Privacy-Preserving Cloud-Based Firewall for IaaS-Based Enterprise

Hualong Sheng*, Lingbo Wei*, Chi Zhang*, Xia Zhang†
*Key Laboratory of Electromagnetic Space Information, Chinese Academy of Sciences
University of Science and Technology of China, Hefei 230027, China
Email: shenghl@mail.ustc.edu.cn, lingbowei@ustc.edu.cn, chizhang@ustc.edu.cn
†School of Computer Science and Technology, Wuhan Univeristy of Technology, Wuhan 430070, China
Email: xia_whut@163.com

***Abstract*—Firewall is the critical first line of defense against cyber attacks. In order to improve security and performance, modern enterprises have to deploy and manage their own firewalls. But the firewall infrastructure is expensive and complex to manage, thus enterprises begin to outsource their firewalls to the cloud to reduce firewall management and deployment costs. The current firewall outsourcing schemes require enterprises to fully trust the cloud. However, firewall policies are confidential information of enterprises, and enterprises do not want to reveal them to the cloud. Such privacy issues hinder the adoption of cloud-based firewall service. With the development of cloud computing, more and more enterprises migrate their IT systems to the cloud (we call these enterprises the IaaS-based enterprises). In this paper, we propose a privacy-preserving firewall outsourcing scheme for a new architecture, in which an IaaS-based enterprise outsources its firewall policies to another cloud. We implement a prototype of our scheme and the experimental results demonstrate the feasibility and performance of our scheme.**

***Keywords*—privacy-preserving; firewall outsourcing; cloud-based firewall**

I. Introduction

Firewalls play an important role in the enterprise network. However, deploying and managing firewalls have brought significant cost for enterprises especially smaller ones. Nemertes estimated the cost of firewall deployment and maintenance as $116,075 for the first year and an annual cost of $108,200 for a midsize US company with 5Mbps of Internet connectivity [1], [2]. The major cost is spent on hiring, training administrators and deploying, upgrading firewalls [1].

Enterprises begin to outsource their firewalls to the cloud due to the promise of cloud computing to decrease costs, easy management, and provide elasticity and fault-tolerance [3]. The environment of a cloud-based firewall implementation is drastically different from that classical in-house version. Comparing with the DIY firewalls, cloud-based firewall can offer some additional features such as availability, scalability and extensibility [4]. However, the firewall policies are confidential and can be used to find the security holes by attackers, thus enterprises do not want to reveal their firewall policies to the cloud because they do not fully trust the cloud. In brief, the challenge here is to outsource firewall to the cloud without revealing the firewall policies to the cloud.

In this paper, we consider a scenario that an IaaS-based enterprise migrates its IT system to a cloud; On the other hand, the IaaS-based enterprise outsources its firewall to another cloud for security and privacy reasons. The two clouds are independent and they work together to offer firewall service to the IaaS-based enterprise. We propose a privacy-preserving firewall outsourcing scheme that works in the above scenario.

Contribution. To the best of our knowledge, we propose the first IaaS-based enterprise firewall outsourcing framework that preserves the privacy of firewall policies. We exploit the additional homomorphic encryption to obfuscate the firewall and keep the firewall policies confidential. To demonstrate the feasibility and performance of our scheme, we present a proof-of-concept implementation of our scheme. The experimental results show that our scheme is practical.

II. Related Work

In this section we review some works on privacy-preserving network functions outsourcing.

Khakpour and Liu [1] propose the first privacy-preserving firewall outsourcing approach. In order to anonymize firewall policies, they introduce Bloom Filter Firewall Decision Diagram(BFFDD) [5] as a data structure. However, Bloom filters introduce false positives [6] and are not very secure [7], [8].

In order to provide privacy-preserving network functionality outsourcing, Shi et al. [8] use the construction of conjunction obfuscator [9] to obfuscate the firewall rules. [8] exploits the cryptographic multilinear map constructed in the form of Graded Encoding System [10]. However, the construction of multilinear maps used in [8] has been proved to be insecure due to the attack on the isZero routine [11].

The work of Melis et al. [7] proposes two practical solutions based on partial homomorphic encryption and public-key encryption with keyword search (PEKS). Note that when outsourcing firewall to the cloud, the cloud middlebox has to send all the results of every rule in the firewall to the client middlebox due to the cloud middlebox does not know which rule the packet matches with. This increases communication overhead because the results to be transmitted increase linearly

This work was supported by the Natural Science Foundation of China under Grants 61202140 and 61328208, by the Program for New Century Excellent Talents in University under Grant NCET-13-0548, and by the Fundamental Research Funds for the Central Universities under Grand WK2101020006.

