Cloudlet Mesh for Securing Mobile Clouds from Intrusions and Network Attacks*

Yue Shi, Sampatoor Abhilash, and **Kai Hwang**
University of Southern California
Los Angeles, CA. 90089 USA
{yueshi, sabhilas, kaihwang}@usc.edu

***Abstract*—This paper presents a new cloudlet mesh architecture for security enforcement to establish trusted mobile cloud computing. The cloudlet mesh is WiFi- or mobile-connected to the Internet. This security framework establishes a cybertrust shield to fight against intrusions to distance clouds, prevent spam/virus/worm attacks on mobile cloud resources, and stop unauthorized access of shared datasets in offloading the cloud. We have specified a sequence of authentication, authorization, and encryption protocols for securing communications among mobile devices, cloudlet servers, and distance clouds. Some analytical and experimental results prove the effectiveness of this new security infrastructure to safeguard mobile cloud services.**

Keywords — Mobile cloud, cloudlet mesh, inter-cloud protocol, collaborative intrusion detection, cloud mashup, and ***MapReduce spam filtering.***

I. INTRODUCTION

Network attacks are a serious matter that confront both cloud providers and massive number of mobile users who access distance clouds in our daily-life operations. Unauthorized intrusions or networks attacks have endangered cloud applications, such as threats from hidden viruses, embedded spams and worm outbreaks. In this paper we propose a novel approach to constructing a Wifi-enabled mesh of cloudlets to shield and safeguard cloud accesses by pervasive mobile users. [11, 18, 20].

Mobile cloud computing becomes an emerging field with high expectation by massive users [17, 24]. We aim to support mobile devices (smartphones, tablets. Etc.) to access cloud services via WiFi or mobile networks. Cloudlets have been proposed as wireless gateways to access remote clouds [20]. Cloudlets and WiFi access points (wireless routers) are integrated to form WiFi-enabled cloudlets.

The cloudlet mesh idea was originally proposed by Khan, et al [11] for real-time applications. We extend their work to support security functionalities in offloading the distance clouds. We secure with security mechanisms and special communication protocols. The purpose is to cope with attacks by viruses, worms and malware.

* Manuscript accepted to appear ***The Third IEEE International Conference on Mobile Cloud Computing, Services and Engineering, (Mobile Cloud 2015),*** March 30- April 3, 2015, San Francisco, CA. USA. The corresponding author is Yue Shi.

For data-intensive applications, the speed and efficiency demands are satisfied by offloading heavy workloads to the remote cloud. Mobile security includes data loss during transmission, storage, and processing stages over mobile devices and remote clouds. In the past, CloudAV [19] was suggested to use virtualized container of sensitive files in mobile devices. Data coloring was suggested to secure big data over the encryption solutions [9]. To protect mobile devices from spam/virus attacks, we suggest to use the remote cloud for data intensive filtering and updating the attack signature databases.

Mobile devices submit their cloud access requests through the cloudlet mesh. Our cloud-based security system works as an intelligent firewall or *intrusion detection system* (IDS) to secure mobile devices within the range of the underlying WiFi mesh. This approach extends from previous approaches [11, 22, 23]. We aim to improve in the following aspects:

- We propose a hierarchically structured security architecture. A trust chain is established between mobile devices, the cloudlet mesh, and remote cloud platforms.
- Predictive security analytics are processed at the backend cloud for virus signature scanning and update with automated virus/spam filtering and removal.
- We emphasize real-time filtering or removal of malicious attacks or fast response to intrusions with the help of trusted remote clouds.

In the past, mobile cloud security was studied by many authors [2, 6, 13, 19, 22]. Our work focuses on the use of cloudlet mesh to establish a cloud access defense shield. Our previous work on IDS [9, 17, 23], worm containment [3], DDoS defense [4], cloud data coloring [8], and trust management with reputation systems [25] are useful to reinforce the mobile security. Other work on cloud data storage protocols [12, 24] are also related our work.

The rest of the paper is organized as follows: Section II introduces the cloudlet mesh architecture. Section III suggests the countermeasures against attacks and outline the protocols developed. In section IV, we present some ideas and analytical results on collective IDS on the mesh. Section V presents the idea of offloading heavy spam filtering tasks to the cloud using the MapReduce filtering approach. Experimental results collected from EC2 cloud are reported. Finally, we summarize the contributions and suggest further work.

978-1-4799-8977-5/15 $31.00 © 2015 IEEE
DOI 10.1109/MobileCloud.2015.15

II. SECURE MOBILE CLOUD ENVIRONMENT

In this section, we introduce the cloudlet mesh architecture. Security threats to mobile cloud computing are assessed. Then we make plausible suggestions for the desired defense infrastructure in a mobile cloud computing environment.

A. The Cloudlet Mesh Architecture

The architecture of a *cloudlet mesh* is shown in Fig.1. All cloudlets are Wifi-enabled. Each cloudlet server has an embedded WiFi access point. Therefore, each cloudlet could connect to many mobile devices within the WiFi range. The cloudlets are interconnected by either wired or wireless links to form the mesh. All cloudlets operate essentially as gateways at the edge network of the Internet. Remote clouds are assumed accessible through the Internet backbone.

In Table 1, we summarize potential threats or attacks in 6 categories at the left column. The column headings correpond to each functional group. Per each threat category, we make suggestions to leverage some existing defense tools. We explore ways to cope with the network threats on the mobile devices, cloudlets and remote clouds. Cloudlets are shown powerful to support data caching, encryption, authentication, distributed intrusion detection and large-scale spam/virus filtering operations.

Mobile devices are subject to virus or network worm attacks. Encryption may not be the best solution for mobile devices due to their limited computing power and energy consumption constraints. Some special software tools are available to resist virus or worm attacks on mobile devices. Authentication and URL checking and spam filtering are needed on mobile devices. With large storage and backup services, we suggest offload the data file to the cloud as most smartphone users choose to do.

