

Multilanguage Software License Key Generator: Accented Characters Approach

¹Olawale John A, ²Moradeyo Oluwatomilola M, ³Ojoawo Akinwale O, ⁴Olaniyan Abolade S, ⁵Adeyeye A. Hannah

^{1,2,3,4}Department of Computer Science, Adeseun Ogundoyin Polytechnic, Eruwa, Oyo State, Nigeria

⁵Department of Electrical Engineering, Adeseun Ogundoyin Polytechnic, Eruwa, Oyo State, Nigeria

Authors E-mail: loolabode47@yahoo.com, tokenny2003@yahoo.com

Abstract - License key generator is an ideal program use to protect proprietary application against unauthorized access. From findings, the existing license key generator only support English speaking countries, which limits the participation of other language character sets. These Non-latin sets are called accented characters which are important elements in both written and spoken language while their use in English is greatly confined to proper words borrowed from certain countries. However; this research enhances the existing license key algorithm through the use of accented characters which support multiple languages. In other to enhance the key generator, the existing data set were updated with accented characters using C# language. Advance Encryption Standard (AES) symmetric was used to secure the generated key at the cache level while linear congruential generator algorithm was used to generate the characters in a random format. Eureqa, an automated machine learning model was used to generate an equation that checks for the vulnerability, accuracy of the key generator. Therefore, the developed license key is more secure compare to the existing algorithm. Also, it support Multilanguage software key which improves the performance and allows more participation in its use. Hence, this developed system is useful for Non-English speaking countries, which provides more security through the encryption method.

Keywords: License Key, Linear Congruential generator, Eureqa, Accented characters, Advance Encryption Standard.

I. INTRODUCTION

Accented characters are important words used in both written and spoken language and their use in English is majorly limited to borrowed words of foreign countries such as resume and tete-a-tete. These so called elements appear frequently in so many European languages which include Spanish, French, German and Portuguese. Kerckhoffs' doctrine, put forward several centuries ago says that, the design of a system is dependent on its access key and not its design being hidden. License key generation and its validation is one important requirement for system applications. It is a

known fact that no license scheme is fully secure but it is an ideal program use to protect applications against piracy and illegal usage. When developing an application, one of the problems that a programmer may encounter is trying to protect his/her application against piracy. Most of the times, people can work break through a proprietary software protection through cracks. This is done by updating the application's installation or executable files, by camouflaging as a valid license. One of the most secure widespread methods to avoid illegal copies or to stop cracking is to create unique serial or product access key but these hackers has a way of unlocking this generated key through some programs that are auto generic. Furthermore, to proffer solution to the problem of illegal and unauthorized access, it is necessary to create license file that will be unique for each computer. In addition, the license file will be associated with the machine. Programs such as Serial Key Generator can be very useful for this function but generate many valid serial numbers. In other to prevent license keys from working for all versions of software, a partial key verification technique was developed. This technique uses some bits to randomly check the validity of a license key and will only support a Latin set for English speaking European countries.

From the foregoing, there will be need to enhance existing license key generator that will support other Multilanguage key gen European countries that do not speak English through the use of accented characters. A license key or registration validation codes that are offline will be more reliable due to intermittent network failure. This technique when implemented will make operational transparency increase so as to prevent software misuse.

II. LITERATURE REVIEW

Over the years one of the contradictions of security through obscurity was the researcher's ideas that was demonstrated to the public on how software could be cracked or bypass the validation checks. Again, responding to the security concerns on the flaws which may occur in the design of lock or key can make software more exposed to attacks Alfred C.H (1851). Criminals are very serious in their

