Protecting composite IoT server by secure secret key exchange for XEN intra virtual machines

Anil Yadav*

Information Technology,
Amity University,
Uttar Pradesh, Noida, 201313, India
Email: anilyadav05@outlook.com
*Corresponding author

Anurag Tripathi

Electrical Engineering,
IIT Delhi,
New Delhi, India
Email: eez128368@ee.iitd.ac.in

Nitin Rakesh

Computer Science and Engineering,
ASET, Amity University,
Uttar Pradesh, Noida, India
Email: nitin.rakesh@gmail.com

Sujata Pandey

Electronics and Telecom Engineering,
Amity University,
Uttar Pradesh, Noida, 201313, India
Email: spandey@amity.edu

Abstract: Challenges presented by ever increasing volumes of smart objects are huge in terms of security and privacy of data generated out of these objects. These challenges range from the usage scenario with other objects and dedicated servers which act as a middleman among various networks such as internet and wireless sensor network. In this paper, attempt has been made to highlight a simple use case of a composite server in a XEN-based virtualised system. Security threats like sniffing and spoofing are isolated and analysed with respect to a novel key transfer method between host and guest operating system in XEN-based virtualised system. In addition, secure and trust model to cater the said security threats like sniffing and spoofing is also presented. The performance of the proposed system in terms of CPU usage and network bandwidth is also shown.

Keywords: smart objects; security; privacy; XEN; IOT; sensor.

Reference to this paper should be made as follows: Yadav, A., Tripathi, A., Rakesh, N. and Pandey, S. (2020) 'Protecting composite IoT server by secure secret key exchange for XEN intra virtual machines', *Int. J. Information and Computer Security*, Vol. 12, No. 1, pp.53–69.

Biographical notes: Anil Yadav is a PhD Research Scholar in Information Technology at Amity University, Noida, India. He received his ME degree in Computer Science from Birla Institute of Technology and Science, Pilani, India in 2002, and his BTech degree from BIET Jhansi India in 2000. He is a Senior Researcher at the Security Lab, SRI Delhi, India, since 2003. He has extensive experience in protocol design and implementation for consumer electronics devices. He has contributed to security and privacy protocol design for secure access control, key provisioning. He has developed, productised as well as lead the analysis of related security solutions. His research interests include but not limited to protocol design for security and privacy for the internet of things and blockchain.

Anurag Tripathi earned his MTech from IIT Guwahati in Computer Science and currently pursuing his PhD from IIT Delhi in EE (C.Tech.) his research areas are image/video annotation and captioning. He has over 13 years of experience in the industry. Prior to joining Accenture, he worked as a Principal Engineer in Samsung from 2008 to 2017. He was awarded SW Architect certificate from Samsung Electronics for his contribution in architecting various software framework for Consumer Electronics devices. Prior to joining Samsung, He has worked in Aricent Technologies from 2005 to 2007 where he worked on the optical network, WiMAX etc. He worked in multiple domains like Linux kernel, device driver, image/video processing, machine learning, etc. As an academic researcher, he has shown his presence in the different area e.g., IOT, intelligent image/video processing by publishing research papers in those areas.

Nitin Rakesh received his Doctorate in Department of Computer Science and Engineering from JUIT, Waknaghat in 2012. In 2007, he received his Master of Technology Degree in Computer Science and Engineering from Jaypee Institute of Information Technology, Noida, India and received Bachelor in Technology Degree in Information Technology from AEC, Agra in 2004. He is the Deputy Head Corporate Resource Centre and Associate Professor in the Department of Computer Science Engineering and with Amity School of Engineering and Technology, Amity University Campus, Noida, India-201313. He is member of IEEE, ACM, SIAM, IAENG and Life member of CSI. He is a recipient of Drona Award for TGMC-2009 by IBM. His research outlines emphasis on network coding, interconnection networks and architecture, fault-tolerance and reliability, networks-on-chip, systems-on-chip, network algorithms, parallel algorithms and fraud detection, online phantom transactions.

Sujata Pandey earned her Master's in Electronics (VLSI) and PhD in Electronics from university of Delhi. Currently, she is working as a Professor at Amity University Uttar Pradesh. She has over 200 research publications in reputed international journals/ conferences. Her areas of research are microelectronics, analogue/digital VLSI design, and energy harvesting. She is a member of IEEE, USA, member of Electron Device Society, IET UK, founder member of VLSI and Semiconductor Society of India, ISTE and life member of Indian Science Congress.

1 Introduction

Creation and management of ecosystem of smart things along with traditional internet have thrown huge challenges. Such ecosystem required basic services along with smart services (Mukherjee et al., 2014) to form the major building blocks of the ecosystem. Various smart applications such as smart-city (Valerio, 2016; Zanella et al., 2014), medical industry, commuting, shopping and education constitute diverse network of connected sensor devices that seamlessly interact to each other. Moreover, these sensors capture and process data for their assigned roles. These devices are resource hungry in terms of processing power, storage but have sufficient intelligence to execute their assigned functions. In addition, regular status update for notification or transmission of data sensed to a remote observation system through Internet is important. Also, seamless and un-interrupted connectivity between the networks is required. Transmission of sensitive and confidential data must meet various security and privacy requirements (Arias et al., 2015) and cater the threats evolved with such interaction.

