keep-it-secure whitepaper

Definitions

F : Plaintext file content

t_F : Plaintext name of the file

k : Symmetric key

f : Ciphertext file content

Structs

Record: (a, b) where a is f and b is $chk(t_F)$

Functions

e, encryption method

$e(F, t_F, k) \rightarrow Record(f, chk_{t_f})$

d, decryption method

$r \in$ record_book; $d(f, r, k) \rightarrow (F, t_F)$

Runtime

pack.py

1. A plaintext file is given
    1. F, file contents and t_F file name is extracted
2. $e(F, t_F, k)$ is called
    1. $f = \text{gpg\_encrypt}(F, k)$
    2. $r = \{chk(t_F) \rightarrow (t_F, chk(f), chk(F))\}$
    3. r is appended to `record_book`

3. Encrypted file is written as $(f, chk(t_F))$ to disk
4. Hash block $h = (chk(F), chk(f))$ with name $chk(t_F)$ is created
5. Hash block is committed to vault with commit message $chk(t_F)$

unpack.py

1. A ciphertext is given
    1. f, file contents and $chk(t_F)$ file name is extracted
2. $d(f, r, k)$ is called
    1. An entry for $chk(t_F)$ in `record_book` is queried
        1. If no record is found, an error is shown and process exits
        2. Returned values $t_F, h_F = chk(F), h_f = chk(f)$ are stored
    2. Ciphertext′s current checksum $chk(f_{rn})$ is compared against stored record from `record_book` h_f
        1. If values do not match, an error is shown and process exits
    3. $F_{rn} = \mathrm{gpg\_decrypt}(f, k)$
    4. Plaintext′s current checksum $chk(F_{rn})$ is compared against stored record from `record_book` h_F
        1. If values do not match, an error is shown and process exits
3. Decrypted file is written as (F, t_F)

Document

Revision 1; 15.12.2023

Ferit Yiğit BALABAN, fyb@fybx.dev

https://fybx.dev

2023