Data Security Class

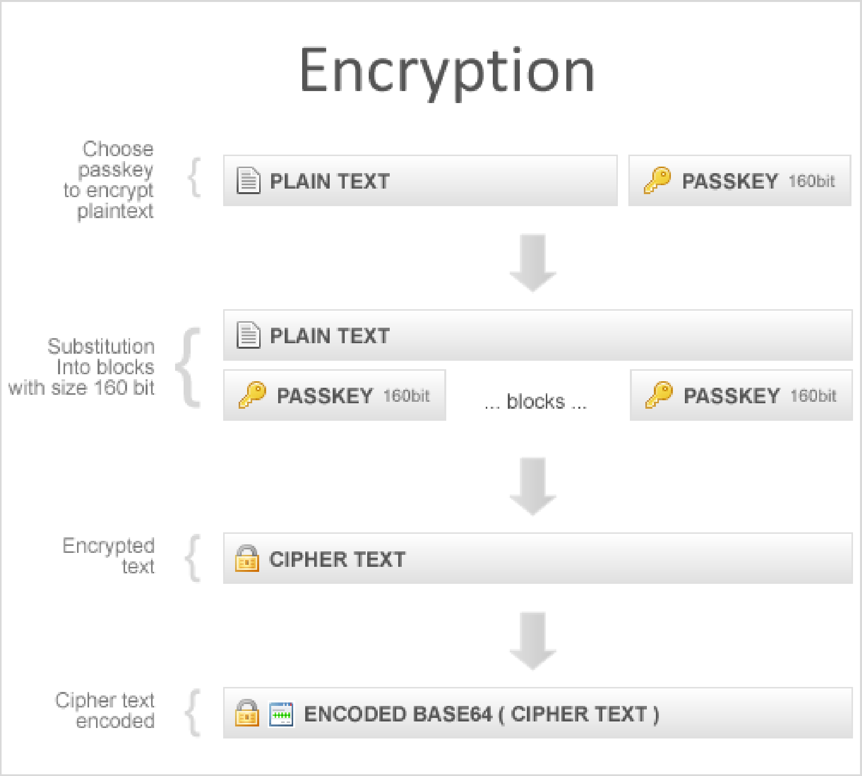
Users Manual

About

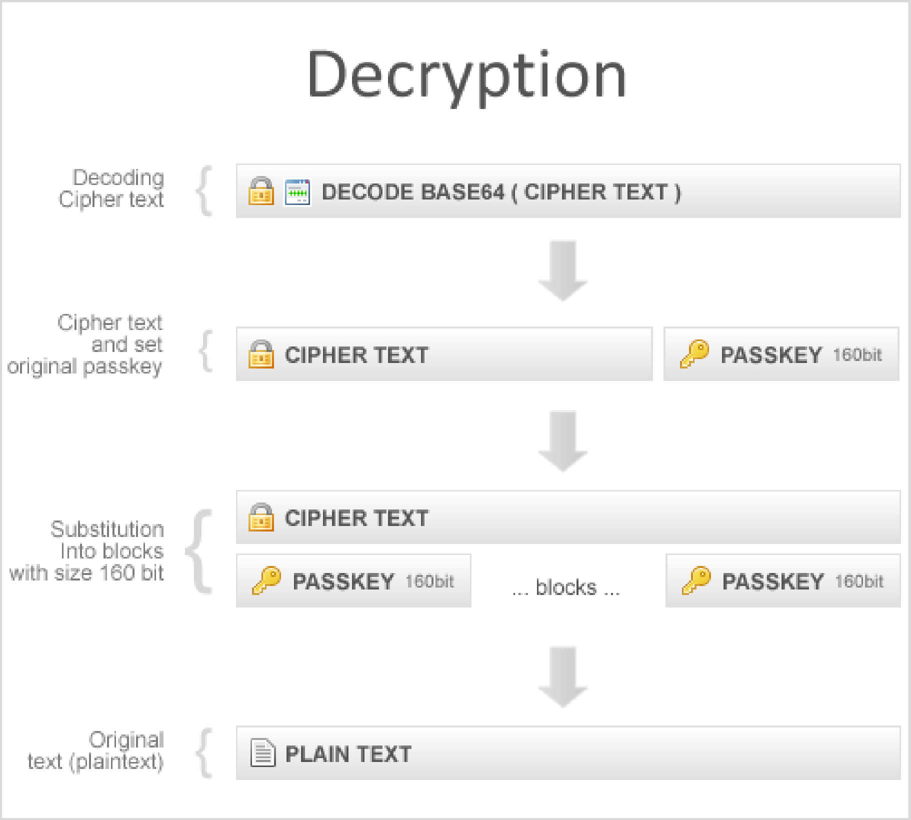
Data Security is encryption class for transforming plain text into cipher text. This ensures data confidentiality and uses 160 bit key to encrypt blocks of plain text. The strength of key it's very high and to break the key it is needed 2^160 which is number with length of 17 numbers and today's computers aren't strong enough to break this key for a short time. Encryption algorithm is designed and discovered by Arlind Nushi, author of this script and named ANCrypt.

