CS 70 Discrete Mathematics for CS
Fall 2004 Rao

Problem 11

Solution to Problem 11

11. **Applications of Polynomials**

(a) Let $\overline{a_1 a_2 a_3 \ldots a_b}$ be the secret, where a_i are bits and $\overline{a_1 \ldots a_b}$ denotes the number represented by the bit-string. Using Lagrange interpolation we find a polynomial on $GF(p)$ of degree $k-1$ for which:

$$\begin{aligned} P(1) &= \overline{a_1 a_2 \ldots a_{\lceil b/k \rceil}} \\ P(2) &= \overline{a_{\lceil b/k \rceil + 1} \cdots a_{2\lceil b/k \rceil}} \\ \ldots &= \ldots \\ P(k) &= \overline{a_{(k-1)\lceil b/k \rceil + 1} \cdots a_b} \end{aligned}$$

Then we give $P(k+i)$ to person i for $i = 1, 2, \ldots n$. If any k people get together they can reconstruct the polynomial and the whole secret, since k points completely determine a polynomial of degree $k-1$.

Any $k-1$ people won't be able to reconstruct any chunk of the secret, and in particular the first $\lceil b/k \rceil$ bits. This is true because for any value of the first $\lceil b/k \rceil$ bits $c_1, c_2, \ldots c_{\lceil b/k \rceil}$ there exists a polynomial Q with $Q(1) = \overline{c_1 c_2 \ldots c_{\lceil b/k \rceil}}$ and the $k-1$ values of the present people.

The size of the field p has to be such that most of its elements are written using $\lceil b/k \rceil$ bits. So choose a prime p such that $2^{\lceil b/k \rceil - 1} < p < 2^{\lceil b/k \rceil}$.

(b) To correct k errors using an $n-1$ degree polynomial, it suffices to send $n+2k$ points. One way to decode is to try all combinations of k errors, and check if there is a degree $n-1$ polynomial that matches the remaining $n+k$ of the points. To prove that we'll never go wrong (assuming there are in fact only $\leq k$ error), suppose there are two different polynomials that work. Then they agree on $n+k$ points with the original set, and therefore they overlap on at least n points. Since they are degree $n-1$ that would make them identical.

In the next part we show that $n+2k$ points are the best we can do.

(c) i. First we look at the possible messages before any errors occur.
Suppose we increase the message by c, so the message consists of $n+c$ elements of the field. The field is of size f (for $GF(p)$ the size of the field is $f = p$). Therefore one may think that there are f^{n+c} messages that can be sent. However since the message is generated using a degree $n-1$ polynomial, and there are only f^n such polynomials (think of the possible values of the coefficients), in fact there are only f^n legal messages.

ii. Next we look at the possible events in the channel.
The channel introduces k errors, and it can choose any k locations for them. There are $\binom{n+c}{k}$ such possibilities ($\binom{a}{b} = \frac{a!}{b!(a-b)!}$ is the number of ways to choose b objects out of a set of a objects). Each error consists of changing the value to some other element of the field, so

the channel has f choices. The channel gets to do this for every of the k errors, so overall the number of actions the channel can take is $\binom{n+c}{k} f^k$.

iii. If we received the corrupted message and are able to recover the original, we can recover both the polynomial and the action of the channel.

iv. If we are able to decode, we can recover all this information - the polynomial and the action of the channel. So the message we received must have been long enough to carry this information. Notice that a message that tells us which of N things happened has to be at least $\log N$ bits long (this is a basic fact of information theory, that takes half a minute of thought). In our case, since the message told us which of $f^n \binom{n+c}{k} f^k$ things happened it must have been at least $\log\left(f^n \binom{n+c}{k} f^k\right)$ bits long.

$$\begin{aligned}
\log\left(f^n \binom{n+c}{k} f^k\right) &= (n+k)\log f + \log \binom{n+c}{k} \\
&\approx (n+k)\log f + \log n^k \\
&= (n+k)\log f + k\log n \\
&\approx (n+k)\log f + k\log f \\
&\approx (n+2k)\log f
\end{aligned}$$

Which is exactly how long a message of $n+2k$ elements of the field is.