A Connection between Indistinguishability and Semantic Security

Sven Laur

University of Tartu

1 Formal Definitions

Intuitively, if objects are indistinguishable then we cannot determine their personal properties which vary among the population. Indeed, if we could reliably determine wheter an object is green or not then we could easily distinguish green objects from yellow ones. Hence, indistinguishability indirectly implies that one can reliably detect only trivial properties that either hold or do not hold for the entire population. This basic argumentation template forms the cornerstone of contemporary cryptography. In the following, we fill out all details that are needed to convert this informal argumentation to a formal proof.

Let $\mathcal{S}$ be a distribution of secret values s and let $\mathrm{supp}(\mathcal{S})$ be the corresponding support. Then we can define indistinguishability of states $\mathrm{supp}(\mathcal{S})$ w.r.t. a function $f : \mathcal{S} \to \mathcal{X}$. We say that states from $\mathrm{supp}(\mathcal{S})$ are (t, ε)-*indistinguishable* if for any $s_0, s_1 \in \mathrm{supp}(\mathcal{S})$ and for any t-time algorithm $\mathcal{A}$:

$$\mathsf{Adv}^{\mathsf{ind}}_{s_0,s_1}(\mathcal{A}) = \left|\Pr\left[x \leftarrow f(s_0) : \mathcal{A}(x) = 1\right] - \Pr\left[x \leftarrow f(s_1) : \mathcal{A}(x) = 1\right]\right| \leq \varepsilon \enspace .$$

To define semantic security, we have to formalise which properties of a hidden state are plausible or relevant. Essentially, we can talk about semantic security w.r.t. *computable* functions $g : \mathcal{S} \to \mathcal{Y}$. Of course, for fixed input and output domains $\mathcal{S}$ and $\mathcal{Y}$ all functions are computable and thus this restriction is only a cosmetic addition that represents our intent. In some scenarios, there are obvious restrictions to the function g. For instance, we might require that the output of g must be computable during the next one hundred years.

Now note that for any function g there is a trivial predictor algorithm $\mathcal{A}_*$ that outputs the most probable outcome of g over the distribution $\mathcal{S}$ and thus the advantage of an algorithm $\mathcal{A}$ is defined as the difference

$$\Pr\left[s \leftarrow \mathcal{S}, x \leftarrow f(s) : \mathcal{A}(x) = g(s)\right] - \Pr\left[s \leftarrow \mathcal{S}, x \leftarrow f(s) : \mathcal{A}_*(x) = g(s)\right] \enspace .$$

More formally, we say that states $\mathrm{supp}(\mathcal{S})$ are (t, ε)-*semantically secure* w.r.t. functions f and g if for any t-time adversary $\mathcal{A}$ the corresponding advantage

$$\mathsf{Adv}^{\mathsf{sem}}_{f,g}(\mathcal{A}) = \Pr\left[\mathcal{A}(x) = g(s)\right] - \max_{y_* \in \mathcal{Y}} \Pr\left[s \leftarrow \mathcal{S} : g(s) = y_*\right] \leq \varepsilon \enspace .$$

2 Indistinguishability Implies Semantic Security

The main aim of this section is to prove the classical theorem about semantic security and introduce basic concepts of game-playing proofs.

Theorem 1. *If states from* $\mathrm{supp}(\mathcal{S})$ *are* $(2t, \varepsilon)$-indistinguishable *w.r.t. the function* f*, then states* $\mathrm{supp}(\mathcal{S})$ *are also* (t, ε)*-semantically for all functions* g*, i.e.,* $\mathsf{Adv}^{\mathsf{sem}}_{f,g}(\mathcal{A}) \leq \varepsilon$ *for all* t*-time adversaries* $\mathcal{A}$*.*

