# Test Plan for JScrypt

Jean Lucas Ferreira, Ocean Cheung, Harit Patel

October 23, 2015

# Contents

# 1 Test Plan Identifier

JScrypt Test Plan

# 2 References

- JScrypt Software Requirement Specification Revision 0
- IEEE 829 Test Plan Template

# 3 Introduction

This is the test plan document for JScrypt revision 0, and will be primarily responsible for addressing the testing plan of the functions implemented for the project.

## 3.1 Objectives

The following objectives must be accomplished in the test plan revision 0, to facilitate the creation of the test document by the developers.

- Describe the environment and testing tool(s) required in order to produce test cases.
- Define all functionalities to be tested, and how they will be tested. For example, which inputs and testing method will be used for a given function.
- The schedule and workload for each person responsible in creating the test cases for this project.

## 3.2 Scope

This Test Plan is focused on describing how each test case should be constructed, and the appropriate input, and expected outputs. It will not go into detail of how each function being tested is implemented.

## 3.3 Constraints

The Test Plan is susceptible to two constraints:

1. Time constraint: the test plan must be complete by October 23, 2015
2. Personnel Constraint: there is only 3 individuals working on the test plan.

### 3.4 Abbreviations, Acronyms and Definitions

*Many keywords used in this document can be found in section 5.1 of JScrypt Software Requirements Specification Revision 0 document.*

- **ASCII:** The most common format for text files in computers and on the internet.
- **Front-End:** The visual representation of a website, made from HTML.
- **Hash String:** A string generated by the program, and will consist of the version, number of rounds, salt, and encrypted key. It has the format: *$ version $ roundNumbers $ salt...encryptedkey.*
- **HTML:** Hyper Text Markup Language, a language that allows constructing of websites.
- **Salt:** Random data that is used as an additional input into a function which encrypts a string.

## 4 Test Items

The following is a list of items which may be used to test the JScrypt implementation:

1. Implementation is able to hash a string.
2. Implementation is able to compare a string with a hash.
3. Implementation is able to take a hashed string and unhash it.
4. The encryption time changes exponentially with increase cost values.

## 5 Software Risk Issues

There exists a few risks with the software, and the development of the software that could affect the final implementation of the project: **Developer Risk Issues:** Implementing an encryption algorithm is not easy, and should be left by security and encryption specialist. Given the knowledge scope of the developers, this project may be susceptible to loopholes in the encryption process.

## 6 Features to be Tested

The following features of JScrypt will be focused on during testing of the application:

1. The ability to convert defined character(s) into an abstract set of characters with no meaning.

2. The ability to compare character(s) with an abstract set of characters to compare if they match.

3. Implementation is able to take a hashed string and unhash it.

4. The encryption time changes exponentially with increase cost values.

# 7 Features not to be Tested

The following features of JScrypt will **not** be tested in revision 0:

1. **Feature:**The front-end of JScrypt, where an HTML form accepts an input key, and on submit, it provides the encrypted key produced by the JScrypt algorithm.
   **Reason for not testing:** his feature is does not affect the implementation of the project, and is simply a means for the user to see the format for which the algorithm encrypts a key, and some of the features that are available with the project. But again, since it is not a critical feature for the functionality of the project, it will be tested in a later revision.

# 8 Approach

## 8.1 Testing Tools

The following tools will be required for testing this project:

- Mocha: a JavaScript test framework
- Chai: a JavaScript assertion library
- Sinon: a JavaScript stubbing and mocking library

## 8.2 Testing Tools Requirement

**Environment:** The testing frameworks require a NodeJS environment in order to successfully run the test cases
**Training:** The testers of the project must be familiar with implementing JavaScript modules within a NodeJS environment, and be responsible for learning the required testing frameworks.

## 8.3 Overview of Testing

JScrypt will be required to go through various tests of its functions. The developers will be in charge of testing the software they implement. The test report for the proof of concept with consist of unit testing, and white box testing. Unit testing will help the developers validate that the correct output is being generated by the function, and white box testing will ensure that all components of the function are working accordingly.

## 8.4 Tests Specific to Proof of Concept (PoC)

These are the functions that will be implemented for the PoC, and will be tested accordingly.

1. **Function Name:** generateRandomSalt
   **Description:** Generates a random padded string of length 24, which is used for hashing the key. This string includes padding which will later be removed before hashing the key.
   **Input:** Rounds; number of times the key is hashed.
   **Expected Output:**A random string of length 24 which is comprised of ascii characters.
   **Test Implementations:**
   *Test case 1 implementation:* Given input rounds as an integer between 6 and 31, the function shall generate some random data. If the generated data is of type string, the test shall pass.
   *Test case 2 implementation:* Given input rounds as an integer between 6 and 31, the function shall generate some random data. If the generated data has a length of 24, the test shall pass.
   *Test case 3 implementation:* Given input rounds as an integer between 6 and 31, the function shall generate some random data. If the generated data comprises entirely of ascii characters, the test shall pass.

