SWAP: Security Aware Provisioning and Migration of Phone Clones Over Mobile Clouds

Seyed Yahya Vaezpour*, Rui Zhang†, Kui Wu*, Jianping Wang†, and Gholamali C Shoja*
*Computer Science Dept., University of Victoria, B.C., Canada. Email: {vaezpour,wkui}@uvic.ca, gshoja@cs.uvic.ca
†Computer Science Dept., City University of Hong Kong, Hong Kong. Email: zhangrui.ray@gmail.com,jianwang@cityu.edu.hk

***Abstract*—Mobile cloud provides smart phone users with unprecedented opportunities to enjoy the abundant computing and storage resources of cloud computing. One viable scheme is to offload computational intensive applications to a mobile phone's agent in the cloud, which could be implemented as a thin virtual machine (VM), also termed as phone clone, in the cloud. Due to shared hardware components (e.g. memory bus and CPU cache) among co-resident VMs, a VM is subject to covert channel attacks and may potentially leak information to other VMs located in the same physical host. Due to the large number of phone clones, it is not practical to guarantee absolute physical isolation of phone clones, and as such a phone clone may have to "dance with strangers" on the same host. In this paper, we address two critical problems in such a computing platform: how to allocate phone clones to minimize the risk of information leakage and how to migrate phone clones whenever the risk becomes higher than a given threshold. We design SWAP: a security aware provisioning and migration scheme for phone clones. Our solution utilizes the spatial and temporal features of phone clones, and by considering the online social connection of mobile users, we greatly simplify the search space of the optimal solution. Experimental results indicate that our algorithms are nearly optimal for phone clone allocation and are effective to maintain low risk with a small number of phone clone migrations.**

I. Introduction

Telecom cloud refers to cloud computing services provided by telecom companies. Compared to third-party cloud service providers, telecom companies, particularly the mobile communication companies, have a unique advantage in the service provisioning: they provide the last-mile connection to users and thus have direct knowledge of users' activities. A telecom company can leverage the above advantages to provide its customers with high-quality and more attractive cloud computing services. One type of such services is to build phone clones with cloud computing to help mobile users augment the functionality of their smart phones and increase the lifetime of the battery.

The concept of phone clones [8] is to build software clones of smart phones on the cloud and enable mobile users to offload computation intensive tasks and backup data to the cloud. Current smartphone devices are normally installed with many useful applications to make users' daily life much easier and more efficient than before. Some applications, however, may require intensive calculation and consume a large amount of energy. In this case, the smartphone could offload the tasks to its clone in the cloud [7], [10], [15].

ISBN 978-3-901882-58-6

Depending on different functionalities supported by the phone clones, they could be implemented as a process or a *thin* virtual machine (VM) [8], [15], [21] on a cloud host. Due to the ease of management and the richer functionalities of VM, we assume the VM version of phone clones in this paper. To allow resource multiplexing, multiple VMs are usually allocated and managed with a hypervisor, such as KVM [2] or Xen [4], in one physical machine. Due to the large number of mobile users, it is not surprising if one hypervisor hosts hundreds of phone clones. Under such circumstances, provisioning and migration strategies for phone clones become critical to the success of mobile telecom cloud.

We need to tackle two constraints in the allocation and migration of phone clones: security and resource. To give users' a reasonable sense of security, phone clones should be physically isolated. For example, users should feel more comfortable if their phone clones are collocated with those of their friends rather than strangers. Nevertheless, due to the limited number of physical hosts and the large number of mobile users, it may not be possible to find a good isolation for all phone clones. As a result, a phone clone may have to live together with strangers' phone clones on the same host.

It has been shown that a VM can attack another VM on the same host via covert channel attacks [18], [25]. Such attacks exploit the CPU cache or the memory bus in a virtualized environment to steal information from other VMs. It has been demonstrated in [18], [25] that covert channel can be effective and very hard to detect. The big challenge we are faced with in mobile cloud is: how to *mitigate* the risk of covert channel attacks when we do not have enough resource to physically isolate strangers' phone clones?

In this paper, we present our first attempt to tackle this problem. We propose and evaluate SWAP: a security aware provisioning and migration scheme for phone clones. For the provisioning of new phone clones, we take advantage of the mobile telecom cloud where it is easy to build a communication graph based on mobile users' communication history. The communication graph reflects the relationship between mobile users, and it should be safe to allocate together the phone clones that have a direct communication link, since they need to communicate anyway. Whenever this requirement cannot be met due to resource constraint, we then solve

the optimization problem that minimizes the risk posed by potential covert channels. Our way of mitigating the negative impact of covert channels comes from the observation that covert channels normally need time to build [11], [28]. By constraining the collocating time duration of two strangers' phone clones on the same host, we can counter the host limitation problem by migrating strategically selected VMs.

The contributions of this paper are two-fold.

- First, we propose a system model that captures the mobile users' communication relationship and the potential risks when collocating phone clones, and solve the optimization problem that minimizes the risks in the provisioning of new phone clones. The optimization problem requires intensive computations due to the large number of phone clones. To avoid this problem, we present a clique-covering method to pre-process the communication graph and significantly speed up the optimization algorithm.
- Second, we propose a phone clone migration strategy to mitigate the impact of potential covert channels. We introduce a decay function to model the time-varying feature of covert channels, and migrate some phone clones whenever the risk among the phone clones in a host becomes high. In this context, we minimize the total number of migrations to meet a given security requirement.

II. Related Work

Mobile cloud computing [22] and cross-VM covert channels [18] are the most related research areas to this paper. We first discuss relevant state-of-the-art mobile cloud computing research. Then, we review a series of groundbreaking works demonstrating the tangible threat of cross-VM covert channels in public cloud services. Lastly, we point out the difference between existing methods on mitigating the cross-VM covert channel risk in mobile cloud computing and our proposed method.

Led by the pioneering mobile cloud computing projects like MAUI [9] and CloneCloud [8], a growing amount of research efforts focusing on different aspects of the topic have recently been sighted. For instance, Ravi et al. [17] and Cuervo et al. [9] concentrate on alleviating the energy consumption for mobile devices and, hence, lengthening their battery lifetime by outsourcing computational tasks to remote servers via communication network. Barbera et al. [7] and Kosta et al. [15] focus on evaluating the performance tradeoff for workload outsourcing. In addition, the enhancement of system architecture has been a major research direction of mobile cloud computing [6] [21] [10].

