Face recognition with image encryption

Numan Mohammed Abbas
CSE dept.
PES UNIVERSITY
Bangalore, Karnataka
Mohd123456789numan@gmail.com

Sushma G Herakal
CSE dept.
PES UNIVERSITY
Bangalore, Karnataka
sushmagh92@gmail.com

Deepa S
CSE dept.
PES UNIVERSITY
Bangalore, Karnataka
deepadhruthi13@gmail.com

***Abstract*— Automated face recognition is gaining popularity in various domains. It's used in catching criminals to identifying VIP customers, surveillance and has many more application. But the downfall of the existing face recognition model is that it required huge number of images to train the model. This leads to breach in privacy and if the dataset falls into wrong hands, it will pose a threat to the people whose face is trained on the model. Another issue faced is that somebody else might tamper with the dataset and add their own face and retrain the model to identify themselves has VIP customers to get access to vaults.**

This paper proposes a method where the entire collection and training of the model is done on encrypted images. Even the person who trains the model will be unknow of the fact that whose face is trained hence having complete privacy. And even if someone manages to get hold of the dataset the images are in encrypted form which is entirely in a non-readable format and can't be decrypted until the person has a key. This leads to complete secured training of the model without any threat on privacy.

Keywords: VGG-16, Face recognition, Image encryption, Privacy

I. Introduction

The image encryption is to transmit the image securely over the network so that no unauthorized or any unknown user can able to decrypt the image. Image encryption, video encryption have applications in many fields . The progression of encryption is moving towards a future of endless possibilities. The image data have special properties such as bulk capability, high redundancy and high correlation among the pixels. Encryption techniques are very useful tools to protect secret information. Encryption defined as the conversion of plain message into a cipher text that cannot be read by any people without decrypting the encrypted text. . Decryption is the reverse process of encryption which is the process of converting the encrypted text in to original plain text, so that it can be read .Encryption of data has become an important way to protect data resources especially available on the internet, intranets and extranets and at any kind of network. Encryption is the process of applying special mathematical algorithms and keys to transform digital data into cipher code before they are transmitted and decryption involves the application of mathematical algorithms and keys to get back the original data from cipher code. The main goal of security management is to provide authentication of users, integrity, accuracy and safety of data resources.

Face recognition is a technology capable of matching a human face from a digital image or a video frame against a database of faces. Face recognition is one of the few biometric methods that possess the merits of both high accuracy and low intrusiveness. It has the accuracy of a physiological approach without being intrusive. For this reason, since the early 70's (Kelly, 1970), face recognition has drawn the attention of researchers in fields from security, psychology, and image processing, to computer vision. Numerous algorithms have been proposed for face recognition; for detailed survey please see Chellappa (1995) and Zhang (1997).

II. LITERATURE REVIEW

A. IoT security challenges and countermeasures(2020)

Considering the IoT acceptance rate, several devices connected to IoT is increasing every day; however, various Security challenges facing the IoT layers, some of the methods are as follows.

1. Security mechanisms for PL

Pursuing protective mechanisms should be observed to preserve the PL from malicious attacks: 1.1. Hashed based encryption Hashed based encryption security provides a facility of encryption in which a message is converted into an the length of the message. It always has a key of double-length from the message. Therefore, it is not an easy task to break a key. The key is also Forwarded to the receiver. The receiver can convert the ciphertext into an original message by using the key .1.2. PKI Protocol a Public Key Infrastructure (PKI) like protocol mechanism is a combination of all the mechanisms described. Above the encryption method such as authorization, authentication, and Intrusion detection; it is implemented in the Recognition layer of IoT architecture. It is better than using different mechanisms individually. There are many nodes Connected, and they make a network. It has a responsibility to provide security. Therefore, it does not trust anyone to Send a message. It uses an RSA encryption algorithm as the public key and private key, respectively. The public key Is stored at the base station while the private key is distributed to each node by a base station.1.3. Lightweight cryptography Lightweight cryptography [is a sub-category in cryptography that intends to provide security solutions for Resource-constrained devices. Cryptography means secret writing; there are three types of lightweight cryptography Mechanisms whose names are symmetric key lightweight cryptographic algorithm, public key lightweight crypto-Graphic algorithm, and hash functions. Whereas for communication security, symmetric-key encryption is suit-able, one such lightweight symmetric key encryption scheme has been developed, which provided very effective in Securely transferring data in IoT networks.

