Quantum Resistant Algorithm Encryptor

FOR CISC-4900-VC1B WITH PROFESSOR CHUANG

SUPERVISED BY MATTHEW MCNEILL
<MCNEILL@BROOKLYN.CUNY.EDU>

BY AHMED OMER
<AHMED.OMER68@BCMAIL.CUNY.EDU>

ABSTRACT

Quantum Chips are on the rise;

And they seek to completely invalidate our cyber security. To combat this, many cryptographers are working time in and out around the globe, to workshop solutions. Of these solutions, there resulting were many quantum resistant algorithms; but they are of little public awareness; which is an issue. As we stand; we are ahead of what could legitimately be called a decryption crisis- and consequently, all we *need* to do to combat it, is spread awareness.

The project's core response;

Which leads to my project, specifically- my project is a runnable windows application with its own install wizard; focusing primarily on the *educational* function of introducing, and explaining, multiple encryption types, while also fulfilling the *exemplary* function of working wholesale also as a file encryptor *using* said types.

The projected end-result of the project is a Windows Application (7+) with it's own install wizard, internal-file-storage within the user's directory(ies), and a hookup to Window's uninstall manager system client.

ABSTRACT(cont.)

The project's core response(cont.);

From there, the user will have access to a minimum of three encryption types; AES standard encryption, a Quantum-Resistant-Algorithm Based Vector-Lattice encryption (exemplifying IRL minor usage via algorithms such as KYBER or DILITHIUM), and an unchosen third. Then, beyond access to the base encryption & decryption of their files (such as when in using the apps the likes of WinRAR or 7-Zip); there will additionally be a blurb *explaining* each algorithm, its current history, common places of usage, and, if a *Quantum-Resistant* Algorithm, how it's stated in in tech literature to stack up to projected findings and thoughts on Quantum-*Chips*.

For complete accuracy of the project, while unflattering, it must be said that this is majorly a *learning project*. Before this project, I have had never made a proper Windows application, much less an install-wizard; and while my knowledge of cyber-security is *possibly* better than most; it is not my major for a reason. As such; the projected completed project, while *looking* to be satisfactory in its totalized development; it will most likely NOT be anything… striking.

