IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 4, April 2017

Performance Analysis on Different Permutation Algorithms for Security Applications

Nilesh Bhavsar[1], Tejas Shinde[1], Satish Narwade[1], Aarti Shinde[1]

Student, Computer Engineering, Dr. D. Y. Patil Institute of Technology, Pimpri, Pune, India[1]

Abstract: Permutation is different arrangements that can be made with a given number of things taking some or all of them at a time. The notation P(n,r) is used to denote the number of permutations of n things taken r at a time. Permutation used in the pattern analysis, Databases and data mining, simulation, accumulation of electronic communication data, Homeland security, wireless networks. Bottom-Up, Lexicography, Johnson-Trotter, Back-tracking, Heap algorithm, Brute-Force are of the most popular permutation algorithms that emerged during the past decades. Lexicography used in graph colouring. Backtracking are used in Sudoku. A performance analysis in terms of time complexity is done by implementing algorithms in different programming languages on different platforms. In cryptography, cipher text is the result of encryption performed on plain text using an algorithm called cipher which is used in client-server application.

Keywords: Permutation, Time complexity, Pseudo Random generator, Authentication, Passkey.

I. INTRODUCTION

In modern era, organizations greatly rely on computer networks to share information throughout the organization in an Efficient and productive manner. Organizational computer networks becoming large and ubiquitous. Assuming that each staff member has a dedicated workstation, a large scale company would have few thousands workstations and many server on the network. These workstations may not be centrally managed, nor would they have perimeter protection. They may have a variety of operating systems, hardware, software, and protocols, with completely different level of cyber awareness among users. Now imagine, these thousands of workstations on company network are directly connected to the Internet. This sort of unsecured network becomes a target for an attack which holds valuable information and displays vulnerabilities. There are numerous network security and access control issues to be considered when implementing a client/server system. Because the server is usually the central location for critical data, adequate physical security and operational security measures need to be taken to insure the safety of the data. Although there are a large number of tools available to perform security and control functions on mainframe systems, there are significantly less tools available that are designed specifically for client/server systems. Until these tools are developed, companies must exercise extreme caution when placing mission critical applications on a client/server system. We execute different permutation algorithm on various platform such as windows, Ubuntu, android etc. and different languages likes C, C++, Java etc. for comparative study of permutation algorithm. And we find efficient algorithm from several algorithm such as lexicography, Heap algorithm, Brute-Force, Johnson Trotter, Backtracking etc. from their time complexity for executing algorithm and space complexity for storing generated permutation. Creating permutation using variations likes substring, Pseudo-random number generation. A condition is found for getting permutations with the property that every object abandons its initial position. Since cryptography is among the possible applications, the generator performing the inverse transformation is also given. Substring divide string into different Subset and generate permutation.

In the process of encoding, the encoder uses words, number symbols for which he believes the decoder will understand. The symbols can be words and number. It is very important how a message will be encoded; it partially depends on the purpose of the message.

The decoding of a message is how able to understand, and interpret the message. It is a process of interpretation and translation of coded information into a comprehensible form. The receiver is trying to reconstruct the idea by giving meanings to symbols and by interpreting the message as a whole. Effective communication is accomplished only when the message is received and understood by both party. However, it is still possible for the message recipient to understand a message in a completely different way from what was the encoder was making an attempt to convey. This can be once "distortions" or "misunderstanding" arise from "lack of equivalence" between the two sides in communicative exchange.

In client-server model, client uses ID to define he is member of client-server model.

Client sends ID to server for validation. Server generates permutation for encoding passkey and then sends it to the client. Client also generating permutation for decoding passkey. If passkey matches then server authenticate to client and connection established between client and server.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 4, April 2017

II. LITERATURE SURVEY

Random permutation generation algorithms namely, Fisher-Yates algorithm, its variant Sattolo algorithm and Josephus algorithm are discussed. Both chaos and non-chaos based approaches have been analysed and compared. Their efficiency has been tested against three criteria: no ideally fixed points, no un-separated pairs and shift factor greater than one third the degree of permutation. A random permutation algorithm should successfully satisfy the above criteria. The OTP is typically generated by a token possessed by the user and is input to an authentication system. The input OTP is compared to an OTP generated by the system using the same information and encryption algorithm as is used by the token. If the input OTP matches the OTP generated at by the system, the user is allowed access to the system. Sedgewick summarizes a number of algorithms for generating permutations, and identifies the minimum change permutation algorithm of Heap to be generally the fastest .Another method of enumerating permutations was given by Johnson.

III. PROPOSED SYSTEM

In this model we are using the concept of permutation for generating a passkey. Client having unique Identification (ID) and this ID's are already stored in server database. The passkey generated on server side and send to client. Server send generated passkey only intended ID. Client also generates permutation for matching passkey. If ID is not valid or not stored at Server database then server will not generate passkey.
When client matches passkey with received passkey from server then authentication successful and secure connection established between client and server. Through this way we use permutation toprovide secure connection between client-server models that cannot be accessed by intruder.

