iG Protection (Data Encryption using Gestures)

Namit Mohale, Sourabh Gupta, Shubham Goyal, Ruchi Goel

Abstract: ***In today's world, unimaginable bytes of data is being generated every second around the globe. A lot of this data is private and confidential to individuals or organizations, and they do not wish for others to access this data. They might keep the data safe in portable drives, but as soon as the data gets connected to a network or is being transferred through a network, it becomes highly vulnerable to threats of it being stolen by hackers, as network has the biggest loopholes as compared to rest when it comes to security. The data is also vulnerable to being physically stolen. Depending on the delicacy of data, it could be highly misused for all you know it could be the nuclear launch codes. Here we propose a method to not only encrypt the data for protection, but to take it to a different level. Whilst data encryption algorithms auto-generate secret keys that are used to encrypt the data, we propose in our method to use hand gestures to generate to generate the key, each key being unique to a certain set and sequence of hand gestures, to be used as encryption key in encryption algorithm. Hence we are using Blowfish algorithm for data encryption and Open CV for gesture recognition. Blowfish being a symmetric key algorithm, the same set of gestures that were used to encrypt the data would be used to decrypt it.***

Keywords: Data Encryption, Blowfish Algorithm, Gesture Recognition, Open CV, Artificial, Intelligence, Symmetric Key Encryption, Decryption.

I. INTRODUCTION

We live in a world where roughly 2.5 quintillion bytes of data is generated every data. 90% of the data total data in this world today has been generated in the last two years, and that is a lot for such a small time. And with data switching to digital form from manual data every second, threats are more imminent with this shift. Digital data is more vulnerable to threats than hard data. Now, what are threats? The simplest threat that comes to our mind when we think about data is it being stolen. But that is not the real threat, instead, what all can happen with or to that data is what really makes the difference.

Revised Manuscript Received on 30 May 2018.
***** Correspondence Author
Namit Mohale*, Department of Computer Science and Engineering, Maharaja Agrasen Institute of Technology (Affiliated to Guru Gobind Singh Indraprastha University), Rohini Sector-22, New Delhi, India. E-mail: mohale.namit95@gmail.com
Sourabh Gupta, Department of Computer Science and Engineering, Maharaja Agrasen Institute of Technology (Affiliated to Guru Gobind Singh Indraprastha University), Rohini Sector-22, New Delhi, India. E-mail: gsourabh92@gmail.com
Shubham Goyal, Department of Computer Science and Engineering, Maharaja Agrasen Institute of Technology (Affiliated to Guru Gobind Singh Indraprastha University), Rohini Sector-22, New Delhi, India. E-mail: shubham.goyal25sep@gmail.com
Ruchi Goel, Department of Computer Science and Engineering, Maharaja Agrasen Institute of Technology (Affiliated to Guru Gobind Singh Indraprastha University), Rohini Sector-22, New Delhi, India. E-mail: ruchigoel@mait.ac.in

The data stored digitally or being wired could be anything, from someone's personal pictures to a business's money related secrets to an entire nation's defense strategies. Now, with technology, the attempts and methodologies to increase the security for the digital data has become more sophisticated.

Now, it doesn't matter if the data is stored on the most remote system or is being transferred through the most secure network, there can always be someone who will be able to find out a loophole in the system and use the vulnerability to his advantage to harm the data. The data could be erased, stolen, sold to rivals or simply manipulated to change its meaning. Hence, there is a need to keep changing the way data is kept secured and transferred. Encryption came out as an effective manner to secure data in which the original data is scrambled with the help of a secure key, which is very long and is kept a secret for the safety of the data, and can only be decrypted with the same or the pair key. Here, we propose a different way to generate the key to encrypt the data using blowfish algorithm, which being a symmetric key algorithm, can unlock the data only with the same key.

