Smart Secure USB SSU-256

Muhammad Ehsan ul Haq, Zeeshan Ali, Muhammad Taimoor Ali, Ruqiya Fazal, Waseem Iqbal, and Mehreen Afzal

Abstract USBs are the most common devices for data sharing and transferring either for personal day to day use or at the organizational level. Its usage is increasing exponentially despite the data breaches occurring due to the noncompliance of security measurements. Consumers are at risk when sensitive data is stored on unsecured USBs. The consequences of losing drives (or when picked up by unauthorized persons) loaded with sensitive information can be significant, including the loss of customer data, financial information, business plans and other confidential/sensitive information, risk of reputation damage. Apropos, this problem of keeping the data confidential from unauthorized users need to be addressed immediately. Therefore, in this paper we present an indigenous solution for this problem which can easily be used by general users and sensitive organizations (strategic, banks, academia, law enforcement, armed forces, telco's and many others) to overcome the above stated confidentiality problem. Our proposed Smart Secure USB (SSU-256), will serve as secure channel for both data storage and transfer.

Keywords USB · Authentication · SSU · Cyber attack · Encryption

M. Ehsan ul Haq · Z. Ali · M. T. Ali · R. Fazal · W. Iqbal (✉) · M. Afzal
Department of Information Security, National University of Sciences and Technology, NUST, Islamabad 44000, Pakistan
e-mail: waseem.iqbal@mcs.edu.pk

M. Ehsan ul Haq
e-mail: 354721ehsan@gmail.com

Z. Ali
e-mail: zeeshan.ali92171@yahoo.com

M. T. Ali
e-mail: muhammadtaimoor1999@gmail.com

R. Fazal
e-mail: ruqiya830@gmail.com

M. Afzal
e-mail: mehreen.afzal@mcs.edu.pk

H. Jahankhani et al. (eds.), *Cybersecurity, Privacy and Freedom Protection in the Connected World*, Advanced Sciences and Technologies for Security Applications, https://doi.org/10.1007/978-3-030-68534-8_9

1 Introduction

In year 2000 USB memory stick was launched which was light in weight, portable, comparatively cheap and offered high transfer rate. Since then the usage of USBs is increasing exponentially ignoring the fact that they offered no kind of security to user's data (Fig. 1).

As both consumers and businesses have increased the demand for these drives, manufacturers are producing faster devices with greater data storage capacities [2], despite the data breaches occurring due to the noncompliance of security measurements. To avoid these data losses secure USBs must be brought into use (Fig. 2).

Data breaches and cyber-attacks are anticipated to increase in the due course of time as the computer networks are expanding. As technology progresses, more and more of our information is at risk. As a result, cyberattacks have become increasingly common and costly. As per Ponemon report 70% of businesses have traced the loss of sensitive or confidential information to USB memory sticks [3]. It is crucial and mandatory to protect our data to get rid of cyberattacks (Figs. 3 and 4).

The basic idea behind the project being proposed is to design a secure USB which will serve as secure channel for both data storage and transfer. This project is going to be the productive insight in the industry due to its Multi-layer security i.e. Biometric Authentication and Encryption which will work completely independent of each other, beside that it provides User specific Compartments and an Admin for users registration and deletion as well as being Indigenous, Cost Effective, assuring integrity and an easy to use, plug n play portable device.

