Public Key Cryptography Using Genetic Algorithm

Swati Mishra, Siddharth Bali

Abstract*—Cryptography is an imperative tool for protecting and securing data. Security provides safety and reliability. Genetic Algorithm (GA) is typically used to obtain solution for optimization and search problems. This paper presents application of GA in the field of cryptography. Key Selection in public key cryptography is a selection process in which keys can be categorized on the basis of their fitness, making GA a good candidate for key generation. Primary goals of our algorithm was to provide fast and improved performance results having practical and feasible implementation. GA correlates nature to a great extent and produce population of keys such that keys with higher fitness value is replicated often. Good Fitness function helps in exploring search space more efficiently and effectively while bad fitness function traps GA operating in local optimum solution and losing its discovery power. Pearson's Coefficient of auto-correlation was used to calculate the fitness of keys. Ranking of keys was performed to find the best fit key. The private key generated cannot be derived from public key. The key samples satisfy gap and frequency test. Thus, purely random and non-repeating final keys were obtained by application of GA which increased the keys strength and security.

***Index Terms*— About four key words or phrases in alphabetical order, separated by commas.**

I. INTRODUCTION

GENETIC Algorithms (GAs) are adaptive heuristic search algorithms based on mechanics of natural selection and natural genetics. They belong to the class of Evolutionary Algorithms (EAs), which are used to find solutions to optimization problems using mechanisms based on biological evolution such as mutation, crossover, selection and inheritance. Cryptography is an imperative tool for protecting information. Public Key Cryptography (PKC) is an asymmetric outline that uses a pair of keys : a public key for encryption, and a private key for decryption. The fact that selecting key for the PKC is a selection process in which various keys can be categorized on the basis of their fitness, makes GAs a good candidate for the process to be followed for generating keys. The point which we intend to make in this paper is that if the quality of the random numbers produced to generate keys is good then the keys generated will always be purely random and non-repeating and hence increasing the strength of keys and security.

The main idea behind GAs is to replicate the randomness of the nature where population of individuals adapts to its surroundings through natural selection process and behavior of natural system. This means that survival and reproduction of an individual (i.e chromosome) is promoted by the elimination of unwanted traits.

GAs produce a population in such a way that the trait which is dominant, that is has higher fitness value is replicated more likewise rest is rejected based on threshold which is then evolved by the iterative application of a set of stochastic operators like mutation, crossover, and selection. This is also the fundamental concept behind evolution in nature. Highest rank of key fitness signifies the better fitness. The results obtained for generating keys should be good in terms of coefficient of autocorrelation. The samples satisfy the tests including gap test, and frequency test.

A simple GA makes use of following operators to transform a population into new population with better fitness :

A. Crossover

Crossover is a genetic operator that helps in joining two chromosomes to form a new chromosome. The newly generated chromosome is called child which takes one part of chromosome from each parent.

Number of crossovers depends on crossover-rate. Generally crossover rate is 2 to 5%.

Number of Crossover =

$$\frac{(\text{No. of cells in a chromosome} * \text{No. of chromosomes} * \text{crossover rate})}{200} \quad (1)$$

Types of Crossover :

Crossover can be classified into following types :

1) Single point crossover

In this type of crossover, only one crossover point is chosen to generate new child.

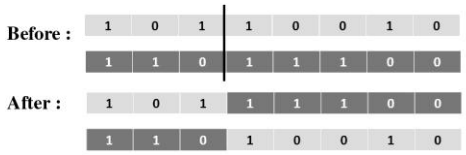

Fig. 1. Single Point Crossover

2) Two point crossover

This type of crossover involves selecting two crossover points to generate new child.

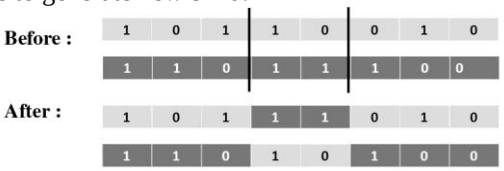

Fig. 2. Two Point Crossover

3) Uniform crossover : In this type of crossover bits of child are uniformly taken from both the parents.

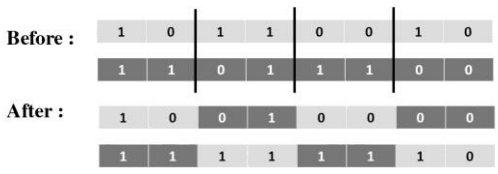

Fig. 3. Uniform Crossover

B. Mutation

Manuscript received on May 2013.

