# A Remote-Attestation-Based Extended Hash Algorithm for Privacy Protection

Yongxiong Zhang
Department of Economics and Trade
Guangzhou College of Technology and Business,
GCTB
Guangzhou, China
E-mail: csyxzhang@qq.com

Yucong You
Department of Economics and Trade
Guangzhou College of Technology and Business,
GCTB
Guangzhou, China
E-mail: 61070262@qq.com

Liangming Wang *
School of Software Engineering
South China University of Technology, SCUT
Guangzhou, China
E-mail: lmwang@scut.edu.cn
*The corresponding author

Luxia Yi
Department of Economics and Trade
Guangzhou College of Technology and Business,
GCTB
Guangzhou, China
E-mail: cilu-5@qq.com

***Abstract*—Compared to other remote attestation methods, the binary-based approach is the most direct and complete one, but privacy protection has become an important problem. In this paper, we presented an Extended Hash Algorithm (EHA) for privacy protection based on remote attestation method. Based on the traditional Merkle Hash Tree, EHA altered the algorithm of node connection. The new algorithm could ensure the same result in any measure order. The security key is added when the node connection calculation is performed, which ensures the security of the value calculated by the Merkle node. By the final analysis, we can see that the remote attestation using EHA has better privacy protection and execution performance compared to other methods.**

***Keywords-Trusted computing; Remote attestation; Privacy protection; Merkle hash tree; Extended hash algorithm***

## I. INTRODUCTION

With the extensive and widely use of Cloud Computing technology, more and more application services require completion of the interaction between multiple machines to achieve the tasks, how to ensure the machine's mutual-trustworthiness and security during the process has become a very important problem. The traditional security mechanisms are basically built upon  the application level, which is difficult to ensure the machine's mutual-trustworthiness and security. Trusted Computing Group (Trusted Computing Group, referred to as TCG) [1] has proposed the trusted computing (referred to as TC) technology development, leading to more and more system security solutions based on trusted computing technology, of which the trusted root and trust chain delivery mechanism provides a basic environment for the solution. The remote attestation technology in trusted computing provides a solution to the trust of both parties. The standard remote attestation mechanism given by the Trusted Computing Specification can be divided into three steps: (1) integrity status measurement; (2) measurement results query communication; (3) integrity status attestation. Those three steps constitute, as a whole, a remote certification mechanism framework. The integrity state measure is mainly to collect information on the hardware and software stack integrity status of the platform so as to be verified, mainly through a large number of Hash operations, and the measurement process to generate specific results stored in the TPM PCR. The integrity measure functions as the basis of the entire telematics technology, being a reliable attestation of whether it is safe and effective.

In 2004, based on the research of Trusted Computing, the IBM Research Center presented the design of the Integrity Measurement Architecture (IMA) [2] based on the Trusted Platform Module TPM, which was designed on the Linux operating system to accomplish the task. When the system opens the file setting into the memory, the IMA code set in the system will evaluate the integrity of the file, then saves the measurement results in the metric list, meanwhile it extends the metric to the TPM chip. The integrity of the IMA definition is based on the simple metric load code and some system static data, moreover, the IMA inserts a large number of metric points when performing the integrity measure, and thus increases the inaccuracy of the metric and the redundancy of the metric. Both of the IMA and other customary binary method, conduct the operation of each file via the simple Hash connection during the process of measurement , the attestation process requires the entire the process log, introducing the privacy exposure , and because the attestation of the log requires the re-completion of the entire process, the performance tends to be inferior. As for the attribute-based remote attestation in [3,4], the proving party only needs to give the corresponding attribute declaration, which is according to the target attribute of the verifier, and does not need to expose the entity uniqueness mark to the verifier. At the same time, because the unique identity of the running entity in the system can be regarded as one of the attributes of the system, the attributes-based remote attestation improves the flexibility of the certification program on the protection of system privacy. However, there exists a problem to perfect the completeness of attributes-based remote attestation method. The other methods in [5,6,7] also cannot solve the completeness problem.

