MPRI 2.30: Proofs of Security Protocols

TD: Relations Among Hash Functions Cryptographic Assumptions

Adrien Koutsos

2023/2024

Questions marked with a star ($\star$) can be omitted without impacting the rest of the exercise.

1 Hash Functions

Let $\Sigma = \{0, 1\}$. A cryptographic hash function $\mathsf{H} : \Sigma^* \mapsto \Sigma^L$ allows to compute, for every message m, a *digest* $\mathsf{H}(m)$ – often called the *hash* – of fixed length L.[1] Examples of such functions are SHA-2, or the more recent SHA-3.

There are many security properties that we may want from a cryptographic hash function. A common property is to require that the hash function has no **collision**, where a collision is a pair of distinct messages m_0, m_1 such that $\mathsf{H}(m_0) = \mathsf{H}(m_1)$. Of course, for cardinality reasons, this cannot be achieved.

Therefore, we are going to slightly change the setting. A *keyed* cryptographic hash function $\mathsf{H} : \Sigma^* \times \Sigma^K \mapsto \Sigma^L$ takes as input a message m of any length and a key k of length K, and compute the hash of m under k. A keyed hash function could be implemented, for example, by taking $\mathsf{H}(m, k) \stackrel{\text{def}}{=} \mathsf{SHA\text{-}3}(k||m)$. To simplify things, we assume $K = L = \eta$ from now on.

2 Hardness Hypotheses on Hash Functions

We now present three different security notions for keyed hash functions.

Collision-Resistance A keyed cryptographic hash $\mathsf{H}(\_, \_)$ is computationally collision resistant if no PPTM adversary can built collisions, even when it has access to a hashing oracle.

Formally, a hash is *collision resistant under hidden key attacks* (CR-HK) iff. for every PPTM $\mathcal{A}$:

$$\mathsf{Pr}_{\mathsf{k}}\left(\mathcal{A}^{\mathcal{O}_{\mathsf{H}(\cdot,\mathsf{k})}}(1^\eta) = \langle m_1\,,\, m_2\rangle, m_1 \neq m_2 \text{ and } \mathsf{H}(m_1, \mathsf{k}) = \mathsf{H}(m_2, \mathsf{k})\right)$$

is negligible, where k is drawn uniformly in $\{0, 1\}^\eta$.

Unforgeability A keyed hash function is computationally unforgeable when no adversary can forge new hashes, even when the adversary has access to a hashing oracle.

Formally, a hash is *unforgeable against chosen-message attacks* (EUF-CMA) iff. for every PPTM $\mathcal{A}$:

$$\mathsf{Pr}_{\mathsf{k}}\left(\mathcal{A}^{\mathcal{O}_{\mathsf{H}(\cdot,\mathsf{k})}}(1^\eta) = \langle m\,,\, \sigma\rangle,\ m \text{ not queried to } \mathcal{O}_{\mathsf{H}(\cdot,\mathsf{k})} \text{ and } \sigma = \mathsf{H}(m, \mathsf{k})\right)$$

is negligible, where k is drawn uniformly in $\{0, 1\}^\eta$.

Pseudo-Random Function A keyed hash function $\mathsf{H}(\cdot, \mathsf{k})$ is a PRF if its outputs are computationally indistinguishable from the outputs of a random function.

Formally, a hash function is a *Pseudo Random Function* iff. for any PPTM $\mathcal{A}$:

$$\left|\mathsf{Pr}_{\mathsf{k}}(\mathcal{A}^{\mathcal{O}_{\mathsf{H}(\cdot,\mathsf{k})}}(1^\eta) = 1) - \mathsf{Pr}_g(\mathcal{A}^{\mathcal{O}_{g(\cdot)}}(1^\eta) = 1)\right|$$

is negligible, where:

- k is drawn uniformly in $\{0, 1\}^\eta$.
- g is a random function from $\{0, 1\}^*$ to $\{0, 1\}^\eta$.

[1] L is more or less the security parameter.

2.1 Relations Among Security Notions and Rule Schemata

Show that we have the following relations among keyed hash function security notions.

Exercise 1 ($\star$)**.** *Show that PRF $\Rightarrow$ EUF-CMA $\Rightarrow$ CR-HK.*

We now consider the problem of designing sound rules of the indistinguishability logic capturing these different keyed hash function security notions.

Exercise 2. *Design and prove sound a rule schemata for CR-HK.*

Exercise 3. *Design and prove sound a rule schemata for PRF. In a first time, assume that there are at most two calls to the hash oracle. Then, generalize to any number of calls.*

2.2 EUF Rule and Variation

If H is an $\mathsf{EUF\text{-}CMA}$ keyed hash function, then the *ground* rule:

$$\frac{}{\left(s \doteq \mathsf{H}(m, \mathsf{k}) \mathrel{\dot{\rightarrow}} \dot{\bigvee}_{u \in \mathcal{S}} m \doteq u\right) \sim \mathsf{true}}\ \mathsf{EUF}$$

is sound, when:

- $\mathcal{S} = \{u \mid \mathsf{H}(u, \mathsf{k}) \in \mathsf{st}(s, m)\}$;
- k appears only in H key positions in s, m, i.e. $\mathsf{k} \sqsubseteq_{\mathsf{H}(\_,\cdot)} s, m$.

We assume that the EUF rule given above is sound. We are now going to prove an improved, more precise, version of the rule.

Ignoring Hashes in Conditions We show that we can ignore some hashes appearing in conditions in s or m. To simplify matter, we only do it for a single condition.

Exercise 4. *Assume that H is EUF-CMA. Show that the following rule is sound:*

$$\frac{}{(\mathit{if}\ b\ \mathit{then}\ s_0\ \mathit{else}\ s_1) \doteq H(m, k) \mathrel{\dot{\rightarrow}} \dot{\bigvee}_{u \in \mathcal{S}_1 \cup \mathcal{S}_2} m \doteq u \sim \mathit{true}}\ \mathit{EUF}_{\mathit{nc}}$$

when b, s_0, s_1, m are ground *terms, and:*

- *$\mathcal{S}_i = \{u \mid H(u, k) \in st(s_i, m)\}$, for $i \in \{0, 1\}$;*
- *k appears only in H key positions in s_0, s_1, m.*

Remark that we do not make *any* assumption on b, except that it is ground. E.g., we can have $b \equiv (\mathsf{att}(\mathsf{k}) \doteq \mathsf{H}(0, \mathsf{k}))$.

Exercise 5 ($\star$)**.** *What is the relation between the advantage against EUF_{nc} and the advantage against the EUF-CMA security assumption? How would this advantage evolve if we generalized the EUF_{nc} rule to N conditions $b_1, \ldots, b_n$?*