NMAC: Security Proof

Hyrum Mills, Chris Soghoian, Jon Stone, Malene Wang

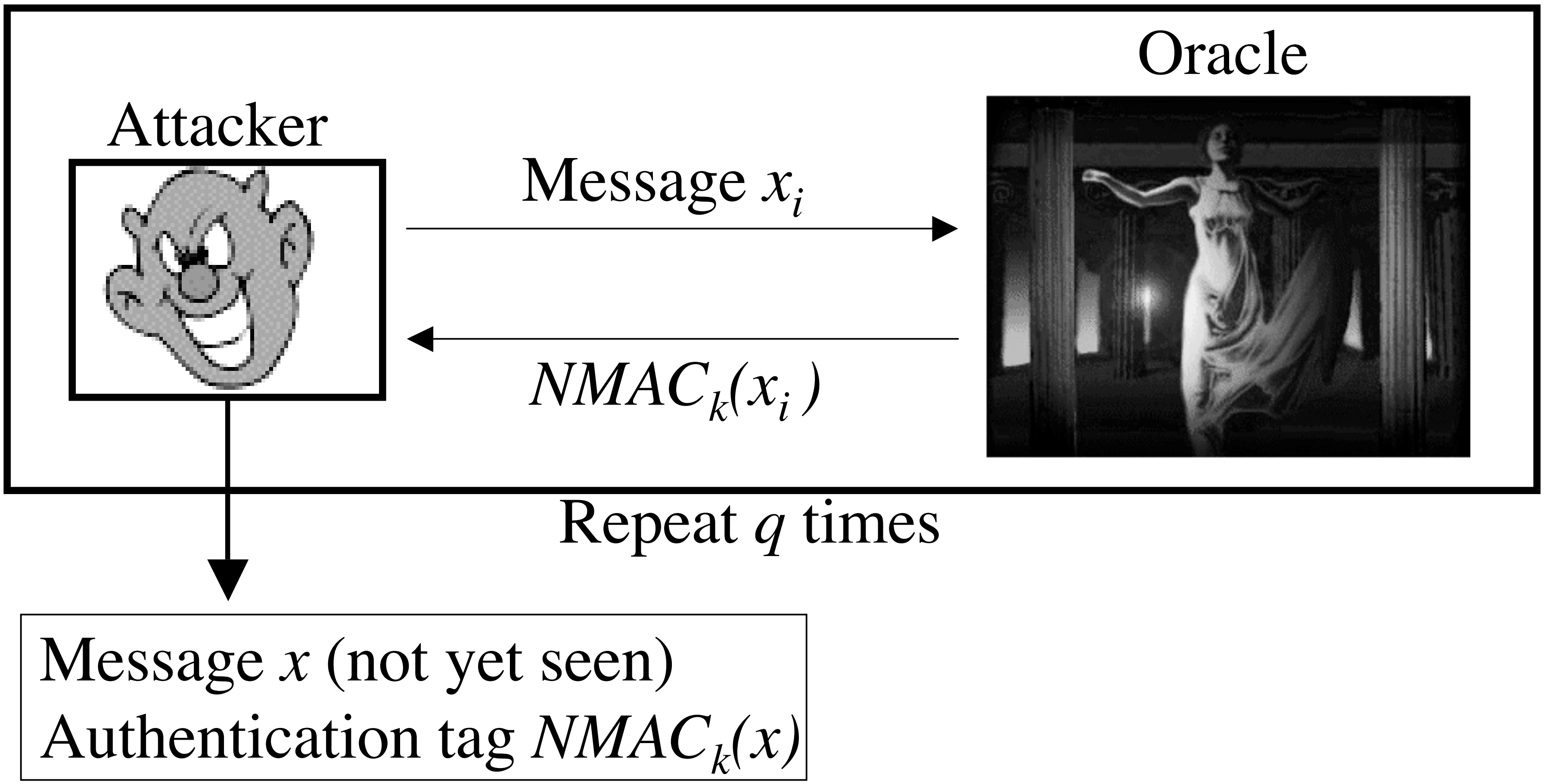
Theorem 4.1

- If the keyed compression function f is an (ε_f, q, t, b)-secure MAC on messages of length b bits, and the keyed iterated hash F is (ε_F, q, t, L)-weakly collision-resistant, then the NMAC function is an ($\varepsilon_f + \varepsilon_F$, q, t, L)-secure MAC

