Problem Statement for JScrypt

Jean Lucas Ferreira, Ocean Cheung, Harit Patel

September 27, 2015

1 What problem are we trying to solver?

When developers are creating authentication services for their web applications, there are multiple ways to store sensitive information about the user. The common techniques for this can usually be resolved with hash functions, such as MD5, SHA1, SHA256, and many more. Even though these hash functions are irreversible, they come with a weakness. Due to their popularity and usage in many applications over the years, hackers have found efficient brute force techniques for cracking the hashes, through rainbow tables, and hash dictionaries. The problem we are trying to solve is to make brute force attacks on encrypted information seemingly impossible, which can be solved by using bCrypt.

2 Why is this an important problem?

Encryption techniques are important for creating a secure and reliable world where technology can be trusted with sensitive information. Society has learned to trust technology and have also benefited greatly from the conveniences created by online banking and online shopping. Even in the event of a security breach into a database, the encryption adds an extra layer of security to protect sensitive information such as passwords or credit card numbers. For example, when Sonys PlayStation Network encountered a security breach in 2011, 77 million users were affected [1]. Even though hackers found a security vulnerability to breach the network, information such as user credit card numbers were still secure and encrypted, making it useless to the hackers.

3 What is the context of the problem you are solving?

Scope of the problem includes developers lacking a reliable, user-friendly, and a robust encryption library to bar access to sensitive information from unauthorized users. By utilizing bCrypt, developers will be able to create more robust encrypted passwords with minimal effort. Additionally, they will be able to authenticate users easily given functions that will compare the user provided password with the encrypted password. Moreover, the functions and algorithms provided by bCrypt will not impede or affect the developers projects or the user experience in any way. Developers worldwide will be able to take advantage of this lightweight and user-friendly library to secure sensitive information.

References

[1] Keith Stuart, Charles Arthur, *PlayStation Network Hack*, The Guardian, Wednesday 27 April 2011.