_1

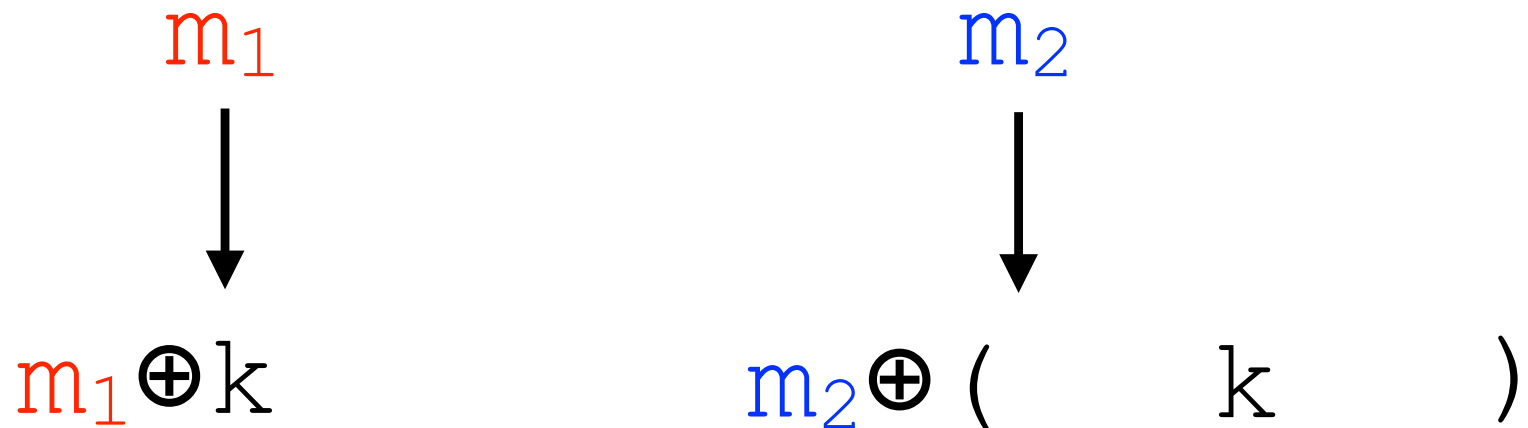
m_2



$m_1 \oplus k$

Revisiting the example

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



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?

Revisiting the example

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

$$\begin{array}{ccc} m_1 & & m_2 \\ \downarrow & & \downarrow \\ m_1 \oplus k & & m_2 \oplus (m_1 \oplus k \oplus m_2) \end{array}$$

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

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

Properties of bitwise xor

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

Example:

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

$$100 \oplus 001 = 101$$

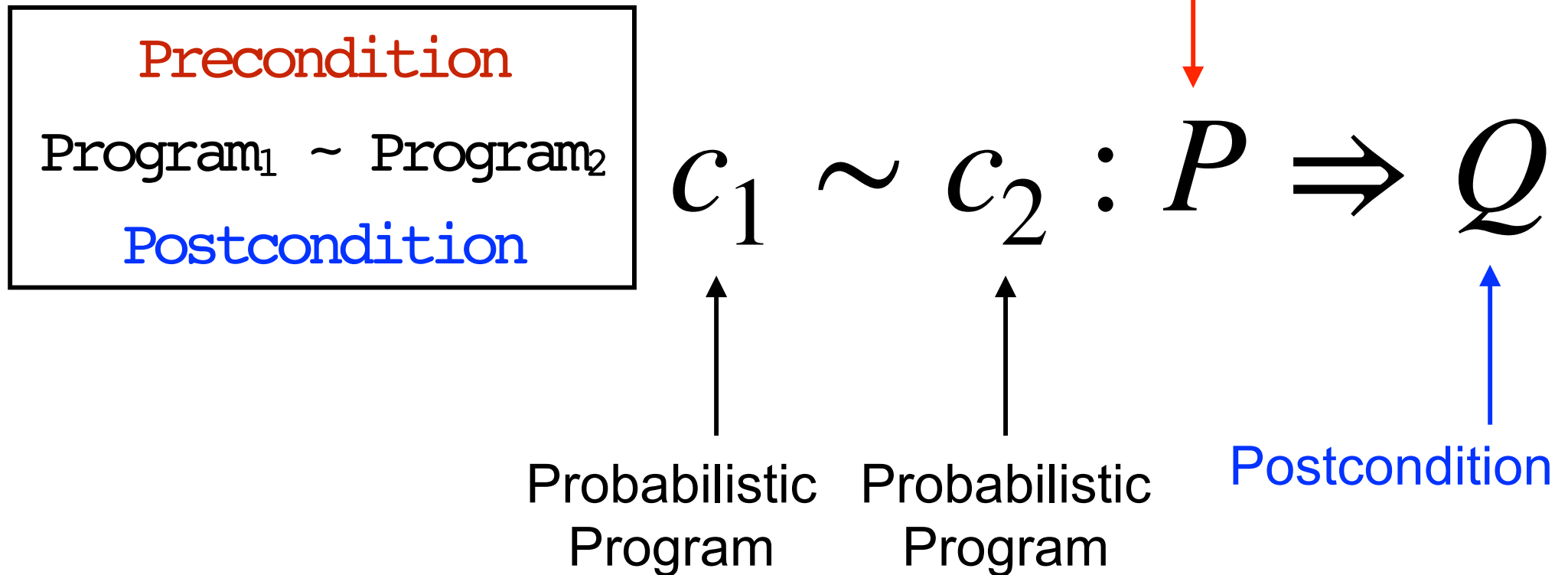
Revisiting the example

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



Applying the property above

Probabilistic Relational Hoare Quadruples



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
(i.e., relation over memories)

logical formula
over ????

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

k_1

relation

$$k_1 = 10 \oplus k_2 \oplus 00$$

00	0.25
01	0.25
10	0.25
11	0.25

k_2

Coupling Example 1

00	0.25
01	0.25
10	0.25
11	0.25

k_1

relation

$$k_1 = 10 \oplus k_2 \oplus 00$$

k_2

00	0.25
01	0.25
10	0.25
11	0.25

k_2

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

$$k_1 = k_2 \text{ or } k_1 = k_2 \oplus 11$$

00	0.00
01	0.50
10	0.00
11	0.50

k_2

Coupling Example 2

00	0.25
01	0.25
10	0.25
11	0.25

k_1

relation

$$k_1 = k_2 \text{ or } k_1 = k_2 \oplus 11$$

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

$$\pi_1(\mu)(a) = \sum_b \mu(a, b)$$

$$\pi_2(\mu)(b) = \sum_a \mu(a, b)$$

$$\pi_1(\mu) = \mu_1$$

$$\pi_2(\mu) = \mu_2$$

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 $\mu(a, b) > 0$, 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 $R \subseteq A \times B$, denoted $\mu_1 R^* \mu_2$, if and only if there exists a sub-distribution $\mu \in D(A \times B)$ such that:

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

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 $R \subseteq A \times B$, denoted $\mu_1 R^* \mu_2$, if and only if there exists a sub-distribution $\mu \in D(A \times B)$ such that:

- 1) if $\mu(a, b) > 0$, 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

Consequences of Lifting

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

$\mu_1 \ (E\langle 1 \rangle \Rightarrow F\langle 2 \rangle) * \mu_2,$

then we can conclude that

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

Consequences of Lifting

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

$\mu_1 \ (E\langle 1 \rangle \Rightarrow F\langle 2 \rangle) * \mu_2,$

then we can conclude that

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

Consequences of Lifting

Consequences of Lifting

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

$\mu_1 \ (E\langle 1 \rangle \Rightarrow F\langle 2 \rangle) * \mu_2,$

then we can conclude that

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

Consequences of Lifting

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

$\mu_1 (E\langle 1 \rangle \Rightarrow F\langle 2 \rangle) * \mu_2,$

then we can conclude that

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

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

Probabilistic Relational Hoare Logic

Skip

$$\vdash \text{skip} \sim \text{skip} : P \Rightarrow P$$

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

$$\vdash C_1 \sim C_2 : P \Rightarrow R \qquad \vdash C_1' \sim C_2' : R \Rightarrow S$$

$$\vdash C_1 ; C_1' \sim C_2 ; C_2' : P \Rightarrow S$$

Probabilistic Relational Hoare Logic

Consequence

$$\frac{P \Rightarrow S \quad \vdash C_1 \sim C_2 : S \Rightarrow R \quad R \Rightarrow Q}{\vdash C_1 \sim C_2 : P \Rightarrow Q}$$

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

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

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

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

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

Probabilistic Relational Hoare Logic

While

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

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

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

Probabilistic Relational Hoare Logic

If-then-else - left

$$\vdash c_1 \sim c_2 : e < 1 > \wedge P \Rightarrow Q$$

$$\vdash c_1' \sim c_2 : \neg e < 1 > \wedge P \Rightarrow Q$$

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

Probabilistic Relational Hoare Logic

If-then-else - right

$$\vdash c_1 \sim c_2 : e < 2 > \wedge P \Rightarrow Q$$

$$\vdash c_1 \sim c_2' : \neg e < 2 > \wedge P \Rightarrow Q$$

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

Probabilistic Relational Hoare Logic

Assignment - left

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

How about the random
assignment?