Despite the benefit of mobile cloud computing, one major drawback is its security concerns. In addition to the security concerns inherited from generic cloud computing, works have been published to expose the security threats specifically induced by mobile cloud computing [22]. The major concerns can be summarized as information leakage and unauthorized access [14]. Furthermore, among all system components of mobile cloud computing, the risk concealed in the cloud end is the most hazardous one since the malicious users can invest money to become legitimate users of the cloud. Physical isolation between the mobile devices is breached at cloud side as virtualization technologies, such as Xen [4], Linux KVM [2] and VMWare [3], multiplex the physical resources of cloud servers between different users.

As one concrete example of mobile cloud computing security threat, cross-VM covert channel has recently emerged as a serious concern over cloud computing services. Ristenpart et al. [18] point out the feasibility of conducting cross-VM attack in amazon EC2 [1] which can potentially host mobile cloud computing services. There are a number of follow-up works [25] [26] [24] which explore the amount of damage that can be dealt by cross-VM covert channels in a public cloud. These works are also relevant to mobile cloud computing, since, at the cloud end, the mobile device of each user corresponds to a unique virtual machine, a.k.a. phone clone, which this mobile device offloads computational tasks to.

Covert channel is a classic security problem, as elaborated in a U.S. NCSC report a.k.a the light pink book [11]. However, cross-VM covert channels are proven to offer much faster data leakage rate. Existing methods against cross-VM covert channel threat can be categorized into architectural, monitoring or fuzzy time based methods. Architectural based methods rely on modifying the system components, e.g. Kadloor et al. [13] propose a method to modify resource schedulers to mitigate the efficacy of timing channels. Monitoring based methods rely on inserting additional system components dedicated to analyzing the status of shared resources between VMs for detecting the presence of covert channels [27] [20]. Fuzzy time based methods [23] hinder the establishment of timing channels by disabling direct access to high precision timers.

All of the above mentioned methods induce deployment cost. Furthermore, most of them are specific to one type of cross-VM covert channels, leaving space for light weight approaches which provide *general* protection over the cloud users. Jaeger et al. [12] assume the awareness of adverse relationship between cloud tenants and logically enforce the safety of a given VM placement plan by prohibiting potential covert information flow between users with an adverse relationship. Zhang et al [28] propose a game strategy that encourages the cloud tenants to frequently migrate their VMs so that the time window for potential cross-VM covert channels is reduced. Although both of them are inspirational to our work, the former solely provides placement plan evaluation without delivering efficient algorithms on producing a valid placement plan, and the latter ignores the knowledge on cloud tenants and may induce heavy overhead on the cloud.

III. System Model

A. Communication Model

We assume that the system has m phone clones and n hosts. Based on the communication history of phone clones, we can build a communication graph $G =< V, E >$, where V denotes the set of nodes representing the phone clones and E denotes the set of edges representing the communication

between phone clones. When two phone clones i and j communicate with each other, we establish a link between them. The communication graph can be represented with an adjacency matrix $A = [a_{ij}]_{m\times m}$, where $a_{ij} = 1$ indicates that there is a communication link between phone clones i and j and $a_{ij} = 0$ otherwise. To facilitate later calculation, we assume that $a_{ii} = 1$ for any phone clone i.

We use the above communication graph for ease of illustration of problem formulation. The model can be easily extended by introducing trust weights for communication links and by introducing the dynamic changes of communication graphs. Nevertheless, the core problems addressed in this paper remain the same, and all the algorithms proposed in the paper are applicable with slight modifications.

B. Potential Risk and Its Calculation

Definition 1: (Potential Risk): When two phone clones that do not have a direct communication link between them are allocated to the same physical host, the allocation scheme may introduce a potential breach. Given a communication graph $G =< V, E >$, the potential risk of a phone clone allocation scheme, $\mathcal{I}$, is the total number of potential breaches introduced by the scheme.

We use an allocation matrix $X = [x_{ij}]_{m\times n}$ to represent a phone clone allocation scheme, where $x_{ij} = 1$ indicates that phone clone i is allocated to host j and $x_{ij} = 0$ otherwise. Given the adjacency matrix A and the allocation matrix X, we wish to mathematically calculate the potential risk of the phone clone allocation scheme.

It turns out that the calculation could be well formulated with matrix trace. We use $\bar{A}$ to denote the complementary matrix of A, i.e., $\bar{A} = [\bar{a}_{ij}]_{m\times m} = [1 - a_{ij}]_{m\times m}$. Define an $n\times n$ risk indicator matrix, $I = X^T\bar{A}X$, where X^T represents the transpose of matrix X. Then the potential risk of the phone clone allocation scheme can be calculated with

$$\mathcal{I} = \frac{1}{2}tr(I) = \frac{1}{2}tr(X^T\bar{A}X), \tag{1}$$

where $tr(\cdot)$ represents the trace of a matrix.

As a simple example to explain Equation (1), Fig. 1(a) shows the communication graph of 4 phone clones. In this example, phone clone 1 and phone clone 4 are placed on host 1, and phone clone 2 and phone clone 3 are placed on host 2. Since there is no communication edge between phone clones 1 and 4 in the communication graph, and they are placed on the same host, the potential risk of this allocation scheme is 1 in this example. Fig. 1(b) illustrates the calculation of potential risk with matrices A and X.

For ease of reference, the main notations used in the paper are listed in Tables I.

IV. Minimizing The Potential Risk in Phone Clone Allocation

A. Problem Formulation

For provisioning a new system, we first need to solve the following problem to minimize the risk of a phone clone allocation scheme.