978-1-4673-9803-9/15 $31.00 © 2015 IEEE
DOI 10.1109/NaNA.2016.37

with the number of firewall rules. Moreover, the client middlebox has to decrypt the results one by one until the client middlebox finds that the packet matches with a rule and this increases the computation costs.

There have been some works done on the network middleboxes outsourcing in Software Defined Networks(SDN) [3], [12]. More specifically, Sherry et al. [3] present a prototype of APLOMB architecture, by which the network middleboxes of the enterprise can be outsourced to the cloud. This architecture is mainly designed to redirect the traffic from/to the enterprise to the cloud without greatly damaging throughput. Gibb et al. [12] designed an architecture that can be used to outsource network functionality to external feature providers and what the enterprises only needs to do is to forward data. However, these works only deal with the problem of network middleboxes outsourcing and do not consider the privacy of the outsourced network middleboxes.

III. System Model and Primitives

In this section we first present the system architecture and then introduce the cryptographic primitives.

A. System Architecture

In this paper we describe a new scenario of cloud-based firewall, which is different from the scenario in [7], [8], where an IaaS-based enterprise outsources its firewall to another cloud for security concerns. As illustrated in Figure 1, the two clouds are named CloudA and CloudB for ease of description. Most of the enterprise's IT infrastructure (e.g., database server, application server) is located in the CloudB and the enterprise outsources its firewall to CloudA. CloudA and CloudB cooperate with each other to fulfill the firewall function. A packet from external site to enterprise needs to be through CloudA first and then CloudB, and finally to reach the enterprise.

TABLE I: Firewall Rule Example

rule #	Source		Destination		Protocol	Action
	IP	Port	IP	Port		
1	192.168.45.*	*	*	*	*	deny
2	192.168.16.*	*	202.38.64.1	23	TCP	deny
3	202.38.64.*	8090	*	*	UDP	permit
4	10.*.*.*	*	115.25.209.39	8080	TCP	deny

A typical IP firewall is a packet filter which looks at the network addresses, ports and protocol of the packet and determines if that packet should be allowed or blocked. A firewall rule is composed of 5-tuple (Source IP, Source Port, Destination IP, Destination Port, Protocol) and corresponding Action, as shown in Table I. In our architecture, we split the firewall function into two parts, Matching and Action. CloudA does not know the action of a firewall rule and all it knows is the ciphertext of the action. CloudA is only responsible for the matching operation that is to find out which rule the packet matches with. When CloudA finds a matching between a packet and a firewall rule, it sends the packet and the corresponding action's ciphertext to CloudB. CloudB is responsible for the action operation. When CloudB receives a packet and corresponding ciphertext, it decrypts the ciphertext and allows or blocks the packet according to the decryption result.

In this paper, for each rule in the firewall we denote it as $r = (\overrightarrow{v}, W, A)$, where $\overrightarrow{v} \in \{0,1\}^k$ is the bitwise representation of the rule and the i-th bit is 0 or 1 as expected if $i \notin W$, the set $W \subseteq [k]$ represents the "wildcard" bits, and $\overrightarrow{v}[i] = 0$ if $i \in W$ for ease of presentation, and A is the corresponding action. For example, the rule #1 in Table I, "192.168.45.*", can be modeled as $\overrightarrow{v}$ = "11000000 10101000 00101101...", $W = \{25, 26, ..., 32\}$, and $A = \{deny\}$.

B. Trust Assumptions

We assume that CloudA and CloudB are semi-trusted. They will honestly perform the protocol. On the other hand, the two clouds are curious and they will try to infer the firewall rules. Furthermore, we assume that CloudA and CloudB will not collude.

C. Primitives

Homomorphic encryption is a form of encryption that allows computations to be carried out on ciphertext, thus generating an encrypted result which, when decrypted, matches the result of operations performed on the plaintext. In this paper, we exploit Paillier cryptosystem [13] to obfuscate the firewall rules. Paillier cryptosystem works as follows:

Key generation: Choose two large prime numbers p and q randomly and independently of each other such that $gcd(pq, (p-1)(q-1)) = 1$. Compute $n = pq$ and $\lambda = lcm((p-1)(q-1))$. Select random integer g where $g \in Z^*_{n^2}$. Ensure n divides the order of g by checking the existence of the following modular multiplicative inverse: $\mu = (L(g^\lambda mod n^2))^{-1} \ mod \ n$, where function L is defined as $L(\mu) = \frac{\mu-1}{n}$. Finally, the public (encryption) key is (n, g) and the private (decryption) key is (λ, μ).

