Ecryption / Decryption for external disk

Function Description

The code snippet defines a function named `encrypt_decrypt_external_disks` with the following signature:

```
pub fn encrypt_decrypt_external_disks(private_public_key: String, user_id: String, is_encryption: u8)
```

Parameters

- `private_public_key` (String): A string representing the private/public key used for encryption/decryption.
- `user_id` (String): A string representing the user ID.
- `is_encryption` (u8): An unsigned 8-bit integer representing the operation mode. It determines whether encryption or decryption should be performed.

Function Logic

The `encrypt_decrypt_external_disks` function performs encryption or decryption on external disks using the provided private/public key. It iterates over the available disks, excluding the "C:\" disk, and calls the `multi_threaded_encrypt_decrypt_files` function to perform encryption or decryption on the files within each disk.

The function uses the `get_disks` function from the `crate::system::info` module to obtain a list of available disks. For each disk (excluding the system disk "C:\"), it calls the `multi_threaded_encrypt_decrypt_files` function, passing the disk path, private/public key, user ID, and operation mode as arguments.

Example Usage

Here is an example of how you can use the `encrypt_decrypt_external_disks` function:

```
fn main() {
    let private_public_key = "your_private_public_key".to_string();
    let user_id = "your_user_id".to_string();
    let is_encryption = 1; // 1 for encryption, 0 for decryption

    encrypt_decrypt_external_disks(private_public_key, user_id, is_encryption);
}
```

In the example above, the function is called with the appropriate arguments to perform encryption on external disks using the provided private/public key and user ID.

Make sure to replace `"your_private_public_key"` and `"your_user_id"` with the actual values you want to use.

Revision #2
Created 3 July 2023 10:14:25 by Makito
Updated 3 July 2023 10:25:39 by Makito