Optimal Key Size of the AVK for Symmetric Key Encryption

Shaligram Prajapat[1] **&**

Dr. Ramjeevan Singh Thakur[2]

[1] International Institute of Professional Studies, Devi Ahilya University, Indore, India.

[2] Department of Mathematics and Computer Applications, Maulana Azad National Institute of Technology, India

(shaligram.prajapat@gmail.com, ramthakur2000@gmail.com)

Abstract: The security of AVK based cryptosystem can be enhanced merely by exchanging the key using parameters. Today, the major challenge we face in design of AVK model of symmetric key encryption is fixing key length for AVK. On deeper scrutiny, it was revealed that a key of shorter length increases vulnerability of the system. On the other hand, key length beyond optimum length involves unnecessary overheads (suboptimum utilization of bandwidth). Thus, this paper resolves the conundrum of research questions, and answers estimation of optimum key size for AVK model. The paper provides useful insights towards decision making for optimal key length.

Index Terms—AVK (Automatic variable key), Symmetric Key.

I. Introduction

The security of information transmitted over a public network depends upon 3 characteristics namely: (1) Enciphering algorithm (2) Protocol and (3) Key. It is often assumed that a system is secure if we have a strong encryption algorithm or a secure protocol. But, security of the whole cryptosystem may be compromised if the key is mishandled: If somehow opponents get hold on keys, the security of information is compromised. So, a systematic and appropriate use of keys is essential to ensure security of information. In principle, any cryptic key can have following features: Key-Length, Key-Randomness, Key-Lifetime, and Key-Secrecy. Although the dynamism, randomness or variability in key increases security of information transmitted over the communication channel, a number of techniques and methods are still under investigation for its efficient implementation.

The strength of traditional crypto-algorithm is the function of key length which takes longer computation time for large key length. This is an overhead associated with Key Length. Alternative approach for ensuring security is use dynamic keys or fix up the key length and vary it for every new session. But, the major issue would be: if key is kept fixed then what should be its length? Shorter key length may be easy to predict and higher the key would lead to overheads similar to big key size .In the subsequent sections, the paper highlights alternative approaches for the issue. It would also recommend time variant key approaches for better safety. The paper also finds answers for key size of AVK to balance the vulnerabilities and computation time.

II. Related Work

Although numerous literature works are available on key based algorithm for securing and comparing efficiency of information, rare amount of work is available on guidelines for choosing Key-size for AVK based cryptosystem. The documentation [1] of Microsoft insinuates that the chance for success of systematic attacks (where intruder tries each permutation of the key until the desired key to decipher the message is explored) depends on the key size. According to this document, the best alternative for minimizing the success rate of brute force attacks are: (1) Select small key-Lifetime or (2) Select key- size, which is large in length. In the former approach, the smaller key-lifetime minimizes the probability of attacks (even if one of the keys is known). This leads to conclusion of key variability. On the other hand, limitation of key selection in fixed key of longer size is used to minimize the probability of successful attacks by increasing the number of combinations [1].

Almost 13 years back, in 2002, Hellman highlighted the effects of increasing the key-length, for improving security level on the vulnerable network. In his article, the suggestion of suitable key-length for secure information exchange was claimed to be of 90 bits with accordance of trends of increasing key size. As indicated in the Table 1.0. The information exchange with key of larger length is assumed to be more secure due to large computation time requirements. One can straight forward infer that, to secure data we need to increase the key length [2].This is a limitation and impractical aspect from future computing perspective.

According to Moore’s law, the computing power of personal computer doubles approximately in every 18 months. If we check key-length problem in alignment with Moore’s law, the effect of proliferating computing power can be experienced on computation of key size of large lengths. The availability of fast multi-core processors and low cost hardware will equip the cryptanalyst. So, intruders and hackers can rigorously develop new techniques and algorithms to exploit and improve the efficiency of key search to breach system in less time. The estimated time for successful key search attacks must be revised as computing power and resource availability increases.

