Passphrase with Semantic Noises and a Proof on Its Higher Information Rate

Kok-Wah Lee
Faculty of Engineering & Technology
Multimedia University
75450 Bukit Beruang, Malaysia.
kwlee@mmu.edu.my

Hong-Tat Ewe
Faculty of Information Technology
Multimedia University
63100 Cyberjaya, Malaysia
htewe@mmu.edu.my

Abstract

Key size becomes very important to a cryptographic algorithm according to Kerckhoff's law where a civilian cryptosystem shall depend fully on key secrecy. Currently, there are four passphrase generation methods: Sentence, acronym, diceware, and coinware. Unicity distance is the minimum size of ciphertext for unique decipherability of ciphertext when number of spurious keys is zero. A key with size less than unicity distance is good where there are spurious keys which allow a protection method using limited unsuccessful logins. Here, stronger forms of passphrases using textual semantic noises like punctuation marks, mnemonic substitution, misspelling, and associative morphing, which improve the key entropy, are proposed. An ASCII mutual substitution table is presented together with its proof on information rate increment. Higher information rate has lower redundancy, and hence bigger unicity distance ensures encrypted keys the short cryptogram in a key vault, like Password Safe, cannot be cryptanalyzed within certain limited login attempts.

Keywords: Key security, passphrase, semantic noises, unicity distance, ASCII mutual substitution table.

1. Introduction

Key is one of the four main groups of entity authentication for identification. These four groups are "something known", "something possessed", "something inherent", and "someone known" [1]. Key is a secret only known to the authenticated entity, where its low cost, mobility, and wide compatibility, makes it to be the most popular authentication method.

Kerckhoff's law is applied in civilian cryptosystem, where the strength of a cryptosystem is fully dependent on the key secrecy [2]. In other words, key size is the main factor of a cryptographic algorithm. Short key is called password and long key is called passphrase. FIPS PUB 112 (Federal Information Processing Standards Publication 112) dated 30 May 1985 on password usage [3] defined password to be a key with a length of 4 to 8 characters and passphrase as a key with a length of 9 to 64 characters.

To estimate the key entropy, NIST (National Institute of Standard and Technology), USA, published an electronic authentication guideline [4]. However, the estimation of user-chosen key entropy is based on Shannon's English information rate [5-6] using 26 English alphabets plus one space character. For more accurate figures, the English information rate is in fact shall be based on 95 ASCII printable characters, where its limiting conditional entropy is available in [7-8].

A good key shall be strong and memorizable. A strong key has high entropy and high randomness. Meanwhile, a memorizable key has reasonable secrets to be remembered. Weak key is not in favourite [9-11]. Random key is strong but not memorizable. Hence, the researchers proposed keys with balanced features of strength and memorability [12-14].

Due to the long key size demand for symmetric key algorithm and asymmetric key algorithm [15-16], password is no longer enough and we need passphrase [1-2, 17]. For the popular email encryption software, PGP (Pretty Good Privacy) version 9.0, the allowed key size comes to a maximum of 255 characters [18]. Now, many modern operating systems support a maximum key field of 255 characters.

For the entry of passphrase, there are currently four input methods: Sentence, acronym, diceware [18] and coinware [19]. In this paper, a stronger form of passphrase is proposed to have more spurious keys by using the semantic noises or semantic errors [20] like punctuation marks, misspelling, mnemonic substitution and associative morphing [12].

In the following sections, Section 2 discusses on key sizes and passphrases. Section 3 explains the unicity distance and presents the stronger forms of passphrases using semantic noises. Section 4 gives an ASCII mutual substitution table together with its proof of higher information rate. Finally, Section 5 concludes this article.

2. Key sizes and passphrases

The minimum symmetric key sizes for different protection periods are given by [15-16]. We present again a table from [16] for easy reference as in Table 1.