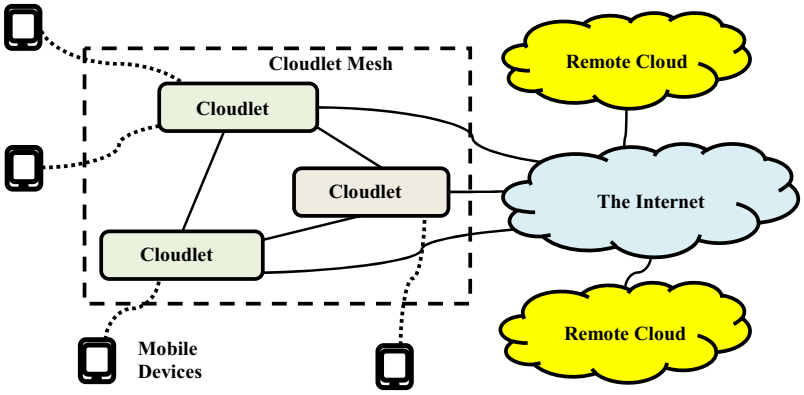


Figure 1: Cloudlet mesh architecture for secured mobile cloud computing, where the cloudlets are WiFi-enabled.

Table 1: Threats and Defense Strategies in Protecting Mobile Cloud Computing

Threats/Defense	Mobile Device	Cloudlet	Public Clouds
Encryption for data protection	Energy cost for encryption is high on mobile devices	Encryption supported and used for communication with remote cloud.	Encryption fully supported. to protect user data lost
Virus, Worms, or Malware attacks	Privacy and energy cost for detection of malware on mobile device is high.	Protect mobile device by verifying files and content used by the user.	Perform analytics on the cloud to detect new types of malware.
Identity theft and Authentication	User authentication before offloading compute or storage tasks.	Need to authenticate all three parties involved	Authentication as a Service. (AaaS) is needed
Cloud offloading and File Transfer	Mobile device migrates part of its task execution to the cloudlet	Data caching at cloudlet to improve performance at mobile devices	High latency to offload to remote cloud may create a QoS problem
Data integrity and storage protection	May use secure storage outsourcing protocols to solve the problem	Data stored by the cloudlet is vulnerable to attacks.	Clouds may compromise user data through phishing attacks.
URL and IP and Spam Filtering	Checking blacklist of IP addresses and URLs consumes much power on mobile devices	Cloudlet mesh performs quick search with low latency and alert the mobile devices with attacks..	Performs predictive analytics and provides database updates for the cloudlets.

We suggest to use multiple cloudlets in the mesh for three purposes: (1) Widening the wireless coverage range to serve many more mobile devices. (2) Coordinated defense among the mesh cloudlets make it possible to build a shield and alert system for all mobile users. (3) Caching and load balancing among the cloudlets in offloading tasks for the cloud.

The remote cloud essentially provide *Security as a Service* (SECaaS) to all end users. The cloud has the data-mining power to provide security intelligence and analytics tools. The overall security of the mobile environment could be improved with some existing cloud services such as notification service, scalable storage, elastic MapReduce in the AWS clo*ud.*

B. Cloud-Based Security Defense

Some sophysticated threats require to deploy the right countermeasures to establish the trust or assure the security. One central idea of this paper is to offload some security datamining and analytics tasks to the remote clouds ov er the cloudlet mesh. The cloud has the capacity to scan large signature databases. One can use MapReduce or Hadoop tools to support fast virus scanning or launch worm containment operations [4]. These tasks are data-intensive, which cannot be handled in real time by the small mobile devices.

One can consider deploy a hybrid intrusion detection systems (HIDS) [14] on the cloud. This HIDS combines both signature matching and anomaly detection into an integrated system for automated detection of virus/worm attacks and deter DDOS attacks over multiple network domains [7]. Or one can consider build reputation systems [25] to support accountability checking in case any trust violations are detected. Trust management demand heavy data mining and machine-learning analytics, which can be handled effectively on the clouds

C. Security Mechanisms and and Analytics

Figure 2 summarizes all the security mechanisms or software modules that are needed to secure a mobile cloud environment . The cloudlets provide mobility management to serve all mobile devices connected to the mesh. Three groups of functional modules are needed: namely *packet filtering, secure communication and cloud access control*.

We will further study these modules in subsequent sections. These modules are most implemented in software. Some may require special hardware, middleware or virtualization support. The cloudlets themselves are all *virtual machine* (VM) enabled. In fact, the cloudlets shake hands with the end users through a VM integration protocol. [20].

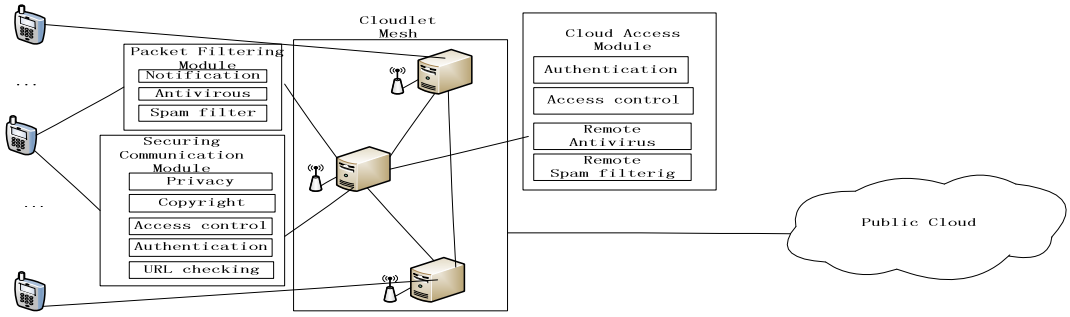


Figure 2: Security mechanisms or software tools needed to secure communication and data transfer among mobile devices, cloudlets and the distance clouds. Some protection features are also characterized in Table 1.

III. SECURITY PROTOCOLS FOR MOBILE CLOUDS

Three security protocols are specified below. These protocols are applied to secure the mobile devices on the outskirt and the cloud in distance. The protocols are illustrated in Fig.3 in several hand shaking processes.