Table 1. Key Size and its impacts on time and speed

Key Size (in bits)	Significance	Traditional-Time and Present speed
40 bits with 2 ^{40} permutations	1 PC with rate of 1 million keys per second, will take 13 days to try out all possible keys	13 days and in Hours rather than in days
56 bits with 2 ^{56} permutations	DES-with supercomputer of entry rate 92 billion keys per second decrypted the message in 56 hours after trying about 25 percent of the possible keys.	Remined safe for 56 hours and after six months in 22 hrs.An improved version of it was decoded by supercomputer (56-bit DES encrypted message)in about 22 hours,.
64 bits with 2 ^{64} permutations	Better Performance over 56 bit key size	Relatively more time is required with respect to traditional DES,generally provides strong protection against brute force attacks.
128 bits with 2 ^{128} permutations	10 million PC trying 100 billion keys per second will take about 10 ^{13} years to try every possible 128-bit key value.	1000 times longer than the estimated age of the universe (15 billion to 20 billion years).But, symmetric keys that are 128 bits or longer are considered unbreakable by brute force attacks.
192 bits with 2 ^{192} permutations	Excellent performance	Relatively hard and secure
256-bit with 2 ^{256} permutations	Excellent performance	Hard but Highly Secure

In traditional approach, user chooses/creates a key containing a string of characters. The key string may be in the form of alphanumeric, numeric, special symbols etc. Depending upon type of implementation checked by source or destination computer, if the supplied key matches with the one which is associated with the actual user’s resource (files, databases, etc), access is granted to all resources belonging to the authorized user.

Primary approach for inexpensive key design is choosing a relatively short string of characters and allowing the user to decide the key, in such a way that the selected key is memorable. However, with such type of keys are easier to guess. Likewise, if user selects a key arduous to remember, it is most probable that he will save the key somewhere (either electronically in hard disk or non-electronically in paper slips).In both cases, the system is equally vulnerable to attack.

An alternate approach to this problem can be implemented by increasing the key length. This would make the system relatively more secure to stand against exhaustive cryptanalyst search. The expected "safe-time" and "breaking-threshold" can prevent(secure) the key from brute force attack. It can be computed by expression (1) and (2). It can also be used to indicate effectiveness of key by its length (used in a given system) [3].

Safe Time: It is the maximum time required to guess a key (in a brute force attack) and is computed using the formula (1):

Safe time =0.5*Total number of possible keys*Time to enter one key (1)

Breaking Threshold: It is the optimum time taken to find the right key to breach the system. For a given key, selected from characters set domain of size N, breaking threshold for a given key is given by:

Breaking threshold = 1/2*N^x*L/(R) (2)

Where: X = length of key (in number of characters), R = Data entry and transmission rate: R characters per minute. N = Size of character set domain i.e. number of letters, numbers, and special symbols (from which the key is selected). Thus, Number of characters involved for entry and replying in a login attempt is N characters.

III. Optimum Key for AVK Model

The AVK model of variability of key for symmetric key has already been proposed in the literature for improved security aspects. In AVK approach, key is varied in respect of time over sessions. Initially key is generated by variable information (exchanged in prior session).The superiority of AVK over a fixed key or key with variable length has been proved in literatures [4, 5, 6, and 7]. Fibonacci–Q matrix, Sparse matrix based approaches are recent approaches that can be used to generate variable keys. Dynamic keys can be generated among a number of users. [8]. A number of techniques to generate time variant key are proposed. In reference [7], we studied a comparative study performed to find out the best techniques among different key generation techniques.

IV. Experimental Setup & Results

A Python 2.7.8 script using pandas library and matplotlib library were used to plot the result under various parameter of x and y. The Pandas library was used to compute the data using the formula (2).The calculated value of time to guess (in Hrs) was stored with corresponding key size in a data frame. The plot ( ) function was used to demonstrate the result.

The effect of increasing the key size was investigated for finding out what would be the optimum key length. It is assumed that data entry rate for entering key is 120 characters per minutes, the character set size is 20 characters (Frequent characters are not considered in character set) login length is 15 characters. Using the Anderson's formula,

N^x≥(4.32*104 *T* M)/(L* p0) (3)

In (3) it is also assumed that the probability for a correct guess is p and the time period in months for systematic attack has been made 24 X 7 in M months, and the lower bound probability is p0 .This expression can be used to decide length of the key (x). Such that it reduces the possibility middle attack as compared to p0.

