Enhanced SHA-1 on Parsing Method and Message Digest Formula

Christine Charmaine G. San Jose
Graduate Programs
Technological Institute of the Philippines
Quezon City, Philippines
tinsanjose11@gmail.com

Bobby D. Gerardo
Institute of Information and
Communications Technology
West Visayas State University
Lapaz, Iloilo City, Philippines
bgerardo@wvsu.edu.ph

Bartolome T. Tanguilig III
College of Information Technology Education
Technological Institute of the Philippines
Quezon City, Philippines
bttanguilig_3@yahoo.com

ABSTRACT

The Secure Hash Algorithm is one of the most commonly used hashing algorithms on the time being. Experts are proposing for much secured SHA-1 because of some reports and study conducted on SHA-1 collision attacks. The aim of the enhanced Secure Hash Algorithm -1 (SHA-1) is to strengthen its original version that is expected to resist possible SHA-1 collision attacks. The enhanced SHA-1 had integrated the following modification: 1) Enhancement on the pre-processing, specifically on the parsing method. 2) The message digest and the final message digest formula was enhanced by giving additional shifting, xoring and improved mathematical formula. The enhanced SHA-1 maintains its original rounds which consist of 80 rounds and message digest output of 160 bit. Based from the result, the enhanced algorithm specifically on mathematical calculation of parsing method and message digest had shown great effects on its result on the hash value despite of a very minimal time delay, the enhanced algorithm is better and more secure.

KEYWORDS

Message Digest, Hash Function, Parsing Method, Cryptography, Digital Signature

1 INTRODUCTION

As perceived by R. Rivest [1], information-processing telecommunication revolution has paved the way in the twentieth century. In the advent of internet technology, many had switch into adopting the technology because of real-time means of communication. Despite of many benefits that can be obtained from on-line technology, problems on security and threats have emerged.

Cryptography used hashing as another means of technology which can be implemented as add-ons for security issues. Hashing functions [2], [3] are applied into many applications such as digital signature, storing password file, key derivation and many others.

Digital signature had played an important role in the recent technology because it provides integrity, authentication and undeniability and could give solution to the four elements of security: the confidentiality, authenticity, integrity and availability. Digital signature is being utilized in electronic commerce where the need to protect sensitive information such e-mails and financial transaction are the main concern.

The hash function [1] is practically difficult to invert because of its one-way property. Moreover, a good cryptographic hash function [10] should preserve the following: efficiency, fast processing time, that a hash function is a pre-image resistant meaning no one could produce the input message based from the given hash value and it should be 2nd pre-image resistant meaning that no one could produce two different documents that have the same hash.

Many hashing technique were developed, among them are: DMDC (Des-like Message Digest Computation), MD5 (Message Digest 5), HMAC (Hashed Message Authentication Code) and SHA (Secure Hash Algorithm).

The National Institute of Standard and Technology (NIST) had developed the Secure Hash Algorithm (SHA) which is then used for Digital Signature Algorithm (DSA). It was published in the year 1993 as a Federal Information Processing Standard [3].

There has been an identified security flaws in SHA-1 [7], [8], [9] this is due to some problems on the existing algorithm. According to experts there is a need for a much powerful hash function because of some weaknesses on its mathematical function. Yiqun Lisa Yin [11] was able to exploit SHA-1 and announces its two weaknesses: the pre-processing steps and problem on its math operation on the first twenty rounds.

Inspired by R. Rivest on [1], which states that every theoretical work is refined and improved through practice and every practice challenges a theoretical work.

With these, the author is proposing for the development of enhanced SHA-1 algorithm. This study will simulate the original SHA-1 algorithm and determine its weaknesses; to enhance the secure hash algorithm specifically on parsing method and message digest formula and to be able to evaluate the performance of the enhanced algorithm in terms of processing time and security.

2 REVIEW OF RELATED LITERATURE

T. Lakshmanan and M. Muthusamy proposed new Secure Hash Algorithm called SHA-192. The original SHA-1 introduced by NIST produces 160 bit message digest while the proposed SHA-192 produce a 192 output length. The authors made some revision to its original function and observed that SHA-192 to be better than the existing SHA-1 hashing algorithm in terms of number of brute force attack but in terms of time performance of the algorithm the proposed SHA-192 has a time delay since it needs to generate a 192 bit of message digest [2].