2. Security mechanisms for AL

The following security mechanisms should be adopted to protect the application layer from any nasty attack. 2.1. Special policies and permissions Particular policies and permissions should be observed for accessing and controlling the IoT structure. Such as XACML terminology, policies are composed by a set of rules. They are in turn made of an effect (e.g., allow/deny) and A condition [11]. Outgoing/incoming traffic and the system's access request can be permitted or restricted by access control lists. 2.2 Risk assessment techniques Risk assessment techniques detect threats of the IoT system, so the application layer should be secured by the risk assessment [30]. For this need, update the firmware of the system devices to build up the security measures.

Results:

1. Hashed based encryption

For the hashed-based encryption, this method has two weaknesses, which make it very difficult to use in almost All real-life situations. First, it requires the presence of a perfect random number generator. A pseudo-random number Generator will not be sufficient, since it will have periods in the key set that it creates .The second problem is Even harder to address.

2. PKI protocol

The proposed system does not itself protect against denial-of-service attacks [14], in which the perpetrator floods A victim with an erroneous request; however, there is a method PKI4IoT makes it very hard to an attacker to take Control of an IoT device in the first place. For DoS protection, additional firewall mechanisms should be deployed. As mentioned, since we assume the hardware to be trusted, an attacker who could gain physical access to the device Could potentially extract a private key, compromising the system.

3. Lightweight cryptography

Since the lightweight algorithms were developed for specific requirements, then it is directly followed by the ad-Vantages and disadvantages of this family of algorithms. A large number of lightweight algorithms have been pro-Posed such as PRESENT, CLEFIA, LED, KANTAN, we observe that applications do not always use cryptographic.

B. *Enhanced encryption technique for secure iot data transmission(2019)*

In this paper, we have proposed an alternate enhanced cryptographic solution combing the characteristic of symmetric, asymmetric encryption algorithms and Public Key Server. Here, the key pairs of end points (User's Device and IoT device) are generated using Elliptic Curve Cryptography and the respective public keys are registered in Public Key Server along with their unique MAC address. Thereafter, both the ends will agree on one common private secret key using Elliptic curve Diffie Hellman, which will be the base for further cryptographic process using AES algorithm. This model can be called as multi-phase protection mechanism. It will make the process of data transmission secure enough that no intermediate can tamper the data.

Results:

In our proposed model, we have four main key stages for secure data transmission. proposed model established a secure channel along with sub sequence cryptographic processes. Main goal is to transfer data/commands effectively and efficiently from one end (IoT) to another (User) securely while maintaining the Confidentiality, Integrity, Authentication. Initially, entities (IoT devices, User/Users device) have to register itself in Public Key Server by passing its Public Keys, which is generated using Elliptic Curve Cryptography, Mac Address, which is unique to everyone as a parameter. If already registered then the registration process is skipped. After that, both the entities (IoT Device and User) will exchange their public keys securely with proper authentication so that no interception happens. Upon successfully exchanging the Public Keys, both entities will agree upon one common shared secret key which is computed using Elliptic curve Diffie Hellman. The Shared Secret Key will be an input to AES encryption algorithm, which will be responsible for encrypting and decrypting our commands/data being sent over network.

C. *SIT: A Lightweight Encryption Algorithm for Secure Internet of Things (2017)*

In this paper we propose a lightweight encryption algorithm named as Secure IoT (SIT). It is a 64-bit block cipher and requires 64-bit key to encrypt the data. The architecture of the algorithm is a mixture of feistily and a uniform substitution-permutation network. The hardware implementation of the algorithm is done on a low cost 8-bit micro-controller. In this algorithm the encryption process consists of encryption rounds, each encryption round includes mathematical operations that operate on 4 bits of data. To create sufficient confusion and diffusion of data in order to confront the attacks. However, the proposed algorithm is restricted to just five rounds only, to further improve the energy efficiency.

