Network-Based Secret Communication in Clouds: A Survey

Johanna Ullrich, Tanja Zseby, Joachim Fabini, Edgar Weippl

***Abstract*—The cloud concept promises computing as a utility. More and more functions are moved to cloud environments. But this transition comes at a cost: Security and privacy solutions have to be adapted to new challenges in cloud environments. We investigate secret communication possibilities – data transmission concealing its mere existence or some of its characteristics – in clouds. The ability to establish such secret communication provides a powerful instrument to adversaries and can be used to gather information for attack preparation, to conceal the coordination of malicious instances or to leak sensitive data.**

In this paper, we investigate potentials for secret communication in cloud environments and show possible application scenarios. We survey current approaches of different kinds of secret communication including covert channels, side channels and obfuscation techniques. While most existing work focuses on covert and side channels within a physical server (cross-VM channels), we place emphasis on network-based covert and side channels, which are rarely addressed in current literature about cloud security. We then discuss secret communication techniques with respect to the application scenarios and show their advantages and limitations.

I. Introduction

Cloud computing has changed the way we think about computing and has made the dream of computing as a utility come true. The paradigm of cloud computing is based on an offer of "*reliable services delivered through next-generation data centers that are built on virtualized compute and storage technologies*" [1]; the data centers themselves are called clouds [2]. Cloud services are accessible from everywhere in the world, and are considered more cost-effective than solutions that are based on local resources within enterprises [3].

Migration of systems to cloud environments is ongoing, and there is barely anything that has not yet been moved: enterprise systems in the oil and gas industry reducing support calls by a fifth and costs by a third [4], service-oriented architectures [5], legacy software that was originally implemented without even a vague idea of cloud computing [6], or a vast scientific database of astronomical data that contains more information than the US Library of Congress [7]. Recently, mobile clouds have been developed where cloud services augment mobile devices, mainly to overcome the latter's performance limitations [8], [9], [10].

As with the development of other utilities like water, electricity or telephony, the implementation of ubiquitous cloud computing creates higher expectations. Among other requirements, clouds have to be secure and fulfill requirements on data confidentiality, integrity and availability. At the same time, clouds demand a fundamental rethinking of existing assumptions on security. For example, its pervasive networking challenges the traditional perimeter protection of enterprises [11].

Related publications focus on specific aspects of cloud computing security like, e. g., hardware virtualization [12], data protection from the infrastructure provider [13], or interconnected clouds [14], to name a few. Beyond, security challenges in the cloud have been investigated in general, e. g., [11] emphasizes the risk of transition and further names identity and access management, data security, trust as well as assurance as security issues. [15] adds resource location and the involved co-residency, service-level agreements and accountability.

In our paper we focus on the specific problem of secret communication in cloud environments. The ability to establish such secret communication provides a powerful instrument to adversaries and can be used to gather information for attack preparation, to conceal the coordination of malicious instances or to leak sensitive data. Means of secret communication include hiding of information transmission using *covert channels*, extraction of secret information using *side channels* and the concealment of communication partners through *obfuscation*.

Cloud-based covert and side channels have been already addressed for instance in [16] and [17], but both concentrate on channels among virtual machines on the same physical server (*cross-VM channels*). But since networking is an inherent part of cloud computing, network-based covert and side channels are highly relevant. Beyond covert and side channels, we include a third class of secret communication: obfuscation techniques to conceal the IDs (usually IP addresses) of communication partners, e. g., by using techniques for anonymous network communication such as onion routing [18]. Further, we show potentials to exploit side channels techniques to realize bidirectional communication via covert channels. Cloud solutions operate in very specific network environments that have to be taken into account when assessing possibilities for secret communication. This paper addresses these challenges and investigates network-based secret communication in clouds.

We first highlight potentials for secret communication in typical cloud structures and describe scenarios, both typically considered benign and malicious, in which such communication channels are used for the sake of secrecy. We then

Johanna Ullrich and Edgar Weippl are with SBA Research, Austria; Joachim Fabini and Tanja Zseby are with the Institute of Telecommunications, TU Wien, Austria.
jullrich,eweippl@sba-research.org, joachim.fabini,tanja.zseby@tuwien.ac.at

present currently known approaches for secret communication channels, investigate their usefulness in cloud environments and classify them according to characteristics relevant to the cloud application scenarios.

Our literature survey reveals 9 covert channels, 20 side channels and 5 obfuscation techniques that are specific to cloud computing. While previous work focused on channels exploiting co-residency, just a minority of network-based channels that were discovered in the literature are of this kind: 2 of 9 covert channels, and 8 of 20 side channels.

We additionally show potentials to exploit side channels as covert channels by intentionally influencing the information revealed in a side channel. Our corresponding analysis shows that out of 20 side channels from the literature, 18 have the potential to be used as covert channels. Some side channels even bear potential for two distinct covert channels leading to a total of 20 covert channels that have not been covered by literature yet.

Approaches from the literature primarily establish communication channels between external nodes and ignore the potential involvement of other stakeholders. Our analysis, however, indicates that cloud-based secret communication is multifaceted, and there are more possible communication partners.