TOOLS & REQUIREMENTS

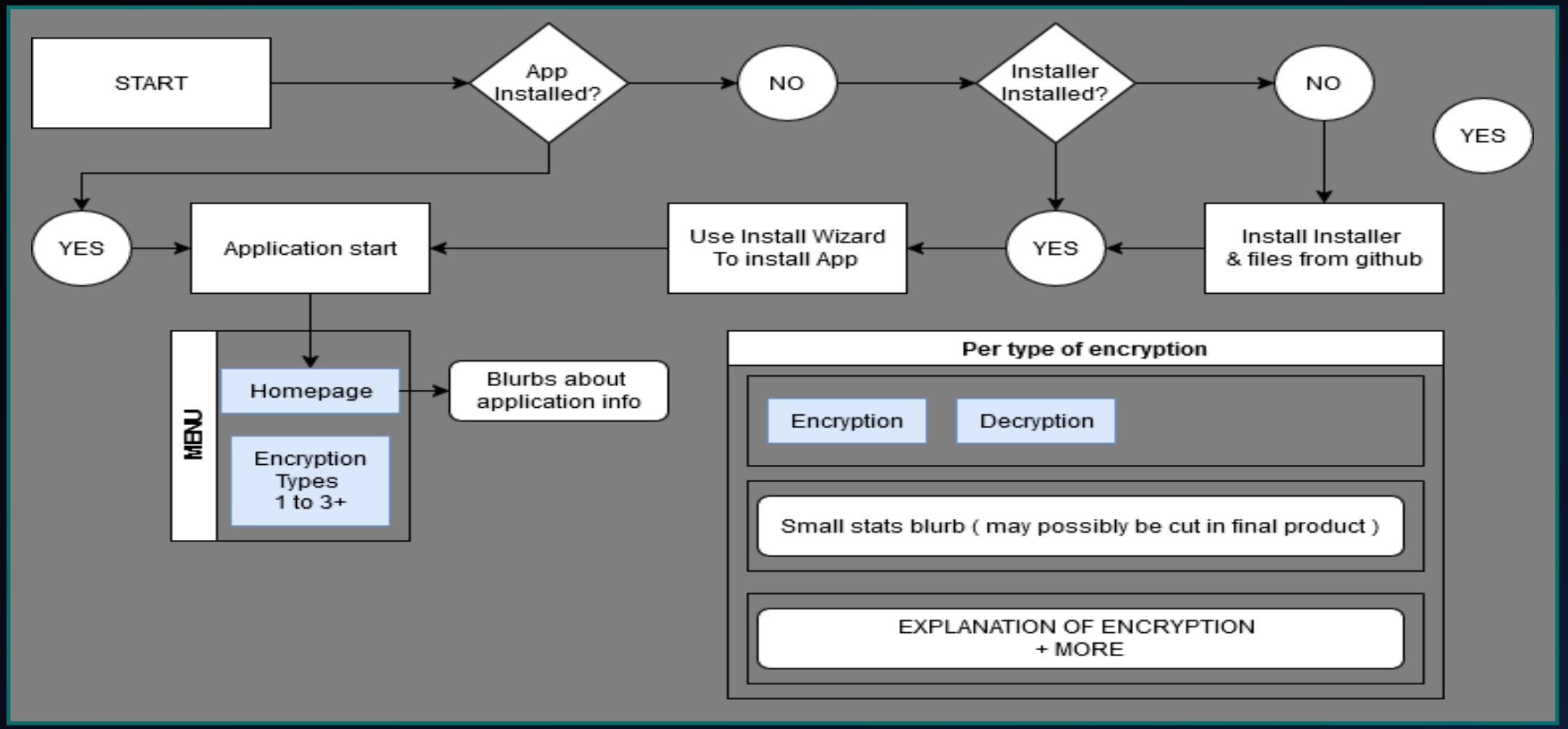
User-end;

- Windows 7+
- Upwards to 3 MB of free space (looking to be much, much smaller; >1MB, but overestimation is safer)
- Internet access to download from Github, and nothing else! Every other file/thing required is packaged within the installer.