Results:

The simulation of the algorithm is done to perform the standard tests including Avalanche and image entropy and histogram. The memory utilization and execution time of the proposed algorithm is observed. The execution time is found to be 0.188 milliseconds and 0.187 milliseconds for encryption and decryption respectively and the proposed algorithm utilizes the 22 bytes of memory on AT mega 328 platform. The Avalanche test of the algorithm shows that a single bit change in key or plain text brings around 49% change in the cipher bits, which is close to the ideal 50% change. To perform entropy and histogram tests we have chosen five popular 8-bits grey scale images. An 8-bits grey scale image can achieve a maximum entropy of 8 bits. From the results it can be seen that the entropy of all encrypted images is close to maximum, depicting an attribute of the algorithm. A good cipher is expected to remove the dependency of the cipher text from the original message. Therefore, Original data, which in our case is an image can be seen to be highly correlated and detaining a high value for correlation coefficient. Whereas the encrypted image does not seem to have any correlation.

D. *Towards Designing Efficient Lightweight Ciphers for Internet of Things (2017)*

In this paper, a total of 13 lightweight cryptographic algorithms are evaluated based on their implementation results on 8-bit, 16-bit, and 32-bit microcontrollers and their appropriateness is examined for resource-constrained scenarios like IoT. These algorithms are analyzed by dissecting them into their logical and structural elements. This paper tries to investigate the relationships between the structural elements of an algorithm and its performance. Association rule mining is used to find association patterns among the constituent elements of the selected ciphers and their performance. Interesting results are found on the type of element used to improve the cipher in terms of code size, RAM requirement and execution time. This paper will serve as a guideline for cryptographic designers to design improved ciphers for resource constrained environments like IoT. first, we present the performance analysis of implementing the lightweight cryptographic algorithm on three platforms: AVR microcontroller, MSP microcontroller and ARM microcontroller. The analysis is based on three factors: code size, RAM foot print, and execution time. Secondly, we present the results of Association Rule Mining applied on constituent elements of lightweight cryptographic algorithms.

Results:

In this paper, we presented an evaluation of 13 light weight block ciphers used for secure communication in Internet of Things. We compared and ranked the ciphers based on three metrics: code size, RAM foot print and execution time. We analyzed the performance of these ciphers on three different platforms: 8-bit, 16 bit and 32 bit. We further dissected these ciphers into their constituent elements and investigated the role of these elements in the performance of ciphers. We used association rule mining to find associations among the constituent elements. Based on the results, we come up with few guidelines regarding the design of lightweight ciphers. Designer must always remember the algorithm prerequisites to be implemented into the devices. So, intention must be to consume less device resource e.g., memory (RAM), code size, execution time etc. The S-box have to be small generally (4 × 4) bits for compact operation. Simultaneously, it must deliver compulsory non-linearity to the algorithms. Key schedule has to be easy so that it takes small space, hence the recently planned cipher keeps the keys fixed. As the algorithms are sprightly implemented into the device, therefor no need for re-keying. The permutation has to be designed in such a way that it attains optimum stability among mixing of bits and areas. The designer must attempt to accomplish an optimum balance amid the different parameters of cost, security and performance. In short, this research work aimed to provide basement to improve the cipher in several ways like code size, size of memory (RAM Requirement), and execution time.