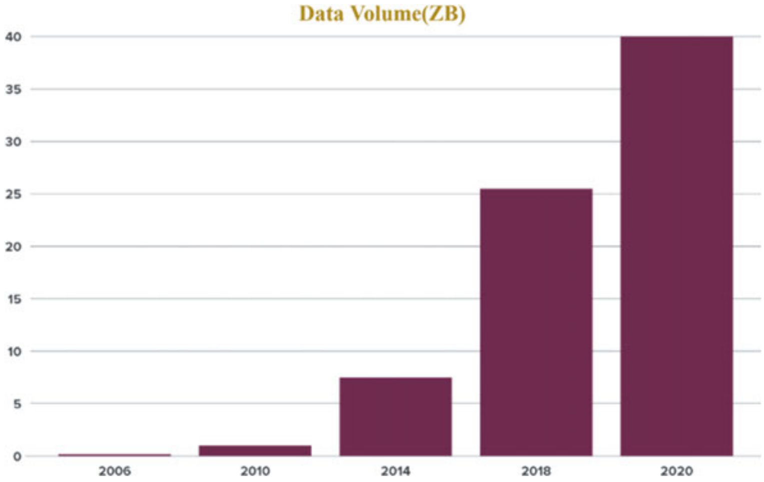


Fig. 1 Global trends of data volume [1]

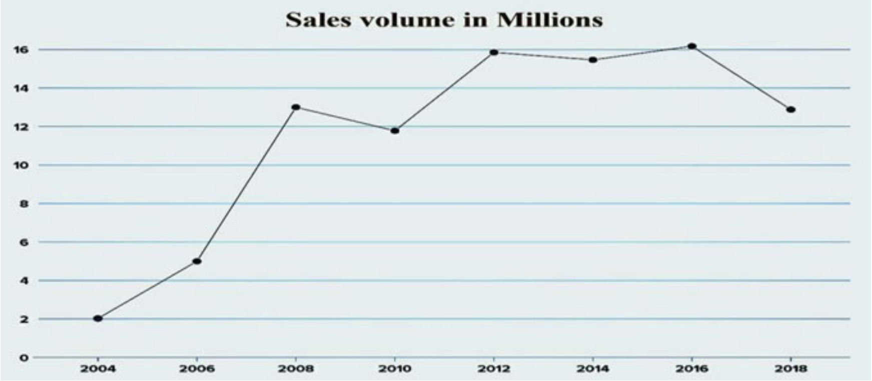


Fig. 2 USBs sales volume [2]

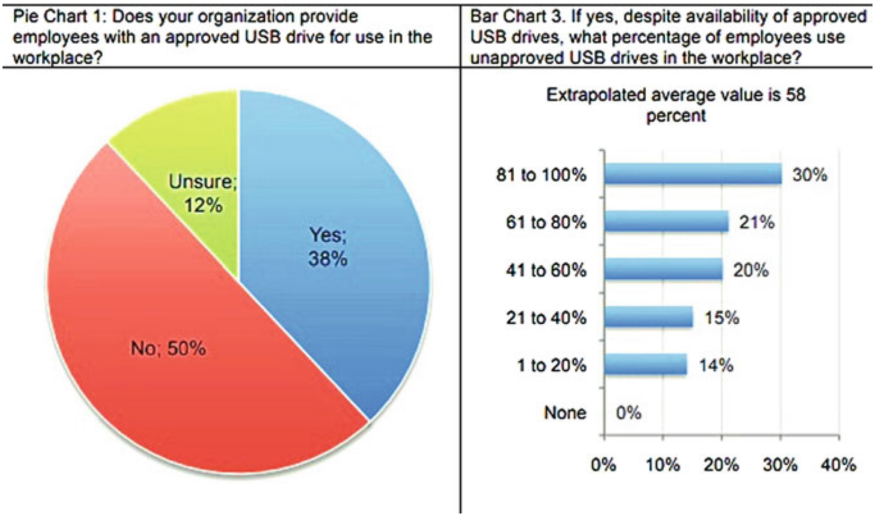


Fig. 3 USBs data loss [3]

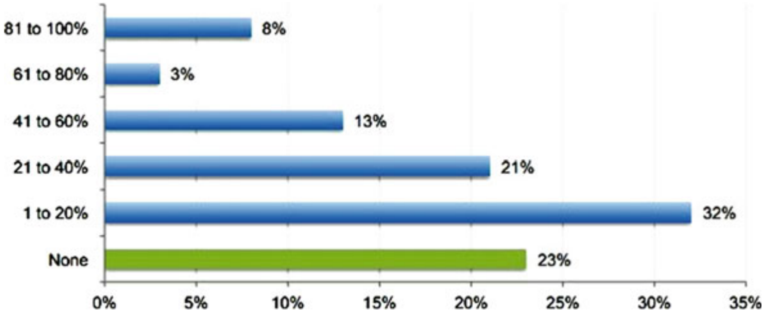


Fig. 4 What percent of USB drives used in the workplace are safe and secure? [3]

