Karol Grondzak - Juraj Branicky *

ON THE OPTIMAL PIN SET GENERATION

Security of the information is one of the recent topics, attracting attention of security professionals as well as common people. With deep penetration of electronic devices into our everyday life the challenge to protect our privacy is increasing. Simple mechanism for authentication is shared secret in the form of password. In this paper we describe a method for generating a set of passwords in the form of Personal Identification Number (PIN), which can be used to provide an authentication mechanism for simple electronic devices with limited input possibilities.

Keywords: *Information security, PIN code generation, maximum clique, genetic algorithm.*

1. Introduction

Information is the value of today. We experience a deep penetration of modern information and communication technologies into our everyday life. With this, electronic crime is a serious threat. Smart phones, tablets and similar devices contain a lot of sensitive data about their owners. The protection of privacy has become a serious topic for electronic devices manufacturers.

There are different levels of security depending on the sensitivity level of protected information. For very sensitive, personal data the highest levels of security should be applied. For such purposes special cryptographic tokens are available.

On the other hand there are situations, when some kind of authorization is requested, but the security hazard is not very high. As an example let us consider a shared office printer. Many employees share the printer and some of the print jobs are not to be seen publicly. In such case it could be possible to add a security feature to the printer. A print job can be assigned some authorization code and only upon typing this code on the printer keyboard the job would be printed. The same mechanism could be used to protect an access to some special features of the printer. Modern office printers can serve as copy machines, fax machines, etc. If the management wants to restrict access to these services to a limited set of employees, the password protection is one of the possible mechanisms. The easiest way is to assign each employee unique secret information, which can identify her/him. Because of the limits of the input device capabilities usually the password has to consist of digits only. Such password is known as a Personal Identification Number (PIN).

Another example of PIN application for authentication is the access to land lines from company's private telephone network. If there is a request for different levels of landline access - e.g. limit the access to the local telephone network, or allow a long-distance calls, some kind of authentication mechanism is required. Then the access to landline is granted only after entering the authentication code (PIN).

This simple form of authentication exhibits several security risks. Let us consider two different cases. In the first case we consider that above mentioned authentication mechanism is used in a hostile environment, e.g. there are persons who want to gain illegal access to the services of electronic device, or to the information stored in it. In the second case we consider friendly environment, without illegal attempts to break into device.

In the case of hostile environment, the attackers' strategy can be to simply try different values of the PIN until they succeed. If there is no measure applied, the attackers sooner or later succeed in their attempts.

The second case might seem to be without security risks, but there is at least one. If the user of electronic device enters by mistake some misspelled PIN and this value would be evaluated by device as a valid code, then access to a protected resource can be granted to a person who is not allowed to use it. Primary goal of this paper is to demonstrate a method, how to avoid such a situation.

According to the best authors' knowledge this problem has not been solved yet. Our approach to use graph theory and maximum clique algorithm presents a unique and new idea to tackle this problem.

* **Karol Grondzak, Juraj Branicky**
Faculty of Management Science and Informatics, University of Zilina, Slovakia
E-mail: Karol.Grondzak@fri.uniza.sk

2. Formulation of the problem

For the purpose of this paper, we will consider that electronic device is equipped with some input device (e.g. keyboard), with limited number of characters (e.g. cellular phone keyboard). Let us denote the set of characters, which can be entered on the keyboard as:

$$\mathbf{S} = \{s_0, s_1, ..., s_{k-1}\}, \quad (1)$$

where s_i is i-th character and k is number of different characters, available on the keyboard. Then the set of all available PIN codes of length n can be constructed by Cartesian product:

$$\mathbf{P} = \mathbf{S}'' = \underbrace{\mathbf{S} \times \mathbf{S} \times ... \times \mathbf{S}}_{n}. \quad (2)$$