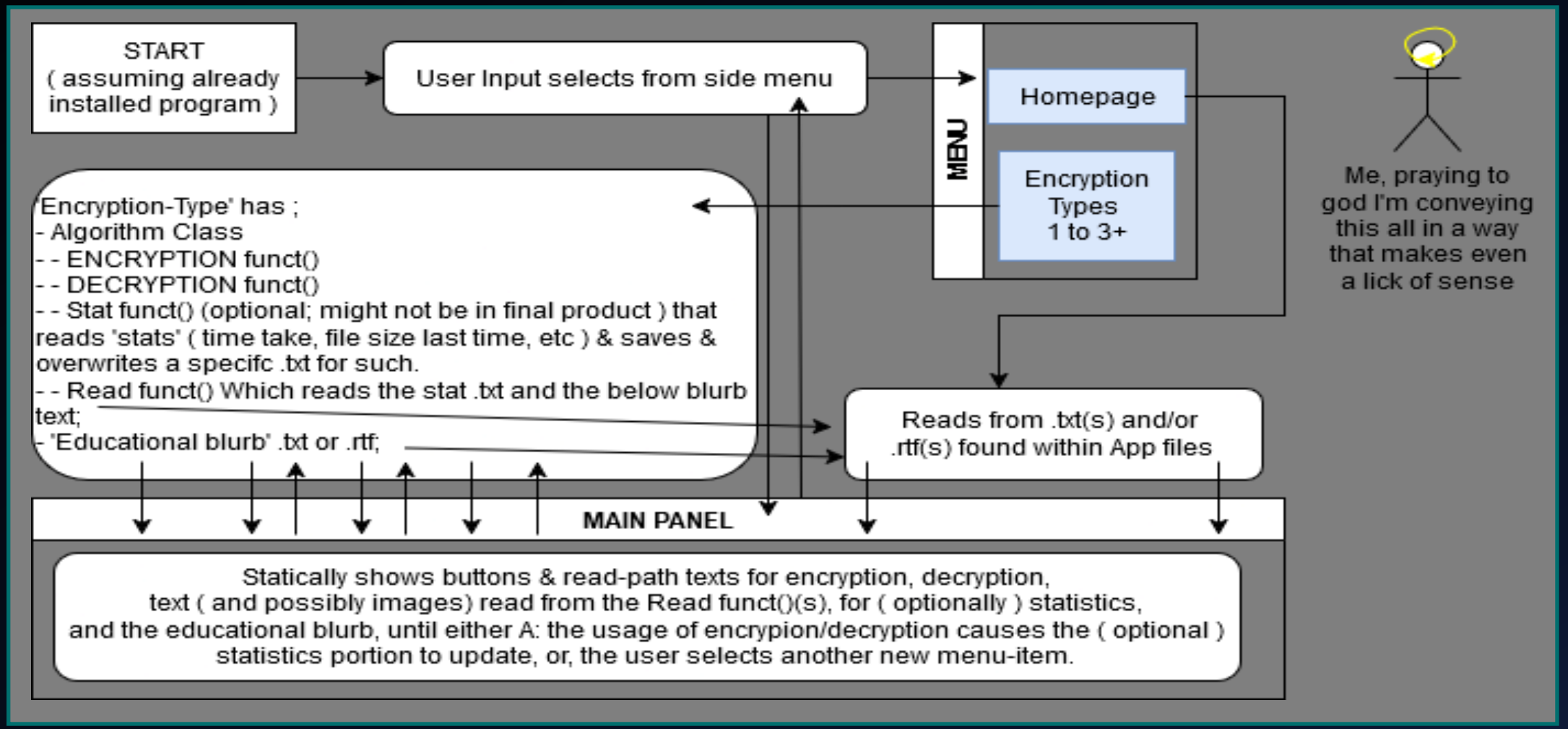
Project-end;

- Visual Studio 2022, version 17.12.4+
- My Laptop, & it’s Windows OS
- Microsoft’s development resources (online)
- And nothing else!