A digital signature consists of a mathematical calculation that demonstrates the authenticity of a message. Dr. Herong Yang [4] discussed the use of hash algorithm in a digital scheme for e-mail messages.

Bruce Schneier discussed the need to enhance the SHA-1. He state that in 2005, cryptanalysts found attacks on SHA-1 suggesting that the algorithm might not be secure enough for ongoing use. He further suggest that NIST orchestrate a worldwide competition for a new hash function. He also made emphasis that NIST should issue a call for algorithms, and conduct a series of analysis rounds, where the community analyzes the various proposals with the intent of establishing a new standard [5].

M. Alam and S. Ray made use of CPSO (Canonical Particle Swarm Optimization) approach in the enhancement of Secure Hash Algorithm-1. The scheme consists of prediction control block which takes the message stream from user and provides a log-list with an equal length with the message stream. The prediction scheme does impede the CPU utilization a bit but the author is confident that the new scheme will create a new venue in designing cryptographic hash function [6].

One study conducted [11] on designing cryptographic hash function is to consider its *"avalanche effect"*. The term was created by Horst Fiestel, meaning that any single changes made from the input message could drastically affect the output of hash value.

3 SECURE HASH ALGORITHM

3.1 SHA-1 General Properties

Pre-processing

There are three basic steps involve in the pre-processing stage. These are the following: *Message Padding, Parsing Method and Initializing of the 160 bit buffer*. This is performed in order to prepare the message for further mathematical calculations.

The First Step: Message Padding
The goal of message padding is to make the final padded message a multiple of 512 bits. Message padding involves three parts.

- Padding the Original Message by adding one "1" at the end of the message.
- Adding many "0's" to form 512 bits of message length.
- Appending 64 bit integer at the end of the zero appended message to form the final padded message. This is performed by determining the length of the original message in bits. The bits value is converted into hexadecimal value which is then appended at the end of the message to form the final 512 bits.

The Second Step: Parsing Method
The parsing method is simply performed by dividing the final padded message consisting of 512 bits into sixteen 32 bit words or blocks from M_0, M_1 . . .M_{15}.

The Third Step: Initializing the 160-bit buffer.
The 160-bit buffer consist of five 32 bit registers (A, B, C, D and E).

H0 = 67 45 23 01
H1 = ef cd ab 89
H2 = 98 ba dc fe
H3 = 10 32 54 76
H4 = c3 d2 e1 f0

Functions Used

The set of SHA primitive functions, ft (B, C, D) is defined as follows:

ft (B, C, D) = (B • C) + (B • D), $0 \leq t \leq 19$ (1)
ft (B, C, D) = B $\oplus$ C $\oplus$ D, $20 \leq t \leq 39$ (2)
ft (B, C, D) = (B • C) + (B • D) + (C · D), $40 \leq t \leq 59$ (3)
ft (B, C, D) = B $\oplus$ C $\oplus$ D, $60 \leq t \leq 79$ (4)

where B • C = B and C
B $\oplus$ C = B xor C
B = Complement of B
+ = addition modulo 2^{32}

Constant Used

There are four values of constant to be used which is in hexadecimal value.

K_t=5a827999, $0 \leq t \leq 19$
K_t=6ed9eba1, $10 \leq t \leq 39$
K_t=8f1bbcdc, $20 \leq t \leq 59$
K_t=ca62c1d6, $30 \leq t \leq 79$

ISBN: 978-1-9491968-07-9

Computing the Message Digest

The message digest or also known as the processed message by using mathematical calculation is generated by using the final padded message. As discussed in 3.1 SHA General Properties, the pre-processing involves three basic steps: the message padding, message parsing and initializing 160-bit buffer. Both padding and parsing the message is used to prepare the message for further calculation. Parsing method is simply dividing the final padded message into sixteen 32 bit block (M_0, M_1 . . . M_{15}). These bit blocks will be substituted to the value of W_t, such that $W_0 = M_0$, $W_1 = M_1$ . . .$W_{15} = M_{15}$. A different computation is involved in the calculation of W_{16}. . . W_{79} which uses the following formula:

For t = 16 to 79,

$W_t = S^1(W_{t-16} \text{ xor } W_{t-14} \text{ xor } W_{t-8} \text{ xor } W_{t-13})$ (5)

The message digest is calculated using the following formula:

Let $A = H_o$, $B = H_1$, $C = H_2$, $D = H_3$, $E = H_4$.

