Encrypted Packet Forwarding in Virtualized Networks

Sriram Natarajan and Tilman Wolf
Department of Electrical and Computer Engineering
University of Massachusetts
Amherst, MA, USA
{snataraj,wolf}@ecs.umass.edu

ABSTRACT

Virtualized networks provide a shared infrastructure platform for hosting multiple independent networks with different protocol stacks. The infrastructure and the virtual networks are operated by different entities who may not trust each other. In our work, we address one of the arising security issues by providing data confidentiality for forwarding network traffic. We propose an encrypted representation of IP addresses and forwarding data structures that hides the operations of the virtual network from the infrastructure provider. We describe the cryptographic computations and data structures that forwards network traffic and discuss their space requirements.

General Terms

Design, performance, security

Categories and Subject Descriptors

C.2.6 [**Computer-Communication Networks**]: Internetworking—*Routers*

1. INTRODUCTION

Network virtualization provides a platform to host diverse set of protocol suites (multiple logical networks) to coexist on a shared network infrastructure [1]. The network virtualization platform proposes a three-tier architecture with the following entities: (1) network infrastructure (NI), which provides the required physical resources (e.g., routers and switches) to setup the network, (2) virtual networks (VN), which deploy the customizable services (e.g., network protocols), and (3) users, who connect to the desired virtual network services. Although the architectural separation of virtual networks from the network infrastructure introduces flexibility to deploy new protocol, there are various new security issues that arise in this context.

Each component of the architecture may be operated by different administrative entities and hence mutual distrust may exist between the hosted VNs and the NI. A VN operator may not want to disclose information about traffic and route configurations to the NI, but existing cryptographic protocols only protect packet payloads (and some higher-layer packet header). The control information in the IP header needs to be visible to the NI in order to perform packet forwarding. In our work, we propose a technique for encrypting IP addresses and forwarding tables in such a way that packet forwarding can still be performed by the NI but without gaining insight on the operation of the VN.

2. IP ADDRESS ENCRYPTION

For an effective packet forwarding functionality, the proposed technique should be: (1) secure from attacks, (2) efficient in forwarding performance, and (3) effective with respect to time and space complexities. End-nodes in the VN encrypt each packet's IP addresses using a probabilistic encryption technique. Payloads are encrypted using conventional cryptographic protocols (e.g., VPN, SSL).

To avoid inferences about traffic patterns, the IP address is transformed into multiple different cipher representations using a probabilistic encryption technique. The following steps describe the forwarding information encryption process: (1) Choose a function, f, number randomly (each encryption function is associated with a function number) from the range of available encrypted functions, (2) concatenate 32 bits from the IP address with k bits from the function number, (3) encrypt the forwarding information bits using a stream or a block cipher scheme as shown in [4], precomputed and represented using a binary decision diagram (BDD), (4) encrypt the function number bits using an Optimal Asymmetric Encryption Padding (OAEP) scheme as shown in [2], precomputed and represented using a hash table, (5) concatenate the encrypted function number and the encrypted forwarding information bits to generate the final encrypted forwarding information bits.

To decrypt the IP address, the following steps are used: 1) Receive the encrypted forwarding information bits, 2) consider the first k bits to determine the function number (representing the encrypted function) used in encrypting the forwarding information bits, 3) decrypt the function number using the OAEP scheme, precomputed and represented using a hash table, 4) decrypt the forwarding information bits using the stream or a block cipher scheme, precomputed and represented using the BDD, 5) retrieve the original forwarding information bits and determine the outgoing interface, 6) update the forwarding table in the network infrastructure.

The encrypted packet forwarding can be effective only when the forwarding rate (lookup) performance is not significantly constrained by the cryptographic computations. Hence, we precompute the entire encryption and decryption functions and reduce the computations to performing lookup operations on the data structures. In the following section we discuss the required lookup data structures to perform the precomputed cryptographic computations.

