امنیت و حریم خصوصی در رایانش ابری

چکیده

پیشرفت های اخیر موجب افزایش محبوبیت و موفقیت محاسبات ابری شده است. با این حال، هنگامی که برون سپاری داده ها و برنامه های تجاری به شخص ثالث باعث موضوعات امنیتی و مسائل خصوصی می شود, این مقوله به یک نگرانی حیاتی تبدیل می شود. در سراسر مطالعه پیش رو، نویسندگان, به دنبال هدف مشترک ارائه یک بررسی جامع از مسائل امنیت و حریم خصوصی موجود در محیط های ابری هستند. ما پنج صفات نماینده امنیت و حریم خصوصی (به عنوان مثال، محرمانه بودن، صداقت (انسجام)، در دسترس بودن، مسئولیت پذیری (پاسخگویی)، و حفاظت از حریم خصوصی) را شناسایی کرده ایم. با شروع از این ویژگی ها، ما روابط در میان آنها را ارائه می دهیم، آسیب پذیری هایی که ممکن است توسط مهاجمان، مدل های تهدید، و همچنین استراتژی های دفاعی موجود در یک سناریوی ابر مورد سوء استفاده قرار گیرد. جهت تحقیقات آینده برای هر یک از ویژگی ها از قبل تعیین شده است.

عبارات شاخص- محاسبات ابر، امنیت، حریم خصوصی، اعتماد، محرمانه بودن، صداقت، مسئولیت پذیری، در دسترس بودن.

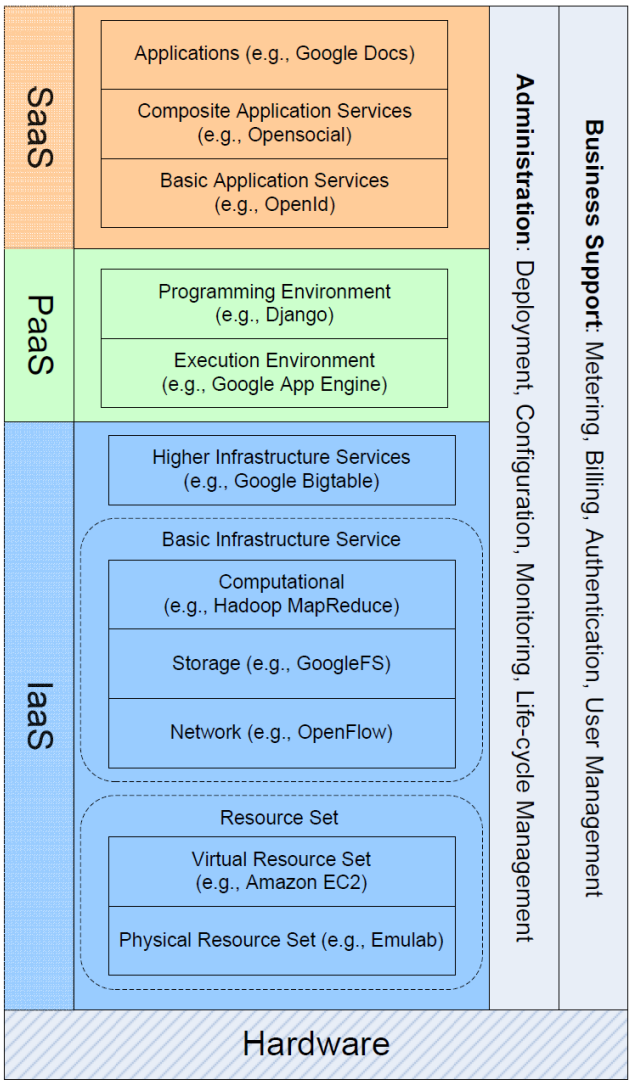
1. مقدمه

رایانش ابری به عنوان یک نقطه ارتباط در صنعت و دانشگاه ظهور یافته است؛ این مقوله نشان دهنده یک مدل کسب و کار جدید و الگوی محاسبات است که تأمین منابع محاسباتی و ذخیره سازی را بر حسب تقاضا میسر می سازد. منافع اقتصادی به منزله یک محرک اصلی برای محاسبات ابری هستند, با توجه به این واقعیت که محاسبات

ابری یک موثر راه برای کاهش هزینه های سرمایه (CAPEX) و هزینه های عملیاتی ار ارائه می دهد(OPEX) .
تعریف محاسبات ابری در بسیاری از متون ارائه شده است [1]، [2]، [3]، [10]، اما هیچ یک, به طور گسترده ای شناخته نشد. در سراسر این متن [1] استناد می کنیم که محاسبات ابری بدین صورت تعریف می شود: " الگوی محاسبات توزیع شده در مقیاس بزرگ که توسط اقتصادهای مقیاس راه اندازی می شود که در آن, حوزه ای از قدرت ، ذخیره سازی، سیستم عامل ها، و خدمات انتزاعی، مجازی، قابل مقیاس از نظر پویایی و مدیریت شده بر حسب تقاضا به مشتریان خارجی بر روی اینترنت تحویل داده می شوند".

A. معماری ابر

شکل. 1 معماری کلی از یک پلت فرم ابر را به تصویر می کشد که پشته ابر [61] نیز نامیده می شود. با ساختمانی بر اساس امکانات سخت افزاری (که معمولاً توسط مراکز داده مدرن پشتیبانی می شود)، خدمات ابر را می توان در اشکال مختلف از لایه پایین تا لایه بالا ارائه نمود. در پشته ابر، هر لایه یک مدل سرویس را نشان می دهد. زیرساخت به عنوان یک سرویس (IaaS) در لایه پایین ارائه می شود، که در آن منابع جمع می شوند و از لحاظ فیزیکی مدیریت می شوند (به عنوان مثال، Emulab) و یا عملاً (به عنوان مثال، ,(Amazon EC2)، و خدمات در شکل های ذخیره (به عنوان مثال، GoogleFS) شبکه (به عنوان مثال، OpenFlow) یا قابلیت محاسباتی (به عنوان مثال، Hadoop MapReduce) تحویل داده می شوند. لایه میانی, سیستم عامل به عنوان خدمت را تحویل می دهد (PaaS) ، که در آن خدمات به عنوان یک محیط برای برنامه نویسی (به عنوان مثال، Django) و یا اجرای نرم افزار (به عنوان مثال، Google App Engine) ارائه می شوند. نرم افزار به عنوان خدمات(SaaS) واقع در لایه بالاست که در آن یک ارائه دهنده ابر انعطاف پذیری مشتری را صرفاً با ارائه برنامه های کاربردی نرم افزار به عنوان یک سرویس بیشتر محدود می کند. به غیر از تأمین خدمات، ارائه دهنده ابر, یک مجموعه از ابزارها و امکانات مدیریت (به عنوان مثال، به عنوان مثال مدیریت خدمات لحظه ای چرخه عمر، اندازه گیری و صدور صورت حساب، پیکربندی پویا) را به منظور مدیریت یک سیستم ابری بزرگ حفظ می کند.



شکل 1. معماری رایانش ابری [61]

B. ویژگی های ابر و چالش های امنیتی

اتحادیه امنیت ابر پنج ویژگی ضروری [6] را خلاصه نموده است که نشان دهنده رابطه با و تفاوت از، الگوی محاسبات سنتی است.

• سلف سرویس بر اساس تقاضا - یک مشتری ممکن است به طور یک جانبه توانایی های محاسبات را به دست آورد، مانند استفاده از سرور های مختلف و ذخیره سازی شبکه، بر اساس تقاضا، بدون تعامل با ارائه دهنده ابر.

• دسترسی به شبکه گسترده - خدمات در سراسر اینترنت از طریق یک مکانیسم استاندارد تحویل داده می شوند که اجازه می دهد تا مشتریان از طریق ابزارهای ناهمگن نازک یا ضخیم کاربر (به عنوان مثال، رایانه های شخصی، تلفن همراه، و PDA ها) به خدمات دسترسی داشته باشند.

• ادغام منابع - ارائه دهنده ابر با ادغام منابع محاسباتی، یک مدل چند-مستاجر را برای خدمت به مشتریان متعدد به کار می گیرد که منابع فیزیکی و مجازی مختلفی هستند که به طور پویا با توجه به تقاضای مشتری اختصاص داده و یا منصوب می شوند. نمونه هایی از منابع شامل ذخیره سازی، پردازش، حافظه، پهنای باند شبکه، و دستگاه های مجازی می شوند.

• انعطاف پذیری سریع - قابلیت ها را می توان به سرعت و با انعطاف پذیری برای مقیاس بندی خارجی سریع تدارک دید یا به سرعت برای مقیاس بندی داخلی سریع منتشر نمود. از نقطه نظر مشتریان، قابلیت های موجود باید نامحدود به نظر برسند و دارای قابلیت خریداری در هر مقدار در هر زمان باشند.

• خدمات اندازه گیری شده - خدمات خریداری شده توسط مشتریان را می توان تعیین کمیت نمود و اندازه گیری کرد. برای ارائه دهنده و مشتریان، استفاده از منابع باید نظارت، کنترل، اندازه گیری و گزارش شوند.

رایانش ابری, با توجه به ویژگی های جذاب آن, به یک مدل کسب و کار موفق و محبوب تبدیل شده است. علاوه بر مزایای در دست، از ویژگی های سابق نیز به مسائل امنیتی جدی مخصوص-ابر منجر می شوند. افرادی که نگرانی آنها, امنیت ابر است, همچنان برای انتقال کسب و کار خود به ابر تردید دارند. مسائل امنیتی, مانع غالب توسعه و استفاده گسترده از محاسبات ابری است. سه چالش اصلی برای امن و قابل اعتماد نمودن سیستم ابر وجود دارد:

• برون سپاری - برون سپاری, هزینه های سرمایه (CAPEX) و هزینه های عملیاتی را برای مشتریان ابر پایین می آورد. با این حال، برون سپاری نیز بدان معنی است که مشتریان از لحاظ فیزیکی کنترل بر داده ها و وظایف خود را از دست می دهند. مشکل از دست دادن کنترل به یکی از علل ریشه ای ناامنی ابر تبدیل شده است. برای رسیدگی به مسائل امنیت برون سپاری، ابتدا، ارائه دهنده ابر باید با ارائه رایانش و ذخیره سازی داده ها مورد اعتماد و امن, قابل اعتماد باشد؛ دوم، داده های برون سپاری شده و محاسبات باید از نظر محرمانگی، یکپارچگی، و دیگر سرویس

های امنیتی, برای مشتریان قابل تایید باشند. علاوه بر این، برون سپاری به طور بالقوه متحمل نقض حریم خصوصی خواهد شد، با توجه به این واقعیت که داده های حساس / طبقه بندی شده خارج از از کنترل مالکان هستند.

• چند-اجاره (مستاجر) - چند اجاره بدان معنی است که پلت فرم ابر به اشتراک گذاشته می شود و توسط مشتریان متعدد استفاده می شود. علاوه بر این، در یک محیط مجازی، داده های متعلق به مشتریان مختلف را می توان در همان دستگاه فیزیکی با سیاست تخصیص منابع معین قرار داد. دشمنانی که می توانند مشتریان ابر مشروع باشند, ممکن است از این موضوع سوء استفاده نمایند. مجموعه ای از مسائل امنیتی مانند نقض داده ها [5]، [17]، [29]، نقض محاسبات [5]، حمله سیل [26]، و غیره، متحمل بدین ترتیب ایجاد می شود. با اینکه چند-اجاره یک انتخاب قطعی فروشندگان ابر با توجه به بهره وری اقتصادی آن است، آسیب پذیری های جدیدی را برای پلت فرم ابر فراهم می کند. بدون تغییر الگوی چند-اجاره، طراحی مکانیزم های امنیتی جدید برای مقابله با خطرات احتمالی ضروری است.

• داده های عظیم و محاسبات شدید - محاسبات ابری قادر به هدایت ذخیره سازی داده های انبوه و وظایف محاسباتی شدید است. بنابراین، مکانیزم های امنیتی سنتی با توجه به محاسبات غیر قابل تحمل یا سربار ارتباطات کافی نیستند. به عنوان مثال، به منظور بررسی صداقت داده ها که از راه دور ذخیره می شوند، مخلوط کردن کل مجموعه داده ها غیر عملی است. برای این منظور، استراتژی ها و پروتکل های جدید انتظار می روند.

C. تکنیک های پشتیبانی

رایانش ابری از مجموعه ای از تکنیک های موجود، مانند شبکه بندی مرکز داده (DCN)، مجازی سازی، ذخیره سازی توزیع شده، MapReduce، برنامه های کاربردی وب و خدمات، و غیره استفاده کرده است.

مرکز داده های مدرن عملاً به عنوان یک حامل موثر از محیط های ابر به کار گرفته شده است. که عظیم محاسبات و قابلیت ذخیره سازی را با ترکیب هزاران دستگاه با تکنیک های DCN فراهم می کند.

تکنولوژی مجازی سازی به طور گسترده در محاسبات ابر برای تخصیص پویای منابع ارائه دهنده و تأمین خدمات ، به خصوص در IaaS استفاده می شود. با مجازی سازی، چند OS را می توان در دستگاه فیزیکی بدون تداخل با یکدیگر قرار داد.

MapReduce [53] یک چارچوب برنامه نویسی است که پشتیبانی از محاسبات توزیع شده بر روی مجموعه داده های جمعی حمایت می کند. این چارچوب, مجموعه های عظیمی از داده ها را به بلوک های کوچک تجزیه می کند که به سرورهای ابری برای محاسبات موازی توزیع می شوند. MapReduce پردازش دسته ای بر روی داده های عظیم را تسریع می کند که باعث می شود این کار به اولویت مدل محاسبات برای فروشندگان ابر تبدیل شود.

جدای از مزایا، تکنیک های سابق نیز تهدیدات جدیدی را نشان داده اند که دارای توانمندی به خطر انداختن امنیت ابر هستند. به عنوان مثال، مرکز داده های مدرن از مسائل تامین پهنای باند ضعیف رنج می برند [24]، که ممکن است مورد سوء استفاده قرار گیرد و ممکن است در نتیجه یک حمله DOS جدید [20] را با توجه به زیرساخت های به اشتراک گذاشته شده در محیط های ابری انجام دهند. همچنین روش دستگاه مجازی (VM) دارای قابلیت فعال کردن دشمنان برای انجام حملات متقابل VM [17] و حملات زمانی [60] با توجه به اقامت همزمان VM است. اطلاعات بیشتر در بخش دوم و بخش سوم مورد بحث قرار می گیرند.

D. روش صفت-محور

مسائل امنیتی و حفظ حریم خصوصی در محیط های ابری شده مطالعه شده اند و در نوشته های قبلی بررسی شده اند. برای درک بهتر این مسائل و ارتباطات آنها، محققان, معیارهای مختلف برای ساخت یک تصویر جامع را به کار گرفته اند.

Gruschka و همکاران [32], مدل سازی اکوسیستم امنیتی بر اساس سه شرکت کنندگان از سیستم ابر را نشان داده اند: خدمات کاربر، مثال خدمات و ارائه دهنده ابر. این نویسندگان, حمله را به شش رده طبقه بندی نموده اند (به عنوان مثال، کاربر به سرویس، خدمات به کاربر، کاربران به ابر، ابر به کاربر، خدمات به ابر و ابر به خدمات). این طبقه بندی برای توصیف تهدیدها و آسیب پذیری های ارائه شده در محاسبات ابری پیشنهاد شده است.

Subashini و همکاران [39] مسائل امنیتی ابر را بر اساس مدل های ارائه خدمات خلاصه نموده اند (به عنوان مثال، IaaS , PaaS، SaaS). Vaquero و همکاران [4] بررسی جامع دیگری را با توجه به هفت تهدید اصلی ارائه شده در [7] ارائه نموده اند. Grobauer و همکاران [70] به اهمیت تشخیص مسائل امنیتی به طور کلی از مسائل امنیتی ابر خاص اشاره کرده اند. علاوه بر این، آسیب پذیری های محاسبات ابری بحث می شوند و خلاصه ای از جنبه های مختلف ارائه می شوند.