Encryption: Let m be a message to be encrypted where $m \in Z_n$. Select random r where $r \in Z^*_n$. Compute ciphertext as: $c = g^m r^n \ mod \ n^2$.

Decryption: Let c be the ciphertext to decrypt, where $c \in Z^*_{n^2}$. Compute the plaintext message as: $m = L(c^\lambda \ mod \ n^2) \cdot \mu \ mod \ n$.

The Paillier cryptosystem has the following properties:

Homomorphic properties:

$D(E(m_1, r_1) \cdot E(m_2, r_2) \ mod \ n^2) = m_1 + m_2 \ mod \ n$

Self-Blinding:

$D(E(m, r_1) \cdot r^n \ mod \ n^2) = m$, where $r \in Z^*_n$

IV. Our Scheme

A. Fundamental Framework

A program can be converted into a new program by a cryptographic obfuscator such that the new program can perform the same function as the original program and no one can restore the original program from the new program[8]. We can solve the privacy-preserving cloud-based firewall problem by constructing a cryptographic obfuscator $\mathcal{O}$. $\mathcal{O}$ can be used to construct a semantically equivalent secure firewall f' from firewall f such that f' has the same functionality as f.

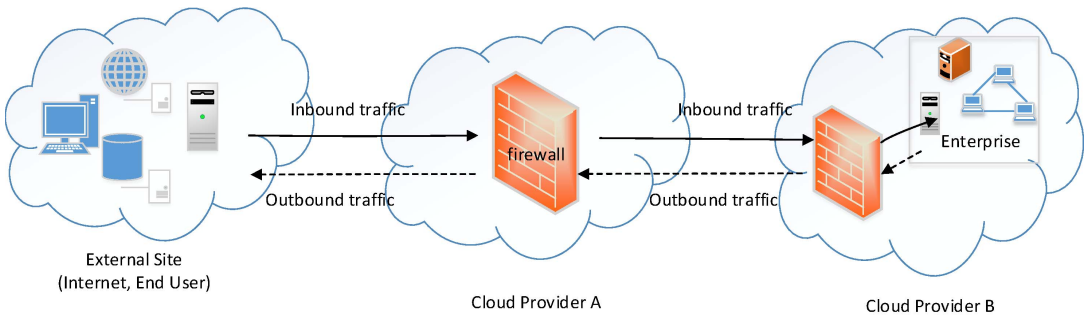


Fig. 1: System Architecture

Our framework consists of three phases. The first phase is the construction of an obfuscator for original firewall f. And the work is done by the administrator of the enterprise. The second phase is to match a packet with an obfuscated firewall rule. And this phase is done by CloudA. The third phase is to execute the action of the corresponding firewall rule. And this work is done by CloudB.

Obfuscation Phase: In the first phase, the administrator of the enterprise sets up the basic cryptographic parameter of the Paillier cryptosystem and CloudB sets up an asymmetrical public key cryptosystem. Then for every 5-tuple rule r and corresponding action in firewall f, such as these in Fig I, the administrator generates a corresponding obfuscated rule r' and generates the action's ciphertext under CloudB's public key. Subsequently, administrator aggregates these obfuscated rules to be an integral secure firewall f'. Finally, the administrator sends f' to CloudA.

Matching Phase: This phase is performed by CloudA. After receiving the obfuscated firewall f', CloudA starts filtering the packets. For each arrived packet, CloudA checks its packet header according to f'. If the packet matches with a rule r', then CloudA sends the packet and the ciphertext of action corresponding to r' to CloudB.

Action phase: When CloudB receives a packet and corresponding ciphertext from CloudA, it decrypts the ciphertext to get the corresponding plaintext action. Then CloudB processes the packet according to the plaintext action.

B. Our Scheme

[8] uses conjunction obfuscator [9] to obfuscate the firewall rules and then sends the obfuscated firewall to the cloud. As mentioned before, the construction of the multilinear maps used in [8] has been shown to be insecure. Thus in this paper, we modify the conjunction obfuscator by replacing the multilinear map with additional homomorphic encryption, the Paillier cryptosystem.