Live gesture recognition is trickier than using machine learning for image recognition. Image recognition makes use of training data to train a model to predict the new fed data, whereas live gesture recognition takes in live feed from the camera to analyze the input gestures using the inbuilt algorithm. For this purpose, we are using the Open CV library by Intel in its Android implementation form to analyze for 4 different gestures; left, right, up and down. A group of these gestures could be used to generate the key to encrypt the data, which could only be decrypted later with the same set and sequence of gestures.

A. Blowfish Algorithm

Blowfish is a symmetric-key block cipher, designed in 1993 by Bruce Schneier and follows a Feistel Structure. Blowfish was designed as an alternative to the aging DES and free of the problems and constraints associated with other algorithms. It is unpatented and hence open to use for free world-wide. Though AES works better, is new and more efficient, but its implementation is not free. An overview is given in Figure 1.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

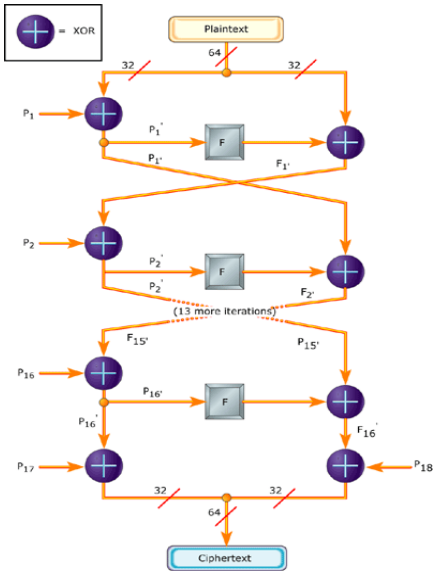

Figure 1: An overview of Blowfish Algorithm

B. Open CV

OpenCV (*Open Source Computer Vision*) is a library of programming functions aimed at real-time computer vision. It was developed by Intel Russia Research Centre. It is written in C++ but its full interfaces are available in Java, Android, Python and MATLAB/OCTAVE. It works better than MATLAB because it occupies lesser resources (approx. 70 MB of RAM) to operate while MATLAB occupies more than 1GB of RAM for operating.

II. OBSERVATION

Live palm tracking using OpenCV required high system configuration and a good quality camera to accurately extract the palm from the live feed and differentiate it from the background using colour profile, hence we preferred developing the application for the task on the Android platform, as the mobile phones nowadays are fast, high in memory and include high-end cameras.

A. Problem Definition

Data safety is one of the major concerns of today's world where the major part of information exists in digital form. With advancements in technologies, it has become easier to exploit the vulnerabilities present in the systems and hence it is required to use more sophisticated safety or encryption methodologies. The data to be secured can range from personal to highly confidential to dangerous data belonging to individuals or organizations or governments. The need to data security has led to development of various encryption algorithms which scramble the original data using the secret keys and can only be decrypted using that particular algorithm and the secret key, which as the name suggests, is kept secret with the user. If the data is to be shared, the key is shared through secure networks. Since there are so many levels involved in this process of data encryption and sharing, it is highly possible that some vulnerabilities are left unnoticed and face the danger of being exploited.

Hence, our attempt is to use blowfish algorithm to encrypt the data but instead of using the traditional method to generate the key, we aim at using a set of gestures to generate a key to encrypt the data and only those gestures can then be used to decrypt the data.

B. Proposed Work

For analyzing live gestures using input camera, we are using OpenCV which is a real-time computer vision library developed by Intel Russia Research Centre. We aim to use its Android implementation since the camera specifications in Android smart phones is appreciable and hence will facilitate better image capturing for gesture/palm tracking.

For encrypting purpose, we aim at using blowfish algorithm which was designed to overcome the problems and constraints associated with DES algorithm. Though it was later replaced by AES algorithm, but we use blowfish because it is less complex and its implementation is free. To generate the key for the encrypting algorithm, we use the Touch-Free Library which is a part of OpenCV android implementation. The touch-free library accesses the camera and the accelerometer and then uses built-in methods to analyze the directional gestures. Every gesture is given a unique key. The camera keeps on taking gesture inputs until no gesture is provided for 2 seconds. It then combines the unique keys of all the gestures to make a single key, which is then converted to 'bytes' and used as the input key for blowfish algorithm. The algorithm then scrambles the data in the original file and can only be decrypted using the same set of gestures in the same application.

