Project deliverable 2

Alex Abrahamson - Mitchell Baker - Brock Ellefson - Yue Hou - Seth Severa

29 September 2017

BLAKE2 hash function

Problem description

Secure hashing of messages is a necessary part of nearly all authentication. It allows computers to determine authenticity on their own, and to ensure data is only given to whomever requests data with the proper authority. An algorithm that performs secure hashing needs to ensure that each input produces a (nearly) unique result, and that the result is computationally hard to guess.

The problem that BLAKE2 solves is the problem of being a faster, more efficient, and more secure version of MD5 and SHA-1 (both MD5 and SHA-1 are secure hashing algorithms, which are explained above).

To show that BLAKE2 is a faster algorithm, let's look at the features outlined in the proposal of BLAKE2 for SHA-3. In the document, it states that BLAKE2 performs fewer rounds (operations on a block), it will only pad the last data block if necessary, and has direct support (with no overhead) of parallelization.

To show that BLAKE2 is a more secure version of MD5/SHA-1, let's look at an example. MD5 is considered insecure for its ability in not being collision resistant. So if someone was attempting to exploit a system that used MD5, they could attempt to use MD5 to hash out millions of passwords and find the one hashed output (there are more than one) that match the hashed output produced by the real password (this is just a basic description of how MD5 is broken). In addition, SHA-1 is considered broken for basically the same reason that MD5 is – low collision-resistance. This is exactly what happened in March 2005, where Arjen Lenstra, Xiaoyun Wang, and Benne de Weger used MD5 to create two X.509 certificate hashes with different public keys (Lenstra, Wang, Weger:

Colliding X.509 Certificates).

Introspective Sort

Introspective sort (or Introsort for short) is a sorting algorithm created by Dr. David Musser in 1997. The algorithm seeks to address the primary problem with popular algorithm Quicksort. In Quicksort, for most inputs, the sorting time is faster than most other algorithms- on average $\Theta(nlogn)$. But the problem arises when we select for our pivot an element that is either greater or lesser than the majority of our remaining elements. This results in a large amount of partitions that achieve little, leading to a worst case running time of n^2. Numerous methods have been implemented to fox this, whether it takes the median of the first, last, and middle element, or randomized the pivot each time. But though the probability of choosing a bad pivot goes down in these cases, it is not eliminated.

What Introspective sort does is combine the quick average time of Quicksort with the stable worst case performance of Heapsort. When running Introsort, it functions as Quicksort until a certain number of partitions has been reached, in which case it transitions into the Heapsort algorithm. Heapsort is on average slower than Quicksort, but on the specific instances where our input array would tend Quicksort towards quadratic time, Introsort is able to detect this and transition to the algorithm with a better worse case but same average case.

In conclusion, the problem Introspective Sort solves is the balance between a fast common case algorithm, and a fast worst case algorithm. By combining the speed of Quicksort with the reliability of Heapsort, Introsort guards itself from ”median-of-three” killers but maintains a faster runtime than other $nlogn$ worst case algorithms by themselves.

Shor’s Algorithm

This is a quantum algorithm created by Peter Shor. It’s purpose is to solve integer factorization and finding discrete logarithms. So, given some integer N, this algorithm can give its prime factorization in polynomial time. This is incredibly faster than the previous most efficient algorithm, the general number sieve, which runs in sub-exponential time.

This problem is generally considered to be rather difficult on classic computers and have been used as the basis for several proposed cryptosystems. However, efficient randomized algorithms are given for these two problems on a hypothetical quantum computer. These algorithms than give the factorization of the input in polynomial time.

References

[1] Musser, David. (1997). Introspective Sorting and Selection Algorithms. Software Practice and Experience. 27. . 10.1002/(SICI)1097-024X(199708)27:8<983::AID-SPE117>3.0.CO;2-#.

[2] Shor, Peter W. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Review*, vol. 41, no. 2, 1999, pp. 303332. JSTOR, JSTOR, www.jstor.org/stable/2653075.

[3] Aumasson, J. P., Neves, S., Wilcox-OHearn, Z., & Winnerlein, C. (2013, June). BLAKE2: simpler, smaller, fast as MD5. *In International Conference on Applied Cryptography and Network Security* (pp. 119-135). Springer, Berlin, Heidelberg.