Preventing Ddos Attack Using Sf with Thumb Impression

V.Shyamaladevi[1], Dr.R.Umarani[2], M.Aishwarya[3]
[1]Associate Professor, Department of MCA, KSR CT, Tiruchengode, Tamilnadu, India
[2]Professor, Department of MCA, Saradha Womens, Salem, Tamilnadu, India
[3]Research Scholar MCA, Tamilnadu, India
[1]shyamalamanikandan@gmail.com, [2]umaninweb@gmail.com, [3]aishwaryammb@ymail.com

***Abstract* : The theoretical performance of linear and nonlinear collusion attacks under the assumptions that orthogonal or regular-simplex fingerprints are used, and that the detector performs a linear correlation test in order to decide whether a user of interest is among the colluders. The colluders create a noise-free forgery by applying a mapping *f* to their individual copies, and then add a noise sequence e to form the actual forgery. They seek the mapping *f* and the distribution of *e* that maximize the probability of error of the detector. The performance of mappings such as linear-averaging and interleaving can be compared in this framework. It is also shown that impulsive noise attacks are far more effective than Gaussian attacks. Digital fingerprinting schemes are devised for traitor tracing. In applications such as copyright protection, the goal is to deter users from illegally redistributing the digital content. Each user is provided with his own individually marked copy of the content. Although this makes it possible to trace an illegal copy to a traitor, it also allows for users to collude and form a stronger attack. One form of such attacks is linear averaging, where the colluders average their copies and add noise to create a forgery. Averaging reduces the power of each fingerprint and makes the detector's task harder. Another collaborative attack is interleaving, where the colluders form a pre-forgery by contributing samples from their copies and contaminating the pre-forgery with noise.**

***Keywords* : colluders, actual forgery.**

I INTRODUCTION

A distributed denial-of-service (DDoS) attack is one in which a multitude of compromised systems attack a single target, thereby causing denial of service for users of the targeted system. The flood of incoming messages to the target system essentially forces it to shut down, thereby denying service to the system to legitimate users.In a typical DDoS attack, a hacker (or, if you prefer, cracker) begins by exploiting a vulnerability in one computer system and making it the DDoS master. It is from the master system that the intruder identifies and communicates with other systems that can be compromised. The intruder loads cracking tools available on the Internet on multiple sometimes thousands of compromised systems. With a single command, the intruder instructs the controlled machines to launch one of many flood attacks against a specified target. The inundation of packets to the target causes a denial of service. The target of DDoS attacks as the victim, in reality there are many victims in a DDoS attack -- the final target and as well the systems controlled by the intruder. Although the owners of co-opted computers are typically unaware that their computers have been compromised, they are nevertheless likely to suffer degradation of service and malfunction. Both owners and users of targeted sites are affected by a denial of service. A computer under the control of an intruder is known as a zombie or bot. A group of co-opted computers is known as a botnet or a zombie army. Both Kaspersky Labs and Symantec have identified botnets -- not spam, viruses, or worms -- as the biggest threat to Internet security.

1.1 Preventing DDoS Attacks

The DDoS problem can only be remedied by a community effort and stricter security standards. First, administrators and home users alike need to make sure their machines are secure. The slaves used in DDoS attacks are often the product of auto rooters, programs which scan thousands of machines, crack vulnerable ones and install software. Keeping patches up to date, closing open services, and implementing basic firewall filtering can help keep your machines from falling prey and participating in such an attack. The major difficulty in defeating a DDoS lies in the spoofed IP addresses of the attackers. This problem can be solved using a technique called ingress filtering on routers. Ingress filtering inspects packets destined for the Internet at the border router, one hop prior to the core router. These routers should know the address of every device behind them; therefore, anything outside of this range is spoofed. Spoofed packets should be dropped before they reach the Internet backbone (or core router). If network administrators implemented such filtering by default, spoofing a packet would become nearly impossible, eliminating the timely identification process in the DDoS investigation. Unfortunately, most networks do not have these crucial filters in place, and spoofed packets abound. IPv6, which will be deployed in the future, also has security features in place that address this fundamental networking problem. The community also recognizes the difficulty of reaching the proper technical contacts on neighboring networks and is actively working on a solution. We should have in place a list of administrative and technical contacts at your ISP.