در این مقاله، ما محیط ابر را به عنوان یک پلت فرم محاسبات جدید در نظر می گیریم که روش کلاسیک تحقیقات امنیتی را می توان به خوبی استفاده نمود. بنابراین، ما به کارگیری یک روش ویژگی صفت-محور را برای انجام بررسی خود تعیین می کنیم. ما اکوسیستم امنیت ابر و حریم خصوصی را از نظر پنج ویژگی امنیتی / حفظ حریم خصوصی (به عنوان مثال، محرمانگی، یکپارچگی، در دسترس بودن، مسئولیت پذیری و حفاظت از حریم خصوصی) نشان داده شده در شکل 2، به کار می گیریم که بسیاری از آنها, موارد نماینده در پیشرفت های پژوهش حاضر هستند. برخی از محققان, حریم خصوصی را به عنوان یکی از اجزای امنیتی در نظر می گیرند، در حالی که در این مقاله، به دلیل اهمیت و تخصص در محیط های ابری, حریم خصوصی را از امنیت جدا می کنیم. حریم خصوصی بسیار مرتبط با به امنیت و همچنین دیگر ویژگی های امنیتی در نظر گرفته می شود که دارای تاثیرات مثبت یا منفی بر حریم خصوصی هستند. اکوسیستم امنیتی, عام است و قابل اجرا برای هر کامپیوتر و سیستم های شبکه یم باشد. به منظور ساخت یک تصویر جامع از امنیت ابر و حریم خصوصی، ما به دنبال معانی اکوسیستم برای سازماندهی این نظرسنجی هستیم. با شروع از پنج ویژگی، آسیب پذیری هایی را مورد بحث قرار داده ایم که می توانند توسط دشمنان استفاده شوند. ما همچنین باید مدل های تهدیدی را بررسی نماییم که مهاجمان می توانند به منظور به خطر انداختن اهداف امنیتی استفاده نمایند. برخی تهدیدها و حملات به درستی انجام شده است، در حالی که دیگران هنوز هم باید حل شوند. این جنبه ها به امنیت و حریم خصوصی سیستم هایی غیر از محاسبات ابری مرتبط می شوند [99], [100], [101], [102], [103], [104], [105], [106], [107], [108],

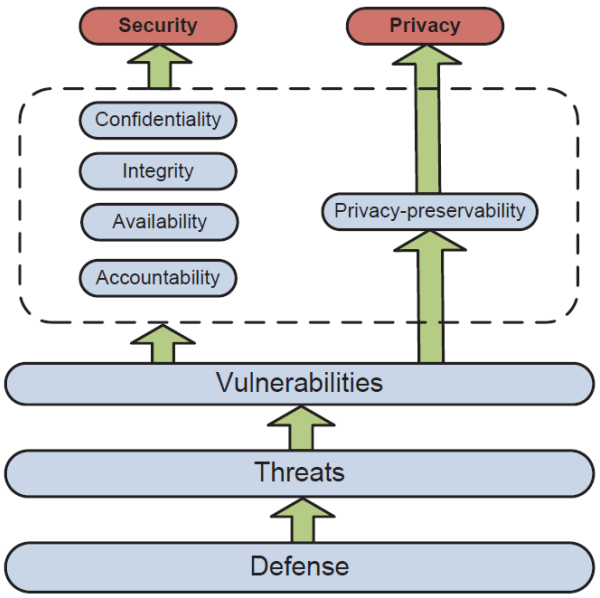
[109], [110], [111], [112], [113], [114], [115], [116], [117], [118], [119], [120], [121], [122], [123].

E. سیستم نماد

برای نشان دادن بهتر ارتباط میان آسیب پذیری، تهدید، و مکانیسم دفاعی. ما سیستم نماد زیر را به کار می گیریم: در نظر بگیرید که Vi, یک نوع از آسیب پذیری را نشان دهد، Ti.j نشاندهنده نوع تهدیدی است که از Vi استفاده می کند، و Di.j.k یک مکانیسم دفاعی را نشان می دهد که به Ti.j. می پردازد برای مثال، V1 آسیب پذیری می تواند توسط دشمنان به منظور ایجاد یک مدل تهدید T1.1 مورد سوء استفاده قرار گیرد که باید توسط راه حل امنیتی D1.1.1 رفع شود.

F. آسیب پذیری های ابر

1) اقامت همزمان VM - V1: در محاسبات ابری، اقامت همزمان (یا اجاره همزمان) بدان معنی است که چند مشتری مستقل, یک زیرساخت فیزیکی یکسان را به اشتراک می گذارند. بطور مشخص، دستگاه های مجازی متعلق به مشتریان مختلف می توانند در یک دستگاه فیزیکی قرار داده شوند. اقامت همزمان VM, مسائل امنیتی خاصی را پیش آورده است، مانند حمله VM-عرضی و Malicious SysAdmin [60].



شکل. 2. اکوسیستم امنیت ابر و حریم خصوصی

2) V2 - از دست دادن کنترل فیزیکی: مشتریان ابر, داده ها و برنامه های خود را به سرورهای ابری برون سپاری می نمایند. در نتیجه، مالکان, کنترل مستقیم بر مجموعه های داده ها و برنامه ها را از دست می دهند. از دست دادن کنترل فیزیکی بدان معنی است که مشتریان قادر به مقاومت در برابر حملات و حوادث خاص نیستند. به عنوان مثال، داده ها و یا نرم افزارها ممکن است تغییر یابند، گم شوند، و یا حتی حذف شوند؛ علاوه بر این، اطمینان از یکپارچگی و محرمانگی داده ها / محاسبات با روش های سنتی دشوار و غیر عملی است.

3) V3 - تأمین ضعیف پهنای باند: یک حمله سنتی DOS / DDOS در محاسبات ابری وجود دارد، و راه حل های نسبی در پژوهش های قبلی [21]، [22] ارائه شده اند. مخصوصاً برای محاسبات ابری، یک نوع جدید از حمله DOS [20] وجود دارد که از زیرساخت رایانش ابری کنونی ضعیف بهره می برد. با توجه به راهنمای طراحی سیسکو [24]، یک مرکز داده ها معمولاً برای تدارک ضعیف با یک ضریب 2.5: 1 تا 8: 1 طراحی می شود، به این معنی که ظرفیت شبکه واقعی بسیار کمتر از مجموع ظرفیت میزبان های واقع در همان زیر شبکه است.

4) V4 - مدل قیمت گذاری ابر: ابر رایانه پایبند به مدل قیمت گذاری پرداخت به محض اقدام [10] است که هزینه خدمات را از نظر معیارهایی مانند ساعت سرور، پهنای باند، ذخیره سازی، و غیره تعیین می کند. از آنجا که همه مشتریان ابر از نظر مالی مسئول خدمات مورد استفاده خود هستند، حمله کنندگان همیشه با بهره برداری از مدل قیمت گذاری, انگیزه هایی برای آزار و اذیت فرآیند صدور صورت حساب دارند. به عنوان مثال، حمله انکار اقتصادی توسعه پایدار (EDoS) [19] مدل قیمت گذاری آب و برق را دستکاری می کند و باعث هزینه های غیر قابل مدیریت برای مشتریان ابر می شود.

ادامه این مقاله به شرح زیر ساختاریابی شده است: بخش دوم تا VI به ترتیب به بحث در مورد محرمانگی، یکپارچگی، در دسترس بودن، مسئولیت پذیری، و حریم خصوصی در رایانش ابری می پرازد؛ سرانجام، این مقاله در بخش هفتم به پایان می رسد.

II. محرمانگی ابر

در هنگام برخورد با محیط های ابر، محرمانه بودن دلالت بر این دارد که داده های مشتری و وظایف رایانش باید از ارائه دهنده ابر و دیگر مشتریان محرمانه نگه داشته شوند. محرمانه بودن همچنان یکی از بزرگترین نگرانی ها با توجه به محاسبات ابری است. این مقله عمدتاً ناشی از این واقعیت است که مشتریان, داده ها و وظایف رایانش خود را به سرورهای ابر برون سپاری می کنند که توسط فراهم کنندگان ابر غیرقابل اطمینان بالقوه کنترل و مدیریت می شوند.

A. تهدیدات برای محرمانگی ابر

1) T1.1 - حمله عرضی-VM از طریق کانال های جانبی: Ristenpart و همکاران [17] وجود حملات عرضی-VM در پلت فرم AMAZON EC2 را نشان داده اند. یک حمله عرضی-VM از ماهیت چند اجاره سوء استفاده می کند که قادر می سازد که VMهای متعلق به مشتریان مختلف بتوانند در یک دستگاه فیزیکی یکسان بمانند. Aviram و همکاران [60], کانال های-جانبی زمانبندی را به عنوان یک تهدید موذی برای امنیت رایانش ابری در نظر گرفته اند, با توجه به این واقعیت که a) کانال های زمان بندی به طور فراگیر وجود دارند و کنترل آنها با توجه به ماهیت موازی سازی گسترده و زیرساخت های به اشتراک گذاشته سخت هستند. b) مشتریان مخرب قادر به سرقت اطلاعات بدون ترک دنباله و یا فعال کردن آلارم از مشتریان دیگری هستند. دو مرحله اصلی عملا برای آغاز چنین حمله ای وجود دارد:

• مرحله 1: قرار دادن. یک دشمن به قرار دادن یک VM مخرب بر روی سرور فیزیکی نیاز دارد که در آن VM هدف مشتری واقع شده است. برای رسیدن به این مورد، دشمن ابتدا باید تعیین کند که نمونه هدف VM در کجا واقع شده است؛ این کار را می توان با ابزارهای کاوش شبکه انجام داد مانند nmap, hping, wget غیره. یک مهاجم باید قادر به تعیین این مورد نیز باشد که آیا دو نمونه VM وجود دارند یا خیر؛ 1) مقایسه آدرس IP Domain0 برای دیدن اینکه آیا آنها مطابقت دارند یا خیر، و 2) اندازه گیری زمان رفت و برگشت کوچک بسته می تواند این کنترل را انجام دهد. صحت چک کردن های همزمان-مقیم می تواند توسط انتقال پیام ها بین نمونه ها از طریق یک کانال مخفی تایید شود. پس از همه کار آمادگی، یک نمونه مخرب VM باید در دستگاه فیزیکی هدف با

مشخص نمودن مجموعه ای از پارامترها ایجاد شود (به عنوان مثال، منطقه، نوع میزبانی)؛ دو استراتژی اساسی برای راه اندازی چنین VM وجود دارند: 1) استراتژی نیروی-حیوانی، که به سادگی بسیاری از نمونه ها را راه اندازی می کند و اقامت همزمان را با هدف چک می کند؛ 2) یک دشمن می تواند از این تمایل بهره برداری نماید که EC2, موارد جدیدی را در همان مجموعه کوچک از دستگاه های فیزیکی راه اندازی نماید. استراتژی دوم, از الگوریتم اختصاص دادن EC2 VM با شروع یک VM مخرب استفاده می کند, پس از اینکه یک VM قربانی راه اندازی می شود به طوری که آنها به احتمال زیاد به همان سرور فیزیکی اختصاص داده می شوند؛ این رویکرد قطعاً دارای میزان موفقیت بهتر در قرار دادن است.

• مرحله 2: استخراج. بعد از مرحله 1، VM مخرب اقامت همزمانی با VM قربانی خواهد داشت. از آنجا که VM مخرب و قربانی منابع فیزیکی خاصی را به اشتراک می گذارند، مانند حافظه نهان داده ها، دسترسی به شبکه، پیش بینی کننده های شاخه CPU، خطوط لوله CPU، و غیره، راه های بسیاری وجود دارند که یک دشمن می تواند برای حملات استفاده کند 1) اندازه گیری استفاده از حافظه پنهان که می تواند بار کنونی سرور را تخمین بزند؛ 2) برآورد یک نرخ ترافیک که می تواند تعداد بازدید کنندگان را به دست آورد و یا حتی صفحات غالب درخواست شده را به دست آورد 3) یک حمله زمانبندی ضربه زدن به کلید که می تواند رمز عبور قربانی را توسط اندازه گیری زمان بین زدن کلیدها سرقت نماید.

یک کار پیگیری، کانال های مختلف پنهان مورد بررسی قرار می گیرد و تحلیل عمیق ارائه می شود. مهاجمان به راحتی می توانند از حافظه پنهان L2، با توجه به پهنای باند بالای آن بهره برداری نمایند. Xu و همکاران به طور خاص کانال حافظه پنهان تبدیل L2 را با ارزیابی کمی بررسی نمودند. [71]. نشان داده شده است که حتی نرخ بیت کانال از کار سابق [17] بالاتر است، قابلیت کانال برای فیلتر کردن اطلاعات مفید خارجی هنوز محدود است، و تنها برای نشت اسرار کوچک مانند کلیدهای خصوصی عملی است. Okamura و همکاران یک حمله جدید را توسعه دادند که نشان می دهد که بار CPU را نیز می توان به عنوان یک کانال مخفی برای رمزگذاری اطلاعات مورد استفاده قرار داد [72]. حمله افشای حافظه [81]، [82] نوع دیگری از حمله متقابل VM است. در یک محیط

مجازی، تکثیرزدایی حافظه, یک تکنیک برای کاهش استفاده از حافظه فیزیکی با به اشتراک گذاشتن صفحات حافظه با همان مطالب است. با اندازه گیری زمان دسترسی نوشتن که بین صفحات تکثیرزدایی شده و صفحات منظم متفاوت است, یک حمله افشای حافظه, قادر به تشخیص وجود یک نرم افزار و یا یک فایل در یک VM اقامت همزمان است.

2) sysadmin -T1.2 مخرب: حمله عرضی-VM بحث می کند که چگونه دیگران ممکن است محرمانه بودن مشتریان ابر را که به طور همزمان با قربانی اقامت دارند، نقض نماید, هر چند تنها تهدید نیست. sysadmin ممتاز ارائه دهنده ابر می تواند حملات را با دسترسی به حافظه VMهای مشتری انجام دهد. به عنوان مثال، Xenaccess [30]], sysadmin را قادر می سازد تا در زمان اجرا, به طور مستقیم با اجرای یک فرایند سطح کاربر در Domain0 به حافظه VM دسترسی داشته باشد.

B. استراتژی های دفاع

رویکردها برای پرداختن به حمله VM-عرضی در شش رده قرار می گیرد: a) به نظر می رسد که پیشگیری از قرار دادن موجب کاهش نرخ موفقیت قرار دادن می شود؛ b) اجرای جداسازی فیزیکی [80]؛ c) طرح های حافظه پنهان جدید [75]، [76]، [77]، [78]؛ d) زمان فازی در نظر دارد تا توانایی مخرب VM برای دریافت سیگنال را با حذف تایمرهای ریز دانه تضعیف نماید [73]؛ e) جبر اجباری VM [60] عدم زمان و یا دیگر نشت های غیر قطعی اطلاعات به دشمنان را تضمین می کند؛ f) پیاده سازی رمزنگاری حافظه پنهان مقاوم در برابر زمانبندی [79]. از آنجا که c) d)، e)، و f), استراتژی های دفاعی مخصوص-ابر نیستند، ما جزئیات را در این بخش نمی گنجانیم.

1) D1.1.1 - پیشگیری از جایگذاری: به منظور کاهش خطر ناشی از زیرساخت به اشتراک گذاشته، چند پیشنهاد برای دفاع از حمله در هر مرحله در [17] داده شده است. برای مثال، ارائه دهندگان ابر می توانند اقامت همزمان را با داشتن Dom0 برای پاسخ ندادن به Traceroute و / یا به طور اختصاص دادن تصادفی آدرس های داخلی IP برای VMهای راه اندازی شده مبهم نمایند. برای کاهش نرخ موفقیت قرار دادن، ارائه دهندگان ابر به کاربران اجازه

می دهند تا تصمیم بگیرند که در چه جایی دستگاه های مجازی خود را قرار دهند. اما، این روش از استراتژی نیروی-حیوان جلوگیری نمی کند.

2) D1.1.2 - تشخیص اقامت همزمان: راه حل نهایی حمله متقابل-VM, از بین بردن اقامت همزمان است. مشتریان ابر (به خصوص شرکت ها) ممکن است به جداسازی فیزیکی نیاز داشته باشند, که حتی می توانند به صورت موافقت نامه های سطح خدمات نوشته شوند (SLA ها). با این حال، فروشنده ابر ممکن است تمایلی به رهاسازی مجازی سازی نداشته باشد که برای صرفه جویی در هزینه و منابع مفید است. یکی از گزینه های باقیمانده, به اشتراک گذاشتن زیرساخت تنها با VMهای "دوستانه" است، که متعلق به همان مشتری یا مشتریان قابل اعتماد دیگر هستند. برای اطمینان از جداسازی فیزیکی، مشتری باید قادر به تایید استفاده دستگاه های مجازی خود از یک دستگاه فیزیکی باشد. HomeAlone یک سیستم [80] است که اقامت همزمان را با استفاده از یک کانال-جانبی (در حافظه کش L2) به عنوان یک ابزار تشخیص تشخیص می دهد. ایده این کار, ساکت کردن فعالیت دستگاه های مجازی "دوستانه" در یک بخش انتخاب شده از حافظه پنهان L2 برای یک مقدار مشخصی از زمان، و سپس اندازه گیری استفاده از حافظه پنهان برای بررسی این مورد است که آیا هر گونه فعالیت غیر منتظره وجود دارد که نشان دهد که دستگاه فیزیکی با مشتری به طور همزمان اقامت دارند یا خیر.