Table 1. Minimum symmetric key sizes for different protection periods

Security Level	Security (bits)	Protection
1	32	Only acceptable for authentication tag size.
2	64	Very short-term protection.
3	72	Short-term protection.
4	80	Smallest general-purpose protection for ≤ 4 years.
5	96	Legacy standard level for 10-year protection.
6	112	Medium-term protection for 20 years.
7	128	Long-term protection for 30 years.
8	256	Foreseeable future. Good protection against quantum computers.

The ASCII entropy is 6.57 bits/letter. Hence, many key sizes are challenging the human memorability limit. For asymmetric key algorithm, the long key sizes make it impossible to be memorizable in ASCII encoding code and require encrypted private key stored in the computing device. A user has to remember the shorter symmetric key used to protect the encrypted private key. Hence, asymmetric key algorithm is normally having the portability problem of private key.

From Table 1, the minimum key size for smallest general purpose level with a maximum of 4-year protection is 80 bits. If ASCII is used, it needs 13 characters to fulfill the key size requirement. Hence, passphrase shall be used as compared to password.

There are four types of passphrase generations with examples given in [19]: Sentence, acronym, diceware [18] and coinware [19]. Sentence-type and acronym-type passphrases are subject to computational analysis of word frequency distribution. Meanwhile, diceware and coinware are immune to the computational analysis due to the feature of random word selection.

Sentence-type passphrase uses an entire phrase or full sentence to form a key. Acronym-type passphrase applies abbreviation of first, second, last, etc., letters of each word in a sentence. Diceware uses dice to choose a word from an ordered word list. The word list can be in any language and based on senary or base-6 numeral system. Coinware is similar to diceware except it uses coin to select from a word list that can be monolingual, bilingual, or multilingual. It is especially efficient for word list in binary, octal, and hexadecimal orders. There are readily built word lists for Han characters in Unicode-encoded CJK languages.

3. Unicity distance and passphrases with semantic noises

Capitalization and *permutation* are the prior arts to increase the passphrase entropy [17]. *Mnemonic substitution* and *associative morphing* are other forms of prior arts. The latter two methods are presented in very brief manner without a proof [12]. Here, we generalize these four methods together with another two methods from us, i.e. *punctuation marks* and *misspelling*, as passphrase with semantic noises, and propose a user template of ASCII mutual substitution table, which comes together with a proof on the information rate increment.

Passphrase with semantic noises has higher information rate (r), lower redundancy (D), and hence bigger unicity distance (n_0). Unicity distance [21-22] is the minimum ciphertext size for unique decipherability of ciphertext given sufficient decryption time, when number of spurious keys (F) is zero. The larger the difference between the unicity distance and key length, the more the spurious keys, and the stronger is the protection method using limited login attempts. Special key management algorithms allow multiple site keys to be created from a master key [23-24]. This further permits each short cryptogram of keys in a key vault like Password Safe to be encrypted by different site keys. The decoding of short cryptogram is studied in [25]. Faster decryption can be achieved using [26].

The relationships of information rate, absolute rate like random signal (R), key entropy (H(K)), ciphertext size (n), and unicity distance are given in Eqs. (1-4).

$$D = R - r \tag{1}$$

$$F \geq 2^{H(K) - nD} - 1 \tag{2}$$

$$n_0 \geq H(K) / D \qquad \text{when } F = 0 \tag{3}$$

$$r \uparrow \; \Rightarrow \; D \downarrow \; \Rightarrow \; F \uparrow, n_0 \uparrow. \tag{4}$$

For instance, English text has r = 1.3 bits/letter for 27 symbols (a, b, c, …, z, space) [5-6]. The redundancy is R = 4.75 bits. If AES-128 is used, H(K) = 128 bits. Hence, n_0 = 128/3.45 = 38 characters. For 95 ASCII printable characters, the upper bound of r based on English language becomes 1.75 bits/letter. The revised n_0 = 128/(8-1.75) = 21.

