Suppose you wanted to extend the JavaScript Object such that getting and setting data in the object is only possible when providing a predetermined access key. The key is first used to instantiate the SecureObject and subsequently used when storing and retrieving data from the SecureObject. Provide the implementation for a JavaScript function that satisfies the following use case:

/\*\*

\* Instantiate a SecureObject

\* SecureObject(accessKey)

\* @accessKey - a string representing the read/write access key

\* returns an instance of a SecureObject

\*\*/

var secureObject = SecureObject("secretKey");

/\*\*

\* Store some data into the SecureObject

\* SecureObject(accessKey, identifier, data)

\* @accessKey - the key used to instantiate this SecureObject

\* @identifier - a String used to map to the data

\* @data - anything can be used as the data

\*

\* returns nothing (undefined)

\*\*/

secureObject("secretKey", "foo", {a: 1, b: 2});

secureObject("secretKey", "bar", "myValue");

/\*\*

\* Retrieve data from the SecureObject

\* SecureObject(accessKey, identifier)

\* @accessKey - the key used to instantiate this SecureObject

\* @identifier - a String used to map to the data you want returned

\*

\* returns the data originally stored in the SecureObject using the identifer or null if not found

\*\*/

secureObject("secretKey", "foo"); // Returns {a: 1, b: 2}

secureObject("secretKey", "bar"); // Returns "myValue"

/\*\*

\* Providing an invalid accessKey or providing an unknown identifier when getting values always returns null

\*\*/

secureObject("wrongKey", "foo"); // Returns null

secureObject("secretKey", "wrongIdentifier"); // Returns null

Notes:

* It should \*not\* be possible to inspect the lockbox object and sniff out the contents. i.e. inspecting in Firebug or iterating over properties with a for/in loop would yield no valuable information whatsoever.
* It should \*not\* be possible to peek into the lockbox to obtain the accessKey used to instantiate the SecureObject
* Don’t bother encrypting/decrypting the stored data – just store data in such a way that you can only gain read/write privileges by supplying the accessKey