Additionally, determine if they have a procedure in place for identifying and dealing with DDoS attacks on their own backbone network. Some of the major providers have sensors in place that can identify sudden increases in traffic at certain points, which serves as a useful alarm for discovering and isolating major DDoS incidents. If you're currently shopping for an access provider, ask them about dealing with DoS attacks.

2 RELATED WORKS

2.1 AN AUTOMATIC IDENTITY AUTHENTICATION

An Identity Authentication System Using Fingerprints submitted by Anil Jain, Lin Hong, Sharath pankanti and Ruud Bolle at department of computer science Michigan State University.To introduce a prototype automatic identity authentication system is capable of automatically authenticating the identity of an individual using fingerprints. The architecture of the automatic identity authentication system is shown in Figure 2.4. It consists of four components: user interface, system database, enrollment module, and authentication module. The task of enrollment module is to enroll persons and their fingerprints into the system database. If a fingerprint image is of poor quality, it is enhanced to improve the clarity of ridge/valley structures and mask out all the regions that cannot be reliably recovered. The enhanced fingerprint image is fed to the minutiae extractor again. Because the current quality checking algorithm is very slow, it is only used in the enrollment module. The task of authentication module is to authenticate the identity of the person who intends to access the system. The person to be authenticated indicates his/her identity and places his/her finger on the fingerprint scanner; a digital image of his/her fingerprint is captured; minutiae pattern is extracted from the captured fingerprint image and fed to a matching algorithm which matches it against the person's minutiae templates stored in the system database to establish the identity.

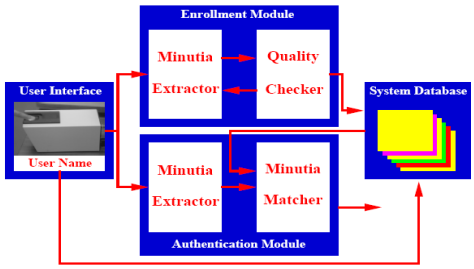


Fig. 2.1 Architecture of the Automatic Identity Authentication System.

2.2 FINGER PRINTS RECOGNITIONS SYSTEM

There are many types of biometric systems commercially available such as fingerprints, iris/retina and hand shape devices. Each of these systems has merits and demerits. In the case of fingerprints, direct contact of the finger with the fingerprint-image-extracting sensor causes degradation in performance, especially in factory construction sites where good-quality fingerprints are hard to obtain due to oil from the finger, moisture, dirt, etc.