Available obfuscation techniques can be categorized into two groups according to their application scenarios: Obfuscation evading censorship, and transmission of illegal or regime-critical content seem to use more sophisticated techniques like *Tor* [18]. The majority of alternatives seems to serve *Command & Control* infrastructures of botnets and are of comparatively low technical finesse.

The remaining paper is organized as follows: Section II defines the fundamental terms of cloud computing, discusses its key technologies as well as participating stakeholders and presents related work on cloud computing security. Section III presents patterns of secret communications – covert channels, side channels and obfuscation techniques – and compares them. Section IV describes the potential of secret communication in cloud environments, while Section V discusses scenarios in which these means of communication might be applied. A survey of currently known approaches is provided in Section VI, and their classification in Section VII. Our findings are discussed in Section VIII. Section IX concludes the paper.

II. Cloud Terminology

In this section, we introduce terminology related to cloud computing. Cloud terminology is ambiguous in some cases; thus, we define common terms to generate a common understanding. In a first step, we define cloud computing in general and present service models that assign responsibility to different stakeholders in manifold ways. These definitions, however, are rather vague from a technical point of view. Thus, we review key technologies that lay the foundation for cloud computing, and highlight security challenges in cloud computing despite its lacking novel technologies. In a second step, we shed light on different roles and entities in cloud computing. Finally, we review related work on cloud computing and discuss it with particular focus on secret communications in cloud environments.

A. Clouds and the Challenge of Security

The term cloud computing emerged in the late 2000s [1], and rather describes a mode of operation that combines several other technologies than a novel technology itself [19]. Definitions are diverse [20], and rather comprehensive. The definition from the *National Institute of Standards and Technology (NIST)* appears to be the most popular, and sees cloud computing as *"a model for enabling ubiquitous, convenient, on demand network access to a shared pool of configurable computing resources (e. g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction"* [21]. In addition, five essential cloud characteristics are defined, namely "*on-demand self-service*", "*broad network access*", "*resource pooling*", "*rapid elasticity*" and "*metering*".

Real-world cloud computing encompasses a broad range of applications: social networking, tax and health applications, storage solutions, virtual machine rentals, and many more. Generally, these services are classified into the three service models *SaaS*, *PaaS* and *IaaS* [21]. We added *StaaS* as a fourth service model due to its characteristics. The service models assign responsibilities to customer and cloud provider in different combinations.

- In *Software as a Service (SaaS)*, the customer uses "*provider's applications running on a cloud infrastructure*" [21] by means of a web browser or a certain client. Examples are *Twitter*[1] and *Google Cloud Messaging*[2].
- *Platform as a Service (PaaS)* clouds provide a platform including programming languages, libraries, etc. to run "*consumer-created or acquired applications*" [21]. An example is *Google App Engine*[3].
- In *Infrastructure as a Service (IaaS)*, the cloud provides resources to the customers. The latter are able to "*run arbitrary software, which can include operating systems and applications*" [21] on this resources. Examples are *Amazon Elastic Compute Cloud (EC2)*[4] and *Google Compute Engine*[5].
- *Storage as a Service (StaaS)* offers synchronization into the cloud and a possibility for storing backups. While some see it as a specialization of *IaaS* due to storage provision [22], its aspect of offering a certain client for easy up- and downloading tends to be *SaaS*. For the purpose of this paper, we use *StaaS* as a service model sui generis. A real-world example is *Dropbox*[6].

From a technology perspective, the definition of cloud computing appears vague. Following the idea of cloud computing

[1] www.twitter.com
[2] developers.google.com/cloud-messaging/
[3] cloud.google.com/appengine/
[4] aws.amazon.com/de/ec2/
[5] cloud.google.com/compute/
[6] www.dropbox.com

