Overview of the System

Why Multi-Project Files/What the Software Does

This software that reads in the file is designed to open one or more projects at a time. The user can make edits to each project within the same application instance and save them accordingly. How the product actually uses these projects can be considered out of scope and will not impact the overall design of the encryption/decryption mechanisms.

Key Management

The asymmetric key pair is stored using the user's own Cryptographic Service Provider, provided by Windows. Today, this means that the key is unique for each machine. The product owners are aware that once these files become encrypted using these keys, the projects can no longer be shared across machines. This is a loss they are working to settle with some future technologies they are investigating (such as Public Key Infrastructure tools). They still insist that the current set of keys be used in the design in the meantime, but to allow for this to be easily changed out in the future.

How the Software Reads from and Writes to the File

The project files have the following nodes for each project: name, project ID, last modified, and data. The name and project ID work together to reduce complications around projects that share the same name further down the road. When the software launches, the user is presented with a "Welcome Screen" that lists "Recent Projects" they've opened recently along with an "Open Projects" button. This gives them the flexibility of opening a project they just worked with or open a new project or one they haven't opened in a while. This has always been a very fast process and the application launches and presents this screen within a few seconds of the user launching the app.

Upon selecting the project(s) they wish to open, the software reads in the "data" portion of the project(s) they selected and attempts to load in as much data as possible. If there are complications with opening a project, an error message is displayed within the project's console window at the bottom of the user's screen indicating that there were some parts that could not be read in successfully. These errors rarely ever happen, and if they do, an anonymous bug report is submitted so that this can be patched in the next release. Everything else within that project is opened, so as to minimize frustrations the user might have with losing data.

Writing to the file involves the software reading in and parsing the general xml structure to determine where in the file that project lives. Then it will overwrite the appropriate contents within that file (including changing the project name if the user so chooses) and updating the "Last Modified" field before writing that back out to the file system. Auto-Save is not supported at this time.

What's inside the <data> Node?

The data node contains the entirety of the project's core data that's needed for the project to work, *including product-sensitive data which spurred this need for encrypted files.* The structure of the contents within this data node is varied with every release of the product. There's no real set structure that has been decided upon yet as the software is still fairly new. Assume that the entire contents should be encrypted and that there's not a good way to structure any encryption within the node at this time. Any attempts at encrypting individual elements within the data node would require constant maintenance with each release.

Desire for File Integrity Checks

This is a feature that the business is greatly considering adding to the product. There's a desire that, while the contents of the data node continue to vary between version releases, they want to ensure that the files don't get modified outside the product so as to cause greater risk of file corruption upon opening a particular project. This is certainly not part of the scope of the problem, but it is an *optional* feature they are considering adding in at some point.

Project File Encoding

This project file has always been stored as plain text and will be for the foreseeable future. Any non-ascii characters should be converted to ascii, through the use of Base64 encoding.