2 Literature Review

Biometric authentication is a process which uses specific modality to verify user such as fingerprint, retina etc. But we are using fingerprint for authentication purpose. Since everyone on Earth has unique and distinctive fingerprint and it is one of the most used modalities for authentication. It is also quite accurate and quick process.

Advanced Encryption Standard AES comes under the category of symmetric cryptography. In symmetric cryptography same key is used to encrypt and decrypt message. That key is called shared key/private key. AES is based upon substitution/permutation and works on block of either 128,192 or 256 bits. Main advantage of using symmetric encryption over asymmetric is that its fast and efficient for larger data. And there is need to keep the key secret in asymmetric encryption. Amongst other modes of AES, XTS mode [4] is selected.

XTS was added to the catalogue of AES block cipher modes back in 2010 by NIST of Science and Technology). It is used by Data Traveler 4000G2 and Data Traveler Vault Privacy 3.0. The intention of designing AES-XTS was to develop such a move that vanquishes the shortcomings of other modes of AES. It eradicates the possible vulnerabilities linked with some side channel attacks which can be used to exploit drawbacks of other modes.

Two AES keys are used by XTS. Where one key is used for block encryption whereas the other is used to encrypt a value known as tweak value. Galois polynomial function is used to bring further modification in tweak value and then it is XORed with both plaintext and ciphertext of each block. The purpose of GF is to provide diffusion and to make sure that identical ciphertexts are not produced by identical plaintexts. This factor allows XTS to produce unique ciphertexts from identical plaintext without making use of initialization vector and chaining. Consequently, the text is now double encrypted using two independent keys, and text is decrypted by reversing the process. Since there is no chaining involved so if stored cipher is damaged or becomes corrupted, only the data of that specific block will be affected. With chaining involved there is possibility of error propagation when decrypted (Fig. 5).

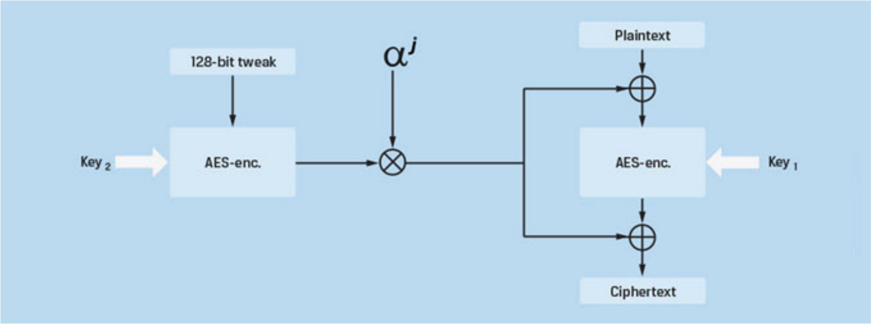


Fig. 5 XTS mode of AES [5]

For authentication purpose we will be using R305 sensor [6]. Users are authenticated through their fingerprint, R305 is one of the most widely used module for fingerprint authentication, with the assistance of DSP in its core. User communicates with this module by using hex codes in a certain specific format command. The red LED on this scanner indicates regarding the state of sensor (ON or OFF).

Each R305 module has specific identifying address and while communicating with another system each instruction is transferred in the form of packet which indicates the address of the device. This module only gives response to those data packages whose address is same as its identifying address. This address is 4 bytes long. By default, its factory value is 0xFFFFFFFF. Almost 500 ms are required for initialization of R305.