The Effect of increasing the key length is shown in Table 2. The comprehensive security was studied for various character lengths. The plotted graph demonstrates that the optimum key length lies in between 4 to 8.

Table 2 Effect of increasing key length

Size of key (in characters)	**Time taken to successful guess (In Hrs.)**
20	7.610350076
26	36.73370624
52	2350.957199
62	6754.213706
72	16566.07562

For plotting the graphs for estimation we have considered the system with: Key length = 6, Login length = 6, Typing speed: 120 characters per minutes, x = Length of character set and Time taken to guess (in Hrs) is y then

$$y = \frac{0.5*x^6*15}{120*60} \qquad (2)$$

The plot is shown in Fig. 1.

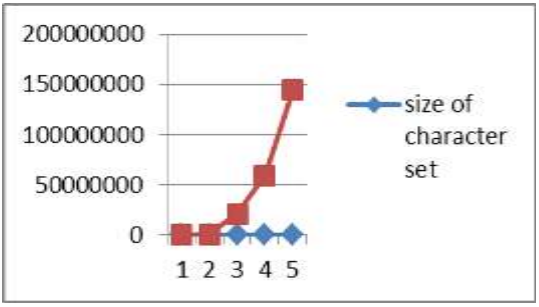

Figure 1. Time to Guess Key vs Length of Character Set

$$y = \frac{0.5*72^{x}*15}{120*60} \qquad (4)$$

Assuming length of character set= 72 (including alphanumeric key elements) and login length =15 and typing speed: 120 characters per minutes. The plot of "key length" with respect to "time taken to guess (in Hrs)" is presented in Fig. 2.

Table 2. Key Length V/S Time

Key Length	Time to guess (in Hrs)
1	0.020833333
2	0.416666667
3	8.333333333
4	166.6666667
5	3333.333333
6	66666.66667
7	1333333.333
8	26666666.67
9	533333333.3
10	10666666667
11	2.13333E+11
12	4.26667E+12
13	8.53333E+13
14	1.70667E+15
16	6.82667E+17
20	1.09227E+23
25	3.49525E+29
50	1.17281E+62
100	1.3205E+127=

The graph indicating key length and time to guess the key seems linear, assuming length of key i.e. number of characters on X axis and time to guess all permutation (in hours) is as shown below in (Fig.2).

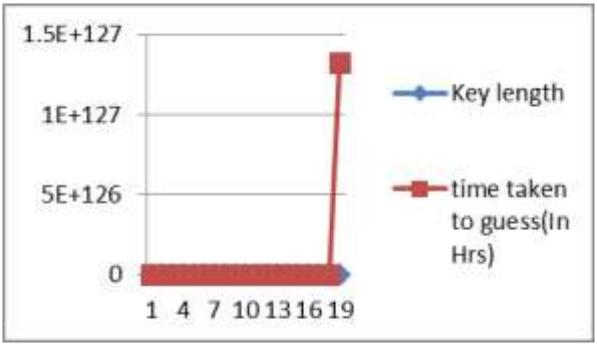

Figure 2. Time to Guess Key vs Length of Character Set

The plot of Fig.2 displays homogeneous behavior over all key size. So, a deeper introspection look is required to observe its effect over time.

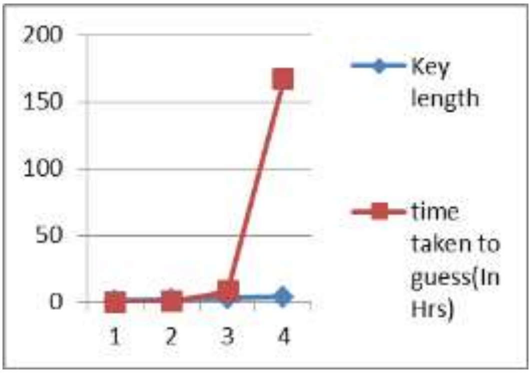

Figure 3. Time to Guess Key vs Length of Character Set

Now to explore the effect of time (to guess a key) for Key length varied from 4 to 8 characters , as per the plot in Fig. 3.

