Collision Resistant Hashing: Can Composition Help?

Dan Boneh

Joint work with Xavier Boyen

Collision Resistant Hashing

- Function H : $\{0,1\}^* \rightarrow \{0,1\}^n$

  is collision resistant if "difficult" to find

  $M_0 \neq M_1$ s.t. $H(M_0) = H(M_1)$

- Used for digital signatures, e.g. certs.

- Note: not needed for HMAC
  - …. and not really needed for digital sigs.

The bad news …

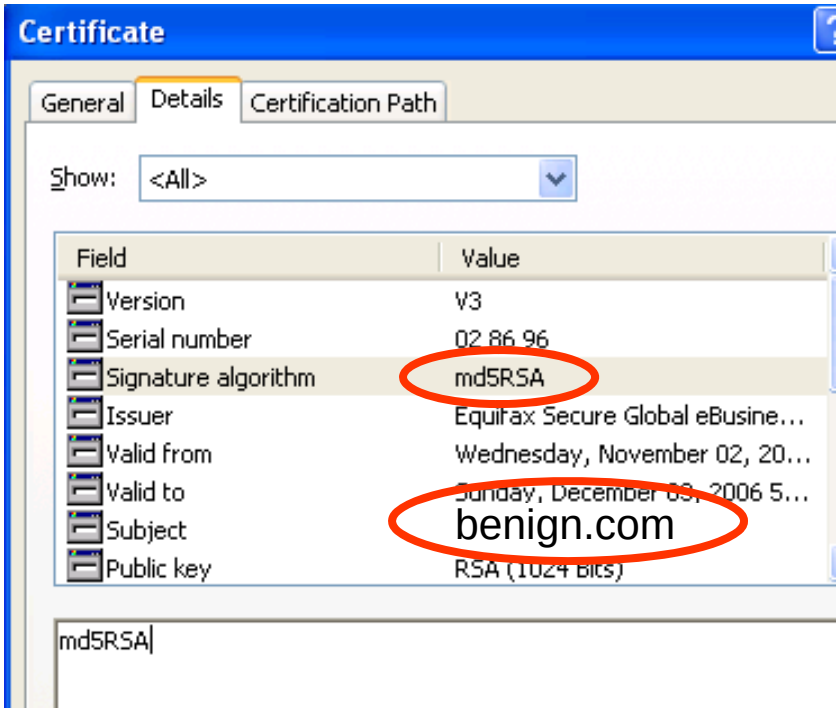
- 2005 was a tough years for CRHFs.

	Digest Length	Brute-force Attack	Better Attack	
MD4	128	2^{64}	2^{1}	[NSKO'06]
MD5	128	2^{64}	2^{30}	[WY'05,LL'05]
RIPEMD-160	160	2^{80}	2^{18}	[WLFCY'05]
SHA-1	160	2^{80}	2^{63}	[WYY'06]

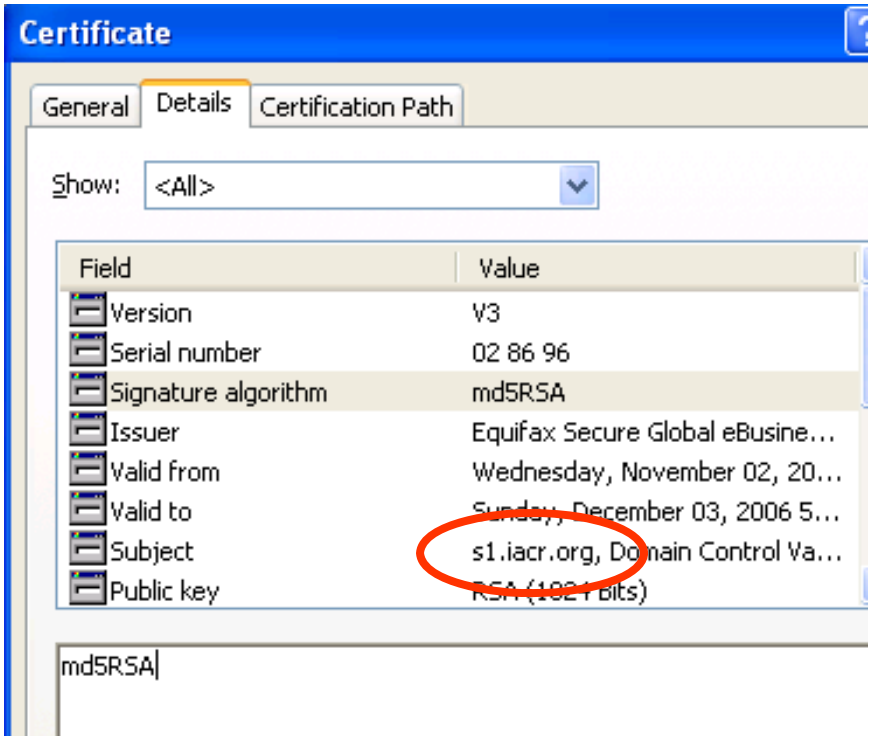
- Remaining functions (for now):
  - SHA-256, SHA-512, Whirpool
  - …. and algebraic functions.