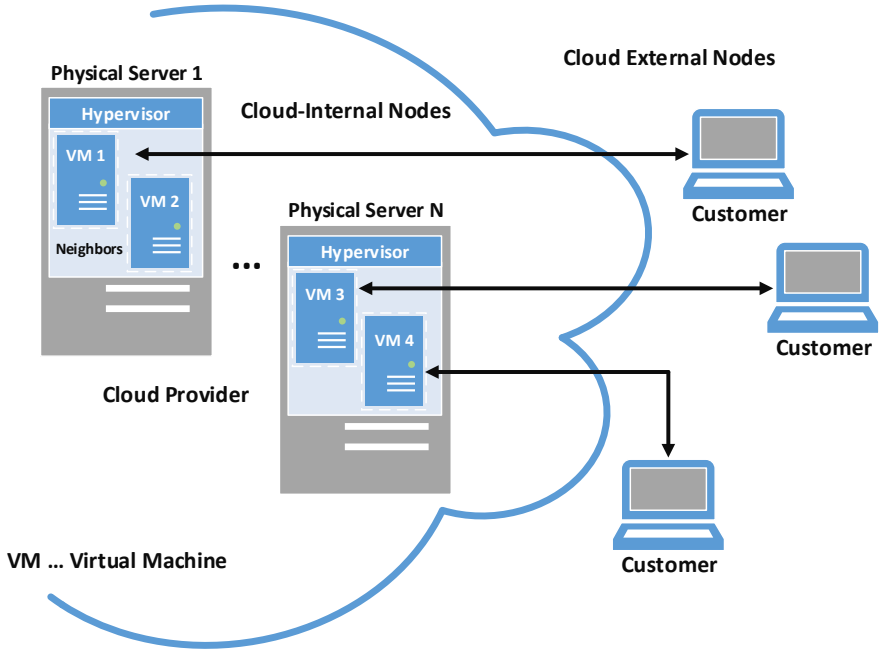


Fig. 1: Roles in Cloud Computing

as an operational model and a confluence of existing technologies, [22] highlights the *(1) spread of devices*, *(2) the trend towards browser interfaces or thin clients*, *(3) provisioning of services over the network*, *(4) dropping of hardware prices*, *(5) sharing of data centers* as well as *(6) the development of Application Programming Interfaces (APIs)* as relevant for the establishment of clouds.

Still, one might ask for the reasons of seemingly excessive engagement with the cloud. Cloud computing is certainly a tremendous economic success story: This year's global market size is estimated to $96.98 billions, and its annual growth rate to 9.14 % [23]. Market leader *Amazon* alone generates an annual $6 billion revenue [24]. By 2014, 69 % of enterprises had an application or infrastructure in the cloud, another 18 % planned to do so within the following year [25]. Beyond, cloud services for end users have large user bases. Moving into the cloud is primarily an economic decision: Cloud computing does not require upfront capital investment for infrastructure, and typically provides (almost) immediate access. Flexibility allows starting with little resources, and increase later if needed. It unfolds its full effect with the creation of new services based on existing ones, and is a key enabler of many novel services due to low economic risk. Low entry costs further enable small companies access to computing facilities that were previously only accessible by large players [22], [26].

Despite lacking of novel technology, there is a decent technical aspect in the deployment of cloud computing: The combination of several existent technologies into a new operation model implies that they are now used in an environment with different characteristics. As a consequence, several (formerly) basic assumptions might be broken and partial redesign required. Among them are:

- Traditional enterprise architectures follow a zoned approach. Internal is considered as benign, external as potentially malicious and thus perimeter protection is applied. Potentially sensitive data is now traveling the Internet on its way to the cloud or back to the customer. Even the cloud-internal traffic is not necessarily benign as the cloud-internal network is shared among (potentially malicious) customers. A traditional zoned approach becomes unfeasible.
- Moving to the cloud changes infrastructure from a white box to a black box. An operator knows the details of his infrastructure and thus its advantages as well as its disadvantages, or is at least able to find out. Cloud providers however see their internal structure as their company secret, and disclose only a limited view thereof. This means that the customers are not fully aware of the provider's intentions and vice versa; and their combination might introduce risks that neither of the two is aware of.
- The customer is not solely unaware of the infrastructure, but also dependent on the cloud provider's offer. While a minor change in configuration might be easy in a self-operated infrastructure, it is almost impossible to do so in the cloud. In the worst case, a provider has to be replaced by another. However, migration is another challenge as

there are barely any standards.

Such broken assumption due to a changed use of technology might negatively impact non-functional requirements. Among them is security, and it is thus of utmost importance to address this issue. Indeed, concerns of security are considered the major obstacle by enterprises when moving to the cloud [25].

B. Roles and Entities in Cloud Computing

The definition of service models in cloud computing has already revealed two roles – the cloud provider and the customer. These and other roles are defined for the purpose of this paper based on descriptions found in the literature [27], [28], [29], [1], [22], [30], [21], and are also depicted in Figure 1.

- A *cloud provider* develops, operates and offers cloud services. Others are able to access these services via the Internet, and use them for their own purpose. The offered services follow one of the cloud service models. Major cloud providers not solely offer a single service, but a variety thereof and also span among more than a single service model.
- A *cloud customer* is somebody who accesses and uses a cloud service; cloud customers are also referred to as *cloud users* or *consumers*. Typically, they pay the cloud provider for the service. Customers might be enterprises or private individuals.
- *Virtual machines* pretend to be real computers, but are virtual representations thereof. They are a consequence of virtualization technology that enables different cloud customers to utilize the same hardware simultaneously.
- The *physical server* is the hardware that is abstracted by means of virtualization. Atop runs a *hypervisor* (or *virtual machine monitor*). This is a software that manages virtual machines. Multiple virtual machines might reside on the same physical server and share its hardware.
- A *neighbor* is a virtual machine that resides on the same physical server as another virtual machine, i. e., they share a server's hardware resources. Neighbors are *co-resident* to each other; the situation is referred to as *co-residency*.
- A *node* is an entity that is connected to the network – independently whether its role is data redistribution or being a data end point. The term includes virtual machines, but also non-virtualized (ordinary) computers, e. g., adversaries that reside outside the cloud. For clarity, we do not apply the term *node* for physical servers in this paper. We distinguish *cloud-based nodes* from *cloud-external nodes*. Cloud-based nodes reside in a cloud, while external nodes are outside and reside at an arbitrary place on the Internet. Customers access the cloud typically by means of external nodes, e. g., laptops, smart phones, etc.

C. Related Work on Cloud Computing Security

Security in cloud computing has been a great challenge; related work on this topic is manifold and a variety of surveys as well as tutorials are available. We identified two distinct groups: (1) general surveys and (2) specific surveys on cloud computing security.