Swati Mishra, Department of Computer Engineering, Delhi Technological University, Delhi, India .

Siddharth Bali, Department of Computer Engineering, Delhi Technological University, Delhi, India.

Mutation is a genetic operator which changes one or more bit values in a chromosome. It is performed on a child after crossover which guarantees the entire state-space will be searched. Its performed infrequently (depending upon probability of altering a cell in a chromosome).
Number of Mutation =

$$\frac{(\text{No. of cells in a chromosome} * \text{No. of chromosomes} * \text{mutation rate})}{200} \quad (2)$$

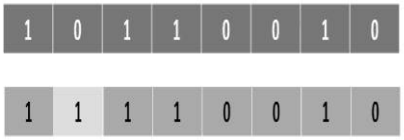

Fig. 4. Mutation

Types of Mutation :

1) Flipping of Bits : It involves selecting one or more bits of chromosome and inverting it.

2) Boundary Mutation : It involves randomly replacing chromosome with either lower or upper bound.

3) Non-Uniform Mutation : It is used to increase the probability that amount of mutation will go to 0 with the next generation.

4) Uniform Mutation : A chosen chromosome cell is replaced with a uniform random value whose range is selected by user.

5) Gaussian Mutation : It involves adding a unit gaussian random value to a chromosome cell.

C. Selection

Selection is the stage of a GAs in which individual chromosomes are chosen from a population for recombination (or crossover).The chromosome with higher fitness value will be considered better.

Classification of Selection :

1) Roulette-wheel Selection
If procedure is repeated until there are enough selected individuals.

2) Stochastic Universal Sampling
If instead of a single pointer spun multiple times, there are multiple, equally spaced pointers on a wheel that is spun once.

3) Tournament Selection
It refers to repeatedly selecting the best individual of a randomly chosen subset.

4) Truncation Selection
It involves taking the best half, third or another proportion of the individuals.

Population Size : The population size remains constant from generation to generation. Determining the size of the population is a crucial factor. Too small population size increases the risk of converging prematurely to a local minima. Initial Population can be determined by randomly or using some heuristic.

Fitness Function : The fitness function plays a very important role in guiding GA. Good fitness functions will help GA to explore the search space more effectively and efficiently. Bad fitness functions, can easily make GA get trapped in a local optimum solution and lose the discovery power. Fitness Function can be classified as Constant fitness function and Mutable fitness function.

II. LITERATURE REVIEW

This work proposes application of GA in the field of PKC which is an essential component of information security. The work makes an attempt to explore the key generating process for PKC to be unique and non-repeating by exploiting GA thus making PKC more secure as par to AES and DES.
To get an idea of the previous attempts made in this field, research papers have been studied and analyzed. In one of the work, GAs is used for searching the key space of encryption scheme [1]. In this work, cryptanalysis of vigenere cipher is done using GA. Another work proposes key generation using GA and deals with only vernam cipher [2].
Computer Scientist in the field of information security and privacy will be curious to investigate the role of GA in cryptography.
The concept of GAs has been highlighted in detail in one of the thesis, GAs in Cryptography by Bethany Delman [3]. In another work, key based bit level Cryptography is done using GA [4] consisting of two levels. In this Bitwise XOR operation is performed which is followed by Genetic Crossover and Mutation.
The primary goals of this work were to provide improved performance and to determine the validity of GA-based approach in the field of cryptography. Currently another work concerning PKC using neural network and GA is also been carried out. The present work is an essential part of bigger perspective work.

III. ALGORITHM FOR PUBLIC KEY CRYPTOGRAPHY USING GENETIC ALGORITHM

Step 1. Initial Population Generation : A GAs begins with a randomly generated set of individuals which is called initial population. Initial population array having values of 192 bit each respectively. Each bit is randomly assigned 0 or 1 bit value corresponding using a random generator function. If value obtained from generator is greater than 50 then bit value 1 is assigned otherwise 0 is assigned. Cell size of chromosomes depicts the key length. All 192 bits of chromosomes cell values are randomly assigned.
Size of initial 2D population array depends on value of MAX_POPULATION which is defined as macro.
Data Structure used : initPop[MAX_POPULATION][192], finalPop[MAX_POPULATION][192].
Step 2. Calculation of Chromosome Number and Threshold Check : Each chromosome should meet a threshold standard. This means that above average chromosomes should have more copies in the population, while below average chromosomes are subjected to extinction based on threshold. For each chromosome, a number is calculated, if its found to be greater than threshold then the corresponding chromosome is selected otherwise rejected. This threshold check is also performed in later stages.
Step 3. Now the GA enters a loop. At the end of each iteration, a new population is produced by applying a certain number of stochastic operators to the previous population. Each such iteration is known as a generation.