Number of available PIN codes is then given by the size of set ***P***:

$$N_p = |\mathbf{P}| = |\mathbf{S}|^n, \quad (3)$$

and the elements of set **P** are in the form of n-tuples:

$$\mathbf{p} = \{(s_1, s_2, ..., s_n) | s_i \in \mathbf{S}, i = 1, 2, ..., n\}, \quad (4)$$

If all possible PIN values were assigned to the users, then any mistake when entering PIN code leads to granting access to protected resource. It would mean that any entered PIN code is valid and simply hitting neighboring key by chance would mean wrong user identification. It is not acceptable in situations when correct user identification is important, e.g. when the management wants to keep track of how many pages were copied by the particular employee. To reduce this possibility, only subset of all possible PIN codes should be assigned to the users. Such subset can be generated by the following procedure.

Let us consider a set of PIN values, defined by (2). To decrease the possibility of incorrect user identification caused by entering value of PIN code assigned to another user, we want to choose only subset of that set. The requirements on the subset are:

1. PIN codes in the selected subset should minimize the possibility of incorrect user identification,
2. subset should contain as many PIN codes as possible while preserving previous requirement.

To meet the first requirement, we need a qualitative criterion according to which we choose elements of the subset. We want to avoid the possibility that mistyped value of PIN code is recognized as correct one. This will happen when two PIN codes are "similar" to each other, e.g. when they differ only in one digit and these digits are close on the keyboard. One can easily see that it is relatively easy to enter PIN code 1112 instead of 1111, if keys 1 and 2 are close on the keyboard. It is much more difficult to hit by mistake key 9 instead of key 1 than it is for key 2. To formalize this empiric observation, we proposed to define a metric for PIN codes, based on the distance between characters of the input device.

This metric is defined such a way that it would help us to compare the PIN codes when constructing their subset. We start with metrics of single character and then we expand it to the whole PIN code. Let us denote the distance of two characters:

$$d_s(s_i, s_j), \forall s_i, s_j \in \mathbf{S}. \quad (5)$$

Then distance of any two PIN codes, defined by (4) is:

$$d(\mathbf{p}_i, \mathbf{p}_j) = \sum_{m=1}^{n} d_s(s_m^i, s_m^j), \forall \mathbf{p}_i, \mathbf{p}_j \in \mathbf{P}, \quad (6)$$

where s_m^i is m-th character of the PIN code $\boldsymbol{p}_i$. Having defined metrics, we can transform qualitative requirement that PIN codes in some subset should minimize the possibility of incorrect user identification into quantitative criterion.

Let us choose some minimal value of distance between PIN codes according to defined metrics (6), called threshold (*T*). We can expect that when we construct subset of PIN codes set such that the distance (as defined by (6)) of any PIN code from all other PIN codes in the subset will equal or will be larger that given threshold T, the possibility of incorrect user identification will be reduced. So we satisfy the first requirement by construction of subset P_O such that:

$$\mathbf{P}_O = \{p_i | p_i \in \mathbf{P} \wedge \forall \mathbf{p}_i, \mathbf{p}_j \in \mathbf{P}_O : d(\mathbf{p}_i, \mathbf{p}_j) \geq T\}. \quad (7)$$

It is obvious that construction of some subset which holds property (7) can be accomplished without any difficulties. The problems will be more difficult if subset with maximum possible number of PIN codes has to be constructed, which the second requirement is.

To make sure that subset with maximum number of elements was obtained, it is necessary to construct all possible subsets of set P and then find one with maximum number of elements. These subsets form a power set of set ***P*** and it is well known that a superset of some set contains $2^{|P|}$ elements.

The problem of finding maximum subset of PIN codes can be formulated using the terms of graph theory. Let us construct a weighted undirected graph G:

$$G(V, E), \quad (8)$$

where V is the set of vertices and E is the set of edges. Vertices represent elements of all possible PIN codes of set ***P***. The graph is complete, e.g. there is edge between any two vertices in the graph. Edges are weighted by the value of distance of the PIN codes which are connected by the edge.