General surveys aim to raise fundamental awareness on security issues with respect to cloud computing, and address a broad spectrum of potential pitfalls. As a consequence of this broad spectrum, functionality and details of underlying technologies are typically treated in a marginal way. Specific attacks are not described, or solely a single or a few attacks are emphasized as representatives. General surveys tend to have a lower number of references than the group of specific surveys. The level of detail with regard to technology is low, and proposed countermeasures are rather of organizational or legal nature like, e. g., standards, service level agreements, compliance checks or caution in choosing a trustful cloud computing provider.

Many general surveys have been published so far: *Kaufman* [31] considers minimal requirements for cloud computing in order to guarantee the CIA (*confidentiality, integrity, availability*) triad. The paper calls for organizational countermeasures; especially for increased cooperation between the technical and the legal sphere. *Jensen et al.* [32] highlight four technical aspects of importance in cloud computing security: XML signatures, browser security, integrity and binding issues as well as flooding attacks, and emphasize them by means of some practical attacks. *Chen et al.* [17] differentiate between "real" security challenges in cloud computing and the ones that arise from the ordinary evolution of the Internet. In particular, they identify multi-party trust issues and a lack of mutual auditability. *Krutz et al.* [33] provide a comprehensive guide on security in cloud computing targeting rather security practitioners than academia, and especially consider the topic of unauthorized data access in all its facets as well as compliance, availability and robustness of security protection. *Popovic et al.* [34] recommend control objectives to the technical and business community by highlighting ten high-level security issues, and propose organizational countermeasures as a solution. *Takabi et al.* [35] highlight unique characteristics of cloud computing that exacerbate security or privacy like handling of data by third parties, shared responsibility or compliance. *Dorey et al.* [11] emphasize the risk of transition, and further name identity and access management, data security, trust and assurance as security issues. *Subashini et al.* [36] discuss security threats that arise from cloud computing's service delivery models. Attacks are listed without detailed analysis and complemented by an overview on general security shortcomings in the cloud. *Zissis et al.* [37] run in the same vein as *Takabi et al.* by "demystifying" unique cloud security challenges, and propose a trusted third party to relieve clients from the security burden. *Rong et al.* [15] provide a high-level overview on general security challenges, and propose the extension of service-level agreements to include security requirements. With respect to our topic of secret communication, *Subashini et al.*, *Tabaki et al.*, *Zissis et al.* and *Rong et al.* identify multi-tenancy as a threat; indeed, co-residency represents a substrate for secret communication. However, they neither name the risk of

secret communication, nor describe specific attacks. Beyond, *Zissis et al.* and *Krutz et al.* anticipate disclosure of private data as a consequence of multi-tenancy; *Chen et al.* and *Krutz et al.* explicitly name the risk of covert and side channels without going into details.

In comparison, specific surveys highlight a specific security aspect of cloud computing, e. g., interconnected clouds, or security issues of a certain technology that is used in cloud computing, e. g., hardware virtualization. This thematic restriction leads to a narrower focus, and a more in-depth look on the respective topic, i. e., a higher level of detail. Specific surveys discuss a high number of related attack vectors and security shortcomings. Considering a cloud computing technology at a very fine-grained level, proposed countermeasures concentrate on architectural and technological improvements like, e. g., a different networking architecture, different scheduling or modified priorities of virtual machines, etc.

Specific surveys investigate diverse topics: *Aceto et al.* [38] focus on cloud monitoring, *Pek et al.* [12] on hardware virtualization, *Del Piccolo et al.* [39] on network isolation, *Ryan* [13] on data protection from the infrastructure provider's perspective, and *Toosi et al.* [14] on interconnected clouds. *Ahmad et al.* [40] survey Software-Defined Networking's (SDN) security of the application, control and data planes, *Rawat et al.* [41] SDN's security with respect to its architecture and energy efficiency, *Khan et al.* [42] investigate threats in SDN's topology discovery and develop a taxonomy, *Scott et al.* [43] focus on solutions to overcome such security threats. *Yan et al.* [44] elaborate on SDN's vulnerability with respect to becoming a target of denial-of-service attacks as well as its capability to participate in such attacks. *Colman-Meixner et al.* [45] consider cloud architectures with respect to resilience against accidents, disasters and attacks.

The work of *Xiao et al.* [16] is closest to our paper; the authors identify five most representative security and privacy attributes and present a large collection of attack vectors that adversaries may exploit. The authors dedicate a section to covert and side channels. However, they concentrate on channels among virtual machines on the same physical server (*cross-VM channels*). The discussed channels exploit resources as L2 caches or the CPU, but not networking capabilities. In contrast, our paper focuses on covert and side channels that arise from networking. Network-based covert and side channels are highly relevant since networking is an inherent part of cloud computing. Beyond covert and side channels, we include a third class of secret communication: obfuscation techniques to conceal the IDs (usually IP addresses) of communication partners, e. g., by using techniques for anonymous network communication such as onion routing [18].