Step 4. Selection and Crossover : First a selection operator is applied, in which two parents are selected randomly from

initial Population. The selected parents are used to produce the individuals for next generation through the application of crossover operator.

For binary string individuals, one-point ring crossover, two-point, and uniform crossover are often used. To apply a crossover operator, parents are paired together using one-point Ring Crossover. The Two parents are combined in the form of ring and a random cut point is generated. With reference to this cutting point, one of the children (C1[ ] array) is created in the clockwise direction, while the other one (C2[ ] array) is created in anti-clockwise direction, as shown in Fig. 5. Every position of the child must retain a value found in the corresponding position of a parent, and the child must be a valid permutation. After Crossover, threshold check is performed.

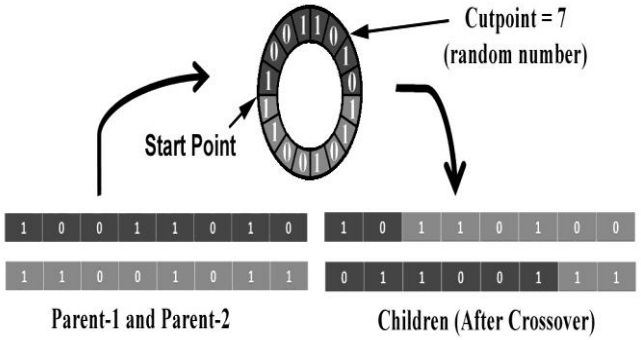

Fig. 5. Ring Crossover

Step 5. Mutation : Next, apply the mutation operator in which a child is randomly changed from what its parents produced in crossover. Number of Mutation is calculated using (2). Thus, Number of Mutation = (192 * 200 * 0.5) ÷ 200 = 96

Note : Sample space taken to be 200 and Mutation rate taken as 0.5%.
For each mutation, select bit number to be modified using random number function : rand()%192 (so one bit is only selected and flipped between 0 and 1).

Threshold check is performed after mutation.

Step 6. Fitness of key Calculation : Repeat the above mentioned steps until the final population array gets full. Chromosomes in final population are arranged according to their fitness values and the chromosome with the highest fitness value is selected.

Fitness value is calculated using following steps :

1) Calculate Shannon Entropy (H(X)) by comparing the percentage of randomness of bits in finalPop[][] against initialPop[][] array using the formula :

$$H(X) = -(p * (\log_2(p))) - ((1 - p) * (\log_2(1 - p))) \quad (3)$$

where p denotes percentage of randomness of finalpop[i][j] from initialpop[i][j] i.e. degree of movement from each bit.

2) Calculate Chi Square value against Shannon Entropy of each finalPop[][] array sample using the formula :

$$X^2 = (\text{observed\_H} - \text{expected\_H})^2/(\text{expected\_H}) \quad (4)$$