To construct subset $\boldsymbol{P}_O$ according to (7), the edges with smaller weight than chosen threshold value T are removed from the graph. Resulting graph ***G'*** is in general not complete. To obtain subset $\boldsymbol{P}_O$ with maximum number of elements, maximal clique of graph ***G'*** has to be extracted.

This way we have transformed the problem of determination of optimal PIN set for the task of user identification into a problem of searching for a maximum clique in an undirected graph.

The problem of maximum clique determination is well known NP-complete problem. It is a combinatorial problem which can be solved either by exact algorithm or by different heuristic methods.

Exact algorithm for finding maximum clique in general constructs and evaluates all possible subsets of set ***P*** defined by (2). Many techniques have been proposed to reduce the amount of evaluated subsets. Some of them are based on the branch-and-bound algorithm, proposed by Carraghan and Pardalos [1 and 2]. We can mention here the work of Konc and Janezic [3], Ostergard [4] or Tomita and Kameda [5]. For many practical problems the size of the graphs does not allow to apply exact algorithm.

As the maximum clique problem is applicable for many theoretical and practical problems, a lot of heuristic methods were proposed to solve it. We can mention the simulated annealing approach proposed by Geng, Xu, Xiao and Pan in [6], the ant colony optimization approach proposed by Solnon and Fenet in [7] or the genetic algorithm approach proposed by Marchiori [8]. Local search algorithm was proposed by Katayama, Hamamoto and Narihisa in [9]. The published algorithms were tested on DIMACS benchmark set [10], which is available on Internet (ftp://dimacs.rutgers.edu/pub/challenge/graph/benchmarks/clique/). It consists of graphs of different properties concerning the density and maximum clique size. Reference implementation of the exact algorithm is also available for download at DIMACS site.

3. Experimental results

To test the possibility of optimal PIN set generation, we considered the following configuration. The device access to which should be protected by the PIN code is equipped with a simple keyboard containing keys to enter digits from zero to nine (Fig. 1). For simplicity we will refer to each key by the character it is representing, e.g. key with symbol zero is zero character, or zero key, etc. Keys are arranged into two-dimensional lattice. Each key is assigned position in the lattice in the form of tuple describing the row and column of the lattice where the key is located. Upper left corner of the lattice has coordinates *(0,0)* and they increase in the directions to the right and down. To calculate the distance between any two characters on the keyboard we use Manhattan metrics:

$$d_s(s_i, s_j) = |r_i - r_j| + |c_i - c_j|, \quad (9)$$

where s_i, s_j are any characters of the keyboard, r_i, r_j are row coordinates and c_i, c_j are column coordinates of the corresponding characters s_i, s_j respectively.

From (9) it is clear that the distance between digits one and nine is:

$$d(1,9) = |0-2| + |0-2| = 4. \quad (10)$$

It is also obvious for the keyboard setup as shown in Fig. 1 that maximum distance between any two characters cannot be larger than four. The distance for PIN codes of length n can be calculated using above mentioned formula (6) and maximum distance between any two PIN codes, considering keyboard in Fig. 1 cannot be larger than 4n.

Fig.1 An example of the keyboard configuration

To verify our proposal and to better understand the properties of PIN codes sets, we have experimented with PIN codes of length two, three and four digits. Four digits PIN codes are quite popular for real devices, smaller sizes (two and three) are suitable for verification of the algorithms because of fast evaluation of the cliques.

As was mentioned in the previous paragraph, DIMACS library contains referential implementation of branch-and-bound algorithm. This algorithm was used to evaluate exact solutions for graphs, for which it was possible to find the solution in reasonable time. These results gave us some general information about the properties of graphs, which represent the optimal PIN set problem.

We also experimented with heuristic algorithm for finding maximum clique. We choose the Marchiori's algorithm, which is combination of genetic and simple novel heuristic. The novel heuristic is executed in each step of the genetic algorithm and is based on three steps. In the first step, the relaxation algorithm is executed to randomly enlarge particular solution. Next, there is executed the repair algorithm to extract the clique from the enlarged solution. Finally the extend algorithm is executed to enlarge the clique using simple greedy algorithm. Detailed description of the algorithm can be found in [8].