Targeting obfuscation in clouds, *Pearson et al.* [46] propose a privacy manager that uses obfuscation to protect cloud data. However, this kind of obfuscation differs from our paper's meaning of obfuscation. Obfuscation with respect to the privacy manager means encryption that is applied to sensitive data before the latter is uploaded into the cloud. In our paper, obfuscation means a technique like for example *Tor* [18] that allows anonymous communication through hindering the correlation of sender and receiver or concealment.

	General Surveys	Specific Surveys	Our Survey
Focus	Broad	Narrow	Narrow
Level of Detail	Low	High	High
Number of References	Low	High	High
Description of Specific Attacks	No[a]	Yes	Yes

TABLE I: Related Work: Surveys considering Cloud Computing Security in General vs. Surveys considering a Specific Cloud Security Aspect

[a] at most as representative examples

Applying the above mentioned criteria to the survey at hand, we classify our paper as a specific survey for the following reasons:

- We discuss the aspect of secret communication in cloud computing. Although including three distinct types of secret communication, the paper's focus is considered to be narrow as these types of clandestine communication represent a specific aspect of the general topic of cloud computing security.
- The paper reviews and discusses a multitude of actual covert channels, side channels and obfuscation techniques at a high level of detail, providing a high number of references.

Our survey differs significantly from previous surveys in that it focuses strictly on network-based secret communication in cloud computing. Table I summarizes the characteristics of our survey on secret communications against the characteristics of existing general and specific surveys.

III. Secret Communication

Data transmission over networks typically follows a regular pattern: A sender wants to transmit a message to a receiver, and thus embeds data in packets. Packets do not only hold a payload but also a header containing information for delivery. The receiver extracts the data from the packets and processes the data according to its needs. For protection, transmitted data might be encrypted so that only the intended receiver can access the actual content. Still, it remains clear who intends to communicate with whom, who is the sender/receiver, what remains control data for purposes of delivery and what is the delivered data (albeit an outsider might not access it due to encryption).

This pattern is contrasted by the concept of secret communication that deviates from the classic pattern insofar as communication takes a more clandestine way. Clandestineness manifests through stakeholder's nescience of certain aspects of communication or even its mere existence. We identified three means of such secret communication, namely covert channels,

side channels and obfuscation, that are defined and discussed in this section.

A. Covert Channels

The *United States Department of Defense* standard 5200.28-STD defines a covert channel as "*any communication channel that can be exploited by a process to transfer information in a manner that violates the system's security policy*" [47]. Literature highlights further aspects: A covert channel "*exploits a shared resource*" [48], and the used channel "*is not designed to be a communication mechanism*" [48] or "*contrary to design*" [49]. Typically it is of "*malicious or unwanted nature*" [50] and "*can be used to leak information*" [51].

If communication partners want to prevent unauthorized parties from eavesdropping on transmitted data, end-to-end encryption is a practical countermeasure. Covert channels go beyond insofar as their intention is concealment of the communication's mere existence from third parties [50], [52], and are applied in case an observer should not even know that communication is on-going. They exploit senders' degrees of freedom [53] that are also accessible to receivers. Traditional network-based covert channels use for example unused or vaguely defined bits; values that are chosen by senders without having strict criteria (e. g., identification numbers, fragmentation offsets or hop counts); values that are typically not parsed by receivers (e. g., timestamps); checksums in case the payload can be modified in a way to correspond to the checksum value; and timing as IP-based traffic shows indeterministic behavior [50].

Observers are usually unaware of the fact that some of these characteristics can be used to communicate. Nevertheless, even if the method is known, the content of the communication should remain confidential by using encryption and ideally should not be distinguishable from typical (e.g. random) values used in such fields. So the secrecy of the communication should additionally lie in the knowledge of a secret key and not solely in the unawareness of external observers of potential communication channels [54]. Therefore classical cryptographic methods should be applied before a message is encoded in a covert channel.

Cryptography differentiates between symmetric and asymmetric algorithms [55]. Symmetric cryptography uses a single secret key that is used by the sender to encrypt, and by the receiver to decrypt the message accordingly. In contrast, asymmetric cryptography operates with two keys – a private key, and a public key per communication partner. While the first is kept private by all means, the public key is made available to the general public. A sender encrypts a message with the public key, and only the receiver is able to regain the message by decryption with the private key. Asymmetric approaches rely on one-way and trapdoor functions, i. e., functions that are computationally inexpensive in one direction but only solvable in the other direction if additional knowledge (private key) is known.

The differences between symmetric and asymmetric cryptography have an impact on covert channel application scenarios. An autocratic regime might force communication partner A within its jurisdiction to reveal the secret key. In a symmetric key setting the regime is able to decrypt all traffic of both communication partners A and B, even if communication partner B is outside its area of influence. If asymmetric cryptography is used, and the regime gets hold of the private key of A (communication partner in its jurisdiction), but does not know the private key of the communication partner outside B, it can decrypt incoming messages to A (encrypted with the public key of A) but has no possibility to decrypt outgoing traffic from A to B (encrypted with the public key of B). Nevertheless, a regime in possession of a private key of A is able to sign messages and pretend to be A. Asymmetric encryption appears further suitable for encrypting unidirectional data extraction, e. g., from several compromised hosts, via covert channels to one data collector. Whenever data is ready to be delivered to the data collector it is encrypted with the same public key and the reporting hosts do not need a key pair of their own. Beyond, an administrator discovering compromization of his own host (and the public key used for data extraction) is not able to decrypt messages that have been sent from his host, neither to decrypt communication of hosts that are still compromised. However, asymmetric cryptography bears the drawback of being computationally more expensive than symmetric approaches and thus might be more conspicuous for the owner of a compromised host. Furthermore, if signing or mutual authentication is needed, also the reporting hosts need to create key pairs.