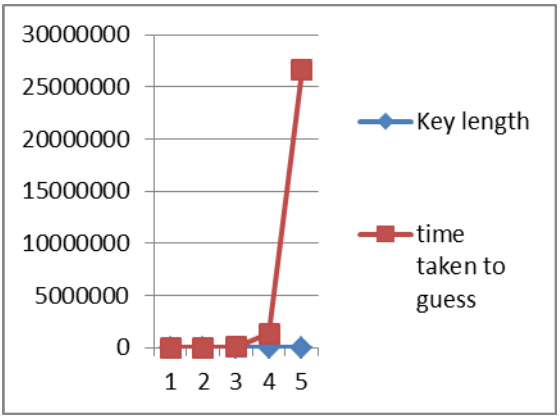

Figure 4. Time to Guess Key vs Length of Character Set
Effect of time to guess a key for Key length varied from 6 to 10 characters.

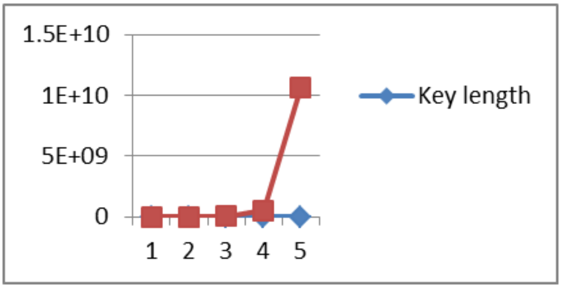

Figure 5. Time to Guess Key vs Length of Character Set
Effect of time to guess a key for Key length varied from 8 to 14 characters.

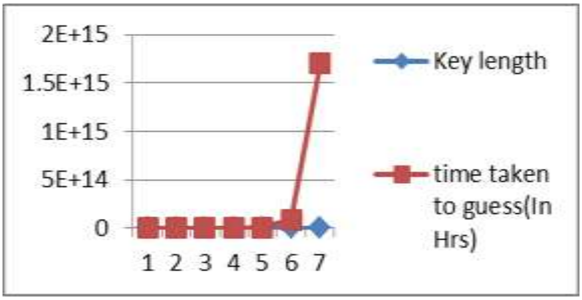

Figure 6. Time to Guess Key vs Length of Character Set
Effect of time to guess a key for Key length varied from 12 to 16 characters.

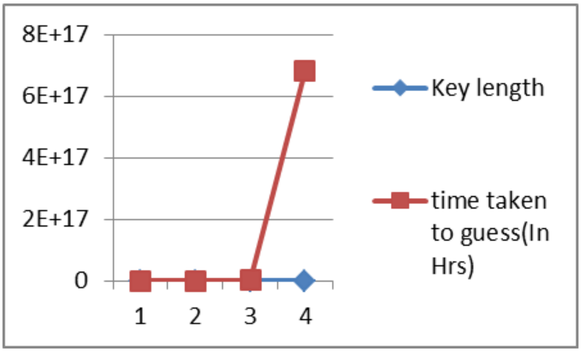

Figure 7. Time to Guess Key vs Length of Character Set

Effect of time to guess a key for Key length varied from 16 to 25 characters.

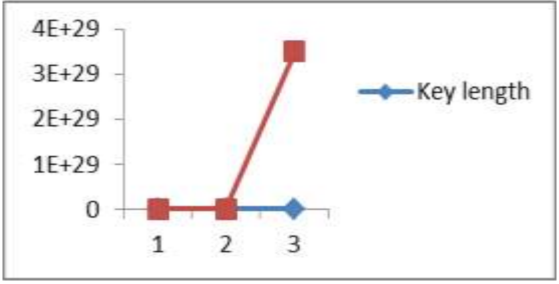

Figure 8. Time to guess key vs length of Character set

For implementation of Automatic Variable key with fixed length but dynamic in nature from session to session should be at least 5 or 6 characters for sufficient resistance against vulnerability.