3) *D1.1.3 – NoHype*: NoHype ([83], [84]) برای به حداقل رساندن درجه زیرساخت به اشتراک گذاشته شده توسط از بین بردن هایپروایزر تلاش می کند در حالی که هنوز هم ویژگی های کلیدی مجازی سازی را حفظ می کند. معماری NoHype, چند ویژگی را فراهم می کند: i) ویژگی "یک هسته در هر VM" مانع تداخل بین دستگاه های مجازی می شود، کانال های جانبی مانند حافظه نهان L1 را از بین می برد، و چند اجاره را حفظ می کند، زیرا هر تراشه دارای چند هسته است. ii) پارتیشن حافظه, هر دسترسی به حافظه VM بر روی یک طیف اختصاص داده شده را محدود می کند؛ iii) دستگاه های I / O مجازی اختصاص داده شده, اجازه می دهد تا به هر VM, دسترسی مستقیم به یک دستگاه I / O مجازی اختصاصی اعطا شود. No-Hype به طور قابل توجهی, سطح حمله مجازی را کاهش داده است و سطح ایزولاسیون VM را افزایش داده است. با این حال، NoHype به تغییر

سخت افزار نیاز دارد که در هنگام در نظر گرفتن اعمال آن برای زیرساخت های ابر جاری, آن را کمتر عملی می سازد.

4) D1.2.1 - بستر رایانش ابری مورد اعتماد: سانتوس و همکاران [29] یک پلت فرم رایانش ابری مورد اعتماد (TCCP) را ارائه نمودند که محیط اجرایی جعبه بسته برای خدمات IaaS را ارائه می دهد. TCCP, اجرای محرمانه دستگاه های مجازی مهمان را تضمین می کند. همچنین مشتریان را قادر می سازد تا به فراهم کننده IaaS گواهی بدهند و تعیین نمایند که آیا خدمت قبل راه اندازی VMها در ابر, امن است یا خیر.

اهداف طرح TCCPعبارتند از: 1) محدود کردن اجرای VM در داخل محیط امن؛ 2) که sysadmin ریشه قادر به دسترسی به حافظه یک VM میزبانی شده در یک گره فیزیکی نیست. TCCP تکنیک های موجود برای ساخت سیستم عامل محاسبات ابری مورد اعتماد را اعمال می کند. و بر حل مسائل محرمانگی برای داده های کاربران و برای محاسبه برون سپاری شده به ابر تمرکز می کند. با TCCP, sysadmin قادر به بازرسی یا مذاکرات پنهانی در مورد محتوای VMهای در حال اجرا نیست.

5) دیگر نظرات: حفظ کنترل داده ها به مشتری: با توجه به ترس مشتری از گم شدن کنترل داده ها در محیط ابر، Descher و همکاران [40] حفظ کنترل داده ها برای مشتریان ابر را به سادگی با ذخیره سازی رمزگذاری صوتی بر روی سرورهای ابر پیشنهاد نمودند. تصاویر رمزگذاری شده VM, کنترل دسترسی را تضمین می کند, زیرا فقط کاربران مجاز شناخته شده به عنوان دارندگان کلید مجاز به دسترسی هستند. با توجه به رمزنگاری، داده ها قابل نصب نیستند و بدون یک کلید دسترسی, درون ابر اصلاح می شوند و محرمانه بودن و یکپارچگی را تضمین می کنند. این رویکرد, تضمین های امنیتی قبل از راه اندازی VM را ارائه می دهد. هرچند, روش هایی برای حمله به VM در طی زمان اجرا [30] و برای به خطر انداختن داده ها و محاسبات وجود دارد.

C. با توجه به محرمانه بودن، حمله متقابل VM و sysadmin مخرب به طور عمده یک سیستم ابر را تهدید می کند؛ هر دو تهدیدات از آسیب پذیری های مجازی سازی و اقامت همزمان استفاده می کنند. دیگر مستاجران, حمله متقابل VM را انجام می دهند، در حالی که sysadmin و مخرب در داخل حمله فروشگاه ابر هستند. دفاع از این

تهدیدات یک کار بی اهمیت با توجه به حقایق زیر است: 1) کانال های جانبی مختلف و دیگر اجزای مشترک را می توان مورد سوء استفاده قرار داد، و دفاع از هر یک از آنها است، کار آسانی نیست. 2) مسائل معدودی هستند که باید بررسی شوند:

- تشخیص اقامت همزمان به عنوان یک روش قطعی در نظر گرفته می شود زیرا مشتریان باید قادر به بررسی این مورد هستند که آیا جداسازی فیزیکی به خوبی اجرا شده است یا خیر. HomeAlone [80] دارای توانایی برای رسیدن به دقت تشخیص در کانال سمت حافظه پنهان L2 است. با این حال، علاوه بر حافظه نهان L2، کانال های جانبی دیگر نیز می توانند به خوبی مورد استفاده قرار گیرند. بنابراین، به منظور ارائه تشخیص کامل اقامت همزمان، مجموعه ای از روش های تشخیص با هدف کانال های جانبی مختلف باید توسعه یابند.
- NoHype پنجره دیگری را برای مقابله با تهدید متقابل VM باز کرده است. با این حال، سخت افزار کالای کنونی, محدودیت را برای اجرای NoHype تحمیل نموده است. علاوه بر این، مهاجرت VM زنده به خوبی توسط این معماری جدید پشتیبانی نشده است. بنابراین، قبل از ساخت یک گام رو به جلو، محققان به پرداختن به تغییرات سخت افزاری به جای NoHype و حفظ ویژگی های بیشتر برای مدیریت VM نیاز دارند.

III. انسجام ابر

مشابه با محرمانه بودن، مفهوم صداقت در محاسبات ابری مرتبط با هر دوی صداقت داده ها و یکپارچگی محاسبات است. صداقت داده ها نشان می دهد که داده ها باید صادقانه در سرورهای ابری ذخیره شوند، و هر گونه تخلف (به عنوان مثال، اطلاعات از دست رفته ، تغییر داده، یا در معرض خطر) باید شناسایی شوند. یکپارچگی محاسبات, این تصور را نشان می دهد که برنامه ها بدون تحریف توسط نرم افزارهای مخرب، ارائه دهندگان ابر، و یا دیگر کاربران مخرب اجرا می شوند، و اینکه هر محاسبات نادرست تشخیص داده خواهد شد.

A. تهدیدات برای انسجام ابر

1) $T2.1$ – گم شدن/ دستکاری داده ها: در ذخیره سازی ابر، برنامه های کاربردی, ذخیره سازی را به عنوان یک خدمت ارائه می دهند. سرورها مقادیر زیادی از داده را حفظ می کنند که در موارد نادر قابل دسترسی است. سرور ابر

از نظر هر دو امنیت و قابلیت اطمینان [14]، که بدان معنی است که داده ها ممکن است از دست بروند و یا به طور بدتر یا به طور تصادفی اصلاح شوند. اشتباهات اجرایی ممکن است موجب از دست رفتن داده ها (به عنوان مثال، تهیه پشتیبان و بازیابی، انتقال داده ها و تغییر عضویت در سیستم های P2P [11]) شود. علاوه بر این، ممکن است دشمنان, حملات را با استفاده از از دست دادن کنترل صاحبان داده ها بر داده های خود شروع کنند.

2) T2.2 - محاسبات نادرست در سرورهای از راه دور: با محاسبات برون سپاری شده، قضاوت در این مورد دشوار است که آیا محاسبه با صداقت بالا اجرا شده است یا خیر. از آنجا که جزئیات محاسبات به اندازه کافی برای مشتریان ابر شفاف نیست، سرورهای ابری ممکن است با سوء نیست رفتار کنند و نتایج محاسباتی نادرست را ارائه دهند؛ آنها ممکن است از مدل نیمه صادق پیروی نکنند. به عنوان مثال، برای محاسباتی که نیاز به مقدار زیادی از منابع محاسباتی دارد، انگیزه هایی برای ابر به منظور "تنبل" شدن وجود دارد [85]. از سوی دیگر، حتی اگر مدل نیمه صادق پیروی شود، مشکلاتی ممکن است بوجود می آیند, زمانی که که یک سرور ابری از کد آسیب پذیر و از رده خارج استفاده می کند که قبلا توسط روت کیت یا کدها و یا داده های [86] مخرب مورد حمله قرار گرفته است.

B. استراتژی های دفاع

1) D2.2.1 - مالکیت قابل اثبات داده ها (PDP): چک کردن صداقت داده ها, یک موضوع بلند مدت تحقیقاتی [62]، [68] است. با این حال، روش های سنتی نمی توانند به درستی با چالش های چک کردن صداقت ارائه شده در ذخیره سازی ابر مقابله نمایند. چالش اصلی چک کردن صداقت اینست که مقدار زیادی از داده ها از راه دور بر روی سرور ابر غیر قابل اعتماد ذخیره می شوند. به عنوان یک نتیجه، روش هایی که نیاز به هش کردن برای کل فایل دارند, سنگین می شوند. علاوه بر این، دانلود فایل از سرور و انجام چک کردن یکپارچگی با توجه به این واقعیت عملی نیست که این کار از نظر محاسباتی گران است و همچنین پهنای باند مصرف می کند. هر یک از مفاهیم سابق در محیط های ابری قابل قبول نیست.

مالکیت قابل اثبات داده ها، که به عنوان ((PDP 11]] ، [12] [14] [15] نامیده می شود, از طریق فرایند چک کردن صداقت داده ها با ذخیره سازی ابر به منظور پاسخ به این سوال به کار گرفته می شود، "آیا مطمئن شدن از اینکه اطلاعات برون سپاری شده صادقانه در ابر ذخیره می شوند, برای مشتریان ممکن است؟ "

الف) یک روش ساده و بی تکلف

برای مقایسه، یک روش ساده و بی تکلف در [12] داده شده است. این ایده شامل محاسبه یک مقدار هش برای فایل F با کلید k (به عنوان مثال، h(F ،K)) و پس از آن ارسال F به سرور توسط مشتری می شود. هنگامی که مشتری یک ضرورت برای بررسی فایل پیدا می کند، k را آزاد می کند و k را به سرور می فرستد، که پس از آن خواسته می شود تا دوباره مقدار هش را بر اساس F و K محاسبه نماید؛ پس از آن، سرور به مشتری با نتیجه هش برای مقایسه پاسخ می دهد. مشتری می تواند چک های متعدد را با نگه داشتن کلید ها و مقادیر هش متفاوت آغاز کند. این رویکرد, یک اثبات قوی را فراهم می کند که سرور هنوز هم F را حفظ می کند. با این حال، جنبه منفی, سربار بالا است که تولید شده است. این سربار وجود دارد، زیرا هر زمان تایید به سرور برای اجرای یک فرآیند هش کردن روی کل فایل نیاز دارد. مفهوم در این لحظه, از نظر محاسباتی، حتی برای عملیات های هش کردن سبک پر هزینه است.

b) مالکیت قابل اثبات داده های اصلی (PDP)

مدل PDP اصلی [11] مستلزم آن است که داده ها در مرحله راه اندازی به منظور ترک برخی از فراداده ها در سمت سرویس گیرنده جهت مقاصد تایید پیش پردازش شوند, زیرا آن داده ها باید به ابر سرور ارسال شوند. هنگامی که مشتری, یک ضرورت برای بررسی صداقت داده ها را در زمان بعدی احساس می کند، یک چالش را به سرور ابر می فرستد، که با یک پیام بر اساس محتوای داده ها پاسخ خواهد داد. بعد از ترکیب پاسخ و فراداده های محلی، مشتری قادر به اثبات این مورد است که آیا یکپارچگی داده ها نقض شده است یا خیر. تضمین احتمالاتی PDP نشان می دهد که PDP می تواند تشخیص سوء رفتار سرور را با سربار کم محاسباتی و ذخیره سازی به احتمال زیاد به دست آورد.

PDP فقط برای فایل های استاتیک (به عنوان مثال، فقط فایل های-الحاقی) قابل اجرا است، به این معنی که زمانی که داده ها برای سرور آپلود می شوند, قابل تغییر نیستند. این محدودیت, کاربرد آن برای رایانش ابر را محاسبات با توجه به این واقعیت کاهش می دهد که این کار با مدیریت داده های دینامیک انجام می شود.

c) اثبات بازیابی (POR)

اثبات قابلیت بازیابی (12] (POR], یک پروتکل ممیزی را در هنگام حل یک مشکل مشابه برای PDP به کار می گیرد. مشکل این است که هر یک از دو مشتری را قادر به بررسی یکپارچگی اطلاعات برون سپاری شده بدون نیاز به بازیابی آن می سازد. POR به صورت بسیار سبک وزن طراحی شده است. به عبارت دیگر، برای به حداقل رساندن ذخیره سازی در سمت کلاینت و سرور، پیچیدگی ارتباطات یک ممیزی، و تعداد بلوک های-داده مورد دسترسی در طول یک ممیزی [12] تلاش می کند.

پروتکل POR در شکل 3 نشان داده شده است. کاربر, تنها یک کلید را ذخیره می کند که برای رمزگذاری فایل F به منظور رسیدن به فایل رمزگذاری شده F استفاده می شود. وظیفه این است که مجموعه ای از مقادیر نگهبان برای F تعبیه شوند، و سرور فقط F را بدون دانستن اینکه مأموران ممکن است در کجا باشند, ذخیره می کند, زیرا مأموران از بلوک های داده منظم غیر قابل تشخیص هستند. در پروتکل چالش و پاسخ، از سرور خواسته می شود تا زیر مجموعه ای معین از مأموران در F را بازگرداند. اگر سرور دستکاری شود و یا F پاک شود، این احتمال بالاست که مأموران خاص نیز خراب شوند و یا از دست بروند؛ این باعث سرور عدم توانایی در تولید یک اثبات کامل برای فایل اصلی می شود. بنابراین، مشتری دارای شواهد برای ثابت نمودن این مورد است که سرور فایل را خراب کرده است. با توجه به این واقعیت است که تعداد مأموران محدود است، POR, کدهای تصحیح-خطا را برای بازیابی فایل تنها با بخش کوچکی خراب اتخاذ می کند.

مشابه PDP، POR فقط برا فایل های استاتیک قابل استفاده است. یک تغییر ظریف برای فایل, پروتکل را خراب خواهد کرد و به طور کامل کاربران را به اشتباه می اندازد.

d) PDP مقیاس پذیر

PDP مقیاس پذیر 14]] یک نسخه بهبود یافته از اصل PDP است. تفاوت در این دو به صورت زیر توصیف می شود: 1) PDP مقیاس پذیر, رمزگذاری کلید متقارن را به جای کلید عمومی به منظور کاهش سربار محاسبه اتخاذ می کند، اما PDP مقیاس پذیر, تأیید عمومی را با توجه به رمزگذاری متقارن پشتیبانی نمی کند. 2) PDP مقیاس پذیر, عملیات های پویا را بر روی داده های از راه دور اضافه نموده است. یکی از محدودیت های PDP مقیاس پذیر اینست که همه چالش ها و پاسخ ها از پیش محاسبه می شوند و تعداد به روز رسانی ها محدود و ثابت است.

e) PDP پویا

هدف از PDP پویا 15] (DPDP)] حمایت کامل از عملیات های پویا (به عنوان مثال، اضافه، درج، ویرایش، و حذف). هدف از عملیات پویا, فعال کردن تصدیق درج و حذف توابع با دایرکتوری های تصدیق شده مبتنی بر رتبه است که بر روی یک لیست گذر ساخته می شوند. نتیجه آزمایش نشان می دهد که، اگر چه حمایت از به روز رسانی های پویا دارای برخی از پیچیدگی های محاسباتی است، DPDP عملا کارآمد می باشد. به عنوان مثال، برای تولید اثبات یک فایل GB1، DPDP تنها KB 415 داده اثبات و 30 میلی ثانیه سربار محاسباتی را تولید می کند.