Safe Authentication Protocol for Secure USB Memories [7], User and device Authentication protocol is way more important than data encryption itself, manufacturers mainly focus on keeping data secure which leaves a flaw in the security of secured USB drives making them vulnerable, Fingerprint is the most used biometric modality because of its two traits uniqueness and permeance Fingerprint based symmetric cryptography [8] uses same fingerprint for encrypting and decrypting data, it uses a string of binary number extracted from fingerprint template act as cryptographic key. This cryptographic key is used for encrypting messages and for decryption process the key is generated from fingerprint instance and then both cryptographic keys are compared. Fingerprint is the most used biometric modality because of its two traits uniqueness and permeance Apart from being user friendly this algorithm got vulnerable due to fingerprint cloning. Other issues related with integration of biometric system with cryptographic system [9]. Techniques which are based on biometric authentication generate a biometric key which is not beneficial in cryptographic applications since they involve sharing unencrypted biometric data over an insecure channel thus, such applications require spawning of biometric keys to release the secret encrypted message that was sent.

Some other kind of secured USB drives like EAGET, having features like fingerprint encryption, user segregated compartments, and dual storage. FU5 is combination of biometric technology and storage. The data stored in USB can only be accessed by authentication through fingerprint. Extreme by SanDisk and Kingston's defender providing symmetric encryption only.

3 Proposed Solution

Smart and secure USB which will serve as authenticated storage medium for data transfer our proposed designs will fulfill following requirements.

The biometric authentication will be done using fingerprint authentication. Data is further secured through Encryption standards i.e. AES-256, which further includes file-based encryption. An administrator in the form of admin is there to register and delete users. This project will help to protect and secure user's data from unauthorized access by having user segregated compartments. Keeps record of all the recent plugins. It will need no external power source. Providing top level features in a minimal price, it's an indigenous and first of its kind smart secure USB. It uses Raspberry pi zero having R305 attached to it with an external SD card to have user segregated compartments. Architectural design of it is as follow.

- Authentication: In this part our device will authenticate its owner and other users.
- Encryption: For security purpose we will be encrypting user's data with AES-256.
- User specific Compartments: Each user will be allowed to have specific compartment and no one else would be allowed to have access to that compartment.
- Multi- Layer Security: Both encryption and authentication together will provide multi- layer security (Figs. 6 and 7).

Experimental setup:

For authentication purpose SSU is using R305 sensor. R305 is one of the most widely used module for fingerprint authentication, with the assistance of DSP in its core. User communicates with this module by using hex codes in a certain specific format.

