Mathematics of the SHA-1 Algorithm

Brendan Schlaman

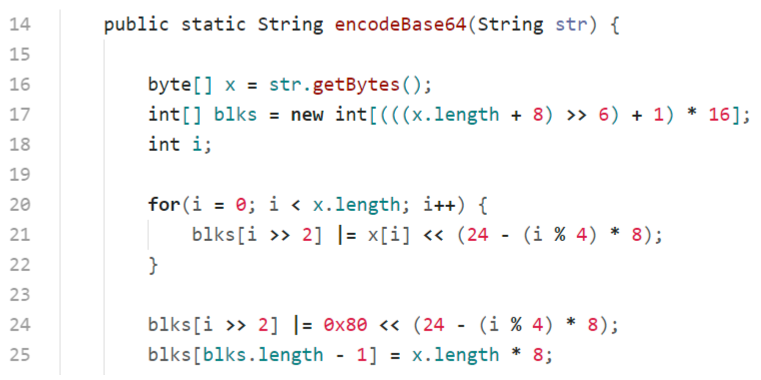
December 2018

Abstract

The purpose of this document is to explain the SHA-1 Hashing Algorithm and it's security to someone with a moderate understanding of mathematics and computer science. The code is first stepped through, and then the security of the algorithm is evaluated. The version of the source code examined here was converted to Java by Russell Beattie and Sam Ruby, and can be found at
github.com/opendatakit/javarosa/blob/master/src/org/javarosa/core/util/SHA1.java

1 SHA-1 Algorithm Stepthrough

1.1 Message is converted to bytes



```java
public static String encodeBase64(String str) {

    byte[] x = str.getBytes();
    int[] blks = new int[(((x.length + 8) >> 6) + 1) * 16];
    int i;

    for(i = 0; i < x.length; i++) {
        blks[i >> 2] |= x[i] << (24 - (i % 4) * 8);
    }

    blks[i >> 2] |= 0x80 << (24 - (i % 4) * 8);
    blks[blks.length - 1] = x.length * 8;
```

Line 61 first makes an array of bytes. This array is the length of the original string (in characters), where each character is represented by a 7 bit number, following the ASCII standard. For example,

$$"A" = 66 = 0b1000010$$

1.2 Message is padded

SHA1 uses particular padding standards in order to ensure that the final message length is a multiple of 512 bits, e.g. that $message.length = 512k, \{k \in \mathbb{N}\}$.
The following standard is used:

1. The bit 1 is added to the end of the message.
2. A 64 bit representation of the size of the message is generated.
(NOTE: the Java Program that we are following uses a 32 bit representation. This means that it can only handle messages of length at most $2^{32} - 1$ bits).
3. An amount of 0 bits are added in between the 1 and the 64 bit size representation such that the final length is a multiple of 512 bits.

The message then takes the form:

$$message \mid 1 \mid k \text{ 0 bits } (0 \leq k < 512) \mid \text{64 bit length representation}$$

Example message lengths and resultant padding:

440 bits	1	0000000	64 bit length representation
448 bits	1	0 x 511	64 bit length representation
448 bits	1	0 x 511	64 bit length representation

2 Conclusion