Homework 1

Nikita Borisov

February 12, 2002

1 Length-doubling hashing

1.1 Collision resistance

All of the constructions are vulnerable to easy collision attacks vulnerable to easy collision attacks, with about $O(2^{32})$ time necessary. To begin, find a collision of h using the birthday paradox, using $O(2^{32})$ applications of h. Now we have that $h(a,b) = h(a',b')$. Then,

$$R(a,b,a,b) = (h(a,b), h(a,b)) = (h(a',b'), h(a',b')) = R(a',b',a',b')$$

Further,

$$R^n(a,b,a,b) = R^{n-1}(R(a,b,a,b)) = R^{n-1}(R(a',b',a',b')) = R^n(a',b',a',b')$$

Hence we were able to find a collision for any number of rounds of R.

1.2 Preimage resistance

This attack applies regardless of whether h is bijective or not. Given an image $R(w,x,y,z) = (a,b,c,d)$, we perform an exhaustive search of the h function to find a solution for $h(w,y) = (a,b)$. Similarly, we solve $h(x,z) = (c,d)$. Both of these require $O(2^{63})$ time on average, resulting in a preimage attack requiring $O(2^{64})$ time. This is much worse than what one should expect for a 128-bit random bijective function.

We can iterate this attack to break an n-round construction. E.g. to find a preimage of $R^2(w,x,y,z) = (a,b,c,d)$, we first find $h(w',y') = (a,b)$ and $h(x',z') = (c,d)$ and then $h(w,y) = (w',x')$ and $h(x,z) = (y',z')$. The complexity of an n-round attack is therefore $O(n2^{64})$. However, since an n round H function requires $2n$ applications of the h function, the n-round attack requires time equivalent to $O(2^{63})$ computations of H, regardless of n.

2 k-fold encryption

Consider the decryption function, D_k. It is natural to implement it as a look-up table; however, an alternative is to define:

$$D_k(c_0, c_1) = a_0c_0 \oplus a_1c_1 \oplus a_2$$

where a_0, a_1, a_2 are one-bit values, depending on the key. (In fact, each of the 6 possible keys maps into a distinct triple (a_0, a_1, a_2), where at least one of a_0 and a_1 is non-zero.)

What happens when we decrypt a "two-fold" encryption?

$$\begin{aligned}
& D_{k_0}(D_{k_1}(c_0, c_1), D_{k_2}(c_2, c_3)) \\
= \; & a_0 D_{k_1}(c_0, c_1) \oplus a_1 D_{k_2}(c_2, c_3) \oplus a_2 \\
= \; & a_0(a_3 c_0 \oplus a_4 c_1 \oplus a_5) \oplus a_1(a_6 c_2 \oplus a_7 c_3 \oplus a_8) \\
= \; & a_0 a_3 c_0 \oplus a_1 a_4 c_1 \oplus a_1 a_6 c_2 \oplus a_1 a_7 c_3 \oplus a_0 a_5 \oplus a_1 a_8
\end{aligned}$$

We can simplify this equation by renaming coefficients:

$$b_0 c_0 \oplus b_1 c_1 \oplus b_2 c_2 \oplus b_3 c_3 \oplus b_4$$

It is easy to see that for "five-fold" decryption, we will have

$$D^5(c_0, \ldots, c_{31}) = a_0 c_0 \oplus \cdots \oplus a_{31} c_{31} \oplus a_{32}$$

Each bit of known plaintext gives us a linear equation in the unknown a_i's. Therefore, given about 33 known plaintexts, we should be able to recover the full matrix by Gaussian elimination, using about 33^2 work. (So in general, an m-fold cipher can be broken with $O(2^{2m})$ work.)

If we do not have any known plaintexts, but we do know the expected distribution of them, we can still break the cipher, but with a bit more work. This attack relies on the observation that if we change the decryption function by changing a_j from 0 to 1 (for $j < 32$), we will expect to obtain a random distribution when we decrypt. Therefore, given enough ciphertext that we would be able to statistically distinguish the distribution bias of the corresponding plaintext from random, we can compute the distribution of $D_k^5(c)$, where $k = (a_0, \ldots, a_3 1, 0)$ are the coefficients of the linear equation. Any incorrect choice will produce a random distribution, while the correct one will result in a bias. If this bias matches the expected one, then we know the correct coefficients; otherwise, the coefficients are $(a_0, \ldots, a_3 1, 1)$.