Proof. In order to present the proof modularly in easily understandable steps, we start from the basic semantic security game

```
$\mathcal{G}^{\mathcal{A}}$
    $s \leftarrow \mathcal{S}$
    $x \leftarrow f(s)$
    $y \leftarrow \mathcal{A}(x)$
    return $[y \overset{?}{=} g(s)]$
```

and then gradually rewrite the game until we obtain the desired bound on the success probability

$$\Pr\left[\mathcal{G}^{\mathcal{A}} = 1\right] \leq \Pr\left[\mathcal{G}^{\mathcal{A}_*} = 1\right] + \varepsilon \enspace .$$

Coin fixing. Note that g does not have to be a deterministic function. However, if $\mathsf{Adv}^{\mathsf{sem}}_{f,g}(\mathcal{A}) > \varepsilon$ for some randomised function $g : \mathcal{S} \times \Omega \to \mathcal{Y}$, then there exists a deterministic function g_* such that $\mathsf{Adv}^{\mathsf{sem}}_{f,g_*}(\mathcal{A}) > \varepsilon$. Indeed, by definition

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{sem}}_{f,g}(\mathcal{A}) &= \Pr\left[\mathcal{G}^{\mathcal{A}} = 1\right] - \Pr\left[\mathcal{G}^{\mathcal{A}_*} = 1\right] \\
&= \sum_{\omega_0 \in \Omega} \Pr\left[\omega \leftarrow \Omega : \omega = \omega_0\right] \cdot \left(\Pr\left[\mathcal{G}^{\mathcal{A}}_{\omega} = 1\right] - \Pr\left[\mathcal{G}^{\mathcal{A}_*}_{\omega} = 1\right]\right) \\
&\leq \max_{\omega \in \Omega} \left\{\Pr\left[\mathcal{G}^{\mathcal{A}}_{\omega} = 1\right] - \Pr\left[\mathcal{G}^{\mathcal{A}_*}_{\omega} = 1\right]\right\}
\end{aligned}$$

where the game $\mathcal{G}_\omega$ is defined as follows

```
$\mathcal{G}^{\mathcal{A}}_{\omega}$
    $s \leftarrow \mathcal{S}$
    $x \leftarrow f(s)$
    $y \leftarrow \mathcal{A}(x)$
    return $[y \overset{?}{=} g(s;\omega)]$
```

Since $g_\omega(\cdot) = g(\cdot;\omega)$ is a deterministic function, we have obtained

$$\mathsf{Adv}^{\mathsf{sem}}_{f,g}(\mathcal{A}) \leq \max_{\omega \in \Omega} \mathsf{Adv}^{\mathsf{sem}}_{f,g_\omega}(\mathcal{A})$$

and without loss of generality we can assume that g is deterministic. This kind of coin-fixing argument is common in many cryptographic proofs.

Sampling Idiom. As a next step, we will split the domain of $\mathcal{S}$ into a set of disjoint sub-domains that cover the entire domain

$$\mathcal{S}_y \doteq \{s \in \mathcal{S} : g(s) = y\} \quad .$$

Since g is deterministic the partition is indeed a well defined. Now a distribution $\mathcal{S}$ naturally defines a distribution over indices y. Let $\mathcal{Y}_0$ be the distribution over elements of $\mathcal{Y}$ such that for all $y_0 \in \mathcal{Y}$

$$\Pr\left[y \leftarrow \mathcal{Y}_0 : y = y_0\right] = \Pr\left[s \leftarrow \mathcal{S} : s \in \mathcal{S}_{y_0}\right] \quad .$$

Secondly, let $\mathcal{S}_y$ denote also a conditional distribution that emerges if we restrict the set of outputs to the domain $\mathcal{S}_y$, that is, for all $s_0 \in \mathcal{S}$ and $y \in \mathcal{Y}$:

$$\Pr\left[s \leftarrow \mathcal{S}_y : s = s_0\right] = \Pr\left[s \leftarrow \mathcal{S} : s = s_0 | g(s) = y\right] \quad .$$