profession, and already know more than they can be taught. Product keys are design for proprietary software which consists of different set of numbers and combination with letters. This sequence will be typed by the user during the installation of the software, after which it will be passed to a function that will manipulate the access key sequence based on a set of mathematical algorithm then try to match the results to some set of acceptable solutions which is known as validation of the key generated (Vangie Beal, 2011). Software license keys can be used in various methods for copy protection. The basic idea is that only users that have acquired the appropriate license will be issued with a key which enables them to install or use the software. From a software programmer opinion, the purpose of a software licensed key is to prevent the software from being shared, copied illegally by non-licensed users. There are two primary ways to go about creating a key, it's either online and offline mode. For an online key generation, it is a simple matter of generating a random sequence of characters, storing that in a database, and then using what is known as a license server to validate the serial number by communicating over the Internet every time the software starts up or does something significant. Online serials eliminate the key generation aspect of piracy since you directly control the keys that are possible and how many times they are used to install the software. One major immediate downside or disadvantage is that the software won't run at all without an Internet connection, which is a non-starter for some people. However, there are also long-term downsides as well when we consider what happens when you remove support for an older version of the product or too many people hit the license server at once, you stop running your business, you forgot to pay a bill and the domain your software talks to goes away, the outcome is that the license key generation server that validates the serial numbers on the internet will go offline or will not work properly. Those things actually happen with unfortunate regularity. Eventually, software installations of your clients will eventually fail when using online.

Any License generator algorithm could be developed to generate keys for any authorized user for protecting their application against unauthorized access. This is done by generating unique number or hidden access key that only recognized the authorized user of such software application. Furthermore, in one of the existing algorithm that uses MAC address, it generates main license key which is used for the main program and an advance key which will be used for the subprogram and option will be selected or chosen by the user. It also helps to generate license key for a particular time period. The use of MAC address technique to generate license and time code keys will make provision for more security for the application and avoid copyright (Ruchi Rautela *et al.*, 2017). Various journal publications have seriously looked into the motive behind license choices and also gave certain

guidelines on how to go about selecting which license to use and how will it be good for the project. Lerner, J. *et al.* (2005) considers the scope of licensing in open source and lists the various areas of considerations which will determine the license of open source programs while Lindman, J. *et al.* (2010) in his own opinion, gives guideline on how to choose an open source license in a commercial context. Laurent, A.M.S. (2004), examines the licenses and their implications in great detail considering the conviction for choosing a particular license but also how to design an open source license. Lindberg, V., (2008) gives his own remark about trademark, patent, and copyright law how it relate to open source and display how to choose a commercial open source license. Under a specific country's copyright protection law, all developed software should be protected from the source to object code format against unauthorized access, unless it was developed by the United States government Raphael C. (2013). A Standard key generation technique, where the software license keys are generated mathematically, is not completely effective to put an end to illegal copying of software, as these. In addition, with improved communication from the advent of the Internet, more critical attacks on keys such as to remove the need for a key have made product key generators become more popular. As a result of this, software publishers use additional method or style to activate and verify the validity of key field. One of the techniques used was to assign a product key which is based on a special feature of the purchaser's computer hardware, which cannot be easily duplicated. Another method involves requiring one-time or periodical validation of the product key with an internet server which is done when the user log on. The server can block unrecognized client software presenting invalid or wrong keys while some experienced clients may bypass these checks, but the server can still deny those clients information or communication Chang *et al* (2002). Software license keys may be verified offline while the software is running or, better still, over the Internet through a license verification system which will be done online (Vangie Beal, 2011). Furthermore, serialization method was also used which saves the data content of your objects through the help of data output and data input in other to create License file that does not contain database. The above method need to capture, collect and arrange the entire objects that are interconnected back together on the other side but in case of any unforeseen circumstance in the process then the original data content cannot be unlocked (Anirudha A. *et al.*, 2015). Again, one wrong notion about software license generator is that the key is referred to as the license but it is only the right given to the software user that validates the user as the authorized customer. Moreover, a key generator that uses MAC address was used which will be supplied by the user during the operation of the program, and is then passed to a verification function in the program. This

function manipulates the key sequence according to a mathematical algorithm which attempts to match the results to a set of valid solutions. This License generator mainly focuses on providing ease and operational transparency to the Client in order to prevent software misuse (Vangie Beal, 2011). This proprietary software is secured with a serial number key which when it is typed in verifies the correctness of the software algorithmically, rather than searching through the entire database (Anirudha A. *et. al.*, 2015). The use of MAC address in generating license key eliminate the problem of serialization by providing a uniquely generated basic key, a uniquely generated advance key and a uniquely generated time code key for particular time period and stores the data contents in Database. The software generates a key by entering MAC Address & program mask for a particular program. The key generated by any cryptographic protocols to generate a code is used as an encryption key at one end of communication, and as an unlock key at the other Ruchi R. *et. al.*, (2017).