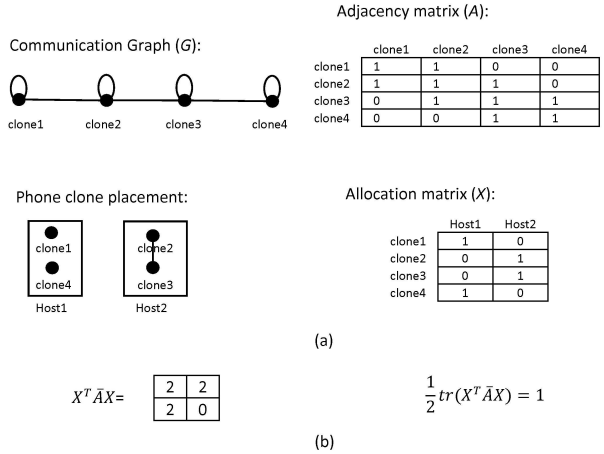


Fig. 1. (a) An example of system model and phone clone placement (b) The calculation of potential risk

Problem 1:

$$\min_{X=[x_{ij}]_{m\times n}} \mathcal{I} = \frac{1}{2}tr(X^T\bar{A}X) \tag{2}$$

subject to

$$x_{ij} \in \{0, 1\} \tag{3}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \text{ for } i = 1, 2, \ldots, m \tag{4}$$

$$\sum_{i=1}^{m} x_{ij} \leq M[j], \text{ for } j = 1, 2, \ldots, n \tag{5}$$

Equations (3) and (4) together assume that a phone clone is allocated to only one host. Inequality (5) indicates that the total number of phone clones assigned to a host should be less than its maximum capacity.

The potential risk is tightly associated with the sparsity of the communication graph. We disclose this relationship with the following lemma and theorem, the proofs of which are provided in Appendix.

Lemma 1: Assume that m phone clones are allocated to n hosts, and that the communication graph is (k, h)-sparse[1]. The potential risk of any reasonable phone clone allocation scheme[2] satisfies the following condition:

$$\mathcal{I} \geq \frac{m^2}{2n} - \frac{m}{2} - (km - nh). \tag{6}$$

Theorem 1: Assume that m phone clones are allocated to n hosts, and that the communication graph is a random graph with link probability of p_l (i.e., each pair of nodes has a link with probability of p_l). For $\forall h > 0$, the potential risk of any phone clone allocation scheme satisfies the following

[1]A graph $G = (V, E)$ with m vertices is called (k, h)-sparse if every subset of $m' \leq m$ vertices spans at most $km' - h$ edges.

[2]An allocation scheme is called reasonable if no host is left empty.

TABLE I
LIST OF NOTATIONS

Notations for Phone Clone Allocation (Section IV)	
Notation	**Definition**
m	Number of phone clones
n	Number of hosts
p	Number of cliques
$A = [a_{ij}]$	Adjacency matrix of communication graph
$X = [x_{ij}]$	Allocation matrix
I	Indicator matrix
$\mathcal{I} = tr(I)$	Potential risk
$R = [r_{ij}]$	Risk matrix
$M[i]$	Maximum capacity of host i
s	Column vector where s[i] represents the number of phone clones in clique i
Notations for Phone Clone Migration (Section V)	
$f(t)$	Decay function
$D(t) = [d_{ij}(t)]$	Decay matrix at time t
$X(t) = [x_{ij}(t)]$	Allocation matrix at time t
$I(t)$	Risk indicator matrix at time t
$\mathcal{I}_v(t)$	Time-variant risk indicator vector
δ	Risk threshold
q	Total number of risky phone clones
l	Number of risky phone clones on the host under investigation
Z	A $l \times 1$ vector where $Z[i,1] = 0$ indicates phone clone i is selected for migration and 1 otherwise.
R_1	$q \times n$ matrix representing the potential risk between a risky phone clone and a host
R_2	$q \times q$ matrix representing the potential risk between two risky phone clones
U	$q \times n$ reallocation matrix where $u_{ij} = 1$ indicates risky phone clone i is assigned to host j and 0 otherwise.
$C[i]$	Remaining capacity of host i

probabilistic lower bound:

$$Pr\{\mathcal{I} \geq (1-p_l)(\frac{m^2}{2n} - \frac{m}{2}) - p_l h \frac{m-n}{2}\} \geq 1 - e^{-p_l h^2(\frac{1}{2} - \frac{1}{2m})/3}. \quad (7)$$

Remark 1: With slight changes, Problem 1 can be used to address more complicated problem settings. For instance, it can be easily extended to study the case where different phone clones demand different computing/memory resources, or the case where a user has different levels of trust with respect to other users.

Remark 2: Problem 1 is a quadratic integer programming problem. It is NP-hard [19], since $\bar{A}$ could be indefinite for a general communication graph. The common way to obtain a lower bound is to use Linear Programming (LP) relaxations. The problem, however, translates to the problem of relaxing Quadratic programming with LP involving indefinite constraints, which is still an open issue [16].

In the following we introduce various heuristics to solve the problem. We first propose a greedy algorithm, called *maximum-conflict-first*, as shown in Fig. 2. The main idea is to handle the "hard" ones first, i.e., the phone clones that have more conflicts with others are allocated first.

In a real system, the number of phone clones is usually big. To speed up the processing, we propose another heuristics in the next section.

1: Sort phone clones in the ascending order of their node degree in the communication graph;
2: **for** each phone clone i in the sorted list **do**
3: Find the host that has the minimum number of phone clones in conflict with i;
4: Allocate phone clone i to this host;
5: **end for**

Fig. 2. The maximum-conflict-first algorithm

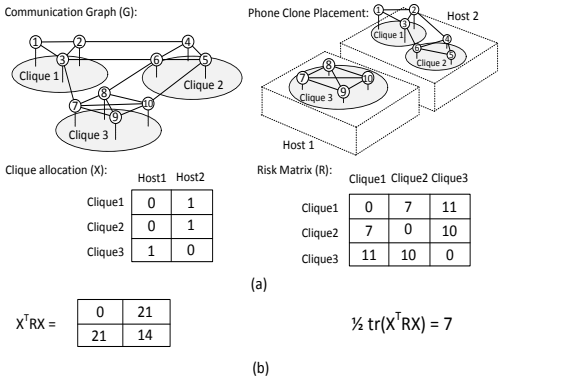


Fig. 3. (a) An example clique-based phone clone placement (b) The calculation of potential risk

B. Reducing Model Complexity with Cliques

Since phone clones that already have communication links are unlikely to attack each other with covert channels, we could consider them as a group and allocate them to the same host. Mathematically, we need to find a clique cover of the communication graph. Since the problem of covering a graph with the minimum number of cliques is proven NP-complete and even does not allow constant approximation [29], we adopt the following well-known heuristic method to obtain an approximate solution:

We iteratively search for cliques that cover more vertices that have not been covered so far. Heuristically, the vertices with larger degrees may have a better chance of appearing in larger cliques. Thus, the search starts from the vertex with the largest degree, until all vertices are covered.