Researchers (Reiser, 2009) have highlighted things' usage scenarios. Open authorisation protocol is used to provide the authorisation services for smart things. Challenges of such services for smart objects are similar to servers that have traditionally based on service oriented architecture. In such scenario, virtualisation (Miao and Han, 2011; Leja et al., 2008) facilitates the sharing of resources and information to other connected components such as computer or devices. Traditionally, hypervisor (http://en.wikipedia.org/wiki/Hypervisor) allows deployment of virtual machine on same hardware platform to construct a server. Security vulnerability of a virtual machine (guest) is a major issue to be addressed in such scenario. Bridge and route modes for virtual network a combined approach is proposed (Wu et al., 2010) so that guests can communicate securely in a virtual network. Authors Kirch (2007) and Goyette and Karmouch (2011) have highlighted various security issues like sniffing, spoofing and denial of services (DOSs) in traditional virtualisation techniques. Virtualisation needs separation for guests within virtual environment of the hardware platform and such separation introduces security threats in the virtualised environment (Discover the Linux Kernel Virtual Machine, 2007; Garfinkel and Rosenblum, 2005). A dynamic model for assessing the security of a virtualised system in routing, firewall and shared network layers is available (Bazargan et al., 2011). Current virtual network model highlights the guests' vulnerabilities like sniffing and spoofing. Bridge acts as a virtual hub in bridge mode with which all guests communicate in the network. A guest in such scenario will be sniffed by a sniffer tool 'Wireshark' (2010). Route acts as 'virtual switch' in route mode. The virtual switch utilises an associated virtual interface to connect to each guest in the network. A guest in such case can perform an address resolution protocol (ARP) spoofing (Phelan, 2008) and redirect packets to respective guest and sniff the packets traversing among the guests. A secured environment model only for inter-domain (inter virtual machine) communication that is a major security threat for a server environment which exists in between two heterogeneous networks is available (Wu et al., 2010). Security for the trusted guests belonging to the same group is not provided. Additionally, creation and configuration of the guests specifically for an organisation required dedicated administrator. Furthermore, it does provide security solution neither for an individual guest nor for an application who is more interested to utilise the services of a guest on the server.

Reiser (2009) additionally described the secure services provided by internet of things server (IoTS). Also the trust problems with the wireless sensor networks (WSN) system combined with traditional internet protocols is highlighted. Advantages and limitations of the open authorisation model are also described by authors. In this model major security vulnerabilities existing due to absence of trust model, man in the middle (MITM), DOS by server flooding.

Primarily these servers must ensure to facilitate functional services to the smart objects. Additionally these services must be extended to build a trust model among the smart sensor objects and their users. The functional aspect of services is different for each object in interaction but to create a trust model it must include secure authentication and authorisation, secure communication and storage as the primary security services to cater the security issues arisen from internet and within servers. Over the years virtual machines have been utilised to deliver the various requests on demand. Such requests include from resource allocation to execution of specialised tasks. However, there are various security concerns in the services offered.

In this paper, attempt has been made to highlight a simple use case of such composite server in a XEN-based virtualised system, security threats like sniffing and spoofing are isolated and finally presented a simple and secure trust model to cater the said security threats like sniffing and spoofing. Current work focuses to design and implement a method to cater security issues of sniffing, spoofing-based MITM attack for virtual network based on XEN. It is a server that processes protocols (standard internet protocol and IOT protocols) such as Transmission Control Protocol (TCP), User Data-gram Protocol (UDP), Constrained Application Protocol (CoAP), Pv6 over low power wireless personal area networks (6LoWPAN), etc. In the current work attempt has been made to create a key transfer method between guests and host. The key is symmetric in nature and used to encrypt the packets transmitted across the virtual machines. System design considerations are discussed in Section 2, test results and security analysis are done in Section 3. Finally, conclusion and future improvements are mentioned in Sections 4.

2 System design consideration

Simple use case scenario of smart sensor network and Internet with XEN-based server interacting with remote observation system is shown in Figure 1.

A sample smart home application depicted in Figure 1 requires the services from device discovery and registration, device authentication and authorisation, event notification and alarming, privacy, secure storage and communication (Phelan, 2008). The services offered by composite server are shown in Figure 2.

A XEN-based virtual machine hosts the services in a virtual network and a logical view of Figure 1 is shown in Figure 3. The virtual machine processes both internet protocols and WSN protocols for IOT [such as 6lowpan (Varadarajan and Crosby, 2014; Kim et al., 2013), CoAP (https://datatracker.ietf.org/wg/core/charter/), etc.] with a translation module between them. The scenarios could be described like a remote monitoring system requests the data of a smart object in the smart home. This request travels to composite IOT server over a traditional Internet and translation module coverts this request to IOT compliant request to sense the data out of the smart object.

