Dynamic SHA2 is vulnerable to generic attacks.

According to security requirements (part 4.A iii) of the hash functions NIST expects the SHA-3 algorithm should be resistant to length-extension attacks.

Length-extension attack is not correctly understood and described in paragraph 6.1 of submitted Dynamic SHA2 documentations.

As a consequence, Dynamic SHA2 (with 256-bit and 512-bit outputs) function \( h \) is trivially vulnerable to length-extension attacks. Given \( h(m) \) and \( \text{len}(m) \) but not \( m \), the attacker easily creates \( m' \) (with correct padding) and calculates \( h(m \| m') \) simply from \( h(m) \) and \( m' \).

Moreover, in the function's design there are no precautions against other generic attacks (multi-collisions etc.).

Best regards,
Vlastimil Klima
Hi!

I write the documentation too hurriedly. I make a mistake at "Length-extension attack". If I can change it, I will change it.

Because it is hard to find the collision of Dynamic SHA2, I use no precautions against other generic attacks (multi-collisions etc.). If I know it is most important and it is not enough, I will some precautions against other generic attacks (multi-collisions etc.) such as message length.

Regards
Xu Zijie

2008/12/15, v.klima@volny.cz <v.klima@volny.cz>:

Dynamic SHA2 is vulnerable to generic attacks.

According to security requirements (part 4.A iii) of the hash functions NIST expects the SHA-3 algorithm should be resistant to length-extension attacks.

Length-extension attack is not correctly understood and described in paragraph 6.1 of submitted Dynamic SHA2 documentations.

As a consequence, Dynamic SHA2 (with 256-bit and 512-bit outputs) function (h) is trivially vulnerable to length-extension attacks. Given h(m) and len(m) but not m, the attacker easily creates m' (with correct padding) and calculates h(m || m') simply from h(m) and m'.

Moreover, in the function's design there are no precautions against other generic attacks (multi-collisions etc.).

Best regards,
Vlastimil Klima