After partitioning the graph into p cliques, we treat the cliques as the basic allocation units, and adjust the measure of potential risk accordingly. If there is no link between a node i in one clique and a node j in another clique, we count a missing edge between the two cliques. We introduce a risk matrix $R = [r_{ij}]_{p \times p}$ where p is the number of cliques after the partition and r_{ij} is the total number of missing edges between clique i and clique j. If we allocate phone clones with cliques as the basic allocation unit, the indicator matrix, $I = X^T RX$. Here we slightly abuse the notation of X by reusing $X = [x_{ij}]_{p \times m}$ to denote the allocation matrix with cliques as the basic allocation unit, because it is easy to figure out the dimension of X from the context.

Therefore, the potential risk of an allocation scheme, $\mathcal{I}$, could be calculated with

$$\mathcal{I} = \frac{1}{2} tr(I) = \frac{1}{2} tr(X^T RX). \quad (8)$$

For example, Figure 3 shows the communication graph of 10 phone clones. In this example, the graph is partitioned into clique 1, clique 2, and clique 3. After partitioning, clique 3 is placed on host 1, and clique 1 and clique 2 are placed on host 2. The potential risk of this allocation scheme is 7.

We thus consider the following optimization problem:

Problem 2:

$$\underset{X=[x_{ij}]_{p\times n}}{\text{Min}} \quad \mathcal{I} = \frac{1}{2} tr(X^T R X) \tag{9}$$

subject to

$$x_{ij} \in \{0, 1\} \tag{10}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \text{ for } i = 1, 2, \ldots, p \tag{11}$$

$$\sum_{i=1}^{p} x_{ij} s[i] \leq M[j], \text{ for } j = 1, 2, \ldots, n \tag{12}$$

where $X = [x_{ij}]$ is an $p \times n$ allocation matrix where $x_{ij} = 1$ indicates clique i is assigned to host j and 0 otherwise. Equations (10) and (11) together assume that a clique is allocated to only one host. Inequality (12) indicates that the total number of phone clones assigned to a host should be less than its maximum capacity.

Problem 2 has the same structure of Problem 1 but has less number of parameters. Since it is also NP-hard [19], we propose a heuristic algorithm, called *QP with Rounding*, as follows:

QP with Rounding: We relax the requirement (10) to allow x_{ij} as real numbers between 0 and 1; we then solve the relaxed problem using quadratic programming. For the x_{ij} values that are either close to 1 or close to 0, we round them up or down to 1 and 0, respectively. In the next round of iteration, we fix those rounded values, and perform the same procedure again until the convergence condition is met.

V. Dynamic Migration of Phone Clones

A. The Problem and The Basic Idea

In the previous section, we considered the *static* phone clone allocation problem. In practice, covert channels need time to build [28] [11]. As such phone clones, even those belonging to strangers, can be safely allocated together during a certain time period. In other words, the risk of covert channels is time dependent.

To address dynamic risk changes over time, we consider the fact that the risk of covert channel attacks becomes higher when two phone clones that initially have no communication link stay too long on the same host. We introduce a *decay function* $f(t)$ and a decay matrix $D(t) = [d_{ij}]_{m\times m}$ for this purpose. If phone clones i and j have a communication link, then $d_{ij}(t) = 1$, otherwise $d_{ij}(t) = f(t)$. The function $f(t) \in [0, 1]$ is the decay function where t indicates the time duration when phone clones i and j stay on the same host. It is a non-increasing function with $f(0) = 1$.

Accordingly, we use $\bar{D}(t)$ to denote the complementary matrix of $D(t)$, i.e., $\bar{D}(t) = [\bar{d}_{ij}(t)]_{m\times m} = [1 - d_{ij}(t)]_{m\times m}$, and $X(t)$ to denote the allocation matrix at time t. The $n \times n$ indicator matrix also becomes time variant and is calculated as $I(t) = X^T(t)\bar{D}(t)X(t)$.

We introduce a time-variant risk indicator vector to model the dynamic risk states of the system:

Definition 2: (Time-variant risk indicator vector): Assume that at time t, the indicator matrix of the system is $I(t)$. The time-variant risk indicator vector of the system at time t, $\mathcal{I}_v(t) = \frac{1}{2} diag(I(t))$, where $diag(.)$ returns a vector with values as the diagonal elements of a matrix. The i-th value in $\mathcal{I}_v(t)$ reflects the risk value on host i at time t.

Note that unlike potential risk, which is a constant for a given allocation scheme, the values in $\mathcal{I}_v(t)$ changes over time.

We call the system is resilient to covert channel attacks if the following condition holds:

$$max(\mathcal{I}_v(t)) \leq \delta, \tag{13}$$

where $max(.)$ returns the largest element in a vector. When this condition is violated, we need to migrate some phone clones and answer the following problem: *how can we maintain the resilience of the system with the minimum number of phone clone migrations?*

To solve the problem, we propose an event-driven heuristics method, i.e., phone clone migration is triggered by the violation of condition (13). Once the phone clone migration is required, the process consists of three steps:

1) Select hosts with the risk value higher than the threshold. These hosts are called risky hosts.
2) Select phone clones from the risky hosts for migration. The selected phone clones are called risky phone clones.
3) Reallocate the risky phone clones.

B. Step 1: Selection of Risky Hosts

After obtaining the time-variant risk vector $\mathcal{I}_v(t)$, the risky hosts can be identified easily by checking the i-th value ($1 \leq i \leq n$) in the vector. Note that the i-th value in the time-variant risk indicator vector represents the risk value of host i.

C. Step 2: Selection of Risky Phone Clones

In this step, we determine the risky phone clones that need to migrate to other hosts.

For each risky host, we calculate the minimum number of phone clones for migration such that the risk value on the host remains below the threshold δ. Assume that there are l phone clones in a chosen risky host k. We construct a $l \times l$ submatrix of $D(t)$, denoted as $D_s(t)$, which is obtained by selecting the l rows and l columns from $D(t)$ corresponding to the phone clones in risky host k. We consider the complementary matrix $\overline{D}_s(t) = \mathbf{1}_{l\times l} - D_s(t)$, where $\mathbf{1}_{l\times l}$ is a $l \times l$ matrix with all 1s.

