1 CBCMAC and Variants

1. Given some (known or chosen) sample pairs message-code (m_i, c_i), the goal of a MAC forgery attack is to output a valid pair message-code (m, c).

2. It is simply $\mathcal{O}(2^n)$.

3. Since there is a xor between one message block let x_i and the result of $\mathsf{CBCMAC}(K, x_1, \ldots, x_{i-1})$ they should have the same bit length:

$$n = b\ .$$

4. As seen in the course:

   - choose m_1 and obtain $c_1 \leftarrow \mathsf{CBCMAC}(K, m_1)$
   - choose m_2 and obtain $c_2 \leftarrow \mathsf{CBCMAC}(K, m_2)$
   - choose B_1, let $m_1' = m_1 \| B_1$ and obtain $c_1' \leftarrow \mathsf{CBCMAC}(K, m_1')$
     Note that $c_1' = \mathsf{CBCMAC}(K, B_1 \oplus \mathsf{CBCMAC}(K, m_1)) = \mathsf{CBCMAC}(K, B_1 \oplus c_1)$
   - let $m_2' = m_2 \| B_2$ for some B_2
     Note that c_2' should be $\mathsf{CBCMAC}(K, B_2 \oplus \mathsf{CBCMAC}(K, m_2)) = \mathsf{CBCMAC}(K, B_2 \oplus c_2)$
     So, if $B_2 \oplus c_2 = B_1 \oplus c_1$ then $c_2' = c_1'$
     Fix $B_2 = B_1 \oplus c_1 \oplus c_2$
   - output $(m2 \| B_2, c_1')$

2 Modulo 33 Calculus

1. Note that we can write

$$N = d_{n-1} \cdot 10^{n-1} + \ldots + d_2 \cdot 10^2 + d_1 \cdot 10 + d_0$$

   which can be written as

$$N = \sum_{i=0}^{n-1} d_i \cdot 10^i$$

   So computing modulo 3 we find

$$N \equiv \sum_{i=0}^{n-1} d_i \cdot 10^i \stackrel{10 \equiv 1}{\equiv} \sum_{i=0}^{n-1} d_i \pmod 3$$

2. To compute N mod 3:

   $n = 0$
   for $i = 0$ to $n - 1$
   $\quad n = n + d_i \bmod 3,$
   output n

3. Computing modulo 11we find

$$N \equiv \sum_{i=0}^{n-1} d_i \cdot 10^i \overset{\substack{11^{i_{\text{even}}} \equiv 1,\\ 11^{i_{\text{odd}}} \equiv -1}}{\equiv} \sum_{i=0, i \text{ even}}^{n-1} d_i - \sum_{i=0, i \text{ odd}}^{n-1} d_i \pmod 3$$

4. To compute N mod 11:

$$\begin{aligned}&n = 0\\&\textbf{for } i = 0 \text{ to } n-1\\&\qquad n = n + (-1)^i \cdot d_i \bmod 11,\\&\textbf{output } n\end{aligned}$$

5. $N = 22a + 12b = 3 \cdot (7a + 4b) + a \equiv a \pmod 3$
   $N = 22a + 12b = 11 \cdot (2a + b) + b \equiv b \pmod{11}$

6. By using the CRT we know $\mathbb{Z}_{33}$ is isomorph to $\mathbb{Z}_3 \times \mathbb{Z}_{11}$. So, any $(a, b) \in \mathbb{Z}_3 \times \mathbb{Z}_{11}$ has a unique representation in $\mathbb{Z}_{33}$.

7. First compute 12341234 mod 3: $12341234 \equiv 1+2+3+4+1+2+3+4 \equiv 2 \pmod 3$.
   The order of $\mathbb{Z}_3^*$ is 2. So, compute 56789 mod 2 = 1.
   So,
$$a = 12341234^{56789} \equiv 2^1 \equiv 2 \pmod 3$$
   Then, compute 12341234 mod 11: $12341234 \equiv 4-3+2-1+4-3+2-1 \equiv 4 \pmod{11}$.
   The order of $\mathbb{Z}_{11}^*$ is 10. So, compute 56789 mod 10 = 9. So,
$$b = 12341234^{56789} \equiv 4^9 \equiv 4^{-1} \equiv 3 \pmod{11}$$
   Finally compute
$$N = 22a + 12b \bmod 33 = 14$$

3 RSA with Faulty Multiplier

1. Write
$$\sum_{i,j} y_i \cdot y_j^* \cdot 2^{32(i+j)}$$

2. At least once there will be the multiplication α times β. So, there will be an incorrect value and the square y^2 will be incorrect.

3.
$$x > 0 \;\Rightarrow\; y > 2^{\ell-1} + 2^{\ell-3} \;\Rightarrow\; y > p$$
   and
$$x < 2^{\ell-3} \;\Rightarrow\; y < 2^{\ell-1} + 2^{\ell-3} + 2^{\ell-3} \;\Rightarrow\; y < 2^{\ell-1} + 2^{\ell-2} \;\Rightarrow\; y < q$$

4. Note that y contains at least one 32-bit word equal to α and another equal to β. Since $y < q$ we will have $y_q = q$. So α and β will be used in a square which lead us to incorrect decryption of y_q. So, $y'_q = y' \bmod q \neq y_q$

5. Note that y contains at least one 32-bit word equal to α and another equal to β. Since $y > p$ we will have $y_p \neq p$ and with high probability α and β will disappear and there will be no computation error. So, $y'_p = y' \bmod p = y_p$.

6. If an error occured, we have two different values y and y'. Note that $y_p = y \bmod p$ is equal to $y'_p = y' \bmod p$. So, $y - y_p$ is a multiple of p as well as $y' - y_p$.
   Computing $\mathsf{gcd}(y - y_p, y' - y_p)$, we will obtain p.
   Then obtain q by computing N/p.