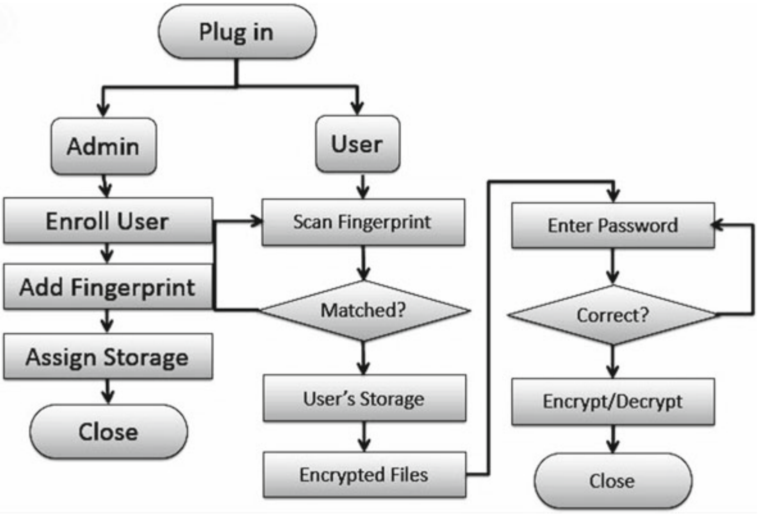


Fig. 6 Block diagram of SSU working

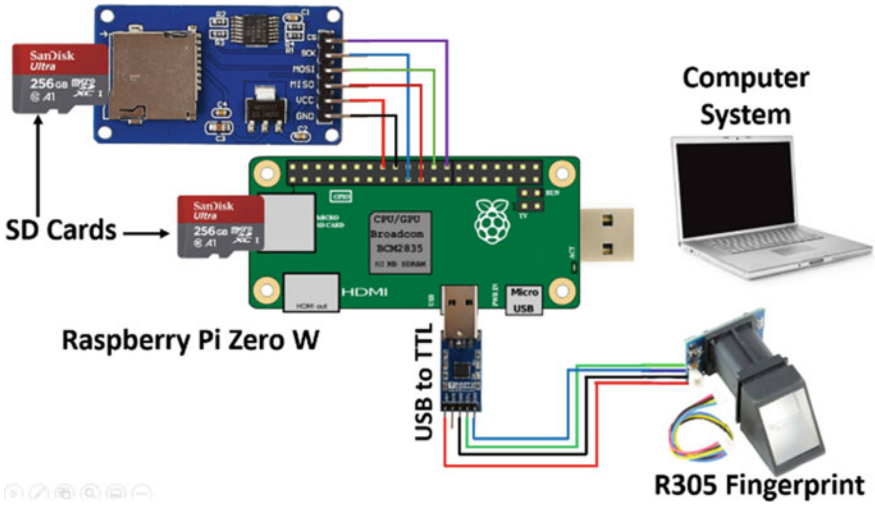


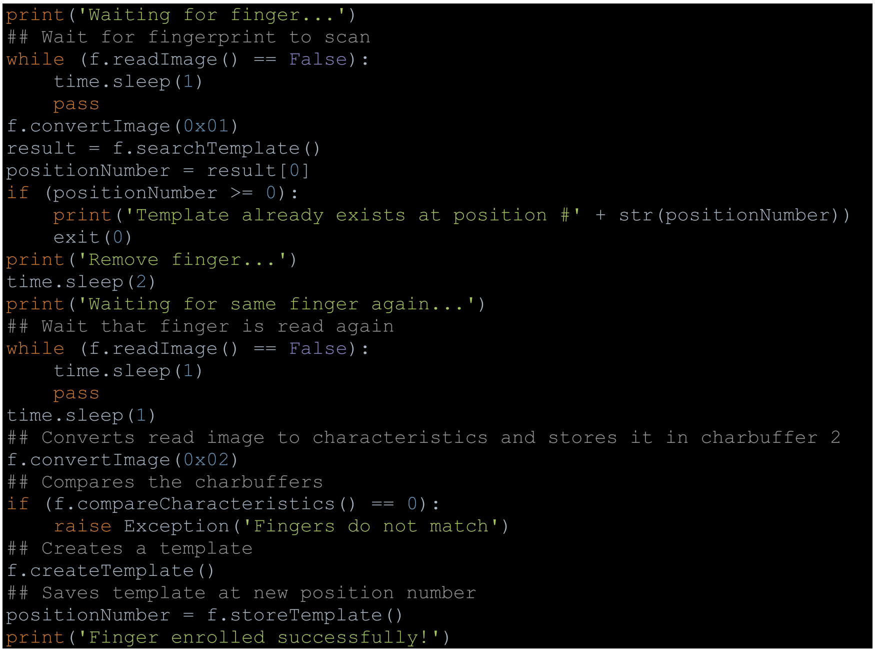
Fig. 7 SSU-256 proposed circuitry

For enrollment and deletion of fingerprints we have used library of Python which is named as “PyFingerprint”.