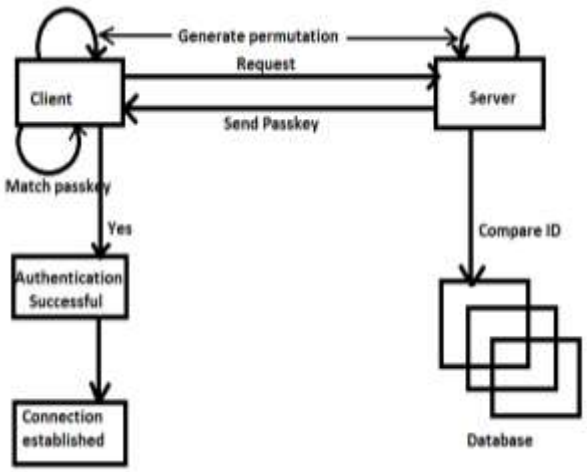

Fig. 1. Block Diagram of System

Block Diagram consists of a client through login page it enters ID and send that ID to Server. Then server will match that ID with existing database if match found then server itself generate permutation and send generated passkey to client. For matching the same passkey client generates permutation and match passkey. If both passkey matches with each other, then connection established.

Application of the proposed system:

- In network devices for validating node.
- Client server model for authentication of a client to established secured connection between them.

VI. METHODS

Algorithm checks whether or not a mobile element exists or not, if yes the algorithm performs the following:
1. Find the largest mobile element k
2. Swap k and the adjacent element
3. Reverse the direction of larger than k As long as there exists a mobile repeat all the above.

The straight-forward algorithm for generating permutations is a typical decrease-by-one algorithm:
a. find all permutations of size n-1 of elements a1, a2, .., an-1

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 4, April 2017

b. Construct permutations of n elements as:
i. append an to each permutation of size n-1
ii. for each permutation of size n-1 for k from 1 to n-1 insert an in front of ak

The algorithm works with directed numbers - each numbers has directions – left or right. A directed integer is said to be mobile if it is larger than its immediate neighbour in the direction it is looking at. Directed number can be represented by a structure that contains the number and its orientation.
For this we used swapping technique to swap given number for permutation. Generate permutations of n items in which successive permutations differ from each other by the swapping of any two items.

Pseudo-Code
// ALGORITHM
// performs permutation using algorithm
// INPUT: input (array of integers)
// OUTPUT: list (array of integers)

The algorithm contains two nested Loops whose bodies are executed consecutively n!-1 times where n is the total length of the input, and n-1 wherever n is that the total length of the input. Ignoring the instructions outside the loop and taking into consideration the most costly instruction into program as the basic operation as follow:

$$\sum_{k=0}^{n!-1} \sum_{x=0}^{n-1} 1 = n\,(n!) = n!$$

And algorithm program is of time complexity O (n!) timesno matterthe worth of the input, we have a tendency to$C_{Best}(n!) = C_{Worst}(n!) = C_{Average}(n!) = n!$
The complexity of this algorithm is O(n!) where n is the length of input to permute. We have two loops and one basic operation executed n! times over n elements.

Implementation:
The six permutation algorithms were all implemented using C, C++ and Java under eclipse, JDBC on Ubuntu, Windows, AIDE android platform.

IV. FLOWCHART

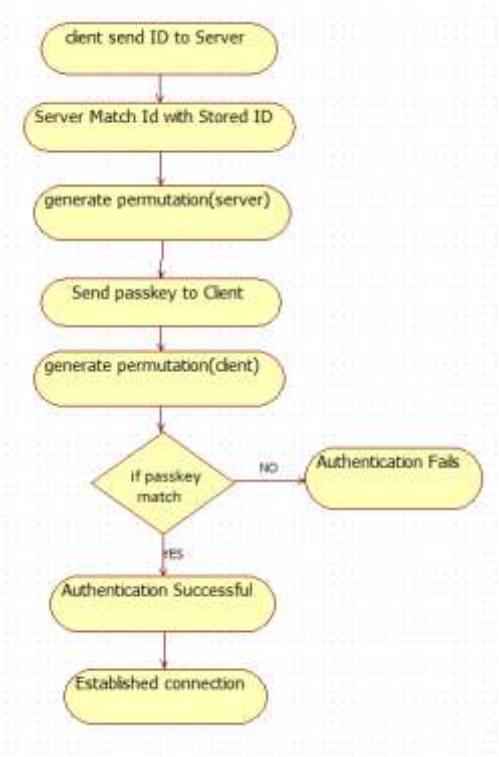

Fig. 2. Flow chart of system

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 4, April 2017

V. TESTING & EXPERIMENTS

A comparison of the execution time between the six permutation algorithms was undertaken using a laptop with Intel i3 dual core processor with 2GB of RAM. The operating system used was Windows ten, Ubuntu, AIDE Android platform. It is worth noting that the execution time for all different algorithms is the average time obtained after five consecutive runs of the same test. Table delineates the execution time of the permutation algorithms using the Johnson Trotter method.
The table shows the time complexity of different generated permutation on different platform which is clearly shows Ubuntu operating system with C++ language is efficiently run generate permutation using minimum time compare to other ones.