3. ENCRYPTED FORWARDING TABLE LOOKUP

Since each function number is represented by k bits, the

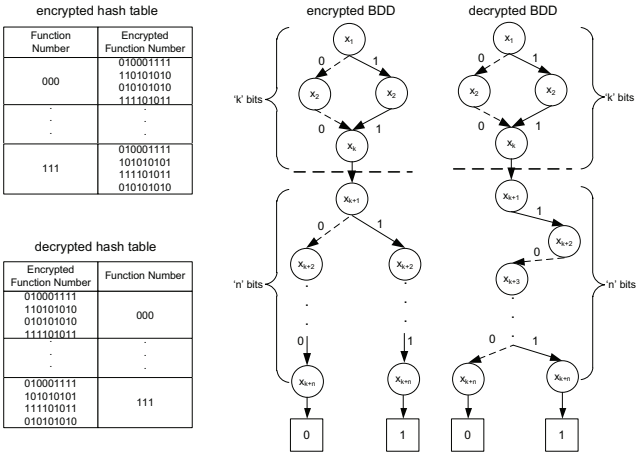

Figure 1: Data Structure Representation

total number of possible encryption functions is 2^k. The number of different cipher texts for each f is denoted as s. Hence, the total number of encrypted forwarding addresses for a single forwarding address is given by the product of number of distinct encryption functions (2^k) and the number of different cipher texts associated with each function number (s).

Using the OAEP technique, the encrypted function numbers are pre-computed and stored in a hash table. In the hash structure, multiple encrypted function numbers are mapped to the same original function number. In the encryption hash table, each function number is added as the hash key and the encrypted values are stored as the hash values. Figure 1 shows the pre-computed encryption/decryption hash tables. To encrypt a packet, a function number is chosen and an associated encrypted function number is randomly picked from the encryption hash table to concatenate with the encrypted forwarding address, resulting in the final encrypted forwarding address. To decrypt the function number, extract the encrypted function number bits from the encrypted packet and lookup in the decryption hash table to determine the original function number.

To perform destination address lookups in the encrypted domain, we consider a data structure to store the pre-computed cryptographic function bits that is efficient in lookup time and total memory requirements. For an effective forwarding rate performance, we pre-compute all the computations involved in encrypting and decrypting the forwarding address. Although the pre-computation improves the processing and lookup time, the total memory required is large. To avoid the memory space explosion, we consider a binary decision diagram that reduces the total memory required to store the cryptographic function bits.

Figure 1 shows the BDD representation of the cryptographic function bits. The first k variables (bits) in the BDD represent the plain text function number $(x_0, x_1, ...x_{k-1})$ bits. The next n variables (bits) represent the cryptographic function bits $(x_k, x_{k+1}, ...x_{k+(n-1)})$. To generate the encrypted forwarding address, we traverse the encryption BDD based on the original address bits and record the bit pattern in each node in the BDD. The original address bits are XORed with the cryptographic (encryption) function bits to generate the encrypted address bits. The decryption process similarly XOR's the received encrypted address bits with the cryptographic (decryption) function bits to retrieve the original address and determine the packet's outgoing interface.

Table 1: Space Complexity Analysis

Forwarding information	size (bits)	data structure size
Original address	n	$O(2^n)$
Encrypted address	$n+k$	$O(2^{n+k})$
ROBDD-based address	$n+k$	$O(2^{n+k}/(n+k))$

The forwarding rate performance of the proposed encrypted packet forwarding technique is dependent on both the size of the encrypted address and the data structure used for lookup computation. The encrypted address is not prefix preserving and is larger in size than the original forwarding address size. Hence, an effective data structure is required that is efficient in both time and space complexities. To address this issue, we consider a reduced ordered binary decision diagram (ROBDD) which performs lookup operation on the compressed representation of the encrypted address bits as shown in [5]. The encrypted address size is reduced by applying the ordering and reduction properties to construct the ROBDD, as shown in [3]. An analysis on the space complexity requirement for the ROBDD based lookup computation is shown in Table 1.

4. SUMMARY

In this work, we discussed how secure forwarding can be implemented in network virtualization. We presented an encrypted packet forwarding technique that encrypts IP addresses and allows the untrusted infrastructure to perform forwarding decisions based on the encrypted forwarding information. We illustrated the encryption and decryption process and identify the set of data structures that can satisfy the forwarding rate performance requirements. In the near future, we plan to prototype and evaluate this technique is a virtualized network testbed.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. CCF-0952524.

5. REFERENCES

[1] T. Anderson, L. Peterson, S. Shenker, and J. Turner. Overcoming the Internet impasse through virtualization. *Computer*, 38(4):34–41, Apr. 2005.

[2] M. Bellare and P. Rogaway. Optimal asymmetric encryption how to encrypt with RSA. pages 92–111. Springer-Verlag, 1995.

[3] R. E. Bryant. Graph-based algorithms for boolean function manipulation. *IEEE Trans. Comput.*, 35:677–691, August 1986.

[4] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Oct. 1996.

[5] R. Sangireddy and A. Somani. High-speed IP routing with binary decision diagrams based hardware address lookup engine. *Selected Areas in Communications, IEEE Journal on*, 21(4):513 – 521, may 2003.