Proof notation

- ε_F - max probability that NMAC will be broken, given q, t, L.
- ε_f - max probability that the compression function f will be broken, given q, t, b.
- A_N - attacker that tries to break NMAC as a MAC
- A_f - attacker that tries to break f as a MAC
- ε_N - probability that A_N succeeds
- x_i - the messages chosen by A_N in her attack

Chosen message attacks against a MAC



Proof: chosen message attack against NMAC by A_N

For $i = 1, \ldots, q$ do

$A_N \to x_i$

$A_N \leftarrow f_{k1}(\overline{F_{k2}(x_i)})$

A_N outputs (x, y)

$x \neq x_1, \ldots, x_q$

(i.e. not yet seen)

$y = NMAC_k(x)$

$k = (k_1, k_2)$

Voila! NMAC has been forged.

Proof: attack against f_{k1} as a MAC by A_f

Using A_N, we build an attacker A_f …

- Choose random k_2
- For $i = 1, …., q$ do
  - A_N -> x_i
  - A_f computes $\overline{F_{k2}(x_i)}$
  - A_f queries f_{k1} to get $f_{k1}(\overline{F_{k2}(x_i)})$
  - A_N <- $f_{k1}(F_{k2}(x_i))$
- A_N outputs (x, y)
- A_f outputs $(F_{k2}(x), y)$

Proof: attack against f_{k1} as a MAC by A_f

Choose random k_2

For $i = 1, \ldots, q$ do

Transparent to A_N

A_N thinks she's querying an oracle

$A_N \text{ -> } x_i$

A_f computes $\overline{F_{k2}(x_i)}$

A_f queries $\underline{f_{k1}}$ oracle to get $f_{k1}(\overline{F_{k2}(x_i)})$

$A_N \text{ <- } f_{k1}(F_{k2}(x_i))$

A_N outputs $\underline{(x, y)}$

A_f outputs $(\overline{F_{k2}(x), y})$

Proof: attack against f_{k1} as a MAC by A_f

Choose ra[...]

For $i = 1$ [...]

A_N [...]

A_f [...]

A_f [...]

A_N [...]

- A_N outputs (x, y)
- A_f knows x, y, and k_2
- But A_f also knows that:

MAC function *message*

$$y = \mathrm{NMAC}_k(x) = \boxed{f_{k1}}\left(\boxed{\overline{F_{k2}(x)}}\right)$$

A_N outputs (x, y)

A_f outputs $(\overline{F_{k2}(x)}, y)$

Proof: attack against f_{k1} as a MAC by A_f

Choose ra[...]

For $i = 1$[...]

A_N[...]

A_f[...]

A_f[...]

A_N[...]

- A_N outputs (x, y)
- A_f knows x, y, and k_2
- But A_f also knows that:

MAC function *message*

$$y = \mathrm{NMAC}_k(x) = \boxed{f_{k1}}\left(\boxed{\overline{F_{k2}(x)}}\right)$$

So she simply computes $\overline{F_{k2}(x)}$…

A_N outputs (x, y)

A_f outputs $(\overline{F_{k2}(x)}, y)$

Voila! f_{k1} as a MAC has been broken.

Proof: Probabilities

- A_f fails when:
  1) A_N fails (with probability ε_1)
  2) A_N succeeds, but $\overline{F_{k2}(x)} = \overline{F_{k2}(x_i)}$ (with probability ε_2)

Proof: Probabilities

- A_f fails when:

  1) A_N fails (with probability ε_1)

  2) A_N succeeds, but $\overline{F_{k2}(x)} = \overline{F_{k2}(x_i)}$ (with probability ε_2)

This means that A_f tries to use $\overline{F_{k2}(x)}$ as its forged message, but if $F_{k2}(x) = F_{k2}(x_i)$, then this message has already been seen and is not a valid forgery. Therefore, A_f fails.