Table 1: Results in second run on different platform.

I/P	C	C++	Java		
I/P	Ubuntu		Windows 10	Ubuntu	AIDE
3	0.001	0.0001	0.0001	0.001	
4	0.002	0.0012	0.005	0.002	
5	0.003	0.01	0.012	0.014	
6	0.005	0.07	0.127	0.035	
7	0.02	0.44	0.529	0.39	
8	0.17	2.7	2.12	3.21	
9	1.41	22.72	20.41	31.12	
10	14.06	241	209.85	305.56	

An authentication scheme can be considered secure if during the authentication process no sensitive data like username etc. is exchanged publicly i.e. without encryption, only the right or legitimate user is given access to an account, or data while transmit is practically impossible to be spoofed.

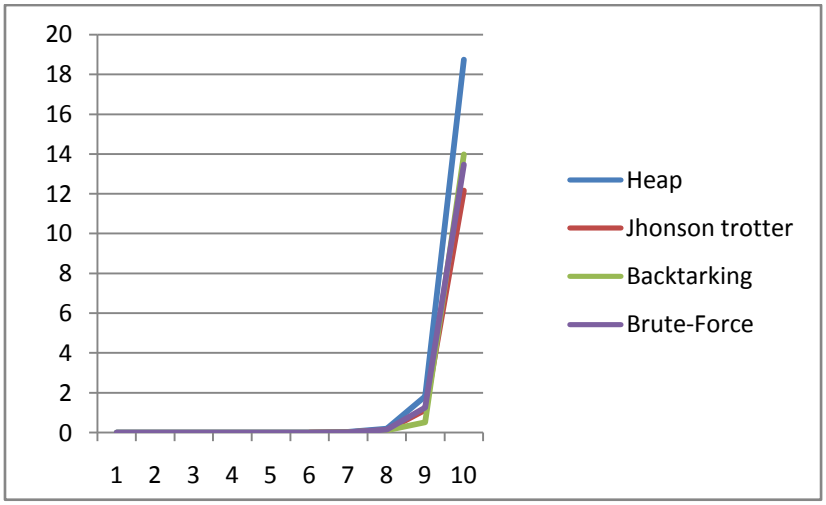

Fig. 2. Sample graph of time complexity of different algorithms

.

Thus, keeping all these above points in mind; the resulting prototype is named as Robust Login Authentication using permutation through secure transmission. This is generates permutation at the server side and sent to client side. Passkey is generate value at both the ends. i.e. client and server end, then authentication only occur if the codes generated on both sides are same. Everything that travels along the network is matches passkey at both sides if passkey are matches then connection will be established. There are only two entities that are involved in this authentication mechanism: a server that accepts all the request and response corresponding and second is client which is an user that is used to login into the user account. In the first step, the user is required to pass ID to the system, and then using permutation generate passkey to server. After successful passkey matches on both sides, then connection will be established and its authorized connection.
Login
As the user sign up his account, all this detailed are stored on server along with his unique ID. Then, whenever user wants to login into his account the id filled during sign up will be considered as the username and password is

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 4, April 2017

previously chosen by the user. Password chosen by the user should be complex enough so that cannot be easily guessed by anyone. For logging into the account, user enters his login ID or username and password and when the click on the login button, its enter into system and client generate passkey to server, server also generated permutation when its matches then authorized connection will be established.

On server side, after receiving the passkey, firstly it will be decrypted and then server checks it into the table whether the password corresponding to username is correct, if it is correct, then the ID which is stored corresponding to that username is verified with the ID number of passkey from which the login request came. If both passkey number matches and the respective unique identification number are also same, then the authentication occur.

VI. CONCLUSION

Permutation used in the pattern analysis, databases and data mining, simulation, accumulation of electronic communication data, homeland security, wireless networks. A performance analysis in terms of time complexity is calculating by implementing algorithms in different programming languages on different platforms. The best algorithm is use for encoding the passkey to secure the client-server model for secure connection establishment which cannot be accessed by intruders.

REFERENCES

[1] Heap, Permutations by interchanges, Computer Journal, Knuth, The Art of Computer Programming, vol. 4 sec. 7.2.1.1 www-cs-faculty.stanford.edu/ knuth/taocp.html
[2] Ord-Smith, Generation of permutation sequences,Computer Journal, 1970-71 Sedgewick, Permutation Generation Methods, Computing Surveys, 1977
[3] Journal of Computer Science & Research (JCSCR) -SSN 2227-328X http://www.jcscr.com Vol. 1, No. 1, Pages.7-19,February 2012
[4] "Permutation Generation Methods" ROBERT SEDGEWlCK Program and Computer Science and Dwlsmn of Applled Mathematics.