III. PROPOSED METHODOLOGY

Our algorithm follows the following steps:

1. First, we will install Android Studio as an environment to code in. Then, we will install the Android implementation of OpenCV library.
2. We will use the OpenCV library to extract the hand from the live feed of camera using colour-profile.
3. We will then use the Touch-Free Library of OpenCV to analyze the gestures, which for every gesture is done by comparing the velocity of palm movement to the successive position of the palm, which gives the direction of its movement.
4. We will assign a unique key to every gesture, for example, 'PQR' is assigned to 'Left' gesture and so on.
5. The camera will keep on reading gestures until no gesture is provided for a span of 2 seconds. Hence, we will be able to provide a sequence of multiple gestures.
6. A final key will be generated by combining the keys of the input gestures in the respective sequence.
7. This key will be converted to its byte form using the 'getbytes()' method.
8. Then we will download the Blowfish algorithm implementations since they are free to use.
9. Then the final key will be used as the input key to the encryption algorithm and will be used to generate the P-array and S-array for the algorithm.
10. The algorithm will then scramble the original data.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

IV. EXPERIMENTAL AND RESULT ANALYSIS

All experiments were performed on both Android Studio installed on a Sony VAIO laptop with 8GB RAM and 1GB Graphic Card and on an Android Smart Phone: One Plus 3 with 6GB RAM running on Android Oreo (8.0.0).

To achieve the proposed methodology, we followed the below listed methods:

1) *Creating Android Application:* To be able to perform any sort of analysis on the live feed of camera, we first need to create an android application to give our algorithm a UI. The android application is a must as it makes the user experience better in choosing the file to be encrypted and using the application effectively. To do this, we use android studio on the laptop. We execute the end application on both the laptop and the mobile phone. The end application is of '.apk' format and is supported in any android smart phone running on Android version 5.0+.

2) *Extracting Hand from background*: We use the OpenCV library and its Touch-Free sub-library to analyze gestures or track palm movement direction.

3) The palm is extracted using the colour-profile by colour sampling, which in turn is done by first covering all the medians for colour extraction on screen with our hand and letting the Open CV library do its magic. The background is detected using the following equation:

$$CurBG[i][j] = alphaCurBG[i][j] + (1 - alpha)CurFrame[i][j]$$

which assumes that the background is static. Background subtraction is done using an inbuilt background subtractor based on Gaussian Mixture based Background/ Foreground Segmentation Algorithm. After these steps, a binary representation of the hand is made for each frame. Contour extraction in then performed using OpenCV's inbuilt edge extraction function. Each sampled colour in the colour-profile produces a binary image of the hand which all are then summed together to get a smooth and noise-free representation of hand. The process is depicted in Figure 2 below.