Below are the examples of passphrases with semantic noises using punctuation marks, misspelling, mnemonic substitution, associative morphing, capitalization, and permutation. Punctuation mark is the easiest. A user is encouraged to embed semantic noise for all types of passphrases: Sentence, acronym, diceware, and coinware. The presence of spurious keys is very useful for the case of limited login attempts, where unique cryptanalysis is impossible.

Actual key: Woman without her man is a savage.
Semantic noises: *Punctuation marks* and *permutation* {
Woman without her, man is a savage.
Woman without her man, is a savage.
Woman, without her man is a savage.
Woman without her, man is a savage?
Woman without reh man, si a savage?
Woman, without reh man si a savage?
Woman without her man is a savage!
Woman without her, man is a savage!
Woman without reh man, si a savage!
Woman, without reh man si a savage! }

Actual key: To be, or not to be: That is the question.
Semantic noises: *Misspelling* and *capitalization* {
To be? or not to be? That is the question?
To be, or not to be? that is the question!
T0 6e? 0r n0t t0 6e? th@t i5 the que5ti0n?
To be! Or not to be! That is the question!
TO BE! OR NOT TO BE! THAT IS THE Question!
To we, or not to we: That is the question.
To me, or not to me: That is the question.
To be, of not to be: That is the question. }

Actual key: Ballon, Address? Atmel. ~Star~
Semantic noise: *Mnemonic substitution* {
B@!!0n, Address? Atmel. ~Star~
Ballon, Address? @mail. ~Star~ }
Semantic noise: *Associative morphing* {
Ballon, +++re$$? Atmel. ~Star~
Ballon, Address? Atmel. ~****~
B@!!0n, +++re$$? @mail. ~****~ }

4. Proof of higher information rate

Here, we present an ASCII mutual substitution table as user template to create passphrase with semantic noises. A user can modify any mutual substitution of these ASCII characters in Table 2. CamelCase makes compound words or phrases in which the words are joined without spaces and are capitalized within the compound [27]. The ASCII substitution is a token with one or more characters. The probability of the initial token letter is used as the token probability, where we assume that the difference is small and negligible.

Table 2. ASCII mutual substitution table

aA	bB	cC	dD	eE	fF	gG	hH	iI	jJ
^	6	<	o|	3	|=	9	|-|	!	?
kK	lL	mM	nN	oO	pP	qQ	rR	sS	tT
|<	1	TV|	TV	0	|o	&	|-	5	+
uU	vV	wW	xX	yY	zZ	0	1	2	3
[_]	\/	vv	><	`/	2	O	I	Z	E
4	5	6	7	8	9	+	-	*	/
h	S	b	L	B	g	t	_	x	|
%	=	[	]	{	}	(	)	<	>
o/o	eq	{	}	[	]	<	>	(	)
!	“	#	$	&	‘	,	.	:	;
i	,,	n	m	Q	,	‘	* dot	;	:
?	@	\	^	_	`	|	~	space	
j	at	`	A	-	\	/	^v	CamelCase	

The upper bound of information rate (r) [5-6] is given by Eqs. (5-6), where $q^N{}_i$ is the probability for predictor to discover the correct letter following a sequence of N-1 symbols in i guesses. i indexes one of the 95 ASCII printable characters.

$$q_i^N = \sum p(j_1, j_2, \ldots, j_{N-1}, j_N) \tag{5}$$

$$\sum_{i=1}^{95} i(q_i^N - q_{i+1}^N)\log_2 i \le r \le -\sum_{i=1}^{95} q_i^N \log_2 q_i^N \tag{6}$$