We try to minimize the number of migrations such that the risk indicator value on host k remains below the threshold δ. We thus consider the following optimization problem:

Problem 3:

$$\underset{Z_{l\times 1}}{\text{Min}} \ -\mathbf{1}_{1\times l} Z \tag{14}$$

subject to

$$z_i \in \{0,1\} \tag{15}$$

$$\frac{1}{2} Z^T \ \overline{D}_s(t) \ Z < \delta, \tag{16}$$

where Z is a $l \times 1$ vector where $Z[i,1] = z_i = 0$ indicates phone clone i is selected for migration and 1 otherwise. Equation (14) tries to minimize the number of phone clone migrations. Inequality (16) ensures that the risk of the remaining phone clones on the host is less than the threshold δ.

Problem 3 belongs to the category of integer linear programming with quadratic constraints, which is hard to solve. We propose a greedy algorithm as follows: At each step, the greedy algorithm selects a phone clone with the maximum risk value, and removes it from the host until the risk value on the host drops below the threshold δ. We run the above algorithm for each risky host separately and select the risky phone clones from each risky host. In the next step, we try to reallocate these risky phone clones to minimize the total risk in the system.

D. Step 3: Migration of risky phone clones

In this step, we reallocate the risky phone clones to the other hosts to minimize the potential risk. Suppose, in the previous step, we have selected a total of q risky phone clones from risky hosts. To migrate the risky phone clones, we need to consider the potential risk of allocating phone clones to hosts and the potential risk among these q risky phone clones.

The potential risk between a risky phone clone and a host can be measured by the matrix $R_1 = [r^1_{ij}]_{q\times n}$, where r^1_{ij} is the total number of missing edges in the communication graph between the risky phone clone i and the phone clones that are currently residing on host j.

There is another potential risk when reallocating two risky phone clones to the same host. The potential risk between two risky phone clones is measured by the matrix $R_2 = [r^2_{ij}]_{q\times q}$, such that $r^2_{ij} = 0$ if the risky phone clone i and the risky phone clone j have a communication link, and 1 otherwise.

We need to solve the following problem:

Problem 4:

$$\underset{U=[u_{ij}]_{q\times n}}{\text{Min}} \ \frac{1}{2} tr(U^T R_2 U) + \frac{1}{2} tr(R_1^T U) \tag{17}$$

subject to

$$u_{ij} \in \{0,1\} \tag{18}$$

$$\sum_{j=1}^{n} u_{ij} = 1, \ \text{for } i = 1,2,\ldots,q \tag{19}$$

$$\sum_{i=1}^{q} u_{ij} \leq C[j], \ \text{for } j = 1,2,\ldots,n \tag{20}$$

where $U = [u_{ij}]$ is an $q \times n$ reallocation matrix where $u_{ij} = 1$ indicates risky phone clone i is assigned to host j and 0 otherwise.

Equation (17) is the objective function to measure the risk of reallocation scheme, in which the first term $\frac{1}{2} tr(U^T R_2 U)$ measures the potential risk of reallocating two risky phone clones to the same host, and the second term $\frac{1}{2} tr(R_1^T U)$ measures the potential risk in reallocating a risky phone clone to a host. Equations (18) and (19) assume that a risky phone clone is reallocated to only one host. Inequality (20) indicates that the total number of risky phone clones assigned to a host should be less than its remaining capacity.

Mathematically, Problem 2 and Problem 4 belong to the same category of integer quadratic programming, and thus can be solved with the same heuristic algorithm introduced in Section IV.

Remark 3: There are generally two concerns in phone clone migration: (1) the migration of a risky phone to a new host might trigger more migrations in the new host, and as such the migration process may not converge to a stable solution; (2) phone clone migration may degrade QoS performance. Nevertheless, the problem in the first concern will not happen, because phone clone migration is triggered by the violation of condition (13). By default, the decay function $f(0) = 1$. That means, when a risky phone clone is re-allocated to a new host, it will not trigger further migration in a short period of time. Further migration is required in the future only when strangers' phone clones stay too long in the same host. By minimizing the potential risk calculated with (17), we minimize the chance for future migration. Regarding the second problem, we can schedule the phone clone migration only when the target phone clone is not being used by mobile users.

VI. Performance Evaluation

A. Simulation Model

We perform comprehensive simulation to evaluate the performance of our phone clone allocation and migration algorithms.

We need to select a decay function $f(t)$ to model the risk of co-placing a pair of phone clones that do not have mutual trust for a continuous time interval t. Zhang et al. [28] and Gligor et al. [11] pointed out that time is needed to establish cross-VM covert channels, including the time spent on compromising the victim VM which leaks data. The time required to establish a covert channel between an arbitrary pair of co-resident VMs is random. Without pre-knowledge of such a random variable, we resort to the common practice that assumes the random variable to be normally distributed. Therefore, the decay function can be modeled using the cumulative distribution function of normal distribution. To simplify the calculation, we estimate this decay function using a Sigmoid function as follows:

$$f(t) = 1 - \frac{1}{1 + \theta \cdot e^{-b(t-a)}} \tag{21}$$

Where θ is a fixed large positive number to bring $f(0)$ close to 1. Then, a and b are adjustable parameters, which control the shape of the decay function. Note that the proposed decay function is just a good example, and other types of decay functions could be used. In our simulation, we set $\theta = 100000$,

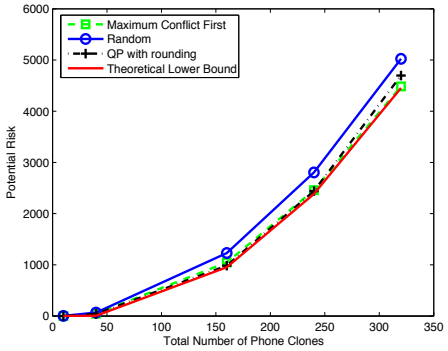


Fig. 4. Potential risk of phone clones with the random communication graph; no of hosts=5; maximum capacity of each host= 100 phone clones

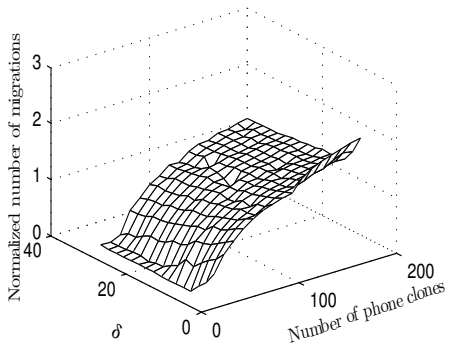


Fig. 5. Normalized number of phone clone migration

$a = 0$, and $b = 1$. Note that we only use the above function as an example, and other functions could be applied within our framework. The "right" selection of decay function depends on the statistical analysis and modelling of covert channel attacks and is beyond the scope of the paper.