E. Secure Data Transmission for Iot Applications (2016)

In this paper, providing Network security and Data security are the point of concern. Network security is the use of software, hardware, and procedural methods to protect IOT applications from attackers and Data security is the use of codes, algorithms and encryption techniques for the protection of IOT applications. Protocols used in this project for securing the data transmitted in either of the communication ways are: Node to gateway, Gateway to node, Node to node, End to end A) MQTT (Message Queue Telemetry Transport): is an application protocol viewed as a publish subscribe model, designed for the communication of M2M. SSL/TLS - To implement security for the data transmission between the nodes (pi) and gateways (pc) in an IOT context. SSL/TLS (Secure Socket Layer/Transport Layer Security) is to be used for MQTT protocol. Since, MQTT relies only on TCP (connection oriented) as transport protocol, by default this connection does not use an encrypted communication. To encrypt the entire MQTT communication, it allows using TLS instead of plain TCP.This is carried out by TCP handshake. B) CoAP: DTLS is used to protect the CoAP protocols. As CoAP (Constrained Application Protocol) is a web protocol which relies over UDP (User Datagram Protocol; which is connectionless) protocol used mainly for the constrained M2M devices in the IOT, TLS is not used here; instead, encryption is done using DTLS (Datagram Transport Layer Security). Most of the constrained device (CoAP) implementations are carried by libcoap packages; this can also be used on the server side. The security implement for MQTT protocol is done using openssl library function of SSL/TLS encryption method and security implementation for CoAP protocol is done using asyncio function of DTLS encryption method at the node transmission of data coming through the gateway. A) OpenSSL Creating the structure of node-to-node communication such as raspberry pi's and PC's acting as gateways for the two nodes. In this scenario data to be transmitted from the gateway are secured at the node point usingSSL/TLScryptographic methods which includes handshake mechanisms to establish the connections i.e., raspberry pi to PC over openssl. B) The data transmission is been carried through the gateway securing it with datagram encryption method of CoAP and passing to the node point using the client and server DTLS encryption mechanism, data coming from the cloud sent by client are secured at the servers. C) C. End to end communication Proposed Secure End to end connection communication system between nodes and gateway is shown in the figure: At right of the figure shown below the security of the data is maintained between the gateway and the node i.e., the data coming from the cloud (gateway) is been secured to read it at the node point terminal. Tool used to detect Network level Vulnerability: Nmap ("Network Mapper") is a free tool available to download and it's also an open source (license) for network discovery and security checking. Tool used to detect application-level vulnerability: Burp suit is the actual tool used to detect the application-level vulnerability. The transmission of data security is completed with different protocols like MQTT and CoAP using its corresponding library functions libssl and libcoap respectively. Different vulnerabilities are mentioned in the project like network layer and application layer vulnerable. Performance of SSL is carried out by Comparing the desktop computer system and raspberry pi platform observing the factor of 33 differences in terms of performance between the computer system and the embedded platform.

Results:

OpenSSL: Allowing the SSL/TLS handshake for the connection establishment at the gateway and the node ensures the end-to-end security. The data to be transmitted now between the node to node or node to gateway or vice version is possible securely.

MQTT and SSL: In this paper A screenshot of the publisher messages received on console window of the subscriber is shown. The screenshots show the output of: Console of the broker, Message received from MQTT QoS, Subscriber.

NAMP: In network level vulnerabilities, using NMAP the following vulnerabilities are detected: SSL is not enforced of login, Poodle, Beast.

HTML: pages are vulnerable to 2 different commands of XSS (third vulnerability in top 10 OWASP) and they are: •Bad attribute XSS command • Bad script XSS command.

III. PROPOSED METHOD

In this work, the dataset is collected by using open cv and Haar Cascade. The collected images are encrypted using our encryption function and stored. The encryption function takes path of the image and the key as input values. The image is converted into byte array code and then we negate the byte array code and apply xor with each individual characters of the key. Then we again apply negation. These coverts the image to a non-readable format and the encryption happens in layers.

For training the model we only pass the path of the dataset and the key for decryption. If the wrong key is passed then the dataset changes to an unreadable format and cannot be recovered.

