Gearing Up: How to Eat your Cryptocake and Still Have it

Alex Shafarenko and Bruce Christianson

Abstract

Often Alice and Bob share a fixed quantity of master key and subsequently need to agree a larger amount of session key material. At present, they are inclined to be cautious about generating too much session key material from a single master key. We argue that this caution arises from their familiarity with keys consisting of a few dozen bytes, and may be misplaced when keys consist of many billions of bytes. In particular, if the proof that the master key was securely distributed depends on a bounded-memory assumption for Moriarty, then the same assumption also imposes constraints upon the cryptanalysis which Moriarty can apply to the generated session material. Block ciphers with (effectively) Terabit blocks allow a much higher ratio of session to master key than can be countenanced with current key lengths, and we construct one such cypher.

Session key material is handed off from the key agreement protocol to application software, and so we shall assume that all session keys, once generated, are immediately shared with Moriarty. We also assume that Moriarty has sufficient computing power to break a one thousand bit key almost instantly[1], but that he cannot do this with non-negligible probability for a thousand billion bit key. Finally, we assume that Moriarty's storage is limited to a few Yottabits[2].

Suppose that we have two master keys, a red key r of the order of five hundred bits, and a longer blue key b of the order of several Terabits. We use the red key to generate a very long pseudorandom bitstring (for example, two independent AES encryptions under counter mode which are then XORed together), which we interpret as a coded one-way transformation of the blue key. The transformed blue key becomes the session key:

$$r = r_1|r_2; \quad q^{(i)} = E_{r_1}[i] \oplus E_{r_2}[i]$$

$$q = q^{(1)}|q^{(2)} \ldots |q^{(N)}; \quad s = F_q[b]$$

We ensure that the transformation F_q is *collision-full*[3] by requiring that s is

[1] Either by brute force, or by solving the equations.

[2] Actually, we can allow Moriarty more capacity, but the current estimated storage capacity of the world is considerably less than this. See Martin Hilbert et al, Science Direct, Feb 2011.

[3] Even with DES in OFB mode, the pairwise XOR of successive encryptions forms a keystream that is harder to break than triple-DES with two keys.

only half the length of b. For example, we might split $b = b_1|b_2$ and $q = q_1|q_2$ into equal halves, define

$$F_{q_i}[b_i] = b_i \cdot (2q_i + 1) \bmod 2^n$$

where $2n$ is the length of b, and set

$$F_q[b] = F_{q_1}[b_1] \oplus F_{q_2}[b_2]$$

.

This gives us less session key material than the amount of master key that we started off with. But now what is to stop us re-using the blue key with a different red key r', giving rise to a different transformation q' of b? Moriarty knows that s and s' arise from the same underlying bits, but he cannot directly attack the keyspace of b. Nor can he mount a meet-in-the-middle type attack to recover b, because he cannot distinguish the different candidate values of b except by storing them, and Moriarty's storage capacity is limited - the binary logarithm of the ratio between his available storage and the length of a single candidate for b is less than forty bits.

The fact that r and r' are relatively short is of no help to him without some knowledge of b: even if he somehow knew r and r' (in spite of the fact that they are never revealed or reused) he still cannot invert F_q and store a significant proportion of the pre-image even for a single session key. This is because, for fixed s and q, the set $F_q^{-1}[s]$ has the same entropy as s originally did: exactly half that of b. Consequently Moriarty cannot learn even a single bit of b no matter how many session keys he has.

Suppose that Alice and Bob share 64 Terabits of master key, perhaps by using a Vintage Bit protocol[4]. They divide this equally into red and blue key material. Each red key consists of two 256 bit AES keys. These red keys are used successively to transform the blue key. The 32 Tb of blue key b is always the same. This allows Alice and Bob to generate a Yottabit of shared session key, all that they will ever need.

[4]cite LNCS xxxx