DESCRIPTION OF THE CRYPTOGRAPHIC SYSTEM

VERSION 2.7.0
MAY 2021

SUPPORT@USECRYPT.COM

Abstract

This paper covers the theoretical background to the UseCrypt Safe system. It includes the course of the most important cryptographic processes related to specific user actions and an assessment of the security level provided by the application.

Keywords: UseCrypt Safe, cryptographic system, encrypted cloud, encryption and data sharing.

Table of contents

1. General description of the system

The UseCrypt Safe system enables convenient and secure storage and sharing of files along with processing related data. The security of cryptographic systems results from the quality of applied algorithms and the way they are combined in operations such as encrypting, signing or key management. The UseCrypt Safe system uses such algorithms as RSA, AES or DH, which are proven and commonly used all over the world. The high quality of their compositions results from the experience and cooperation of leading Polish cryptologists, which is confirmed by the positive rating given by the Institute of Cyber Security of the Military University of Technology in Warsaw.

Selected features of the system:

- Files are encrypted as part of the end-to-end process - this means that only the owner and persons authorised by the owner have access to the content of the files; the data cannot be decrypted on the server.
- Communication is carried out through the UseCrypt Secure Tunnel (UST), which makes a dedicated encrypted connection between the client application and the server. Its use ensures that the communication is not eavesdropped on or manipulated.
- A distinctive feature of the UseCrypt Safe system is the use of the Hybrid Virtual Key Management (HVKM) technology, which is unique in a global scale. HVKM significantly increases the security level of the user's private key by dividing it into two parts. As a result, the user's data remains secure even if their computer is physically stolen.

A detailed description of cryptographic operations employed by UseCrypt Safe can be found in the following chapters.

2. Basic tools and algorithms

The basic cryptographic library used by UseCrypt Safe application is OpenSSL in its latest available version (currently 1.0.2p).

For symmetric encryption CBC mode AES-256 algorithm is used . This is currently the strongest publicly available symmetric encryption algorithm; its advantage comes from the fact that this algorithm is supported by modern processors, which make it much faster than other algorithms.

The basis of asymmetric cryptography is RSA 2048-bit. The choice of key length has a direct impact on security - the longer the key, the greater its strength. A disadvantage related to application of long keys is the increased time needed to make cryptographic operations. It has been recognised that the optimum length is 2048 bits (it is estimated that a key of 2048-bit length will remain resistant to attacks for 10 more years).

For the exchange of keys between the client and the server, the Diffie-Hellman protocol is used. SHA-512 is used as secure hash function.

3. UseCrypt Secure Tunnel (UST)

Once the connection between the client and the server is established, an encrypted UseCrypt Secure Tunnel (UST) is created. The communication is encrypted using AES algorithm.

UST initialisation protocol starts with verification of the server's identity based on its certificate (which is issued by UseCrypt CA). A connection to the server can only be established if the server domain is confirmed by a valid certificate.

The key exchange is carried out using the Diffie-Hellman (DH) protocol. For the initial tunnel, the key is a random value generated for the client. When a user logs into the system, a new proper session key is concurred, which is generated on the server.

UST has an advantage over VPN tunnels as it operates at the application layer and prevents eavesdropping or impersonation by other applications running on the user's computer. Unlike SSL/ TLS, the session key (which is agreed upon in the authentication protocol) is generated not only by the client side, but also by the server - a safeguard in case a client's randomness generator is of low quality (this could make it easy for attackers to steal the password).

4. Generation and split of the user key

As part of the client's registration process , the user's RSAkeys - both public and private - are generated. To strengthenthe qualityof the keys , the randomness generator is enriched with a value created by random movements of the mouse cursormade by the user. The innovation of HVKM technologyin relation to the standard RSA algorithm isthat the private key (d ,e,n) is divided into two complementary parts (du,e,n) and (ds,e,n), which must conform to the following relation:

$$d_u d_s \equiv d \varphi \bmod(n)$$

This means that both parts retain the same strength as the original key (here: 2048 bits), while performing a private key operation requires sequential use of the first one and then the other part of the key. Its part (ds,e,n) is sent to the server and then deleted by the client during the account creation. All private key operations require cooperation of both parties.

5.User configuration

For the security of the client's private key, it is stored as encrypted form in the so- called User Configuration, which is created when the user account is opened. The key to the Configuration can be reconstructed by combining the hash of the user's password , the random factor R (located in the User Configuration) and the secret S , which is located on the server; the user gets access to it upon positive authentication in the system.

5.1.Safe configuration

During registration, a so-called safe configuration is also created. This contains part of the client's private key, but it is encrypted with a separate password. The purpose of the safe configuration is to be able to regain access to the account in case the user loses his user configuration (in case of computer failure or theft) or if he forgets his password.
The procedure for regaining access to an account using safe configuration involves setting a new user password and cancelling the authorization of all existing Trusted Installations. The user is only allowed to save the safe configuration file and its password at the account creation stage, as only at this point is their identity unconditionally trusted. For security reasons, the file should be stored separately from the password, and both should be protected from unauthorised access by third parties and from loss. (If the safe configuration file and password are stolen, an unauthorised person is able to take entire control of the user account. If a file or password is lost (e.g. if the storage medium is lost or fails), it will become impossible to regain access to the account as part of an emergency)

The user may opt out of storing the safe configuration. For corporate environments, it is recommended that the safe configuration and its password be secured by an IT department that is competent to store and use them securely.

