ISSN:0975-9646

Enforced Conviction in Cryptographic Provenance Across Critical System Information

Prof. Asha. N, Prince

School of Information Technology and Engineering (SITE), VIT University,
Vellore - 632014, Tamil Nadu, India

***Abstract*: This project's aim is to describe two application, first one is keystroke integrity verification and secondly, malicious traffic detection or traffic monitoring. In this project we design and implement a cryptographic protocol that impose keystroke integrity by utilizing trusted computing platform and this approach helps to restrict outbound malware traffic. Trusted platform module is an international standard for secure crypto-processor. It offers facilities for the secure generation of cryptographic key but in this paper for generating the public key and private key we are using a novel approach. We apply basic cryptographic mechanism to ensure the correct data flow. We describe how to integrate cryptographic components with operating system and how to use hardware tools for the integrity of cryptographic keys in our verification operations.**

***Keywords*: Public Key, Private Key, Cryptographic Key**

I. INTRODUCTION

Today network security is very challenging task because if we are sending some data from one machine to another machine, it should be secure and correct at the receiver side. For making the data transmission more secure we are presenting a security model and a new approach, cryptographic verification using a trusted platform module that upgrade the truthfulness of a host and data. We present a mechanism that restrict opponent from utilizing host recourses along with a security property that prevents the caricature of the data being created at the source.

This project's aim is to describe two application, first one is keystroke integrity verification and secondly, malicious traffic detection or traffic monitoring. In this project we design and implement a cryptographic protocol that impose keystroke integrity by utilizing trusted computing platform and this approach helps to restrict outbound malware traffic. TPM is a computer chip that can securely store artifacts used to authenticate the platform (PC, Laptops). A trusted platform module can also be used to store platform measurement that helps ensure that the platform remains trustworthy. Trusted platform module is an international standard for secure crypto-processor. It offers facilities for the secure generation of cryptographic key but in this paper for generating the public key and private key we are using a novel approach. We apply basic cryptographic mechanism to ensure the correct data flow. We describe how to integrate cryptographic components with operating system and how to use hardware tools for the integrity of cryptographic keys in our verification operations.

II. PROPOSED METHODOLOGY

In this paper a new cryptographic provenance verification (CPV) approach is being proposed. The paper demonstrates the applications' features in terms of keystroke-integrity service and full-bodied host-based traffic monitoring. The traditional data hiding mechanisms have been inculcated to ensure the appropriate flow of data corresponding to the system properties of the host, especially on verifying the provenance of fuzzy system related data. We also describe how ciphered components or data can be integrated with the operating system ensuring the use of hardware tools for the integrity or uniqueness of cryptographic keys in our provenance verification operations. The Upcoming four modules will elaborate how the titled project works.

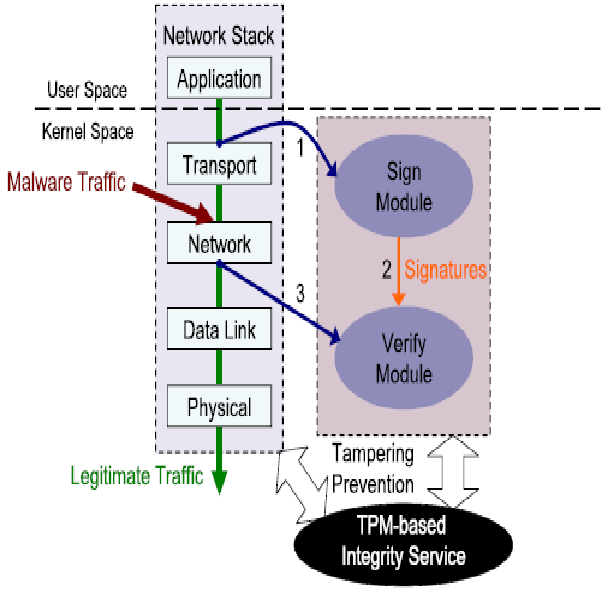
a. Authentication: Authentication is the process of determining whether someone or something is, in fact, who or what it is declared to be. In private and public computer networks authentication is commonly done through the use of logon passwords. Knowledge of the password is assumed to guarantee that the user is authentic. Each user registers initially (or is registered by someone else), using an assigned or self-declared password. The weakness in this system for transactions that are significant is that passwords can often be stolen, accidentally revealed, or forgotten.