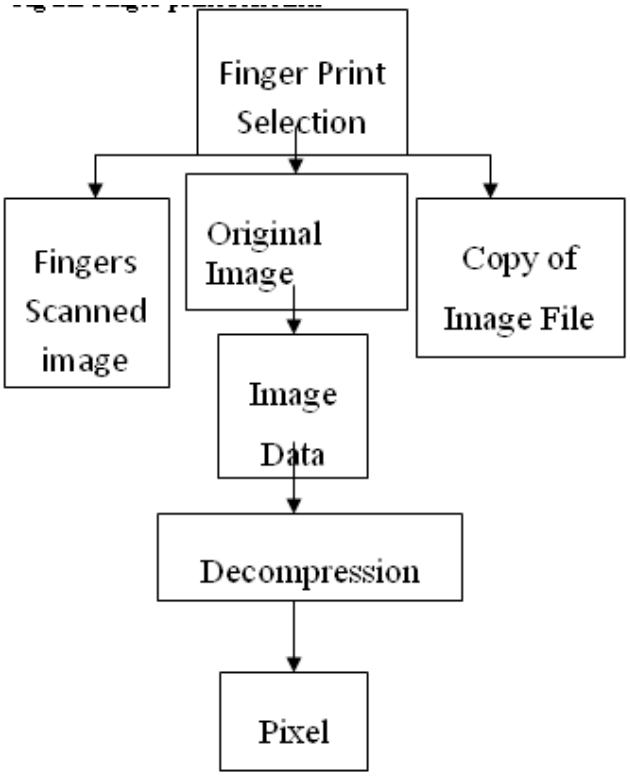
Fingerprinting: Past & Present

First recognized in 1864 as a unique identifier, fingerprint identification is the oldest and most reliable biometric technique. Widely used and accepted in most places as evidence, fingerprints have long been associated with the law enforcement community. The FBI alone has more than 70 million fingerprints on file, most of which were taken using the older inkpad and paper method. This required a human to sit down with a magnifying glass for hours if not days comparing fingerprints. Within the last 10 years, though, they have stopped keeping paper records and have moved to capturing fingerprint data digitally, allowing a computer to sort through hundreds of print an hour with a human doing only the final examinations. Every fingerprint can be broken down into two basic features, called ridges and valleys. By examining these characteristics, it is possible to extract data from raw fingerprints and store it in a computer database for future comparisons. Images can be captured using one of several devices, including: Optical Scanners

- Thermal Scanners
- Capacitive (Solid-State) Scanners

There are currently two accepted methods for extracting this data: minutia-based and correlation-based. Minutia-based is the more microscopic of the two, locating ridge branches and endings and assigning them an XY-coordinate that is then stored in a file. Correlation-based looks into the overall pattern of ridges and valleys. Instead of looking for tiny minutia points, the locations of whorls, loops and arches and the directions that they flow in are extracted and stored on the other hand, correlation-based comparisons can be affected by image translation and rotation. After the initial setup of an authorized user, every time he or she wants to access the system, the fingerprint is run through the same algorithms used when it was stored. This data set is compared to the original data set on file, and then it is either accepted or rejected. Most authentication systems grapple with this False Rejection Rate/False Acception Rate (FRR/FAR). Simply put, authorized users shouldn't be rejected and unauthorized users shouldn't be accepted. No system has been created that is 100% accurate.

Fig 2.2 Finger print selection

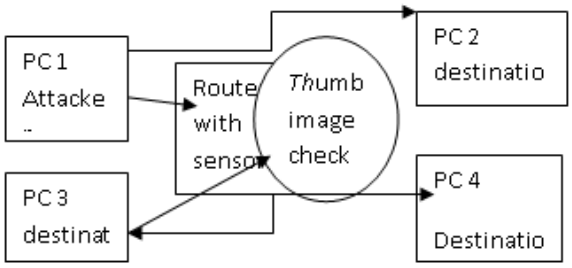


These special sensors can be either deployed independently or the different functionalities can be included in the same sensor nodes. Even data reading and reporting can be generated from these sensors at different rates, subject to diverse quality of service constraints, and can follow multiple data reporting models. For example, hierarchical protocols designate a cluster- head node different from the normal sensors. These cluster heads can be chosen from the deployed sensors or can be more powerful than other sensor nodes in terms of energy, bandwidth, and memory. Hence, the burden of transmission to the BS is handled by the set of cluster-heads.

2.3 Fault Tolerance

Some sensor nodes may fail or be blocked due to lack of power, physical damage, or environmental interference. The failure of sensor nodes should not affect the overall task of the sensor network. If many nodes fail, MAC and routing protocols must accommodate formation of new links and routes to the data collection base stations. This may require actively adjusting transmit powers and signaling rates on the existing links to reduce energy consumption, or rerouting packets through regions of the network where more energy is available. Therefore, multiple levels of redundancy may be needed in a fault-tolerant sensor network.

Fig 2.3 Architecture of router performance



3 PERFORMANCE ANALYSES AND SIMULATION RESULTS

The system produces intermediate results. Encoded documents are applied into the decoding process with the same key and also the module is tested with different finger print values. The testing results show that the system is a reliable one. The system provides a high level security for the documents and also for the key. The decoding process results show that the system returns the original content of the encoded file without any loss.

a. Analysis

The system is tested with different number of inputs. The results of this system are analyzed with the consideration of the following factors. They are time, file type and size, instant key generation process, security and authentication.