6. User authentication

Logging into the system is done with the dedicated UseCrypt Safe client by means of a login (i.e. the user's e-mail address) and a password. Correct submission of this information is the only action required from the user at each logging in - however it is not sufficient to gain access to the account.

During user authentication it is necessary for the client to have access to the current User Configuration file.

Not only does the system verify that the password is correct, but also that the device (or more specifically, the Trusted Installation) is authorised by the user. In the case of a suspected compromise, authorisation can be withdrawn for selected devices while the remaining devices remain functional. In the case of more serious incidents, the account access recovery procedure can be used;

- then all existing Trusted Installations automatically lose their authorizations, and the user can set a new password

The above conditions minimise the chance of unauthorised persons taking control over the user's account.

6.1. Role of HVKM

The unique feature of HVKM technology is that it allows the user to regain exclusive access to the account and restore its security even in case of a break-in.
If the classic private key is stolen , there are irreversible consequences of compromised security, as the attacker can use this key to decrypt any cipher text. Introduction of access restrictions to the server does not give full assurance that cipher texts will not leak; an expensive but necessary solution is to generate a new key pair, encrypt all the data and delete the previous cipher texts. Even so, there is still a chance that the attacker copied the cipher texts before they were destroyed.
Splitting the private key into a client part and a server part makes the application server a necessary partner to use the private key ; access privileges are just as important as physical access to the user's part of the private key. In particular , HVKM provides assurance that changing a user's password or revoking the authorisation of compromised Trusted Installations is sufficient to stop attackers from decrypting data. Non-authenticated access to cipher texts is no longer security-critical as taking over part of the user key and the cipher text is still not sufficient to decrypt its contents. In particular, cipher texts can be stored ignoring numerous access control restrictions, increasing convenience while maintaining a high level of security.

7. Account access authorisation

The UseCrypt Safe system requires authentication of the account and trusted installations by rewriting a token (i.e. a random sequence of characters) sent to the user's email address. The user is considered to have exclusive access to their email account and the token sent through this channel can be used as an additional security factor.

7.1. User account

As part of the registration process, the user provides an email address, which becomes their login in the Use- Crypt Safe system. In order to avoid abuse, it is necessary to confirm that the user is the owner of this address - this is why the system requests authorisation via token during the first login. In the event of positive authorisation, the account is activated and it is not possible to register again with the same email address.

7.2. Trusted Installations

Every time the user logs into the system the UseCrypt Safe client recreates the installation ID and sends it to the server. Based on this, the server can determine whether a particular installation has been authorised by the user to use his account. The installation identifier (which determines a particular Trusted Installation) is recreated based on a random ID a value and the device ID u. ID a is generated when the user configuration is created (as a result of creating, importing or recovering an account), so each recreation will result in a new ID a. ID u is calculated each time based on a set of unique parameters of the current device. Thus , moving the configuration to another computer will be recognized by the server as a new Trusted Installation. Any attempt to log in to a user account from an unauthorized Trusted Installation results in a token authentication request. This event is recorded in the system logs, including registration of IP address from which the attempt to access the account was made.
The user can revoke the authorisation of selected Trusted Installations after logging in to the system or contacting the server administrator. In exceptional cases, the Safe Configuration can also be used, which results in withdrawal of the authorisation of all current Trusted Installations.

8. Transferring configuration to another device

The system allows a user account to be used from multiple devices; this requires the user configuration to be transferred and the new Trusted Installation to be authorised.

8.1. Import from file

Once logged in, the user has the option to export their configuration to a file to import it to a new device. The file must be deleted immediately after use; therefore it is not recommended to send it via email or other media over which the user does not have full control.

8.2. Remote transfer

The UseCrypt Safe system also allows remote transfer of the Configuration via the server. The remote transfer protocol provides that the user logs into the system on S (old) installation and then from N (new) installation sends a request to download the user's configuration - the application displays the DH public key hash to the user . S installation notifies you that another installation is requesting the user configuration. To ensure that the Configuration is sent to a known recipient, the notification shows the previously mentioned DH public key hash - if the digest is different from the one displayed on N installation, the user should absolutely reject the remote configuration transfer request. After acceptance, the actual transfer takes place and the cipher text is deleted from the server.

8.3. Use of safe configuration

A special case of importing a User Configuration is the use of a safe configuration. However, its use involves revoking the authorization of all other Trusted Installations. For this reason it is recommended to choose the other options described above.