b. Malware: It can be used to help keep a network secure. Its primary goal is to control the incoming and outgoing network traffic by analyzing the data packets and determining whether it should be allowed through or not, based on a predetermined rule set. A network's builds a bridge between an internal network that is assumed to be secure and trusted, and another network, usually an external.

c. *Cryptography:* Cryptographic attribution corroboration in the design of a keystroke honesty examine that utilizes the hardware Trusted Platform Module (TPM). Cryptographic attribution corroboration approach in realizing a host-based traffic-monitoring framework. The framework is capable of detecting surreptitious outbound traffic of OS-level malware by enforcing the attribution corroboration for outbound network packets. Malware traffic that bypasses normal network-stack functions can be effectively detected. In general if we are talking about the cryptography it is simply encryption and decryption of the data by using the public key and private key.

d. Provenance: We produce a set of top secret keys from the master key for our cryptographic operations in order to improve the efficiency. Our design involves creating three private/public RSA key pairs: a signing key, a binding key, and a storage key. The signing key is used to sign and encrypt outbound packets as well as TPM quotes.

III. Architecture



IV. Algorithm (Pell's equation)

This is novel approach for generating the public key and private key. This approach based on prime number. Steps are given below-

STEP 1

Select a undisclosed odd prime integer number "B".

STEP 2

Consider the Diophantine Equation:

$Z^2 - BW^2 = 1$ (equation 1)

Let (Z_0 ,W_0) be the least positive integral solution of equation (1).

Here Z_0 ,W_0 are kept undisclosed

STEP 3

Choose two large odd prime c, d

Define M= cd (equation 2)

STEP 4

$\beta = [Z_0 + \varphi(m)]^2 - B[W_0 + e]^2$ (equation 3)

Where "e" can be select such that 1<e< φ(m) and gcd (e,φ(m)) = 1

Since GCD (e, φ (m)) = 1, there is a single positive "p" such that pe $\equiv$ 1(modφ (m))

Assume $p^3 \not\equiv$ 1(modφ (m))

$e^3 \not\equiv$ 1(modφ (m)) (equation 4)

STEP 5

$\beta = Z_0^2 + [\varphi(m)]\text{^}2 + 2Z_0\, \varphi(m) - B\,[W_0^2 + e^2 + 2W_0\, e]$

$= Z_0^2 - BW_0^2 + [\varphi(m)]\text{^}2 + 2Z_0\, \varphi(m) - Be^2 - 2W_0\, eB$

$\beta = 1 - Be^2 - 2W_0\, eB$ (modφ (m))

$\beta + Be^2 - 2W_0\, eB =$ 1 (modφ (m))

Multiply by p^3 on both sides,

$\beta p^3 + Be^2 - 2W_0 p^2\, B \equiv p^3$ (modφ (m))

STEP 6

DEFINE $C = \beta\, p^3 + 2W_0 p^2 B + Bp$

$C = p^3$ (mod(φ (m)))

STEP 7

Represent the given message "n" in the interval (0 ,m-1)

STEP 8

Assume GCD (n,m)=1

STEP 9

Public key = C ,m

ENCRYPTION $E \equiv n^{C}$ (mod m)

$\blacksquare\, n^{p^3 + k.\varphi(m)}$ (mod m)

$\equiv n^{p^3} [n^{\varphi(m)}]\text{^}k$(mod m)

- E $= n^{p^3}$ (mod m)

STEP 10

DECRYPTION $= E^{e^3}$ (mod m)

$= (np^3)^{e^3}$ (mod m)

$\equiv n^{p^3 e^3}$ [Hence $p^3 e^3$ = 1(modφ (m))]

$\equiv$ n (mod m)

V. Conclusion

We gave a common approach for civilizing the guarantee of system data and properties of a host, which will be the helpful for preventing malware activities and attacks. Our host-based system security solutions against malware complement network-traffic-based analysis. We confirmed CPV's application in identifying surreptitious malware activities of a host, in particular how to distinguish malicious illegal data flow from legitimate one on a computer that may be compromised. On the bases of our research we can that these keys (public key,private key) that is generated with the help of the Pell's equation is more secure. These keys are not select by any user it will automatic generate with the help of equation. It is more secure because Pill's equation is combination of both the number that is prime and natural number.

