

**Project:**

**DlpServer.KeySafe**

**Component: DlpServer.KeyManagement.exe**

|  |  |  |
| --- | --- | --- |
| **Author** | **Date** | **Comments** |
| Emad Mohammad | 20th Feb 2017 | First version |
|  |  |  |

DLPSERVER KEY\_MANAGEMENT(new)

**INTRODUCTION**

This component of keysafe actually generates the encrypted user private key, upon request from client, which sends userkey id over sockets. Then, the private key is decrypted and the required user private key is returned. For safe transfer of key from server to client, AES and RSA are implemented together. First, the required private key is encrypted with AES key and then encrypted private key is sent to the client. After this, RSA public/private keypair are generated in client , then public key is sent to the server. In the server, this public key is used to encrypt the AES key that was used to encrypt the private key. After this, encrypted AES key is sent to the client. At the client side, first encrypted AES key is decrypted with the RSA private key, then this decrypted AES key is used to decrypt the user private key.

1. User requests for the key.
2. “get\_key.php” is called which inturn calls “clientSoc.php” and passes the “keyId” to the later.
3. This userkeyId is sent to the server using sockets.
4. In the server ,the sockets in the service, listens to the data from client.
5. A function named, “Listen\_socketGetPublicKeyToEncrypt()” in the service, is called .
6. Firstly,This function passes socket listener and handler and an identifier value to the function “join\_conn()”.

* In this function ,first sockets listen and connect and the data is then received.
* This data, for first time gets the userkey.
* If the identifier value for “\_conn” is “1”, then this data is passed to the “EncryptData\_aes()” function.

**GENERATING THE USER DECRYPTED PRIVATE KEY (SERVER SIDE)**

* In this function, first the ‘userkey” is passed to the “DecryptPrivateKey()” which generates the user private key and then returns this private key.
* In “DecryptPrivateKey()” , first path is made to the configuration file “” , where we get the encrypted “keytype” and “password” of the CEO, which is then decrypted.
* We establish connection to database and using the “userkey” from client, get the publickey and privatekey and generate the keypair for user named “keyUserPair”.
* Also, using the “keysafe” keytype , we get the publickey and privatekey and generate another keypair called “keySafePair”.
* Now, using the “keytype” from configuration file, which is the keytype of “ceo”, we get its public and private keys, we generate the keypair for ceo named “keyCeoPair”.(private key is encrypted).
* Now, using the password from the configuration file, we decrypt and return the privatekey of ceo, using the function, DecryptKeyWithPassword(keyCeoPair,myPassword);
* Using this private key of ceo, we decrypt the “keySafePair” using the function DecryptKeySafe(keySafePair, privatekey), which return decrypted keysafePair private key.
* Finally, using the decrypted keysafePair private key, we decrypt private key of “keyUserPair”, using function DecryptWithPrivate(keyUserPair.PrivateKeyByte, dd) and the decrypted private key of the user is returned.

**ENCRYPT THE USER DECRYPTED PRIVATE KEY WITH AES, STORE AES KEY AND SEND BACK TO CLIENT(SERVER SIDE)**

* In the function, “EncryptData\_aes()”,decrypted private key is returned by the function, “DecryptPrivateKey()”
* Using the “Rijndael” method, we create the rijndael encryptor and encrypt the private key.
* Here , we save the “aes\_key” key to a global variable ,for later use and return the encrypted private key.
* This, returned encrypted private key is sent back over sockets to the client in php file.

**GENERATE THE PUBLIC/PRIVATE KEYPAIR USING RSA(IN PHP)(CLIENT SIDE)**

* Using the  **RSA** ,OPENSSL KEY, we generate the public private key of size 2048.
* Each time generation of public key is random
* Then, using sockets, we send the rsa public key over port “90”.
* And private key is stored.

**GET THE PUBLIC KEY , ENCRYPT AES KEY AND SEND BACK TO THE CLIENT(SERVER SIDE)**

* Sockets listens and the data is received again, this time the rsa public key from client.
* If the identifier value for “\_conn” is “2”, then this data , here which is public key and aes key is passed to the “EncryptKey\_rsa(publickey, Aeskey)” function.
* A **BouncyCastle** library has been used to separate the ***modulus*** and ***exponent*** parameters from the public key.
* Then, these parameters are imported to “*RSACryptoServiceProvider*” and then with the keysize of 2048 , AES key is encrypted and returned.
* Then, the encrypted AES is sent back to the clientside in php.

**DECRYPT THE RECEIVED AES KEY FROM SERVER(CLIENT SIDE)**

* The AES key is decrypted using RSA private key, using function,

***openssl\_private\_decrypt($encrptd\_aes\_key,$aes\_key,$privKey);***

* The decrypted RSA key is then, used to decrypt the encrypted user private key on clientside.

**DECRYPT THE RECEIVED *ENCRYPTED USER PRIVATE KEY* WITH *DECRYPTED AES KEY*(CLIENT SIDE)**

* The encrypted user private key is decrypted using ***MCRYPT\_RIJNDAEL\_256***, using function,

***strippadding(mcrypt\_decrypt(MCRYPT\_RIJNDAEL\_256, $key, $text, MCRYPT\_MODE\_CBC, $IV)***

* And the UserPrivatekey is returned.
* And is passed to “get\_key.php” and echoed there.