III. METHODOLOGY

3.1 Research Approach

In this research, efforts are made to explore many options like, consultations with relevant journals that treated software license key and other books. In all encryption systems, method of generating and encrypting key that is able to produce a secure cipher is required. A high-level view of the proposed encryption system is provided in this section. The main features used in the system are random number generation, verification of the generated values using statistical tests through the use of Eureqa to generate an equation. An appropriate method is required to generate and test random numbers to ensure that they have the necessary characteristics of independence and unpredictability. Eureqa model was implored to test the random numbers for reliability.

Eureqa models are a powerful addition to the Data Robot Automated Time Series product which can help you find human-readable mathematical formulas to explain the patterns in your data and build accurate, transparent, actionable models and solving real-world problems more faster than other approaches. The use of Eureqa models are the secret background operations that help to secure the secret key. But in other to generate any random number, linear congruential generator (LCG) was used to generate random numbers see figure 3.1. Linear congruential generator (LCG) is an algorithm that yields a sequence of pseudo-randomized numbers calculated with a discontinuous piece wise linear equation. It is a very simple example of a random number generator, fast and require minimal memory (one modulo-m number, often 32 or 64 bits) to retain state. This makes them valuable for simulating multiple independent streams. This

method represent one of the oldest and best known pseudo random number generator algorithms. The theory behind them is relatively easy to understand and they are easily implemented and fast, especially on computer hardware which can provide modular arithmetic by storage-bit truncation.

The function which is used in this method was proposed by D. H.Lehmerin 1948

$$X_{n+1} = (aX_n + C) \text{ mod } m$$

Where:

- X is the sequence of pseudorandom values, and
 - m, $0 < m$ is the "modulus" 3.1
 - a, $0 < a < m$ is the "multiplier" 3.2
 - c, $0 \leq c < m$ is the "increment" 3.3
 - X_0 , $0 \leq X_0 < m$ is the "seed" or "start value" 3.4
- are integer constants that specify the generator.

