ONLINE BIOMETRIC VERIFICATION USING APACHE STORM ON PUBLIC CLOUD

Prof. Shraddha More 1, Jigar Maniyar 2, Mrunmayee Vartak3, Prasobh Nair 4, Chirag Raul 5

1,2,3,4,5 Department of Information Technology, SJCEM

Abstract—Biometric is one of the human trait which is being used for the identification and verification purpose. This paper entitles the importance of online Biometric recognition system in real-time application. It covers how this Biometric recognition system is implemented over the open source big data technologies such as Apache Storm and how this technologies can be put on cloud using Windows Azure. It proposes a highly scalable, pluggable and faster online Biometric recognition architecture that can handle a million Biometric events and processed them per second. It can cope with huge amount of load and massive through puts. So this architecture stack is equally suitable for Systems with high ambitions. This system contains biometrics and is real time. Passwords are like a shared symmetric secret which the user know and the online service to whom the user needs to authenticate knows. And as a user subscribes to more and more online service, he reuses the secret to authenticate itself. Moreover passwords are easily phish-able and can be easily exploited through malware exploits such as key loggers. The other and more robust option is biometrics. This System proposes a highly secure, scalable, pluggable and faster biometric system architecture. That can handle a billion of biometric events per second that are going to come from the devices which are connected to internet. The system processing of the biometric events has faster enrollment and verification process; fault tolerant & infinitely scalable database, and lastly removing insights from the biometric event.

Keywords: Apache Storm; Bots; Bot Builder SDK; Blob storage; Spout.

I. INTRODUCTION

People use passwords constantly, to login to a number of online services every single day. And equally as the number of online services, the person subscribes to Increases, the amount of passwords that person has to call up is more. There for it can be easily cracked. The alternative that exits for passwords are one time passcodes, where a five or six digit code is send to the user’s device by an SMS and the user use this code along with password to authenticate an action[1]. The issues with this one-time passcode is that if the user changes his mobile number, then he has re-register everything again, the user may be unable to get the SMS because of some network issues, he may not be able to get where the one time passcode on his device is going etc [2]. still they are phish-able. This is where the role of biometrics comes into the picture by offering quicker, easier and more robust authentication in a seamless manner. As biometrics is going to be applied for online authentication, amount of biometric data generated by it will be increasing at a very quick pace. The mobile phone is today an indispensable companion in both people’s private and professional lives [3][4]. People prefer to use their mobile phone to net banking, pay bills, transfer funds, etc. Thus to use biometric to authenticate a user online, the Smartphone applications, will take advantage of the many inbuilt mobile sensors available on mobile devices, open possibility for analyzing and processing new types of generated data, and possess an impact on almost all activities of societal and business life. Bots are as vital a way for users to interact with a service or application as is a web site or a mobile experience [5]. Developers writing bots all face the same problems: bots require basic I/O; they must have language and dialog skills; and
they must connect to users The Bot Framework provides tools to easily solve these problems and more for developers e.g., automatic translation to more than 30 languages, user and conversation state management, debugging tools, an embeddable web chat control and a way for users to discover, try, and add bots to the conversation experiences they love. The Bot Framework has a number of components including the Bot Connector, Bot Builder SDK, and the Bot Directory. Apache storm is a distributed real-time event processing engine from apache. It’s designed to handle huge amount of incoming load and still gives assurances that every event gets processed. Apache Strom gives a flexible structure for applying logic to events. In the architecture of storm, an storm application is called a topology and a topology is a neat abstraction that defines the application logically and physically.

II. LITERATURE SURVEY

Real-time Online Signature Recognition System on Apache Storm with Cancelable Biometrics on Azure Public Cloud Signature- is one of the human trait which is being used for the identification and verification purpose. This paper entitles the importance of online signature recognition system in real-time application. It covers how this signature recognition System is implemented over the open source big data technologies such as Apache Storm and how this technologies can be put on cloud using Windows Azure. It can cope with huge amount of load and massive throughputs. In this proposed architecture a feature called cancellable biometrics is included to provide additional security to the system[8].

Online Signature Recognition using SaaS Model on Public Cloud- This project describes how the signatures will be acquired through the digitizing tablet, and how they will be stored in a blob storage on cloud. The public cloud based model presented in this paper is successfully implemented on Microsoft Azure cloud and aims to devise a highly scalable, pluggable and faster online signature recognition system. This implementation is of high significance in case on online banking and e-commerce, where handwritten dynamic signatures can be used for authentication of transactions[9].

