Cryptology
Problems

To be discussed Monday, March 27, 2000.

In class we have discussed the discrete logarithm problem modulo a prime, which means that we have discussed them over fields of prime order. There are also finite fields of prime power order, so for any prime p and any exponent $e \geq 1$, there is a field with $q = p^e$ elements, $GF(q)$. The elements of such a field can be represented by polynomials over $GF(p)$ of degree no more than $e - 1$. The operations can be performed by working modulo an irreducible polynomial of degree e. For example, $y = x + x^5 + x^7$ is an element of the field $GF(2^{10})$, represented by $GF(2)[x]/(x^{10} + x^3 + 1)$. One can calculate a representation for y^2, by squaring y and then computing the result modulo $x^{10} + x^3 + 1$, so one gets $x^2 + 2x^6 + 2x^8 + x^{10} + 2x^{12} + x^{14} \pmod{x^{10} + x^3 + 1} = 1 + x^2 + x^3 + x^4 + x^7$. In Mathematica, you can use the PolynomialMod function to do these calculations. Thus, `PolynomialMod[(x+x^5+x^7)^2,x^10 +x^3+1,2]` raises y to the power 2, modulo the polynomial $x^{10} + x^3 + 1$, working modulo 2. Try raising y to the powers $e \in \{33, 93, 341, 1023\}$ to see what result you get. What do you get? What does this prove about y?

On my computer, raising to the power 1023 directly failed due to lack of memory. What does this say about how Mathematica did the calculations? What can you do to get around this problem when you try these calculations?

Why would there be a preference for working in $GF(2^k)$ for some large k, rather than modulo a prime for some very large prime? Hint: think about how arithmetic is performed.