PROPOSED USER INTERFACE FLOW



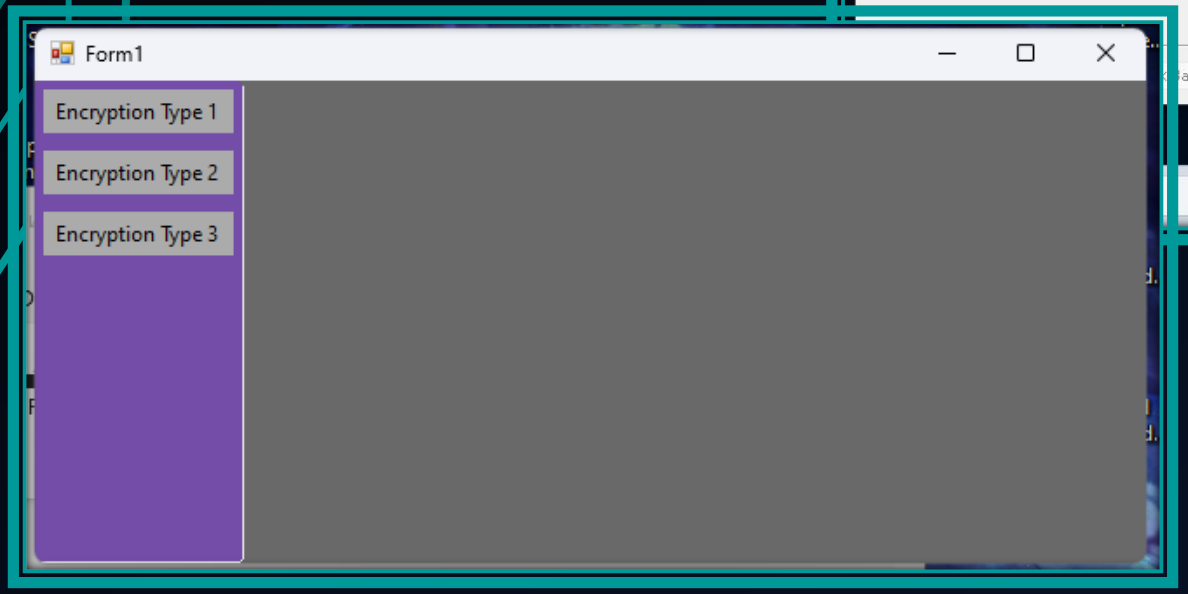
PROPOSED GENERAL APPLICATION FLOW

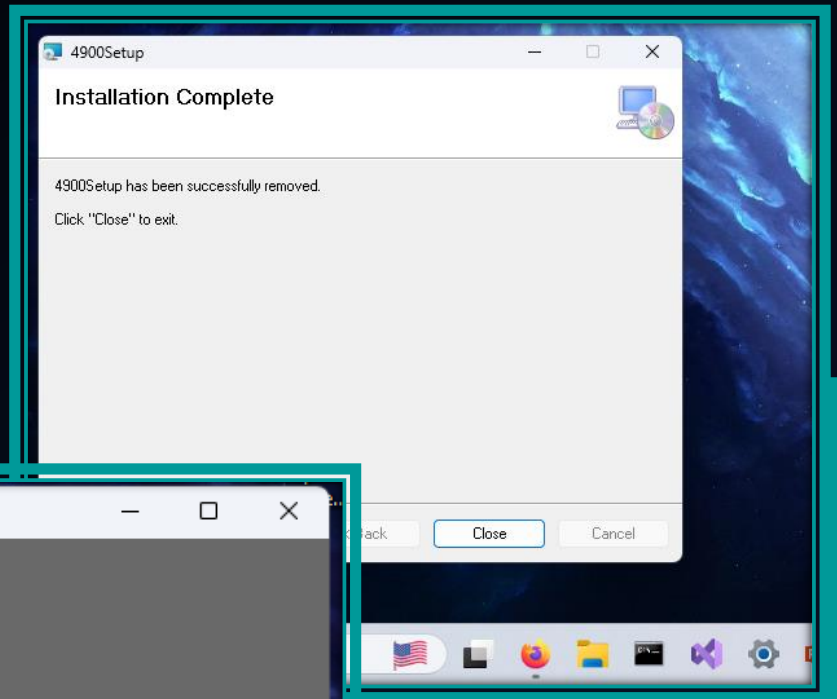


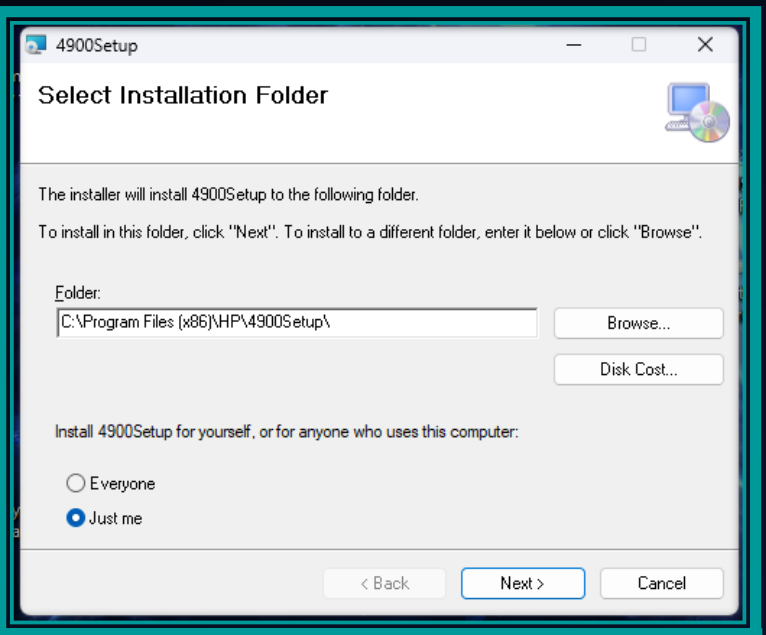
SO FAR IMAGE INSTANCES;