9. Password change

The user password has a double meaning in the system as it is used to verify identity by the server and it is necessary to recreate the key of the User Configuration. Therefore, changing the user password results in changing the key of this Configuration and its encryption on the installation from which the procedure was initiated. The password hash stored on the server will also be changed.

If the user wants to log in on another installation, he should enter the new password in order to be verified by the server. The system will detect that the User Configuration cannot be de- encrypted with the new password provided; it will then ask the user to enter the old password and if successful will encrypt the Configuration with the new key. From now on, the user only needs to enter the new password when logging in.

To change the password, the system requires the user to enter the previous password once and the new password twice. This increases the level of confidence that the password is changed by the account owner and that the new password is set consciously (i.e., no accidentally pressed characters appear in it).

9.1. Emergency password change

By default, only the logged-in user can change the password as the system administrator cannot set a new password or reset it. If the password is forgotten, the account recovery procedure should be carried out using the Safe Configuration and a new password should be set. Note, however, that all previous Trusted Installations will be de-authorised.

10. File key management

The content of the file and its metadata (including its name) are encrypted. Encryption of metadata protects against concentrated attacks as it makes it difficult to evaluate the contents of the cipher text.

10.1. Encryption

The K key- randomly generated for each file and sent to the server as a so - called capsule - - is encrypted with the user's public key. Data encryption is carried out using AES algorithm, and the use of the key encapsulation mechanism (KEM) eliminates possible weaknesses of the asymmetric algorithm that appear during encryption of minor data. Additionally, the system enables effective destruction of a local copy of the original file after sending it to the server (if the user wishes so). Simply deleting a file is not enough to prevent its restoration; file systems do not overwrite its contents when deleting a file. In many cases this means that the secret content can be recovered directly from the disk and thus circumvent the protection. The destruction procedure involves overwriting a file several times with random data and only then deleting the file; this way the user can be sure that the file sent to the UseCrypt Safe server is secure.

10.2. Access to file

Access to the content or metadata of a file requires its decryption with K key. In order to obtain the open form of K key it is necessary to decrypt the capsule sequentially using both parts of the private key - first the server does it (using part of the server's private key) and then the client (using part of the client's private key). After finishing the decryption of the file data, the client removes the key overwriting it in memory with random data.

10.3. File sharing

In order to share the file, it is not necessary to decrypt the cipher text itself, it is only necessary to decrypt the key capsule. For this purpose, the server sends a partially decrypted capsule to the client along with the addressee's public key P.
The client completes the decryption of key K, then encrypts it with the public key P and sends it to the server with the information which permissions it assigns to the addressee. From now on the addressee has access to the content and metadata of the file and depending on the decision of the assigner the level of rights may include reading, overwriting or deleting the file.

Using a separate K key for each file means that the owner has very precise control over anyone who has access to it.

10.4. Local encryption

The system offers possibility of local data encryption. Encryption, decryption and file sharing are the same as above with the difference that the cipher text of the file content is not sent to the server. The purpose of this functionality is to allow sharing of large files without transferring to/from the server any time the Internet connection transfer is too slow and the user prefers to transfer the cipher text e.g. on an external drive.
It may seem that local encryption has a different security level than sharing files via the server, but this is not true. UseCrypt Safe is still used to manage the file keys, and cipher texts are still not useful without the cooperation of the client and the server; no matter where they are located.

11. Bibliography

1. M.Kutylowski, P.Kubiak, M.Tabor, D.Wachnik. Mediated RSA cryptography specification for additive private key split ting (mRSAA).
2. PKCS #7: Cryptographic message syntax standard. Technical note, RSA Laboratories, November 1993.
3. PKCS #12: Personal information exchange syntax. Technical note, RSA Laboratories, 1999.
4. PKCS#1v2.1: Rsa cryptography standard. Technical note, RSA Laboratories, 2002.
5. D. Boneh, X. Ding and G. Tsudik. Identity based encryption using mediated RSA. In 3rd Workshop on Information Security Application, August 2002. KIISC.
6. X.Ding, G.Tsudik. Simple Identity-Based Cryptography with Mediated RSA.
7. H.Krawczyk, IBM. M.Bellare, UCSD. R.Canetti, IBM. HMAC: Keyed-Hashing for Message Authentication. February 1997.
8. E.Rescorla, RTFM INC. Diffie - Hellman Key Agreement Method. June 1999
9. S.Blake - Wilson, A.Menezes. Authenticated Diffie-Hellman Key Agreement Protocols. 1999
10. J.Randall, Randal Consulting. B.Kaliski, EMC. J.Brainard, RSA. S.Turner, IECA. Use of the RSA-KEM Key Transport Algorithm in the Cryptographic Message Syntax. September 2010.
11. XML Security Working Group F2F. Key Encapsulation: A New Scheme for Public-Key Encryption. May 2009.
12. Federal Information Processing Standards Publication 197. Specification for the Advanced Encryption Standard (AEsS). November 2001.
13. D.Eastlake, DEC. S.Crocker, Cybercash. J.Schiller, MIT. Randomness Recommendations for Security. December 1994.