Data Representations:

Gen. - <https://computersciencewiki.org/index.php/Data_representation>

* <https://en.wikipedia.org/wiki/List_of_file_formats>

Sound - <https://cs.wellesley.edu/~cs110/reading/sound.html>

* Hour of cd quality stereo music = ~634 MB
* File size (bits) = bit rate \* recording time {where bit rate = bit-res \* sampling rate}

Hex - <https://en.wikipedia.org/wiki/Hexadecimal#Conversion>

* Bitwise - <https://en.wikipedia.org/wiki/Bitwise_operation>

Ideas:

(idrk how we wanna do anything or what’s important to you exactly, and I’m not trying to push or change your vision and idea for it, but here’s some stuff that popped in my head that could maybe be helpful)

= I feel like data loss or distortion is one of the major things, like if we were able to maintain the data to keep consistency, so we can take an mp3 to jpeg to txt and back to the exact same mp3, it would be really cool.

* Maybe with an option for the user to maintain all data or not? So that way a picture to password would be more viable, and wouldn’t have to be super long, but that depends on how we translate and encode to text and everything.
  + Sub-idea/thought, if we let user discard some data, with this algorithm wouldn’t it essentially drop the resolution (or lose some pieces) of a picture if you translated the password back to the picture?

= Intermediary stage, decompose all data into one generic type ⇒ build files up with generic type

* I don’t know why, but I hate this idea now. It feels like part of this could be used maybe, but in general it feels too oop and bound to give side effects or distort data, might be how I’m thinking about it.

= Could we utilize the structure of different data types/formats in an algorithm for translation across? Like, identifying that an mp3 has structure X and a jpeg has structure Y feels like it would be helpful, right? Am I just high? Nah, 5th

= Honestly I was thinking that if we could make this provable and give the proof, like that input file is exactly equal to the returned file, it would be really cool. I don’t know how you feel about that, but it’d be doable I’d think, especially if we choose the language well and make it in a functional way.

* e.g., file.mp3 == file\_copy.jpeg (maybe needs intermediary for proof; file.mp3 == ??? == file\_copy.jpeg)

I’m thinking the proofs will look like this:

file.txt -> file.png -> file.txt

file.mp3 -> file.png -> file.txt -> file.mp3