For t = 0 to 79 do

$TEMP = S^5 (A) + F_t (B, C, D)+E+W_t+K_t$ (6)

$E = D; D = C; C = S^3 (B); B=A; A=TEMP$ (7)

Model of the SHA-1 Operation

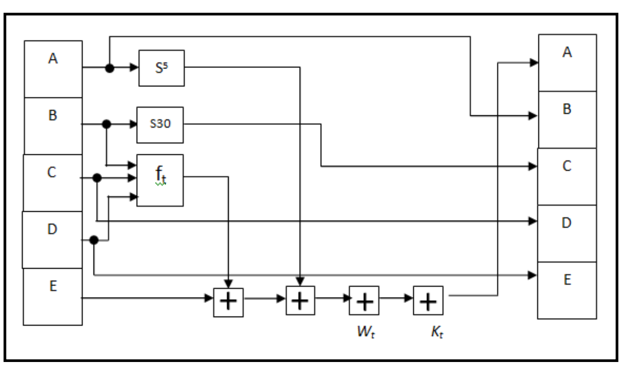

Figure 1. The SHA-1 Operation

The final message digest is the concatenation of the following:

$H_0 = H_0 + a$ (8)

$H_1 = H_1 + b$ (9)

$H_2 = H_2 + c$ (10)

$H_3 = H_3 + d$ (11)

$H_4 = H_4 + e$ (12)

Final Message Digest=H0||H1||H2||H3||H4 (13)

4 PROPOSED ENHANCED SHA-1

4.1. Pseudocode of Enhanced Parsing Method

```
set value of size to 16
for i counter is less than size
{
Num is equal to a certain number
m[i] is equal to s[i](m[i] xor num)
w[i] is equal to m[i]
print value of w[i]
}
```

4.3 Model of the Enhanced SHA-1 Operation

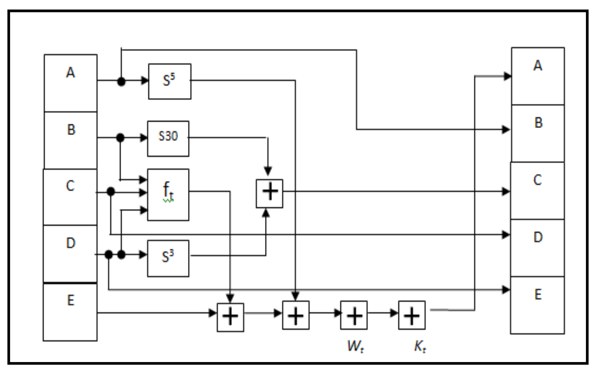

Figure 2.0 Proposed Enhanced SHA-1 Operation

5 DEVELOPMENT OF ENHANCED SHA1

The developed enhanced SHA-1 is explained in detailed below.

1. The original message is padded with one bit "1" first at the end of the original message.
2. The first one bit "1" padded at the end is followed by zero or more bits"0" to form a multiple of 512 bits.

3. The determined length of the original message will be appended to the padded message to form 512 bits.

4. The final padded message consisting of 512 bits will be generated to form 16 word blocks (M_0 to M_{15}). The enhanced SHA-1 on parsing method will include additional mathematical calculations as discussed in no. 4 proposed enhanced SHA-1. The goal of this enhancement is to strengthen the pre-processing function of SHA-1 specifically in parsing method.
   Suppose the original message is: 1a7fd53b4c. After padding one "1" and many "0's", appending and generating 16 word block, the value of M_0 to M_{15} are as follows:

$W_0=M_0$ = 1a7fd53b
$W_1=M_1$ = 4c800000
$W_2=M_2$ = 00000000
$W_3=M_3$ = 00000000
$W_4=M_4$ = 00000000
$W_5=M_5$ = 00000000
$W_6=M_6$ = 00000000
$W_7=M_7$ = 00000000
$W_8=M_8$ = 00000000
$W_9=M_9$ = 00000000
$W_{10}=M_{10}$=00000000
$W_{11}=M_{11}$=00000000
$W_{12}=M_{12}$=00000000
$W_{13}=M_{13}$=00000000
$W_{14}=M_{14}$=00000000
$W_{15}=M_{15}$=00000028

Application of enhanced parsing method

The 16 word block enumerated above will be generated again using the additional mathematical calculation on parsing method and will produce its new value:

$W_0=M_0$= 087ec54d
$W_1=M_1$=bd0220ec
$W_2=M_2$=480443d8
$W_3=M_3$=900883b0
$W_4=M_4$=20110761
$W_5=M_5$=40220ec2
$W_6=M_6$=80441d84
$W_7=M_7$=00883b09
$W_8=M_8$=01107612
$W_9=M_9$=0220ec24
$W_{10}=M_{10}$=0441d848
$W_{11}=M_{11}$=08833090
$W1_2=M1_2$=1107612
$W1_3=M1_3$=220ec240
$W_{14}=M_{14}$=441d8480
$W_{15}=M_{15}$=882f0900

Sample Calculation:

m[0] = s^0 (m[0] xor num)
= s^0 (1a7fd53b xor 12011076)
=s^0(00011010011111111101010100111011
xor (00100100000000010001000001110110)
=s^0 00001000011111101100010101001101)
= 00001000011111101100010101001101
m[0] = w[0] = 087EC54D

m[1] = s^1 (m[1] xor num)
= s^1 (4c800000 xor 12011076)
=s^1 (01001100100000000000000000000000
xor 00100100000000010001000001110110)
=s^1(01011110100000010001000001110110
= 10111101000000100010000011101100
m[1] = w[1] = bd02202ec

The value of W_{16} to W_{79} is calculated using the following formula:

$$W_t=S^1(W_{t-16} \text{ xor } W_{t-14} \text{ xor } W_{t-8} \text{ xor } W_{t-3})$$

5. The initialized 160 bit buffer, function and constant used are the same with the original SHA-1

6. To compute the message digest, the enhanced formula are used:

ISBN: 978-1-9491968-07-9 ©2015 SDIWC

Let A= H_0; B=H_1; C=H_2; D=H_3 and E=H_4 (14)

For t = 0 to 79 do

TEMP= $Ft(B,C,D)+S^5(A)+E+W_t+K_t$ (15)

E = D; D = C; C = $S^{30}(B)$ xor $S^3(D)$; B = A;
A =TEMP (16)

The Final Message Digest is calculated using a new formula:

The Final Message is the concatenation of the following:

H_0 = S1(H_0 + a) (Value on 79th Round) (17)

H_1 = S1(H_1 +b) (Value on 79th Round) (18)

H_2 = S1H_2 + c) (Value on 79th Round) (19)

H_3 = S1(H_3 +d) (Value on 79th Round) (20)

H_4 = S1(H_4 +e) (Value on 79th Round) (21)

Final Message= (H0 || H1 || H2 || H3 ||H4) (22)

6 PERFORMANCE RESULT

Table 1 shows the value of input message and message padding.

Table 1. 512 bit padded Message

Input Message:	**1a7fd53b4c**
512 bit Padded Message	1a7fd53b 4c800000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000028

Table 2 shows the new result of sixteen 32 bit words on blocks from M_0, M_1 ....M_{15} after using the enhanced parsing method.

Table 2. The new value of M_0, M_1 ....M_{15}

t	$M_t = W_t$
0	087ec54d
1	bd0220ec
2	480443d8
3	900883b0
4	20110761
5	40220ec2
6	80441d84
7	00883b09
8	01107612
9	0220ec24
10	0441d848
11	08833090
12	11076120
13	220ec240
14	441d8480
15	882f0900

Table 3 shows the value of W_{16} to W_{79} using the formula: $W_t = S^1(W_{t-16}$ xor W_{t-14} xor W_{t-8} xor $W_{t-13})$

Table 3. Calculated value of W16…W79

t	W_t	*t*	W_t
16	c6c8618e	34	24726d3a
17	d66f97f0	35	01a27d5c
18	c8f72fe3	36	b0c3b0cf
19	3cc2b8d8	37	aa0e9ca6
20	ce7bd86a	38	bf04d3c1
21	54a7b0d1	39	b55301db
22	f316af9d	40	6866100c
23	89f80c8e	41	f8e26a29
24	2e7cfe0b	42	60336035
25	5fb4c9b2	43	6b1134c1
26	a893340a	44	8580ee7b
27	70666806	45	b4632d77
28	89abe8f1	46	9f1bf917
29	ac2a9f36	47	c0a4bbbf
30	034a452a	48	c14a1dc1
31	bc26f51e	49	15f95206
32	18d25ea1	50	684c0cfe
33	6ca74761		