A. Multi-party Authentication Protocol (MAP)

This MAP protocol is extended from the work of Ateniese, et Al [1]. The purpose to authenticate between mobile device and the cloudlet mesh. Three functional modules introduced in Fig.2 are used here. They are mainly used for two purposes:

(a). Mobility Management: When a mobile device submits an access request to the cloudlet mesh, the cloudlet responding with the highest signal strength will be connected. One can also connects to a nearby cloudlet with lower managing workload to act as the access point. When mobile device moves within the mesh range, handover between cloudlets is expected.

(b) Multi-Way Authentication: The incoming message, if it is sufficiently small in size, will be scanned by antivirus and spam filtering package in the receiving cloudlet. However for larger file exceeding certain limit, the filtering tasks will be offloaded to a distance cloud. Finally, when spam or virus removal is done, a notification will be returned to the requesting mobile device. Multi-party authentication is applied in the cloudlet mesh authentication which enables single sign-on within the cloudlet mesh.

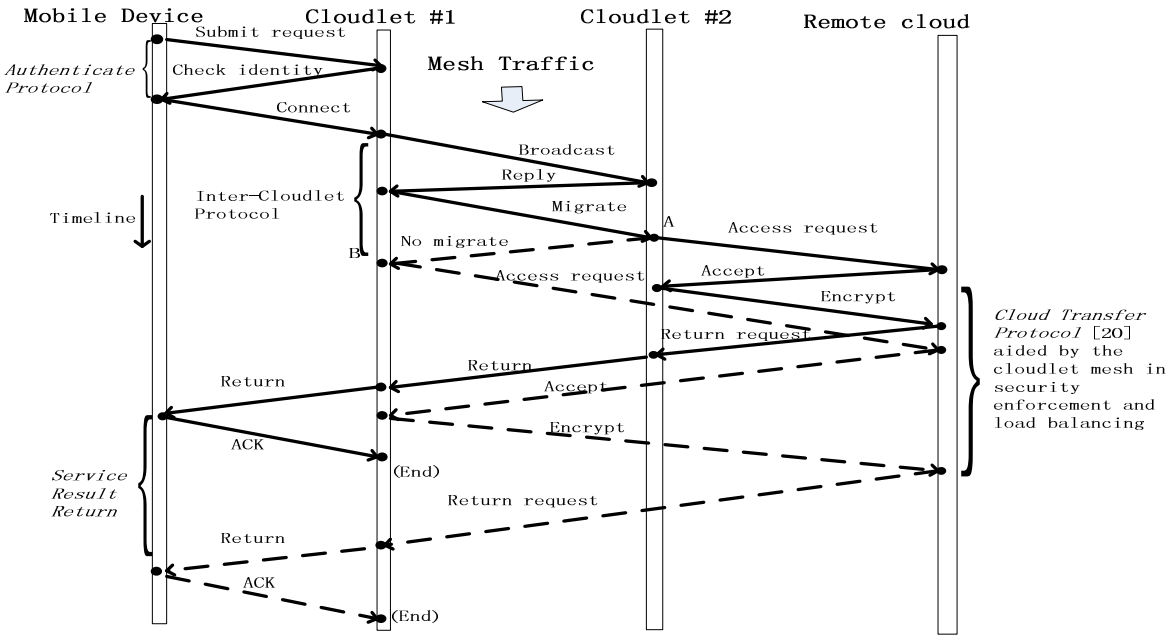


Figure 3 Security protocols applied for interactions among mobile devices, cloudlets and remote clouds.

B. Inter-Cloudlet Protocol: (ICP)

In Fig.3, we define a new *Inter-Cloudlet Protocol (ICP)* for communication among the cloudlets in the mesh. This ICP protocol supports collective intrusion detection and load balancing operations, which are transparent to mobile users. We use multiple cloudlets, each installed with an *intrusion detection system* (IDS). Each cloudlet has a database containing a white list of friendly users. This protocol specifies the steps needed to secure inter-cloudlet communications and load balancing within the cloudlet mesh.

Step 1: Mobile device sends a request to establish connection with a cloudlet after passing the MAP protocol.

Step 2: The cloudlet checks its white list of friendly users, if found, the connection is granted. Otherwise, the cloudlet broadcasts the request to other cloudlets in the mesh to check whether it is registered there.

Step 3: The cloudlet checks neighbour's workload. The mobile device could be migrated to a less loaded cloudlet for load balancing. This cloudlet will handle the access request to the remote cloud as noted by the time instance "A" in Fig.3)

***Step 3'*:** For data intensive applications, the original cloudlet will initiate the access request to the remote cloud as noted at time instance at time instance "B" in Fig.3.

C. Trusted Cloud Transfer Protocol (TCTP):

This protocol was suggested by Slawik [21]. Once a request is accepted, the cloudlet applies the TCTP to encrypt the message or file being transferred to a distance cloud. Note that in Fig.3 either cloudlet #1 or cloudlet #2 could handle this encryption process as shown by the solid and dashed edges starting from point A or point B to the end of the process, respectively. The following steps are taken in the TCTP handshaking process.

Step 1: A *certificate authority* (CA) sends a public key to the requesting cloudlet on behalf of the mobile user.

Step 2: The text body being transferred to the cloud is encrypted. The cloudlet scans every arriving packet for virus detection or spam filtering. Once passing the screening, the packet will be sent to the mobile device. Otherwise, an alert will be issued.

Step 3: The cloudlet waits for the response from the mobile device, if time out, the packet will be resent. After receiving the security report from the remote cloud, the cloudlet notifies the mobile device accordingly.

IV. DISTRIBUTED IDS ON THE CLOUDLET MESH

In this section, we use ROC (*receive operating characteristic*) curves to analyze the performance of the proposed cloudlet mesh for securing mobile cloud applications. We use a collaborative IDS approach to achieving high detection rate.

A. Improvement in IDS Rate with The Mesh

For Intrusion detection systems, false negatives have higher penalty to the system performance than false positives, because an undetected attacker can compromise the entire system once they are able to clear the detection result. Hence to

improve the overall system security it is better to have a multitude of IDS systems executable at individual cloudlets.

The intrusion detection is done by all cloudlets collectively. They alert each other with the detection results. This does not cause a performance penalty due to cooperative efforts by all cloudlets involved. Each IDS can be in one of two states: either an intrusion present (I) or no intrusion present (NI). The IDS reports either an intrusion alarm (A) or no alarm (NA). The result of the ROC curve displays the true positive rate or the probability of an alarm given a false positive rate.