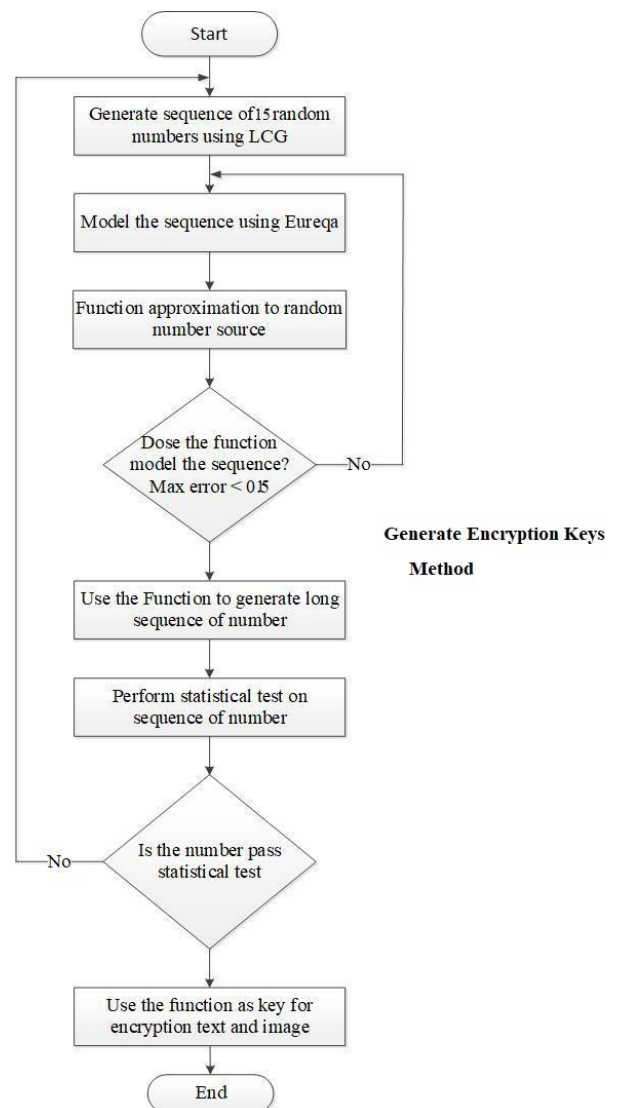


Figure 3.1: Method used to generate encryption keys

3.2 Enhanced Software License Key Generator

The proposed system facilitates a strong and secure encryption and decryption system.

The original data set were updated through Extended ASCII character encodings which are eight-bit encodings that include the standard seven-bit ASCII characters, plus additional character is used in this research to generate Accented characters. The ASCII code, abbreviated American Standard Code for Information Interchange, is a character encoding standard for electronic communication see figure 3.2. It is a code for representing English characters as numbers, with each character assigned a number from 0 to 127. ASCII codes represent text in computers, telecommunications equipment, and other devices. This tool generates random ASCII text from which you can choose the

length of generated text and how many results you want. You can also use a custom ASCII alphabet for generating random results to form Accented Characters, or to select predefined sets of characters for randomization see figure 3.3a.

In this research, a method of generating an encryption key that is able to produce a secure cipher is required. The enhanced key generator will be secured by AES cipher at the cache level while it is being queried by the database of the application to prevent unauthorized access but on the other hand it is an area of vulnerability for other license key generators. When an application is encrypted, the program generates license key and combination of basic key and advance key with some mathematical operation and its able to generate license for selected program. In this key generation, only the programmer can make changes to the encrypted system.