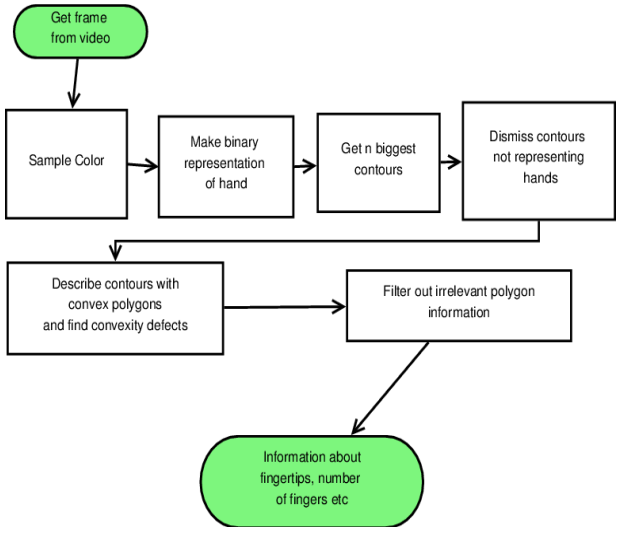

Figure 2: Hand Extraction using Open CV

4) *Tracking hand*: To track the hand, convex hull and defects are analyzed, which are actually the finger tips and the valley between two fingers respectively. Now taking the average of the position of defect points, we get the position of a rough centre of the palm and assume the palm to be a circle, getting its radius. The palm movement direction is then tracked by subtracting the centre position of palm for every successive frame. The final direction is assumed to be the one in which the received difference is maximum and positive. The camera keeps detecting gestures until no gesture is provided for 2 seconds and remembers the set and sequence of multiple gestures provided.

5) *Generating key for Blowfish*: After tracking hand using the Touch-Free OpenCV library, we assign a unique sub-key to each gesture and our algorithm combines all these keys in the sequence of gestures provided to form a primary key. This key is then converted to its byte format using the 'getbytes ()' method, which is then used as the input key for the Blowfish algorithm.

6) *Encrypting data*: Once we have the key, we start with the encryption process. First, P and S arrays are computed. P is an array of 18 32-bit integers and S is a 4x256 2-D array of 32 bit integers. Both the arrays are initialised by constants and are formed when the first 32 bits of the key are XOR-ed and written back to the array and the process is repeated for every position in the array. Then the plaintext of 64-bit is divided into two halves and the first half is XOR-ed with the first entry from the P-array. This output is sent to two places. The first one in first sent into an F function which is elaborated in Figure 3 and then XOR-ed with the second half of the plaintext, and the second one is cross XOR-ed with the output of the next F function and so on. The overall process is shown in Figure 1 above.

7) We finally receive the cipher text which replaces the original data in the file, hence encrypts it.

8) *Decrypting data*: Since it is a symmetric-key algorithm, when the same set of gestures are provided in order to decrypt the data, the same process explained above is performed on the decrypted data which re-writes the cipher-text as the original data, hence decrypting it.

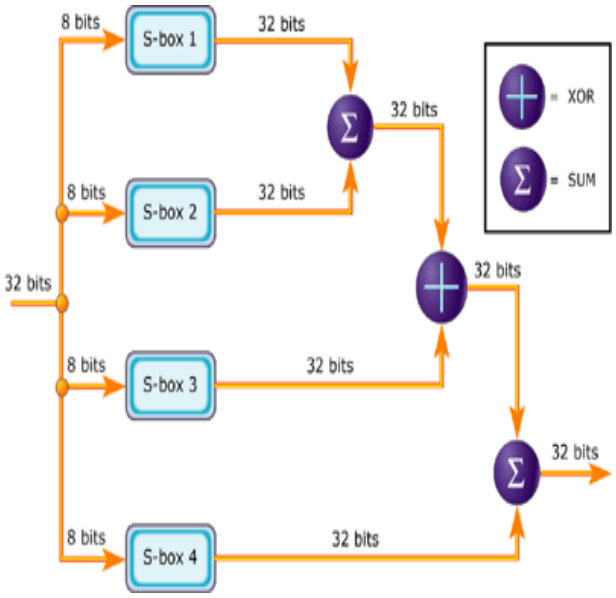

Figure 3: F Function Representation

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

V. TIME EFFICIENCY

Time efficiency is an important aspect where our project stands firm. Lower response time has been achieved by the use of data structures as local variables for P and S arrays and by the use of fast smart phone. Also, the use of OpenCV library has also facilitated the lower response time as it is a light-weight library and hence do not take up much memory.