We test our algorithms with different communication graph models: the random graph model and the Barabsi-Albert (BA) graph model [5], both of which have been used to model the relationship among online users. In the random graph model, each pair of phone clones are assigned a communication link with a probability. In the Barabsi-Albert graph model, the graph begins with an initial connected network of n_0 phone clones. New phone clones are added one at a time, and each new phone clone is connected to $n_e (n_e \leq n_0)$ existing phone clones with a probability that is proportional to the number of links that the existing phone clones already have.

B. Performance of Phone Clone Allocation

We compare our phone clone allocation algorithms with a naïve random method. In the random method, each phone clone randomly selects a host that has free space to join.

Fig. 4 shows the potential risk with a random communication graph generated with the link probability of 50%. The theoretical lower bounds are obtained using Theorem 1 with the probability no less than 98%. When the number of phone clones is small (say smaller than 50), all methods have similar performance. This is because the system has enough capacity to avoid risky allocation. When the number of phone clones becomes large, the *maximum-conflict-first* algorithm works the best and its performance is very close to the theoretical lower bound. This implies that (1) Theorem 1 provides very tight probabilistic lower bound, and (2) our *maximum-conflict-first* and *QP with rounding* algorithms are nearly optimal.

C. Performance of Phone Clone Migration with Random Graph

We test if our phone clone migration scheme can maintain a low risk level with a small number of phone clone migrations. In the first test, we use the random communication graph generated with the link probability of 50%. The number of hosts is set to 10; each host has a maximum capacity of 50 phone clones. We change the number of phone clones from 20 to 200 and change the risk threshold in each host from 2 to 30. We measure the normalized number of phone clone migrations, which is calculated as the total number of migrations divided by the number of phone clones.

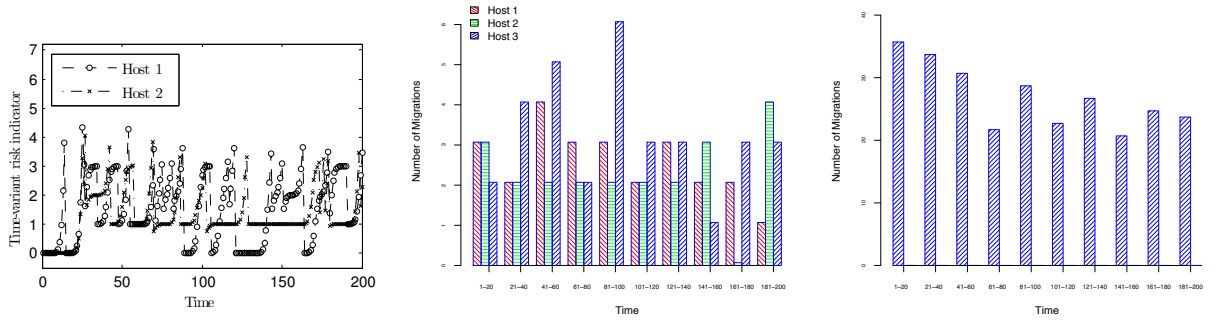
From the results shown in Fig. 5, we can see that although a smaller δ value or a larger number of phone clones results in more phone clone migrations, the normalized number of phone clone migrations is low, e.g., in the worst case (200 phone clones and $\delta = 2$), the average number of phone clone migrations is small in the long term. We also test "heavier" cases by reducing the number of hosts and increasing the number of phone clones, and we observe the similar phenomenon.

To further illustrate the dynamic behavior of phone clone migration, we set the number of phone clones as 100 and randomly select 2 hosts and show the dynamic change of time-variant risk indicator in the two hosts when the risk threshold δ is set to 3. It is clear from Fig. 6(a) that when the time-variant risk indicator in one host increases above the threshold, the migration algorithm kicks in to trigger phone clone migration so that the risk is reduced below the threshold value. Fig. 6(b) shows the number of migrations on three randomly selected hosts. Since phone clone migration is triggered only when the time-variant risk indicator in a host becomes higher than the threshold, we measure the number of migrations every 20 unit time. We have observed that our phone clone migration scheme can maintain a low risk level in all hosts. In addition, we have observed that no host or phone clone behaves significantly different from others, which suggests that our scheme does not penalize a particular host or a particular phone clone with more migrations.

Fig. 6(c) shows the total number of migrations in all hosts. We can easily see that the system is pretty stable in the sense that no sudden large number of phone clone migrations is required. This type of stability is important for ease of system management.

D. Phone clone migration with the BA graph model

We also test the scenarios where the communication graph of phone clones follows the BA model, by setting the model

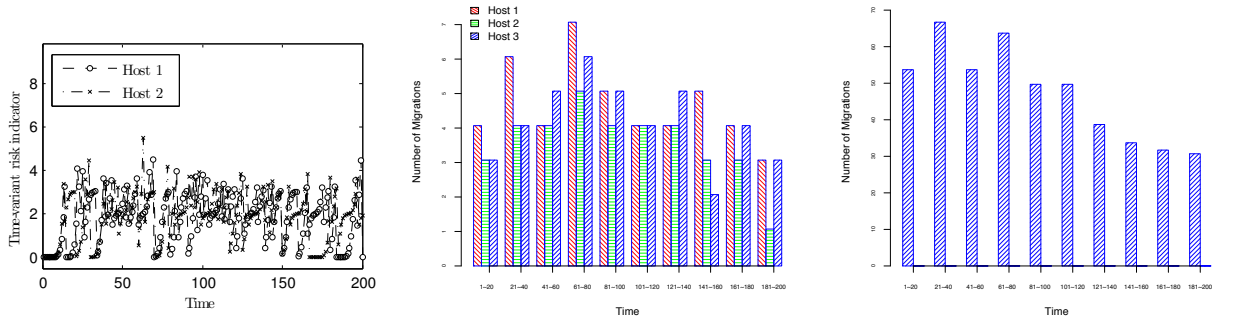


(a) Time-variant risk indicator on two randomly selected hosts

(b) Number of migrations on three randomly selected hosts

(c) Total number of migrations in the system

Fig. 6. The dynamic changes of time-variant risk indicator and number of phone clone migrations with the random communication graph



