

Tutorial Questions

Cryptography and Network Security

Date: 9/11/23

Q1. It is possible to use a hash function to construct a block cipher with a structure similar to DES. Because a hash function is one way and a block cipher must be reversible (to decrypt), how is it possible?

Q2. Suppose $H(m)$ is a collision-resistant hash function that maps a message of arbitrary bit length into an n -bit hash value. Is it true that, for all messages x, x' with $x \neq x'$, we have $H(x) \neq H(x')$ Explain your answer.

Q3. Consider the SHA-3 option with a block size of 1024 bits and assume that each of the lanes in the first message block (P_0) has at least one nonzero bit. To start, all of the lanes in the internal state matrix that correspond to the capacity portion of the initial state are all zeros. Show how long it will take before all of these lanes have at least one nonzero bit. Note: Ignore the permutation. That is, keep track of the original zero lanes even after they have changed position in the matrix.

Q4. Suppose $a_1 a_2 a_3 a_4$ are the 4 bytes in a 32-bit word. Each a_i can be viewed as an integer in the range 0 to 255, represented in binary. In a big-endian architecture, this word represents the integer

$$a_1 2^{24} + a_2 2^{16} + a_3 2^8 + a_4$$

In a little-endian architecture, this word represents the integer

$$a_4 2^{24} + a_3 2^{16} + a_2 2^8 + a_1$$

a. Some hash functions, such as MD5, assume a little-endian architecture. It is important that the message digest be independent of the underlying architecture. Therefore, to perform the modulo 2 addition operation of MD5 or RIPEMD-160 on a big-endian architecture, an adjustment must be made. Suppose $X = x_1 x_2 x_3 x_4$ and $Y = y_1 y_2 y_3 y_4$. Show how the MD5 addition operation $(X + Y)$ would be carried out on a big-endian machine.

b. SHA assumes a big-endian architecture. Show how the operation $(X + Y)$ for SHA would be carried out on a little-endian machine.