Proof: Probabilities

- Case 1: A_N fails

  Then $\varepsilon_1 \leq 1 - \varepsilon_N$

- Case 2: A_N succeeds, but $\overline{F_{k2}(x)} = \overline{F_{k2}(x_i)}$

  This means $F_{k2}(x) = F_{k2}(x_i)$, which is a hash collision on F_{k2} . So ε_2 is the probability of a finding a hash collision.

  By definition, $\varepsilon_2 \leq \varepsilon_F$.

Proof: Probabilities

- Probability that A_f fails $(1 - \varepsilon_f)$ is bounded by the sum of the probabilities in the two cases above $(\varepsilon_1$ and $\varepsilon_2)$:

  $$1 - \varepsilon_f \leq \varepsilon_1 + \varepsilon_2$$

  $$1 - \varepsilon_f \leq (1 - \varepsilon_N) + \varepsilon_F$$

  $$\varepsilon_N \leq \varepsilon_f + \varepsilon_F$$

  (Recall in Theorem 4.1: "… then the NMAC function is an $(\varepsilon_f + \varepsilon_F, q, t, L)$ - secure MAC")

Proof: Probabilities

- Furthermore, since $\varepsilon_N \leq \varepsilon_f + \varepsilon_F \ldots$
  - $\varepsilon_N \leq 2\varepsilon'$, where $\varepsilon' = max(\varepsilon_f, \varepsilon_F)$
    - This says that the probability of breaking NMAC is at most 2 times the probability of breaking the underlying hash function
  - $\varepsilon' \geq 1/2\ \varepsilon_N$
    - This says that if you break NMAC, then you can break the underlying hash function with at least half that probability

Remarks

- Remark 4.2
  - The proof is *constructive*, meaning that if you can show an attacker A_N that breaks NMAC given certain resource constraints, you can also explicitly show an attacker A_f with the same resource constraints that can break the underlying hash function.
  - The proof also shows you can do the latter with at least half the probability of the former ($\varepsilon' \geq 1/2\ \varepsilon_N$).

Remarks

- Remark 4.2, *cont'd*
  - Degradation of security when going from the hash function to NMAC is minimal ($\varepsilon_N \le 2\varepsilon'$).
  - Proof considers a generic attacker, including future advances in cryptanalysis. In reality, the probabilities of success are very low.

Weaker Assumptions, Stronger Statement

- Weaker assumptions mean there are fewer conditions to meet. Therefore, a statement that is conditional upon weaker assumptions is more likely to occur, and is therefore stronger.

Remarks

- Remark 4.4
  - The actual assumptions required by the analysis above are even weaker than what was stated
  - An A_N that tries to break NMAC by attacking the compression function as a MAC is not able to choose or control $F_{k2}(x)$ because she does not know k_2 . She only sees $F_{k1}(F_{k2}(x))$.

Remarks

- Remark 4.4, *cont'd*
  - Similarly, if A_N tried to break NMAC by finding collisions in the internal function $F_{k2}(x)$, she still only sees $F_{k1}(F_{k2}(x))$. This makes it difficult for A_N to actively try and compute possible collisions on F_{k2}.
  - Applying the outer function does not hide the fact that collisions occurred in the inner function.

    If $F_{k2}(x) = F_{k2}(x_i)$

    Then $F_{k1}(F_{k2}(x)) = F_{k1}(F_{k2}(x_i))$

Remarks

- Remark 4.5
  - Because A_N can only see $F_{k1}(F_{k2}(x))$, we can modify the "(ε_F, q, t, L)-weakly collision-resistant" assumption in the theorem to the significantly weaker assumption that the inner hash function is collision resistant to adversaries that see the hash value only after it was hashed again with a different secret key.

Conclusion

- The security of NMAC/HMAC depends on that of the underlying hash. If there is an A_N that can break NMAC with a certain probability, then one can design an A_f that can break the underlying hash with at least half that probability.

Conclusion

- The security analysis assumes:
  - A generic attacker
  - An attacker that knows k_2 and can see and control $F_{k2}(x)$
  - A standalone hash function F_{k2} that is collision-resistant to a certain point
- In reality, the latter two assumptions can be replaced by weaker ones that cause the analysis to be even stronger.

Questions?