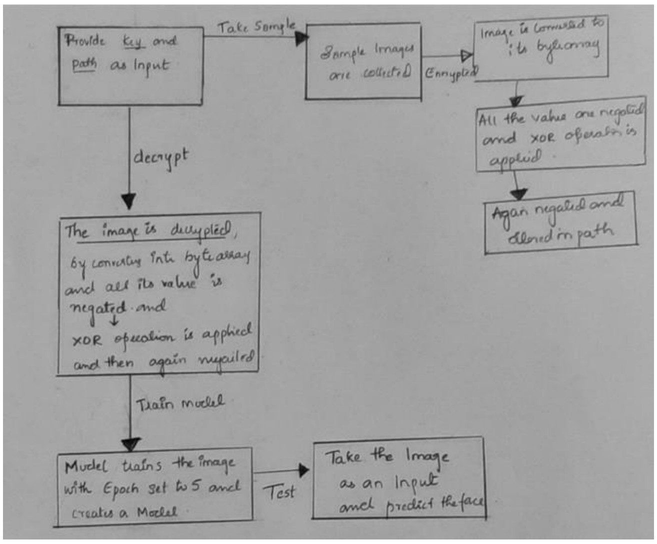

Provide key and path as Input
Take Sample
Sample Images are collected
Encrypted
Image is Converted to its bytearray
All the value are negated and XOR operation is applied
Again negated and stored in path
decrypt
The image is decrypted, by converting into bytearray and all its value is negated and XOR operation is applied and then again negated
Train model
Model trains the image with Epoch set to 5 and creates a Model
Test
Take the Image as an Input and predict the face

Figure1.Diagram depicting working of our Project.

We use transfer learning to train our face recognition model. The pre-trained model used in this work is VGG16.

The model is saved and for the testing of the model the saved model is loaded and run thus preserving complete privacy of the images in the dataset.

IV. EXPERIMENT AND RESULTS

For creation of dataset, we used OpenCV and Haar Cascade to capture 2000 images for each individual whose face had to be trained. Once the dataset is encrypted and stored,

No one can access the trained images since it will not be in a human readable format.

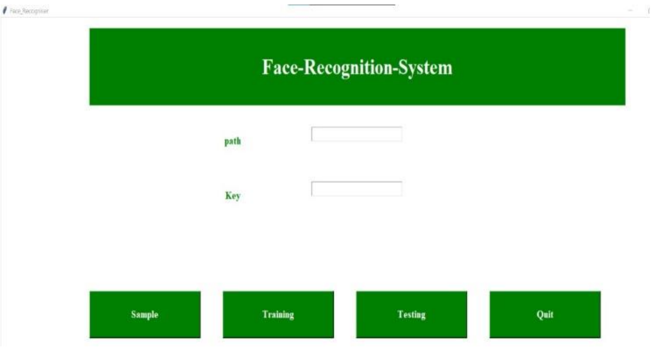
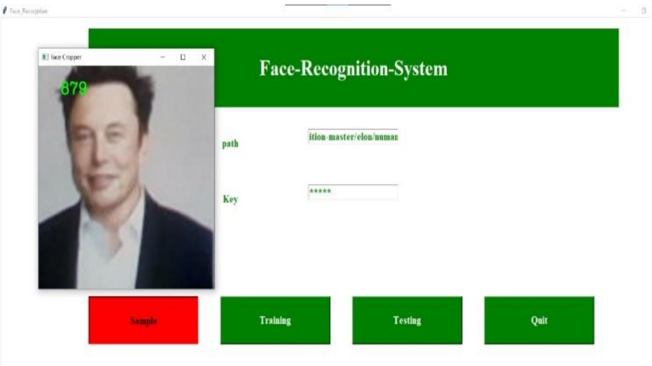


Figure2. Face Recognition system



ition-master/elon/numas

*****

Figure3. Collecting sample images after the path and key are given

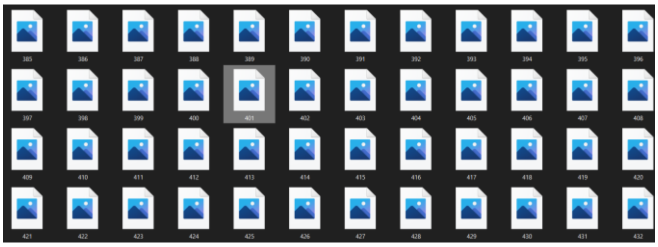
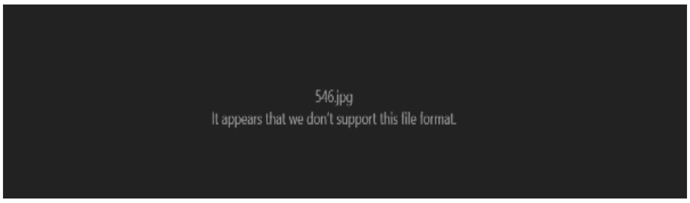


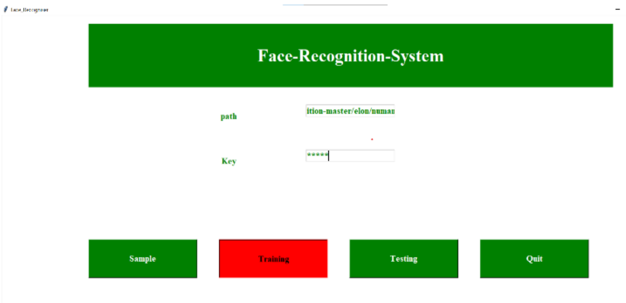
Figure4.The sample images which are encrypted



546.jpg
It appears that we don't support this file format.

