Message Authentication Codes (MAC)

To prove the security of this MAC, we will use the standard technique of reduction to a game between an adversary and a challenger.

Game 0: The challenger chooses a random key k and provides it to the adversary. The adversary can then submit any message m of length at most 2^(n/4)-1. The challenger computes the tag t for the message m using the Mac algorithm and sends it to the adversary. The adversary can then make any number of queries to the tag verification algorithm Verify.

Game 1: This game is identical to Game 0, except that the challenger does not choose a random key. Instead, the challenger chooses a random function f: {0,1} ^n x {0,1} ^n/4 -> {0,1} ^n and provides it to the adversary. The rest of the game proceeds as in Game 0.

We will show that if there exists an adversary A that can win Game 0 with probability greater than 1/2 + negl(n), then there exists a distinguisher D that can distinguish the random function f from a truly random function with probability greater than negl(n).

Distinguisher D:

1. The challenger chooses a random key k and two distinct messages m0 and m1 of length at most 2^(n/4)-1.
2. The challenger computes the tags t0 and t1 for messages m0 and m1 respectively, using the Mac algorithm with key k.
3. The challenger flips a coin b and sends to the adversary.
4. If b=0, the challenger replaces t0 with a random tag chosen uniformly at random from the set of all possible tags of length n/4*(d+1) and sends the modified tag to the adversary. If b=1, the challenger sends t1 to the adversary.

5. The adversary can then make any number of queries to the tag verification algorithm Verify, with the restriction that the adversary cannot make a query using a tag that was already provided by the challenger.
6. After the adversary has made its queries, it outputs a bit b'.
7. The output of D is 1 if b'=b, and 0 otherwise.

We claim that Pr[D(f) = 1] - Pr[D(r) = 1] > negl(n), where f is a truly random function and r is a random function sampled by the challenger in Game 1.

To see why this is true, consider the following cases:

- If b=0, then t0 is a random tag, and the probability that the adversary can guess the correct value of b with high probability is negl(n).
- If b=1, then t1 is a valid tag computed by the Mac algorithm using the random key k. Therefore, the probability that the adversary can guess the correct value of b with high probability is 1/2 + negl(n), because this is the probability that A wins Game 0.

Therefore, we have shown that if there exists an adversary A that can win Game 0 with probability greater than 1/2 + negl(n), then there exists a distinguisher D that can distinguish the random function f from a truly random function with probability greater than negl(n). This contradicts the assumption that f is a truly random function, and therefore **we conclude that the MAC is secure.**