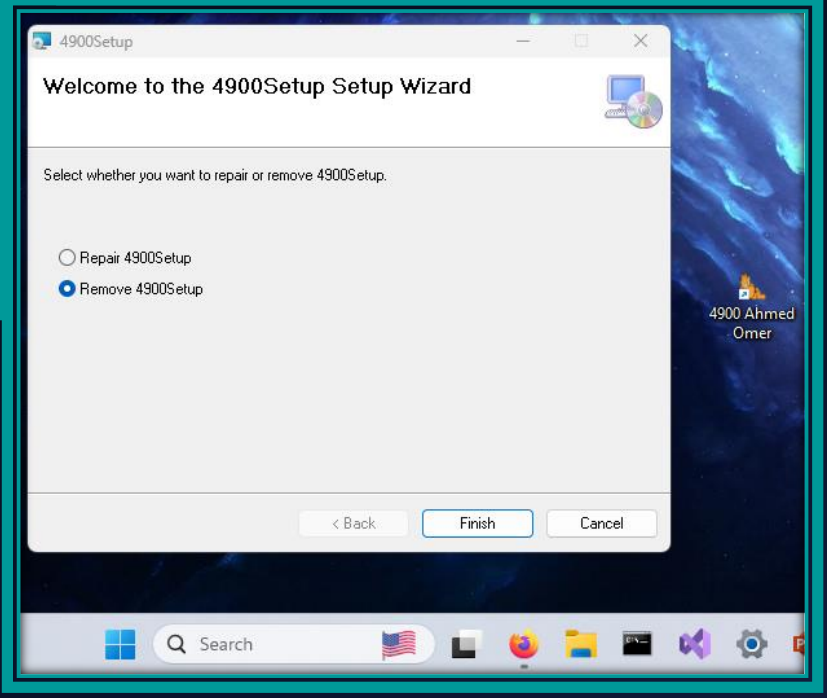
Left to right;

- Install example
- Shown existence
- Uninstall example
- The growing UI of the app itself!









<----------

Note how there isn't a Home Page yet; I want to focus on creating the encryptors first, before looks towards frivolous things like that

THE THREE CHUNKS OF THE PROJECT;

In general;

I've split my project into three 'general chunks'; each expected to mostly take 4 weeks each, maximum. This way, for the last four weeks, I can focus more on general improvements, having already completed the focus-items.

The application wizard, windows, and the App itself;

For instance; the install wizard, and it's install capability is already finished! Throughout these last three weeks, I haven't just been just focusing on it; I've been thinking of a project, finding a supervisor, etc; but despite that, my general time-goal has been accomplished; this week, I'll be mostly adding panel-switch functionality, and buttons, before I quick-switch to adding the encryption functionality, proper;

The encryptors, encryption, and decryption;

In this segment; the en&decryption functionality,; being the incorporation of AES & the lattice encryption; in that order (AES to establish a baseline). Then, DURING this segment *also*, finding another suitable third, alongside it's own implementation.

THE THREE CHUNKS OF THE PROJECT(cont.);

The educational segment, and tidying up;

And finally, possibly the easiest, and simultaneously hardest, segment; writing the educational blurbs, and user-testing. I am not a writer; it's why I didn't go for a CW degree- that being said; user-testing will probably the harder segment. I'm planning on finally fixing an old Desktop I have, and factory-resetting it to a windows standard, before downloading the install wizard, and having some family members, and neighbors try to navigate the download, the application itself, and the educational segments.

I figure they'll be a good litmus test, because in all fair's fair; most of the people that live directly around me are over thrice my age. *IF* I can manage to finagle all these events together, I'll probably have a very, very small survey for them to fill out, and tweak the program a bit after a single full round in response, to submit my results.

If I can't, however, schedule this with my neighbors & family; my secondary plan is, honestly, to either A; annoy my college friends about it, or B; ask for the 4900 faculty's help for possible resources, or background methods of testing the educational/explainational part of my project!

