User Encryption

User Encryption is available in Tiki 12.2 and up.

User Encryption uses CryptLib.
CryptLib (aims to) safely store personal user data, e.g. passwords for external systems, in Tiki.

- The encrypted user data can only be decrypted by the user
- The secret decryption key not stored by Tiki, and is only known when the user is logged in.
- The solution is local to Tiki. No internet/external connection is needed to decrypt data.

User Encryption provides good data privacy, but there are several cases where the data can be lost. See the issues below. It is best suited to store passwords and other data, which can be re-entered by the user.

The targeted use case is being able to securely store data for (Tiki) user A, having an account on system X with username Y and password Z. CryptLib identifies system X as the domain, having the credentials Y/Z for user A. Multiple "password domains" can be defined. One for each system that require user credentials.

System services can access the stored, encrypted passwords, while the user is logged in, without having to prompt the user for additional information or store the password in (near) cleartext.

The module "Domain Password" provides an interface where the user can enter the credentials. See http://doc.tiki.org/User Encryption.

About CryptLib

File: lib/crypt/cryptlib.php

Tiki pre-18

CryptLib uses Mcrypt if the PHP extension is available. The Rijandel-256 bit algorithm is used.
If Mcrypt is not available Tiki reverts to (near-plaintext) Base64 encoding.

In order to use Mcrypt

1. The Mcrypt PHP extension must be available
2. Call the init method before using cryptlib

Tiki18+

CryptLib uses OpenSSL if the PHP extension is available. The AES-256-ctr algorithm is used.
If OpenSSL is not available Tiki reverts to (near-plaintext) Base64 encoding.
In order to use OpenSSL

1. The OpenSSLPHP extension must be available
2. Call the init method before using cryptlib

The functionality

The method setUserData encrypts the value and stores a user preference.
getUserData reads it back into cleartext

The secret key phrase is the MD5 sum of the username + Tiki password.
When a user logs in, Tiki calls onUserLogin, which registers the current secret key in a session variable. This session variable is used to decrypt the stored user passwords when needed. The session variable is destroyed when the user logs out. The decryption key is thus only known to the system while the user is logged in.

Each encryption uses its own initialization vector (seed).
Rehashing the same value should thus yield a different result every time.

When a user changes the password, Tiki will call onchangeUserData, which will rehash the value.

The system needs to have the username + both the old and new passwords in cleartext, in order to be able to rehash the encrypted data. This may not always be possible.

The cleartext data can be recovered knowing the username and Tiki password, which was used to encrypt the data.

Issues

There are some cases which can/will render the encrypted data unreadable, thus lost.

If the user forgets the password, the encrypted data is no longer recoverable.

Direct database updates

Changing a user's Tiki password directly in the database will not fire onchangeUserData, making the stored passwords unreadable.

Tiki cannot maintain the data relations if data are updated directly in the database. This is method of updating is discouraged. Can be destructive when combined with the "User Encryption" feature.

Admin sets user's password

When an admin "hard" sets a user password, the old password is unknown. This makes it impossible to rehash the stored password. The encrypted data can then no longer be decrypted when the user logs in, since the "secret key" has changed.

A warning has been added to the admin panel, where user passwords can be reset.
"The feature User Encryption stores encrypted user information, such as password used to connect to external systems. If the password is changed, it will destroy the user's decryption key, and make the data unreadable. The user will be forced to re-enter the passwords and other data that may be encrypted."

Encryption scheme changes

You should make sure the OpenSSL PHP extension is installed before enabling User Encryption. CryptLib will fallback to Base64 encoding if OpenSSL is not found.

Changing the encryption scheme, e.g. by first using User Encryption then installing OpenSSL, will invalidate all passwords, unless a conversion is done.

It is best to make sure that OpenSSLis available before enabling the "User Encryption" feature. Thus avoiding having to switch encryption schemes.

Only the user can enter the data

It is not possible to have another user, even if this is the admin, to enter encrypted user data instead of the user him-/herself. The encryption is locked to the logged in user's username + Tiki password.

This is not so much a problem issue, but a consequence of the User Encryption.

Using CryptLib

CryptLib integrates at code level with the external service interface. Use the module "Domain Password" to provide a UI, so users can manage the credentials.

Before using CryptLib, the "User Encryption" feature must be enabled (Admin - Security)

Getting an encrypted password

```
// Retrieve username and password
$cryptlib = TikiLib::lib('crypt');
try {
        $cryptlib->init();

        // Get the username and password in the password domain
        $domain = 'Database X';
        $password = $cryptlib->getUserData($domain);
        $username = $cryptlib->getUserData($domain, 'usr');
        // $username = $user;  // ...or use the logged in user
        if ($password == false || $username == false) {
                $error = 'Please enter you account information';
        }

} catch (Exception $e) {
        $error = $e->getMessage();
}
$cryptlib->release();
$cryptlib = null;
```

In the stored user preferences the userPrefKey is altered.
CryptLib will add a prefix: dp.
So, "Database X" becomes "dp.Database X".