Figure5. Error message when we try opening encrypting image

To train the face recognition model we use VGG-16 Deep Learning pre-trained model. This model takes only the path of the dataset and the decryption key as the input. The image is decrypted before its trained.

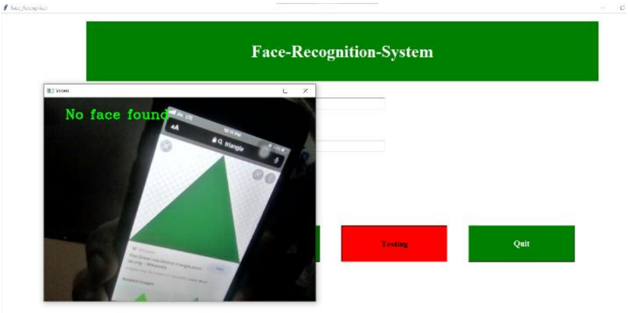


path ition-master/elon/numan

Key *****

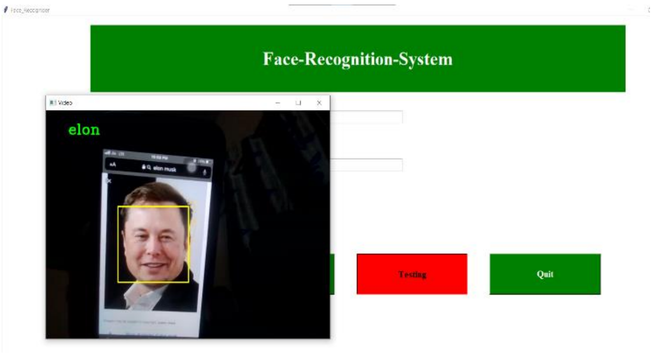
Figure6. Entering path and key for training

We use already pre-existing weights to train the model and SoftMax as its activation function. An epoch of 10 was used and it gave overall accuracy of 98-99 when tested on 400 images of individual person. Once the model is trained, we save the model. Then the model is loaded and is ready for recognising the faces. In the entire process only, the model has access to the encrypted images and therefore no 3rd person can access the images.



No face found

Figure7. Testing a image which is not trained



elon

Figure8. Face Recognition system is identifying the image.

V. Conclusion and Future work

The main goal of our work was to preserve the privacy of the all the images which is used for training the model. We modified the xor encryption so that it's not easily detectable that xor encryption was used. The encryption technique is fast and since it's done in layers it's difficult to decrypt.

Our model runs on a local system. For future work we will try creating a cloud-based application where the encrypted images are stored on the cloud. We will also make use of iot devices to get the images and also try different cryptography algorithms.

VI. Acknowledgment

I would like to thank our guide teacher Asst. Professor Vineetha B Dept. of CSE, PES University, Bengaluru for guiding us and supporting us throughout the project. I would also like to thank our college PES UNIVERSITY, Ring road campus Bengaluru for providing this opportunity to work on this project.

VII. References

[1] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.413.3569&rep=rep1&type=pdf

[2] https://www.researchgate.net/publication/322277374_A_Survey_and_Analysis_of_the_Image_Encryption_Methods

[3] https://www.academia.edu/download/48949207/1.pdf

[4] https://www.researchgate.net/publication/342148798_Enhanced_encryption_technique_for_secure_iot_data_transmission

[5] https://www.researchgate.net/publication/342148798_Enhanced_encryption_technique_for_secure_iot_data_transmission

[6] https://arxiv.org/abs/1704.08688

[7] http://www.itiis.org/digital-library/manuscript/1772