THE THREE CHUNKS OF THE PROJECT(cont.);

When, specifically, do you plan to accomplish these tasks(?);

The Application Wizard & it's basis;	weeks 3-5
The AES & Lattice encryptor;	weeks 5-7
Third encryptor find, & incorporation;	weeks 6-8
The educational segment, & testing;	weeks 10-12
Tidying up & adding anything extra	weeks 12+

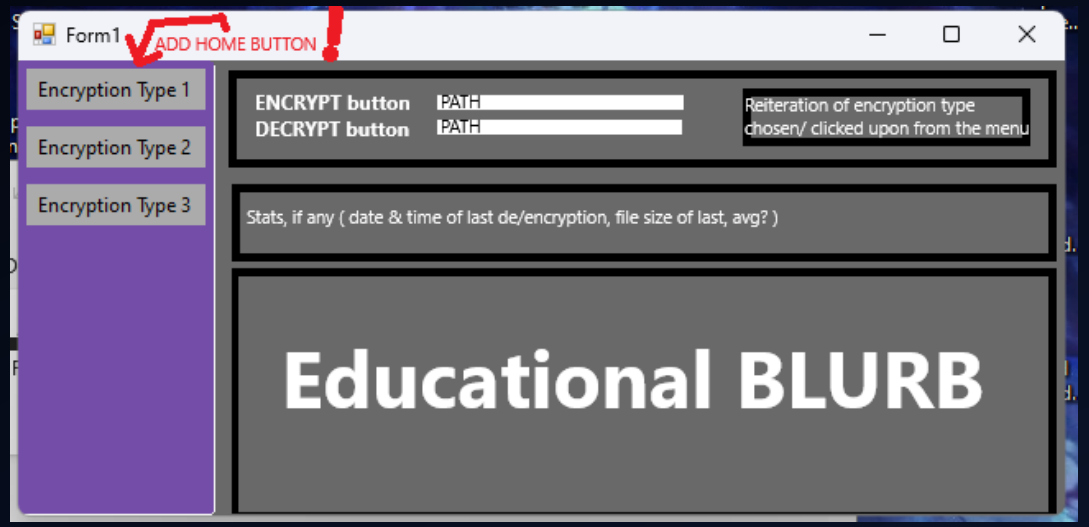
You might notice these are different times than I projected for the Project-Proposal survey; this is because despite the simultaneous workings of finding, researching, securing, and setting up a project's idea, alongside the project itself, I managed to create the installer fairly quickly, thanks to a mixture of Microsoft's resources, and my willingness to switch language types (just a jump from C++ back to C#; but the .NET basis for application-building has been so standard for so long, resources are abundant).

This has been overall a major boon for my projected *best* times, since now, average case, I have some wiggle-room for time-sensitive project emergencies!

Some quick final portions/thoughts;

specifically, Uses for this project (examples);

- As a regular file encryption, decryption program for windows (examples of such include WinRAR, 7-Zip, DiskCryptor, etc)
- As a Working example on your windows machine of Quantum-Resistant encryption, decryption; plus, since consumer versions aren't common, slightly more obtuse file-protection for the average user who choses to do so- an actually legitimate application selling point
- And as an encryption educational tool! While I'm 100% my specific program won't be of said standard, I can easily imagine a more professional version of my project being done by, persay, a educational company, and selling it to add to highschool computer



Final thoughts;

This has been fun so far! As I've mentioned prior, this is majorly a learning project for me- and I've been learning a lot! Both code-base wise, and project management wise (ha)! So, I just want to say two things;

Thank you for the chance for this experience! I'm not gonna start counting my chickens before they hatch, but things have been going swell, so far!

And secondly; I really, really hope I understood this diagram assignment correctly (lmao)!

Cheers!

Quantum Resistant Algorithm Encryptor -END

FOR CISC-4900-VC1B WITH PROFESSOR CHUANG

SUPERVISED BY MATTHEW MCNEILL
<MCNEILL@BROOKLYN.CUNY.EDU>

BY AHMED OMER
<AHMED.OMER68@BCMAIL.CUNY.EDU>