Probabilistic Relational Hoare Logic

Random Assignment

$$\vdash x_1 := \$ d_1 \sim x_2 := \$ d_2 : ?? \Rightarrow Q$$

We would like to have:

for all $m_1, m_2, P(m_1, m_2) \Rightarrow$
let $a = \{d_1\}_{m_1}$ in unit($m_1[x_1 \leftarrow a]$)
 Q^*
let $a = \{d_2\}_{m_2}$ in unit($m_2[x_2 \leftarrow a]$)

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

We would like to have:

for all $m_1, m_2, P(m_1, m_2) \Rightarrow$
let $a = \{d_1\}_{m_1}$ in unit($m_1[x_1 \leftarrow a]$)
 Q^*
let $a = \{d_2\}_{m_2}$ 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.

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

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.

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

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 $h:A \rightarrow B$ 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 $\text{supp}(\mu_1)$ and $\text{supp}(\mu_2)$,
- 2) for all $a \in A$, $\mu_1(a) = \mu_2(h(a))$

Probabilistic Relational Hoare Logic

Random Assignment

$P = h \triangleleft (d_1, d_2) \wedge$

$\forall v,$

$v \in \text{supp}(d_1)$

$\Rightarrow$

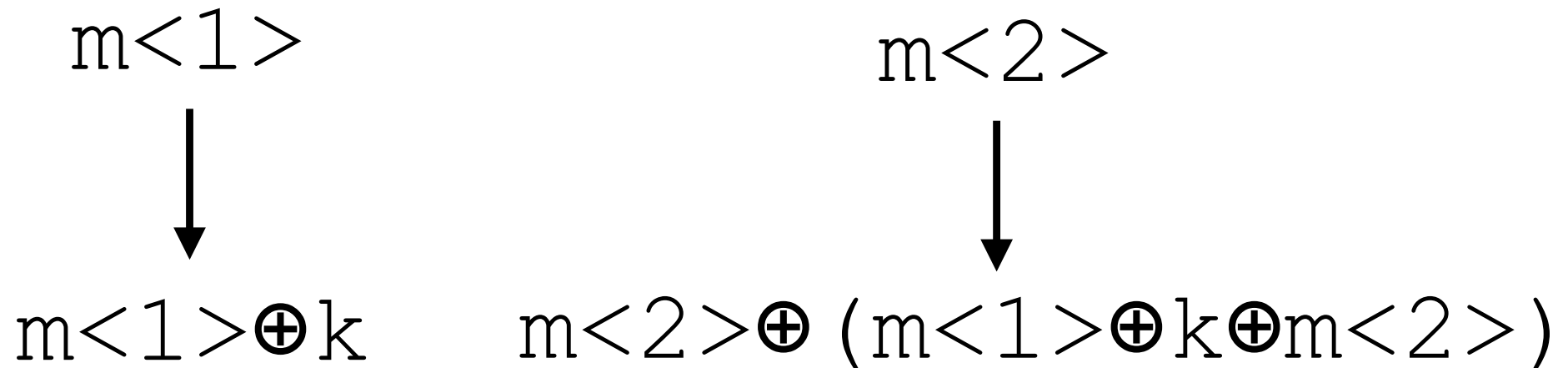
$Q[v/x_1\langle 1 \rangle, h(v)/x_2\langle 2 \rangle]$

we let h 's definition refer
to program variables tagged
with $\langle 1 \rangle$ or $\langle 2 \rangle$

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

Back to our example

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



Back to our example

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

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

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

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 $v \in \{0, 1\}^n$, $d_1(v) = d_2(h(v))$?

Back to our example

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

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

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

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

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

$$P = h \triangleleft (d_1, d_2) \wedge$$

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

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

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

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

Back to our example

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

$$P = h \triangleleft (d_1, d_2) \wedge$$

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

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

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

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

Back to our example

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

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

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

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

Back to our example

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

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

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

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

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?