Decision Mechanism of Password Recovery Based on Bloom Filter in Heterogeneous Platform

Quanquan Xie[1, a] , Qinglei Zhou[1] , Xueming Si[2] , Bin Li[2]

[1]College of Information Engineering, Zhengzhou University, Henan Zhengzhou 450000, China;

[2]College of Information Engineering, the PLA Information Engineering University, Henan Zhengzhou 450000, China.

[a]736053294@qq.com

Keywords: password recovery, bloom filter, heterogeneous platform, network security.

Abstract. According to the password recovery task's computation and local communication intensive features, we design a decision mechanism of password recovery based on bloom filter in the CPU+FPGA system architecture. Utilizing the advantages of the bloom filter in the searching of massive data, bloom filter uses the generated hash vector to filter the feature string and the result becomes the basis of crack mode decision before the start of the task. At the same time, it can accurately position where the password dictionary file is. The experiment results show that this method takes up smaller time of decision and reduces the huge overhead of dictionary transmission and mode switching. Compared with the traditional mode, this method can improve the efficiency of the whole task.

1. Introduction

Password authentication mechanism is an important means to ensure the security of file. But Encryption file will bring inconvenience to the National Security Department of the investigation and evidence collection work. In this case, the password recovery technology appears. However existing computational structure and calculation ability is unable to meet the needs of encrypted document's password recovery. Take the Office 2010 document as an example, exhaustive searching all passwords which length is 6, character set size is 62 (contains upper and lowercase letters and numbers), we need approximately 438 days in the HPC [1]. The calculated efficiency is unable to meet the user's demand. Due to the realization of the principle of FPGA, CPU+FPGA heterogeneous computing system can play a very good performance for computing intensive tasks [2, 3], such as password recovery.

In this paper, we use the filter bloom to express a dictionary, and query whether a password belongs to the dictionary. And we choose the dictionary mode or exhaustive mode according to the query results. In this way, we can achieve maximize the efficiency of the password recovery application in the CPU + FPGA heterogeneous platform.

2. Decision Mechanism of Password Recovery

2.1 Structure of Password Recovery System.

According to the characteristics of password recovery task and the existing resources, we design the password recovery application on four parts: communication module, password generation module, decision module and computing module. The overall structure design of password recovery system is shown in Fig 1.

Communication module: This module is used to transmit the task information, control signal and result. The server is on general PC. And the client needs to install on the FPGA board to obtain the real-time state.

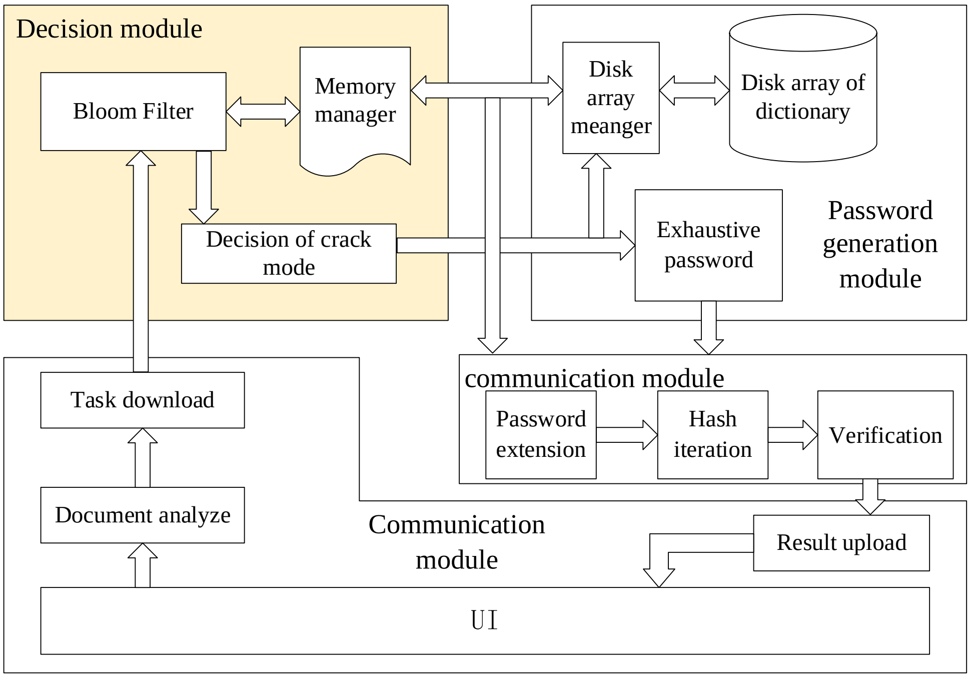

Fig. 1 Structure of password recovery system

Password generation module: when the crack mode is dictionary mode, the password is read from disk array. When the crack mode is exhaustive mode, the password is generated by exhaustive module according to the character set, the length of password and other information of password. The dictionary file is mount to the disk array server. And the exhaustive module is put on FPGA to improve the password throughput.

