0001	1000
0010	0100
0011	0000
0100	0101
0101	1101
0110	1001
0111	0001
1000	0011
1001	1110
1010	0110
1011	1010
1100	0111
1101	1011
1110	1110
1111	0011

Suppose that initially, blocks A and B are cached, but C and D are not. Blocks A and B are dirty and recently written with values of 1010 and 1111, respectively.

a) Assume a direct encryption scheme (i.e. electronic codebook mode). If block A is written back to memory, the cipher text of A that is written to memory is 0110 . If block B is then written back to memory, the cipher text of B that is written to memory is 0011

b) Assume a counter-mode encryption using a 4-bit global counter. The counter value is currently 0010. If block A is written back to memory, the encryption seed that is used is 0011 , its pad is 0000 , and the cipher text of A that is written to memory is 1010 . If block B is then written back to memory, the encryption seed that is used is 0100 , its pad is 0101 , and the cipher text of A that is written to memory is 1010 . The global counter value after this is [0100]. Assume that a counter is incremented before used for encryption (and not the other way around).

c) Assume that counter mode encryption with per-block (or local) counter is used. Each per-block counter is 2-bit long. Initially, the counters for A and B are both 00. If block A is written back to memory, the encryption seed that is used is [0001], its pad is [1000], and the cipher text of A that is written to memory is [0010]. If block B is then written back to memory, the encryption seed that is used is [0101], its pad is [1101], and the cipher text of A that is written to memory is [0010]. The local counter values after this are [01] for block A and [01] for block B. Assume that a counter is incremented before used for encryption (and not the other way around).

Answer 1:
0110
Answer 2:
0011
Answer 3:
0011
Answer 4:
0000
Answer 5:
1010
Answer 6:
0100
Answer 7:
0101
Answer 8:
1010
Answer 9:
0100
Answer 10:
0001
Answer 11:
1000
Answer 12:

0010

Answer 13:

0101

Answer 14:

1101

Answer 15:

0010

Answer 16:

01

Answer 17:

01

Question 2

10 / 10 pts

a) Suppose that the cache block size is 128 bytes, and we encrypt the memory using global counter with 32-bit counter size. If the memory is 8GB, how many additional GBs are needed to store counters in memory? 0.25 (To simplify, pretend that 8GB = 8,000,000,000 bytes. Provide your answer to two digits decimal. For example, 11.00, 7.22, etc.)

b) Suppose that the cache block size is 32 bytes, and we encrypt the memory using global counter with 64-bit counter size. If the memory is 8GB, how many additional GBs are needed to store counters in memory? 2.00 (To simplify, pretend that 8GB = 8,000,000,000 bytes. Provide your answer to two digits decimal. For example, 11.00, 7.22, etc.)

Answer 1:

0.25

Answer 2:

2.00

Question 3

20 / 20 pts

Suppose that we have a hypothetical machine with the following characteristics

- A cache block is 4-bit in size
- The memory can hold four data blocks: A, B, C, and D, laid out contiguously in that order in memory.
- We employ regular (non-Bonsai) Merkle Tree that covers data to protect its integrity. For the Merkle Tree, we employ a keyed hash (MAC) function that produces the following 2-bit output given some 4-

bit input (shown in the table below). Assume that Merkle Tree root is always on chip, while all other Merkle Tree nodes are cacheable.

Input, i.e. x	MAC Output, i.e. H_K(x)
0000	00
0001	01
0010	10
0011	11
0100	00
0101	01
0110	10
0111	11
1000	00
1001	01
1010	10
1011	11
1100	00
1101	01
1110	10
1111	11

Suppose that initially, none of blocks A-D are cached. In memory, blocks A, B, C, and D, have the following values: 0001, 0101, 1011, and 0010.

a) What is value of the root of the tree? 0101

b) Suppose that block A is brought on chip and a new value of 1010 is written to it. Then, it is evicted from the cache and written back to memory. Assuming that all Merkle Tree nodes are cached on chip, what is the value of Merkle Tree node that is the parent to A? 1001

c) Then, block D is brought on chip and a new value of 1111 is written to it. Then, it is evicted from the cache and written back to memory. Assuming that all Merkle Tree nodes are cached on chip, what is the value of Merkle Tree node that is the parent to D? 1111

d) Continue from part (c). Now the Merkle Tree node that holds the parent of D is evicted. What would be the value of Merkle Tree root node? 0111

Answer 1:
0101
Answer 2:
1001
Answer 3:
1111
Answer 4:
0111

Question 4
10 / 10 pts

a) Suppose that we use Merkle Tree that uses 64-bit hash with 64-byte block size. The arity of the Merkle Tree is 8 and the maximum ratio of Merkle Tree size to data is 0.143 (round to nearest 3 digit decimal).

b) Suppose that we use Merkle Tree that uses 128-bit hash with 32-byte block size. The arity of the Merkle Tree is 2 and the maximum ratio of Merkle Tree size to data is 1.00 (round to nearest 3 digit decimal).

Answer 1:
8
Answer 2:
0.143
Answer 3:
2
Answer 4:
1.00

Quiz score: 70 out of 70