Certificate trouble

- Lenstra, Wang, de Weger ’05:



Requested cert



Obtained cert

What to do?

- **Option 1**: Design new hash functions.
  - NIST hash function competition.
  - Hash function workshop (Aug 24-25).

- **Option 2**: Strengthen existing functions.
  - e.g. Double number of rounds of SHA-1.

- **Hedging our bets**:

  Suppose H_1 , H_2 are two CRHFs (currently).

  Goal: build a new hash H s.t.

  either H_1 , H_2 is a CRHF $\Rightarrow$ H is a CRHF.

Hedging our bets

- Simple construction: $H(M) := H_1(M) \,\|\, H_2(M)$

- Property (*):

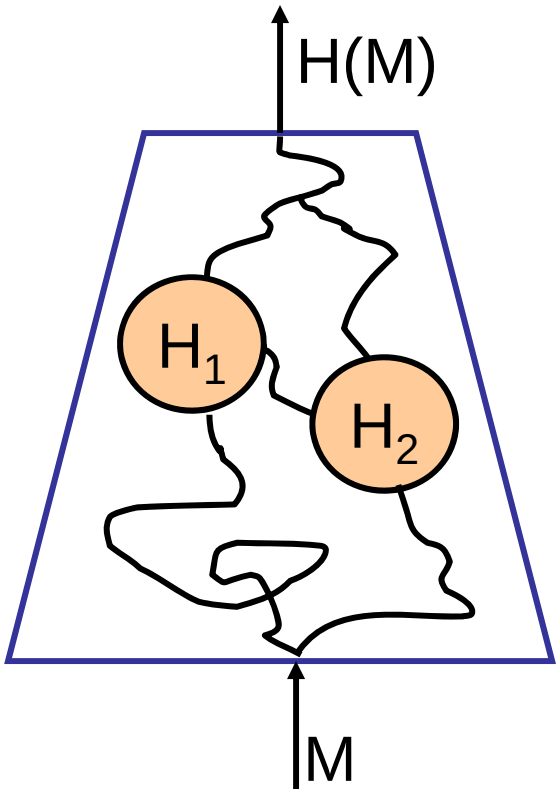
  Any collision M, M' on H $\Rightarrow$

  Collision on both H_1 and H_2

$\Rightarrow$ If either H_1 or H_2 is CRHF then H is CRHF

- … but long digests. (and twice as slow as H_1 or H_2)

Can we do better?



- Can we combine H_1 , H_2 so that:
  1. H outputs shorter digests, and
  2. Property (*) holds: collision on H gives collisions on both H_1 , H_2
- Answer: NO [BB'06]
  - Suppose H_1 , H_2 output n-bit digests.
  - H outputs fewer than 2n bits $\Rightarrow$ no proof of security.

$\Rightarrow$ Concatenation is the optimal way to hedge bets.

Composition: a few details

- A secure CRHF composition is a pair **(C, P)** where:
  - $\mathbf{C}^{H_1,H_2}(M)$ is a hash function. Uses two oracles H_1 , H_2 .
  - $\mathbf{P}^{H_1,H_2}(M,M')$ is an "efficient" algorithm such that:
    - If (M,M') are a collision for C^{H_1,H_2} then
      P outputs collisions (M_1,M_1') , (M_2,M_2') for H_1, H_2
    - P is a "proof of security" for C.

- Thm [BB'06]: If C outputs fewer than 2n bits then
  there exist H_1,H_2 and M,M' such that P fails w.h.p