Figure 1 Simple use case scenario of smart sensor network and internet with XEN-based server interacting with remote observation system (see online version for colours)

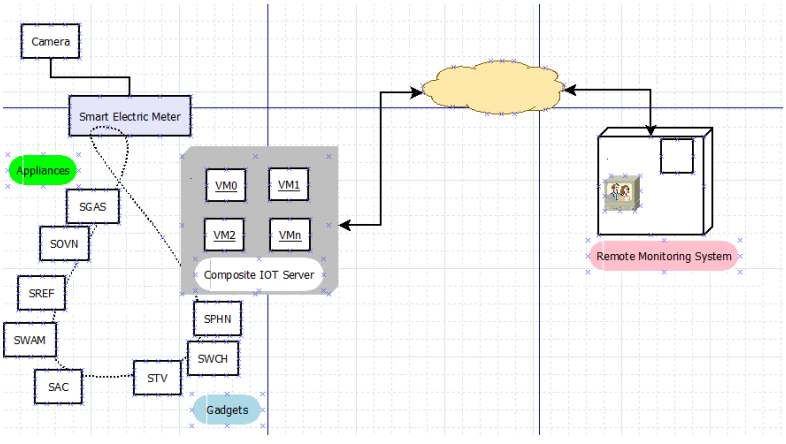


Figure 2 Services offered by composite server (see online version for colours)

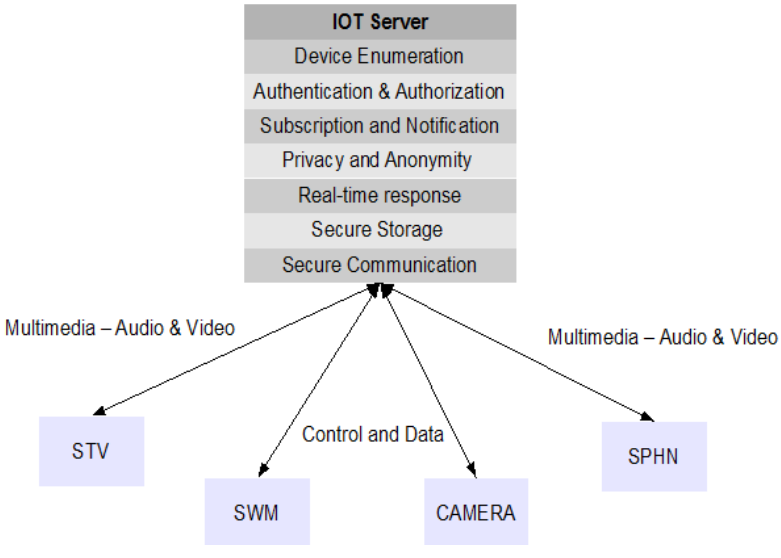
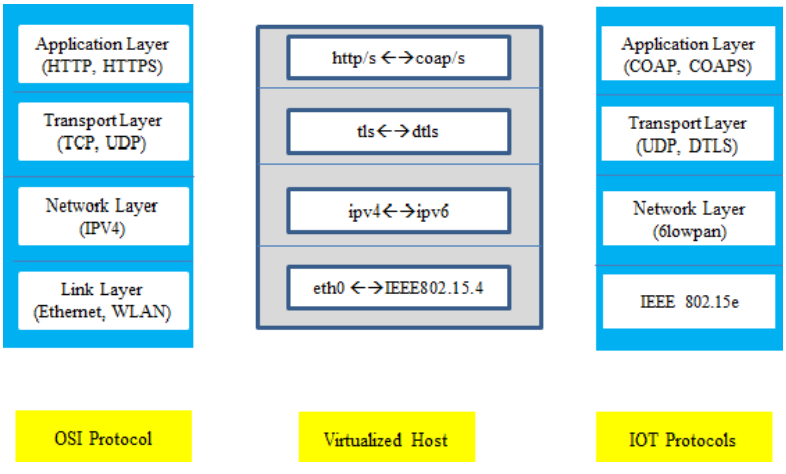


Figure 3 Visualised host with composite server (see online version for colours)



To meet the IOT application's requests working groups over the globe have defined several specifications. Primary focus of these standards and specifications are to device protocols to meet the requirements of resource constrained sensor objects. The Internet Engineering Task Force (IETF) has devised 6LoWPAN (Varadarajan and Crosby, 2014; Kim et al., 2013) for IPV6 packet transmission over IEEE 802.15.4 (http://tools.ietf.org/wg/6lowpan/). Framework for light-weight versions of already existing popular protocols are defined by constrained restful environments (CoRE) (Montenegro et al., 2007). CoAP (https://datatracker.ietf.org/wg/core/charter/) is analogous to HTTP but is a lighter version of HTTP. CoAP is defined by CoRE working group and have analogous operations such as GET (operation performed by HTTP and CoAP to get data), operation performed by HTTP and CoAP to send data (POST), and DELETE (operation performed by HTTP and CoAP to delete data). Instead of TCP, UDP is the layer underneath CoAP. TCP is used by transport layer security (TLS) whereas UDP is used by DTLS. Security at transport layer is provided by TLS and datagram transport layer security (DTLS) (Shelby et al., 2013; Dierks and Rescorla, 2008).

Service-based architecture depicted in Figure 3 caters the needs of sample smart home application. A virtualised network is depicted where virtual machines utilise a common physical network to isolate from each other with the help of a common host. A virtual network (http://wiki.kartbuilding.net/index.php/XEN_Networking) can be configured with operational modes that are briefly described below.

