Department of Computer Science
University of California, Berkeley
CS 294-90, Cryptanalysis
John Manferdelli

Final Exam

Date distributed: May 2, 2013
Date due: May 8, 2013, 6PM PDT
Send completed exam to: JohnManferdelli@hotmail.com

Rules of the Road: Before you read this exam, you may prepare a two-sided (one sheet) set of notes (the "Cheat Sheet"). Once you begin the exam, you may only consult the Cheat Sheet. During the exam, you may use a manual or computer based -calculator. You may not consult others during the exam but you can send me email if you have any questions. You may also use "Excel" or similar tools, using functions you define and simple packaged functions like "mod" and "power," but not complex functions like "GCD." Please comply with UC's policies on academic conduct. You do not have to prove standard results you use but please state any assumptions or theorems you rely on as clearly as you can.

Complete all problems

1. On a machine where you plan to generate symmetric keys for, say, AES, you have access to a single entropy source E. E has the following statistical properties: You can draw four bits of entropy, a_1,a_2,a_3,a_4, from E per call. The process generating these bits is not "memory-less" and has the following distribution: $Pr(a_1=0)= Pr(a_1=1)= 1/2$. For $i>1$, $Pr(a_i=0|a_{i-1}=0)= 2/3$, $Pr(a_i=1|a_{i-1}=0)= 1/3$, $Pr(a_i=1|a_{i-1}=1)= 3/4$ and $Pr(a_i=0|a_{i-1}=1)=1/4$. You can call access E many times and each time you receive a four-bit response with the foregoing distribution but there is a time delay between calls and you don't want to "waste" entropy.
    a. What is the entropy (or information rate) of E? Remember if E consisted of independent, identically distributed bits (which is doesn't) each with equal probability for drawing 0 or 1, the information rate would be 4 bits.
    b. You wish to use E efficiently to generate keys. Key bit should be independent, identically distributed bits with maximum information rate (i.e.-for 128 bit key, you want 128 bits, **k**, with $H(\mathbf{k})=128$). Design a mechanism to obtain **k** from E. You may assume a cryptographic hash, like SHA-1 is a "perfect" mixer that loses no entropy.

c. What is the most likely and least likely sequences, a_1,a_2,a_3,a_4, that E would generate?
d. Your ambitious, but lazy, companion decides to use concatenated outputs from E for key generation. If E had been perfect, AES-128, exhaustive search (given a corresponding plain and cipher text) would find the key used in 2^{127} trials on average. How would you calculate, how many trials, on average, would a key search from this flawed distribution take? You may not be able to find a closed form expression but you should be able to write a general expression and examine some smaller cases.

2. What is a cryptographic hash? What three properties should a cryptographic hash have? Let's explore one of these.

   Suppose we have n objects (n is large) and we select r of them (with replacement). Let's calculate the probability that we will have r *distinct* (i.e.- non-colliding) objects when we're done: There are n^r ways to pick the r objects. There n ways to pick the first object (without duplication), (n-1) ways to pick the second and so on, the rth object can be selected (n-(r-1)) ways so the probability we will have *no* duplication is: (n/n) ((n-1)/n)...((+1-r)/n). We write this as

$$\Pr^{\text{no-collision}}(n,r)= \prod_{i=1}^{r-1}(1-\frac{i}{n}) \approx 1-\sum_{i=1}^{r-1} i \approx (1-\frac{r \bullet r}{2n})$$

   Now $\lim_{n\to\infty}\left(1-\frac{x}{n}\right)^n = e^x$, so $\Pr^{\text{no-collision}}(n,r)= \left(\left(1-\frac{r^2}{2n}\right)^n\right)^{1/n} \approx e^{-\frac{r^2}{2n}}$

