CS 346 Class Notes

Mark Lindberg

Mar 7, 2016

Last Time:
5.3 Message Authentication using Hash functions.
"Hash and Mac" paradigm.
Construction 5.5:
$\Pi = (\mathsf{Mac}, \mathsf{Vrfy})$. A fixed-length MAC for messages of length $\ell(n)$.
$\Pi_H = (\mathsf{Gen}_H, H)$. A hash function with output length $\ell(n)$.
Construct Mac: $\Pi' = (\mathsf{Gen}', \mathsf{Mac}', \mathsf{Vrfy}')$.
Gen': Run Gen_H to get s. Also get random n-bit k. Key is (s, k).
$\mathsf{Mac}'_{s,k}(m) = \mathsf{Mac}_k(H^s(m))$.
$\mathsf{Vrfy}'_{s,k}(m) = \mathsf{Vrfy}_k(H^s(m), t)$.
This Time:
Theorem 5.6: If Π is secure and Π_H is collision resistant, then Π' is secure.
Proof: Let $\mathcal{A}'$ be an arbitrary PPT adversary in the $\mathsf{Mac\text{-}forge}_{\mathcal{A}',\Pi'}(n)$ experiment.
Split the $\mathcal{A}'$ successes into "Type I" and "Type II".
$\mathcal{A}'$ succeeds if it produces (m, t), such that $m \notin Q$ (set of messages $\mathcal{A}'$ has previously made oracle calls on), and $\mathsf{Vrfy}'_{s,k}(m, t) = 1$. From Vrfy' definition, $\mathsf{Vrfy}_k(H^s(m), t) = 1$.
Note that $\Pr[\mathcal{A}' \text{ succeeds}] = \Pr[\mathcal{A}' \text{ has Type I success}] + \Pr[\mathcal{A}' \text{ has Type II success}]$.

$$\begin{aligned} \text{Type I} &: H^s(m) = H^s(m') \text{ for some } m' \in Q. \\ \text{Type II} &: \text{Otherwise.} \end{aligned}$$

Type I breaks collision resistance of Π_H. We can then relate Type II to breaking Π's security.
Type I:
To prove: $\Pr[\mathcal{A}' \text{ has Type I success}]$ is `negl`:
Construct a PPT adversary $\mathcal{A}_H$ in $\mathsf{Hash\text{-}coll}_{\mathcal{A},\Pi}(n)$ that simulates $\mathcal{A}'$.
$\mathcal{A}_H$ gets s and outputs m, m'. It succeeds iff $m \neq m'$ and $H^s(m) = H^s(m')$.
To simulate $\mathcal{A}'$. $\mathcal{A}'$ calls oracle $\mathsf{Mac}'_{k,s}(m) = \mathsf{Mac}_k(H^s(m))$ at outset, $\mathcal{A}_H$ generates a random n-bit k when $\mathcal{A}'$ outputs (m, t).
If $\mathcal{A}'$ does not have a Type I success, $\mathcal{A}_H$ outputs arbitrary messages.
Otherwise, output m, m' such that $H^s(m) = H^s(m')$ and $m' \in Q$.

Type II:

To prove: $\Pr[\mathcal{A}'$ has Type II success$]$ is $\texttt{negl}$.

Construct a PPT adversary $\mathcal{A}$ for $\mathsf{Mac\text{-}forge}_{A,\Pi}(n)$. $\mathcal{A}$ simulates $\mathcal{A}'$.

To simulate a call to $\underbrace{\mathsf{Mac}'_{s,k}(m)}_{\text{Oracle of } \mathcal{A}'} = \underbrace{\mathsf{Mac}_k(H^s(m))}_{\text{Oracle of } \mathcal{A}}$.

At outset, $\mathcal{A}$ run $\mathsf{Gen}_H(1^n)$ to get s. When $\mathcal{A}'$ outputs m, t, if $\mathcal{A}'$ does not get a Type II success, give an arbitrary output.

Otherwise, $\mathcal{A}$ outputs $(H^s(m), t)$. This will pass Vrfy. Note that $H^s(m) \neq H^s(m')$ for any $m' \neq m$, $m, m' \in Q$, because it's a Type II success, and therefore cannot be a Type I success.

Done!

HMAC construction!

Um, I took a picture. That thing was ridiculous.

Generic birthday attacks on Hash functions.

THE BIRTHDAY PARADOX IS NOT A FREAKING PARADOX, DANGIT.

On a hash function with $\ell(n)$-bit output strings, $O(2^{\frac{\ell(n)}{2}})$ evaluations are sufficient to find a collision with good probability.

Analysis: Assume idealized Hash function (worst case), n bins, throw balls into random bins until we have a collision. Expected time, $\Theta(\sqrt{n})$. Hehehe. This is mathematically provable, and makes perfect sense probabilistically, and there is no paradox. Period. *sigh*

Constant space birthday attack: Pick an IV, and keep hashing the output, and try to see when it loops on itself.

There's a solution to this. Huh.