(a) Time-variant risk indicator on two randomly selected hosts

(b) Number of migrations on three randomly selected hosts

(c) Total number of migrations in the system

Fig. 7. The dynamic changes of time-variant risk indicator and number of phone clone migrations with the BA communication graph

parameters $n_0 = 5$ and $n_e = 4$. Other system settings are the same as in the previous section. Fig. 7(a) shows the dynamic changes of time-variant risk indicator on two randomly selected hosts; Fig. 7(b) shows the number of migrations of three randomly selected hosts; Fig. 7(c) shows the total number of phone clone migrations in all hosts. From the figures, we observe the phenomena similar to those obtained with the random graph model.

VII. Conclusions

Cloud computing opens a new era for mobile communications industry. Mobile users are presented with the great opportunity of utilizing the abundant computing resources in cloud to execute powerful applications. One way to harness this opportunity is to build phone clones in cloud and allow users to offload computation intensive tasks to the phone clones. Associated with the enhanced services to users, however, is the potential information leak over the cloud. In this paper, we systematically study this problem and develop SWAP: a security aware provisioning and migration scheme for phone clones. We solve the core technique challenges in SWAP and develop algorithms that have been demonstrated to work effectively for phone clone allocation and migration.

Acknowledgment

This research was partly supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), Hong Kong General Research Funding under project CityU 9041787, and National Science Foundation China Under project 61272462.

References

[1] Amazon elastic compute cloud (ec2), 2013. http://aws.amazon.com/ec2.
[2] Linux kvm, 2013. http://www.linux-kvm.org/.
[3] Vmware, 2013. http://www.vmware.com/.
[4] Xen, 2013. http://www.xenproject.org/.
[5] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47–97, 2002.
[6] M. V. Barbera, S. Kosta, A. Mei, V. C. Perta, and J. Stefa. Cdroid: Towards a cloud-integrated mobile operating system. In *Proc. of IEEE INFOCOM*, 2013.
[7] M. V. Barbera, S. Kosta, A. Mei, and J. Stefa. To offload or not to offload? the bandwidth and energy costs of mobile cloud computing. In *Proc. of IEEE INFOCOM*, 2013.
[8] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti. Clonecloud: elastic execution between mobile device and cloud. In *Proceedings of the sixth conference on Computer systems*, pages 301–314. ACM, 2011.
[9] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl. Maui: making smartphones last longer with code offload. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 49–62. ACM, 2010.
[10] N. Di. Gearing resource-poor mobile devices with powerful clouds: Architectures, challenges, and applications. *IEEE Wireless Communications*, page 3, 2013.
[11] V. Gligor. *A Guide To Understanding Covert Channel Analysis of Trusted Systems*. National Computer Security Center, U.S., 1993.
[12] T. Jaeger, R. Sailer, and Y. Sreenivasan. Managing the risk of covert information flows in virtual machine systems. In *Proceedings of the 12th ACM symposium on Access control models and technologies*, SACMAT '07, pages 81–90, 2007.
[13] S. Kadloor, N. Kiyavash, and P. Venkitasubramaniam. Mitigating timing based information leakage in shared schedulers. In *Proc. of IEEE INFOCOM*, pages 1044–1052, 2012.

[14] A. N. Khan, M. M. Kiah, S. A. Madani, M. Ali, et al. Enhanced dynamic credential generation scheme for protection of user identity in mobile-cloud computing. *The Journal of Supercomputing*, pages 1–20, 2013.
[15] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang. Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading. In *Proc. of IEEE INFOCOM*, pages 945–953, 2012.
[16] A. Qualizza, P. Belotti, and F. MargotKosta. Linear programming relaxations of quadratically constrained quadratic programs. *The IMA Volumes in Mathematics and its Applications*, 154:407–426, 2012.
[17] A. Ravi and S. K. Peddoju. Energy efficient seamless service provisioning in mobile cloud computing. In *SOSE*, pages 463–471, 2013.
[18] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage. Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. In *Proceedings of the 16th ACM conference on Computer and communications security*, CCS '09, pages 199–212, 2009.
[19] S. Sahni. Computationally related problems. *SIAM Journal on Computing*, 3(4):262–279, Dec. 1974.
[20] B. Saltaformaggio, D. Xu, and X. Zhang. Busmonitor: A hypervisor-based solution for memory bus covert channels. In *Proceedings of the 6th European Workshop on Systems Security*, EuroSec'13, 2013.
[21] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies. The case for vm-based cloudlets in mobile computing. *Pervasive Computing, IEEE*, 8(4):14–23, 2009.
[22] P. K. Tysowski. *Highly Scalable and Secure Mobile Applications in Cloud Computing Systems*. PhD thesis, University of Waterloo, 2013.
[23] B. C. Vattikonda, S. Das, and H. Shacham. Eliminating fine grained timers in xen. In *Proceedings of the 3rd ACM workshop on Cloud computing security workshop*, CCSW '11, pages 41–46, 2011.
[24] Z. Wang. *Information Leakage Due to Cache and Processor Architectures*. PhD thesis, Princeton University, 2012.
[25] Z. Wu, Z. Xu, and H. Wang. Whispers in the hyper-space: high-speed covert channel attacks in the cloud. In *Proceedings of the 21st USENIX conference on Security symposium*, Security'12, pages 9–9, 2012.
[26] Y. Xu, M. Bailey, F. Jahanian, K. Joshi, M. Hiltunen, and R. Schlichting. An exploration of l2 cache covert channels in virtualized environments. In *Proceedings of the 3rd ACM workshop on Cloud computing security workshop*, CCSW '11, pages 29–40, 2011.
[27] Y. Zhang, A. Juels, A. Oprea, and M. K. Reiter. Homealone: Co-residency detection in the cloud via side-channel analysis. In *Proceedings of the 2011 IEEE Symposium on Security and Privacy*, SP '11, pages 313–328, 2011.
[28] Y. Zhang, M. Li, K. Bai, M. Yu, and W. Zang. Incentive compatible moving target defense against vm-colocation attacks in clouds. In *SEC*, volume 376 of *IFIP Advances in Information and Communication Technology*, pages 388–399, 2012.
[29] D. Zuckerman. Np-complete problems have a version that's hard to approximate. In *Proceedings of the 8th IEEE Annual Structure in Complexity Theory Conference*, May 1993.

