Abstract

Security and privacy are hot topics now. Today there are many systems that store our files but you are definitely not sure whether your files are 100% safe. The reason is you are storing the files directly into their servers, which is clearly not the one you want. On thinking about the way we can store data securely, we found a model that encrypts the file in client side without even referring the server. Therefore, the file that you will get is simply an encrypted file that is going to be stored in client machine. The application that we built is a responsive web and a hybrid mobile application, which means that it will work the same in all screens and in all platforms, whether it is IOS, Android or Windows, the application is same. We found issues with the system that we built but we also found the solution for the same. When it comes to encrypting data and securing information, people naturally expect the page to be loaded through HTTPS. In this case, I believe it is not necessary, as apart from the initial download of the HTML and assets, no data is been transferred between you and the server – everything is done client-side with JavaScript. User who encrypt the file does not need to remember the password all the time. They can provide some questions which gives the password as the answer. No worries we are not encrypting the file simply from the answer for those questions. We are generating a random number for each user and doing a simple calculation based on the random number. Thus, every time when the user does an encryption operation a random number is been added with the passphrase and the encrypted file will be available for them. Thus even if they remember the answer for those questions, they cannot see your file unless they login. We can share the encrypted files only with few group of people thus providing access only to them and creating a secured workgroup. Thus, we do not need to worry about the file that they are storing in different servers because they anyway need to use the above model to decrypt it thereby creating the most secured file system.