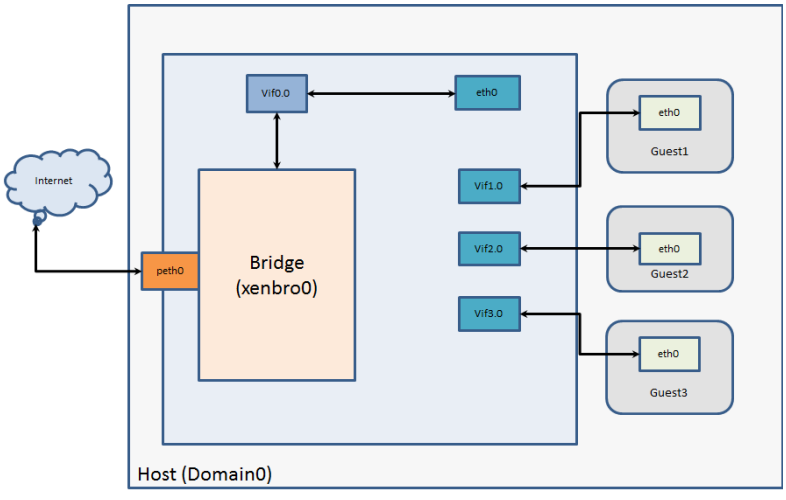
2.1 Bridged network mode

The simplest form of bridge networking as shown in Figure 4 is the depicted for XEN (http://XEN.org/files/Marketing/WhyXEN.pdf and http://www.XEN.org/files/Marketing/HowDoesXENWork.pdf); it is the most convenient technique to configure with. Such networking techniques facilitate virtual machine users to configure a virtual Ethernet card to connect to an already existing network (Netfilter kernel module, 2012). Figure 4 explains the 'bridge network' and its major characteristics are:

1 Every guest machine exists within same IP range.

2 No active participation of virtual hub called as domain0, except forwarding traffic to destination guest machines and internet.

Figure 4 XEN-based bridge network (see online version for colours)



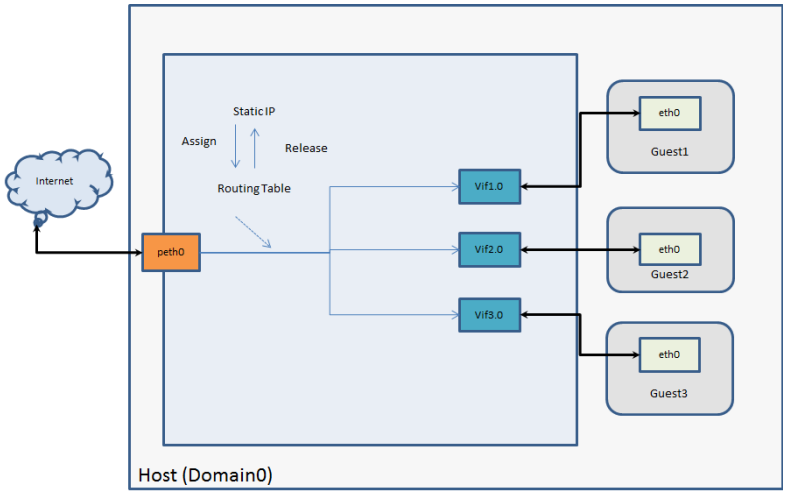
2.2 Routed network mode

Characteristics of a XEN-based routed network as shown in Figure 5 guests are:

1 guests are in separate IP LAN

2 guest's traffic can be routed to internet.

Figure 5 XEN-based routed network (see online version for colours)



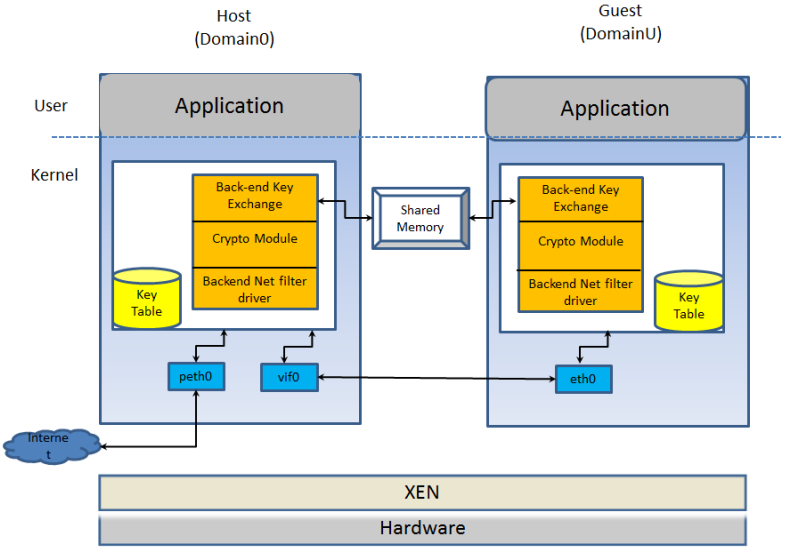
2.3 Proposed composite server approach

Figure 6 represents the composite server architecture for the implementation of secure key exchange and packet encryption. In the proposed approach packets transmitted across virtual interface (vifn) of host and network interface (eth0) of client domain are encrypted/decrypted and vice-versa as illustrated in Figure 6. Prior to transmission of any packets across the host and guest, they secret keys are exchanged. In the model, the packet transmission between host and guests appears in following two scenarios:

1 When a packet received from internet on to peth0 of host and destined towards a guest (virtual machine). Host encrypts packet and then transmits it to virtual interface (vifn) on the receiving guest (virtual machine). Guest decrypts the received packet and resumes with its remaining operations.

2 If a packet is required to be transmitted from a guest over through eth0 existed in client domain and the destination is Host. In this case guest encrypts the packet and sends it to host. On receipt of the packet, Host decrypts the encrypted packet and resumes the normal operation. Shared memory-based IPC is used to transmit the secret key between the host (Domain0) and guest (client domain). XEN hypervisor supports such infrastructure. After successful sharing of secret key between host and guest, encrypted data transfer start within the virtualised network.