APPENDIX: PROOFS

A. Proof of Lemma 1

Proof: Given a phone clone allocation scheme $m_1, m_2, \ldots, m_n$, where $m_i (i = 1, 2, \ldots, n)$ is the number of phone clones allocated to host i, we have $m_i > 0$ and $m = \sum_{i=1}^{n} m_i$. Since the communication graph is (k, h)-sparse, every subset of $m' \leq m$ vertices spans at most $km' - h$ edges. Because the total number of possible links among the phone clones on host i is $\frac{m_i(m_i-1)}{2}$, the potential risk at host i, denoted by $\mathcal{I}_i$, satisfies:

$$\mathcal{I}_i \geq \frac{m_i(m_i-1)}{2} - (km_i - h).$$

We thus obtain the potential risk of the allocation scheme:

$$\begin{aligned}\mathcal{I} = \sum_{i=1}^{n} \mathcal{I}_i &\geq \sum_{i=1}^{n} \frac{m_i^2}{2} - \sum_{i=1}^{n} \frac{m_i}{2} - \sum_{i=1}^{n} (km_i - h) \\ &\geq \frac{m^2}{2n} - \frac{m}{2} - (km - nh).\end{aligned}$$

B. Proof of Theorem 1

Proof: Given a phone clone allocation scheme $m_1, m_2, \ldots, m_n$, where $m_i (i = 1, 2, \ldots, n)$ is the number of phone clones allocated to host i, we have $m_i > 0$ and $m = \sum_{i=1}^{n} m_i$. Denote the number of links among the phone clones on host i as $\mathcal{L}_i$. Since the communication graph is a random graph with link probability of p_l,

$$E[\mathcal{L}_i] \equiv \mu = p_l \frac{m_i(m_i-1)}{2}. \tag{22}$$

Since the links in a random graph are independent, based on Chernoff bound, $\forall \epsilon_i h > 0 (\epsilon > 0)$,

$$Pr\{\mathcal{L}_i \geq \mu(1+\epsilon_i h) < (\frac{e^{\epsilon_i h}}{(1+\epsilon_i h)^{(1+\epsilon_i h)}})^u$$

Since $\epsilon_i h > 0$, we can use $e^{-u(\epsilon_i h)^2/3}$ to approximate $(\frac{e^{\epsilon_i h}}{(1+\epsilon_i h)^{(1+\epsilon_i h)}})^u$. When $0 < \epsilon_i h \leq 1$, we have $(\frac{e^{\epsilon_i h}}{(1+\epsilon_i h)^{(1+\epsilon_i h)}})^u \leq e^{-u(\epsilon_i h)^2/3}$. When $1 < \epsilon_i h$, the maximum gap between $e^{-u(\epsilon_i h)^2/3}$ and $(\frac{e^{\epsilon_i h}}{(1+\epsilon_i h)^{(1+\epsilon_i h)}})^u$ is less than 3%. Therefore, it is appropriate to use $e^{-u(\epsilon_i h)^2/3}$ to approximate $(\frac{e^{\epsilon_i h}}{(1+\epsilon_i h)^{(1+\epsilon_i h)}})^u$. We have

$$Pr\{\mathcal{L}_i \geq \mu(1+\epsilon_i h)\} \leq e^{-u(\epsilon_i h)^2/3}$$

Equivalently,

$$Pr\{\mathcal{L}_i < \mu(1+\epsilon_i h)\} \geq 1 - e^{-u(\epsilon_i h)^2/3}$$

Following the same reasoning in Lemma 1, the potential risk at host i, denoted by $\mathcal{I}_i$, satisfies:

$$Pr\{\mathcal{I}_i \geq \frac{m_i(m_i-1)}{2} - \mu(1+\epsilon_i h)\} \geq 1 - e^{-u(\epsilon_i h)^2/3} \tag{23}$$

Since Inequality (23) holds for all $\epsilon_i > 0$, we set

$$\epsilon_i = \frac{1}{m_i} \tag{24}$$

Replacing (22) and (24) into (23), we have

$$\begin{aligned}&Pr\{\mathcal{I}_i \geq \frac{m_i^2 - m_i}{2}(1 - p_l(1 + h\frac{1}{m_i}))\} \geq 1 - e^{-u(\epsilon_i h)^2/3} \\ &Pr\{\mathcal{I}_i \geq \frac{m_i^2 - m_i}{2}(1-p_l) - \frac{m_i - 1}{2} p_l h\} \geq 1 - e^{-p_l h^2(\frac{1}{2} - \frac{1}{2m})/3}\end{aligned} \tag{25}$$

Note that the right-hand side is due to the fact that $\frac{1}{m} \leq \frac{1}{m_i}$. Since $\mathcal{I} = \sum_{i=1}^{n} \mathcal{I}_i$, we have

$$\begin{aligned}&Pr\{\mathcal{I} \geq \sum_{i=1}^{n} \frac{m_i^2 - m_i}{2}(1-p_l) - \sum_{i=1}^{n} \frac{m_i - 1}{2} p_l h\} \\ &\geq 1 - e^{-p_l h^2(\frac{1}{2} - \frac{1}{2m})/3}\end{aligned} \tag{26}$$

That is,

$$Pr\{\mathcal{I} \geq (1-p_l)(\frac{m^2}{2n} - \frac{m}{2}) - p_l h \frac{m-n}{2}\} \geq 1 - e^{-p_l h^2(\frac{1}{2} - \frac{1}{2m})/3}.$$