III. PROPOSED WORK

Passwords are like a shared symmetric secret which the user know and the online service to whom the user needs to authenticate knows. And as a user subscribes to more and more online service, he reuses the secret to authenticate itself. Due to this, now it’s not only two who knows the secret, anyone who intercepts the secret, can then reuse into all the online services the user has subscribed. Moreover passwords are easily phish-able and can be easily exploited through malware exploits such as key-loggers. All these issues and many more, make passwords untenable credentialing system. The more robust option for proposed system is biometrics[6]. This paper proposes an a highly secure, scalable, pluggable and faster biometric system architecture, that can handle a billion of biometric events per second that are going to come from the devices, guaranteed delivery and processing of the biometric events, faster enrolment and verification process, fault tolerant & infinitely scalable database, and lastly removing insights from the biometric event. Additionally, this proposed architecture comes the combination of multimodal biometrics and cancellable biometric to add a layer of security over various attacks[7].
Fig 1: System Architecture (Enrollment)

Enrolment:
1. The biometric is captured onto the system using the hardware component such as web cam and fingerprint.
2. This biometric trait which has been captured is then contorted further.
3. The contorted biometric trait is then uploaded to the enroll blob storage.
4. A message with the details of the enrolled user and the container name is sent to the spout through service bus.
5. Then the Service Bus then sends this message to the spout of apache storm which formats the event data.
6. Spout gives the formatted event data to the bolt.
7. The bolt check which type of request is received and forwards the request to the appropriate bolt based on the request received (i.e. enroll or verify bolt).
8. The enroll bolt read the event data and points to the respective blob storage container as specified in the event data.
9. All available trait files are downloaded from the respective blob storage container.
10. Peculiarity (Feature) vector extraction is performed on this trait files.
11. The resulted feature vector files are saved to the Feature Vector blob storage.
12. The event data and outcome of the feature extraction process is sent to the Result bolt.
13. Result bolt sends the message to the service bus.
14. Then service bus sends message to the user.
Fig 2: System Architecture (Verification)

**Verification:**
1. The biometric is captured onto the system using the hardware component such as web cam and fingerprint.
2. This biometric trait is then contorted.
3. The contorted biometric trait is then uploaded to the enroll blob storage.
4. A message with details of the enrolled user and the container name is also sent to the service bus.
5. Then the Service Bus sends this event to the spout of apache storm which formats the event data.
6. Spout gives the formatted event data to the bolt.
7. The verification bolt check which type of request is received and forwards the request to the appropriate bolt based on the request received (i.e. enroll or verify bolt). The verify bolt read the event data and points to the respective blob storage container as specified in the event data.
8. The verification bolt read the event data and points to the respective blob storage container as specified in the event data.
9. Peculiarity (Feature) vector extraction is performed on this live signature file.
10. The verify bolt now points to the Feature vector blob storage with whom the live signature is to be compared as specified in the event data.
11. The respective feature vector files of that user id is downloaded from the Feature Vector blob storage.
12. Both the calculated live feature vector file and the downloaded feature vector file are compared.
13. The Confidence Level (probability value within the specified range of values) and the event data is forwarded to the result bolt.
14. This result is send as a message to the service bus.
15. Service bus queue forwards this message to the client.
IV. EXPERIMENTAL RESULTS

The system implementation is divided into following phases:

Fig. 4 sending message on server.

It is a service bus which is hardware component. This service sends message to the spout in Apache storm to take appropriate action.

Fig. 5 Microsoft Azure message storage.
This is a Microsoft Azure interface which shows you the message which has been received and not has been read. Initially the storage is empty, as the message gets received it shows the number of message that has received.

**Fig. 6 sending message to service bus.**

This interface shows that message has been successfully send on server

**Fig. 7 receiving message on server.**

This is on receiver end which is receiving the message that has been send by the bot.
Fig. 8 notification of approved message.

This is the message which is received by the server.

V. CONCLUSION AND FUTURE WORK

In this system, we presented a biometric authentication which can be used for various devices and software’s. The environment for this system is user friendly and simple to get register. The system can store n number of users. The system provides high security to data. The system keeps the user credentials classified. Service can be accessible from anywhere through an internet. The system works on Azure cloud in which it uses Apache storm for processing of data. Cloud services such as blobs are used for storing the data. The data is the biometric imprints of individual

REFERENCES

VIII. Biometric Authentication using Software as a Service (SaaS) Architecture with Real-time Insights
IX. Mr. Godson Michael D’silva Information Technology Department, St. John College of Engineering & Technology, Palghar, India, Dr. Vinayak Ashok Bharadi Information Technology Department, Thakur College of Engineering & Technology, Mumbai, India, Information Technology Department, Mr. Shridhar Kamble Thakur College of Engineering & Technology, Mumbai, India.
X. Real-time Online Signature Recognition System on Apache Storm with Cancelable Biometrics on Azure Public Cloud
   Mr. Godson Michael D’silva, Mr. Sagar Kumar, Mr. Urvesh Patel, Mr. Cleon Barretto Information Technology
XI. Real world smart chatbot for customer care using a software as a service (SAAS) Architecture
XII. Mr. Godson Michael D’silva, Mr. sanket thakare, Ms. Shraddha more, Mr, Jeril kuriakose 2017 International Conference on I-SMAC (Iot in Social, Mobile, Analytics, and Cloud) (I-SMAC) Year 2017, Accessed on 10.01.2018