All experiments were performed on commodity computer equipped with two Intel CPUs (Intel(R) Xeon(R) CPU E5530 2.40GHz) and 4GB of RAM memory and the results are summarized in Table 1. First columns denote the size of the PIN code and the value of the threshold.

Following are the columns describing the properties of the graphs obtained after applying the above described algorithm. As we expected, when the threshold value increases, the amount of edges in graph decreases. Next column represents the minimum value of node degree. It can be seen that for larger thresholds nodes with zero degree appear in the graph. For thresholds

Results of heuristic

Table 1

PIN code size	Threshold Value	Edge count	Min Degree	Count of 0-degree	Max Degree	Density	Exact solution size	Size of the heuristic solution	In iteration	Time to find solution [s]
2	2	4690	91	0	97	0.9475	52	52	4	0.02
2	3	4012	65	0	90	0.8105	15	15	3	< 0.01
2	4	2908	25	0	78	0.5875	10	10	4	< 0.01
2	5	1678	0	1	59	0.3390	5	5	0	< 0.01
2	6	722	0	9	37	0.1459	4	4	0	< 0.01
2	7	208	0	35	16	0.0420	2	2	0	< 0.01
2	8	32	0	75	4	0.0065	2	2	0	< 0.01
3	2	495600	987	0	996	0.9922	N/A	504	10	2.02
3	3	480360	924	0	984	0.9417	N/A	94	210	7.79
3	4	441752	740	0	956	0.8844	N/A	59	570	13.82
3	5	371576	425	0	896	0.7439	N/A	20	157	2.68
3	6	275504	125	0	794	0.5516	15	14	5	0.07
3	7	173976	0	1	644	0.3483	8	8	125	1.78
3	8	90456	0	13	464	0.1811	6	6	5	0.07
3	9	37212	0	76	281	0.0745	3	3	0	0.02
3	10	11440	0	260	134	0.0229	3	3	0	0.03
3	11	2368	0	575	44	0.0047	2	2	0	0.03
3	12	256	0	875	8	0.0005	2	2	0	0,03
4	2	49943000	9983	0	9995	0.9990	N/A	4992	40	1073.07
4	3	49672200	9867	0	9977	0.9935	N/A	605	1327	3752.56
4	4	48746280	9371	0	9925	0.9750	N/A	354	747	870.15
4	5	46432752	8005	0	9790	0.9287	N/A	90	185	87.5
4	6	41968304	5525	0	9502	0.8395	N/A	60	123	38.69
4	7	35097872	2625	0	8962	0.7020	N/A	26	1020	229.55
4	8	26503760	625	0	8090	0.5301	N/A	19	73	14.02
4	9	17672856	0	1	6851	0.3535	N/A	10	937	156.45
4	10	10188920	0	17	5327	0.2038	9	8	28	4.35
4	11	4968168	0	133	3701	0.0994	4	4	0	2.26
4	12	1995400	0	629	2225	0.0399	4	4	13	1.8
4	13	636544	0	1995	1104	0.0127	3	3	2	0.27
4	14	152320	0	4475	424	0.0030	2	2	0	1.97
4	15	24576	0	7375	112	0.0005	2	2	0	1.96
4	16	2048	0	9375	16	0.0000	2	2	0	1.93

smaller than half of the PIN code size minimum degree is nonzero, but significantly larger than the size of found maximal cliques. It means that newly generated graphs are still complex and finding maximal clique by exact algorithm is time consuming procedure. To be able to compare our graphs with graphs contained in DIMACS benchmark set, we have calculated other graphs characteristics, like maximum node degree, number of nodes with zero degree and the graph density.

Finally the last columns contain information about the maximum clique size found by exact algorithm and by a simple genetic algorithm published by Marchiori [8].

