MAD 4471: Cryptography and Coding Theory **Fall 2020**

Lecture 4: Hash Functions and the Birthday Paradox

Lecturer: Jean-François Biasse *TA: William Youmans*

4.1 Hash functions

A hash function is a function $H : \mathcal{M} \rightarrow \mathcal{T}$ where typically $|\mathcal{M}| \gg |\mathcal{T}|$.

Definition. H is collision - resistant if there is no efficient algorithm $\mathcal{A}$ that can find $m_0, m_1 \in \mathcal{M}$ such that $H(m_0) = H(m_1)$ with non negligible probability.

We can use hash functions to derive MACs. The trivial construction consisting in defining:

- $S(m, k) = H(k||m)$
- $V(m, k, t) = \text{true if } t = H(k||m)$

is not secure because the Merkel-Darmgard construction to hash messages of arbitrary length easily allows an adversary to compute $H(k||m||\text{something new})$ from $H(k||m)$ without knowing k, which consistutes a valid forgery in the MAC security game (even though this is not a collision for the hash function since we don't necessarily have that $H(k||m||\text{something new}) = H(k||m)$. Instead, HMAC repeats this construction twice:

- $S(m, k) = H(k \oplus \text{opad}||H(k \oplus \text{ipad}||m)$
- $V(m, k, t) = \text{true if } t = H(k \oplus \text{opad}||H(k \oplus \text{ipad}||m)$,

where ipad and opad are fixed (public) bit strings.

4.2 Finding Collisions

To ensure the security of HMAC, we must use collision- resistant hash functions. Let $H : \mathcal{M} \rightarrow \mathcal{T}$ be a hash function. There is a trivial way to find messages in $\mathcal{M}$ with the same tag in $\mathcal{T}$ (i.e. to find collitions). It consists in drawing elements of $\mathcal{M}$ at random until we find one. It is not very smart, but the expected number of trials before finding a collision is on average significantly less than $N := |\mathcal{T}|$. In the worst case however, one might have to draw $N + 1$ messages in $\mathcal{M}$ before obtaining a collision, but this statistically never happens. This phenomenon is called the "Birthday paradox".

Theorem. Let $0 < x < 1$. If we draw $n \geq \sqrt{2 \ln\left(\frac{1}{1-x}\right)}\sqrt{N} + 1$ elements uniformly at random in $\mathcal{M}$, the probability of finding a collision is at least x.

Proof. Let us calculate the probability of <u>not</u> finding a collision after trying n times.

$$\begin{aligned}
Pr(\text{no collision}) &= (\frac{N-1}{N})(\frac{N-2}{N})....(\frac{N-n+1}{N}) \\
&= \prod_{i=1}^{n-1}(1-\frac{i}{N}) \\
&\leq \prod_{i=1}^{n-1} e^{\frac{-i}{N}} \text{because} 1-y \leq e^{-y} \\
&= e^{\sum_{i=1}^{n-1}\frac{-i}{N}} = e^{\frac{-n(n-1)}{2N}} \leq e^{\frac{-(n-1)^2}{2N}}
\end{aligned}$$

Therefore, the probability of finding a collision satisfies:

$$\begin{aligned}
Pr(\text{collision}) &= 1 - Pr(\text{no collision}) \\
&\geq 1 - e^{\frac{-(n-1)^2}{2N}}
\end{aligned}$$

To ensure that this probability be at least x it suffices that

$$\begin{aligned}
e^{\frac{-(n-1)^2}{2N}} \leq 1-x &\iff \frac{-(n-1)^2}{2N} \leq \ln(1-x) \\
&\iff (n-1)^2 \geq 2\ln\left(\frac{1}{1-x}\right)\cdot N \\
&\iff n \geq \sqrt{2\ln\left(\frac{1}{1-x}\right)}\cdot\sqrt{N}+1
\end{aligned}$$

This means that the size of $\mathcal{T}$ must account for the "birthday attack". If we want to make some that an attack take at least 2^{128} operations, $|\mathcal{T}|$ must be at least 2^{256} (which is the case for SHA256).