Based on a more direct and complete binary remote attestation method, this paper proposes an Extended Hashing Algorithm with a more efficient and privacy protection. Extended Hashing Algorithm will be used in the process of integrity measurement in the process of remote attestation.

The rest of this paper is organized as follows. Section II introduces relevant background knowledge. The third section introduces the Extended Hashing Algorithm based on Merkle Hash Tree. In the fourth section, we analyze the Extended Hashing Algorithm proposed in this paper. Section 5 summarizes this paper and prospects for the next step.

## II. Background

### A. Binary-based Remote Attestation

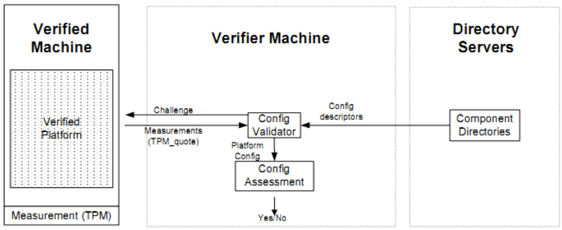

Figure 1. Framework of TCG’s binary-based remote attestation

As it is shown in Fig .1, in the Framework of binary remote authentication, the Verifier Machine and the Verified Machine communicate directly through the Config Validator module. The Config Validator module obtains the results of the binary measurement by the proving party through the TPM and reconstructs the configuration information of the platform. During the refactoring process, both of the report log information generated by the TPM metric process and the component configuration information provided by Direcory Servers are required to be used.

### B. Merkle Hash Tree

Hash Tree, also known as the Merkle Hash Tree [8], was coined by Merkle as a method to establish the shared secret between the two entities by using a public key infrastructure in 1980. Merkle Hash Trees are now commonly used to protect the data blocks in memory [9,10,11]. It is shown by Figure 2 that a binary Merkle Hash Tree with four-leaf nodes. Apparently ,on the Merkle Hash Tree, the data object is created as a leaf node, and the tree's internal node is its sub node is the Hash Value concatenation. The root of the tree is named the “root hash”, which represents all data objects, because changes to arbitrary data objects tends to cause changes in the root Hash Value.

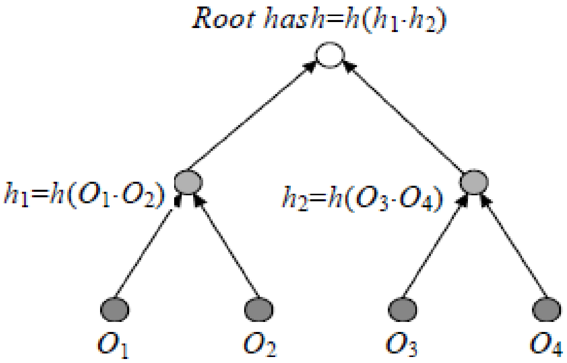

Figure 2. Merkle Hash Tree

To check the integrity of the leaf nodes, the following points are needed: (1) reading the contents of the node and its brother's nodes; (2) connecting their data; (3) calculating the Hash Value of the data after the connection; 4) Repeating the above steps to the Root Hash; (5) Checking whether the calculated result is consistent with the content stored in the Root Hash node.

To update the leaf node value, firstly check the integrity of the leaf node (process as mentioned above). If the leaf node is complete, then the followings are needed: (1) modifying the node value; (2) calculating and updating the value of its parent node; (3) repeating the above steps until the Root Hash is eventually updated.

## III. A Merkle Hash Tree -Based Extended Hash Algorithm

As in [12], the calculation method of Hash Value of the inner node connection on the Merkle Hash Tree generates the h (o1.o2) <> h (o2.o1), so the different order of the leaf node data read during the process generated on the Merkle Hash Tree leads to different results.