User Enrollment:

```
initialize fingerprint sensor
new_image = readimage ()
if new_image in stored_images
Print (ERROR! This user already exists)
else
verify_image = readimage ()
if new_image = = verify_image
image_hash = hash(image)
save_image(new_image)
username = input(username)
assign_image_hash_to_username ()
assign_storage_to_user ()
else
print (ERROR! Image not matched)
```

Code:

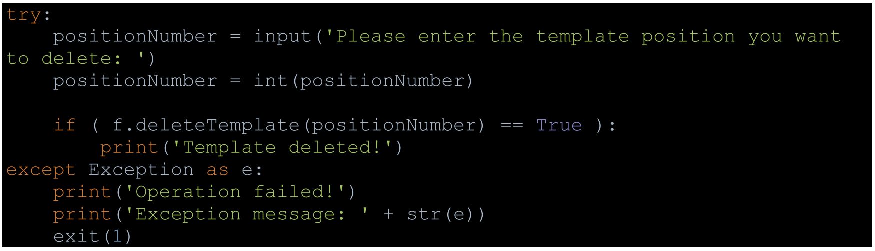


```
print('Waiting for finger...')
## Wait for fingerprint to scan
while (f.readImage() == False):
    time.sleep(1)
    pass
f.convertImage(0x01)
result = f.searchTemplate()
positionNumber = result[0]
if (positionNumber >= 0):
    print('Template already exists at position #' + str(positionNumber))
    exit(0)
print('Remove finger...')
time.sleep(2)
print('Waiting for same finger again...')
## Wait that finger is read again
while (f.readImage() == False):
    time.sleep(1)
    pass
time.sleep(1)
## Converts read image to characteristics and stores it in charbuffer 2
f.convertImage(0x02)
## Compares the charbuffers
if (f.compareCharacteristics() == 0):
    raise Exception('Fingers do not match')
## Creates a template
f.createTemplate()
## Saves template at new position number
positionNumber = f.storeTemplate()
print('Finger enrolled successfully!')
```

Delete User:

```
username = input(username)# Enter username of user to be deleted
If username exist
image_hash = read_hash(username)
Delete(image_hash)
Delete(username)
Else
Print (ERROR! This user does not exist)
```

Code:



```
try:
    positionNumber = input('Please enter the template position you want
to delete: ')
    positionNumber = int(positionNumber)

    if ( f.deleteTemplate(positionNumber) == True ):
        print('Template deleted!')
except Exception as e:
    print('Operation failed!')
    print('Exception message: ' + str(e))
    exit(1)
```

SSU-256 is using XTS mode of AES 256. It is the standard mode of AES 256 used in market since, it caters for the issues in other modes. Lack of diffusion is exhibited by ECB mode since it generates same ciphertexts for identical plaintexts. Whereas in

CBC encryption of each block is sequential so each block needs to be calculated to calculate next block. So, encryption/decryption cannot be parallelized in this mode. In this XTS mode randomization and efficiency is achieved through Tweak value and Galois function.

In XTS each data block is assigned a positive integer value known as tweak value. It starts off from a random integer and then assigned one after the other. Tweak value must be converted to little endian byte array.

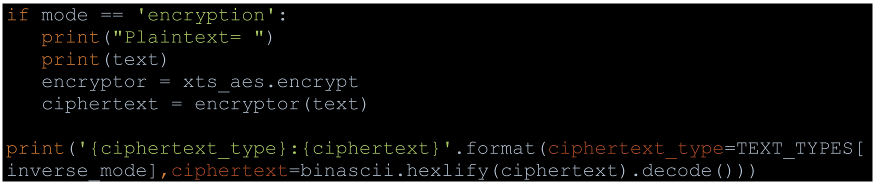
For the sake of key derivation SSU-256 is using a function called Password Based Key Derivation Function (PBKDF-2) [10]. It is just a simple cryptographic key derivation function. It is immune to dictionary attacks and rainbow table attacks. It takes several parameters as an input and generates a derived key as an output with following intakes:

- Password
- Salt
- Count of iterations
- Hash Function
- Derived-key-length.