   A. Set $r=\alpha\sqrt{n}$. For what α, is $\Pr^{\text{no-collision}}(n, \alpha\sqrt{n})$= 1/2, 3/4, 99/100? These represent a 50%, 25% and 1% probability of collision
   B. Suppose we use two Merkle-Damgard hashes of output size 256 bits ~~(say, SHA-256 and SHA-3 with 256 bit output)~~ in which any collision could be used to produce other collisions by appending the same bit strings to the original (non-identical) colliding strings. ~~Assuming (which is the best case) each has a 50% chance of collision, what is the probability that *both* have a collision? For what α, is there a 50% probability that both have a collision? What is r in this case? Finally, what r is there a 50% chance that a (good) hash with 512 bits of output has a collision?~~ How much time (in the size of the output hash) does it take to find a string that produces the same hash for both hash algorithms with 25% probability? 50% probability? What do you conclude about building cryptographic hashes out of independent hash functions is?
3. The k-linear feedback shift register LFSR(a_1, a_2,... a_k) is defined by $x_{k+n+1}= x_{k+1}a_1+ x_{k+2}a_2+x_{k+n}a_k$, where $a_k \neq 0$, and all arithmetic is over GF(2). We say x_t=LFSR$_t$(a_1, a_2,... a_k). (x_1, x_2 ,..., x_k) is the key and the plaintext message is m_1, m_2 ,..., m_l, the ciphertext message is m_1+x_1, m_2+x_2 ,..., x_l+m_l. Given a corresponding plaintext and ciphertext messages of length t, "break" this cipher system. What is the minimum size to t for k?
4. Describe its construction of DES in terms of basic transformations. What is the role of the key schedule? S-boxes? What is linear cryptanalysis? Consider a

DES like cipher with the same key schedule and same high level single round, namely, ρ_i: (L,R)→(R, L+ f($K^{(i)}$, R), however, f is different in the following respects: (1) there is no expansion matrix and the first 32 key bits of the traditional DES round key is xored with R to produce the S-box input, (2) the permutation matrix P is replaced by the identity permutation, and (3) there is a single S-box, S, which takes four bit inputs and produces four bit outputs. The S-box is applied to each four bits of $K^{(i)}$+R, in succession. Thus f($K^{(i)}$, R)= $S(K^{(i)}+R)_{1,2,3,4}$ || $S(K^{(i)}+R)_{5,6,7,8}$ ||… || $S(K^{(i)}+R)_{29,30,31,32}$. S is defined as:

S(t,u,v,w)= (t+tw, u+uv, v+uvw, w+tw).

Find linear and differential characteristics of this per round function and analyze generally, without implementing, the prospects for linear and differential cryptanalysis of this modified cipher.

5. Describe the RSA public key system. Key generation, basis for safety, encryption process. Suppose p= 1493 and q= 1499. Calculate n and ϕ(n). If e=5 is the encryption exponent, calculate the decryption exponent, d.
6. Factoring using the $x^2=y^2$ (mod n) a la quadratic sieve. Suppose n=3837523. Observe that $9398^2= 5^5 x19^1$ (mod n), $19095^2= 2^2x5^1x11^1x13^1x19^1$ (mod n), $1964^2= 3^2 x 13^3$ (mod n), and $17078^2= 2^6x3^2x11^1$(mod n). Use these to find (x,y): $x^2=y^2$ (mod n). Finally, calculate (x-y, n) where (a,b) is the gcd of a and b to find the factors of n.
7. Describe a discrete log public key cipher over a finite field of characteristic p. What is the public key? The private key? Describe the encryption process using a "small" p (for example, p= 3467). Set k=[$\sqrt{3467}$]=58. Suppose α=5 and β =2717 where $\beta= \alpha^x$ (mod p). Find x as follows (Baby step, giant step). Compute a table (α, α^j (mod p)), for j= 1,2,…,k. Now compute, $\beta\alpha^{-kj}$ (mod p), for j= 1,2,…,k and find the intersection in the first table. Compute x from this. Finally, describe the Diffie Hellman key exchange protocol using exponentiation mod p.
8. Suppose $E_p(a,b)$ is the set of points (including the point at ∞) on the equation $y^2=x^3+ax+b$. Recall $E_p(a,b)$ is non-singular if $D=4a^3+27b^2 \neq 0$. How many points are on $E_{23}(2,13)$? Describe ECC encryption on $E_{23}(2,13)$? In the role of Alice, pick a public key for an ECC public key system on $E_{23}(2,13)$. Show how to embed the message m=7 in an point P_M on $E_{23}(2,13)$. In the role of Bob encrypt the message and in the role of Alice, decrypt it. What was the most computationally expensive procedure called for in this problem?

I hope you enjoyed the class. Please feel free to contact me if you have any questions in the future.

John