REFERENCES

[1] A. Baliga, V. Ganapathy, and L. Iftode, "Automatic Inference and Enforcement of Kernel Data Structure Invariants," Proc. 24th Ann. Computer Security Applications Conf. (ACSAC '08), 2008.

[2] A. Baliga, P. Kamat, and L. Iftode, "Lurking in the Shadows: Identifying Systemic Threats to Kernel Data," Proc. IEEE Symp. Security and Privacy, pp. 246-251, 2007.

[3] B. Blackburn and R. Ranger, Barbara Blackburn, the World's Fastest Typist. 1999.

[4] M. Christodorescu, S. Jha, and C. Kruegel, "Mining Specifications of Malicious Behavior," Proc. Sixth Joint Meeting of the European Software Eng. Conf. and the ACM SIGSOFT Symp. the Foundations of Software Eng. (ESEC-FSE '07), pp. 5-14, 2007.

[5] W. Cui, R.H. Katz, and W. tian Tan, "Design and Implementation of an Extrusion-Based Break-in Detector for Personal Computers," Proc. 21st Ann. IEEE Computer Security Applications Conf. (ACSAC '05), pp. 361-370, 2005.

[6] D.E. Denning, "A Lattice Model of Secure Information Flow," Comm. ACM, vol. 19, pp. 236-243, May 1976.

[7] D.E. Denning and P.J. Denning, "Certification of Programs for Secure Information Flow," Comm. ACM, vol. 20, pp. 504-513, July 1977.

[8] M. Dhawan and V. Ganapathy, "Analyzing Information Flow in Javascript-Based Browser Extensions," Proc. Ann. IEEE Computer Security Applications Conf. (ACSAC '09), pp. 382-391, 2009.

[9] A. Dinaburg, P. Royal, M. Sharif, and W. Lee, "Ether: Malware Analysis via Hardware Virtualization Extensions," Proc. 15th ACM Conf. Computer and Comm. Security (CCS '08), pp. 51-62, 2008.

[10] S. Garriss, R. Ca´ceres, S. Berger, R. Sailer, L. van Doorn, and X. Zhang, "Trustworthy and Personalized Computing on Public Kiosks," Proc. Sixth Int'l Conf. Mobile Systems, Applications, and Services, pp. 199-210, 2008.

[11] J. Goebel and T. Holz, "Rishi: Identify Bot Contaminated Hosts by IRC Nickname Evaluation," Proc. First USENIX Workshop Hot Topics in Understanding Botnets, Apr. 2007.

[12] J.B. Grizzard, V. Sharma, C. Nunnery, B.B. Kang, and D. Dagon, "Peer-to-Peer Botnets: Overview and Case Study," Proc. First USENIX Workshop Hot Topics in Understanding Botnets, Apr. 2007.

[13] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "BotMiner: Clustering Analysis of Network Traffic for Protocol- and Structure-Independent Botnet Detection," Proc. 17th USENIX Security Symp., 2008.

[14] R. Gummadi, H. Balakrishnan, P. Maniatis, and S. Ratnasamy, "Not-a-Bot: Improving Service Availability in the Face of Botnet Attacks," Proc. Sixth USENIX Symp. Networked Systems Design and Implementation (NDSI '09), 2009.

[15] M.G. Jaatun, J. Jensen, H. Vegge, F.M. Halvorsen, and R.W. Nergaˆrd, "Fools Download where Angels Fear to Tread," IEEE Security & Privacy, vol. 7, no. 2, pp. 83-86, 2009.