پروتکل DPDP سه عملیات جدید که به عنوان PrepareUpdate، PerformUpdate و VerifyUpdate شناخته شده است را معرفی می کند. PrepareUpdate توسط یک مشتری به منظور تولید یک درخواست به روز رسانی اجرا می شود که شامل به روز رسانی ها برای انجام (به عنوان مثال، تغییر بلوک **i**، حذف بلوک **i**، و غیره) می شود. PerformUpdate توسط یک سرور برای انجام به روز رسانی فایل واقعی اجرا می شود، و پس از آن یک اثبات به روز رسانی را به مشتری باز می گرداند که به نوبه خود رفتار سرور را در طول روز رسانی تایید می کند.

f) Hail: یک دسترس بودن بالا و لایه صداقت برای ذخیره سازی ابر

HAIL [16] متفاوت از کار قبلی است, با توجه به این واقعیت که یک محیط توزیع شده را در نظر می گیرد که در آن مشتری باید یک فایل را در چندین سرور با افزونگی گسترش دهد و تنها یک حالت کوچک ثابت را در دستگاه محلی ذخیره نماید. تهدیدهایی اصلی که HAIL با آنها مبارزه می کند, دشمنان تلفن همراه هستند، که ممکن است فایل F را احتمالا با تضعیف سرورهای متعدد فاسد نماید.

g) خلاصه ای از PDP

PDP یک کلاس از مشکلاتی است که رویکردهای کارآمد و عملی را به منظور تعیین این مورد فراهم می کنند که آیا داده های برون سپاری صادقانه ذخیره می شوند یا خیر. ما چندین روش تازه در حال ظهور را در جدول I خلاصه نموده ایم و مقایسه کرده ایم. تکامل PDP نشاندهنده بهبودها از داده های استاتیک به داده های پویا و همچنین تنظیم تک سرور برای تنظیمات سرویس دهنده های توزیع شده است.

2) D2.1.2 – ممیز شخص ثالث: به جای اجازه دادن به مشتریان برای بررسی یکپارچگی داده ها، وظیفه چک کردن صداقت داده ها برای شخص ثالث که می تواند توسط هر دو ارائه دهنده ابر و مشتریان مورد اعتماد قرار گیرد, نیز ممکن است. Wang و همکاران [45] اتخاذ یک ممیز شخص ثالث (TPA) را برای بررسی یکپارچگی داده های برون سپاری شده در محیط های ابری پیشنهاد دادند. TPA موارد زیر را تضمین می کند: 1) داده های ابر می توانند به طور موثر بدون کپی داده های محلی ممیزی شوند، و مشتریان ابر از عدم سربار آنلاین برای ممیزی رنج می برند؛ 2) هیچ آسیب پذیری جدیدی برای به خطر انداختن حفظ حریم خصوصی داده ها ایجاد نخواهد شد.. این روش کلیدی یک تصدیق کننده هومومورفیک است که در نوشته های موجود [46] استفاده شده است. هنگامی که ترکیب یک تصدیق کننده هومومورفیک با پوشش تصادفی، TPA قادر به دسترسی به محتوای داده هاست در حالی که ممیزی انجام می شود.

3) مبارزه با محاسبات نادرست: ویژگی برون سپاری محاسبات ابری به محققان انگیزه می دهد تا دوباره مشکل کلاسیکی را بررسی کنند که به یکپارچگی محاسبات خارجی می پردازد. چگونه ممکن است یک دستگاه, محاسبات خود را به دستگاه دیگری برون سپاری نماید و پس از آن، بدون اجرای محاسبه به صورت محلی، صحت نتیجه خروجی را توسط دستگاه دیگری بررسی نماید [87]؟ استراتژی های متعارف برای بررسی صداقت محاسبات خارجی در چهار گروه قرار می گیرند:

• D2.2.1: **محاسبات دوباره** به یک دستگاه محلی برای انجام دوباره محاسبه، و سپس مقایسه نتایج نیاز دارد. دوباره محاسبات, دقت 100٪ آشکارسازی اشتباه را تضمین می کند و نیازی به اعتماد نمودن به فروشنده ابر ندارد.

با این حال، توجه به این واقعیت که هر یک از تاییدیه ها به حداقل زمان زمان با محاسبات اصلی نیاز دارند, معمولا هزینه غیر قابل تحمل است. برای این منظور، مشتریان احتمالا انگیزه ای برای بررسی یکپارچگی محاسبات در این روش ندارند. یک تغییر در محاسبات دوباره, نمونه برداری است [66]، که حداقل تضمین های احتمالاتی تشخیص اشتباه را بسته به درجه نمونه برداری ارائه می دهد. نمونه برداری, دقت اثربخشی را سبک سنگین می کند.

• D2.2.2: **تکرار**, یک وظیفه محاسبه را به چند دستگاه اختصاص می دهد و سپس نتایج را مقایسه می کند. رای گیری اکثریت را می توان برای تعیین صحت استفاده نمود. تکرار, نیمه اعتماد به فروشنده ابر را فرض می کند زیرا هر دوی محاسبات و تایید از راه دور انجام می شوند. دشمنان هوشمند که مقادیر معینی از دستگاه ها را کنترل می کنند, ممکن است چک کردن تکرار را با بازگشت همان نتایج نادرست دور بزنند.

• D2.2.3: ممیزی [51]، [89] معمولاً همراه با ثبت وقایع کار می کند. در طول اجرای یک محاسبه، جزء ورود به سیستم, همه وقایع مهم را برای یک فایل ورود به سیستم ثبت می کند که بعد از آن به یک یا چند ممیز برای بازبینی ارسال می شود. ممیزی یک رویکرد معمول برای انجام تحقیق و بررسی پزشکی قانونی است. یک اشکال ممیزی این است که اگر مهاجم, محاسبات را بهتر از ممیز درک کند، دستکاری بیت داده ها بدون شناسایی شدن ممکن است.

• D2.2.4: با محاسبه مورد اعتماد [29] ، 58]] کامپیوتر را برای رفتار مداوم به روش های مورد انتظار با پشتیبانی سخت افزار و نرم افزار به اجرا در می آورد. تکنیک کلیدی چک کردن صداقت به عنوان گواهی از راه دور شناخته شده است که با داشتن سخت افزار, یک گواهی را تولید می کند که بیان می کند چه نرم افزاری در حال اجرا است. پس از آن گواهی را می توان به تصدیق کننده ارسال نمود تا نشان دهد که این نرم افزار بدون تغییر است. یک فرض از محاسبات قابل اعتماد اینست که برخی از اجزای مانند سخت افزار و هایپرویزور از لحاظ فیزیکی تغییر نمی کنند. برخی از روش های تایید ([85]، [90]، [91]، [92]), مخصوص دامنه و یا نرم افزار هستند. به عنوان مثال، Wang و همکاران یک مکانیسم عملی برای برون سپاری برنامه ریزی خطی را (LP) طراحی نموده اند [85]؛ با کاوش در قضیه دوگانگی محاسبات LP, و استخراج شرایطی که نتیجه درست باید برآورده سازد، مکانیزم امنیتی تنها متحمل

هزینه اضافی نزدیک-به-صفر در هر دو سرور ابری و مشتریان می شود. روش Freivalds [89], ضرب ماتریس m*m را در o(M2) با یک الگوریتم تصادفی تأیید میکند. بلانتون و همکاران [92] یک رویکرد برون سپاری محاسبات بیومتریک در مقیاس بزرگ با سربار مناسب را ارائه می دهد.

روش های بالا بر فرضیات مختلف تکیه می کنند که دارای محدودیت هایی هستند و یا متحمل هزینه محاسبات بالا می شوند. با این حال، هدف نهایی، فراهم نمودن تایید انسجام عملی و بی قید و شرط برای محاسبات از راه دور است. برای رسیدن به این هدف، SETTY و همکاران [95] جستجوی کمک از برخی از نتایج اولیه تحقیقات مانند اثبات تعاملی [93]، و اثبات احتمالاتی قابل کنترل (PCP ها) [94] را پیشنهاد نمودند که برای طراحی یک روش ایده آل تلاش می کند که یک مشتری را قادر می سازد تا به بررسی صحت پاسخ در زمان ثابت، با یک اثبات کد گذاری شده مناسب و تحت شانس ناچیز مثبت نادرست بپردازد [94]. تا به امروز، روش تأیید مبتنی بر PCP, برای اهداف عمومی، با توجه به تحقیقات SETTY, عملی نبوده است. اگر چه استفاده از آن در یک مشکل خاص (به عنوان مثال، ماتریس ضرب) دلگرم کننده به نظر می رسد، نویسنده همچنین به برخی از مسائل جدی مانند زمان راه اندازی گران اشاره نموده است.

C. خلاصه و مسائل باز

صداقت ابر به این دلیل آسیب پذیر می شود که مشتریان از لحاظ فیزیکی داده ها و نرم افزار خود را کنترل نمی کنند. برای صداقت داده ها، دو چالش وجود دارد: i) حجم بزرگ داده ها, طرح هش کردن معمول را دوام پذیر نمی سازد. ii) چک کردن صداقت تنها زمانی می تواند عملی شود که الزامات اضافی وجود دارند، که دشواری را افزایش می دهند؛ به عنوان مثال، عملیات پویا بر روی داده های از راه دور, با تضمین های صداقت، غیر بدیهی است و در تنظیم توزیع شده، هر دوی انسجام و هماهنگی در نظر گرفته می شوند. از سوی دیگر، صداقت محاسبات سخت تر است. چالش اصلی, عدم دانش از موارد داخلی محاسبات است؛ اگر روش تایید برای محاسبات عمومی اعمال شود. یک روش چک کردن صداقت به خوبی طراحی شده شرایط زیر را برآورده می سازد: i) برای نگرانی های عملی، حجم کار محاسبه محلی تایید باید کمتر از محاسبات اصلی باشد. در غیر این صورت, انجام برون سپاری چندان موثر

نیست. ii) اثبات می تواند توسط هر شخص برای اطمینان از عدم انکار تایید شود؛ iii) هیچ و یا چند فرض معدود تحمیل می شوند.

پتانسیل تحقیقات زیادی در این حوزه وجود دارند:

• ترکیب چک کردن صداقت با سایر الزامات واقع گرایانه, یک روند تحقیقات امیدوار کننده است. تحقیقات مدرن, حمایت از عملیات پویا در اطلاعات ابر را مطالعه کرده اند؛ همچنین، تنظیم دستگاه واحد به تنظیم توزیع شده گسترش می یابد. با این حال، هیچ یک از آثار قبل به بررسی چک کردن صداقت همراه با هر دو عملیات پویا و تنظیم توزیع شده نپرداخته اند. علاوه بر این، دیگر ویژگی های سنتی در سیستم توزیع شده مانند تحمل خطا را نیز می توان در نظر گرفت.

• در صورتی پیشرفت قابل توجه حاصل خواهد شد که یک روش تایید عملی و ساخته شده بی قید و شرط برای صداقت محاسبات توسعه یابد. این یک تلاش مهم [95] به منظور کاهش پیچیدگی محاسبات در هنگام استفاده از اثبات های احتمالاتی برای ضرب ماتریس به عنوان مطالعه موردی است. با این حال، حتی در یک مطالعه موردی، چند مشکلات جدی بوجود می آیند. فضای بهبود هنوز هم بزرگ است.

• از سوی دیگر، روش مخصوص-دامنه می تواند به اثرات رضایت بخش دستیابی پیدا کند به دلیل اینکه این روش ها معمولا از ویژگی های درونی محاسبه استفاده می نمایند. برخی از محاسبات علمی مانند برنامه ریزی خطی و عملیات ماتریس دارای نسخه های برون سپاری خود هستند. با این حال، انواع دیگر مانند بهینه سازی غیر خطی همچنان باید به ابر با اطمینان از صداقت قوی برون سپاری شوند.

IV. در دسترس بودن ابر

در دسترس بودن از توابع اصلی محاسبات ابری بسیار مهم است که خدمات بر حسب تقاضا را در سطوح مختلف ارائه می دهد. اگر یک خدمت خاص دیگر در دسترس نباشد و یا کیفیت خدمات نتواند توافقنامه سطح خدمات (SLA) را

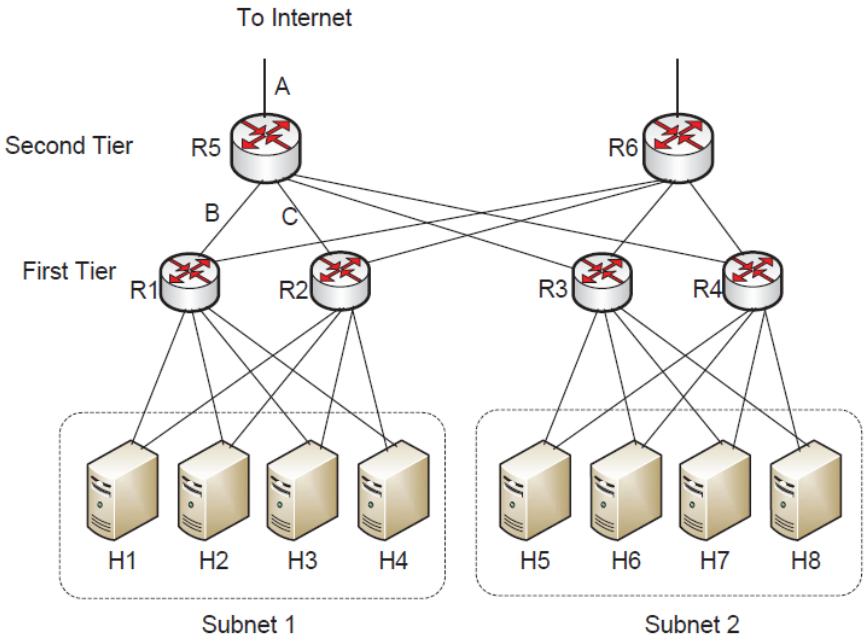
برآورده سازد، مشتریان ممکن است ایمان به سیستم ابر را از دست بدهند. در این بخش، ما دو نوع از تهدیدات که دسترس بودن ابر را مختل می کنند, مورد مطالعه قرار داده ایم.

A. تهدیدات برای در دسترس بودن ابر

1) T3.1 - حمله جاری شدن سیل از طریق گرسنگی پهنای باند: در حمله جاری شدن سیل، که می تواند باعث جلوگیری از سرویس (DoS) شود، مقدار زیادی از درخواست های مزخرف به یک خدمات خاص برای ممانعت از عملکرد درست آن ارسال می شوند. در محاسبات ابری، دو نوع اساسی [34] از حملات سیل وجود دارند:

• DOS مستقیم - هدف حمله تعیین می شود، و در دسترس بودن سرویس ابری هدف به طور کامل از دست خواهد رفت.

• DOS غیر مستقیم – معنا آن, دو قسم است: 1) همه خدمات میزبانی شده در دستگاه فیزیکی با قربانی هدف تحت تاثیر قرار خواهند گرفت. 2) حمله بدون یک هدف خاص آغاز می شود.



شکل 4. معماری شبکه مرکز داده های سنتی

همچنین نویسندگان در [34] اشاره می کنند که یکی از پیامدهای یک حمله سیل اینست که اگر یک خدمات ابر معین در دسترس نباشد و یا کیفیت خدمات تنزل یابد، مشترکین تمام خدمات تحت تاثیر قرار گرفته نیاز به ادامه پرداخت قبض دارند. با این حال، ما استدلال می کنیم که چون ارائه دهندگان ابر باید قبلاً یک توافقنامه سطح خدمات را (SLA) با مشتریان خود امضا کرده باشند، یک حزب مسئول باید زمانی تعیین شود که سطح خدمات تنزل یافته است, زیرا مشتریان از آن تنزل آگاه خواهند. ما از این مشکل (به عنوان مثال، مسئولیت پذیری ابر) را در بخش بعدی بیشتر توضیح خواهیم داد.

ماهیت تأمین ضعیف و باز بودن عمومی در یک سیستم ابر, آسیب پذیری جدیدی را به همراه می آورد که می تواند برای انجام یک حمله جدید DOS به منظور به خطر انداختن سرویس ابر ارائه شده توسط اشباع شدن پهنای باند شبکه محدود مورد سوء استفاده قرار گیرد. همانطور که در شکل 4 نشان داده شده است، لینک های A، B، C, به ترتیب لینک های بالای روتر R5، R1 و R2 هستند. فرض کنید که لینک B, لینک فعال است و لینک C, لینک خراب می باشد (به عنوان مثال، یک لینک زمانی فعال خواهد شد که لینک فعال, خاموش باشد). با توجه به تأمین ضعیف, ظرفیت کل H1، H2، H3، H4 و (که به صورت زیر شبکه 1 است) چند برابر بزرگتر از هر ظرفیت برای لینک های A، B، یا C است. به منظور اشباع لینک B، حمله کنندگان (که ممکن است چند میزبان کنترل شده توسط دشمن باشند) در زیر شبکه 1 تنها به تولید ترافیک کافی برای هدف قرار دادن میزبان در زیر شبکه دیگر نیاز دارند (مثلا زیرشبکه 2). هنگامی که لینک B توسط ترافیک غیر حسی اشباع می شود، میزبان ها در زیرشبکه 1 قادر به ارائه خدمات به کاربران ابر می باشند.

برای شروع چنین حمله DOS (گرسنگی پهنای باند) به طور موثر، چند مرحله وجود دارد:

1) شناسایی توپولوژی - از آنجا که تنها میزبان ها در زیرشبکه های مختلف توسط لینک های تنگنا متصل می شوند, دشمن اول باید توپولوژی شبکه را شناسایی نماید. با بهره برداری ماهیت چندگانه از یک روتر، تعداد روترها بین دو میزبان را می توان تعیین نمود. این کار کمک می کند تا میزبان های انتخاب شده, تصویری از توپولوژی داشته باشند.