Authenticated and eavesdropping-secured key exchange is a vital part of symmetric cryptography, and a major challenge. The Diffie-Hellman protocol is nowadays the major approach to overcome this issue on a non-tap-proof communication link, and is based on the concept of finite cyclic groups [56]. Diffie-Hellman (and also other key exchange protocols like Needham-Schroeder [57]) require a two-way data exchange, i. e., a bidirectional channel – an assumption that does not necessarily hold for covert channels. In unidirectional covert channel settings, the communication partners might perform an out-of-band key exchange. For example, Diffie-Hellman could be performed over regular networking instead of the covert channel. Such an unexpected key exchange might, however, appear illegitimate to an observer, and might be a hint for covert communication. This especially holds for countries prohibiting cryptography, see application scenario *Superficial Compliance to Cryptography Laws* in Section V, as a Diffie-Hellman key exchange is an unambiguous characteristic of cryptography.

With respect to covert channels in cloud computing we further have to highlight two distinct aspects: First, a shared resource in the context of covert channels as highlighted in [48] does not necessarily mean resource sharing in the sense of cloud computing. A shared resource of a covert channel is one that both sender and receiver can access, i. e., read and/or write. However, this resource does not necessarily have to be physically shared in the cloud computing's sense of resource-sharing. I. e., sender and receiver need not be co-resident virtual machines (neighbors) and compete for the resource. For example, on the one hand, CPUs are shared

among virtual machines in the sense of cloud computing. In addition, they represent a shared resource in the sense of secret communication as modulating CPU loads might allow data transmission among neighbors. On the other hand, virtual machines might exploit packet header fields for secret communication, and the latter form the shared resource in the sense of secret communication. However, in comparison to the CPU, the virtual machines do not share the packet header fields in the cloud computing sense. A shared resource in the sense of cloud computing is a more stringent condition than a shared resource in the sense of covert channels; in consequence, covert channels might exist between cloud-internal hosts albeit not residing on the same physical server; the channel might for example exploit packet timing as its shared resource. This fact further implies that dedicated instances, i. e., physical servers that are intended for a single customer and thus not shared with unknown parties, are not an all-embracing countermeasure in order to prevent cloud-based covert channels.

Second, an observer that is looking for suspicious traffic has to reside on the communication path between sender and receiver. For determination of the non-existence of suspicious traffic, the observer has to control all alternative paths that the communication might take. Depending on the observer's power this appears to be a minor or major challenge. In cloud computing such a potential observer is always present with the cloud provider. In dependence of the service model, the provider controls the underlying infrastructure including computing, storage and network facilities, virtualization, the operating system, the platform and/or the applications and traffic has to pass in any case. Summarizing, a cloud provider has the ability to become an utmost powerful adversary. In consequence, selection of a provider is a task of high importance for security. However, today's cloud providers typically refrain from revealing details of their infrastructure or internal processes and selection of a provider is based on rather "weak" criteria like reputation or popularity. Cloud customers sit on the shorter end of the lever.

B. Side Channels

Side channels root in the field of cryptographic engineering and *"exploit characteristic information extracted from the implementation of the cryptographic primitives and protocols. This characteristic information can be extracted from timing, power consumption, or electromagnetic radiation features"* [58]. The classic way of exploiting a side channel is the extraction of a secret key. A recent prominent example is the extraction of the RSA private key from noise that emerges from a laptop during the performance of cryptographic algorithms [59].

But even beyond breaking cryptography, side channels are prevalent whenever an implementation's behavior reveals systems internals that should be kept secret. A well-known side channel is operating system detection (fingerprinting): Although protocols like IP are standardized [60], stack implementations show (subtle) differences in behavior and allow to determine a host's operating system. For example, operating systems initialize the IP *Time to Live (TTL)* field with different values. Adding the measured hop distance to the received TTL allows drawing conclusions on the remote host's operating system [61].

In comparison to the classic communication pattern, side channels are a side effect of the system architecture or implemented algorithms and unintended by the sender. Such channels can leak (confidential) information, and transmitted data is neither encrypted nor otherwise protected due to the channel's unplanned nature.

Cloud computing adds new aspects to such channels' application: Cloud providers conceal their infrastructure and configuration following a security-by-obscurity concept. In-depth and verifiable knowledge on cloud internals remains widely inaccessible for customers, and this black-box approach impedes checks on the provider's compliance with service level agreements. In such scenarios, side channels can be used as a source of information gathering that allow plausibility checks. For example, cloud providers offer dedicated instances[7], i. e., physical servers that just run instances of a single customer to mitigate the threat of co-residency. A customer might use a side channel checking for co-residency as a defensive tool and verify whether there is a stranger's virtual machine on the same physical server [62]. This example also emphasizes a modified standing of side channels. In cloud computing, side channels are not exclusively means of attacking and thus evil, but also serve the benign purposes, e. g., protection of customers against a typically far more powerful provider that might silently disobey service level agreements. This modification goes hand in hand with a change in perspective from building systems as in traditional engineering to discovering phenomena in a way that is comparable to the natural sciences.

