HW 10 SOLUTIONS CMSC 389. DUE Jan 19
SOLUTIONS

1. (0 points) READ my NOTES on RSA and SECRET SHARING- PARTICULARLY SHORT SHARES.

2. (40 points) (In this problem we outline how you can have a finite field of 4 elements.) Let $F = \{0, 1, x, x+1\}$. The coefficients are in mod 2, so $x + x = 2x = 0x = 0$. Multiplication will be such that whenever you multiply two numbers you replace any term of the form x^2 with $x + 1$.

   (a) Form the addition table for F. You need not tell us what 0 plus stuff is since $0 + blah = blah$. You can assume addition is commutative so you don't have to tell us both $a + b$ and $b + a$.

   (b) For every element in F say what its additive inverse is.

   (c) Form the mult table for F. You need not tell us what 1 times stuff is since $1 \times blah = blah$. You need not tell us what 0 times stuff is since $0 \times blah = 0$. You can assume mult is commutative so you don't have to tell us both ab and ba.

   (d) For every NONZERO element in F say what its mult inverse is.

   SOLUTION TO PROBLEM 2

   GRADING NOTE: The answers HAD TO be any of $\{0, 1, x, x+1\}$ could NOT be things like -1 or x^{-1} or $3x + 2$.

   - $1 + 1 = 0$
   - $1 + x = 1 + x$
   - $1 + (x + 1) = x$
   - $x + x = 0$
   - $x + (x + 1) = 2x + 1 = 1.$
   - $(x + 1) + (x + 1) = 0.$
   -
   - The additive inverse of 0 is 0.
   - The additive inverse of 1 is 1.

- The additive inverse of x is x.
- The additive inverse of $x+1$ is $x+1$.

- $x \times x = x^2 = x+1$
- $x(1+x) = x + x^2 = x + (1+x) = 1$

- The mult inverse of 1 is 1.
- The mult inverse of x is $x+1$.
- The mult inverse of $x+1$ is x.

3. (60 points) In the notes and class I told you how to, using RSA, have a secret sharing scheme where every share was $2|s|/t$. In the notes (and maybe in class- I am writing this before I gave class) I gave a scheme where you use two polys for the encoded secrete and one for the key that used shares of size $3|s|/t$.

   (a) Describe rigorously the scheme where you use three polys for the encoded secrete and one for the key. How short are the shares?

   (b) Describe rigorously the scheme where you use L polys for the encoded secrete and one for the key. How short are the shares?

   (c) Is there a limit to how many polys you should use?

SOLUTION TO PROBLEM 3

3a)

(a) Zelda picks a p, q, e, d that (1) satisfy the conditions of RSA, and (2) p and q are roughly $2^{|s|/3t}$, so $|p| = |q| = |s|/3t + O(1)$ and $n = pq$ is such that $|n| = |s|/3t + O(1)$. Henceforth we ignore $O(1)$ terms, so we take $|p| = |q| = |n| = |s|/3t$.

(b) Zelda computes $u = RSA(s)$. We assume $|u| = |s|$.

(c) Zelda takes $u = u_{1,0} \cdots u_{1,t-1} u_{2,0} \cdots u_{2,t-1} u_{3,0} \cdots u_{3,t-1}$. where all of the u_i's are of roughly the same length. We take $|u_i| = |s|/3t$.

(d) Let $F = GF(2^{|s|/3t})$. Note that all u_i are in F.

(e) Zelda forms THREE polynomials (over F)

$$f_1(x) = u_{1,t-1}x^{t-1} + u_{1,t-2}x^{t-2} + \cdots + u_{1,1}x + u_{1,0}.$$

$$f_2(x) = u_{2,t-1}x^{t-1} + u_{2,t-2}x^{t-2} + \cdots + u_{2,1}x + u_{2,0}.$$

$$f_3(x) = u_{3,t-1}x^{t-1} + u_{3,t-2}x^{t-2} + \cdots + u_{3,1}x + u_{3,0}.$$

(f) Let $k = (p, d)$ be a way to code p and d into one number. We can arrange things such that $|(p, d)| = 2|s|/3t$.