2) به دست آوردن دسترسی به میزبان های کافی – تعداد میزبان ها برای انجام این حمله توسط ظرفیت لینک بالایی تعیین می شود که می توان آن را توسط برخی از ابزارها مانند [26] Pathload، 27] Nettimer]، و یا 25] Bprobe] برآورد نمود.

3) انجام حمله – نویسنده, استفاده از ترافیک UDP را پیشنهاد می دهد, زیرا جلسات دیگر TCP را گرسنه می سازد.

2) T4.1 - حمله مصرف منابع متقلبانه (FRC): حمله انکار اقتصادی پایداری (EDoS) [96] FRC، 97]] یک حمله ظریف است که ممکن است طی یک دوره طولانی (معمولا برای هفته ها طول می کشد) به منظور اثربخشی انجام شود. در محاسبات ابری، هدف از حمله FRC, محروم نمودن قربانی (به عنوان مثال، مشتریان منظم ابر) از در دسترس بودن اقتصادی بلند مدت خود به مطالب وب سایت به میزبانی وبی است که در دسترس عموم می باشد. به عبارت دیگر، مهاجمانی که به عنوان مشتریان خدمات ابر حقوقی عمل می کنند، به طور مداوم درخواست هایی را به وب سایت میزبانی وب در سرورهای ابری برای مصرف پهنای باند ارسال می کنند که برای مشتری ابر دارای وب سایت, صورتحساب صادر می شد؛ و تشخیص ترافیک FRC از دیگر ترافیک های مشروع مشکل است. یک حمله FRC زمانی موفق می شود که موجب بار مالی بر قربانی شود.

B. استراتژی دفاع

1) D3.1.1 - دفاع از حمله جدید DOS: این نوع جدید حمله DOS از DOS سنتی و یا DDOS حملات [24] که در آن DOS سنتی, ترافیک را به نرم افزار / میزبان هدف به طور مستقیم می فرستد در حالی که حمله جدید DOS این کار را نمی کند, متفاوت است؛ بنابراین، برخی از تکنیک ها و اقدامات متقابل [21]، [22] برای هدایت DOSهای سنتی, دیگر قابل اجرا نیستند.

یک راهبرد اجتناب DOS به نام مهاجرت خدمات [20] برای مقابله با این حمله جاری شدن سیل جدید توسعه داده شده است. عامل نظارتی واقع در خارج از ابر, برای تشخیص این مورد تنظیم می شود که آیا ممکن است گرسنگی پهنای باند توسط کاوش مداوم برنامه های کاربردی ابر وجود داشته باشد یا خیر. هنگامی که تخریب پهنای باند

شناسایی می شود، عامل نظارتی, مهاجرت نرم افزار را انجام خواهد داد که ممکن است به طور موقت خدمات را متوقف نماید و آن را بعداً از سرگیری نماید. مهاجرت, برنامه فعلی را به زیر شبکه دیگر حرکت می دهد که مهاجم از آن بی اطلاع است نتایج آزمایش نشان می دهند که تنها چند ثانیه طول می کشد تا مهاجرت یک برنامه تحت وب بدون تابعیت از یک زیر شبکه به دیگری صورت پذیرد.

2) D4.1.1 - تشخیص حمله FRC: کلید تشخیص FRC, تشخیص ترافیک FRC از ترافیک فعالیت طبیعی است. Idziorek و همکاران بهره برداری از ثبات و خود-تشابه فعالیت وب تجمعی را پیشنهاد نموده اند [96]. برای رسیدن به این هدف، سه معیار تشخیص استفاده می شوند: i) قانون Zipf برای [97] برای اندازه گیری فرکانس نسبی و خود شباهت محبوبیت صفحات وب اتخاذ می شود؛ ii) قانون پای Spearman برای پیدا کردن نزدیکی بین دو لیست رتبه بندی شده استفاده می شود که امتیاز شباهت را تعیین می کند؛ ج) همپوشانی بین لیست مرجع و لیست مقایسه کننده, شباهت بین داده های آموزش و داده های آزمون را اندازه گیری می کند. ترکیبی از سه معیار, یک راه قابل اعتماد تشخیص FRC را ارائه می دهند.

C. خلاصه و مسائل باز

جمع و جور کردن سرویس را می تواند به تهدیدات داخلی و خارجی منجر شود. یک تهدید داخلی از مشتریان مخرب ابر می آید که از ویژگی تامین ضعیف پهنای باند در معماری فعلی DCN برای گرسنه کردن ترافیک خدمات مشروع استفاده می کنند. از سوی دیگر، تهدید خارجی به حمله EDoS اشاره می کند که موجب تنزل دردسترس بودن اقتصادی دراز مدت قربانی می شود. به نظر می رسد هر دوی Doss و EDos در حالات دیگر ظاهر می شوند، با این حال، روش هایی که آنها برای حمله به پلت فرم ابر استفاده می کنند, جدید هستند و ارزش بررسی دارند. موضوعات زیر می توانند, جهات تحقیقاتی آینده باشند:

• D3.1.1 یک راهبرد عدم مقابله را ارائه می دهد که مهاجرت خدمات را اتخاذ می کند. اجتناب DoS با این حال برای دفاع کامل از این حمله کافی نیست، چرا که دشمنان هنوز مشخص نشده اند. این موضوع را می توان بیشتر

توسط مسئولیت پذیری بررسی نمود. در این مورد، برای پیگیری رفتار مخرب، شناسایی منشاء غیر حسی ترافیک که سعی در اشباع لینک اتصال دارد مهم است.

• D4.1.1 یک مکانیزم را برای تشخیص FRC توصیف می کند. با این حال، روشن نیست که چگونه یک قربانی به حمله واکنش نشان می دهد و شناسایی مهاجمان به خوبی ارائه نمی شود. برای تکمیل D4.1.1، تحقیقات جدید در این حوزه انتظار می روند.

• حمله FRC توسط مصرف پهنای باند انجام می شوند، که یکی از منابعی است که قابل پرداخت می باشد. با این حال، منابع دیگر، از جمله قابلیت های محاسبات و ذخیره سازی، به طور بالقوه به حمله EDoS آسیب پذیر هستند. بنابراین، کشف مدل های تهدید عملی و استراتژی های دفاعی به سمت مطالعه جامع از حمله EDoS ضروری است.

V. مسئولیت پذیری ابر

در حالی که پاشخگویی در سیستم های دیگر مورد مطالعه قرار گرفته است [124]، [125]، [126]، [127]، [128]، [129]، [130]، [131]، ایجاد روابط اعتماد در محیط ابری ضروری است [35]، [47]، [48]، [50]، [52]، [88]. مسئولیت پذیری به معنی اینست که قابلیت شناسایی یک حزب، با شواهد غیر قابل انکار، مسئول رویدادهای خاص [124] ، [125]، [126]،[127]، [128]، [129]، [130]، [131]. در هنگام مقابله با محاسبات ابر، احزاب متعددی وجود دارند که ممکن است درگیر شوند؛ یک ارائه دهنده ابر و مشتریان آن, دو پایه اساسی هستند، و مشتریان عمومی که از برنامه های کاربردی استفاده می کنند (به عنوان مثال، یک نرم افزار وب) برون سپاری شده توسط مشتریان ابر می توانند طرف دیگر باشند. با این حال، هویت ریز دانه ممکن است برای شناسایی یک دستگاه خاص و یا حتی برنامه معیوب / مخربی که مسئول است, به کار گرفت.

A. تهدیدات برای مسئولیت پذیری ابر

1) T2.2 - نقض SLA: A . Haeberlen به اهمیت مسئولیت پذیری در محاسبات ابری [47] می پردازد که در آن از دست دادن کنترل داده ها زمانی مشکل ساز می شود که چیزی از دست می رود. به عنوان مثال، مشکلات زیر ممکن است احتمالا بوجود آیند: 1) دستگاه ها در ابر ممکن است به اشتباه پیکربندی شوند یا معیوب باشند و در

نتیجه می توانند داده های مشتری را خراب کنند یا موجب محاسبه او برای بازگشت نتایج نادرست شوند؛ 2) ارائه دهنده ابر به طور تصادفی می تواند منابع کافی برای مشتری را تخصیص دهد, عملی که می تواند عملکرد خدمات مشتری را کاهش دهد و سپس SLA را نقض کند؛ 3) یک مهاجم می تواند یک ایراد را در نرم افزار مشتری به منظور سرقت اطلاعات با ارزش و یا تسلط بر دستگاه های مشتری برای هرزه نگاره و یا حملات داس جاسازی نماید. 4) مشتری نمی تواند دسترسی به داده های خود دسترسی داشته باشد, زیرا ابر آن را از دست داده است و یا به سادگی به خاطر اینکه داده ها در زمان نامناسب دردسترس نیسند. اگر چیزی اشتباه شود، برای مثال، نشت داده ها به رقیب، یا محاسبه, نتایج نادرست را بازگرداند؛ برای یک مشتری و ارائه دهنده دشوار است که تعیین کنند کدامیک از آنها موجب این مشکل شده اند، و، در صورت عدم وجود شواهد محکم، یکدیگر را مسئول مشکل مناقشه می دانند.

2) T2.3 - *MapReduce* متقلب: *MapReduce* [53] یک الگوی محاسبات موازی است که به طور گسترده ای توسط ارائه دهندگان ابر عمده به کار گرفته می شود (گوگل، یاهو، فیس بوک، و غیره). *MapReduce* یک مجموعه داده های بزرگ را به بلوک های متعدد تجزیه می کند که هر یک از آنها متعاقبا ورودی را به یک دستگاه کارگر تک برای پردازش وارد می کنند. با این حال، دستگاه های کاری ممکن است به اشتباه پیکربندی شوند و یا مخرب باشند، در نتیجه، نتایج پردازش بازگردانده شده توسط ابر ممکن است نادرست باشد. علاوه بر این، بررسی صحت نتایج به غیر از اجرای دوباره همان وظیفه به صورت محلی برای مشتریان دشوار است. MapReduce نادرست را می توان به عنوان یک مورد محکم از مشکل صداقت محاسبات مشاهده نمود، همانطور که ما در بخش III بحث نمودیم (به عنوان مثال، صداقت ابر). این موضوع بیشتر توسط مسئولیت پذیری بررسی خواهد شد، به دلیل اینکه حتی پس از اینکه مشتریان صحت خروجی MapReduce را تایید کرده اند، هنوز هم یک ضرورت برای شناسایی دستگاه های معیوب و یا هر دلیل ممکن دیگر که منجر به پاسخ های اشتباه می شود وجود دارد.

3) T2.4 - هویت پنهان دشمنان: با توجه به نگرانی های حریم خصوصی، ارائه دهندگان ابر نباید به مشتریان ابر, اطلاعات مربوط به هویت را فاش کنند. دسترسی ناشناس برای مقابله با این موضوع استفاده می وشد؛ اگرچه

ناشناس بودن, حریم خصوصی را افزایش می دهد، مشکلات امنیتی را نیز به وجود می آورد. ناشناس ماندن کامل نیاز دارد تا اطلاعات مشتری به طور کامل از هر کسی یا هر چیز دیگری پنهان بمانند. در این مورد، کاربران مخرب می توانند صداقت داده ها را بدون شناسایی شدن به خطر اندازند, زیرا مخفی کردن هویت آنها آسان تر می شود.

4) T4.2 – صورتحساب دهی نادرست مصرف منابع: مدل پرداخت به محض وصل شدن, مشتریان را قادر می سازد تا تصمیم بگیرند که چگونه کسب و کار خود را بر اساس نیازهای خود و همچنین شرایط مالی برون سپاری نمایند. با این حال، بررسی هزینه های مصرف منابع با توجه به جعبه سیاه و ماهیت پویای محاسبه ابر, برای مشتریان بسیار مشکل است. از دیدگاه فروشنده ابر، به منظور دستیابی به حداکثر سود دهی، ارائه دهندگان ابر, برنامه های کاربردی متعدد متعلق به مشتریان مختلف را برای حفظ بهره برداری بالا انتخاب می کنند. چندگانه کردن ممکن است موجب شود تا ارائه دهندگان به اشتباه مصرف منابع را به مشتریان نسبت دهند و یا به طور ضمنی متحمل هزینه های اضافی شوند، بنابراین اثربخشی آنها کاهش می یابد [18]. به عنوان مثال، زمان I / O و پهنای باند شبکه های داخلی اندازه گیری نمی شوند، حتی اگر هر یک متحمل هزینه غیربدیهی شوند. علاوه بر این، سنجش اثرات به اشتراک گذاری ، مانند استفاده به اشتراک گذاشته شده از حافظه، دشوار است.

B. استراتژی های دفاع

1) D2.2.1 - مسئولیت پذیری در توافقنامه سطح خدمات (SLA): برای مقابله با این اختلاف از نقض SLA، یک AUDIT ابتدایی (A، S، T1، T2) در [47] پیشنهاد شده است که اجازه می دهد تا مشتریان بررسی کنند که آیا ارائه دهنده ابر, SLA (مشخص شده توسط A) را برای خدمات S بین زمان های داخلی t1 و t2 برآورده ساخته است یا خیر. AUDIT در صورتی اوکی خواهد داد که هیچ خطایی شناسایی نشود؛ در غیر این صورت AUDIT, شواهد قابل افشا را برای طرف مسئول را ارائه خواهد کرد. نویسنده در [47], جزئیات AUDIT را بیان نمی کند، به جای آن, نویسنده مجموعه ای از بلوک های ساختمانی را فراهم می کند که ممکن است مؤثر باشند، از جمله 1) سیاههای مربوط به رشوه دادن آشکار که می تواند تمام اعمال سابقه یک برنامه را ضبط کند, 2) تکرارهای مبتنی بر مجازی سازی که می تواند اقدامات برنامه های کاربردی دیگر را با پخش مجدد سیاههای مربوطه ممیزی نمایند

3) مهر زنی مدت زمان مورد اعتماد که می تواند برای شناسایی خطای عملکرد مورد استفاده قرار گیرد (یعنی، زمان تاخیر و یا توان نمی تواند با سطح در SLA مطابقت داشته باشند)، و 4) نمونه برداری که می تواند تضمین های احتمال را ارائه می کند و می تواند بهره وری پخش را بهبود بخشد.

2) D2.2.2 - دستگاه مجازی مسئولیت پذیر: دستگاه مجازی مسئولیت پذیر (50] (AVM], کار پیگیرانه [47] است. هدف از AVM, توانمندسازی کاربران برای ممیزی اجرای نرم افزار روی دستگاه های از راه دور است. AVM قادر به 1) تشخیص خطا ، 2) شناسایی گره معیوب، 3) فراهم نمودن شواهد قابل اثبات از یک خطای خاص و اشاره به مسئول است. AVM قابل اجرا برای رایانش ابری است که در آن مشتریان داده ها و نرم افزار خود را به سرورهای ابری بی اعتماد برون سپاری می نماید. AVM اجازه می دهد تا کاربران ابر, صحت کد خود را در سیستم ابر بررسی نمایند. روش این کار, قرار دادن هر نرم افزار در حال اجرا در یک دستگاه مجازی است که ورود رشوه دادن آشکار [51] را برای ثبت اجرای کامل نرم افزار نگه می دارد. اگر ما فرض کنیم که یک مرجع پیاده سازی وجود دارد، که اجرای صحیح نرم افزار را تعریف می کند، توسط تکرار ورود به سیستم فایل و مقایسه آن با کپی مرجع, کاربران ابر دارای اطلاعات کافی به منظور بررسی صحت نرم افزار هستند. اگر تناقضی وجود داشته باشد، عدم تطابقاتی تشخیص داده خواهند شد. ورود به سیستم, رشوه دادن-آشکار، به این معنی که هیچ کس نمی تواند برای ورود به سیستم فایل بدون شناسایی شدن رشوه دهد. هنگامی که یکپارچگی ورود به سیستم فایل تضمین باشد، شواهد به دست آمده از آن, قابل اعتماد است. شواهد توسط هر شخص خارجی قابل اثبات هستند. یکی از محدودیت های AVM اینست که تنها می تواند خطاهای ناشی از عملیات های شبکه را تشخیص دهد زیرا فقط پیام های ورودی / خروجی شبکه را گزارش می دهد.