The extended Merkle Hash Tree uses an alternative method when connecting joints between nodes in the generation process so as to ensure that the order of the nodes' data is the same.

### A. Implementation of EHA Algorithm

EHA (A, B, K), A and B represent two node extended, k represent the security key.

**Definition:**

$N_L=\{0,1\}^{160}$, the leaf node on the tree, representing the 160-bit Hash Value of the file

$N_B=\{0,1\}^{168}$, the inner node in the tree, representing the 168-bit Hash result obtained by connecting the Hash by two other nodes

K, key

$K^*=\{0,1\}^{160}$, which represents the 160-bit Hash Value produced by the Hash of the key K

**Algorithmic process:**

Calculating the Hash of K to get the 160-bit Hash result K *

Calculating the number of bits in 1 in K *, denoted as CK

*1) Connection between leaf nodes*

NL1, NL2 represent , respectively , two leaf nodes,

*a) The NL1 and K * bits are XOR (XOR), and the result is recorded as S1*

*b) The NL2 and K * bitwise exclusive OR, the result is recorded as S2*

*c) Calculate the number of bits in the NL1 and NL2 values of 1, denoted as C1, C2, respectively*

*d) S1 cycle left shift C1 bit, the result is recorded as CS1*

*e) S2 cycle left shift C2 bit, the result is recorded as CS2*

*f) The CS1 and CS2 are exclusive-OR, and the result is denoted by R*

*g) Calculate the number of bits in the value of R for 1, denoted as CR, CR is in the range of 0 to 160, so it can be represented by 8-bit binary*

*h) Connect the 160-bit R to the 8-bit CR to get 168 bits, denoted as Y*

*i) Move the Y cycle to the left to get the final result*

*2) Connection between the nodes*

NB1, NB2 represent, respectively , two inner nodes

*a) NB1 cycle right shift CK bit, the result is recorded as NB1 **

*b) Take NB1 * before the first 160, recorded as NC1*

*c) NB2 cycle right shift CK bit, the result is recorded as NB2 **

*d) Take NB2 * before the 160, recorded as NC2*

*e) The NC1 and NC2 are exclusive-OR, and the result is denoted by R*

*f) Calculate the number of bits of value 1 in R, denoted as CR*

*g) The 160-bit R and 8-bit CR connected to the results of 168, recorded as Y*

*h) Move the Y cycle to the left by the CK bit to get the final result*

*3) The connection between the leaf node and the inner node*

NL, NB decibels represent leaf nodes and inner nodes

*a) NB cycle right shifts CK bit, the results recorded as NB **

*b) Take NB * the 160 downwards, recorded as NC*

*c) Calculate the number of bits in the NL with a value of 1, denoted as C*

*d) NL and K * bitwise exclusive OR, the result is recorded as S*

*e) S cycle left shift C bit, the result is recorded as CS*

*f) The CS and NC bitwise exclusive OR, the result is recorded as R*

*g) Calculate the number of bits of value 1 in R, denoted as CR*

*h) Connect the 160-bit R to the 8-bit CR to get 168 bits, denoted as Y*

*i) Move the Y cycle to the left to get the final result*

## B. *Production Process Analysis of Extended Merkle Hash Tree*

In light of the definition of Extended Merkle Hash Tree, we can see that any extension of the Merkle Hash Tree , during the production process , is a number of times between the leaves of the connection between the Hash operation, a number of connections between the nodes of the Hash operation and a number of inner nodes and a leaf node, and the inner node is generated by the connection operation of two leaf nodes or one leaf node to another inner node.