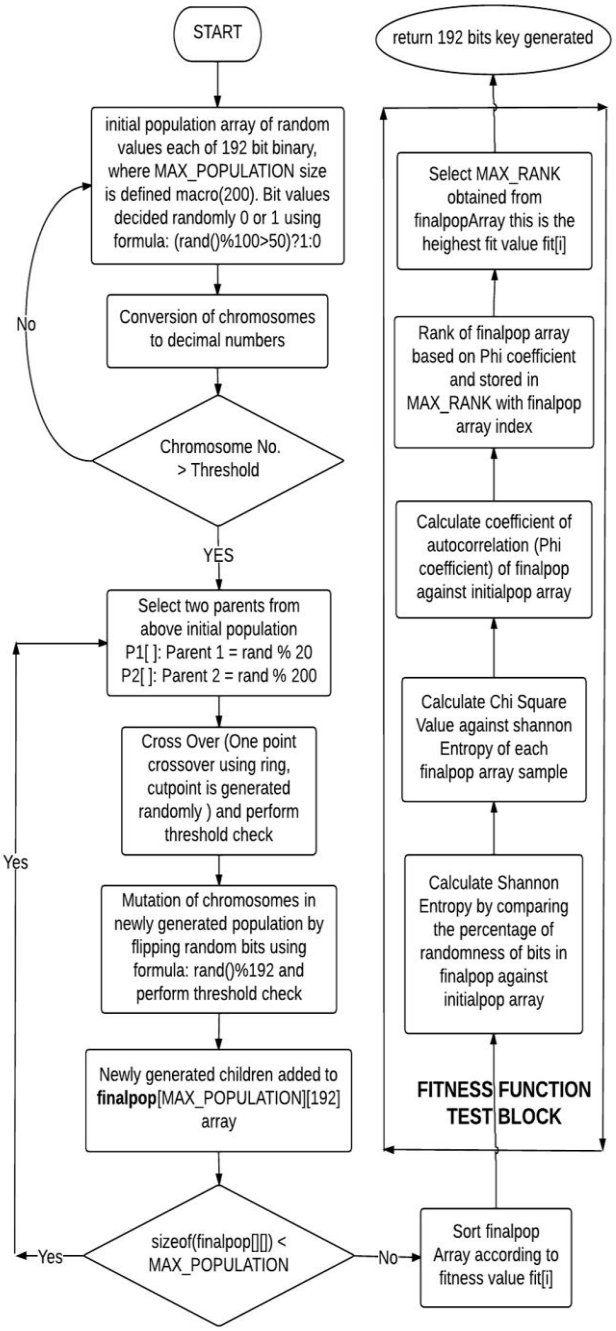

Fig. 6. Flowchart

3) Calculate the Coefficient of Auto-correlation (Phi-coefficient) of finalpop[][] against initialpop[][] array.

$$\varphi^2 = X^2 \div n, \text{ where n is sample space (n = 200).} \quad (5)$$

This coefficient of auto-correlation serves as fitness function and is used to find best fit random key.

Step 7. Ranking of Chromosomes based on Phi-coefficient : Calculate Ranks of chromosomes in finalpop[][] array based on Phi-coefficient and store the index of chromosome with maximum rank in MAX_RANK variable which serves as rank of maximum fit chromosome.

Step 8. Public and Private Key : Thus, chromosome with maximum fitness value serves as 192 bit key. This algorithm is run two times to get private and public keys respectively.

```
1  for(i=0; i<200; i++)
2  {       d1=0; d2=0;
3        for(j=0; j<192; j++)
4        {   int DM = abs(finalPop[i][j] - initPop[i][j]);
5           d1 += j * DM;
6           d2 += j * initPop[i][j];
7        }
8     // percentage of randomness observed
9        p = ((float) abs(d1-d2))/d2;
10   //H(X) calculation :
11  float observed_H=-(p*(log2(p)))-((1-p)*(log2(1-p)));
12   //X^2 Calculation
13    float X_2 = ((observed_H - expected_H) *
14          (observed_H - expected_H))/(expected_H);
15       summation_X_2 = summation_X_2 + X_2;
16      //PHI calculation
17    //PHI corresponding value against total 200 keys
18       float PHI=sqrt(X_2/200);
19          if( max_rank < PHI )
20          {   max_rank = PHI; //update max_rank to new
PHI
21              rank_index = i;
22          }
23   }
24   // find chi sqaure value avg of summation X_2
25   float chi_sqaure = summation_X_2/200;
```

Fig. 7. Pseudo-code

IV. RESULTS AND ANALYSIS

Degree Movement (DM) is calculated to depict the randomness of bit observed in final population against initial population. The implementation proposed generates good random sequences of keys and it is observed to have low running time . Index value is used to represent weights of equivalent position of bits. It is used for calculating expected value of chromosome whereas DM is used to calculate observed value. Percentage of Randomness is denoted by p.

Observed Shannon's Entropy (H(X)) is calculated by comparing the percentage of randomness of bits in final population with respect to initial population array using formula (3). Expected Shannon's Entropy is given by H(X) for p = 0.5 (when binary key consists of equal 0's and 1's).