The detection probability P(A|I)=*1– β* corresponds to the case of *true positive rate* for an alarm given an intrusion. The *false positive rate* is denoted as P(A|NI) = *α.* On the ROC curve, the α is plotted on the x-axis and the true positive rate 1-β is plotted at the Y-axis. We derive a formula of a composite ROC curve for a mesh of *n* cloudlets. We aim to detect any intrusion with accuracy.

The following Equations cover all intrusion detection conditions that a cloudlet supposes to handle with confidence.

$$P(A1,A2,\ldots,AN \mid I) = \prod_{i=\{1,..n\}} (1 - \beta_i)$$

$$P(A1,A2,\ldots, NAj,\ldots,AN \mid I) = \prod_{(i=\{1,..n\},j=\{1,..,n\},i\neq j)} (1-\beta_i) \times \beta_j$$

$$P(NA1,NA2,\ldots,NAN \mid I) = \prod_{i=\{1,..n\}} \beta_i \quad (1)$$

$$P(A1,A2,\ldots,AN \mid NI) = \prod_{i=\{1,..n\}} \alpha_i$$

$$P(A1,A2,\ldots, NAj,\ldots,AN \mid NI) = \prod_{(i=\{1,..n\},j=\{1,..,n\},i\neq j)} (1 - \alpha_i) \times \alpha_j$$

$$P(NA1,NA2,\ldots,NAN \mid NI) = \prod_{i=\{1,..n\}} (1 - \alpha_i)$$

Since we report an intrusion with at least one IDS reporting correctly. Thus β for the entire system is given by Eq. (2). The α is calculated by Eq.(3) as follows:

$$\beta = P(NA1,NA2,\ldots,NAN \mid I) = \prod_{i=\{1,..n\}} \beta_i \quad (2)$$

$$\alpha = 1-\prod_{i=\{1,..n\}} (1 - \alpha_i) \quad (3)$$

B. Optimization of the Collaborative IDS

We model the arrival of traffic to each individual cloudlet is self-similar and maximize an objective function for calculating the improvement in IDS performance gained by using N cloudlets in the cloudlet mesh.

Let node 0 be the master cloudlet. Led d_i represent the delay of the cloudlet mesh from node 0 to each individual cloudlet node i in the mesh. Lett P_i be the expected processing time at node *i* for the IDS database search. If a cloudlet node *i* is itself loaded, it will not be able to support the computation of the MapReduce job.

Let L_i be the load at node *i.* If a cloudlet presents itself as busy, L_i for that node will be 0, and the value of L_i will rise according to the load on the cloudlet, up to a value of 1 which indicates that the cloudlet is free, as communicated in its heartbeat message. Let F_i be fraction of the job the master cloudlet decides to distribute to the cloudlet node *i* based on the decision made by the master cloudlet on the delay and the loading of cloudlet.

Also the individual node delay and processing times have to be under a network-wide delay *D* for the network so that queueing does not build up for the packets. For *i=0,1,..,N-1,* we have $0 <= L_i <= 1$; $d_{min} <= d_i <= D$ $P_{min} <= P_i <= P_{max}$.. We formulate a linear programming model as follows:

$$\text{Maximize: } \sum_{i=\{0,\ldots,N-1\}} (F_i * (1 - L_i) / (P_i * d_i) \quad (4)$$

Subject to the following constraints:

$$\sum_{i=\{0,\ldots,N-1\}} F_i = 1; \quad (5)$$

$$L_i + F_i <= 1 \quad (6)$$

C. ROC Results on Detection Accuracy

The ROC results of 3 independent IDSs are given in Table 3 and the ROC result for a mesh of 3 cloudlets are given in Table 4. These results are plotted in Fig.4. The false positive rate α was measured in the range {0.001, 0.006} on the X-axis. The true positive rate ranges between 0.45 to 0.91 along the Y-axis. The individual IDS results are shown by dash or dot lines at the bottom of the plot. The composite ROC curve is at the top of the plot.

Table 2: IDS Characteristics for 3 Separate Cloudlets

Cloudlet #1		*Cloudlet #2*		*Cloudlet #3*		*Cloudlet Mesh*	
α_1	$1-\beta_1$	α_2	$1-\beta_2$	α_3	$1-\beta_3$	α	$1-\beta$
0.0020	0.530	0.0023	0.600	0.0020	0.530	0.0063	0.9099
0.0018	0.515	0.0020	0.594	0.0018	0.515	0.0055	0.9031
0.0016	0.504	0.0018	0.59	0.0016	0.504	0.0051	0.8985
0.0015	0.497	0.0018	0.588	0.0015	0.497	0.0048	0.8951
0.0014	0.49	0.0017	0.581	0.0014	0.490	0.0045	0.8908
0.0012	0.482	0.0016	0.576	0.0012	0.482	0.0042	0.8868
0.0011	0.474	0.0014	0.572	0.0011	0.474	0.0038	0.8824
0.0011	0.467	0.0012	0.568	0.0011	0.467	0.0035	0.8786
0.0011	0.46	0.0010	0.561	0.0011	0.460	0.0031	0.8736

From Fig.4, we see a clear advantage of using the composite mesh of 3 cloudlets over the use of individual cloudlet for intrusion detection. With a small false positive rate, say 0.001, the composite ROC curve shows 35 ~ 64% higher true positive rate than that of individual IDS result.

When the false positive rate is high at 0.006 at the extreme right, the performance of the composite IDS is still 10 ~ 20% higher than the individual IDS performance. These results prove the superiority of using the cloudlet mesh for security enforcement in a mobile cloud environment. For other collective defense schemes against viruses and network worms, or DDoS attacks, we expect similar advantage points in using the cloudlet mesh.