By the description of the algorithm of the previous connection operation, it can be seen that the operation of step a , b , c , d of the connection operation between two inner nodes are the reverse operation of step 1 , 2of the operation of one leaf node and one inner node during the last two steps of the operation of, so in order to facilitate the analysis, we put the above three operations simplified process as follows , not affecting the results of the case:

- $\oplus$, XOR operation symbol
- <<<, loop left shift operation symbol

N1, N2 between the two leaves of the operation can be expressed as: $((N1 \oplus K\,*) <<< C1) \oplus ((N2 \oplus K\,*) <<< C2)$;

N1 and N2 two nodes within the operation of $N1 \oplus N2$;

The connection operation between the leaf node N1 and the inner node N2 $((N1 \oplus K\,*) <<< C1) \oplus N2$.

Assuming that the process of Extended Merkle Hash Tree is generated by the leaf node, we use the operator * to describe that the Extended Merkle Hash Tree that generates n leaf nodes is N1 * N2 ••• * Nn.

N1 * N2 is actually the connection between the two leaves, through the previous algorithm Definition: $N1 * N2 = ((N1 \oplus K\,*) <<< C1) \oplus ((N2 \oplus K\,*) <<< C2)$ , Because the XOR operation is consistent with the exchange law, so $N1 * N2 = ((N1 \oplus K\,*) <<< C1) \oplus ((N2 \oplus K\,*) <<< C2) = ((N2 \oplus K\,*) <<< C2) \oplus ((N1 \oplus K\,*) <<< C1) = N2 * N1$.

N1 * N2 * N3 by N1 and N2 to do a connection between the leaves of the operation, and then the results and N3 do a leaf and the connection between the nodes in the operation.

By the previous definition, $N1 * N2 * N3 = ((N1 \oplus K\,*) <<< C1) \oplus ((N2 \oplus K\,*) <<< C2) * N3 = ((N1 \oplus K\,*) <<< (N2 \oplus K\,*) <<< C2) * ((N3 \oplus K\,*) <<< C3) = N1 * (N2 * N3)$.

$N1 * N2 = X1 \oplus X2$, $N1 * N2 * N3 = X1 \oplus X2 \oplus X3$, further result can be obtained, $N1 * N2 * ... * Nn = X1 \oplus X2 \oplus ... \oplus Xn$.

Thus, in the process of generating the Extended Merkle Hash tree, the result of the root node of the EMT is ultimately generated by the n leaf nodes, being consistent regardless of the order of the $N_i$.

## IV. Analysis of the Advantages of Extended Hash Algorithm

### A. *Privacy Protection Analysis*

The traditional TCG remote attestation mechanism extends the metric Hash Value of the metric into the PCR and reconstructs the PCR value when performing integrity verification. The advantage of this mechanism is that it is easy to construct an integrity trust chain. The drawback of it lies at it being necessary to understand the total integrity Hash of the metric and its extension to the PCR, so it is appropriate to measure the entire process from powering up the machine to the start of the operating system, but for the application, it does not trust mutually and generally do not have a strict execution order relationship, and the IMA metric-based metric verification mechanism still requires the application of the integrity of the Hash Value during the whole process aiming to extend the specified PCR to implement the measurement verification, which is the root cause of insufficient privacy protection.

Extended Hash Algorithm is used to remotely attest that the integrity of the program to be verified so as to calculate the integrity of the program, followed by a Merkle Hash Tree developed, and tree nodes preserve the integrity of the program Hash Value, and the tree in the non-leaf nodes are automatically generated by the Extended Hash Algorithm. When the remote attestation is conducted, there is only a node is obtained through the encrypted Hash Value.

Hence, it is clear that the use of Extended Hash Algorithm for remote attestation of privacy can be effectively protected.

### B. *Analysis of Performance*

For the IMA architecture, a list of n nodes is required to perform this Extended Hash operation with a time performance of O (n). For the Extended Hash Algorithm based on the Merkle Hash Tree, the time performance is O (log2n). And this has proved that the implementation is of high efficiency.