The detailed description of our scheme is given here. As discussed before, we present every rule of the firewall as $r = (\overrightarrow{v}, W, A)$. For A, the administrators encrypt it as $E(A)$ with the public key of the CloudB. For each bit in $\overrightarrow{v}$, for example the i-th bit, we generate two pairs of ciphertexts:

$$(u_{i,0}, v_{i,0}) \in (r_{i,0}^n \ mod\ n^2, E_{i,0}() \cdot r_{i,0}^n \ mod\ n^2) \quad (1)$$

$$(u_{i,1}, v_{i,1}) \in (r_{i,1}^n \ mod\ n^2, E_{i,1}() \cdot r_{i,1}^n \ mod\ n^2) \quad (2)$$

In Equation (1) and (2), $r_{i,0}, r_{i,1} \in Z_n^*$ are all chosen randomly and uniformly. $E_{i,0}(), E_{i,1}()$ is ciphertext of 1 or 0 i.e., E(0) or E(1). When $i \notin W$, if the i-th bit is 1, then $E_{i,1}()$ is E(0) and $E_{i,0}()$ is E(1); if the i-th bit is 0, $E_{i,1}()$ is E(1) and $E_{i,0}()$ is E(0). When $i \in W$, $E_{i,1}() = E_{i,0}()$ is E(0).

Except the two pair of ciphertexts of each bit, an additional pair of ciphertext is also generated for the whole rule:

$$(u_{k+1}, v_{k+1}) = (r_{k+1}^n mod n^2, r_{k+1}^n \cdot \prod_{i \in [k]} E_{i, \overrightarrow{v}[i]}() mod n^2) \quad (3)$$

Then the administrator sends the obfuscated firewall i.e., all the ciphertexts for each rule, and all the encrypted actions to CloudA. When CloudA receives the obfuscated firewall, it begins to perform the firewall function.

When a packet $\overrightarrow{p}$ arrives, CloudA computes the following Equations for every rule in the firewall one by one:

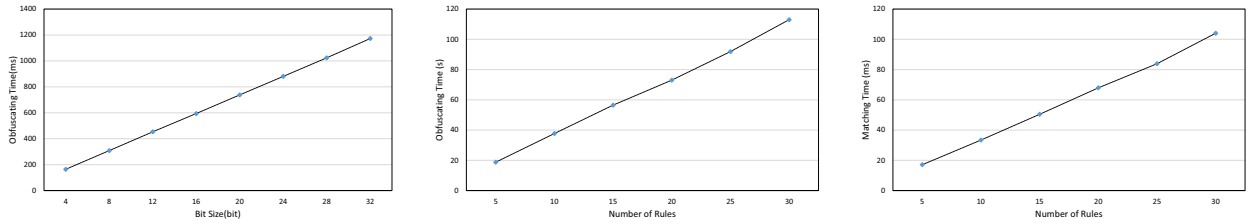
$$LHS = u_{n+1} \cdot \prod_{i \in [k]} v_{i, \overrightarrow{p}[i]} \ mod\ n^2 \quad (4)$$

$$RHS = v_{n+1} \cdot \prod_{i \in [k]} u_{i, \overrightarrow{p}[i]} \ mod\ n^2 \quad (5)$$

After computing the above Equations for a rule, CloudA checks whether $LHS = RHS$. If $LHS = RHS$, then the packet matches with the rule, CloudA stops computing the Equations for next rule and sends the packet along with the encrypted action $E(A)$ to CloudB. Otherwise, the packet does not match with the rule and CloudA continues to check the next rule.

When CloudB receives a packet from CloudA, it first decrypts the encrypted action using its private key. Then CloudB sends the packet to the enterprise or drops the packet according to the decryption result i.e., the Action.

Privacy: Intuitively, CloudA only learns which rule the packet matches with and does know what the rules and the corresponding actions are. CloudB only processes the packet according to the decryption and does not check the packet header, thus CloudB knows nothing about the firewall rules except the action. Note that our scheme can achieve the same security as the scheme in [8], [9]. We omit the proof here due to the page limit.



(a) Performance in Obfuscation Phase with Various Bits

(b) Performance in Obfuscation Phase with 5-tuple Rules

(c) Performance in Matching Phase

Fig. 2: Experimental Results

V. Experiment

We implement our scheme in Java using the Homomorphic Encryption Project [14] and evaluate it using real-life firewall rules. We conduct our scheme on a machine running Windows 7 professional with a 3.4 GHz Inter(R) Core(TM) i3-4310 CPU and 4GB RAM. We model the firewall rules as 5-tuple with the length of 104 bits. We use a 1024-bit modulus for Paillier cryptosystem in our experiments. And now we give the performance analysis of our scheme below.

