1 Warming up ...

1.1 Warm up 1

Consider the following five events:

a. Correctly guessing a random 128-bit AES key on the first try.

b. Winning a lottery with 1 million contestants.

c. Winning a lottery with 1 million contestants 5 times in a row.

d. Winning a lottery with 1 million contestants 6 times in a row.

e. Winning a lottery with 1 million contestants 7 times in a row.

Sort them by likelihood.

1.2 Warm up 2

Suppose that using commodity hardware it is possible to build a computer for about 200 dollars that can brute force about 1 billion AES keys per second. Suppose an organization wants to run an exhaustive search for a single 128-bit AES key and was willing to spend 4 trillion dollars to buy these machines (this is more than the annual US federal budget). How long would it take the organization to brute force this single 128-bit AES key with these machines? Ignore additional costs such as power and maintenance.

2 (A?)symmetric Encryption

a. Are DES/AES symmetric or asymmetric encryption scheme? Why?

b. What is a block cipher? What is a stream cipher?

c. What is a Feistel structure?

d. Is a 1-round feistel network indistinguishable from a truly random permutation? Explain your answer.
e. Is a 2-round feistel network indistinguishable from a truly random permutation? Explain your answer.

f. Explain ECB, CBC, CFB, OFB and CTR modes of encryption for symmetric ciphers. Are these stream or block modes? What are their pros/cons (risk of loop, parallelizability of decryption/encryption)?

g. Let $m$ be a message consisting of $L$ AES blocks (say $L=100$). Alice encrypts $m$ using CBC mode and transmits the resulting ciphertext to Bob. Due to a network error, ciphertext block number $L/2$ is corrupted during transmission. All other ciphertext blocks are transmitted and received correctly. Once Bob decrypts the received ciphertext, how many plaintext blocks will be corrupted?

h. Same question as g but with CTR mode of encryption.

3 RSA textbook

a. Explain the RSA encryption scheme. What is the private key? What is the public key?

b. Given the public key, can I recover the private key? How?

c. Given the private key, can I recover the public key? How?

d. What is indistinguishability in the presence of an eavesdropper?

e. What is CPA security?

f. What is CCA security?

g. Is RSA textbook encryption CPA secure? CCA secure?

4 MAC

a. What is a MAC. What property (security) shall it provide?

b. Consider the CBC-MAC in the lecture note p. 44. Assume Malice has 2 pairs of $(m_1, t_1)$ and $(m_2, t_2)$. Show how Malice can construct a new pair $(m_3, t_3)$ ($m_3$ does not necessarily has the same length as $m_1$ and/or $m_2$). Why does this make it an insecure MAC? What principle does it violate?

c. You want to create your own MAC and you decide to use RSA. Given $e$ the public exponent, $d$ the private one and $N$ the modulo, you generate the tag $T$ for any message $M! = 0$ as $T \leftarrow M^d mod N$, and verify the MAC as follow:

$$Verif(T) = \begin{cases} 0 & T^e \neq M \\ 1 & T^e = M \end{cases}$$

As an attacker, you have access to $e$ and $N$, but not $d$! Is your MAC secure? Why?
5 PRF

Let $R := \{0, 1\}^4$ and consider the following PRF $F : R^5 \times R \to R$ defined as follows:

$$F(k, x) := t = k[0]$$

```plaintext
for i=0 to 3 do
    if (x[i]==1) t = t ^ k[i+1]  //comment: ^ is xor
output t
```

That is, the key is $k = (k[0], k[1], k[2], k[3], k[4])$ in $R^5$ and the function at 0101 is defined as $F(k, 0101) = k[0] \oplus k[2] \oplus k[4]$.

For a random key $k$ unknown to you, you learn that $F(k, 0110) = 0011$ and $F(k, 0101) = 1010$ and $F(k, 1110) = 0110$. What is the value of $F(k, 1101)$? Note that since you are able to predict the function at a new point, this PRF is insecure.

6 WEP 802.11

While TLS is a well-regarded, if overly complex, protocol, the Wired Equivalent Protocol for wireless traffic encryption has earned the nickname “how not to design a crypto protocol.” It is regretfully implemented in millions of wireless routers despite catastrophic security flaws.

The basic setup involves a wireless client (such as a laptop) sharing a 40-bit key with the access point. When a connection is made, encryption is done using RC4-64, with the 64-bit key consisting of the 40-bit shared key and a 24-bit IV which is incremented for each packet. Within the packets, a CRC-32 checksum is appended prior to encryption. The value sent for a packet is:

```
IV|E_K[IV{M|CRC_M}]
```

a. Why is a new IV used with each packet? In practice, is 24 bits enough?

b. Comment on the choice of 40-bit keys. Why might this have been chosen? Also, explain why allowing users to choose their own 40-bit hex value is dangerous.

c. Why is the CRC included? Is this appropriate? Explain specifically what attacks are possible due to the CRC, and what an appropriate fix would be.

d. The use of many closely related keys in RC4 is insecure, there are complicated but devastating algebraic attacks on RC4 given reasonable amounts of known plaintext encrypted under related keys. Explain where known plaintext is likely to come from in the WEP case.

e. At the end of the day, how critical is WEP security? Explain what attacks are possible and not possible on a laptop browsing the web through an insecure WEP channel.
7 Protocol Composition

Suppose you’re designing a data transmission protocol over a faulty, insecure, and expensive channel, such as IP. You want to apply the following transforms to all data packets:

- Encryption of the data using AES-CBC to provide confidentiality and integrity
- Compression of the data using gzip to increase bandwidth efficiency
- Calculation of a BCH error-correcting code to ensure reliability of transmission

The operations Encrypt(), Compress(), and Error-Correct() can be applied to outgoing data packets in any of 3! = 6 orders. Is there a proper ordering for the three? Which of these orders will cause problems? Describe specifically if confidentiality, efficiency, and reliability may be broken with improper ordering.

8 Protocols (2012 Paper 7 Question 12)

The RSA cryptosystem can be tuned to make the workload asymmetric: with \( d = 3 \), encryption (cubing modulo \( n \)).

The following public-key protocol uses the above property to allow two principals \( A \) and \( B \) to establish a common secret key \( N_b \) (invented by \( B \)) without incurring high computational load, thanks to the help of a server \( S \) who computes all the cubic roots in the protocol. Attackers are assumed to be able to overhear, but not alter, the messages between \( A, B \) and \( S \).

\[
\begin{align*}
A \rightarrow S &: B, N_a^3 \mod n \\
S \rightarrow B &: A \\
B \rightarrow S &: A, N_b^3 \mod n \\
S \rightarrow A &: B, N_a \text{ xor } N_b
\end{align*}
\]

a. What is the purpose of \( N_a \)?

b. Describe in detail a protocol attack that will allow two colluding attackers \( C \) and \( D \) to recover \( N_b \). Assume \( S \) is stateless.

c. Stop the attack you described in b by making \( S \) stateful.

d. Describe in detail a more sophisticated attack whereby the colluding attackers will recover \( N_b \) even if \( S \) adopts the precaution you described in c.

e. Fix the attack described in d.