For Encryption:

```
Select file to encrypt
plaintext = read(file)
password = input(password)
cfm_password = input(cfm_password)
if password_validity = = True
if password = = cfm_password
hash = hash(password)
# Store hash to be used for decryption
save(hash)
key = PBKDF2(password)
ciphertext = encrypt (AES, key, plaintext)
else
Print (ERROR! Password Mismatched)
else
Print (ERROR! Invalid_Password)
```

Code:



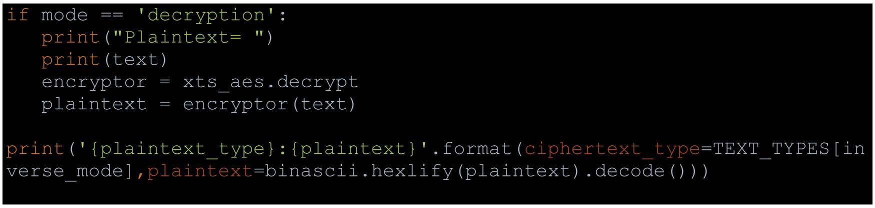
```
if mode == 'encryption':
   print("Plaintext= ")
   print(text)
   encryptor = xts_aes.encrypt
   ciphertext = encryptor(text)

print('{ciphertext_type}:{ciphertext}'.format(ciphertext_type=TEXT_TYPES[
inverse_mode],ciphertext=binascii.hexlify(ciphertext).decode()))
```

For Decryption:

```
file = input (filename of encrypted file) # Enter same password used at the time of encryption
password = input(password)
hash = hash(password)
verify_hash = read(stored_hash)
If hash = = verify_hash
key = PBKDF2(password)
ciphertext = read(filename)
plaintext = decrypt (AES, key, ciphertext)
Else
Print (ERROR! Wrong Password)
```

Code:



```
if mode == 'decryption':
   print("Plaintext= ")
   print(text)
   encryptor = xts_aes.decrypt
   plaintext = encryptor(text)

print('{plaintext_type}:{plaintext}'.format(ciphertext_type=TEXT_TYPES[in
verse_mode],plaintext=binascii.hexlify(plaintext).decode()))
```

We have segregated the memory into user specific compartments. Each user will be allotted a specific compartment and no user will be allowed to look into anyone else's compartment. We are using FAT32 file system. File system basically controls how the data is stored or retrieved. If there was no file system, then the data would be stored as a single piece of data like user cannot tell when one data has stopped and the next has begun. Data is segregated into pieces where each piece is properly identified by giving it a specific name.

Encryption along with authentication provides multi -layer security to the data of user. Plus, our idea also facilitates user with file-based encryption, Additional feature of PBKDF2 makes it harder for attacker to brute force the password.

4 Analysis of the Proposed Solution

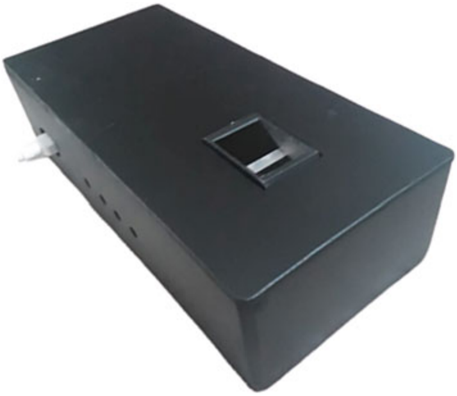
SSU-256 stands out tall amongst its market competitors.

