

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

Probabilistic Expressions

We extend the language with expression describing probability distributions.

$$d ::= f(e_1, \dots, e_n, d_1, \dots, d_k)$$

Where f is a distribution declaration

Some expression examples

`uniform({0, 1}n)` `gaussian(k, σ)` `laplace(k, b)`

Semantics of Probabilistic Expressions

We would like to define it on the structure:

$$\{f(e_1, \dots, e_n, d_1, \dots, d_k)\}_m = \{f\}(\{e_1\}_m, \dots, \{e_n\}_m, \{d_1\}_m, \dots, \{d_k\}_m)$$

but is the result just a value?

Probabilistic Subdistributions

A **discrete subdistribution** over a set A is a function

$$\mu : A \rightarrow [0, 1]$$

such that the mass of μ ,

$$|\mu| = \sum_{a \in A} \mu(a)$$

verifies $|\mu| \leq 1$.

The support of a discrete subdistribution μ ,

$$\text{supp}(\mu) = \{a \in A \mid \mu(a) > 0\}$$

is necessarily countable, i.e. finite or countably infinite.

We will denote the set of sub-distributions over A by $D(A)$, and say that μ is of type $D(A)$ denoted $\mu:D(A)$ if $\mu \in D(A)$.

Probabilistic Subdistributions

We call a subdistribution with mass exactly 1, a **distribution**.

We define the **probability** of an event $E \subseteq A$ with respect to the subdistribution $\mu: D(A)$ as

$$\mathbb{P}_\mu[E] = \sum_{a \in E} \mu(a)$$

Probabilistic Subdistributions

Let's consider $\mu \in D(A)$, and $E \subseteq A$, we have the following properties

$$\mathbb{P}_\mu[\emptyset] = 0$$

$$\mathbb{P}_\mu[A] \leq 1$$

$$0 \leq \mathbb{P}_\mu[E] \leq 1$$

$$E \subseteq F \subseteq A \text{ implies } \mathbb{P}_\mu[E] \leq \mathbb{P}_\mu[F]$$

$$E \subseteq A \text{ and } F \subseteq A \text{ implies } \mathbb{P}_\mu[E \cup F] \leq \mathbb{P}_\mu[E] + \mathbb{P}_\mu[F] - \mathbb{P}_\mu[E \cap F]$$

We will denote by **0** the subdistribution μ defined as constant 0.

Operations over Probabilistic Subdistributions

Let's consider an arbitrary $a \in A$, we will often use the distribution $\text{unit}(a)$ defined as:

$$\mathbb{P}_{\text{unit}(a)}[\{b\}] = \begin{cases} 1 & \text{if } a=b \\ 0 & \text{otherwise} \end{cases}$$

We can think about unit as a function of type $\text{unit}: A \rightarrow D(A)$

Operations over Probabilistic Subdistributions

Let's consider a distribution $\mu \in D(A)$, and a function $M:A \rightarrow D(B)$ then we can define their composition by means of an expression $\text{let } a = \mu \text{ in } M a$ defined as:

$$\mathbb{P}_{\text{let } a = \mu \text{ in } M a}[E] = \sum_{a \in \text{supp}(\mu)} \mathbb{P}_{\mu}[\{a\}] \cdot \mathbb{P}_{(Ma)}[E]$$

Semantics of Probabilistic Expressions - revisited

We would like to define it on the structure:

$$\{f(e_1, \dots, e_n, d_1, \dots, d_k)\}_m = \{f\}(\{e_1\}_m, \dots, \{e_n\}_m, \{d_1\}_m, \dots, \{d_k\}_m)$$

With input a memory m and output a subdistribution $\mu \in D(A)$ over the corresponding type A . E.g.

$$\{\text{uniform}(\{0, 1\}^n)\}_{m \in D(\{0, 1\}^n)}$$

$$\{\text{gaussian}(k, \sigma)\}_{m \in D(\text{Real})}$$

Semantics of PWhile Commands

What is the meaning of the following command?

```
k := $ uniform({0, 1}^n); z := x mod k;
```

We can give the semantics as a function between **command**, **memories** and **subdistributions over memories**.

$$\text{Cmd} * \text{Mem} \rightarrow D(\text{Mem})$$

We will denote this relation as:

$$\{c\}_m = \mu$$

Semantics of Commands

This is defined on the structure of commands:

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

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

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

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

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

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

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

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: $\mu_1 \sim_{\text{low}} \mu_2$

Example: Low equivalence on distributions

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

$$\mu_1([x=2, y=0]) = 2/3, \quad \mu_1([x=3, y=1]) = 1/3$$

and

$$\mu_2([x=1, y=0]) = 1/3, \quad \mu_2([x=5, y=0]) = 1/3,$$

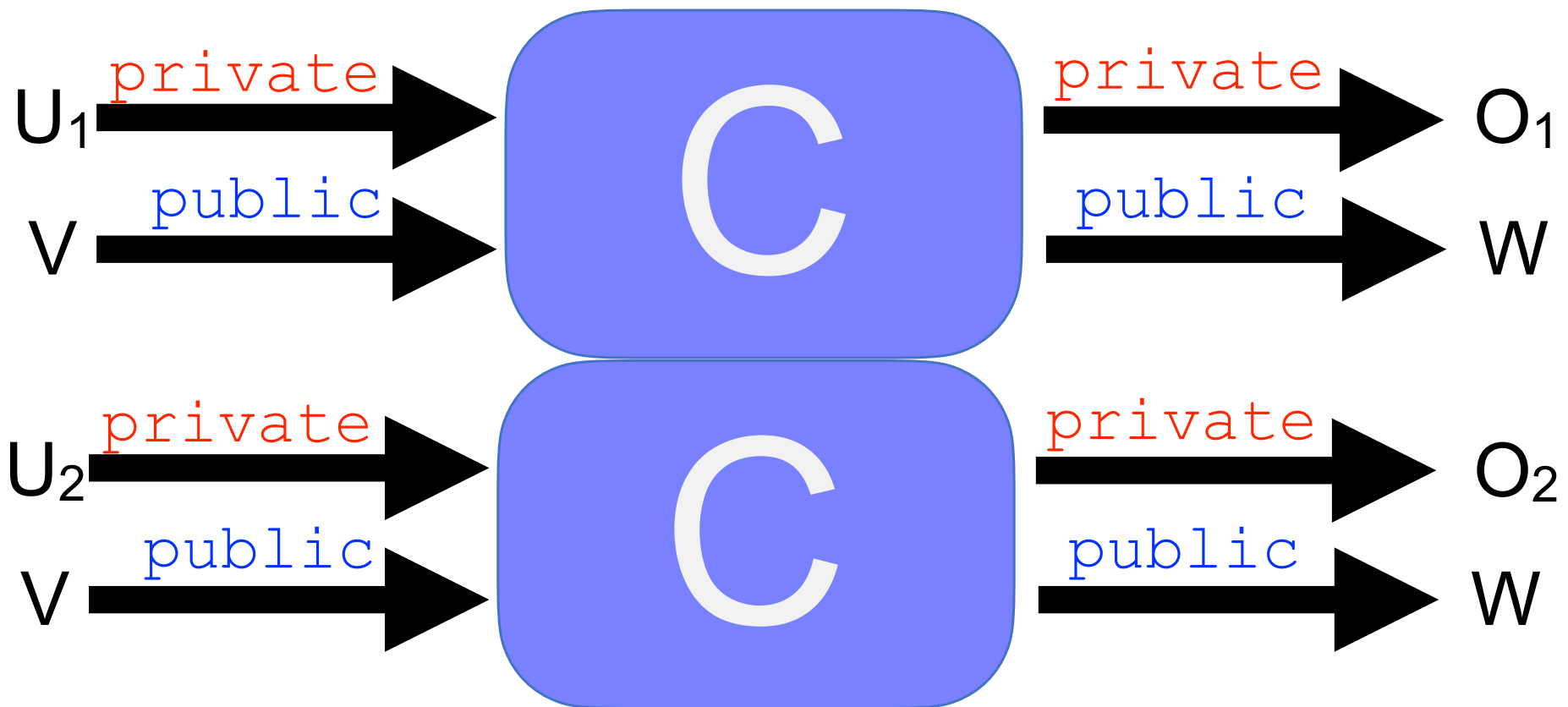
$$\mu_2([x=4, y=1]) = 1/3$$

are low equivalent.

Noninterference as a Relational Property

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

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



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?