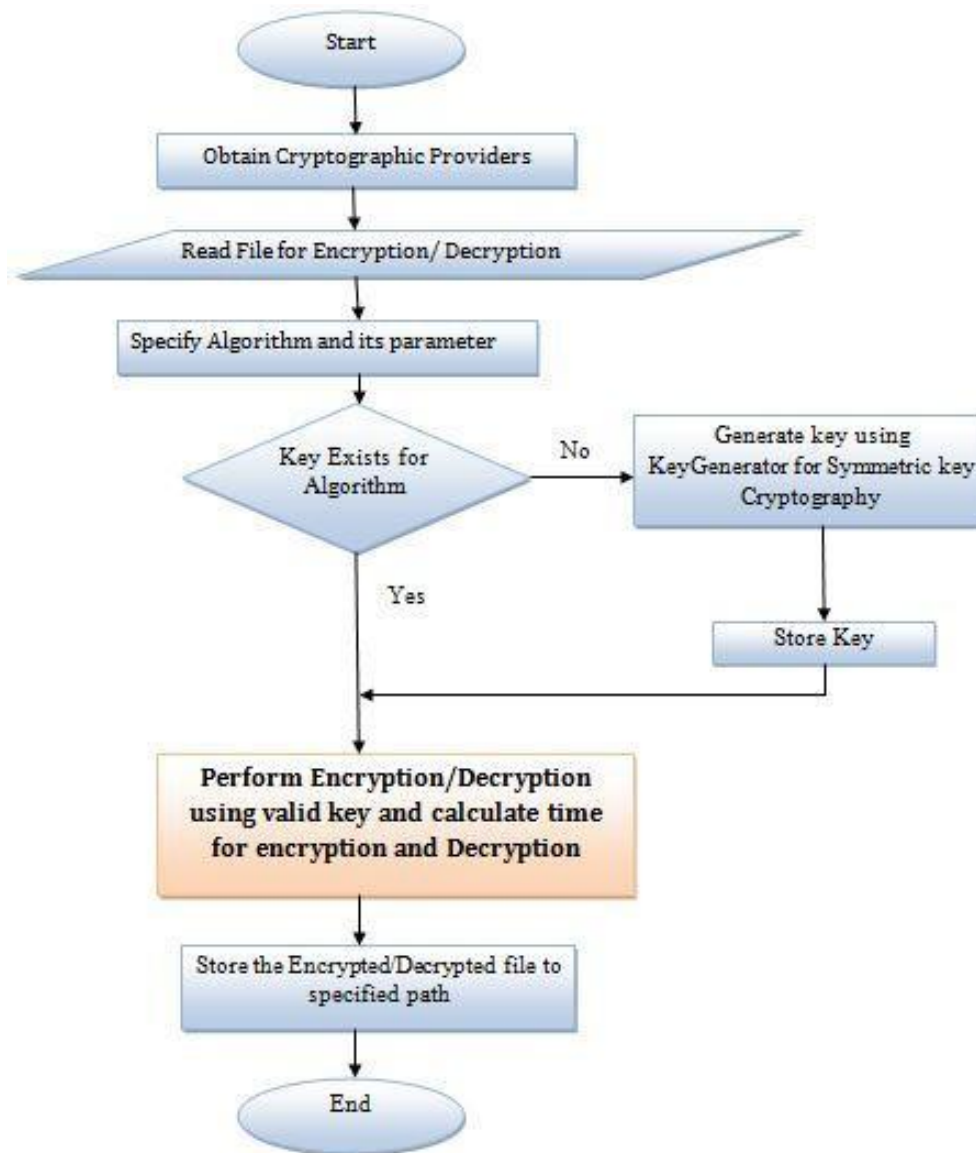


Figure 3.2: Symmetric key cryptography and flowchart

dec	hex	oct	char	dec	hex	oct	char	dec	hex	oct	char	dec	hex	oct	char
0	0	000	NULL	32	20	040	space	64	40	100	@	96	60	140	'
1	1	001	SOH	33	21	041	!	65	41	101	A	97	61	141	a
2	2	002	STX	34	22	042	"	66	42	102	B	98	62	142	b
3	3	003	ETX	35	23	043	#	67	43	103	C	99	63	143	c
4	4	004	EOT	36	24	044	\$	68	44	104	D	100	64	144	d
5	5	005	ENQ	37	25	045	%	69	45	105	E	101	65	145	e
6	6	006	ACK	38	26	046	&	70	46	106	F	102	66	146	f
7	7	007	BEL	39	27	047	'	71	47	107	G	103	67	147	g
8	8	010	BS	40	28	050		72	48	110	H	104	68	150	h
9	9	011	TAB	41	29	051	}	73	49	111	I	105	69	151	i
10	a	012	LF	42	2a	052	*	74	4a	112	J	106	6a	152	j
11	b	013	VT	43	2b	053	+	75	4b	113	K	107	6b	153	k
12	c	014	FF	44	2c	054	,	76	4c	114	L	108	6c	154	l
13	d	015	CR	45	2d	055	-	77	4d	115	M	109	6d	155	m
14	e	016	SO	46	2e	056	.	78	4e	116	N	110	6e	156	n
15	f	017	SI	47	2f	057	/	79	4f	117	O	111	6f	157	o
16	10	020	DLE	48	30	060	0	80	50	120	P	112	70	160	p
17	11	021	DC1	49	31	061	1	81	51	121	Q	113	71	161	q
18	12	022	DC2	50	32	062	2	82	52	122	R	114	72	162	r
19	13	023	DC3	51	33	063	3	83	53	123	S	115	73	163	s
20	14	024	DC4	52	34	064	4	84	54	124	T	116	74	164	t
21	15	025	NAK	53	35	065	5	85	55	125	U	117	75	165	u
22	16	026	SYN	54	36	066	6	86	56	126	V	118	76	166	v
23	17	027	ETB	55	37	067	7	87	57	127	W	119	77	167	w
24	18	030	CAN	56	38	070	8	88	58	130	X	120	78	170	x
25	19	031	EM	57	39	071	9	89	59	131	Y	121	79	171	y
26	1a	032	SUB	58	3a	072	:	90	5a	132	Z	122	7a	172	z
27	1b	033	ESC	59	3b	073	;	91	5b	133	[123	7b	173	{
28	1c	034	FS	60	3c	074	<	92	5c	134	\	124	7c	174	
29	1d	035	GS	61	3d	075	=	93	5d	135]	125	7d	175	}
30	1e	036	RS	62	3e	076	>	94	5e	136	^	126	7e	176	~
31	1f	037	US	63	3f	077	?	95	5f	137	_	127	7f	177	DEL