Here is the image of how this algorithm ensures data confidentiality:

Encryption phase:



Decryption phase:


Decryption
Decoding Cipher text
DECODE BASE64 (CIPHER TEXT)
Cipher text and set original passkey
CIPHER TEXT
PASSKEY 160bit
Substitution Into blocks with size 160 bit
CIPHER TEXT
PASSKEY 160bit
... blocks ...
PASSKEY 160bit
Original text (plaintext)
PLAIN TEXT

This is how encryption phase works

- First, you need to add text for encryption and choose a “passkey” for that text
    - One passkey encrypts and decrypts the same plaint text and cipher text
    - If you encrypt with a passkey and decrypts with different passkey it will generate complete different text (mostly unreadable)
- Step 2, is the substitution of plain text with 160bit key. Example if text length is 480 chars, then there are 480/40 it will result in 12 blocks, and they will be encrypted with the same key.
    - The keys are generated by using US Secure Hash Algorithm 1 (SHA1)
    - The length of generated key is 40 chars
    - Example key: = sha1(“my_key”)
- Then it’s generated the cipher text that can be only decrypted by using the same key as used on encrypt phase.
- Cipher text is encoded using base64 to generate data that can survive in transport because after encryption cipher text contains binary data.

It is clear that the Decryption phase is the reverse of encryption.

SafeCookie

SafeCookie is implemented under ANCrypt algorithm and ensures:

- **Data Confidentiality**
- **Data Integrity**

Data Confidentiality – Ensures that data are hidden from everyone expect persons who have passkey and encryption/decryption algorithm.

Data Integrity – Guarantees that data saved on cookies cannot be altered or modified from another user. If data's are modified you are in knowledge of that by using a method for testing cookie integrity. Data integrity in cookies is very important because we know that every user has possibility to alter cookies on his own browser and those are read by your site. That's say for example if you have a cookie to check if user is logged and on cookie is saved user access privileges in this form: *user_logged=false*, *user_privileges=normal_user*. Any person can make modifications on these cookies because they are easy to understand and they are meaning of a function on website. Then user try and can change *user_logged=true*, *user_privileges=admin*, and this is the reason why un-protected cookies are weakness of your site.

But with SafeCookie, a cookie is saved on this way:

- Hash value of cookie name is generated using MD5 algorithm and saved with that name
- Value of cookie
    - The hash of value is generated using MD5 – for data integrity check
    - Then hash value and data's for that cookie are concatenated
    - Concatenated data are encrypted with ANCrypt using specific passkey

And to test a cookie data integrity this is the flow of how any cookie can be verified if is altered (modified) and it's not the cookie you saved before.

- First to retrieve a cookie you need to specify a cookie name, and that name will be hashed to search if that cookie exists.
- If cookie exists, it decrypts the content of that cookie using the same key as on encryption
- Cookie is splitted into two parts

  - Part that contains saved hash value
  - And contents of cookie
- A hash value is generated for content of cookie and will be compared with saved hash value of cookie
- If values are the same, this means that cookie is not altered otherwise it will return false value by meaning that cookie content has been altered and it's in your hand to deal with that cookie.

Installing and Configuring

Installation

Extract the content of directory src/ to your project folder that will be used.

Configuration

You don’t need special configuration of Data Security Class because it doesn’t requires, the only thing you need its to include the specified class context that is available in Normal and Static. They are both explained in details in class files why you would use ANCrypt (or SafeCookie) in static or normal context.

Examples of Declaration

ANCrypt in normal context

```
Include(“src/ancrypt/ANCrypt.class.php”);

$instance = new ANCrypt(“pass_key”);
$encrypted = $instance->encrypt(“plaintext”);
$instance->decrypt( $encrypted );

$instance->encryptFile(“file_to_encrypt.txt”, “encrypted_file_name.txt”);
$instance->decryptFile(“encrypted_file_name.txt”, “decrypted_file.txt”);
```

ANCrypt in static context

```
Include(“src/ancrypt_static/ANCrypt.static.class.php”);

ANCrypt::setKey(“super_key”);

$encrypted = ANCrypt::encrypt(“plain_text”);
ANCrypt::decrypt($encrypted);

ANCrypt::encryptFile(“file_to_encrypt.txt”, “encrypted_file_name.txt”);
ANCrupt::decryptFile(“encrypted_file_name.txt”, “decrypted_file.txt”);
```

SafeCookie in normal context

```
Include(“src/safecookie/SafeCookie.class.php”);

$instance = new SafeCookie(“pass_key”);
$instance->setCookie(“my_cookie”, “my value 123”, 3600, “/”); // name, value, seconds alive, cookie path
$instance->getCookie(“my_cookie”);
$instance->validateCookie(“my_cookie”); // true if cookie integrity is not touched
$instance->getOriginal(“my_cookie”); // Get original name and value of cookie
```

SafeCookie in static context

Include(“src/safecookie/SafeCookie.class.php”);

SafeCooikie::setSuperKey(“my_super_key”);

SafeCookie::set(array(“my_cookie”, “my value 123”, 3600, “/”)); // array(name, value, seconds alive, cookie path) two last parameters are optional

SafeCookie::get(“my_cookie”);

SafeCookie::validate(“my_cookie”); // true if cookie integrity is not touched

SafeCookie::set(array(“other”, “otherone”), “my other key”);

SafeCookie::get(“other”, “my other key”);

For more examples please check examples directory

examples/

Also there are simulations of how SafeCookie and ANCrypt can be implemented on site for use, both located on

examples/ancrypt/ANCryptor – Encryption/ Decryption Script

examples/safecookie/safecookie_simulator/

Credits

Author:

Arlind Nushi

Author's email:

arlindd@gmail.com

Demo web-site:

http://arlindnushi.dervina.com/data_security_class/

Icons used on some of examples are taken from FAMFAMFAM and IconsFinder.net

This class is bought from Envato® Marketplace