Cryptology ePrint Archive: Report 2012/731

Time-memory Trade-offs for Near-collisions

Gaëtan Leurent

Abstract: In this work we consider generic algorithms to find near-collisions for a hash function. If we consider only hash computations, it is easy to compute a lower-bound for the complexity of near-collision algorithms, and to build a matching algorithm. However, this algorithm needs a lot of memory, and makes than 2^{n/2} memory accesses. Recently, several algorithms have been proposed without this memory requirement; they require more hash evaluations, but the attack is actually more practical. They can be divided in two main categories: some are based on truncation, and some are based on covering codes.

In this paper, we give a new insight to the generic complexity of a near-collision attack. First, we consider time-memory trade-offs for truncation-based algorithms. For a practical implementation, it seems reasonable to assume that some memory is available and we show that taking advantage of this memory can significantly reduce the complexity. Second, we show a new method combining truncation and covering codes. The new algorithm is always at least as good as the previous works, and often gives a significant improvement. We illustrate our results by giving a 10-near collision for MD5: our algorithm has a complexity of 2^45.4 using 1TB of memory while the best previous

Category / Keywords: secret-key cryptography / Hash function, near-collision, generic attack, time-memory trade-off

Date: received 31 Dec 2012

Contact author: gaetan leurent at normalesup org