Next, Chi-Square value is calculated against Shannon's Entropy of each chromosome in final population using formula (4). Statistically, coefficient of correlation gives a better idea of the relation between the items. Thus, coefficient of auto-correction (Phi-coefficient) is calculated using formula (4). The chromosome with maximum value of Phi-coefficient gets Maximum Rank and is taken to be as the best fit key. GAs co-relates the nature to large extent and produce a population in such a way that the trait which has higher fitness value is replicated more.

The chi-square test is used to determine whether there is a significant difference between the expected frequencies and the observed frequencies in one or more categories. Do the number of individuals or objects that fall in each category differ significantly from the number one would expect.

Steps for Chi-square Test :

1) Find observed frequencies
2) Find expected frequencies
3) Find the chi-square value using formula :
4) Find the degree of freedom.
5) Refer Chi Square Table for value.
6) If calculated chi-square value is equal to or greater than the value in table, reject the null hypothesis : differences in data are not due to chance alone.

Gap test counts the number of digits that appear between repetitions of a particular digit. It examines the length of gaps between occurrences of sample values and determines the length of consecutive subsequences with samples not in a specific range. Steps for Gap Test :

1) Specify the class df for the theoretical frequency distribution based on the selected class interval width.
2) Arrange the observed sample of gaps in a cumulative distribution.
3) Find D, the maximum deviation between F(x) and S(x).

Table I
Observed (d1) and Expected (d2) Values for 8-bit key sample

For ith iteration				Observed	Expected
index	initPop[i][j]	finalPop[i][j]	Degree Movement (DM)	index * DM	index * initPop[i][j]
0	1	0	1	0	0
1	1	1	0	0	1
2	1	0	1	2	2
3	1	1	0	0	3
4	0	1	1	4	0
5	1	0	0	0	5
6	0	1	1	6	0
7	0	0	0	0	0
Summation				12	11

The above table depicts calculation for 8-bit key length while in practice 192-bit key length was calculated and analyzed.

Table II
Chi-Square Test Results for 192-bit Key for 10 Sample space

sno	observed	expected	P	H(X) observed	H(X) expected	X^2	PHI=X^2/n
1	7170	7852	0.086856851757514	0.425890454495013	0.5	0.010984449469911	0.002196889893982
2	8618	8932	0.035154500671742	0.219616518939665	0.5	0.157229792903023	0.031445958580605
3	9305	8888	0.046917191719172	0.273154184892371	0.5	0.102918047663689	0.020583609532738
4	8240	7518	0.096036179835062	0.456303141288494	0.5	0.003818830922507	0.000763766184501
5	10114	7841	0.289886494069634	0.868574578083925	0.5	0.271694439219487	0.054338887843897
6	7256	7148	0.015109121432569	0.113018905968287	0.5	0.299508734275963	0.059901746855193
7	9003	8972	0.003455193936692	0.033229351125565	0.5	0.435749677301322	0.087149935460264
8	7580	7250	0.04551724137931	0.267040302883756	0.5	0.108540440960985	0.021708088192197
9	10222	15682	0.348169876291289	0.93242297114832	0.5	0.373979251953482	0.074795850390696
10	7980	8200	0.026829268292683	0.178232593754894	0.5	0.207068527443406	0.041413705488681

The above table is illustration for Chi-square test for 192-bit key. In practice the test was carried on 200 sample space.

4) Determine the critical value, D(α) for the specified value of α and the sample size N.
5) If the calculated value of D is greater, then null hypothesis of independence is rejected.

The frequency test results are shown in Table 1 and Table 2. Here chi-square test is applied on the random keys produced after running the program. Sample with maximum rank was found to have value 0.054 and Observed Chi-square was found to be 0.11 which was observed to be in acceptable range (i.e. below 6.63) according to chi-square table for $\alpha = 0.01$.

In Gap Test, all 200 sample were found to be unique and non-repeating possessing no relation of next random number with previous one. Security Services incorporated :

1) Confidentiality : It is used to protect identifiable information from forced disclosure to prevent its malicious use. Encipherment is done using the proposed algorithm.

2) Data integrity : No data modification confirmed that the information can only be accessed or modified by the authorised users. It is done using Digital Signatures.

3) Authentication : It gives the ability to know the identity of a user, without saying anything about the access rights of the individual.