We use a field F' on $2^{2|s|/3t}$ elements. Zelda will ALSO secret-share the key k. This we do in the standard way; however we still describe it for completeness and so we can our analysis.

Zelda picks random numbers $r_{t-1}, \ldots, r_1 \in F$ (so $|r_i| \leq |s|/t$). Zelda forms the polynomial (over F')

$$g(x) = r_{t-1}x^{t-1} + r_{t-2}x^{t-2} + \cdots + r_1 x + k.$$

(g) Zelda gives A_i the numbers $f_1(i), f_2(i), f_3(i)$ and $g(i)$. Zelda also gives everyone (n, e) but we won't count that as a share since everyone gets it.

How many bits does Zelda give each A_i?

$f_1(i)$ is of length $|s|/3t$.

$f_2(i)$ is of length $|s|/3t$.

$f_3(i)$ is of length $|s|/3t$.

$g(i)$ is of length $2|s|/3t$.

Hence the total length is $5|s|/3t$.

3b)

(a) Zelda picks a p, q, e, d that (1) satisfy the conditions of RSA, and (2) p and q are roughly $2^{|s|/Lt}$, so $|p| = |q| = |s|/Lt + O(1)$ and $n = pq$ is such that $|n| = |s|/Lt + O(1)$. Henceforth we ignore $O(1)$ terms, so we take $|p| = |q| = |n| = |s|/Lt$.

(b) Zelda computes $u = RSA(s)$. We assume $|u| = |s|$.

(c) Zelda takes $u = u_{1,0} \cdots u_{1,t-1} u_{2,0} \cdots u_{2,t-1} u_{3,0} \cdots u_{3,t-1} \cdots u_{L,0} \cdots u_{L,t-1}$. where all of the u_i's are of roughly the same length. We take $|u_i| = |s|/Lt$.

(d) Let $F = GF(2^{|s|/Lt})$. Note that all u_i are in F.

(e) Zelda forms L polynomials (over F)

$$f_1(x) = u_{1,t-1}x^{t-1} + u_{1,t-2}x^{t-2} + \cdots + u_{1,1}x + u_{1,0}.$$

$$f_2(x) = u_{2,t-1}x^{t-1} + u_{2,t-2}x^{t-2} + \cdots + u_{2,1}x + u_{2,0}.$$

$$f_3(x) = u_{3,t-1}x^{t-1} + u_{3,t-2}x^{t-2} + \cdots + u_{3,1}x + u_{3,0}.$$

$\vdots$

$$f_L(x) = u_{L,t-1}x^{t-1} + u_{L,t-2}x^{t-2} + \cdots + u_{L,1}x + u_{L,0}.$$

(f) Let $k = (p, d)$ be a way to code p and d into one number. We can arrange things such that $|(p, d)| = 2|s|/Lt$. We use a field F' on $2^{2|s|/3t}$ elements. Zelda will ALSO secret-share the key k. This we do in the standard way; however we still describe it for completeness and so we can our analysis.

$$g(x) = r_{t-1}x^{t-1} + r_{t-2}x^{t-2} + \cdots + r_1 x + k.$$

(g) Zelda gives A_i the numbers $f_1(i), f_2(i), f_3(i), \ldots, f_L(i)$ and $g(i)$. Zelda also gives everyone (n, e) but we won't count that as a share since everyone gets it.

How many bits does Zelda give each A_i?

$f_1(i)$ is of length $|s|/Lt$.

$f_2(i)$ is of length $|s|/Lt$.

$f_3(i)$ is of length $|s|/Lt$.

$\vdots$

$f_L(i)$ is of length $|s|/Lt$.

$g(i)$ is of length $|s|/Lt$.

Hence the total length is $(L+1)|s|/Lt$.

3c) There is a limit. If the length is too small then RSA can be broken. So you need $|s|/Lt$ big enough so that RSA is still secure with primes this small.