Then the sampling procedure $s \leftarrow \mathcal{S}$ can be rewritten in two steps

```
$\mathcal{G}_*^{\mathcal{A}}$
    $y \leftarrow \mathcal{Y}_0$
    $s \leftarrow \mathcal{S}_y$
    $x \leftarrow f(s)$
    $y \leftarrow \mathcal{A}(x)$
    return $[y \overset{?}{=} g(s)]$
```

so that the success probability does not change for any adversary $\mathcal{A}$. Indeed

$$\begin{aligned} \Pr\left[\mathcal{G}^{\mathcal{A}} = 1\right] &= \sum_{y \in \mathcal{Y}} \Pr\left[s \leftarrow \mathcal{S} : g(s) = y_0\right] \Pr\left[\mathcal{G}^{\mathcal{A}} = 1 | g(s) = y_0\right] \\ &= \sum_{y \in \mathcal{Y}} \Pr\left[y \leftarrow \mathcal{Y}_0 : y = y_0\right] \Pr\left[\mathcal{G}_{y_0}^{\mathcal{A}} = 1\right] \end{aligned}$$

where the game $\mathcal{G}_{y_0}$ is defined as follows

```
$\mathcal{G}_{y_0}^{\mathcal{A}}$
    $s \leftarrow \mathcal{S}_{y_0}$
    $x \leftarrow f(s)$
    $y \leftarrow \mathcal{A}(x)$
    return $[y \overset{?}{=} g(s)]$
```

To be punctual, one has to use the total probability formula and the definition of conditional probabilities to formally prove

$$\Pr\left[\mathcal{G}^{\mathcal{A}} = 1 | g(s) = y_0\right] \equiv \Pr\left[\mathcal{G}_{y_0}^{\mathcal{A}} = 1\right]$$

but this is a trivial exercise that is left to the reader.

Choosing between several simple hypothesis. Already a superficial inspection of the game $\mathcal{G}_*$ reveals that an adversary $\mathcal{A}$ must choose between multiple simple hypotheses $\mathcal{H}_{y_0} = [y \overset{?}{=} y_0]$. As a next step, we can express

$$\begin{aligned}\Pr\left[\mathcal{G}_*^{\mathcal{A}} = 1\right] &= \Pr\left[y \leftarrow \mathcal{Y}_0 : y = y_1\right]\Pr\left[\mathcal{G}_{y_1}^{\mathcal{A}} = 1\right]\\ &\quad + \sum_{y_0 \in \mathcal{Y} \setminus y_1} \Pr\left[y \leftarrow \mathcal{Y}_0 : y = y_0\right]\Pr\left[\mathcal{G}_{y_0}^{\mathcal{A}} = 1\right]\end{aligned}$$

where y_1 is the most probable index element. Now note that

$$\begin{aligned}\Pr\left[y_1\right]\Pr\left[\mathcal{G}_{y_1}^{\mathcal{A}} = 1\right] &= \Pr\left[y_1\right] - \Pr\left[y_1\right] \cdot \sum_{y_0 \in \mathcal{Y} \setminus y_1} \Pr\left[s \leftarrow \mathcal{S}_{y_1} : \mathcal{A}(f(s)) = y_0\right]\\ &\leq \Pr\left[y_1\right] - \sum_{y_0 \in \mathcal{Y} \setminus y_1} \Pr\left[y_0\right]\Pr\left[s \leftarrow \mathcal{S}_{y_1} : \mathcal{A}(f(s)) = y_0\right]\end{aligned}$$

since $\Pr\left[y_0\right] \leq \Pr\left[y_1\right]$ by the choice of y_1. Consequently,