Decision module: this module will be described in detail in the 2.2 section.

Computing module: according to the different types of documents and crack model, this module will load different crack algorithm. The crack process is similar, including three parts of the password extension, hash iteration, verification string comparison. These processes are the logic operation and complex. So we puts it on the FPGA which is good at high speed logic operation

2.2 Design of decision module based on filter Bloom.

Bloom filter proposed by Howard Bloom is an efficient data structure [4, 5, 6]. In the decision module we use bloom filter BF=<R. M, H>. R= $\{r_1, r_2, \ldots, r_n\}$express each dictionary file, and r_i is a password's feature string in dictionary file. M is a bit vector of length m. And M[0], M[1], ... , M[m-1]is used to store the hash value mapped by elements r_i. H is the set of hash functions which range is (0, m-1). H=$\{h_1(), h_2(), \ldots, h_k()\}$, and $h_1(), h_2(), \ldots, h_k()$ are mutually independent.

First, we initialize the vector M to 0. Then, read every element r_i from R and calculate the hash values $h_1(r_i)$, $h_2(r_i)$, ..., $h_k(r_i)$. Search address in the M according to the hash value, and make $M[h_1(r_i)]$, $M[h_2(r_i)]$, ... , $M[h_k(r_i)]$ do OR operation with 1 respectively. Finally, store the vector M in the disk array as an index of the dictionary file. When the task feature string is sent to the decision module, the decision module read the vector M mapped by dictionary file from the disk array server. Then, the BF uses the hash function set H to compute the feature string when. If the result hits the vector M, the crack mode is selected as the dictionary mode, and send the dictionary file number to computing module. If misses, BF read the next vector M`, until all the vectors are not hit. Then we consider disk array does not contain feature string of this password. So we are unable to use the dictionary crack mode, and only choose exhaustive mode.

But the value of the hash function may generate a collision. If all bits in vector M mapped by an element are 1, in fact, we still can't identify this element must belong to the set R. This kind of error division is called false positive [7]. But a certain range of false positive is acceptable for the password recovery task. The false positive is calculated as:

$$f = \left(1 - \left(1 - \frac{1}{m}\right)^{kn}\right)^{k} \approx \left(1 - e^{\frac{kn}{m}}\right)^{k} \quad (1)$$

In order to improve the adaptability of this method, we update the dictionary file by the results of exhaustive mode to improve the efficiency of future tasks.

3. Experiment and Analysis

3.1 Design of Experiment.

In experiment, we use three IBM x3650 with 24G memory and Gigabit LAN, one of them is used as host computer, another is used as disk array server, and the last is used as the CPU computing node. And we use two sets of Xilinx Virtex-6 LX550T [8] as accelerator with 16G memory. The dictionary file contain the common password and some special combinations, such as birthday, name and phone number. And we use 400 SHA1 [9, 10, 11] encryption document as crack task. The first 200 document's password is in the dictionary file, while the last 200 document's password is not. The password character set space is 62 (including the size of the letters and numbers). And the length of the password is in the range of 3-7 bit and obeys the normal distribution, because password bellow 3 bits are cracked in less than one second.

The relevant parameters in the BF are chosen as follow: suppose k= 6 and randomly select 6 hash functions, of which hash function set is H= {RSHash, HFHash, JSHash, PJWHash, elfhash, BKDRHash}, vector length is 20 times the number of entries in dictionary file. We can calculate the false positive f=0.0303% in this experiment by the formula 1. At the end of experiment, we compare the average time of only exhaustive mode, exhaustive mode with dictionary and decision mechanism based on Bloom filter.

3.2 Result Analysis.

Assume that the crack rate is v and the password length is n. The completion time of crack task fluctuates between $(62+62^2+\ldots+62^{n-1})/v$ and $(62+62^2+\ldots+62^{n-1}+62^n)/v$. With the increase of the password length, the crack time is exponential growth. Compared to the exhaustive mode, if the password is in the dictionary, exhaustive mode with dictionary can significantly improve the speed. But when the password cannot be broken with a dictionary mode, the decision module need to read all dictionaries and switch to the exhaustive mode from the dictionary mode. The FPGA needs about 2-3 minutes to download and configure the corresponding. So if the password is not in the dictionary, this model will increase the time. But decision mechanism based on Bloom filter can effectively reduce the time of communication and mode conversion.