Due to the mutual substitution of ASCII printable characters, 95 $q^N{}_i$ becomes about 47 pairs. Every two different $q^N{}_{i-}$ with different probabilities are paired to share the same probability. Let one of these pairs has probabilities A and B before mutual substitution, and probability C after mutual substitution. Let other ASCII characters have a combined probability D, where (A + B + D = 1) and ((A + B)/2 = C). The inequality Eq. (7) can be proven using differentiation of calculus dy_2/dA on Eq. (11) derived from Eqs. (7-10). y_1 is a constant and y_2 has an absolute minimum value equaling to y_1 at A = B = (1 – D)/2. The other two critical points, A = 0 and 1 – D, share the same absolute maximum value.

$$-2C * \log_2 C \ge -A * \log_2 A - B * \log_2 B \tag{7}$$

$$C^{2C} \le A^A B^B \tag{8}$$

$$((1-D)/2)^{1-D} \le A^A (1-D-A)^{1-D-A} \tag{9}$$

$$y_1 = ((1-D)/2)^{1-D} \tag{10}$$

$$y_2 = A^A (1-D-A)^{1-D-A} \tag{11}$$

Inequality (6) can be further extended to three or more mutually substituted ASCII characters for higher information rate increment. The best case is all the

ASCII characters can be mutually substituted, which creates the highest information rate like absolute rate the random signal, where unicity distance will become infinite. However, this is just an ideal dream. What we can do is to approach the dream as close as possible. In Eq. (4), higher information rate has more spurious key.

5. Conclusions

Here, stronger form of passphrase is proposed using semantic noises generalizing the punctuation marks, capitalization, permutation, mnemonic substitution, associative morphing, and misspelling. Passphrase with semantic noises has higher information rate, bigger unicity distance, and more spurious keys, which strengthens the login protection with limited attempts. In addition, an ASCII mutual substitution table and its proof on information rate increment is provided.

6. References

[1] A.J. Menezes, P.C.V. Oorschot, and S.A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Boca Raton, FL, USA, 1996.

[2] B. Schneier, *Applied Cryptography*, John Wiley & Sons, New York, NY, USA, 1996.

[3] NIST, *FIPS Pub 112 Password Usage*, CSRC (Computer Security Resource Center), NIST (National Institute of Standards and Technology), Gaithersburg, MD, USA, May 30, 1985.

[4] W.E. Burr, D.F. Dodson, and W.T. Polk, *Electronic Authentication Guideline*, NIST Special Publication 800-63, CSRC (Computer Security Resource Center), NIST, Gaithersburg, MD, USA, June 2004.

[5] C.E. Shannon, "Prediction and Entropy of Printed English", *Bell System Technical Journal*, vol. 30, no. 1, American Telephone and Telegraph Company, New York, NY, USA, January 1951, pp. 50-64.

[6] T.M. Cover, and R.C. King, "A Convergent Gambling Estimate of the Entropy of English", *IEEE Transactions on Information Theory*, vol. 24, no. 4, IEEE, Piscataway, NJ, USA, July 1978, pp. 413-421.

[7] P.F. Brown, V.J.D. Pietra, R.L. Mercer, S.A.D. Pietra, and J.C. Lai, "An Estimate of an Upper Bound for the Entropy of English", *Computational Linguistics*, vol. 18, no. 1, MIT Press, Cambridge, MA, USA, March 1982, pp. 31-40.

[8] B.K. Tsou, T.B.Y. Lai, and K.-P. Chow, "Comparing Entropies within the Chinese Language", *Proceedings of 1st International Joint Conference on Natural Language Processing (IJCNLP 2004)*, LNCS 3248, Hainan Island, China, March 22-24, 2004, pp. 466-475.

[9] D.V. Klein, ""Foiling the Cracker": A Survey of, and Improvements to, Password Security", *Proceedings of USENIX 2nd Workshop on Security*, USENIX, Portland, OR, USA, August 1990, pp. 5-14.

[10] E.H. Spafford, "Opus: Preventing Weak Password Choices", *Computers & Security*, vol. 11, no. 3, Elsevier Advanced Technology, Oxford, Oxon (Oxfordshire), England, UK, May 1992, pp. 273-278.