4) Non-repudiation : Neither sender nor the recipient can deny later from sending or receiving the message respectively. It can be achieved using digital signatures.

V. CONCLUSION

The work has been implemented using C++. 192 bits random Public and Private key were generated. Public and Private Key Samples have been collected and analyzed in Microsoft Excel. In the analysis, population of 200 chromosomes were considered and were found to be randomly generated as proposed such that for each loop an entire new population is observed. Analysis was done for various sample values of keys generated which included frequency test (chi-square test) and gap test to check the nature of randomness and replication of chromosome, satisfactory results were obtained. In Gap Test, all 200 sample chromosomes were found to be unique and non-repeating possessing no relation of next random chromosome of keys with previous generated one.

Threshold check was also performed after crossover and mutation steps which was a deciding factor for the acceptance of chromosome generated throughout the process. This process is stopped when a given termination condition is met. Roulette Wheel Selection was applied which resulted in the replication of favorable data, thus making the population fitter. Ring crossover was applied and the rate of mutation was taken as 5%.

Fitness of keys was found using Pearson's coefficient of auto-correlation followed by ranking using phi-coefficient which decided the best fitness and has been verified from results. Results were analyzed even for large number of chromosomes > 200. The results have been compared with the standard results and were found to be in accepted and satisfactory range after verification. Coefficient of Correlation is found to be satisfactory, random chromosome is selected which is taken as key for PKC.

The primary goals of this work were to produce better and fast performance results and to determine the validity of typical GA-based methods in the field of cryptography. Thus from statistical analysis of results, final keys obtained from GA were observed to be purely random and hence increasing the strength of keys and security.

ACKNOWLEDGMENT

The authors thanks and gratefully acknowledges the support of Mr. H. Bhasin (DTU, Delhi) and Ms. S. Jhajharia (DTU, Delhi).

REFERENCES

[1] Omran, S.S.; Al-Khalid, A.S.; Al-Saady, D. M., "A cryptanalytic attack on Vigenère cipher using genetic algorithm," Open Systems (ICOS), 2011 IEEE Conference on, vol., no., pp.59,64, 25-28 Sept. 2011.
[2] Goyat, S., "Cryptography Using Genetic Algorithms (GAs). " IOSR Journal of Computer Engineering (IOSRJCE), Volume 1, Issue 5 , Volume 1, Issue 5 , June 2012.
[3] Delman, B., "Genetic Algorithms in Cryptography." Master of Science in Computer Engineering, Rochester Institute of Technology, Rochester, New York, July 2004.
[4] Som, S.; Chatergee, N.S.; Mandal, J.K., "Key based bit level genetic cryptographic technique (KBGCT)," Information Assurance and Security (IAS), 2011 7th International Conference on , vol., no., pp.240,245, 5-8 Dec. 2011.
[5] Goyat, S., "GENETIC KEY GENERATION FOR PUBLIC KEY CRYPTOGRAPHY." International Journal of Soft Computing and Engineering (IJSCE), Volume 2, Issue 3, July 2012.
[6] Sharma, L.; Pathak, B. K.; Sharma, R., "Breaking of Simplified Data Encryption Standard Using Genetic Algorithm ", Global Journal Of Computer Science And Technology, Volume 12, Issue 5, Version 1.0, March 2012.
[7] Khan F. U.; Bhatia, S., "A NOVEL APPROACH TO GENETIC ALGORITHM BASED CRYPTOGRAPHY ", International Journal of Research in Computer Science, Volume 2, Issue 3, pp. 7-10, 2012.
[8] Bhasin, H.; Bhatia, S., "Application of Genetic Algorithms in Machine learning", IJCSIT, Vol. 2 (5), 2011.
[9] Goldberg, D E., Genetic Algorithms in Search, Optimization, and Machine Learning. Reading, MA. : Addison-Wesley, 1989. Stallings, W., "Cryptography and Network Security : Principles and Practice", 3rd Edition. Prentice Hall. Boston Columbus Indianapolis.

Ms. Swati Mishra Currently a B.Tech. Student at Delhi Technological University. Her main area of research includes : Design and analysis of algorithms, intelligent system, cryptography and neural networks.

Mr. Siddharth Bali Currently a B.Tech. Student at Delhi Technological University. His main area of research includes : Design and analysis of algorithms, intelligent system, cryptography and neural networks.