Figure 3.3: Standard ASCII Character chart

Accented letters

á	alt-0225	Á	alt-0193	à	alt-0224	À	alt-0192	â	alt-0226	Â	alt-0194	ä	alt-0228	Ä	alt-0196
é	alt-0233	É	alt-201	è	alt-0232	È	alt-0200	ê	alt-0234	Ê	alt-0202	ë	alt-0235	Ë	alt-203
í	alt-0237	Í	alt-205	ì	alt-0236	Ì	alt-0204	î	alt-0238	Î	alt-0206	ï	alt-0239	Ï	alt-0207
ó	alt-0243	Ó	alt-0211	ò	alt-0242	Ò	alt-0210	ô	alt-0244	Ô	alt-0212	ö	alt-0246	Ö	alt-0214
ú	alt-0250	Ú	alt-0218	ù	alt-0249	Ù	alt-0217	û	alt-0251	Û	alt-0219	ü	alt-0252	Ü	alt-0220
ý	alt-0253	Ý	alt-0221	ÿ	alt-0229	ÿ	alt-0197	ž	alt-0142	Ž	alt-0158	ÿ	alt-0255	ÿ	alt-0159
ã	alt-0227	Ã	alt-0195	ø	alt-0248	Ø	alt-0216	æ	alt-0230	Æ	alt-0198	ð	alt-0240	Ð	alt-0208
ñ	alt-0241	Ñ	alt-0209	ç	alt-0231	Ç	alt-0199	œ	alt-0156	Œ	alt-0140	þ	alt-0254	Þ	alt-0222
õ	alt-0245	Õ	alt-0213	š	alt-0154	Š	alt-0138	ß	alt-0223	Press 'alt' plus the number (on the number pad) to type these letters					