t	W_t	*t*	W_t
51	03ef91f4	66	f1aa2f6f
52	3f7dbee7	67	df99329f
53	86e579e9	68	e21feac1
54	972d565c	69	a6ce540e
55	64d0dd55	70	903f50ea
56	9ff42822	71	b277312b
57	224eb564	72	0ee3e4ea
58	d25ebfcb	73	c55b0296
59	86d340c0	74	3da730a3
60	0f5039de	75	2ea26ec2
61	40f8a1d4	76	6e966a3b
62	9f5fe494	77	4c57be39
63	7dba1a65	78	9bafd65d
64	ec153192	79	eea8cae1
65	560f2405		

Table 4 shows the value of register output A,B,C,D and E in hexadecimal values after passing t ($0 \le t \le 79$).

Table 4. Register output of A,B,C,D and E where t = 0…79.

t	REGISTER OUTPUT				
	A	**B**	**C**	**D**	**E**
0	a8335e00	67452301	be258d16	98badcfe	10325476
1	ece28d0e	a8335e00	a8fd2075	be258d16	98badcfe
2	95c8bb62	ece28d0e	6de5d42d	a8fd2075	be258d16
:	:	:	:	:	:
:	:	:	:	:	:
:	:	:	:	:	:
:	:	:	:	:	:
:	:	:	:	:	:
77	3a58e9cd	23e91a0e	29aff076	393ff37c	6ff56f8f
78	549e5a6d	3a58e9cd	c585c532	29aff076	393ff37c
79	5c89aa66	549e5a6d	62b813e5	c585c532	29aff076

Table 5. Generation of Hash Value

Hash	**Input Message**	**Hash Value in Hexadecimal**
Original SHA-1	1a7fd53b4c	488783979801d679394bd83428c28e412b8dee05
Enhanced SHA-1	1a7fd53b4c	879d9acf88d80bedf6e5e1c7ab703351db05a4cd

On table 5, the original and the enhanced hashing algorithm (SHA-1) were tested with the same input message and produced a different hash value but with the same number of 160 bit.

Table 6. Simulation of result for sample data for enhanced SHA-1

Input Message	**Message Digest**
1a7fd53b4c	879d9acf88d80bedf6e5e1c7ab703351db05a4cd
3a7fd53b4c	1dd5788f18d36ab7309d63e38851eee1f4be66cf

The table 6 shows sample simulation result. The first input message was “1a7fd53b4c” then followed by the second message of “3a7fd53b4c”. Notice that only the first digit of the second message was altered from “1” to “3” and based from the result, the avalanche effect is very evident since the enhanced algorithm produces the hash value with great difference after changing one digit from its original value.

Table 7. Hash Computation for Original and Enhanced SHA-1

Hash Algorithm	Hashing Time in Milliseconds	Input Message	Hash Value/ Message Digest
Original SHA-1	60.6	1a7fd53b4c	488783979801d679394bd83428c28e412b8dee05
Enhanced SHA-1	65.4	1a7fd53b4c	879d9acf88d80bedf6e5e1c7ab703351db05a4cd

The table 7 shows the running time on the generation of hash value. The average running time of five attempts for the original SHA-1 was 60.6 milliseconds and the average running time for the enhanced SHA-1 was 65.4 milliseconds. Based from the result, it shows that there is a minimal delay on the processing speed of the enhanced SHA-1 with 4.8 milliseconds difference from the original algorithm.

7 CONCLUSIONS

The enhancement that is incorporated in this study includes the following: 1) Enhancement on the pre-processing, specifically on the parsing method. Additional mathematical technique is attached to the original method parsing method. 2) The message digest and the final message digest formula was enhanced by giving additional shifting, xoring and improved calculations.

The enhanced SHA-1 maintains its original rounds which consist of 80 rounds and message digest output of 160 bit. To test the enhanced SHA-1, the author entered the same value as input message to the original and enhanced algorithm and based from the figures on table 5 both algorithm have 160 bit message digest output but the enhanced SHA-1 produces a different value. The table 7 shows a minimal delay on the processing speed, this is expected due to some processes incorporated on the enhanced algorithm. On table 6, the enhanced algorithm produces a very evident avalanche effect which is considered a good cryptographic design. The enhanced algorithm is better and more secured version of SHA-1 which is expected to resist possible future collision attacks.