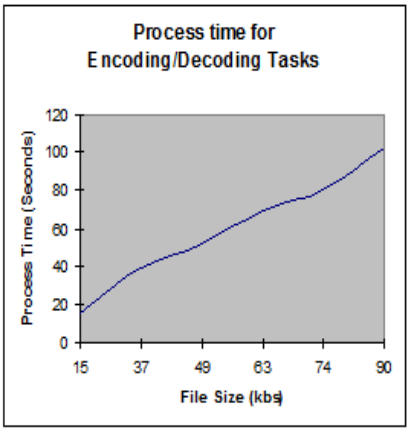
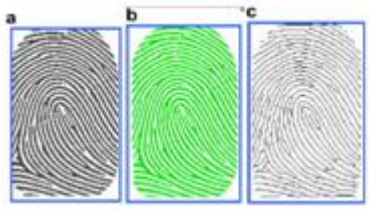
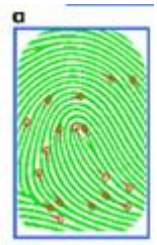


FIG 3.1 process time for the tasks

The duration for the encryption and decryption process are measured and compared. The encryption and the decryption tasks take the same amount of time. The key generation time is very negligible one. The time that can be taken for the encryption and the decryption process is depends upon the input file size. If the file size is increased then the process time is also increased. The key generation time is almost same for any finger print image. The hardware configuration may affect the execution time.

Copy of Orginal images





pixel transforming

4 FUTURE SCOPE DEVELOPMENT

Every application has its own merits and demerits. The project has covered almost all the requirements. Further requirements and improvements can easily be done since the coding is mainly structured or modular in nature. Changing the existing modules or adding new modules can append improvements. Further enhancements can be made to the application, so that the web site functions very attractive and useful manner than the present one.

5 CONCLUSIONS

It is concluded that the application works well and satisfy the administrator and service engineers. The application is tested very well and errors are properly debugged. The site is simultaneously accessed from more than one system. Simultaneous login from more than one place is tested. The site works according to the restrictions provided in their respective browsers. Further enhancements can be made to the application, so that the web site functions very attractive and useful manner than the present one. The speed of the transactions become more enough now. The project is fully user friendly one. So any one of the staff members with little knowledge of computer can handle the software and all the requirements for the user are fully completed.

REFERENCES

[1] H.J. Chao, "Next Generation Routers," Proc. IEEE, vol. 90, no. 9, pp. 1518-1558, Sept. 2002.

[2] M.A. Ruiz-Sanchex, E.W. Biersack, and W. Dabbous, "Survey and Taxonomy of IP Address Lookup Algorithms," IEEE Network, vol. 15, no. 2, pp. 8-23, Mar./Apr. 2001.

[3] G. Varghese, Network Algorithmics: An Interdisciplinary Approach to Designing Fast Networked Devices. Morgan Kaufmann Publishers/ Elsevier, Inc., 2005.

[4] V.C. Ravikumar, R.N. Mahapatra, and L.N. Bhuyan, "EaseCAM: An Energy and Storage Efficient TCAM-Based Router Architecture for IP Lookup," IEEE Trans. Computers, vol. 54, no. 5, pp. 521- 533, May. 2005.

[5] D. Shah and P. Gupta, "Fast Updating Algorithms for TCAM," IEEE Micro, vol. 21, no. 1, pp. 36-47, Jan./Feb. 2001.

[6] D. Mehta and S. Sahni, Handbook of Data Structures and Applications. Chapman and HALL/CRC, 2005.

[7] H. Song, J. Turner, and J. Lockwood, "Shape Shifting Tries for Faster IP Route Lookup," Proc. IEEE Int'l Conf. Network Protocols (ICNP), 2005.

[8] W. Eatherton, "Fast IP Lookup Using Tree Bitmap," master's thesis, Washington Univ., 1999.

[9] B. Lampson, V. Srinivasan, and G. Varghese, "IP Lookups Using Multiway and Multicolumn Search," IEEE/ACM Trans. Networking, vol. 7, no. 3, pp. 324-334, June 1999.

[10] X. Sun and Y. Zhao, "An On-Chip IP Address Lookup Algorithm,"IEEE Trans.Computers, vol. 54, no. 7, pp. 873-885, July 2005.