3) D2.2.3 - مانیتورینگ مشارکتی: یک راه حل که مشابه با AVM است در [48] توسط حفظ یک دستگاه حالت خارجی توسعه داده شد که کار آن اعتبارسنجی صحت داده ها و اجرای منطق کسب و کار در یک محیط چند اجاره است. نویسندگان [48] نقطه پایانی خدمات را به عنوان یک واسطه تعریف نمودند که از طریق آن خدمات ابر به کاربران نهایی تحویل داده می شوند. فرض بر این است که داده ها ممکن است تنها از طریق نقطه پایانی در دسترس

قرار گیرند که با توجه به SLA بین ارائه دهنده ابر و کاربران مشخص می شوند. ایده اصلی, بسته بندی کردن هر نقطه پایانی با یک آداپتور است که قادر به ضبط ورودی / خروجی نقطه پایانی و ثبت تمام عملیات های انجام شده از طریق نقطه پایانی است. پس از آن ورود به سیستم به دستگاه حالت خارجی برای اهداف احراز هویت ارسال می شود. برای تأیید اعتبار صحت، درخت B [49 Merkle B] برای تصدیق داده هایی به کار گرفته می شود که در سیستم ابر ذخیره می شوند. یک عملیات به روز رسانی بر روی داده ها, درخت-MB را نیز نیز به روز رسانی خواهد کرد. یک عملیات پرس و جو توسط یک پرس و جوی وسیع روی درخت-MB تصدیق می شود. هنگامی که چک کردن صحت خراب می شود، دستگاه حالت, مشکلات را گزارش و شواهد را بر اساس نتیجه پرس و جو از MB-درخت تایید می کند.

4) D2.3.1 - *MapReduce:* مسئولیت پذیر: در [54]، این مشکل با SecureMR بررسی شده است که تکرار وظیفه کامل را برای دوبل نمودن کنترل نتیجه پردازش اتخاذ می کند. SecureMR مستلزم آن است که دو دستگاه متفاوت دو برابر شوند که زمان پردازش کلی, اجرای یک کار را دو برابر می کند. علاوه بر این، SecureMR از یک مثبت کاذب رنج می برد هنگامی که یک برنامه معیوب, وظایف تکراری را پردازش می کند.

Xiao و همکاران [66] برای ساخت یک MapReduce مسئولیت پذیر برای تشخیص گره های بدخیم پیشنهاد نمودند. ایده اصلی به صورت زیر است: ارائه دهنده ابر یک دامنه اعتماد را ایجاد می کند که شامل چند دستگاه کارگر منظم به نام ممیزان می شوند. یک ممیز, از جبرگرایی توابع Map/Reduce به منظور اعمال یک آزمون مسئولیت پذیر (A-تست) برای هر وظیفه در هر دستگاه کاری استفاده می کند. AC یک وظیفه را بر میدارد که توسط یک دستگاه M تکمیل شده است, سپس دوباره آن را اجرا می کند و خروجی را با خروجی دستگاه M مقایسه می کند. اگر یک تناقض دیده شود، پس از آن ثابت می شود که M مضر است یا خیر. یک آزمون زمانی متوقف خواهد شد که تمام وظایف آزمایش شوند. یک تکرار کامل از یک اجرا به هزینه های محاسبه بزرگ نیاز دارد. به جای دنبال نمودن یک نرخ تشخیص 100٪، نویسندگان مصمم به ارائه تضمین های احتمال به منظور سرعت

بخشیدن به آزمون-A هستند. در این لحظه، ایده کلی, فقط اجرای بخشی از هر وظیفه است. با انتخاب دقیق پارامترها، نرخ تشخیص بالا (به عنوان مثال، 99٪) با هزینه های محاسبه کم به دست می آید.

5) D2.4.1 - منشأ امن: منشأ امن با یک هدف معرفی می شود تا اطمینان حاصل شود که شواهد قابل اثبات را می توان برای ردیابی مالک داده های واقعی و سوابق تغییر داده ارائه نمود. منشأ امن برای بهبود پزشکی قانونی داده ها و مسئولیت پذیری در سیستم ابری ضروری است. Lu و همکارانش [41] یک طرح منشأ امن را بر اساس تکنیک های دارای دو خط مستقیم بریده پیشنهاد کرده اند که اولی مشکلات منشأ در محاسبات ابری را به دنبال می آورد. با توجه به یک فایل ذخیره شده در ابر، زمانی که اختلاف در آن فایل وجود دارد، ابر می تواند تمام اطلاعات منشأ را با توانایی رسم تمام نسخه های فایل و کاربرانی که آن را تغییر داده اند ارائه دهد. با استفاده از این اطلاعات، یک هویت کاربر خاص را می توان ردیابی نمود.

6) D4.2.1 – حسابداری منبع قابل اثبات: Sekar و Maniatis [18] حسابداری منابع قابل اثبات را پیشنهاد کرده اند، که مشتریان را قادر می سازد تا مطمئن شوند که i) برنامه های کاربردی آنها در واقع منابعی را مصرف نموده اند که برایش پول پرداخته اند و ii) مصرف بر اساس سیاست توافق شده توجیه می شد. این طرح در [18] سه نقش را در نظر می گیرد: مشتری C، ارائه دهنده P، و تصدیق کننده V. اول، C از P می خواهد تا وظیفه T را اجرا کند؛ پس از آن، P یک گزارش R را تولید می کند که توصیف می کند که P فکر می کند که C چه منابعی را مصرف می کند. پس از آن, C گزارش را به R می فرستد و برخی از داده های اضافی را به V می فرستد چک کند که آیا R, یک گزارش مصرف معتبر است. با پیاده سازی یک لایه سخت افزار مورد اعتماد با دیگر فن آوری های موجود از قبیل نظارت بر تخلیه، نمونه برداری، و عکس فوری، می توان تضمین کرد که a) ارائه دهنده, مشتریان را بیش از حد و کمتر از حد مطالبه نمی کند b) ارائه دهنده به درستی یک منبع را به مدیر مسئول برای استفاده از آن منبع مصرف می کند.

7) سایر نظرات: به منظور گنجاندن مسئولیت پذیری عملی به محیط ابر، Ko و همکاران [69], چرخه عمر مسئولیت پذیری ابر (CALC)، توصیف فازهای کلیدی برای ساخت مسئولیت پذیری ابر ارائه نمودند. CalC شامل

هفت مرحله می شود: 1) برنامه ریزی سیاست، 2) حس و ردیابی، 3) ورود به سیستم، 4) امن نگه داشتن سیاههای مربوط، 5) گزارش و پخش دوباره، 6) ممیزی، و 7) بهینه سازی و اصلاح.

C. خلاصه و مسائل باز

مسئولیت پذیری یک ویژگی قابل توجه محاسبات ابری است زیرا الگوی رایانش, دشواری حفظ یک نهاد مسئول برای برخی از اقدامات را افزایش می دهد. پیرو یک مدل صدور صورت حساب به محض ورود، فروشنده ابر, منابع اجاره شده توسط مشتریان را فراهم می کند که میزبان مطالب وب است که برای کاربران عمومی باز است. حتی یک عمل ساده (به عنوان مثال، یک درخواست وب) خواهد احزاب متعدد باشد. از سوی دیگر، مسئولیت پذیری نه تنها موجب رفع و رجوع تهدیدات می شود, بلکه با حوادث مختلف، از جمله اشکال نرم افزار، اشتباه پیکربندی، و خرابی سخت افزاری، برای کمک به شناسایی منشاء رویداد سرو کار دارد. بنابراین، یک ابر مسئولیت پذیر, یک گام بزرگ به سوی یک ابر قابل اعتماد است.

چند مسائل باز برای تحقیقات آینده وجود دارند:

• MapReduce مسئولیت پذیر, مسئولیت پذیری برای MapReduce را با ارائه شواهد قابل اثبات اشاره کننده به گره معیوب / مسئول میسر می سازد, زمانی که هر دو تابع Map یا تابع Reduce, صادقانه انجام نشده باشند. با این حال، راه حل های حال حاضر بر اساس تکرار هستند، که دارای دو اشکال است: a) تکرار، موثر نیست، و فروشندگان ابر دارای مشوق برای انجام این کار نیستند؛ نمونه برداری می تواند بهره وری را بهبود بخشد اما دقت را کاهش می دهد، که برای مشتریان مهم تر است؛ b) تکرار تنها در صورتی موثر است که فروشنده ابر از مدل نیمه اعتماد است، که ممکن است در دنیای واقعی رخ ندهد. بنابراین، برای طراحی طرح های جدید ضروری است که به کارایی و دقت دستیابی پیدا کنند.

• برای پیاده سازی مسئولیت پذیری استفاده منابع، D4.2.1 متکی در فروشندگان ابر برای تولید یک گزارش مصرف است که توانایی تایید توسط شخص ثالث را دارند. یکی دیگر از ایده ها, کاهش حمایت ارائه دهنده ابر است. دو گزینه ممکن وجود دارد [18]. اولین گزینه, پیش بینی منابع است، که توسط استخراج سیاههای مربوطه اجرا,

مشتریان را قادر به پیش بینی حجم کاری می سازد و پس از آن نتیجه را با گزارش استفاده ارائه دهنده مقایسه می کند. نگرانی اصلی پیش بینی منبع اینست که به اندازه کافی دقیق نیست. یکی دیگر از گزینه ها, توانمندسازی مشتریان متعدد به شناسایی نقض های مشترک است. با این حال، حفظ حریم خصوصی مشتریان اگر این دو با هم کار کنند, سخت است.

VI. حریم خصوصی ابر

در عین حال حریم خصوصی, یکی دیگر از نگرانی های مهم با توجه به ابر محاسبات است, با توجه به این واقعیت که داده ها و منطق کسب و کار مشتریان در میان سرورهای ابری اعتماد برقرار است که متعلق به ارائه دهنده ابر است و توسط آن نگهداشته می شود. بنابراین، خطرات بالقوه ای وجود دارند که اطلاعات محرمانه (به عنوان مثال، اطلاعات مالی، پرونده سلامت) و یا اطلاعات شخصی (مانند نمایه شخصی) برای رقبای عمومی یا کسب و کار افشا شوند. حریم خصوصی یک از موضوعات با بالاترین اولویت [132]، [133]، [134] است.

در سراسر این متن، ما قابلیت حفاظت از حریم خصوصی را به عنوان ویژگی اصلی حریم خصوصی در نظر می گیریم. ویژگی های امنیتی معدد به طور مستقیم یا به طور غیر مستقیم حفاظت از حریم خصوصی, جمله محرمانه بودن، صداقت، مسئولیت پذیری، و غیره را تحت تاثیر قرار می دهند. بدیهی است، به منظور حفظ اطلاعات خصوصی از فاش، محرمانه بودن ضروری می شود، و صداقت تضمین می کند که داده ها / محاسبات خراب نمی شوند، که به نحوی حریم خصوصی را حفظ می کند. مسئولیت پذیری، در مقابل، می تواند حریم خصوصی را با توجه به این واقعیت تضعیف کند که روش های دستیابی به دو صفت معمولا متناقض هستند. جزئیات بیشتر در این بخش داده شده است.

A. تهدیدات برای حریم ابر

در بعضی جهات، حفظ حریم خصوصی, یک فرم سختگیرانه تر از محرمانه بودن است، با توجه به این مفهوم, آنها هر دو از نشت اطلاعات جلوگیری می کنند. بنابراین، اگر محرمانه بودن ابر نقض شود, حفاظت از حریم خصوصی نیز

نقض خواهد شد. مشابه با دیگر سرویس های امنیتی، معنی حریم خصوصی ابردو قسم است: حریم خصوصی اطلاعات و حریم خصوصی محاسبات (T2.5).

B. استراتژی های دفاع

Chow و همکاران [59], رویکردهای حفظ حریم خصوصی را نزدیک به سه دسته طبقه بندی کرده اند، که در جدول II نشان داده شده است.

Gentry رمزگذاری به طور کامل هومومورفیک (D2.5.1،FHE) را برای حفظ حریم خصوصی در محاسبات ابری [64] پیشنهاد نمودند. FHE, محاسبات روی داده های رمزگذاری شده را میسر می سازد، که در سرورهای بی اعتماد ارائه دهنده ابر ذخیره می شود. داده ها را می توان بدون رمزگشایی پردازش نمود. سرورهای ابر هیچ دانش یا دانش کمی در مورد داده های ورودی, تابع پردازش, نتیجه و هر مقدار نتیجه فوری دارند. بنابراین, محاسبه برون سپاری شده تحت پوشش ها در یک روش حفظ حریم خصوصی کامل رخ می دهد. FHE به یک ابزار قوی برای اجرای حریم خصوصی در محاسبه ابر تبدیل شده است. هرچند, تمام طرح های شناخته شده FHE نیز برای استفاده در عمل ناکافی هستند. در حالیکه محققان در حال تلاش برای کاهش پیچیدگی FHE هستند, در نظر گرفتن کاهش قدرت FHE برای دستیابی دباره به بازده ارزشمند است. Naehrig و همکاران تا حدودی رمزگذاری هومومورفیک را پیشنهاد داده اند [98] که تنها از تعدادی از عملیات های هومومورفیک پشتیبانی می کند که بسیار سریع تر و فشرده تر از FHE است.

Pearson et al. ([42] and [43]) مدیر حریم خصوصی (D2.5.2) را پیشنهاد دادند که متکی بر تکنیک های مبهم نمودن است. مدیر حریم خصوصی, خدمت مبهم سازی و ضد مبهم سازی را برای کاهش مقدار اطلاعات حساس ذخیره شده در ابر را فراهم می کند. ایده اصلی, فقط ذخیره شکل رمزگذاری شده از داده های خصوصی کاربران در انتهای ابر است. فرآیند داده ها به طور مستقیم روی داده های رمزگذاری شده انجام می شود. یک محدودیت اینست که فروشندگان بار ممکن است تمایلی به پیاده سازی خدمات اضافی برای حفظ حریم خصوصی نداشته باشند. بدون همکاری فراهم کننده, این طرح کار نخواهد کرد.

Squicciarini et al. [31] یک موضوع حریم خصوصی جدید را بررسی نموده اند که از شاخص گذاری داده ها ایجاد می شود. به منظور مقابله با شاخص گذاری داده ها و جلوگیری از نشت اطلاعات, محققان یک معماری حفاظت از داده های سه لایه را برای ارائه سطوح مختلف از حریم خصوصی برای مشتریان ابر ارائه می دهد.

Itani et al. [44] یک حریم خصوصی-به عنوان-یک-خدمت را ارائه می دهد, به طوری که ذخیره سازی و محاسبه امن داده های خصوصی توسط اعمال توانمند های ضد رشوه در پردازنده های همزمان رمزنگاری میسر شود. که به نوبه خود از داده های مشتری در برابر دسترسی غیرمجاز حفاظت می کند.

Sadeghi et al. [58] استدلال می کنند که راه حل های رمزنگاری خالص بر اساس رمزگذاری قابل تایید و هوموفریک کامل از نهفتگی بالا برای ارائه برون سپاری امن عملی محاسبه به یک فراهم کننده خدمت ابر توزیع شده رنج می برند. آنها ترکیب یک نشانه سخت افزاری مورد اعتماد (D2.5.3) را با ارزیابی تابع امن (SFE) به منظور مقایسه توابع دلخواه روی داده ها پیشنهاد نموده اند, زمانی که هنوز در شکل رمزگذاری شده است. محاسبه, هیچ اطلاعاتی را نشت نمی دهد و قابل تایید است. تمرکز این کار, مینیمم سازی نفهتگی محاسبه برای میسر نمودن برون سپاری امن و کارآمد در محاسبه ابر است. یک نشانه سخت افزاری در برابر حملات فیزیکی ضد-رشوه است. اگر نشانه تحت فرض مورد اعتماد بودن باشد, پردازش داده های کاربر را می توان در نشانه ای انجام داد که به یک سرور ابر بی اعتماد وصل می شود. ویژگی یک نشانه می تواند تضمین کند که محاسبه داده ها, محرمانه و قابل تایید است. راه حل پیشنهادی در [58] تنها باید از یک نشانه ضد-رشوه در فاز پیش پردازش تنظیم استفاده می کند. در فاز انلاین, تنها عملیات های رمزنگاری متقارن در ابر انجام می شوند, بدون اینکه به تعامل بیشتر با نشانه نیاز باشد.