Table: 1. Difference

	Existing Method	Proposed Method
Algorithm	No algorithm is used because there is no existing platform offering encryption through gestures.	Blowfish algorithm is used for encryption with some enhancements in gesture recognition robustness.
Security	Data can be compromised as it is present in original format which can be altered or copied.	Data is present only in encrypted form which is not understandable.
Understandability	Easy to use and understand.	User-friendly design which is easily understandable for use.
Efficiency	Not very efficient as it is vulnerable to attacks because the building elements of the cipher keys are transported over the network.	It is highly efficient as it is not vulnerable to attacks because only proposed method can decrypt the data.
Performance	Pattern matching, passwords or any other method used doesn't involve real time image processing. So, its performance is only satisfactory.	Its performance is really good, considering real time gestures recognition and encryption. It does its work really fast.
Enhancements	Better locks can be used such as finger prints, more complicated passwords etc.	Better algorithms can be used for making encryption and decryption more efficient, unique physical attributes can be used as a key such as finger prints, iris recognition etc.

VI. FUTURE SCOPE

At this point, the algorithm can very efficiently and quickly analyze the live feed from camera to track gestures and encrypt/decrypt the data. But there are still a few areas of scope for future. Sometimes when the background is not static or the colour of background is not distinct from that of the hand, the outputs are sometimes slow or in very few cases, incorrect. It is mainly the gesture recognition part of the project that presents challenges, otherwise, the encryption part, once it receives the key from the gestures, always gives the correct output. Also, blowfish is a costly encryption method in terms of memory and time since it creates large arrays to facilitate the process; we can also implement the process using some other more efficient encryption algorithm. Also, mobile phone's memory management system works different from that of the computer and hence makes the process a little slower. Hence, this project can also be shifted to a laptop computer, implementing the code in java and attaching an external camera with better configuration than the inbuilt VGA camera, which was the main reason of implementing this project on Android.

VII. CONCLUSION

Gesture Analyses is a wide branch of Artificial Intelligence which itself is a wide branch of research and has tremendous scope for improvement. As far as data security is concerned, our project manages to give an extra layer of security to the data encryption process by adding the gesture lock feature and ruling out the auto key generation step, which makes it even more difficult to break into the data.

If someone does not have the information about the gestures, even if he happens to get his hands on the system or on the data or even on the entire encryption algorithm itself, it will still be nearly impossible for him to guess the sequence of gestures provided and will be stuck with scrambled data which will make no sense.

By running the application on smart phone instead of laptop computer, the project showed dramatic change in its output, with the smart phone being more efficient due to better camera quality, making the gesture recognition part easier, faster and more correct. Hence with current limitations, the accuracy of the project was found to be nearly 72%.

ACKNOWLEDGEMENT

First I will thank the almighty for giving me the courage & perseverance in completing the project. This project itself is an acknowledgement for all those who have given me their heart-felt co-operation in making it a grand success. It is a pleasure to express my deep and sincere gratitude and sincere thanks to the project guide **Ms. Ruchi Goel (Assistant Professor, CSE Department)** and I am profoundly grateful towards the unmatched help rendered by them. I am also thankful to them for extending her sincere & heartfelt guidance throughout this project work. Without their supervision and many hours of devoted guidance, stimulating & constructive criticism, this thesis would never come out in this form. Last but not the least; I would like to express my deep sense and earnest thanks giving to my dear parents for their moral support and heartfelt cooperation in doing the project. I would also like to thank my friends, whose direct or indirect help has enabled me to complete this work successfully.

REFERENCES

1. Kari Pulli, Anatoly Baksheev, Kirill Kornyakov, Victor Eruhimov, "Realtime Computer Vision with OpenCV", ACM New York, 2012
2. Gary Bradski, Adrian Kaehler, "Learning OpenCV", Willow Garage, ICRA 2010 ROS Tutorial
3. Tanjyot Aurora, Parul Arora, "Blowfish Algorithm", IJCSCE, 2013
4. Bill Gatliff, "Encrypting data with Blowfish Algorithm", 2003
5. Schneier, Bruce, "Applied Cryptography: Protocols, Algorithms, and Source Code in C, Second Edition", New York, NY: John Wiley & Sons, 1995

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)