Figure 6 Composite server architecture (see online version for colours)



This approach constitutes of following major functions.

2.3.1 Generation and transmission of secret key

Secret key is generated while guests are being created as shown in Figure 7(a). Figure 7(b) depicts the detailed secret key generation and transmission process. The major steps involved in the process of secret key generation and exchange are briefly described below:

1 Host (Domain0) randomly generates a secret key while creation of each guest (domains).

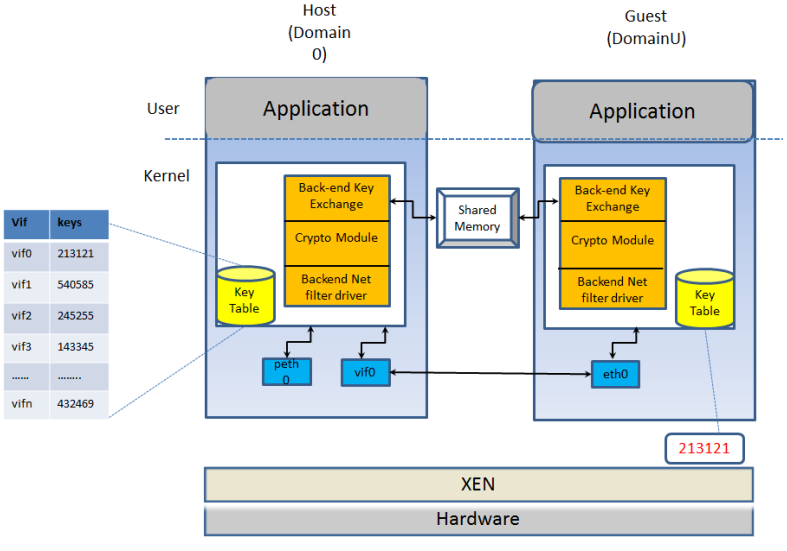
2 The generated secret key is shared with guest through unique key exchange pipe (KEP).

3 KEP uses the shared memory-based IPC mechanism supported by hypervisor (Cho and Jeon, 2007). To share the pages across the guests, XEN internally uses a grant table (Chisnall, 2008) method.

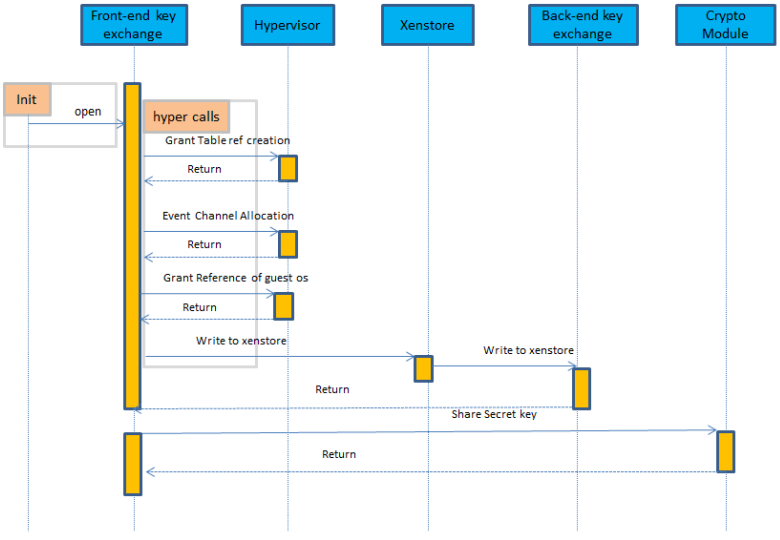
4 The host (Domain0) maintains a secret key table (key DB) of all its guests (client domains).

5 Prior to the packet transmission over the network, every guest (client domains) is assigned with a unique secret key.

Figure 7 (a) Generation and transmission of secret key (b) Sequence diagram of secret key generation and transmission (see online version for colours)



(a)



(b)

2.3.2 Host to guest data transmission

Host to guest data transmission mechanism as shown in Figure 8(a) are described below:

1 If host (Domain0) has received a packet which is directed towards any guest (client domain), before the packet transmission a backend netfilter kernel module (2012) (back-end netfilter driver) at Host catches the packet and its payload is encrypted with the secret key maintained in key-table for the targeted guest (client domain).

2 If a guest has received a packet from a host (Domain0), another netfilter kernel module (2012) (front-end netfilter driver) at the Guest (Client Domain) decrypts the Payload of the Packet and resumes the normal operation.

3 Netfilter modules (front-end and back-end) interact with their respective crypto module over the predefined interfaces for encryption and decryption of the packets transmitted across the virtual network.

2.3.3 Guest to host data transfer

The data transfer mechanism between guests to host as shown in Figure 8(b) involves following steps:

1 On receipt of a packet at host (Domain0) over the virtual machine network for any guest (client domain), before delivery of the packet the backend netfilter module (http://en.wikipedia.org/wiki/Hypervisor) (backend netfilter driver) at Host traps the network packet. After encryption of packet payload with the secret key maintained in the key table (key database) for the respective guest (client domain).

