LexEVS 6.0 Design Document - Detailed Design - Password Encryption

Contents of this Page

Document Information

Author: Craig Stancl
Email: Stancl.craig@mayo.edu
Team: LexEVS
Contract: CBITT BOA Subcontract# 29XS223
Client: NCI CBIIT
National Institutes of Heath
US Department of Health and Human Services

Revision History

Version	Date	Description of Changes	Author
1.0	5/14/10	Initial Version Approved via Design Review	Team

Password Encryption

Encryption is the process of taking data (called *cleartext*) and a short string (a *key*), and producing data (*ciphertext*) meaningless to a third-party who does not know the key. Decryption is the inverse process: that of taking ciphertext and a short key string, and producing cleartext.

LexGrid utilizes the java security API for encryption and decryption of the database passwords. The Security API is a core API of the Java programming language, built around the java.security package (and its subpackages). This API is designed to allow developers to incorporate both low-level and high-level security functionality into their programs.

The Java Cryptography Architecture encompasses the parts of the Java 2 SDK Security API related to cryptography, as well as a set of conventions and specifications provided in this document. It includes a "provider"architecture that allows for multiple and interoperable cryptography implementations.

Encryption/Decryption implementation Details

Creating a Cipher Object

Creating a Cipher Object

```
Cipher cipher = Cipher._getInstance_("PBEWithMD5AndDES");
```

"PBEWithMD5AndDES" is the widely used algorithm used for the encryption process. Other available algorithms are "PBEWithHmacSHA1AndDESede", "AES", "Blowfish", "DES", "DESede" etc.

Initializing a Cipher Object

Initializing a Cipher Object

```
cipher.init(<MODE>, _<KEY>_, <_PBEParameterSpec_>);
```

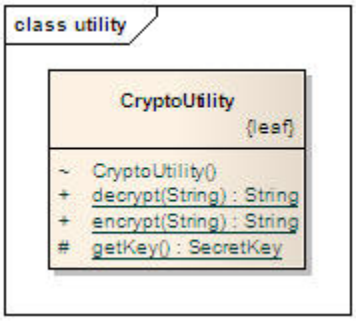
A Cipher object obtained via getInstance must be initialized for one of four modes, which are defined as final integer constants in the Cipher class. The modes can be referenced by their symbolic names, which are shown below along with a description of the purpose of each mode:

- ENCRYPT_MODE: Encryption of data.
- DECRYPT_MODE: Decryption of data.
- WRAP_MODE: Wrapping a Key into bytes so that the key can be securely transported.
- UNWRAP_MODE: Unwrapping of a previously wrapped key into a java.security.Key object.

Encrypting and Decrypting Data

```
cipherBytes = cipher.doFinal(<text to encrypt/decrypt>);
```

Passwords in LexEVS are encrypted /decrypted in one step (*single-part operation*) by passing the text to encrypt/decrypt as a parameter.



Following are the steps to encrypt the password of LexEVS database.

1. Run PasswordEncryptor.sh or PasswordEncryptor.bat (pass password text as a parameter) from lbAdmin to generate the encrypted password.
    - Generated password will be stored in a file @ lbAdmin/password.txt
2. Copy the encrypted password from password.txt and paste it in lbConfig.props file (DB_PASSWORD=<Encrypted_Password>)
3. Set the new lbConfig parameter DB_PASSWORD_ENCRYPTED=true (value case insensitive) .
    - Note : any value other than 'true' (or no value) for DB_PASSWORD_ENCRYPTED is considered as 'false'.
4. When password encryption is off, use the password directly as you have been using till now.