$$\begin{aligned}\Pr\left[\mathcal{G}_*^{\mathcal{A}} = 1\right] &\leq \Pr\left[y_1\right] + \sum_{y_0 \in \mathcal{Y} \setminus y_1} \Pr\left[y_0\right]\left|\Pr\begin{bmatrix} s \leftarrow \mathcal{S}_{y_0} : \\ \mathcal{A}(\cdot) = y_0 \end{bmatrix} - \Pr\begin{bmatrix} s \leftarrow \mathcal{S}_{y_1} : \\ \mathcal{A}(\cdot) = y_0 \end{bmatrix}\right|\\ &\leq \Pr\left[y_1\right] + \max_{y_0 \in \mathcal{Y}}\left|\Pr\begin{bmatrix} s \leftarrow \mathcal{S}_{y_0} : \\ \mathcal{A}(\cdot) = y_0 \end{bmatrix} - \Pr\begin{bmatrix} s \leftarrow \mathcal{S}_{y_1} : \\ \mathcal{A}(\cdot) = y_0 \end{bmatrix}\right| \enspace .\end{aligned}$$

From complex hypotheses to simple hypotheses. Note that in terms of $\mathcal{S}$ the last term in the upper bound obtained above is very close to a computational distance between complex hypotheses $[s \leftarrow \mathcal{S}_{y_0}]$ and $[s \leftarrow \mathcal{S}_{y_1}]$.

In fact, if we know the maximising value y_0, then we can convert a t-time algorithm $\mathcal{A}$ that maximises the term into $2t$-time distinguisher $\mathcal{B} : \mathcal{X} \rightarrow \{0, 1\}$. As y_0 can be at most t-bits long, we can test $[\mathcal{A}(f(s)) \overset{?}{=} y_0]$ in t-time and consequently we can build $2t$-time algorithm $\mathcal{B}$ such that

$$\begin{aligned}p_0 &\doteq \Pr\left[s \leftarrow \mathcal{S}_{y_0} : \mathcal{B}(f(s)) = 1\right] = \Pr\left[s \leftarrow \mathcal{S}_{y_0} : \mathcal{A}(f(s)) = y_0\right] \enspace ,\\ p_1 &\doteq \Pr\left[s \leftarrow \mathcal{S}_{y_1} : \mathcal{B}(f(s)) = 1\right] = \Pr\left[s \leftarrow \mathcal{S}_{y_1} : \mathcal{A}(f(s)) = y_0\right] \enspace .\end{aligned}$$

Since

$$\begin{aligned}p_0 &= \sum_{\substack{s_0 \in S_{y_0}\\ s_1 \in S_{y_1}}} \Pr\left[s \leftarrow \mathcal{S}_{y_0} : s = s_0\right]\Pr\left[s \leftarrow \mathcal{S}_{y_1} : s = s_1\right]\Pr\left[\mathcal{B}(f(s_0)) = 1\right]\\ p_1 &= \sum_{\substack{s_0 \in S_{y_0}\\ s_1 \in S_{y_1}}} \Pr\left[s \leftarrow \mathcal{S}_{y_0} : s = s_0\right]\Pr\left[s \leftarrow \mathcal{S}_{y_1} : s = s_1\right]\Pr\left[\mathcal{B}(f(s_1)) = 1\right]\end{aligned}$$

we can estimate

$$
\begin{aligned}
|p_0 - p_1| &= \left| \sum_{\substack{s_0 \in S_{y_0}\\ s_1 \in S_{y_1}}} \Pr[s_0]\Pr[s_1]\left(\Pr[\mathcal{B}(f(s_0))=1] - \Pr[\mathcal{B}(f(s_1))=1]\right) \right| \\
&\le \sum_{\substack{s_0 \in S_{y_0}\\ s_1 \in S_{y_1}}} \Pr[s_0]\Pr[s_1]\left|\Pr[\mathcal{B}(f(s_0))=1] - \Pr[\mathcal{B}(f(s_1))=1]\right| \\
&\le \max_{\substack{s_0 \in S_{y_0}\\ s_1 \in S_{y_1}}} \left|\Pr[\mathcal{B}(f(s_0))=1] - \Pr[\mathcal{B}(f(s_1))=1]\right| \\
&\le \max_{s_0, s_1 \in \mathcal{S}} \mathsf{cd}_x^{2t}([s \overset{?}{=} s_0], [s \overset{?}{=} s_1]) \le \varepsilon \ .
\end{aligned}
$$