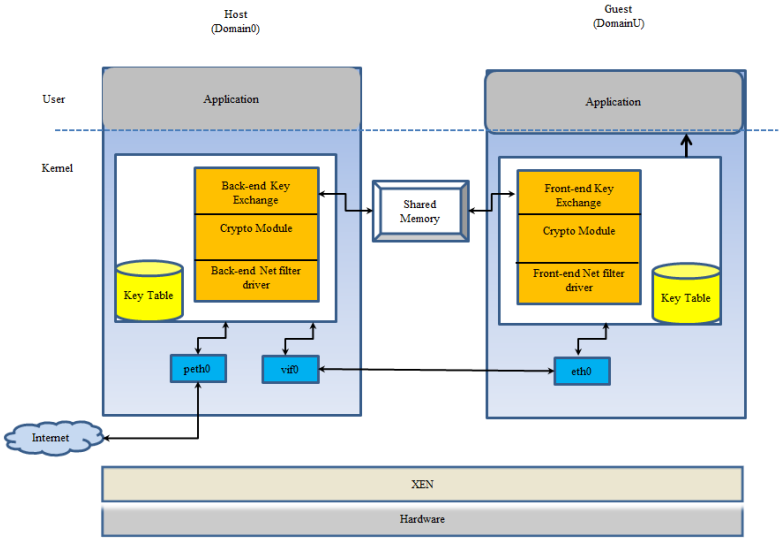
2 If the packet is received from the host (Domain0) another frontend netfilter kernel module (http://en.wikipedia.org/wiki/Hypervisor) at the guest (client domain) traps the packet. Normal operation flow resumes after decryption of the packet payload by the frontend netfilter module.

3 Backend netfilter driver interacts with crypto module for encryption/decryption the packet.

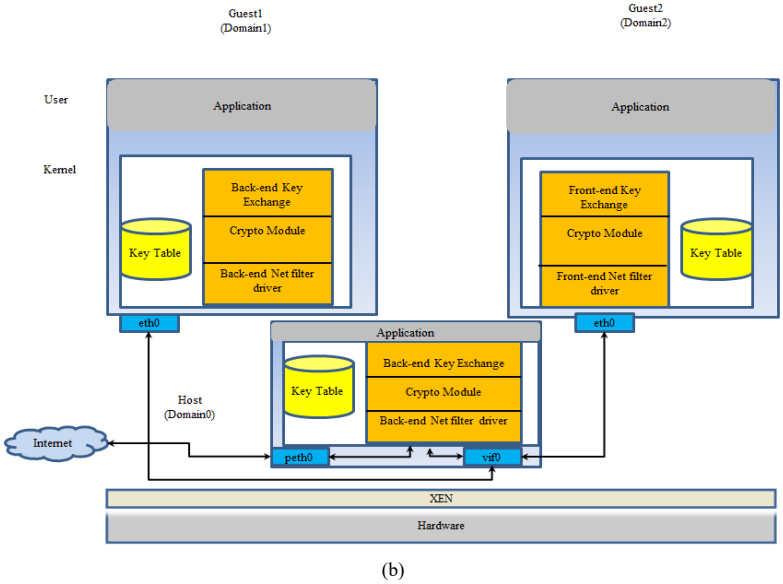
4 Final destination of the packet is retrieved from the routing table kept at host after successful accomplishment of original shape of the packet.

5 Packet encryption/decryption applies to communication within the virtualised network, not to the packets transmitted over the internet.

Figure 8 (a) Host to guest data transmission (b) Guest to host data transfer (see online version for colours)



(a)



(b)

2.3.4 Removal of secret key

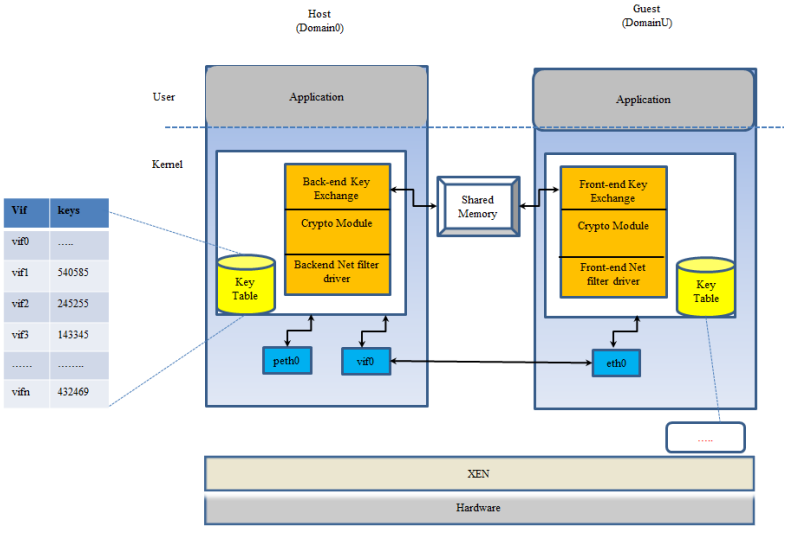
Secret key at guest is removed from the key table while shut-down process at the guest, as shown in Figure 9. The detailed steps of secret key removal is described below:

1 Secret key is removed from a guest's own key table just before it's shut-down.

2 Afterwards, Host is notified with shut-down event by the Guest with a virtual interrupt (Cho and Jeon, 2007).

3 Host deletes the entry of secret key from its key table (key database) pertaining to guest who notified with virtual machine shut-down event.