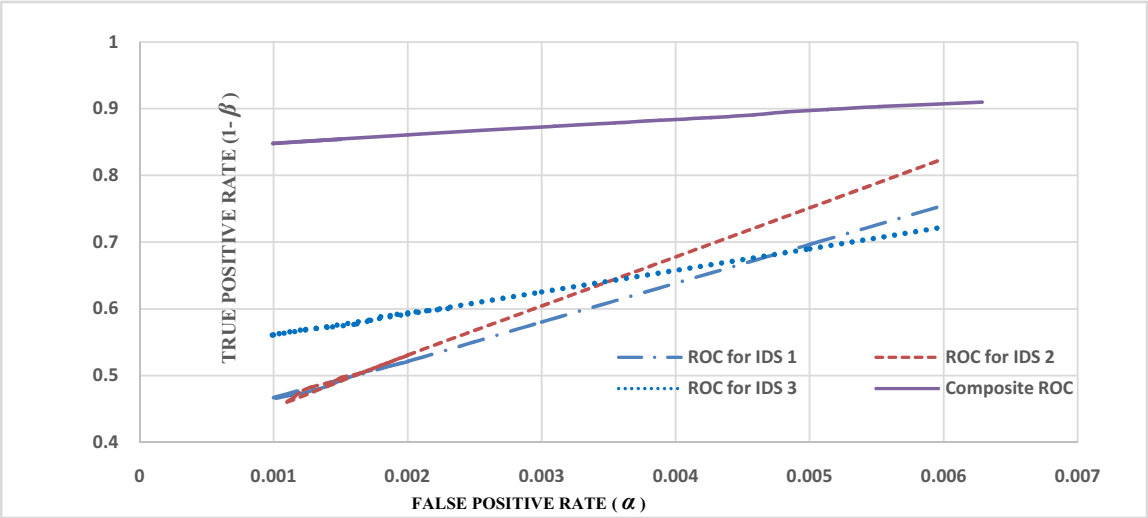


Figure 4: ROC curves showing the improvement of IDS rate in a mobile cloud platform secured by the cloudlet mesh

V. SPAM/VIRUS FILTERING ON MOBILE CLOUD

We present an offloading operation on AWS EC2 cloud for fast filtering of spams from massive number of Twitter blogs. Consider the interaction between two cloud platforms: Twitter and Amazon EC2 to carry out the task of fast spam filtering.

This offers a typical mashup service over social media data. The Twitter social cloud is mainly used for tweet collection and streaming into the AWS S3 (*Simple Storage Service*). The EC2 is used to implement the MapReduce analytics engine for filtering out detected spams on the fly. With twitter REST API, we gather the blogs related to trending topics from different users.

A. MapReduced Spame Filtering

The idea of using MaReduce to do spam filtering is illustrated in Fig.5. On the left initial checking of URL links in the cloudlet mesh. If the file size is larger than 1GB or the cloudlet mesh cannot identify whether or not it is a spam file, that file will be sent to remote cloud for further analysis. To improve performance, MapReduce filtering is applied in a remote cloud.

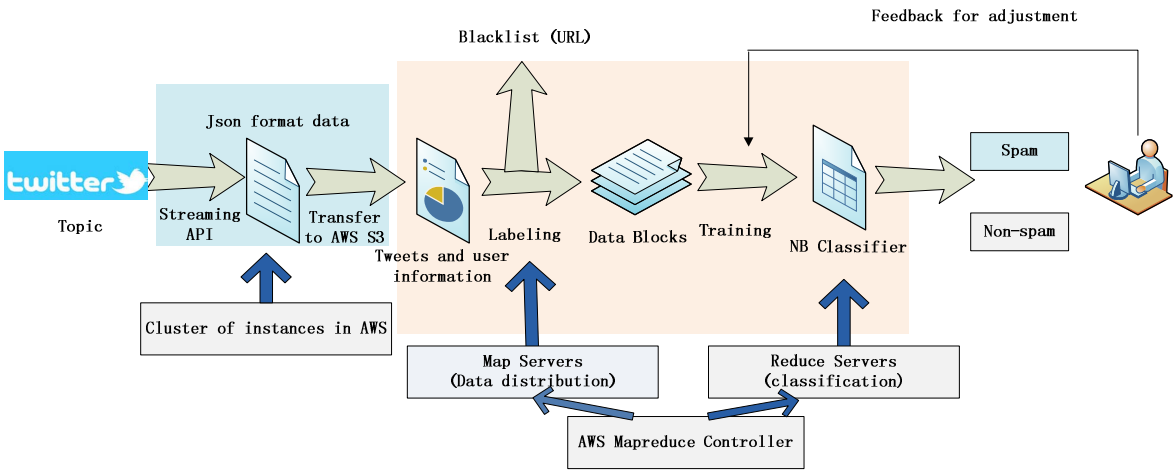


Figure 5 MapReduce spam filtering on remote cloud for securing mobile devices and the cloudlet mesh.

After spam filtering at a remote cloud, a report is sent back to the mobile device through the forwarding cloudlet. The cloudlet mesh notifies the mobile device to make a decision whether or not to receive the suspicious packet. In the remote cloud, we apply Map function to map out the input file to different machine instances, and the Reduce function is applied to use the Naive Bayesian classifier [14, 15].

We use both link screening against a blacklist and a content-based filtering algorithm. If a tweet URL matches with

an entry in the blacklist, we classify it directly as a spam tweet. Otherwise, we will go through a content-based filtering process shown in the middle part of Fig.5.

Algorithm 1: Mapping partitioned data blocks

Input: Twitter file to be scanned for filtering out Spams

Output: (Key, Value) pairs for data partitioning

// Keys are the tweet block numbers and Values \ are the features extracted from all tweets. //

Procedure:

Int n= number of tweet blocks

Int M= number of available Map servers

Forall Data blocks from Amazon S3 storage

Distribute the n data blocks evenly to all Map servers

Int fnb=sequence number of the block file;

Int count= sequence number of tweet in block

Block_number=(count) Mod n;

EndForall

Forall M Map Severs

For each tweet in each Map Serve:

Extract the following features:

String tct= content of the tweet;

Int ufd = user ID number

Int ufr= user number of followers;

String_uvs= user verified status;

Double_ur= user reputation

Int th= number of hashtags in the tweet;

Int nl= number of links in the tweet

EndExtract

Key=(string) Block_number;

Value=fnb+count+tct+ufd+ufr+uvs+ur+th+nl;

Emit (Key, Value) pairs

EndFor

EndForall

B. The Naïve Bayesian Spam Classifer

To classify the users, we apply the following profile information on users: user account reputation, spam URLs, number of hashtags (#), spam terms, and the total number of mentioned users tested. Note that a user can be either a legitimate user or a spammer, but not both. The account reputation is represented by:.

$$P(\varphi) = \frac{n_i(j)}{n_i(j)+n_o(j)} \quad (7)$$

where $n_i(j)$ is the number of followers user j has and $n_o(j)$ is the number of proven friends user j has. The spam terms are high-occurrence words found in known spammers' tweets. The URL begins with http://. The Tweets users will be identified with @ followed by the user account name.