The final step. To summarise, we have obtained

$$
\begin{aligned}
\Pr\left[\mathcal{G}^{\mathcal{A}} = 1\right] = \Pr\left[\mathcal{G}_*^{\mathcal{A}} = 1\right] &\le \max_{y_0 \in \mathcal{Y}} \Pr[y \leftarrow \mathcal{Y}_0 : y = y_0] + \varepsilon \\
&\le \max_{y_0 \in \mathcal{Y}} \Pr[s \leftarrow \mathcal{S} : g(s) = y_0] + \varepsilon \ .
\end{aligned}
$$

□

3 Final Remarks

Note that the proof given above is strictly non-constructive and does not show how to convert a good predictor of g-values into a good distinguisher of hidden states $s_0, s_1 \in \mathcal{S}$. Non-constructivity of the proof is simultaneously the main strength and weakness of this approach. Theorem 1 is extremely powerful as a mathematical claim, since it assumes nothing from the sample distribution $\mathcal{S}$ and holds for all functions g. As a result, non-constructivity is essential in the proof, since we cannot assume that elements of $\mathcal{S}$ can be *efficiently* sampled neither we can assume that the function g is *efficiently* computable.

To be precise, non-constructivity enters into the proof in three places. First, we fix random coins ω so that $g(\cdot, \omega)$ behaves better than $g(\cdot)$. Second, we fix a most probable output y_1 and sub-distributions $\mathcal{S}_y$ for $y \in \mathcal{Y}$. Third, we use two proof steps for finding an output $y_0 \in \mathcal{Y}$ and two states $s_0 \in \mathcal{S}_{y_0}$, $s_1 \in \mathcal{S}_{y_1}$ that maximises the difference $|\Pr[\mathcal{A}(f(s_0)) = y_0] - \Pr[\mathcal{A}(f(s_1)) = y_0]|$.

For a fixed distribution $\mathcal{S}$ and a fixed function $g(\cdot)$, the complexity of these non-constructive steps is irrelevant. For a hypothetical t-time algorithm $\mathcal{A}$ that achieves $\mathsf{Adv}^{\mathsf{sem}}_{f,g}(\mathcal{A}) > \varepsilon$, there exist states $s_0, s_1 \in \mathcal{S}$ and a corresponding $2t$-time algorithm $\mathcal{B}$ such that $\mathsf{Adv}^{\mathsf{ind}}_{s_0,s_1}(\mathcal{B}) \ge \varepsilon$ and consequently the existence of such a t-time algorithm $\mathcal{A}$ directly contradicts the security premise.

However, the efficiency is important if the distribution $\mathcal{S}$ is not fixed beforehand and a contradiction triple $(\mathcal{B}, s_0, s_1)$ must be discovered on the fly. For instance, Theorem 1 does not cover semantic security of ciphertexts in the settings, where an adversary can influence which messages are encrypted. The latter is not so far-fetched assumption. For instance, communication in a war is mostly about the adversarial behaviour and thus clearly controllable by adversary.

To prove security in such settings, we have to weaken the claim so that the construction in the proof would become efficiently constructable. In particular, the distribution $\mathcal{S}$ must be efficiently samplable and g efficiently computable. Then we can convert the original non-constructive proof into a constructive reduction. The details of this approach are thoroughly discussed in [BDJR97].

References

[BDJR97] Mihir Bellare, Anand Desai, E. Jokipii, and Phillip Rogaway. A concrete security treatment of symmetric encryption. Available from http://www-cse.ucsd.edu/users/mihir/papers/sym-enc.html, 1997. Full version of the FOCS publication.