C. Obfuscation

Obfuscation aims at anonymous communication by concealment of sender/receiver or hindering their correlation by third-parties [50]. Communication is not fully covert: Observers are generally aware that nodes are participating in communication and use a certain method of obfuscation [18]. However, they are not able to correlate sender and receiver and/or identify them. This assumption holds even in case the observer joins this technique of obfuscation. Observers might further be unable to read the transmitted content due to encryption.

The most prominent example is *Tor* [18]. Based on onion-routing, packets are detoured over relays that are provided by volunteers. Relays decrypt packets to infer the next address and deliver to the next hop. Although aware of the communication, a third party cannot compromise anonymity by simple actions like the provision of single relays. Obfuscation in general relies on confusion due to the Internet's extent, requires high traffic and many users/participating nodes for successful concealment.

With respect to the classic communication pattern that has been presented at the beginning of this section, obfuscated communication is intended, as with covert channels. An observer however is aware that communication is going on, but cannot find out who is communicating with whom. An

[7]https://aws.amazon.com/de/blogs/aws/amazon-ec2-dedicated-instances/

	Covert Channel	*Side Channel*	*Obfuscation*
Intention	yes	no	yes
Hiding Technique	resource unindented for communication	none	large set of nodes or users
Intermediate nodes	optional	optional	required
Content Protection	encryption	none	encryption
Suspicion Level	medium	low	high

TABLE II: Classification of Secret Communication Variants

observer is also unable to decode transmitted data. A pattern of obfuscation is the involvement of intermediate nodes: Sender and receiver seem to maintain a connection to this intermediate, but as a number of nodes do the same, correlation becomes more difficult.

Clouds appear to be a sound substrate for obfuscation - less due to technology than their impact on economy and society. First, obfuscation depends crucially on the number of participants, and cloud services typically have a large user base. As networking is a prerequisite of cloud computing many participants imply much traffic that can be used to hide. Second, this traffic appears in-dubious as the majority of people use the cloud service as intended. Finally, services with a vast user base are unlikely to be blocked. For example, countries applying Internet censorship refrain from blocking clouds as they fear negative impact on commerce and society. This enables activities to counter censorship by moving content to the cloud or using the cloud as a relay [63], [64].

D. Comparison

We defined three means of secret communication, namely covert channels, side channels and obfuscation, and discussed them with respect to classic communication patterns and their application in clouds. Table II summarizes differences between these three (general) kinds of communication on the basis of five discovered major characteristics:

- *Intention for communication* describes whether the data is transmitted by the sender on purpose. This is the case for communication via covert channels and obfuscation techniques, as the communication partners aim to exchange information. In contrast, side channels leak information unintentionally; the sender might not even know that it is transmitting data and provides sensitive information to third parties.
- *Hiding technique* describes the method that hides the exchanged information from potential observers: As side channels are unintended, they do not hide either. Covert channels hide by using a shared resource which is not intended for communication. Examples are CPUs or caches in a system, header fields or packet timing in network protocols. Obfuscation exploits a large set of nodes or users to hide.
- *Intermediate nodes* reside between sender and receiver on the communication path. Obfuscation requires intermediate nodes for concealment. For example, sender and receiver might maintain connections to the same intermediate node, but correlation of these parties is complicated as they are lost in the multitude of other connections. This intermediate node is used as a reflection point for information. An alternative is the redirection of traffic over a number of relays. Covert and side channel do not necessarily require an intermediate node. It is worth noting, however, that intermediate nodes might be detrimental to the quality of side and covert channels, for instance by rewriting packet headers or changing packet timing.