1) عقاید دیگر: رمزنگاری, همه-منظوره **نیست**

Van Dijk et al. [63] استدلال نمودند که رمزنگاری به تنهایی نمی تواند راه حل های کامل برای تمام موضوعات حریم خصوصی در یک محاسبه ابر, حتی با ابزارهای قوی مانند FHE فراهم کند. نویسندگان به طور رسمی, یک رده از مسائل حریم خصوصی را از نظر حالات برنامه های مختلف را تعریف نمودند. اثبات شده است که

زمانی که داده ها در میان مشتریان به اشتراک گذاشته می شوند, هیچ پروتکل رمزنگاری را نمی توان برای ارائه اطمینان از حریم خصوصی پیاده سازی نمود. تعاریف زیر را در نظر بگیری [63]:

- S – ابر که یک مقوله یکنواخت منبع یابی شده است.
- $C = C_1, C_2, \ldots, C_n$ - مجموعه ای از استعدادها/مشتریان S است.
- Xi- یک مقدار خصوصی استاتیک متعلق به Ci است که اما در S ذخیره می شود.
- وظیفه S, اجرای برنامه ها/توابع مختلف روی $\{x_i\}$ است.

a) کلاس یک: محاسبه خصوصی کاربر-تک است

این برنامه ها تنها داده های xi تحت مالکیت توسط یک مشتری Ci را پردازش می کنند. هیچ شخص دیگری نباید قادر به یادگیری هر اطلاعاتی در مورد فرآیند داخلی باشد. یک نمونه نوعی, یک برنامه آماده سازی-مالیات است که داده های مالی ورودی را اتخاذ می کند که متعلق به کاربر تک هستند و باید از هر دوی S و کاربران دیگران پنهان باشند. این کلاس می تواند به طور مناسب توسط یک FHE حل شود که تمام الزامات امن را برآورده می سازد.

b) کلاس دو: محاسبه چند-کاربر خصوصی

این برنامه ها روی مجموعه داده های $\{x_i\}$ تحت مالکیت توسط کاربران متعدد $\{C_i\}$ عمل می کنند. چون کاربران بیشتری دخیل هستند, حریم خصوصی داده ها در میان کاربران به روشی پیچیده تر حفظ می شود. خط مشی های کنترل-دسترسی وجود دارند که باید در زمان پردازش داده ها دنبال شوند. یک برنامه دنیای واقعی, یک سیستم شبکه بندی اجتماعی است که در آن xi, مشخصات شخصی یک کاربر Ci است؛ Ci قادر به مشخص نمودن این مورد است که کدام دوستان می توانند چه بخش ها/توابعی از داده های او را ببینند (یعنی, یک خط مشی کنترل دسترسی را می دهد). اثبات شده است که محاسبه چند-کاربر با استفاده از رمزنگاری غیرقابل دست یافتن است.

c) کلاس سه: محاسبه چند-کاربر خصوصی واضح

این کلاس, یک نسخه محدود از کلاس دو است. تفاوت اینست که خط مشی کنترل-دسترسی روی داده های کاربر, واضح است, بدان معنی که وابسته به تاریخچه اجرای برنامه است. این کلاس به طور کلی در این مقاله بررسی نمی شود [63], اما نویسندگان اعتقاد دارند که موقعیت مهمی در برنامه های ابر دارد.

2) عقاید دیگر: چارچوب های حفظ حریم خصوصی

Lin et al, یک چارچوب حفاظت داده های کلی [56] را به منظور پرداختن به چالش های حریم خصوصی در تدارک خدمات ابر ارائه نمودند. این چارچوب شامل سه بلوک ساختمانی کلیدی می شود: 1) یک راهبرد رتبه بندی خط مشی برای کمک به مشتریان ابر برای شناسایی یک فراهم کننده ابر که به بهترین شکل با الزامات حریم خصوصی تناسب دارد؛ 2) یک مکانیزم تولید خط مشی اتوماتیک که خط مشی ها و الزامات از هر دوی شرکت کنندگان را یکپارچه می کند و یک خط مشی خاص را به صورت توافق شده توسط آنها تولید می کند؛ 3) اجرای خط مشی که تضمین می کند این سیاست برآورده خواهد شد. یک مسیر سرراست به کسب رتبه بندی خط مشی, مقایسه الزام مشتری با خط مشی های چندین فراهم کننده خدمات و پس از آن انتخاب یکی از آنها به بالاترین رتبه است. مقایسه می تواند در سمت کاربر, سمت فراهم کننده ابر یا از طریق یک کارگزار رخ دهد. جبر خط مشی را می توان برای انجام تولید خط مشی به کار گرفته. هر خط مشی باید در ابتدا رسمی و سپس با جبر خط مشی ریزدانه یکپارچه شود [57]. [28] Pearson نشان می دهد که حریم خصوصی باید از ابتدا در نظر گرفته شود و باید در هر فاز طراحی خدمت ابر در نظر گرفته شود.

C. موضوعات باز

با در نظر گرفتن حریم خصوصی ابر, برخی مسائل باز باید در تحقیقات آینده مطالعه شوند:

- نویسندگان فکر می کنند که مسئولیت پذیری و حریم خصوصی می توانند با یکدیگر در تضاد باشند. اجرای مسئولیت پذیری, تا حدودی ناقض حریم خصوصی است و حفاظت شدید از حریم خصوصی (مثلاً گننامی کامل برای پنهان نمودن هویت کاربران), مسئولیت پذیری را چالش برانگیزتر می سازد. یک مثال شدید, یک فایل به اشتراک

گذاشته شده, تحت دسترسی چندین کاربر است که می توانند هویت های خود را به دلیل گمنامی برای هدف حفاظت از حریم خصوصی پنهان کنند. هرچند کاربران خرابکار به دلیل دسترسی ناشناس با مشکل ردیابی می شوند. از نقطه نظر مسئولیت پذیری, رویکردهای کلی شامل ورود اطلاعات, پخش دوباره, ردیابی و غیره [22] می شوند. این عملیات ها را نمی توان بدون نشان دادن برخی اطلاعات خصوصی کامل نمود (مثلاً نام حساب, آدرس IP). ما باید در جستجوی یک موازنه باشیم که در آن, الزام یک سفت را می توان برآورده نمود و در عین حال به طور همزمان درجه صفت دیگر را حفظ کرد.

- ارزیابی صفات, موضوع مهم دیگر است, زیرا یک روش کمی برای ارزیابی آنها را فراهم می کند. هدف, تعیین میزان امن بودن یک ابر یا میزان ارائه حریم خصوصی است. معنی آن دوقسم است: 1) مقایسه رویکردهای امنیتی به طور مثال, برای دستیابی به 100٪ حریم خصوصی, مفید خواهد بود, طرح A, 100 تا می ارزد؛ طرح B می توانند با هزینه 20 به 99٪ مسئولیت پذیری دست یابد. به طور مشخص, طرح B, از نظر عملی کارآمدتر است, هرچند, یک درصد از مسئولیت پذیری را فدا می کند. بدون یک ارزیابی, مقایسه دو راهبرد از نظر کمیتی مشکل است. 2) بندهای کمیتی الزامات حریم خصوصی/امنیت را می توان در توافق نامه های سطح خدمت (SLAها) پیش نویس نمود.

VII. نتیجه گیری ها

در سراسر این مقاله, نویسندگان به طور نظام مند, موضوعات امنیت و حریم خصوصی را در رایانش ابری بر اساس روش شناسی صفت-محور که در شکل 5 نشان داده شده است مطالعه نموده اند. ما بیشتر صفات نماینده امنیت/حریم خصوصی (مثلاً, محرمانگی, صداقت, موجودیت, مسئولیت پذیری و حفاظت از حریم خصوصی) را شناسایی نموده ایم و آسیب پذیری ها را بررسی نموده ایم که می توانند توسط دشمنان به منظور انجام حملات مختلف بهره برداری شوند. راهبردهای دفاعی و پیشنهادات در این مورد به خوبی بررسی شدند. ما اعتقاد داریم که این بازنگری به شکل دادن به جهات تحقیقات آینده در حوزه های امنیت ابر و حریم خصوصی کمک خواهد کرد.

REFERENCES

[1] I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud computing and grid computing 360-degree compared," Grid Computing Environments Workshop, 2008. GCE'08, 2009, pp. 1-10.

[2] J. Geelan. "Twenty one experts define cloud computing," Virtualization, August 2008. Electronic Mag., article available at http://virtualization.sys-con.com/node/612375.

[3] R. Buyya, C. S. Yeo, and S. Venugopal. "Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities," CoRR, (abs/0808.3558), 2008.

[4] Luis M. Vaquero, Luis Rodero-Merino and Daniel Morán, "Locking the sky: a survey on IaaS cloud security," Computing, 2010, DOI:10.1007/s00607-010-0140-x.

[5] Google Docs experienced data breach during March 2009. http://blogs.wsj.com/digits/2009/03/08/1214/

[6] Cloud Security Alliance (CSA). "Security Guidance for Critical Areas of Focus in Cloud Computing V2.1," (Released December 17, 2009). http://www.cloudsecurityalliance.org/guidance/csaguide.v2.1.pdf

[7] Cloud Security Alliance (CSA). "Top Threats to Cloud Computing V 1.0," released March 2010.

[8] The security-as-a-service model. http://cloudsecurity.trendmicro.com /the-security-as-a-service-model/

[9] S.D. Di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, "A data outsourcing architecture combining cryptography and access control," Proc. 2007 ACM workshop on Computer security architecture, 2007, pp. 63-69.

[10] P. Mell and T. Grance. The NIST Definition of Cloud Computing (Draft). [Online] Available: www.nist.gov/itl/cloud/upload/cloud-def-v15.pdf., Jan. 2011.

[11] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," In ACM CCS, pages 598-609, 2007.

[12] A. Juels and B. S. Kaliski, "PORs: Proofs of retrievability for large files," In ACM CCS, pages 584-597, 2007.

[13] Y. Dodis, S. Vadhan, and D. Wichs, "Proofs of retrievability via hardness amplication," In TCC, 2009.

[14] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," SecureComm, 2008.

[15] C. Erway, A. Küpçü, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," Proc. 16th ACM conference on Computer and communications security, 2009, pp. 213-222.

[16] K.D. Bowers, A. Juels, and A. Oprea, "HAIL: A high-availability and integrity layer for cloud storage," Proc. 16th ACM conference on Computer and communications security, 2009, pp. 187-198.

[17] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, "Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds," Proc. 16th ACM conference on Computer and communications security, 2009, pp. 199-212.

[18] V. Sekar and P. Maniatis, "Verifiable resource accounting for cloud computing services," in Proc. 3rd ACM workshop on Cloud computing security workshop, 2011, pp. 21-26.

[19] C. Hoff, "Cloud computing security: From DDoS (distributed denial of service) to EDoS (economic denial of sustainability)," [Online] Available: http://www.rationalsurvivability.com/blog/?p=66., 2008.

[20] H. Liu, "A New Form of DOS Attack in a Cloud and Its Avoidance Mechanism", Cloud Computing Security Workshop 2010.

[21] S. Kandula, D. Katabi, M. Jacob, and A. Berger, "Botz-4-sale: Surviving organized ddos attacks that mimic flash crowds," In Proc. NSDI (2005).

[22] A. Yaar, A. Perrig, and D. Song, "Fit: Fast internet traceback," In Proc. IEEE Infocom (March 2005).

[23] S. Staniford, V. Paxson, and N.Weaver, "How to own the internet in your spare time," In Proc. USENIX Security (2002).

[24] Cisco data center infrastructure 2.5 design guide. http://www.cisco.com/ univercd/cc/td/doc/solution/dcidg21.pdf.

[25] R. Carter, and M. Crovella, "Measuring bottleneck link speed in packet-switched networks," Tech. rep., Performance Evaluation, 1996.

[26] C. Dovrolis, P. Ramanathan, and D. Moore, "What do packet dispersion techniques measure?" In Proc. IEEE INFOCOM (2001), pp. 905-914.

[27] K. Lai, and M. Baker, "Nettimer: a tool for measuring bottleneck link, bandwidth," In USITS'01: Proc. 3rd conference on US ENIX Symposium on Internet Technologies and Systems (Berkeley, CA, USA, 2001), USENIX Association, pp. 11-11.

[28] S. Pearson, "Taking account of privacy when designing cloud computing services," Software Engineering Challenges of Cloud Computing, 2009. CLOUD '09. ICSE Workshop on, 2009, pp. 44-52.

[29] N. Santos, K.P. Gummadi, and R. Rodrigues, "Towards trusted cloud computing," Proc. 2009 conference on Hot topics in cloud computing, 2009.

[30] B. D. Payne, M. Carbone, and W. Lee, "Secure and Flexible Monitoring of Virtual Machines," In Proc. ACSAC'07, 2007.

[31] A. Squicciarini, S. Sundareswaran, and D. Lin, "Preventing Information Leakage from Indexing in the Cloud," 2010 IEEE 3rd International Conference on Cloud Computing, 2010, pp. 188-195.

[32] N. Gruschka, M. Jensen, "Attack Surfaces: A Taxonomy for Attacks on Cloud Services," Cloud Computing, IEEE International Conference on, pp. 276-279, 2010 IEEE 3rd International Conference on Cloud Computing, 2010.

[33] P. Saripalli, B. Walters, "QUIRC: A Quantitative Impact and Risk Assessment Framework for Cloud Security," Cloud Computing, IEEE International Conference on, pp. 280-288, 2010 IEEE 3rd International Conference on Cloud Computing, 2010.

[34] M. Jensen, J. Schwenk, N. Gruschka, and L.L. Iacono, "On technical security issues in cloud computing," Cloud Computing, 2009.

[47] A. Haeberlen. "A Case for the Accountable Cloud," 3rd ACM SIGOPS International Workshop on Large-Scale Distributed Systems and Middleware (LADIS '09), Big Sky, MT, October 2009

[48] C. Wang and Y. Zhou, "A Collaborative Monitoring Mechanism for Making a Multitenant Platform Accountable." Hotcloud 2010.

[49] F. Li, M. Hadjieleftheriou, G. Kollios, and L. Reyzin, "Dynamic authenticated index structures for outsourced databases," Proc. 2006 ACM SIGMOD international conference on Management of data, June, 2006, pp. 27-29.

[50] A. Haeberlen, P. Aditya, R. Rodrigues, and P. Druschel, "Accountable virtual machines," 9th OSDI, 2010.

[51] A. Haeberlen, P. Kuznetsov, and P. Druschel, "PeerReview: Practical accountability for distributed systems," In Proc. ACM Symposium on Operating Systems Principles (SOSP), Oct. 2007.

[52] S. Pearson and A. Charlesworth, "Accountability as a Way Forward for Privacy Protection in the Cloud," Proc. 1st International Conference on Cloud Computing, Beijing, China: Springer-Verlag, 2009, pp. 131-144.

[53] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," In OSDI'04: Proc. 6th conference on Symposium on Opearting Systems Design and Implementation. USENIX Association, Berkeley, CA, USA.

[54] W. Wei, J. Du, T. Yu, and X. Gu, "SecureMR: A Service Integrity Assurance Framework for MapReduce," Proc. 2009 Annual Computer Security Applications Conference, 2009, pp. 73-82.

[55] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," Operating Systems Review, vol. 33, 1998, pp. 173-186.

[56] D. Lin and A. Squicciarini, "Data protection models for service provisioning in the cloud," Proceeding of the 15th ACM symposium on Access control models and technologies, 2010, pp. 183-192.

[57] P. Rao, D. Lin, E. Bertino, N. Li, and J. Lobo, "An algebra for fine-grained integration of xacml policies," In Proc. ACM Symposium on Access Control Models and Technologies, pages 63-72, 2009.

[58] A.R. Sadeghi, T. Schneider, and M. Winandy, "Token-Based Cloud Computing," Trust and Trustworthy Computing, 2010, pp. 417-429

[59] R. Chow, P. Golle, M. Jakobsson, E. Shi, J. Staddon, R. Masuoka, and J. Molina, "Controlling data in the cloud: outsourcing computation without outsourcing control," Proc. 2009 ACM workshop on Cloud computing security, 2009, pp. 85-90.