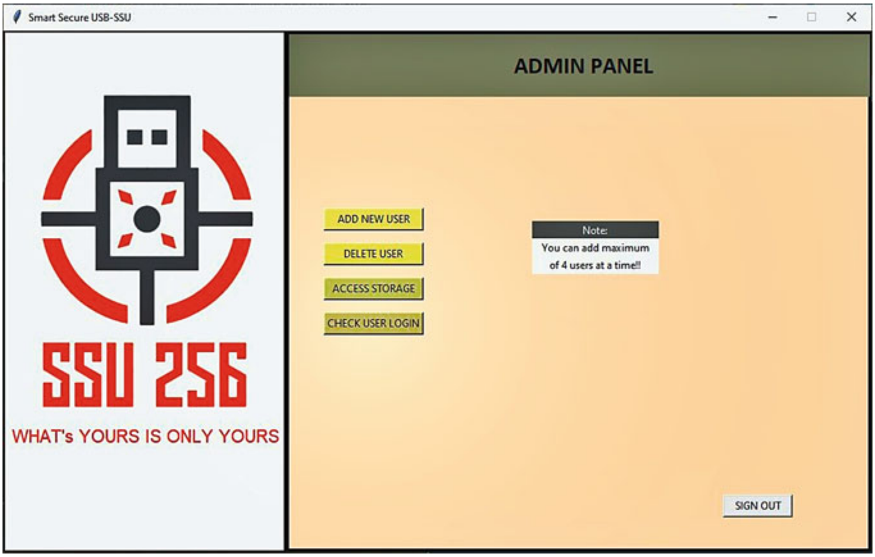
- Providing user, a two-layer security i-e Authentication and encryption.
- File base encryption makes separate passwords for all files hence adding more protection to the secured data.
- Keeps record of all the recent plugins.
- Pakistan based Indigenous solution to unsecured USBs data breaches.
- No external power source is needed.
- Providing top level features in a minimal price (Fig. 8).

Features	SanDisk	Eaget	Kingston	SSU-256
Authentication	✘	✓	✘	✓
Encryption	✓	✘	✓	✓
Multi-users	✘	✓	✘	✓
Multi-layer security	✘	✓	✘	✓
Admin rights	✘	✓	✘	✓
Recent login history	✘	✘	✘	✓

It can serve many different fields of life such as law Enforcements, strategic organizations, Telco's, IT sector, Banks, Corporate Sector and General digital Users (Fig. 9).

Fig. 8 SSU-256





Note:
You can add maximum of 4 users at a time!!

Fig. 9 GUI of SSU-256

5 Conclusion

SSU-256 is going to stand out as a best external hard drive that gives precedence to the security of user's data which they carry and this property will make it one of the indispensable tool for those who carry sensitive data with them. In this era where everyone needs to move data from one system to other, most of the companies have banned the usage of these devices in their office premises which makes life of employers quite tough. So, SSU-256 will serve as a sound solution to these problems of data leakage, privacy issues and other attacks possible through USBs. In this modern era of data where it is the most invaluable thing and every other person wants to get hold of other's data. It is the hour of need to have such a project that is capable enough to combat these problems linked with data security.

SSU-256 with other advancements on encryption techniques, modern authentication techniques can easily enhance scope of SSU and will be even more beneficial for companies and general users.

References

1. https://www.researchgate.net/Global-growth-trend-of-data-volume-2006–2020-based-on-The-digital-universe-in-2020-researchgate_net
2. https://www.statista.com/statistics/485531/sales-volume-usb-flash-drives-germany/
3. U.S. survey of IT and IT security practitioners

4. HARDWARE ENCRYPTED USB USING AES-256 by Amir Abbas, Marvi Waheed, Kamran Anwar
5. https://medium.com/asecuritysite-when-bob-met-alice/who-needs-a-tweak-meet-full-disk-encryption-437e720879ac
6. Python-R305 Documentation, Release 1.0.0
7. Lee K, Yeuk H, Choi Y, Pho S, You I, Yim K Safe authentication protocol for secure USB memories. https://isyou.info/jowua/papers/jowua-v1n1-4.pdf
8. Barman S, Chattopadhyay S, Samanta D Fingerprint based symmetric cryptography.https://ieeexplore.ieee.org/abstract/document/7045306/s
9. Uludag U, Pankanti S, Prabhakar S, Jain AK Biometric cryptosystems: issues and challenges. https://scholar.google.com.pk/scholar?q=Biometric+Cryptosystems:+Issues+and+Challenges+by+Umut+Uludag,+Sharath+Pankanti,+Salil+Prabhakar+and+Anil+K.+Jain&hl=en&as_sdt=0&as_vis=1&oi=scholart
10. https://cryptobook.nakov.com/mac-and-key-derivation/pbkdf