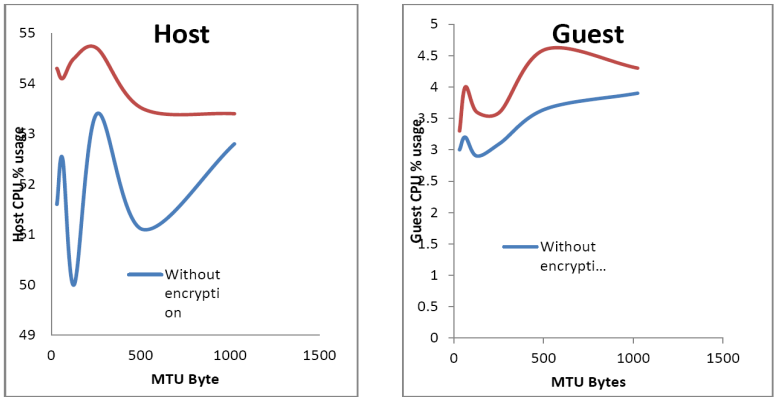
Figure 9 Deletion of secret key (see online version for colours)



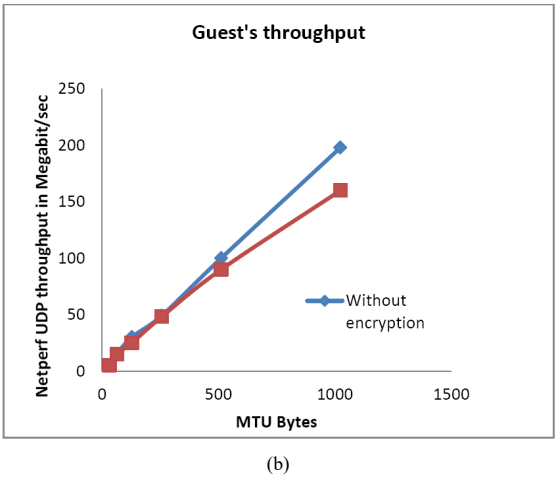
3 Experimental results

For simplicity XOR-based encryption is deployed in the proposed framework. Experimental results are produced by keeping secret key size as 16 byte. Results are illustrated in Figure 10. The CPU utilisation across host and guest are depicted with multiple sizes maximum transmission unit (MTU). CPU usage is calculated with 'top' utility with and without encryption. A benchmarking tool netperf (Jones, 2015) used for evaluation of the performance across the virtualised network. UDP bandwidth is calculated in megabits/sec with UDP_STREAM in netperf. Results of with and without KEP and encryption is depicted in Figure 10(b). The experiment is carried out on XEN hypervisor (3.3.0), both host and guest kernel version is 2.6.18.8. Test bed used is shown in Table 1.

Figure 10 (a) CPU usage chart at host and guest with and without KEP and encryption (b) Guest's throughput with netperf (see online version for colours)



(a)



(b)

Table 1 Test bed

H/W	*Intel Pentium 4*
Frequency	3 GHz
Cache memory	1 MB
RAM	2 GB
Operating system version	FC 8
Host memory	256 MB

3.1 Security analysis

- *Key protection:* the KEP and encryption uses a unique secret key. The key is generated at host while a new guest is created. The secret key is shared to guest at the time of guest's initialisation process. While the shut-down process of the guest, this secret key is deleted. Each of the guests uses their own secret key for encryption and decryption. The KEP is implemented over the shared memory and virtual interrupt mechanisms of XEN.

  Prior to network establishment of guest, secret key sharing between host and guest is done through the shared memory channel, so there is no way this key is exposed to other guests. On closure request of the guest, this secret key is deleted from the guest key table. So, the keys are well protected in the proposed solution. Sniffing and

- *Spoofing:* Wireshark (http://openmaniak.com/wireshark.php) based sniffing attack is eliminated by the current approach. As the packets being transmitted are encrypted by secret key at host and guests, sniffing of packets through wireshark only provides encrypted packets. Same approach applies to spoofing also. Packets being encrypted are of no help to attackers.

4 Conclusions

Authors proposed a composite server for things. The composite server acts as a common interface between two different networks. The security issues pertaining to such server is highlighted by utilising XEN hypervisor's infrastructure. Solution to security issues like sniffing, spoofing in an intra-virtual machine communication on a virtualised platform using a secure key transmission and encryption method are proposed. Finally, the performance in terms of CPU usage and network bandwidth is depicted based on the design and implementation of proposed approach.

The proposed work can further be explored in multiple ways; first more robust encryption and decryption mechanism adaption for implementation of the current work on XEN virtualised environment for the validation of security loop-holes, impact of performance incurred due to encryption and decryption. Second this work can be extended to restructure IPsec architecture in things environment so that secret key along with secure communication methods in proposed server environment can be provided. Third extension of current work is to provide include sensor data generated from sensor devices and providing trusted security services for authentication, authorisation and storage.

References

[online] http://wiki.kartbuilding.net/index.php/XEN_Networking (accessed 20 May 2015).

Arias, O., Wurm, J., Hoang, K. and Jin, Y. (2015) 'Privacy and security in internet of things and wearable devices', *IEEE Transactions on Multi-Scale Computing System*, Vol. 1, No. 2, pp.99–109.