Performance in Obfuscation Phase: Fig. 2a and Fig. 2b show the obfuscating time against increasing number of rule bit sizes and increasing number of firewall rules (with 5-tuple of 104-bit length). We can see that the obfuscating time increases linearly with the number of rule bit sizes and the number of firewall rules from the above two figures. More specifically, the obfuscating time of 32-bit size rule of our scheme is about 1.1 seconds, which is much more efficient comparing with the so-called naive scheme in [8], where the obfuscating time of 32-bit size rule is 4.1 seconds. Note that our scheme can be optimized to improve the efficiency the same as [8]. Fig. 2b plots the obfuscating time with various firewall rules and we can see that obfuscating time is about 112 seconds for 30 firewall rules (with 104-bit length of each firewall rule). The time is acceptable because the obfuscating phase needs to be done only once.

Performance in Matching Phase: Fig. 2c shows the performance of matching phase of our scheme. The matching time of our scheme is linear with the number of the firewall rules. The matching time for a packet and one firewall rule (5-tuple of 104-bit length) is about 3.46 ms as shown in Fig. 2c. Thus our scheme is much more efficient comparing with the scheme in [8], in which the execution time of the three different proposals is separately 300 ms, 100 ms and 6 ms.

Performance in Action Phase: When CloudB receives a packet and corresponding encrypted action, it decrypts the encrypted action. Note that CloudB can use any secure probabilistic public key cryptography. For simplicity, we assume that CloudB uses the RSA cryptosystem and evaluate the performance of Action Phase by evaluating the decryption time. We generate an instance of RSA cryptosystem with 1024-bit modulus for CloudB. In order to show the efficiency of our scheme in Action Phase, we test the decryption time by decrypting 100 ciphertexts. The results show that it takes about 1 ms in Action Phase for one packet.

VI. Conclusion

In this paper, we propose a new cloud-based firewall outsourcing architecture and then we propose a privacy preserving firewall outsourcing scheme for this architecture. We demonstrate the practicality and efficiency of our scheme by implementing a prototype of our scheme on our own commodity machine and we argue that the performance of our scheme can be improved when it is applied in the cloud.

References

[1] A. R. Khakpour and A. X. Liu, "First step toward cloud-based firewalling," in *Reliable Distributed Systems (SRDS), 2012 IEEE 31st Symposium on*. IEEE, 2012, pp. 41–50.

[2] T. Ritter, "Network-based firewall: Extending the firewall into the cloud." [Online]. Available: http://www.business.att.com/content/whitepaper/Nemertes_DN0496_Network-Based_Firewall_Services_May_2009.pdf

[3] J. Sherry, S. Hasan, C. Scott, A. Krishnamurthy, S. Ratnasamy, and V. Sekar, "Making middleboxes someone else's problem: network processing as a cloud service," *ACM SIGCOMM Computer Communication Review*, vol. 42, no. 4, pp. 13–24, 2012.

[4] R. Daley, "Cloud based firewalls: Overview and considerations." [Online]. Available: http://www.infosecwriters.com/Papers/RDaley_CBF.pdf

[5] M. G. Gouda and A. X. Liu, "Structured firewall design," *Computer Networks*, vol. 51, no. 4, pp. 1106–1120, 2007.

[6] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[7] L. Melis, H. J. Asghar, E. De Cristofano, and M. A. Kaafar, "Private processing of outsourced network functions: Feasibility and constructions."

[8] J. Shi, Y. Zhang, and S. Zhong, "Privacy-preserving network functionality outsourcing," *arXiv preprint arXiv:1502.00389*, 2015.

[9] Z. Brakerski and G. N. Rothblum, "Obfuscating conjunctions," in *Advances in Cryptology–CRYPTO 2013*. Springer, 2013, pp. 416–434.

[10] J.-S. Coron, T. Lepoint, and M. Tibouchi, "Practical multilinear maps over the integers," in *Advances in Cryptology–CRYPTO 2013*. Springer, 2013, pp. 476–493.

[11] J. H. Cheon, K. Han, C. Lee, H. Ryu, and D. Stehlé, "Cryptanalysis of the multilinear map over the integers," in *Advances in Cryptology–EUROCRYPT 2015*. Springer, 2015, pp. 3–12.

[12] G. Gibb, H. Zeng, and N. McKeown, "Outsourcing network functionality," in *Proceedings of the first workshop on Hot topics in software defined networks*. ACM, 2012, pp. 73–78.

[13] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in cryptologyłEUROCRYPT99*. Springer, 1999, pp. 223–238.

[14] thep - the homomorphic encryption project - computation on encrypted data for the masses - google project hosting. [Online]. Available: https://code.google.com/p/thep/