Itinerary

Martin Sig Nørbjerg

October 3, 2023

1 1. Meeting

1. I would like to go over how the definition of Goppa codes in the notes, relate with the classical definition using Riemann Roch spaces.

   - Goppa codes werent actually the codes constructed on AG geometry.
   - We normally use sub field sub codes.
   - In addition how i should addapt Patterson's algorithm to use our approach? Alternatively I might use the approach explained in AG codes and some applications. (The PDF i sent to you previusly.)
     - Go through the notes.
   - What is the difference between syndrome decoding algorithms and regular decoding algorithms?
     - Syndrome decoding grows in exponential time.

2. Questions about the McEliece cryptosystem:

   - Why do we use AG codes instead of say RS codes, I am guessing that it is because the family of codes are larger and hence more secure is this correct or is there other factors at play?
     - Nice question: we should be very carefull when choosing our code, we need it to have very little structure.
     - Sub field sub codes.
     - The structure can be extracted by looking at the generator matrix.
   - They also mention that any other alternant code might also provide an equivalently good security, should I investigate alternant codes?
     - NO! forget it. Maybe have a look at McWiliams Lund.
   - Our public key is given by PGS where G is our generator matrix, $P \in \mathbb{F}_q^{n\times n}$ is a permutation matrix and $S \in \mathbb{F}_q^{k\times k}$ is non-singular. But why do we multiply the message by a random non-singular matrix? and how do we know for sure that PGS isn't simply a generator matrix of another algebraic geometry code?
     - We get an equivalent code. (They have the same properties.) This should look more like a random code.

- Why do we use our error to have weight exactly t (our error correcting radius), instead of picking some weight $\leq t$, wouldn't this increase the size of the search space?
    - If we take error $\leq t$ it is easier to decode. There are also algorithms for decoding $\geq t$, however

3. We will go with sage.

2 2. Meeting

1. Syndrome Decoding:

   - Last meeting you mentioned that the syndrome decoding algorithm grows in exponential time, (is this with respect to the number of errors, the dimension or length of the code. Or is it a combination?) also do you have a source that i may reference?
   - What is **complete** nearest neighbour decoding, i have heard of nearest neighbour decoding, what is the difference between the "regular" nn decoding and "complete" nn decoding?

2. Information Set Decoding:

   - Does my algorithm for computing information sets work (please see the lemma and algorithm below)? Most sources refer to different algorithms, where you for instance simply pick k columns uniformly and check for their independence, alternatively they also mention an algorithm where check if each new coloumn we add to our set is linearly independent of the previus columns.
     - I fear that my algorithm may not produce every single information set.

     Let $G \in \mathbb{F}_q^{k\times n}$ be the generator matrix of code $\mathcal{C}$, and G' be G reduced to row echelon form. Let I' be the tuple consisting of the indicies of the k pivot columns of G sorted with repsect to the canonical order $\leq$ on $\mathbb{N}$, then any set of the form $\{i_1, i_2 \dots, i_k\}$ with $i_j \in \mathbb{N}$ and $G'_{i_j,j} \neq 0$ such that $I'_j \leq i_j < I'_{j+1}$ for all $j < k$ and $I'_k \leq i_k \leq n$. Suppose $\{i_1, i_2 \dots, i_k\}$ is such a set, then there exists a permutation matrix such that $P \in \mathbb{F}_q^{n\times n}$ such that the pivot columns in GP has indicies $i_1, i_2 \dots, i_k$, since $G_{i,j} \neq 0$. Hence the columns $g_{i_1}, g_{i_2} \dots, g_{i_k}$ are $\mathbb{F}_q$-linearly independent and thus $\{i_1, i_2 \dots, i_n\}$ forms an information-set.

     [H] Construction of information sets IS Generator$G$: a generator matrix of $[n,k]_q$ code $G' \leftarrow G$ reduced to row echelon form $I' \leftarrow$ The tuple consisting of the indicies of the k pivot columns in G' $I \in \left\{\{i_1, i_2 \dots, i_k\} | i_j \in \mathbb{N} : G'_{i_j,j} \neq 0, I'_j \leq i_j < I_{j+1} \text{ for } j < k, I'_k \leq i_k \leq n\right\}$ **yield** I chosen uniformly and without replacement

   - In plain ISD we look for a information set I such that $wt(y - y_I G_I^{-1} G) \leq t$, where y is the recived word and t is the number of allowed errors, why is this less than or equal to t and not simply equal? (This is regards to p7_r3 i.e. the phd thesis of Carl Löndahl)

3. General Questions:

    - I really want to try decoding of algebraic geometry codes if possible, i can't get it out of my head, hence i have been looking around for proofs of the fact that classical goppa codes are algebraic geometry codes (or atleeast subfield subcodes), in this setting i was wondering about the following question:
        - If $\mathcal{C}$ is a $[n, k, d]_q$ code, with an efficient decoding algorithm $dec_{\mathcal{C}}$ and $\mathcal{C}*$ is a subfield subcode of $\mathcal{C}$, is there a way to modify $dec_{\mathcal{C}}$ to work for C^*?

3. Meeting

1. I'm a little uncertain how the security level (in bits) is computed. My understanding is that it measures the average number of operations needed to crack the encryption by our current best algorithm. However this does not sound very objective. And wouldn't the security level drop over time as we develop new and better algorithms?

2. Should I look into list decoding (or only if I have time at the end of the project)

3. With regards to information set decoding, I have written sections about plain information set decoding and Lee-Brickell's algorithm and computed the expected work factor of each. I know that there exists a better algorithms (sterns) however I don't really find the topic of information set decoding very interesting compared to the rest of the topics, so I am wondering if I could simply explain the algorithm, and skip computing the expected work factor (perhaps simply reference a source instead.)

4. Both the basic and error correcting pairs algorithms require us to have the ability to efficiently compute bases for Riemann-Roch spaces, I found the paper "Computing Riemann-Roch spaces in Algebraic Function Fields and Related Topics", by F. Hess. It seems good but VERY technical and takes the approach of algebraic function fields instead of the approach using algebraic geometry which I'm more familiar with. Do you have an alternative source?

    - I think that it will be sufficient with a way to compute bases for plane projective curves. Please correct me if I'm wrong.

5. With regards to differentials and derivations, why is derivations only required to be $\mathbb{F}$-linear (where $\mathbb{F}$ is our ground field) while differentials etc. are required to be $\mathbb{F}(\mathcal{X})$ linear.

6. Do you know of a proof of the following theorem or do you know of a source which proves it? Found it in "Codes, Cryptology and Curves with Computer Algebra" by Ruud Pellikaan and others, but they do not provide a proof / source:

    - Let t be a local parameter at $P \in \mathcal{X}$. Then there exists a unique derivation $D_t : \mathbb{F}(\mathcal{X}) \to \mathbb{F}(\mathcal{X})$ such that $D_t(t) = 1$.

7. On a more personal note, I could use some career advice as I'm unsure if I should try to go into academia or go into industry:

    - What sorts of jobs can we get in the industry as discrete mathematicians?
    - Do you know of any companies in the area, that I could look into to get an idea for the sort of work?
    - What could a path into academia look like?