Only the results for sparse graphs could be obtained by exact algorithm in reasonable time. Unfortunately these results are not very useful as the resulting cliques are small (only of size of tens). Such small cliques are not sufficient for practical problems when we usually have to assign hundreds of PIN codes for the users.

But the algorithm is useful to verify that heuristic algorithm is working properly.

When comparing the maximum cliques solutions found by exact and heuristic algorithms (when applicable), we can see that the difference between the solution size is mostly zero and in a few cases it is one. This gives us the confidence that heuristic algorithm can give us almost optimal solution in a reasonable time.

Taking into consideration the obtained results, we can conclude that reasonable large set of PIN codes is obtained when the value of threshold is approximately one quarter of maximum possible threshold value. The results demonstrate that for company with several hundreds of employees minimum reasonable PIN code size is four.

4. Conclusions

In this paper, we have presented the problem of finding set of PIN codes for unique user authentication applied for simple electronic device equipped with limited capabilities keyboard. We have proposed a procedure how to obtain maximum possible set of PIN codes by converting the problem into the problem of maximum clique search. As this problem is one of the fundamental and well-studied problems, there are many algorithms proposed to solve it. In our paper we present the results of two algorithms applied to PIN code of sizes from two to four. Exact algorithm gives result only for sparse graphs and was used to verify the results of the heuristic algorithm. Using simple genetic algorithm, we were able to find maximum cliques for all the PIN code sizes and threshold values.

As a future work, we would like to study the properties of graphs, obtained by the proposed procedure into more details. To be able to obtain solution for larger PIN codes, we plan to parallelize the exact and heuristic algorithms. We also plan to extend our research by applying other maximum clique search heuristics, like simulated annealing, particle swarm optimization or differential evolution.

Acknowledgment

This paper is supported by the following project: University Science Park of the University of Zilina (ITMS: 26220220184) supported by the Research & Development Operational Program funded by the European Regional Development Fund.

References

[1] CARRAGHAN, R., PARDALOS P. M.: An Exact Algorithm for the Maximum Clique Problem, *Operations Research Letters,* 9, 1990, 375-382.

[2] PARDALOS, P. M., RAPPE, J., RESENDE, M. G. C.: *An Exact Parallel Algorithm for the Maximum Clique Problem.* High Performance and Software in Nonlinear Optimization, Kluwer Academic Publishers, 1997, 279-300.

[3] KONC, J., JANEZIC, D.: An Improved Branch and Bound Algorithm for the Maximum Clique Problem, *Commun. Math. Comput. Chem.*, 58, 2007, 569-590.

[4] OSTERGARD, P. R. J.: A Fast Algorithm for the Maximum Clique Problem, *Discrete Applied Mathematics,* 120 2002, 197-207.

[5] TOMITA, E., KAMEDA, T.: An Efficient Branch-and-bound Algorithm for Finding a Maximum Clique with Computational Experiments, *J. of Global Optimization,* 37, 2007, 95-111

[6] GENG, X., XU, J., XIAO, J., PAN L.: A Simple Simulated Annealing Algorithm for the Maximum Clique Problem, *Information Sciences*, 177, 2007, No. 22, 5064-5071

[7] SOLNON, CH., FENET, S.: A Study of ACO Capabilities for Solving the Maximum Clique Problem, *J. of Heuristics,* 12, 2006, 155-180

[8] MARCHIORI, E.: *A Simple Heuristic Based Genetic Algorithm for the Maximum Clique Problem*, Proc. of ACM Symp. Appl. Comput.: 366-373, 1998.

[9] KATAYAMA, K., HAMAMOTO, A., NARIHISA, H.: An Effective Local Search for the Maximum Clique Problem, *Information Processing Letters,* 95, 2005, 503-511

[10] HASSELBERG, J., PARDALOS, P. M., VAIRAKTARAKIS, G.: Test Case Generators and Computational Results for the Maximum Clique Problem, *J. Global Optim.*, 3, 1993, 463-482.