If a username is stored, a parameter is appended to the user-pref key. For paramName = 'usr', the result is: "dp.Database X.usr"

The encrypted data is stored in user preferences and key field length is 40 characters. That's the length available to specify a password domain name with pre- and suffix.

Managing a different user

```
// Assign an encrypted user and password to a new user
// Note: Can only be done, when the password is known in clear-text

// Get a new cryptLib instance and initialize the encryption
$userCrypt = new CryptLib();
$seed = $userCrypt->makeCryptPhrase($newUser, $newPass);
$userCrypt->initSeed($seed);

// Save the credentials for the new user
$domain = 'Database X';
$userCrypt->putUserData($newUser, $domain, $newPass);
$userCrypt->putUserData($newUser, $domain, $newUser, 'usr');
```

Questions

- How can passwords be shared? Ex: 3-5 team members that all need access to a series of credentials related to each project. Ex.: Domain name registration / Management, uptime monitoring, Server passwords, etc. (The list is different for each project)
    - Team members can be added and removed at any time, so their password won't be immediately available to use for encryption.
    - Credentials can be added / removed at any time for a project. Maybe projects could be Tiki categories so we can use for listing / filtering / permissions, etc.
    - A new Tiki user could be created for each project, and each person who should have access would store these credentials in their own User Encryption space (A recursive use of the tool!)
        - Maybe this idea could be combined with the switch user feature we already have?
- Why does the list of domain passwords need to be set by the site admin? Why can't each user create as needed?
- Any thoughts on browser integration?
    - "Send these credentials to my User Encryption"
    - Bookmarklet like Clipperz
    - 1-click login
        - https://clipperz.is/features/direct_logins/
- Suggestion: Add a text area field for user notes related to this password
- Any thoughts on offline access? Ex: Save all to one encryped HTML 5 file.
- Idea: could this concept be applied to tracker items?
    - Option on fields (So we can text area, file, etc.) or a dedicated password / secret / encrypted field type
    - Easy to list / filter / sort / etc.
    - Idea: share by email (Unique URL using token feature). Default to only work on 1st click

I am adding the answers here for now

1. "How can passwords be shared?".

In order to access the information, the user's clear-text password must be known. Additionally the username

must be known. If a user need to setup access to a number of projects, add a "Domain Password" module for each project. Each user can then fill in the username and password for each of the relevant projects. If the credential changes for any project, the user must update the Domain Password settings for that project. If a new Tiki user is added, the user must manually set the domain password. The admin user defining the user could also do this, since the clear-text password is known when the user is defined. Such an operation probably requires a custom module or similar, It is probably not possible to combine this with the user switch, since the (switched-to) user's clear-text password must be know to decrypt the info. A browser integration would require knowledge of the username and clear-text password. Probably not suited.
So, how can passwords be shared? By communicating them verbally or written to each user. Each user can then individually enter the login info for the external system...if that is the question.

1. "Send these credentials to my User Encryption"

...not sure what is meant here, Is the clear-text password know at the time? If not decryption is not possible.

1. Suggestion: Add a text area field for user notes related to this password.

This is possible. Should this be general notes, or? Should the info be encrypted?

1. "Any thoughts on offline access?"

No such plans at the moment. However, I have considered "anonymous" access. In some cases it is wanted that anonymous users can access the "resource". Since the user encryption is linked to a Tiki user, this is currently not possible. It could be possible if the "anonymous user-data" is stored in local.php (or similar).

1. Idea: could this concept be applied to tracker items?

I don't see why not, but remember that all data is lost if the decryption is disturbed, e.g. by the admin resetting the users password.Using user encryption to store private info in tracker should be possible.

1. Other questions

- Option on fields (So we can text area, file, etc.) or a dedicated password / secret / encrypted field type
- Easy to list / filter / sort / etc.
- Idea: share by email (Unique URL using token feature). Default to only work on 1st click

I am not sure about these questions.Tokens do not contain the decryption info, so probably not usable.

Brainstorming KeePass interoperability

- Perhaps a driver / adapter could be created for read (or even read/write) access to KeePass databases?
    - KeePass is being considered to be a component of Tiki Suite
    - KeePass has tons of features, integrates well with desktops apps and browsers. Mobile clients are available. Can encrypt files as well.
    - Please see: KeePass

Related

- http://www.open-emr.org/wiki/index.php/Encryption_and_Decryption_of_Documents
- http://doc.owncloud.org/server/7.0/user_manual/files/encryption.html
- https://framework.zend.com/blog/2016-08-19-end-to-end-encryption.html

Alias

- CryptLib