The NB classifier is used to classify Tweet x as a spam c_s, when the following Bayesian condition is met:.

$$P(c_s)\times P(x|c_s) / [\, P(c_s) \times P(x|c_s) + P(c_l) \times P(x|c_l)] > \theta \quad (8)$$

where $P(c_s)$ is the spam probability, $P(c_l)$ is the legitimate user probability obtained at the training stage, and the threshold θ are aprior information known to the classifier.

The conditional probabilities $P(x|c_s)$ and $P(x|c_l)$ are derived from spam terms collected in a training set. Let $P(word_i|c_s)$ be the probability spams containing word i and $P(word_i|c_l)$ be that of legitimate blogs. We have

$$P(x|\, c_s) = \prod_i P(word_i|\, c_s), \quad P(x|\, c_l) = \prod_i P(word_i|\, c_l) \quad (9)$$

Algorithm 2 : Reduction for NB Spam Classifer

Input: Grouped (key,value) from the mapper

Output: Classified (Key, Value) pairs

Procedure:

Int N= number of available Reduce servers

// (Key,Value) pairs with the same key go to the same Reduce server//

Forall N Reduce Servers:

For each value in each Reduce server:

Get features: **s**tring fnb, tct, count,ufd, ufr, uvs, ur,th, nl;

// These are defined in Map Algorithm 1//

Classify using the the NB condition in Eq.(8).

Key=fnb+count;

// Key is tweet sequence number//

Value= tct;

Output (Tweet key, Legitimate tweet) or (Tweet Key, Spam);

End for

EndForall

The spam classifier must be trained to give higher accuracy in the detection process. The detection threshold θ in Eq.(5) is determined by the training set applied. A well prepared training set gives better accuracy in the detection process. Typically, one set $\theta = 0.5$ in the middle initially. We use adaptive machine learning method to adjust the threshold based on real-time feedbacks from historical filtering results.

In Fig.5, the feedback path is used to adjust the training sets or thresholds applied. The following Algorithm adjusts the threshold in the feedback loop. During period T, an offset window $\{\lambda_1, \lambda_2\}$ is applied. If the spam report is greater than a threshold λ_2, we decrease θ by a small value σ. If the spam

report is less than a threshold λ_1, we increase θ by σ Here λ_1, λ_2 and σ are preset by training experience.

Algorithm 3: Training NB classifier threshold

Input: Spam feedback reports and old threshold θ value
Output: New threshold newθ value
Procedure:

For each time period T and spam feedback count ms
Int ms=0;
//Initialize the spam feedback count during T//
Receive a spam count report:
ms = ms+1;
// Let β be the number of tweets processed during T//
// $\lambda_1 < \lambda_2$ are the offset window for threshold adjustment//
Endfor
If ms / β > λ_2, newθ = θ - σ
If ms / β < λ_1, newθ = θ + σ
Endif
Endif
Output new threshold value newθ

C. Spam Filtering Speed Gains

In Fig.6, we report spam filtering time in using the MapReduce pipeline illustrated in Fig.5. Three Tweet datasets are tested, ranging from 10 GB to 100 GB and 1 TB are tested.

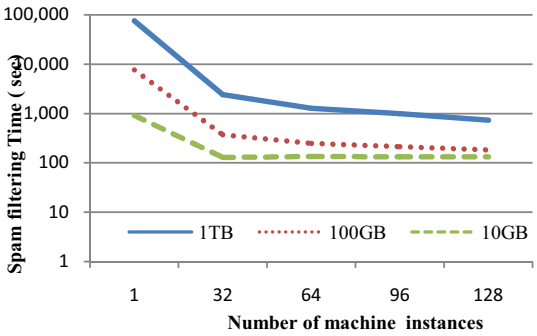


Figure 6 Spam filtering time on AWS MapReduce cluster with increasing number of machine instances

For 1 TB dataset, the filtering time is reduced from 27 hours (100,000 sec) to 15 minutes (900 sec), when the EC2 cluster scales from 1 to 128 nodes. A speedup of 108 is experienced. For 100 GB dataset, we have the execution time of 127 min on a single node and 5 minutes on 128 nodes. A speedup factor of 127/5 ≈ 25. The conclusion is that MapReduce can reduce the filtering time significantly. The larger is the dataset, the higher the speedup is expected.

D. Spam Detection Results on EC2

Consider the set ***L*** of legitimate tweets (blogs or Emails in general) mingled with the set ***S*** of malicious spammers (or attackers in general). In general, there are four possible prediction outcomes (subsets) as defined below. The *true positives* (TP) correspond to the set of spams correctly classified. The *true negatives* (TN) refer to the set of legitimate blogs correctly classified. The *false positives* (FP) are the set of legitimate Tweets incorrectly classified as spams. Finally, *false negatives* (FN) are the spam tweets incorrectly classified as legitimate users.

Note that the set union relationship: ***L*** U ***S*** = TP U TF U FP U FN holds. For simplicity, we use the notations ***a, b, c*** and ***d*** to denote the probability rates of the corresponding events TP, TN, FP and FN, respectively. Sometimes, researchers refer this 4-tuple as a *confusion matrix*. Five *performance metrics* are defined in Table 3 in terms of those parameters.

The *accuracy* is the percentage of all messages (Tweets) and spams classified correctly to all messages and spams scanned. The *precision* is the ratio of correctly classified spams to the total number of predicted spams, including both legitimate Tweets and spams that are classified as spams. Accuracy and precision reflect different concerns by the performance evaluators or system supervisors.