2. **Function Name:** hashKey
   **Description:** Generates the hash string (refer to section 3.4) that will be stored in the applications database. This is the only function that a user of the project will be required to call in their application.
   **Input:** key (string of length 1 to 56 ), rounds (integer from 6 to 31)
   **Expected Output:**A hash string of the format represented in 3.4
   **Test Implementations:**
   *Test case 1 implementation:* Given the input key as an empty string and rounds, a value between 6 and 3. The test case shall not pass.
   *Test case 2 implementation:* Given an input key as a string of length between 1 and 56, and rounds as integer not within the range 6 to 31. The test case shall pass, since invalid rounds are automatically set to a default round.
   *Test case 3 implementation:* Given an input key as a string of length between 1 and 56, and rounds as integer within the range 6 to 31. The test case shall pass.

3. **Function Name:** getRounds
   **Description:** A string can go through a series of rounds to become hashed and finding the number of rounds it went through gives knowledge to help unhash a string.
   **Input:** A hash string (refer to section 3.4)
   **Expected Output:**number of rounds or null
   **Test Implementations:**
   *Test case 1 implementation:* If the hash string inputted did not follow the $version$rounds$... format with the three $ signs specifically then the test will not pass.

*Test case 2 implementation:* If the hash string inputted followed the $version$rounds$salt...key format, the number of characters in the salt and key are equal, if the version of JScrypt is up to date, and if the round number is a number between 6 and 31, then the test will pass.

## 9 Item Pass/Fail Criteria

The tests will provide an input string for JScrypt to encrypt and return the encrypted string. The tests will verify the returned string can also be decrypted, and that the decrypted string is the same as the original input string. Once these tests have successfully completed, the item will be passed. Additionally, the tests shall be comprehensive and cover much of the code.

## 10 Suspension Criteria and Resumption Requirements

If the encrypted value of the input string is comprehensible or does not follow the preferences provided by the user, then the testing algorithm will be suspended. Additionally, if the encrypted value cannot be decrypted, or once the value is decrypted it is different from the original input string, then the testing algorithm will be suspended. The testing algorithm will be resumed once the code representing the stated issues has been changed.

## 11 Test Deliverables

The following deliverables are required:

1. Test Plan Revision 0 document

2. Test Report Revision 0 document

## 12 Remaining Test Tasks

1. **Task Remaining:** Set of unit-tests in mocha test framework.
   **Completed by who/when:** All team members, by October 26, 2015. (revision 0 PoC demonstration)

2. **Task Remaining:** Back-End system implementation test plan.
   **Completed by who/when:** All team members, by October 26, 2015 (revision 0 PoC demonstration)

3. **Task Remaining:** Front-End implementation test plan.
   **Completed by who/when:** All team members, by November 30, 2015 (revision 1 demonstration)

# 13 Environment Needed

The following are needed to test the JScrypt project:

1. Installed Node.js on the machine being used to test.
2. Installed Mocha testing framework on the machine used.
3. Installed Chai testing framework on the machine used.

# 14 Staffing and Training Needs

All developers responsible for creating the test cases must gain basic knowledge of the testing frameworks (Mocha, Chai) used for this project.

# 15 Responsibilities

All team members have equal responsibility among the development of the project. Any design decision chosen by a member must also be approved by all team members. The project supervisors (Dr. Spencer Smith and Li Peng) have superior responsibility among all team members and may mandate design decisions of the project, given logical reasoning.

# 16 Schedule

Table 1: Schedule

| **Task** | **Completed By** | **Member Assigned** |
|---|---|---|
| Proof of Concept Demo revision 0 | October 26, 2015 | All |
| Proof of Concept Demo revision - Unit Tests | October 26, 2015 | All |
| Design Document revision 0 | November 6, 2015 | All |
| Revision 0 Full demonstration | November 9, 2015 | All |
| Revision 0 Full demonstration - Unit tests for all JScrypt functions | November 9, 2015 | All |

# 17 Planning Risks and Contingencies

Overall risks to the project, specifically with the testing process:

- Lack of personnel when testing is to begin.
- Delays in fully understanding the technology used for testing.
- Delays in understanding the algorithms that are to be tested.
- Changes in the implementation or design of the project.

Contingencies for the stated risks:

- Tests may be required to be less comprehensive.
- Number of tests written and performed will be reduced.
- The team will work overtime.
- Certain aspects of the project will be prioritized and only those aspects will be implemented.

# 18 Approvals

Project will be deemed approved and allowed for development continuation by the project supervisors: Dr. Spencer Smith, and Li Peng