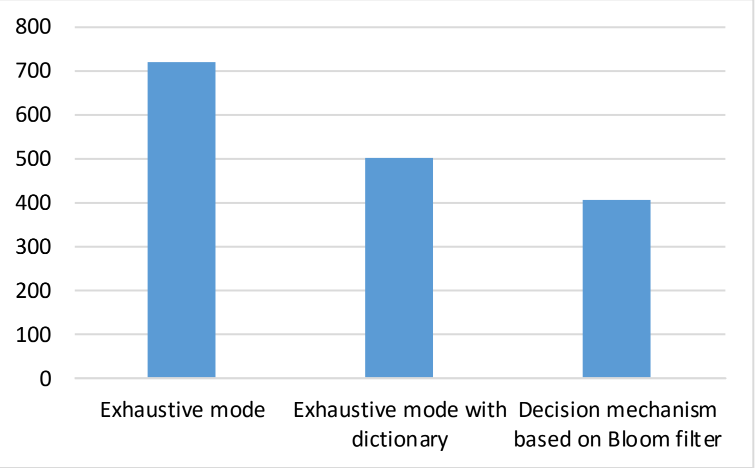

Fig. 2 The average crack time of three modes

The average crack time of three modes for 400 documents is shown in Fig 2. The experiment results show that the decision mechanism based on Bloom filter can significantly reduce the average time of crack tasks. Compared to violence exhaustive mode, the efficiency of this method increased by about 45%. Compared to the exhaustive mode with dictionary, the efficiency of this method is increased by about 20%. When the false positive occurs, this method will degenerate to the exhaustive mode with dictionary. But as long as the false positive is controlled at low level, it will not have a great impact on the system's overall efficiency.

4. Summary

In this paper we use the advantage of bloom filter to optimize the decision mechanism of password recovery task. Compared with the traditional model, this method can not only give full play to the advantages of dictionary model, but also reduce the time of dictionary transmission communication and mode conversion. Experiment shows that this method can greatly improve the efficiency when the task is continuous, and when the scale of task is larger, the more obvious advantages. However, this method also has some limitations and shortcomings. First, this method can only be used for encryption algorithm without salt processing, such as SHA-1 and MD5. Because salt value is random, it will get different feature string, even password is same. So this method is unable to make correct choice. Second, the bit vector mapped by each dictionary file occupies a part of the storage space. Although it can be optimized by reducing the length of the bit vector, but this will increase the false positive. So it is necessary to make a balance in time and space to get the overall best results.

References

[1]. Li Xiaoxiao: Research on file password recovery technology on high performance computing platform (Master, Beijing University of Posts and Telecommunications, china, 2014). p.2.

[2]. Li Longpu, Si Xueming, Zhang Zhihong, et al. Implementing dictionary based password reconvert for ZIP documents on multi-FPGA. Computer Applications and Software. Vol.32 (2015) No. 6, p. 292-295.

[3]. Chen Yijiao, Lu Zexin, Sun Zhigang. Implementation research of reconfigurable hardware based-on FPGA. Journal of Information Engineering University. Vol.10 (2009) No. 1, p. 94-98.

[4]. Burton H. Bloom. Space/Time Trade-offs in Hash Coding with Allowable Errors .Communications of the ACM. Vol.13 (1970) No. 7, p. 422-426.

[5]. Shahabeddin Geravand, Mahmood Ahmadi. Bloom filter applications in network security: A state-of-the-art survey. Computer Networks. Vol.57 (2013) No. 18, p. 4047-4064.

[6]. Zhao Qian, Cui Yimin, Zou Tao. Research of Bloom filter application in network forensics. Computer Engineering and Applications. Vol.46 (2010) No. 3, p. 91-94.

[7]. Yan Huayun, Guan Jihong. Survey of Bloom filter. Telecommunications Science. Vol.26 (2010) No. 2, p. 139-142.

[8]. Information on: http://www.xilinx.com/support/documentation/data_sheets/ds150.pdf

[9]. Zhang Bin, Xu Ming-yang. SHA-1 and application in FPGA encrypted authentication system. China Integrated Circult. Vol.20 (2011) No. 6, p. 57-61.

[10]. Han Jinsheng, Lin Jiajun, Zhou Wenjin, et al. Design of Linux password high-speed crack mode on FPGA. Journal of Chongqing University. Vol.35 (2012) No. 8, p. 42-47.

[11]. Tyler Blaine Johnson: Phillip H.Jomes,et al. An FPGA Architecture for the Recovery of WPA/WPA2 Keys (Master, Iowa State University, the USA, 2014). p.36.