Table 3. Performance Metrics for Spam Filtering

Performance Metric	Probability Expression
Accuracy (*A*)	$A = (a+b)/(a+b+c+d)$
Precision (*P*)	$P = a/(a+c)$
Recall (*R*)	$R = a/(a+b)$
Spam Detection Rate (*S*)	$S = a/(a+d)$
Legitimate Blog Rate (*T*)	$L = b/(b+c)$

On the other hand, we have the *spam recall rate* that reflects the ratio of correctly classified spams to the total number of correctly classified legitimate tweets and spams. The recall rate indicates how often the spams appeared in correctly predicted cases. Finally, we define the *spam detection rate S* by the ratio of positively detected spams over all suspected spams, detected correctly or incorrectly. One can also define a *Legitimate detection rate L* by the ratio of TN to all legitimate tweets correctly or incorrectly detected.

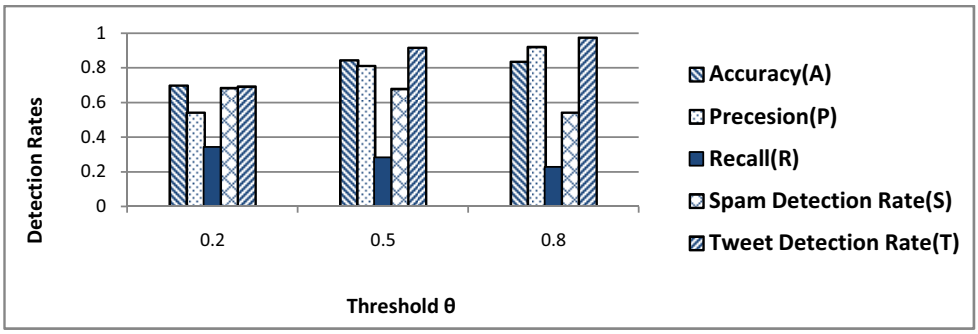


Figure 7: Spam filtering results from MapReduce experiments on the EC2 of AWS cloud over 1 TB of Twitter blogs.

VI. CONCLSIONS AND SUGGESTIONS

Securing mobile cloud services is the major barrier to the integration of BTOD (*bring your own devices*) and BYOC (*bring your own cloud*) in our daily applications. This paper attempted to solve the problem with the use of a mesh of cloudlets to perform the needed authentication, authorization and encryption operations to establish a trusted mobile computing environment.

We use the cloudlet mesh to perform collaborative intrusion detection among multiple cloudlets. This results in significant gain in collective intrusion detection rate. We offload very large Tweet dataset (up to 1 TB) to the AWS cloud for MapReduced filtering of spams hidden in large tweet dataset. Fast filtering of spams from huge dataset can help virus scanning and signature generations from new unknown attacks. We summarize below the major research findings of these studies.

A. Summary of Major Research Findings:

Our contributions are summarized below in 4 technical aspects towards mobile cloud security.

(1) The cloudlet mesh can be applied in many security functions, such as multi-party authentication, anti-versus scanning, distributed intrusion detection, and authorization to access the remote clouds, effectively. This relieves the mobile devices from time/energy consuming security operations.

(2) The collective use the cloudlet mesh can greatly enhance the intrusion detection rate (see Fig.4) by an improvement factor of 36% when the false positive rate is 0.5%. This is very encouraging to see the collective gain using multiple cloudlets for security enforcement.

(3) Another distinct advantage is the introduction of the *inter-cloudlet protocol (ICP)* to support distributed intrusion detection. Once the malicious intrusions are stopped, some potential virus-carry programs or data objects can be eliminated before they take effect. The ICP promotes load balancing among multiple cloudlets operating in the mesh concurrently.

(4) We applied EC2 cloud with MapReduced detection of spams over 1 TB of Twitter blogs. This results in 108 times reduction in filtering time and leads to 70~90% detection rate of the Tweet spams on the Amazon EC 2 cloud. These results clearly demonstrate the effectiveness of offloading heavy jobs to the distance cloud.

B. Suggestion of Further Work

In the past, our research group at USC has developed a number of security mechanisms, data privacy and trust management techniques for securing grid or P2P computing. We suggest below ways to extend these techniques to secure both mobile and non-mobile clouds.

(1) The *hybrid intrusion detection system* (HIDS) [9] can be extended to use the cloudlet mesh in detecting not only virus or worms but also network anomalies such as unusual TCP connections, etc.

(2) The WormShield containment scheme reported in [3] can be extended to cope with network worm spreading in large-scale clouds or data centers.

(3) The data coloring techniques [8] can be applied to protect large database in the cloud. Some trust management in clouds can be supported by the PowerTrust reputation system [25] in the cloud using P2P techniques.

(4) Additional experiments are desired to prove the advantages of offloading some of the big-data mining and analytics operations on the distance cloud. Latency analysis and QoS enhancement are desired in these mobile cloud applications.

ACKNOWLEDGMENTS: The research reported here was supported by Basic Research Program of China under 973 Grant No.2011CB302505. Kai Hwang would like to acknowledge the support of a visiting chair professorship endowed by EMC Cooperation at Tsinghua University.

REFERENCES :

[1] G. Ateniese, M. Steiner, and G. Tsudik ,"New Multiparty Authentication Services and Key Agreement Protocols," *IEEE Journal of Selected Areas in Communications,* April 2000.

[2] A. Bahar, Md. Ahsan Habib and Md. Manowarul Islam, "Security architecture for mobile cloud computing," *International Jounal of Scientific Knowledge,* pp. 11-17, July 2013.

[3] M. Cai, K. Hwang, Y. K. Kwok, S. Song, and Y. Chen, "Collaborative Internet Worm Containment", *IEEE Security and Privacy,* May/June 2005, pp.25-33.

[4] D. Chen, and Hong Zhao, "Data Security and Privacy Protection Issues in Cloud Computing," *IEEE International Conf. Computer Science and Electronics Engineering* (ICCSEE), pp. 647-651, March. 2012.