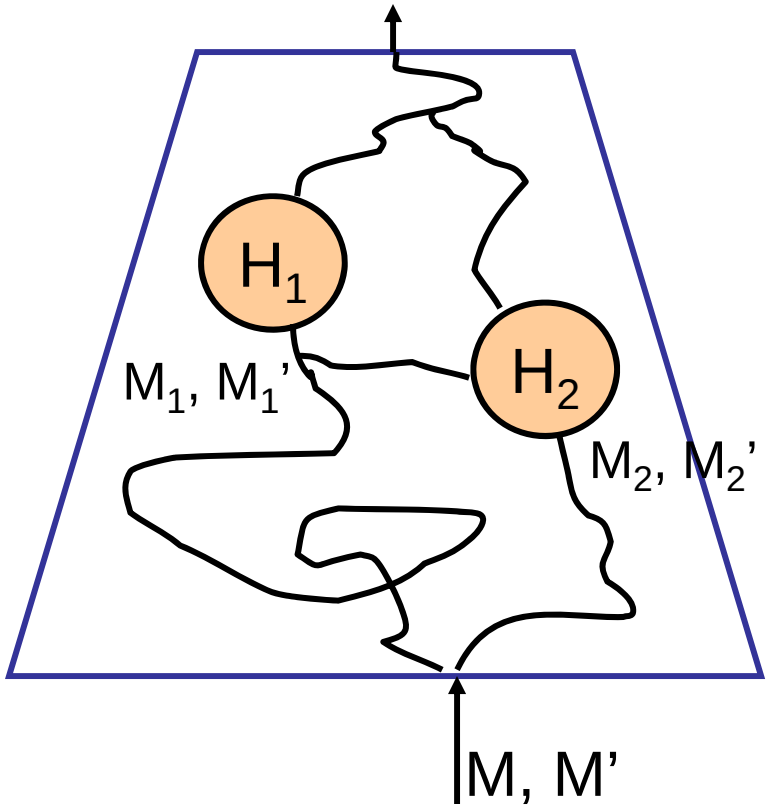
More generally …

- Suppose H_i outputs t_i bit digest, for $i=1,2,\ldots,s$
- Thm: If $C^{H_1,\ldots,H_s}(M)$ outputs fewer than $\sum t_i$ bits

  there exist $H_1,\ldots,H_s$ and M,M' such that P fails whp.
- Our example for $H_1,\ldots,H_s$ is very similar to SHA-1.

Proof Idea

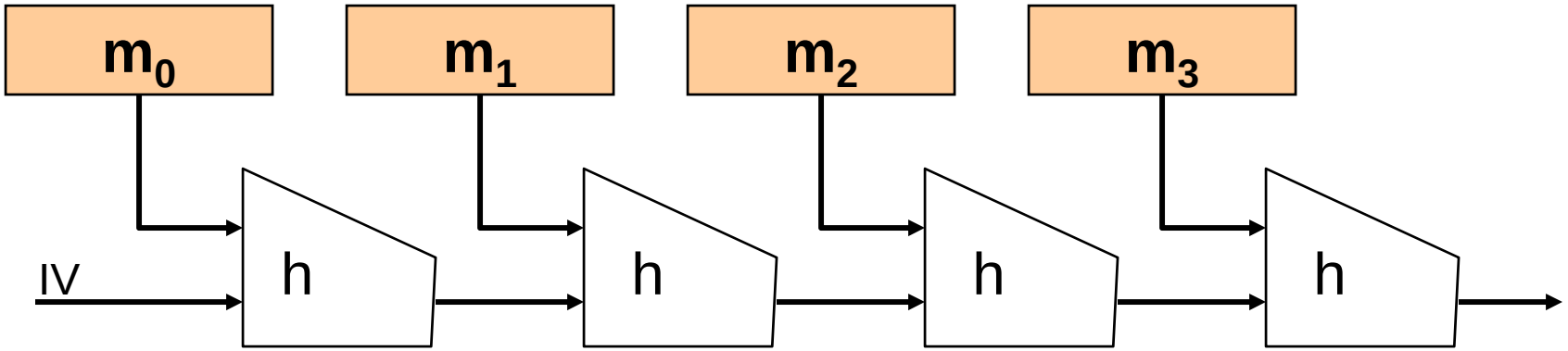
- Step 1: Prove there are H_1 , H_2 and M,M' s.t.
  1. (M,M') are a collision for C
  2. Either (M_1,M_1') or (M_2,M_2') are not a collision for H_1 or H_2



- Step 2: Use H_1,H_2 and M, M' to break P.

Joux's attack on concatenation

- Merkle-Damgard hash functions:



- H_1, H_2 : MD hash functions with n-bit digests.
  - Joux: collision for $H = H_1 \| H_2$ in time $O(n\, 2^{n/2})$

$\Rightarrow$ concat is a good hedge, but does not strengthen hash

Algebraic Compressions Functions

- Example 1: $h(m, t) := g^{m \| t} \pmod N$
  - One "multiplication" per ≈10 message bits.
  - 2048-bit digest.
- Example 2: $h(m, t) := g^m h^t \in G$
  - Two "multiplications" per ≈10 message bits.
  - 192-bit digest (using e.c.)
- Example 3: VSH: $h(m, t) := t^2 \cdot \Pi p_i^{m_i} \pmod N$
  - Contini-Lenstra-Steinfeld '06
  - One multiplication per ≈200 message bits
  - Speed: **1.1MB/sec on 1 GhZ P3**.

Summary

- Can we hedge our bets using current CRHFs?
  - Yes: concatenation.
  - … but no better method exists.
- Promising research on provable algebraic hash functions.
  - Open: can they ever compete with SHA-512 ?