REFERENCES

[1] Alfred M., Oorschot P., and Vanstone S., Handbook of Applied Cryptography, CRC Press, 1997

[2] Lakshmanan T., and Muthusamy M., A Novel Secure Hash Algorithm for Public Key Digital Signature Schemes. The International Arab Journal of Information Technology Vol.9, No.3 pp. 262 – 267, May 2012

[3] Rhee M., Internet Security, Cryptographic principles, algorithms and principles, pp.149, 2003 John Wiley & Sons, Ltd ISBN 0-470-85285-2

[4] Yang H., Digital Signature Scheme for E-mail Messages. PKI Tutorials. http://www.herongyang.com/PKI/SMIME-Digital-Signature-Scheme-for-Email-Messages.html

[5] Schneier B., Schneier Security. Article: Cryptanalysis of SHA-1, https://www.schneier.com/blog/archives/2005/02/cryptanalysis_o.html

[6] Alam M., and Ray S., Design of an Intelligent SHA-1 Based Cryptographic System: A CPSO Based Approach. International Journal of Network Security, Vol. 15, No.6,PP.465-470, Nov. 2013

[7] Wang X., and Yu H., “How to break MD5 and other hash functions”, Advances in Cryptology – EUROCRYPT, LNCS 3494, Springer-Verlag , pp.19-35, 2005.

[8] Goldwasser S., Micali S., and Rivest R., “ A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks”, Journal on Computing vol. 17, no.2. pp281-308, 1988.

[9] “New Europian Schemes for signatures, Integrity and Encryption Project”, available at: http://www.cryptonessie.org.

[10] Rjasko M., Properties of Cryptographic Hash Function, https://www.fmph.uniba.sk/fileadmin/user_upload/editors/studium/svk/2008/INF/rjasko.pdf

[11] The SHA-1 attacks, http://en.wikipedia.org/wiki/SHA-1

Christine Charmaine G. San Jose is currently taking up her Doctor in Information Technology and now on dissertation writing at Technological Institute of the Philippines, Quezon City, Philippines and finished her MSIT degree at

University of La Salette, Santiago City, Philippines in the year 2005. She obtained her Bachelor of Science in Computer Science degree at Emilio Aguinaldo College, Manila, Philippines in the year 1998. She started her teaching profession in the year 2001 at Institute of Information and Communication Technology at Isabela State University, Echague, Isabela, Philippines. She has been designated as an Institute Secretary for four years and has been extensively involved in Research and Extension of the Institute. She has been tapped by various Agencies to perform Job relevant to her field of specialization. She has been a Board of Election Inspector (BEI) of Department of Science and Technology (DOST) and was designated by the Commission on Election (COMELEC) to be a Member of the Board of Canvasser as Consolidated Canvassing System Operator. She is a member and an officer of JCI – Junior Chamber International Philippines. Her field of interest includes information system and data security.

Dr. Bobby D. Gerardo is currently the Vice President for Administration and Finance of West Visayas State University, Iloilo City, Philippines. His dissertation: Discovering Driving Patterns using Rule-based intelligent Data Mining Agent (RiDAMA) in Distributed Insurance Telematic Systems. He has published more than 60 research papers in national and international journals and conferences. He is a referee to international conferences and journal publications such as in IEEE Transactions on Pattern Analysis and Machine Intelligence and IEEE Transactions on Knowledge and Data Engineering. He is interested in the following research fields: distributed systems, telematics systems, CORBA, data mining, web services, ubiquitos computing and mobile communications.

Dr. Bartolome T. Tanguilig III took his Bachelor of Science in Computer Engineering in Pamantasan ng Lungsod ng Maynila, Philippines in 1991. He finished his Master's Degree in Computer Science from De La Salle University, Manila, Philippines in 1999, and his Doctor of Philosophy in Technology Management from Technological University of the Philippines, Manila in 2003. He is currently the Assistant Vice President for Academic Affairs and concurrent Dean of the College of Information Technology Education and Graduate Programs of the Technological Institute of the Philippines, Quezon City.

Dr. Tanguilig is a member of the Commission on Higher Education (CHED) Technical Panel for IT Education, the chair of the CHED Technical Committee for IT, the founder of Junior Philippine ITE Researchers (JUPITER), board member of the Philippine Society of IT Educators (PSITE), member of the PCS Information and Computing Accreditation Board (PICAB), and a member of the Computing Society of the Philippines (CSP).