[60] A. Azimov, S. Hu, R. Ford, and P. Gummadi, "Determining timing

CLOUD'09. IEEE International Conference on, 2009, pp. 109-116.
[35] M. Jensen and J. Schwenk, "The accountability problem of flooding attacks in service-oriented architectures," in Proc. IEEE International Conference on Availability, Reliability and Security (ARES), 2009.
[36] J. Oberheide, E. Cooke, and F. Jahanian, "Cloudav: N-version antivirus in the network cloud," Proc. 17th conference on Security symposium, 2008, pp. 91-106.
[37] F. Lombardi and R. Di Pietro, "Transparent security for cloud," Proc. 2010 ACM Symposium on Applied Computing, 2010, pp. 414-415.
[38] C. Henrich, M. Huber, C. Kempka, J. Müller-Quade, and M. Strefler, "Brief Announcement: Towards Secure Cloud Computing," Stabilization, Safety, and Security of Distributed Systems, 2009, pp. 785-786.
[39] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," J. Network and Computer Applications, vol. 34, Jan. 2011, pp. 1-11.
[40] M. Descher, P. Masser, T. Feilhauer, A. Tjoa, and D. Huemer, "Retaining Data Control to the Client in Infrastructure Clouds," Availability, Reliability and Security, 2009. ARES '09. International Conference on, 2009, pp. 9-16.
[41] R. Lu, X. Lin, X. Liang, and X.S. Shen, "Secure provenance: the essential of bread and butter of data forensics in cloud computing," Proc. 5th ACM Symposium on Information, Computer and Communications Security, 2010, pp. 282-292.
[42] S. Pearson, Y. Shen, and M. Mowbray, "A privacy manager for cloud computing," Cloud Computing, 2009, pp. 90-106.
[43] M. Mowbray and S. Pearson, "A client-based privacy manager for cloud computing," Proc. Fourth International ICST Conference on COMmunication System softWAre and middlewaRE, 2009, pp. 1-8.
[44] W. Itani, A. Kayssi, and A. Chehab, "Privacy as a Service: Privacy-Aware Data Storage and Processing in Cloud Computing Architectures," IEEE International Conference on Dependable, Autonomic and Secure Computing, 2009, pp. 711-716.
[45] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," IEEE INFOCOM 2010, San Diego, CA, March 2010.
[46] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in Proc. ESORICS'09, Saint Malo, France, Sep. 2009.
[60] A. Aviram, S. Hu, B. Ford, and R. Gummadi, "Determinating timing channels in compute clouds," In Proc. 2010 ACM workshop on Cloud computing security workshop (CCSW '10). ACM, New York, NY, USA, 103-108.
[61] A. Lenk, M. Klems, J. Nimis, S. Tai, and T. Sandholm, "What's inside the Cloud? An architectural map of the Cloud landscape," Software Engineering Challenges of Cloud Computing, 2009. CLOUD'09. ICSE Workshop on, 2009, pp. 23-31.
[62] D. L. G. Filho and P. S. L. M. Baretto, "Demon-strating data possession and uncheatable data transfer," IACR ePrint archive, 2006, Report 2006/150, http://eprint.iacr.org/2006/150.
[63] M. Van Dijk and A. Juels, "On the Impossibility of Cryptography Alone for Privacy-Preserving Cloud Computing," IACR ePrint 2010, vol. 305.
[64] C. Gentry, "Fully homomorphic encryption using ideal lattices," In STOC, pages 169-178, 2009.
[65] Cloud computing infrastructure, available: http://en.wikipedia.org/wiki/Cloud_computing.
[66] Z. Xiao and Y. Xiao, "Accountable MapReduce in Cloud Computing," Proc. The IEEE International Workshop on Security in Computers, Networking and Communications (SCNC 2011) in conjunction with IEEE INFOCOM 2011.
[67] M. Christodorescu, R. Sailer, D. Schales, D. Sgandurra, and D. Zamboni, "Cloud security is not (just) virtualization security: a short paper," In Proc. 2009 ACM workshop on Cloud computing security (CCSW '09). ACM, New York, NY, USA, 97-102. DOI=10.1145 /1655008.1655022 http://doi.acm.org/ 10.1145/1655008.1655022.
[68] Y. Deswarte, J.-J. Quisquater, and A. Saidane, "Remote integrity checking," in Proc. Conference on Integrity and Internal Control in Information Systems (IICIS'03), November 2003.
[69] R.K.L. Ko, B.S. Lee, and S. Pearson, "Towards Achieving Accountability, Auditability and Trust in Cloud Computing," Proc. International workshop on Cloud Computing: Architecture, Algorithms and Applications (CloudComp2011), Springer, 2011, pp.5.
[70] B. Grobauer, T. Walloschek, and E. Stocker, "Understanding cloud computing vulnerabilities," Security and Privacy, IEEE, vol. 9, no. 2, pp. 50-57, 2011.
[71] Y. Xu, M. Bailey, F. Jahanian, K. Joshi, M. Hiltunen, and R. Schlichting, "An exploration of L2 cache covert channels in virtualized environments," in Proc. 3rd ACM workshop on Cloud computing security workshop, New York, NY, USA, 2011, pp. 29-40.
[72] K. Okamura and Y. Oyama, "Load-based covert channels between Xen virtual machines," in Proc. 2010 ACM Symposium on Applied Computing, New York, NY, USA, 2010, pp. 173-180.
[73] B. C. Vattikonda, S. Das, and H. Shacham, "Eliminating fine grained timers in Xen," in Proc. 3rd ACM workshop on Cloud computing security workshop, New York, NY, USA, 2011, pp. 41-46.
[74] B. Stone and A. Vance, "Companies slowly join cloud computing," New York Times, 18 April 2010.
[75] Z. Wang and R. B. Lee, "New cache designs for thwarting software cache-based side channel attacks," In 34th International Symposium on Computer Architecture, pages 494-505,June 2007.
[76] Z. Wang and R. B. Lee, "A novel cache architecture with enhanced performance and security," In 41st IEEE/ACM International Symposium on Microarchitecture, pages 83-93, November 2008.
[77] G. Keramidas, A. Antonopoulos, D. N. Serpanos, and S. Kaxiras, "Non deterministic caches: A simple and effective defense against side channel attacks," Design Automation for Embedded Systems, 12(3):221-230, 2008.
[78] J. Kong, O. Aciicmez, J.-P. Seifert, and H. Zhou, "Deconstructing new cache designs for thwarting software cache-based side channel attacks," In 2nd ACM Workshop on Computer Security Architectures, pages 25-34, October 2008.
[79] R. Könighofer, "A fast and cache-timing resistant implementation of the AES," in Proc. 2008 The Cryptopgraphers' Track at the RSA conference on Topics in cryptology, Berlin, Heidelberg, 2008, pp. 187-202.
[80] Yinqian Zhang, A. Juels, A. Oprea, and M. K. Reiter, "HomeAlone: Co-residency Detection in the Cloud via Side-Channel Analysis," in 2011 IEEE Symposium on Security and Privacy (SP), 2011, pp. 313-328.
[81] K. Suzaki, K. Iijima, T. Yagi, and C. Artho, "Memory deduplication as a threat to the guest OS," in Proc. Fourth European Workshop on System Security, New York, NY, USA, 2011, p. 1:1-1:6.
[96] J. Idziorek, M. Tannian, and D. Jacobson, "Detecting fraudulent use of cloud resources," in Proc. 3rd ACM workshop on Cloud computing security workshop, New York, NY, USA, 2011, pp. 61-72.
[97] J. Idziorek and M. Tannian, "Exploiting cloud utility models for profit and ruin," in Cloud Computing (CLOUD), 2011 IEEE International Conference on, 2011, pp. 33-40.
[98] M. Naehrig, K. Lauter, and V. Vaikuntanathan, "Can homomorphic encryption be practical?" in Proc. 3rd ACM workshop on Cloud computing security workshop, New York, NY, USA, 2011, pp. 113-124.
[99] T. Choi, H.B. Acharya, and M. G. Gouda, "Is that you? Authentication in a network without identities," International J. Security and Networks, Vol. 6, No. 4, 2011, pp. 181 - 190.
[100] Q. Chai and G. Gong, "On the (in) security of two Joint Encryption and Error Correction schemes," International J. Security and Networks, Vol. 6, No. 4, 2011, pp. 191 - 200.
[101] S. Tang and W. Li, "An epidemic model with adaptive virus spread control for Wireless Sensor Networks," International J. Security and Networks, Vol. 6, No. 4, 2011, pp. 201 - 210.
[102] G. Luo and K.P. Subbalakshmi, "KL-sense secure image steganography," International J. Security and Networks, Vol. 6, No. 4, 2011, pp. 211 - 225.
[103] W. Chang, J. Wu, and C. C. Tan, "Friendship-based location privacy in Mobile Social Networks," International J. Security and Networks, Vol. 6, No. 4, 2011, pp. 226 - 236.
[104] X. Zhao, L. Li, and G. Xue, "Authenticating strangers in Online Social Networks," International J. Security and Networks, Vol. 6, No. 4, 2011, pp. 237 - 248.
[105] D. Walker and S. Latifi, "Partial Iris Recognition as a Viable Biometric Scheme," International J. Security and Networks, Vol. 6 Nos. 2-3, 2011, pp. 147-152.
[106] A. Desoky, "Edustega: An Education-Centric Steganography Methodology," International J. Security and Networks, Vol. 6 Nos. 2-3, 2011, pp. 153-173.

[82] K. Suzaki, K. Iijima, T. Yagi, and C. Artho, "Software Side Channel Attack on Memory Deduplication," in 23rd ACM Symposium on Operating Systems Principles, poster, 2011.

[83] E. Keller, J. Szefer, J. Rexford, and R. B. Lee, "NoHype: virtualized cloud infrastructure without the virtualization," in Proc. 37th annual international symposium on Computer architecture, New York, NY, USA, 2010, pp. 350-361.

[84] J. Szefer, E. Keller, R. B. Lee, and J. Rexford, "Eliminating the hypervisor attack surface for a more secure cloud," in Proc. 18th ACM conference on Computer and communications security, New York, NY, USA, 2011, pp. 401-412.

[85] C. Wang, K. Ren, J. Wang, "Secure and Practical Outsourcing of Linear Programming in Cloud Computing," In IEEE Trans. Cloud Computing April 10-15, 2011.

[86] A. Baliga, P. Kamat, and L. Iftode, "Lurking in the Shadows: Identifying Systemic Threats to Kernel Data (Short Paper)," in 2007 IEEE Symposium on Security and Privacy, May 2007.

[87] S. Setty, A. Blumberg, M. Walfish, "Toward practical and unconditional verification of remote computations," in the 13th Workshop on Hot Topics in Operating Systems, Napa, CA, USA 2011.

[88] K. Benson, R. Dowsley, and H. Shacham, "Do you know where your cloud files are?," in Proc. 3rd ACM workshop on Cloud computing security workshop, New York, NY, USA, 2011, pp. 73-82.

[89] F. Monrose, P. Wycko, and A. D. Rubin, "Distributed execution with remote audit," In NDSS, 1999.

[90] M. J. Atallah and K. B. Frikken, "Securely outsourcing linear algebra computations," In ASIACCS, 2010.

[91] R. Motwani and P. Raghavan, "Randomized Algorithms," Cambridge University Press, 2007.

[92] M. Blanton, Y. Zhang, and K. B. Frikken, "Secure and Verifiable Outsourcing of Large-Scale Biometric Computations," in Privacy, Security, Risk and Trust (PASSAT), IEEE Third International Conference on Social Computing (SocialCom), 2011, pp. 1185-1191.

[93] S. Goldwasser, S. Micali, and C. Rackoff, "The knowledge complexity of interactive proof systems," SIAM Journal on Comp., 18(1):186-208, 1989.

[94] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy, "Proof verification and the hardness of approximation problems," J. ACM, 45(3):501-555, May 1998.

[95] S. Setty, A. Blumberg, and M. Walfish, "Toward practical and unconditional verification of remote computations," in the 13th Workshop on Hot Topics in Operating Systems, Napa, CA, USA 2011.

[107] N. Ampah, C. Akujuobi, S. Alam, and M. Sadiku, "An intrusion detection technique based on continuous binary communication channels," International J. Security and Networks, Vol. 6 Nos. 2-3, 2011, pp. 174-180.

[108] H. Chen and B. Sun, "Editorial," International J. Security and Networks, Vol. 6 Nos. 2/3, 2011, pp. 65-66.

[109] M. Barua, X. Liang, R. Lu, X. Shen, "ESPAC: Enabling Security and Patient-centric Access Control for eHealth in cloud computing ," International J. Security and Networks, Vol. 6 Nos. 2/3, 2011, pp. 67-76.

[110] N. Jaggi, U. M. Reddy, and R. Bagai, "A Three Dimensional Sender Anonymity Metric," International J. Security and Networks, Vol. 6 Nos. 2/3, 2011, pp. 77-89.

[111] M. J. Sharma and V. C. M. Leung, "Improved IP Multimedia Subsystem Authentication Mechanism for 3G-WLAN Networks," International J. Security and Networks, Vol. 6 Nos. 2/3, 2011, pp. 90-100.

[112] N. Cheng, K. Govindan, and P. Mohapatra, "Rendezvous Based Trust Propagation to Enhance Distributed Network Security," International J. Security and Networks, Vol. 6 Nos. 2/3, 2011, pp. 101-111.

[113] A. Fathy, T. ElBatt, and M. Youssef, "A Source Authentication Scheme Using Network Coding," International J. Security and Networks, Vol. 6 Nos. 2/3, 2011, pp. 112-122.

[114] L. Liu, Y. Xiao, J. Zhang, A. Faulkner, and K. Weber, "Hidden Information in Microsoft Word," International J. Security and Networks, Vol. 6 Nos. 2/3, 2011, pp. 123-135.

[115] S. S.M. Chow and S. Yiu, " Exclusion-Intersection Encryption," International J. Security and Networks, Vol. 6 Nos. 2/3, 2011, pp. 136-146.

[116] M. Yang, J. C.L. Liu, and Y. Tseng, "Editorial," International J. Security and Networks, Vol. 5, No.1 pp. 1 - 3, 2010,

[117] S. Malliga and A. Tamilarasi, "A backpressure technique for filtering spoofed traffic at upstream routers," International J. Security and Networks, Vol. 5, No.1 pp. 3 - 14, 2010.

[118] S. Huang and S. Shieh, "Authentication and secret search mechanisms for RFID-aware wireless sensor networks," International J. Security and Networks, Vol. 5, No.1 pp. 15 - 25 , 2010.

[119] Y. Hsiao and R. Hwang, "An efficient secure data dissemination scheme for grid structure Wireless Sensor Networks," International J. Security and Networks, Vol. 5, No.1 pp. 26 - 34 , 2010.

[120] L. Xu, S. Chen, X. Huang, and Y. Mu, "Bloom filter based secure and anonymous DSR protocol in wireless ad hoc networks," International J. Security and Networks, Vol. 5, No.1 pp. 35 - 44, 2010.

[121] K. Tsai, C. Hsu, and T. Wu, "Mutual anonymity protocol with integrity protection for mobile peer-to-peer networks," International J. Security and Networks, Vol. 5, No.1 pp. 45 - 52, 2010.
[122] M. Yang, "Lightweight authentication protocol for mobile RFID networks," International J. Security and Networks, Vol. 5, No.1 pp. 53 - 62, 2010.
[123] J. Wang and G.L. Smith, "A cross-layer authentication design for secure video transportation in wireless sensor network," International J. Security and Networks, Vol. 5, No.1 pp. 63 - 76, 2010.
[124] Y. Xiao, "Accountability for Wireless LANs, Ad Hoc Networks, and Wireless Mesh Networks," IEEE Commun. Mag., special issue on Security in Mobile Ad Hoc and Sensor Networks, Vol. 46, No. 4, Apr. 2008, pp. 116-126.
[125] Y. Xiao, "Flow-Net Methodology for Accountability in Wireless Networks," IEEE Network, Vol. 23, No. 5, Sept./Oct. 2009, pp. 30-37.
[126] J. Liu, and Y. Xiao, "Temporal Accountability and Anonymity in Medical Sensor Networks," ACM/Springer Mobile Networks and Applications (MONET), Vol. 16, No. 6, pp. 695-712, Dec. 2011.
[127] Y. Xiao, K. Meng, and D. Takahashi, "Accountability using Flow-net: Design, Implementation, and Performance Evaluation," (Wiley Journal of) Security and Communication Networks, Vol.5, No. 1, pp. 29-49, Jan. 2012.
[128] Y. Xiao, S. Yue, B. Fu, and S. Ozdemir, "GlobalView: Building Global View with Log Files in a Distributed / Networked System for Accountability," (Wiley Journal of) Security and Communication Networks, DOI: 10.1002/sec.374.
[129] D. Takahashi, Y. Xiao, and K. Meng, "Virtual Flow-net for Accountability and Forensics of Computer and Network Systems," (Wiley Journal of) Security and Communication Networks, 2011, DOI: 10.1002/sec.407.
[130] Z. Xiao and Y. Xiao, "PeerReview Re-evaluation for Accountability in Distributed Systems or Networks," International J. Security and Networks (IJSN), 2010, Vo. 7. No. 1, in press.
[131] Z. Xiao, N. Kathiresshan, and Y. Xiao, "A Survey of Accountability in Computer Networks and Distributed Systems," (Wiley Journal of) Security and Communication Networks, accepted.
[132] J. Liu, Y. Xiao, S. Li, W. Liang, C. L. P. Chen, "Cyber Security and Privacy Issues in Smart Grids," IEEE Commun. Surveys Tuts., DOI: 10.1109/SURV.2011.122111.00145, in press.
[133] Y. Xiao, X. Shen, B. Sun, and L. Cai, "Security and Privacy in RFID and Applications in Telemedicine," IEEE Commun. Mag., Vol. 44. No. 4, Apr. 2006, pp. 64-72.
[134] H. Chen, Y. Xiao, X. Hong, F. Hu, J. Xie, "A Survey of Anonymity in Wireless Communication Systems," (Wiley Journal) Security and