[5] Y. Chen, K. Hwang, and W. S. Ku, "Collaborative Detection of DDoS Attacks over Multiple Network Domains", *IEEE Trans. on Parallel and Distributed Systems* , Dec. 2007.

[6] Huang, D., Zhou, Z., Xu, L., Xing, T., Zhong, Y.: Secure data processing framework for mobile cloud computing. : *Proc. of the IEEE Conference on Computer Communications Workshop*, Shanghai, pp. 614–618, 2011.

[7] K. Hwang, J. Dongarra, and G. Fox, "Cloud Computing: Trust Managements in Virtual Datacenters", *Microsoft TechNet Magazine,* Dec. 2011.

[8] K. Hwang and D. Li, " Trusted Cloud Computing with Secure Resources and Data Coloring", *IEEE Internet Computing,* Vol.14, Sept. 2010.

[9] K. Hwang, M. Cai, Y. Chen, and M. Qin, "Hybrid Intrusion Detection with Weighted Signature Generation over Anomalous Internet Episodes", *IEEE Trans. on Dependable and Secure Computing,* Vol.4, No.1, Jan-March, 2007.

[10] K. A. Khan ; Q. Wang ; C. Luo ; X. Wang and C. Grecos, " Comparative study of internet cloud and cloudlet over wireless mesh networks for real-time applications ", *Proc. SPIE* 9139, *Real-Time Image and Video Processing* , May 15, 2014)

[11] D. Kovachev, R. Klamma, Beyond the client-server architectures: A survey of mobile cloud techniques, *Proc. of the 1st IEEE International Conference on Communications* (ICCC), Beijing, China, August 2012.

[12] Kumar, P.S., Subramanian, R., Selvam, D.T., "An Efficient Distributed Verification Protocol for Data Storage Security in Cloud Computing," *Proc. IEEE 2nd International Conf. Advanced Computing, Networking and Security* (ADCONS '13), pp. 214-219, Dec. 2013

[13] M. Lacoste, A. Wailly Aymeric, and T. Loïc, H. Xavier L. Guillou, and J.Wary "Flying Over Mobile Clouds with Security Planes: Select your Class of SLA for End-to-End Security," *IEEE/ACM 6th International Conference on Utility and Cloud Computing,* pp. 51-59, 2013

[14] K. Lee, J. Caverlee and S. Webb, "Uncovering Social Spammers: Social Honeypots + Machine Learning", Proc. of *Int'l Conf. on Research and Dev. in Information Retrieval,* ACM SIGIR-2010, N.Y., pp. 435 – 442.

[15] M. McCord and M. Chuah, "Spam Detection on Twitter using Traditional Classifiers", *ATC* , Banff, Canada, Sept. 2011

[16] Mechtri, M. , Zeghlache, D. ,Zekri, E. and Marshall, I.J., " Inter and intra Cloud Networking Gateway as a service," *IEEE Int'l Conf. Cloud Networking* (CloudNet '2013) pp. 156-163, 2013.

[17] Mohamed, H., Adil, L.,Saida, T., and Hicham, M, "A collaborative intrusion detection and Prevention System in Cloud Computing," *Proc. IEEE Conf. AFRICON,* Sept 2013.

[18] M. Reza Rahimi, Jian Ren, Chi Harold Liu, Athanasios V. Vasilakos, and Nalini Venkatasubramanian, "Mobile Cloud Computing: A Survey, State of Art and Future Directions", in *ACM/Springer Mobile Application and Networks* (MONET), Speciall Issue on Mobile Cloud Computing, Nov. 2013.

[19] J. Oberheide, K. Veeraraghavan, E. Cooke, J. Flinn, F. Jahanian, "Virtualized In-Cloud Security Service for Mobile Devices," *ACM Proceedings of the First Workshop on Virtualization in Mobile Computing,* pp 31-35, 2008

[20] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The Case for VM-Based Cloudlets in Mobile Computing," *IEEE Pervasive Computing,* vol. 8, no. 4, pp. 14–23, 2009.

[21] M. Slawik, "The Trusted Cloud Transfer Protocol," *Proc. IEEE 5th International Conf. Cloud Computing Techmology and Science* (CloudCom '13), pp. 203-208, 2013.

[22] H. Suo, Z. Liu, J. Wan, and K. Zhou, "Security and privacy in mobile cloud computing," in *Proceedings of the 9th International Wireless Communications and Mobile Computing Conference* (IWCMC '13), pp. 655–659

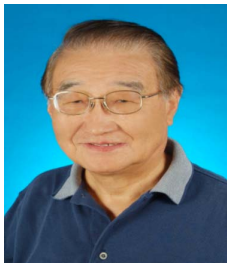
[23] J. W. Ulvila, J. E. Gaffney, "Evaluation of intrusion detection systems", *Journal of Research of the NIST,* 2003, pp. 453-473.

[24] K. Yang and X. Jia, "An Efficient and Secure Dynamic Auditing Protocol for Data Storage in Cloud Computing," *IEEE Trans. Parallel and Distributed Systems,* vol. 24, Issue. 9, pp. 1717-1726, Sep 2013..

[25] R. Zhou and K. Hwang, "PowerTrust: A Robust and Scalable Reputation System for Trusted P2P Computing", *IEEE Trans. on Parallel and Distributed Systems*, April, 2007.

Yue Shi is a PhD student of Electrical Engineering at the University of Southern California. She received the B.S from Huazhong Univ. of Science and Technology in China and M.S. degree at USC. She is interested in big data analytics, mobile and cloud systems, social networks and mashup applications. She can be reached by: yueshi@usc.edu.

Sampatoor Abhilash received the M.S degree in Electrical Engineering (Computer Networks) from the University of Southern California in 2015. His research interests include mobile cloud computing, distributed systems, wireless networks and queueing theory.



Kai Hwang is a Professor of EE/CS at the University of Southern California. He earned the Ph.D. degree from UC Berkeley in 1072. An IEEE Life Fellow, he has published extensively in the areas of computer architecture, parallel processing, network security, P2P/Grid and cloud computing. The *IEEE CloudCom* has extended him a Life-Time Achievement Award in 2012. Contact him via Email: kaihwang@usc.edu