CRC example

Question: You receive the packet 1010110, which includes the 3-bit CRC 110 produced by the generator 1001. Are any errors detected?

Answer: Received payload was 1010 with CRC of 110. To compute CRC, first pad the payload with same number of zeros as are in the CRC, then do long division by the generator (using bitwise XORs for subtractions):

```
             1011
      ----------
1001 |  1010000
        1001
        ----
          110
          000
          ---
          1100
          1001
          ----
           1010
           1001
           ----
            011
```

Remainder is 011, i.e. computed CRC is 011. But received CRC was 110, so computed CRC does not equal received CRC, therefore errors have been detected.

Cryptography examples

All the examples use the following keys, and assume that RSA public key cryptography is used throughout:

Public key for Alice: modulus 22, exponent 3
Private key for Alice: modulus 22, exponent 7
Public key for Bob: modulus 34, exponent 5
Private key for Bob: modulus 34, exponent 13

The following table can be used for numerical calculations:

m	m^3 mod 22	m^7 mod 22	m^5 mod 34	m^{13} mod 34
1	1	1	1	1
2	8	18	32	32
3	5	9	5	29
4	20	16	4	4

5	15	3	31	3
6	18	8	24	10
7	13	17	11	23
8	6	2	26	26
9	3	15	25	25
10	10	10	6	28
11	11	11	27	7
12	12	12	20	14
13	19	7	13	13
14	16	20	12	22
15	9	5	19	19
16	4	14	16	16
17	7	19	17	17
18	2	6	18	18
19	17	13	15	15
20	14	4	22	12

Question: A would like to encrypt and send the number 6 to B. What is the encrypted number sent?

Answer: A encrypts using B's public key. Formula is $c=m^e \bmod n$. Here, m=6, e=5, n=34. So $c = 6^5 \bmod 34 = 24$.

Question: A has received the encrypted number 9 from B. What was the plaintext number sent by B?

Answer: B' s message to A is encrypted using A's public key. A should decrypt using A's private key. Formula is $m = c^d \bmod n$. Here, c=9, d=7, n=22. So $m = 9^7 \bmod 22 = 15$.

For the next questions, use the following hash function h, and assume it is a cryptographic hash function (it isn't, but don't worry about that):

h(m) = sum of decimal digits of m, mod 10

Question: B would like to send the message 39 to A, without encryption but with a digital RSA signature. What is actually sent?

Answer: B computes h(m) = h(39) = 12 mod 10 = 2. To sign, B encrypts the hash with B's private key, so signature s is $s=2^{13} \bmod 34 = 32$. Final data sent is the message (39) followed by the signature (32), or 3932.

Question: B receives the message m=13 followed by the signature s=16. A claims to have sent the message and signed it using RSA. (a) How can B verify that A sent the message? (b) How can B verify that the message received has not been tampered with?

Answer: Compute h(m) = h(13)=4. Decrypt 16 using A's public key, obtaining $h' = 16^3 \bmod 22 = 4$. If h'=h(m), then message is from A and has not been altered, otherwise we have no information about the

authenticity or integrity of the message. In this particular case, h'=h(m)=4, so the message is from A and has not been tampered with.

The remaining questions assume that A and B have established a shared, secret session key s=12 known to no one else. The & sign represents concatenation of decimal numbers, e.g. "523 & 41" means "52341", and "523 & (7mod2)" means "5235". Alice and Bob agree to validate the integrity of their messages by concatenating each message with the shared secret, computing a hash of the result using the above hash function, and appending this value to the original message.

Question: A would like to send the message 18 to B, guaranteeing authenticity and integrity (but not confidentiality), and without the expense of using public key cryptography. What should A send to B?

Answer: Compute MAC = h(18 & s) = h(1812) = 2. Send 18 & MAC = 18 & 2 = 182.

Question: A receives from B the data 4556672, in which the last digit is a MAC based on their shared secret. Has the message being tampered with?

Answer: MAC = last digit = 2. Message m = 455667. Compute h(m & s) = h(455667 & 12) = h(45566712) = 6. Because the computed MAC, 6, does not equal the received MAC, 2, the message appears to have been forged.