Bazargan, F.A., Yeun, C.Y. and Zemerly, J. (2011) 'Understanding the security challenges of virtualized environments', *Internet Technology and Secured Transactions* (*ICITST*), *2011 International Conference*, 11–14 December, pp.67–72.

Chisnall, D. (2008) *The Definitive Guide to the XEN Hypervisor*, Prentice Hall, ISBN 9780132349710 [online] https://books.google.co.in/books?id=km99jpOs4N4C.

Cho, Y.C. and Jeon, J.W. (2007) 'Sharing data between processes running on different domains in para-virtualized XEN', *Control Automation and Systems*, *ICCAS'07*, 17–20 October, pp.1255–1260.

Dierks, T. and Rescorla, E. (2008) '*The Transport Layer Security* (*TLS*) *Protocol Version 1.2*, RFC 5246, August, Updated by RFC 5746, 5878.

Discover the Linux Kernel Virtual Machine (2007) April [online] https://groups.google.com/forum/#!topic/vmkernelnewbies/XjTkP2yLpFA (accessed 26 May 2015).

Engelhardt, J. (2010) *Writing Netfilter Modules*, July [online] http://inai.de/documents/Netfilter_Modules.pdf (accessed 23 May 2015).

Garfinkel, T. and Rosenblum, M. (2005) 'When virtual is harder than real: security challenges in virtual machine based computing environments', in *HOTOS'05: Proceedings of the 10th conference on Hot Topics in Operating Systems*, Berkeley, CA, USA, USENIX Association, p.20.

Goyette, R. and Karmouch, A. (2011) 'A dynamic model building process for virtual network security assessment', *Communications*, *Computers and Signal Processing* (*PacRim*), *2011 IEEE Pacific Rim Conference*, 23–26 August, pp.482–487.

Hypervisor [online] http://en.wikipedia.org/wiki/Hypervisor (accessed 18 May 2015).

IETF 6LoWPAN Working Group [online] http://tools.ietf.org/wg/6lowpan/ (accessed 19 April 2015).

IETF Constrained RESTful Environment (CoRE) *Working Group* [online] https://datatracker.ietf.org/wg/core/charter/ (accessed 18 April 2015).

Jones, R. (2015) *A Network Performance Benchmark* [online] http://www.netperf.org (accessed 15 May 2015).

Kim, E., Kasper, D., Chevrollier, N. and Vasseur, J.P. (2013) *Design and Application Spaces for 6LoWPANs draft-ietf-6lowpan-usecases-09*, January.

Kirch, J. (2007) *Virtual Machine Security Guidelines Version 1.0*, *The Center for Internet Security*, September [online] http://benchmarks.cisecurity.org/tools2/vm/CIS_VM_Benchmark_v1.0.pdf (accessed 26 May 2015).

Leja, C., Richard, C.C.P., Barnier, C., Brown, C.L., Dittmann, P.F., Koziel, P., Welle, M. and Westermeier, J.T. (2008) 'Virtualization and its benefits', *AITP – Research and Strategy Advisory Group*.

Miao, X. and Han, J. (2011) 'The design of a private cloud infrastructure based on XEN', in *Proceedings of Distributed Computing and Applications to Business*, *Engineering and Science* (*DCABES*), 14–17 October, pp.160–164.

Montenegro, G., Kushalnagar, N., Hui, J. and Culler, D. (2007) *Transmission of IPv6 Packets over IEEE 802.15.4 Networks*, RFC 4944, September.

Mukherjee, A., Paul, H.S., Dey, S. and Banerjee, A. (2014) 'ANGELS for distributed analytics in IOT', *IEEE World forum on Internet of Things* (*WF-IOT*), pp.565–570.

Phelan, T. (2008) *Datagram Transport Layer Security* (*DTLS*) *over the Datagram Congestion Control Protocol* (*DCCP*), RFC 5238, May.

Reiser, H.P. (2009) *Security Challenges with Virtualization*, Master's thesis, Universidade de Lisboa, December.

Shelby, Z., Hartke, K., Borman, C. and Frank, B. (2013) *Constrained Application Protocol* (*CoAP*), Draft-ietf-core-coap-04 (Internet Draft), January.

Valerio, P. (2016) 'Is the IoT a tech bubble for cities? With more cities joining the smart city revolution and investing in sensors and other IoT devices, the risk of a new tech bubble is rising', *IEEE Consumer Electronics Magazine*, Vol. 5, No. 1, pp.61–62.

Varadarajan, P. and Crosby, G.V. (2014) 'Implementing IPsec in wireless sensor networks', *New Technologies*, *Mobility and Security* (*NTMS*), *2014 6th International Conference*, IEEE, 30 March–2 April, pp.1–5.

Why XEN & XEN How it Work [online] http://XEN.org/files/Marketing/WhyXEN.pdf and http://www.XEN.org/files/Marketing/HowDoesXENWork.pdf (accessed 18 May 2015).

Wireshark (2010) August [online] http://openmaniak.com/wireshark.php (accessed 18 April 2015).

Wu, H., Ding, Y., Winer, C. and Yao, L. (2010) 'Network security for virtual machine cloud computing', in *Computer Sciences and Convergence Information Technology* (*ICCIT*), pp.18–21, DOI: 10.1109/ICCIT.2010.5711022.

Zanella, A., Bui, N., Castellani, A., Vangelista, L. and Zorzi, M. (2014) 'Internet of things for smart cities', *IEEE Internet of things Journal*, Vol. 1, No. 1, pp.22–32.