## V. Conclusion

This paper conducts an analysis on the existing problem of binary-based and attribute-based remote attestation method, because the attribute-based remote attestation method faces a complete problem, which is difficult to solve. To the binary-based remote attestation method, this paper focused the solution to the privacy protection problem. An Extended Hash Algorithm for privacy protection has been proposed. In the process of integrity measurement, the Extended Hash Algorithm uses the Merkle Hash Tree to store the binary Hash Value. During the process of generating the Merkle Hash Tree, compared to other algorithms, the Extended Hash Algorithm proposed in this paper is not affected by the extended order. Through the final security analysis, the algorithm achieves the desired effect.

In the next process, we will, apply the Extended Hash Algorithm proposed in this paper, to the remote attestation process, aiming to achieve the definition of the completion of the entire remote attestation.

## Acknowledgment

This paper is supported by: key project of the Combination of “Production, Education and Research” supported by Guangdong province and Ministry of Education(project number: 2012B091000109); the second batch of core technology research project on strategic emerging industry of Guangdong Province(project number: 2012A010701005); the Fundamental Research Funds for the Central Universities(project number:2012ZM0051).

## References

[1] “Trusted computing.” [Online]. Available: http://www. trustedcomputinggroup.org/

[2] R. Sailer, X. Zhang, T. Jaeger, and L. Van Doorn, “Design and implementation of a tcg-based integrity measurement architecture.” in USENIX Security Symposium, vol. 13, 2004, pp. 223–238.

[3] L. Chen, R. Landfermann, H. Lohr, M. Rohe, A.-R. Sadeghi, and ¨C. Stuble, “A protocol for property-based attestation,” in ¨ Proceedings of the first ACM workshop on Scalable trusted computing. ACM, 2006, pp. 7–16.

[4] Sadeghi A, Stüble C. Property-Based attestation for computing platforms: caring about properties, not mechanisms. In: Raskin V, ed. Proc. of the 2004 Workshop on New Security Paradigms. New York: ACM, 2004. 67−77.

[5] T. Rauter, A. Holler, N. Kajtazovic, and C. Kreiner, “Privilege-based ¨ remote attestation: Towards integrity assurance for lightweight clients,” in Proceedings of the 1st ACM Workshop on IoT Privacy, Trust, and Security. ACM, 2015, pp. 3–9.

[6] Luo, W., Liu, W., Luo, Y., Ruan, A., Shen, Q., & Wu, Z. (2016). Partial Attestation : Towards Cost-Effective and Privacy-Preserving Remote Attestations.2016 IEEE Trustcom/BigDataSE/ISPA.IEEE,2016,pp.152 – 159

[7] Abir Awad; Sara Kadry; Brian Lee; Gururaj Maddodi; Eoin O'Meara.Integrity Assurance in the Cloud by Combined PBA and Provenance.2016 10th International Conference on Next Generation Mobile Applications, Security and Technologies (NGMAST).2016,pp,127-132

[8] Merkle RC. Protocols for public key cryptosystems. In: Proc. of the IEEE Symp. on Security and Privacy. Washington: IEEE Computer Society, 1980. 122−134.

[9] Merkle RC. A certified digital signature. In: Brassard G, ed. Proc. of the 9th Annual Int’l Cryptology Conf. on Advances in Cryptology. Heidelberg: Springer-Verlag, 1989. 218−238.

[10] Blum M, Evans W, Gemmell P, Kannan S, Naor M. Checking the correctness of memories. In: Proc. of the 32nd Annual Symp. on Foundations of Computer Science. Washington: IEEE Computer Society, 1991. 90−99.

[11] Gassend B, Suh GE, Clarke D, van Dijk M, Devadas S. Caches and hash trees for efficient memory integrity verification. In: Proc. of the 9th Int’l Symp. on High-Performance Computer Architecture. Washington: IEEE Computer Society, 2003. 295−306.

[12] Xu, Z.-Y., He, Y.-P., & Deng, L.-L. (2011). Efficient Remote Attestation Mechanism with Privacy Protection. *Journal of Software, 22*(2), 339–352.