[11] E.H. Spafford, "Observations on Reusable Password Choices", *Proceedings of USENIX 3rd Symposium on UNIX Security*, USENIX, Baltimore, MD, USA, September 1992, pp. 299-312.

[12] S.V. Bugaj, "Passwords for Real Humans", *USENIX Login*, vol. 21, no. 3, Berkeley, CA, USA, June 1996, pp. 41.

[13] S.A. Kurzban, "Easily Remembered Passphrases – A Better Approach", *ACM SIGSAC Review*, (SIGSAC: Special Interest Group on Security, Audit and Control), vol. 3, no. 2-4, ACM, New York, NY, USA, Fall/Winter 1985, pp. 10-21.

[14] J. Yan, A. Blackwell, R. Anderson, and A. Grant, "Password Memorability and Security: Empirical Results", *IEEE Security and Privacy*, vol. 2, no. 5, IEEE Computer Soc., Los Alamitos, CA, USA, September 2004, pp. 25-31.

[15] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid, *Recommendation for Key Management – Part 1: General (Revised)*, NIST Special Publication 800-57, CSRC (Computer Security Resource Center), NIST, Gaithersburg, MD, USA, 2006, ch. 5, pp. 61-71.

[16] C. Gehrmann, and M., Näslund (Ed.), *ECRYPT Yearly Report on Algorithms and Keysizes (2006)*, European Network of Excellence in Cryptology (ECRYPT), Katholieke Universiteit Leuven, Leuven-Heverlee, Belgium, 2006.

[17] W. Stallings, *Protect Your Privacy: A Guide for PGP Users*, Prentice Hall, Englewood Cliffs, NJ, USA, 1995.

[18] PGP Corporation, *PGP Desktop 9.0 for Windows User's Guide*, PGP Corporation, Palo Alto, California, USA, 2006.

[19] K.W. Lee, and H.T. Ewe, "Coinware for Multilingual Passphrase Generation and Its Application for Chinese Language Password", *Proceedings of the 2006 International Conference on Computational Intelligence and Security (CIS 2006)*, Guangzhou, Guangdong, China, November 3-6, 2006, pp. 1511-1514 (Part 2).

[20] K.W. Lee, "Semantic Error Occurrences in the Multimedia Communications", *Proceedings of IASTED 2005 International Conf. on Education & Technology (ICET2005)*, Calgary, Alberta, Canada, July 2005, pp. 227-231.

[21] D.R. Stinson, *Cryptography: Theory and Practice*, Chapman & Hall/CRC Press Boca Raton, FL, USA, 2002.

[22] S.S. Wagstaff Jr., *Cryptanalysis of Number Theoretic Ciphers*, Chapman & Hall/CRC Press, Boca Raton, FL, USA, 2003, ch. 8, pp. 115-117.

[23] K.P. Yee, and K. Sitaker, Passpet: Convenient Password Management and Phishing Protection, Proceedings of Symposium on Usable, Privacy and Security (SOUPS2006), Pittsburgh, PA, USA, July 12-14, 2006, p. 32-43.

[24] K.W. Lee, and H.T. Ewe, "Multiple Hashes of Single Key with Passcode for Multiple Accounts", *Journal of Zhejiang University Science A (JZUS-A)*, vol. 8, no. 8, August 2007, pp. 1183-1190.

[25] G.W. Hart, "To Decode Short Cryptograms", Communications of the ACM, vol. 37, no. 9, ACM, New York, NY, USA, September 1994, pp. 102-108.

[26] K.W. Lee, C.E. The, and Y.L. Tan, "Decrypting English Text Using Enhanced Frequency Analysis", 2006 National Seminar on Science, Technology and Social Sciences (STSS2006), Kuantan, Pahang, Malaysia, May 30-31, 2006.

[27] Wikipedia Contributors, "CamelCase", Wikipedia the Free Encyclopedia, revision 22 May 2007 14:10 UTC, accessed 1 June 2007.