V. Conclusions

In traditional cryptosystem the practical difficulty of increasing the key length to reduce the probability of damage from side channel attacks, makes it arduous and inconvenient to remember long string. User has to record the key either in the system file or on piece of paper, both are undesirable. Relatively shorter keys would be easy to remember and convenient in handling and changing from session to session from the perspective of AVK based cryptosystems. The work described in this paper point outs that optimal key length for fixing up with automatic variable key length of 5 or 6 characters (key length to prevent from system attack) is sufficient for a session. Once a key of this key length is initiated then it can be changed from session to session, so that the security of the system is not compromised. It is also worth mentioning that key generally does not fail because of brute force attack but, fails as a result of mishandling of the key.

VI. Acknowledgement

This work is supported by research

project under Fast Track Scheme for Young Scientist from DST, New Delhi, India. Scheme 2011-12, No. SR/FTP/ETA-121/ 2011 (SERB), dated 18/12/2012References.

References

BlueKrypt, Cryptographic Key length Recommendation http://www.keylength.com/en/4

Martin E. Hellman, “An overview of public key cryptography”, IEEE Communication Magazine- May 202

Lane, “Security of Computer based Information System”, Macmillan series,1990.

R.Goswami, S. Chakraborty, A. Bhunia,C. T. Bhunia ,'Generation of Automatic Variable Key under Various Approaches in Cryptography System", Journal of The Institution of Engineers (India): December 2013, Volume 94, Issue 4, pp 215-220

Galen E. Pickard, Roger I. Khazan, Benjamin W. Fuller, Joseph A. Cooley, "DSKE: Dynamic Set Key Encryption", MIT Lincoln Laboratory.

Shaligram Prajapat, Sachin Saxena, Amber Jain,“ Implementation of Information Security with Fibonacci Q-Matrix”, in the International Conference on Intelligent Computing and Information System (ICICS-2012).

Shaligram Prajapat, Amber jain, R.S.Thakur, “A Novel Approach For Information Security with Automatic Variable Key Using Fibonacci Q-Matrix”, International Journal of Computer & Communication Technology (IJCCT) ISSN (ONLINE): 2231 - 0371 ISSN (PRINT): 0975 – 7449 Vol-3, Iss-3, 2012, p.p. No. 54-57.

Shaligram Prajapat, Dr. R.S. Thakur et al., “Sparse approach for realizing AVK for Symmetric Key Encryption”, International Research Conference on Engineering, Science and Management 2014 (IRCESM 2014) Dubai, UAE –ISBN 978-93-83303-51-9

Shaligram Prajapat, Dr. R.S. Thakur, ”Time variant approach towards symmetric key”, SAI-Conference London, cosponsored by IEEE and Springer , October-2013.

Author’s profiles:

Shaligram Prajapat, He is pursuing Ph.D. under supervision of Dr. Ramjeevan Singh Thakur, from Department of Mathematics and Computer Applications, Maulana Azad National Institute of Technology.He has received B.Sc.(Electronics), M.Sc.(Com.Sc.), M.Tech(Com.Sc.), M.Phil.(Comp.Sc.) from Devi Ahilya University,Indore.He is International Institute of Professional Studies (IIPS) ,Devi Ahilya University Indore, as Reader for MCA and M.Tech Courses since 2007.With over 15 years of teaching experience, He has reviewed five international books of Pearson education, 3 papers in international journals including Springer and Atlantis press. He has also presented paper in international and national conferences. He is member of various professional bodies like IEEE, ISTE, ACM, CSI, CSTA, IAENG, IEEE(Computer Society).

Dr. Ramjeevan Singh Thakur is eminent researcher and Associate Professor in the Department of Computer Applications at Maulana Azad National Institute of Technology, Bhopal, India. He is a Teacher, Researcher and Consultant in the field of Computer Science and Information Technology. He earned his Master Degree from Samrat Ashok Technology Institute, Vidisha (M.P.) in 1999. And Ph.D. Degree (Computer Science) From Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P.) in 2008. He has published more than 75 Research Paper in National, International, Journals and Conferences. He has visited several Universities in USA, Hong Kong, Iran, Thiland, Malaysia, and Singapore. His areas of interest include Data Mining, Data Warehousing, Web Mining, Text Mining, and Natural Language Processing. He has also received DST Young Scientist Award-2011 in Engineering under Fast Track Scheme, Department of Science & Technology, New Delhi, India. His area of interest are Data Mining, NLP Bioinformatics, Soft Computing