Figure 3.3a: Accented letters

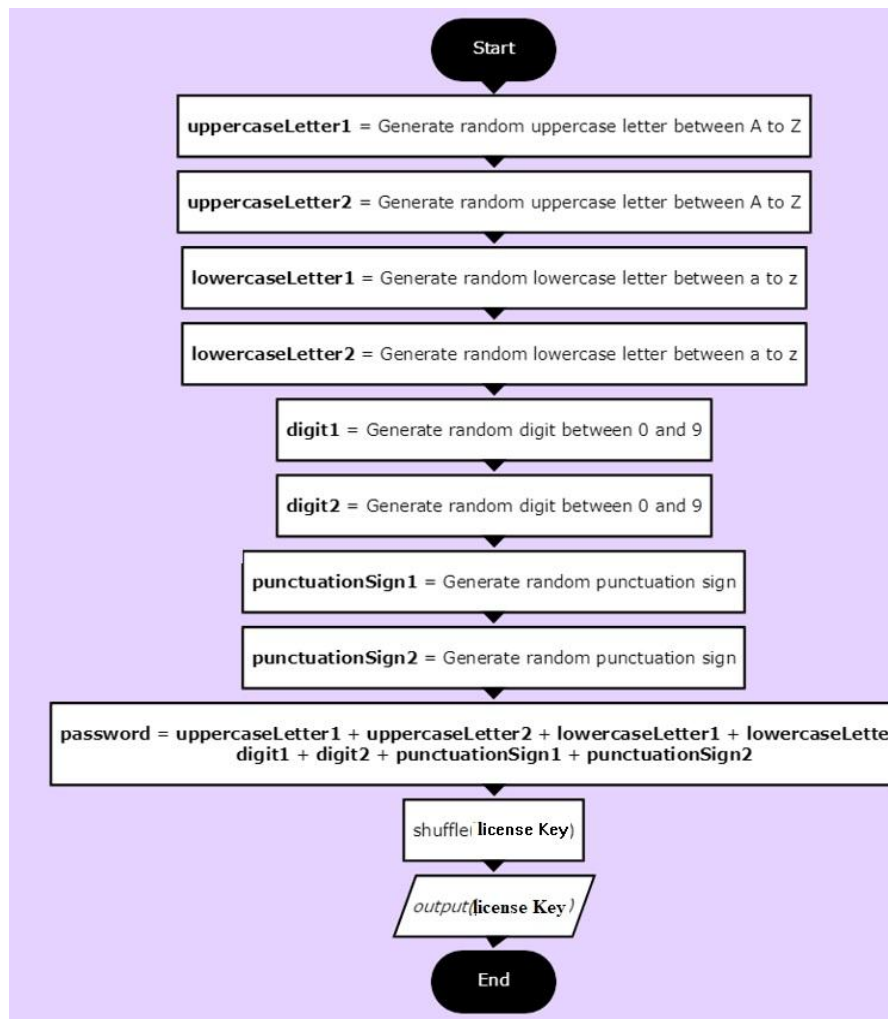


Figure 3.4: Random password generator algorithm Flowchart

3.4 Performance Metrics

Software license key is a data string that verifies the authorization of a product user legal access. It is an important aspect of software development in the sense that it ensures that an application is secured or protected against unauthorized access. This research is designed in such a way that it will be difficult to break the key generated due to the random symmetric cryptography used and to generate the key will require a very short time considering the random means of selecting the numbers or letters. The vulnerability or loopholes of the software was also checked by using Eureka data robot to simulate it. This key is designed to unlock software once and registers the details of that system in its database thereby preventing multi use by other user. The new system in its implementation secures the cache memory by encrypting its content through the use of AES cipher encryption. It also support both accented and Latin set of characters. This new design encourages more participation in license key generation.

IV. RESULT

4.1 Result

The implementation of the enhanced algorithm that generate license key is done using secure symmetric key cryptography algorithm. This algorithm is design in such a way that the strings of numbers or letters will be picked randomly and verified before it is being accepted. The key generator software gives fast performance; establish relationship between the vendor and customer of the product. In addition, the license key cannot be used illegally and could not be crack because it is secured by AES cipher encryption at the cache level before it is moved to the database for onward validation. The enhanced system is designed to accept both accented and other types of characters such as Yoruba, Latin set and it is to be used on a system per use; this is done by registering the details of the system in its memory the moment it is activated. The programming language used to implement this research work is C#, java scripts and Visual basic.Net.

4.2 License Key Generation

Generation of serial numbers and license keys of software products

The set of sources for automated generation of license keys for software products, allow using string templates of keys with set of required key tokens, checksum values for determining authenticity of generated keys and random data which provide more security to the software.

Launching Key gen.

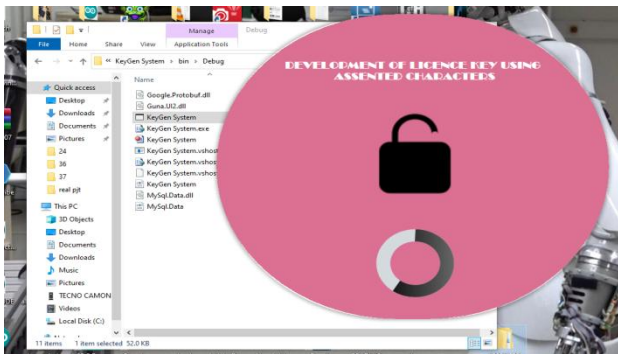


Figure 4.1: Key generated in General Format

4.3 Concepts

For generation of license key, there is need to create string of license template with ASCII characters and add to each token of template initial value. It is demonstrated on screenshot below:

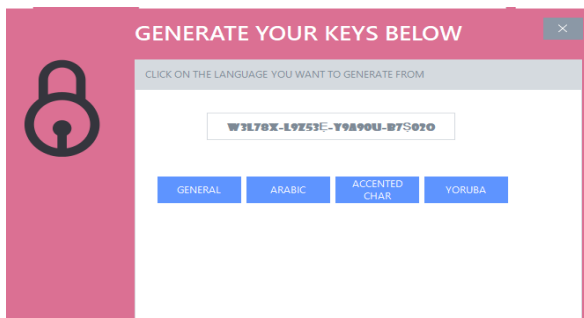


Figure 4.2: Generating the license keys

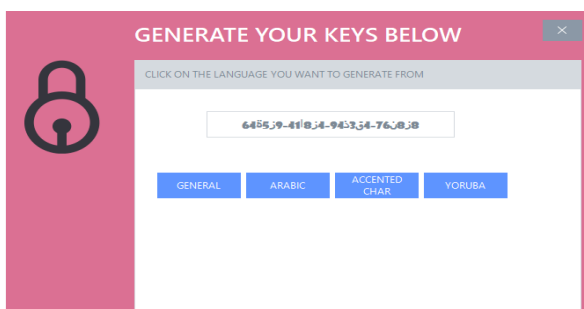


Figure 4.3: Key generated in accented Arabic Character

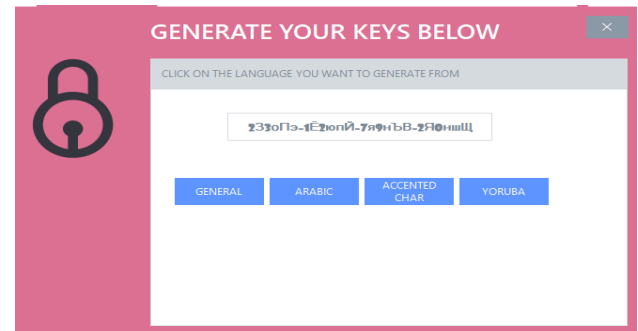


Figure 4.4: Key generated in accented Character with Country language

V. CONCLUSION

Over the years, the Latin set of characters for only English speaking countries was used to develop license key generator which had prevented other non-English speaking countries from participating. Multi language key generator using accented characters is a technique which is developed to eliminate the limitation which was caused due to the non-inclusion of other countries languages or set of characters in the development of license key. The enhanced license key is more secure as it incorporates AES cipher encryption to secure the key, gives fast performance in its implementation. Most importantly, cannot be cracked and copied illegally. This dissertation is expected to improve productivity and encourage more participation.

REFERENCES

- [1] Anirudha A. Kolpyakwar, Piyush M. Dhande, Pallavi A. Chaudhari (2015): "License File Generator Using MAC Address", International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 6. <https://www.irjet.net/archives/V4/i6/IRJET-V4I6719.pdf>
- [2] <https://www.datarobot.com>
- [3] Laurent A.M.S, (2004): "Understanding Open Source and Free Software Licensing", O'Reilly Media, first edition, annotated edition.
- [4] Lerner, J., Tirole, J. (2005): "Examines the scope of licensing in open source and lists the various considerations that determine the license of open source projects".
- [5] Lindberg, V. (July 2008): "Intellectual Property and Open Source: A Practical Guide to Protecting Code", O'Reilly Media, 1 edition.
- [6] Rotenberg, A. (1960). "A New Pseudo-Random Number Generator". Journal of the ACM. page: 75-77. doi:10.1145/321008.321019. S2CID 16770825.
- [7] Ruchi Rautela, Raul Parandwal, Pankaj Upadhyay (2017): "License generator using MAC address for

industrial application”, www.irjet.com, volume 04, issue 06 June.

[8] Vangie Beal (2011): www.webopedia.com/TERM/L/license_generator.html

[9] Holzner, S., Stallings W., (1999): “Cryptography and Network Security”, Visual Basic .NET, Black book, 2nd Edition, Prentice Hall.

Citation of this Article:

Olawale John A, Moradeyo Oluwatomilola M, Ojoawo Akinwale O, Olaniyan Abolade S, & Adeyeye A. Hannah. (2024). Multilanguage Software License Key Generator: Accented Characters Approach. *International Research Journal of Innovations in Engineering and Technology - IRJIET*, 8(6), 176-183. Article DOI <https://doi.org/10.47001/IRJIET/2024.806021>