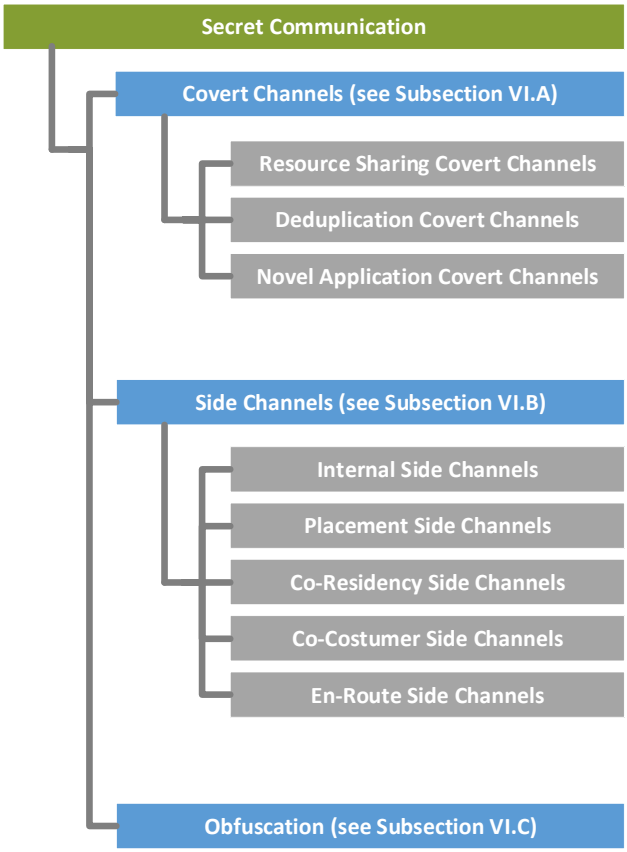


Fig. 2: Taxonomy of Secret Communication in Cloud Computing

- *Content protection* refers to techniques that might be used to prevent others from accessing transmitted information. Covert channels and obfuscation might use encryption, side channels do not include any – again due to their unintended nature. Senders might attempt to close side channels when becoming aware of their existence.
- *Suspicion level* describes the degree to which an observer suspects that communication is taking place. Low means that an observer is unaware of the communication in general, medium means an observer may suspect communication, but cannot find details, and high refers to an observer being (quite) sure that there is communication but cannot find details, either.

With respect to the communication pattern that was presented at the beginning of this section, side channels lack the intention to transmit data; covert channels lead an observer to believe in the non-existence of communication despite the latter might be able to access overt traffic; and obfuscation attempts to hide who is communicating with whom. Finally, a taxonomy is depicted in Figure 2; it distinguishes covert channels in resource sharing, deduplication and novel application covert channels; side channels in internal, placement, co-residency, co-customer and en-route side channels. Details on these sub-categories of covert and side channels are provided in Subsections VI-A and VI-B.

IV. Potentials for Secret Communication

Distinct factors of cloud computing influence networking, and thus provide potential for the establishment of secret communication. These influence factors can be exploited to generate network traffic with certain characteristics like traffic amount, certain header fields or packet timing as well as nodes involved in the communication. Exploiting its influence factors, a sender forms a traffic with certain characteristics in order to deliver the secret message; the receiver in turn interprets these characteristics, and infers this way the secret message. This process is illustrated in Figure 3.

Senders of covert channels aim to deliberately modify these influence factors, while senders of side channels configure them as needed in their environment. In both cases, receivers draw conclusion on the influence factors from the resulting network traffic's characteristics. An obfuscation's sender aims to mimic legitimate traffic and its characteristics.

In detail, we identified the following influence factors on networking. As they can impact traffic characteristics, they bear potential for secret communication in cloud computing.

- Cloud computing requires underlying physical *hardware* and infrastructure that is subject to physical limits. Consequently, cloud computing efforts are also limited.
- *Virtualization* has to partition hardware among virtual instances in a fair and efficient manner, but also has to protect instances from each other facing a natural trade-off.
- *Operating systems* are networking's pivot and support various protocols. Applications utilize them for delivering their messages[8].
- Cloud computing has led to plenty of novel *applications*. These applications provide new features, but also new weaknesses – both forming a substrate for secret communication.
- Cloud services attract large user populations; pervasive *customer behavior* has economic and societal impact bearing potential for secret communication.

Figure 3 depicts the influence factors as well as network traffic characteristics. Further, it highlights that the influence factors are managed either by the cloud provider or the cloud customers; distribution is however dependent on the service model. With management comes the power to influence networking, i. e., the service model defines secret communication's extent and the communication partners. We discuss in the following the influence factors' impact on networking, and sketch their potential exploitation for secret communication in cloud computing.

A. Underlying Physical Hardware and Infrastructure

Clouds ultimately rely on physical computing resources and are thus subject to their physical limitations; although cloud providers put effort in abstraction and concealment of their internal infrastructure, they cannot overcome physical principles. Hardware still has an impact on the service that is accessible by customers, and this becomes visible in manifold ways. For example, a network interface card cannot serve two or more virtual machines at the same time because it is able to send or receive just one packet at a time [65].

Neither are access times detached from their physical background. Responses from distant machines (in remote data centers) take longer than those from one close-by [66]. Round-trip times within a data center's segment or to a virtual machine on the same physical server are lower; and typically, there are also less intermediate hops [67]. Round-trip times might be artificially prolonged by intentionally delaying the response, but not speeded up. These general statements also hold for resources beyond networking. Drive redundancy reduces response times, and fetching information from a number of drives is faster than gaining the same from only one [68].

Clouds reuse known technologies. Providers aim to conceal their utilization or detailed configuration from customers. They frequently prohibit the Internet Control Message Protocol (ICMP) and thus standard tools like *ping* and *traceroute*. Networking however still works as expected and replacement of diagnosis tools by crafted, benign network probes is feasible. Development of such replacements is especially fruitful as knowledge and experience is existent from the pre-cloud era. This is highlighted at the example of the IPv4 *Time To Live* (TTL) field. The amount of intermediate hops can be extracted from the minimal TTL that is required without any need of ICMP messages [69].

In conclusion, layers of abstraction and concealment cannot hide all ground truths of the cloud and such limitations

[8]As an operating system's role does not differ in cloud computing from traditional use, we refrain from highlighting its impact on networking in this section.

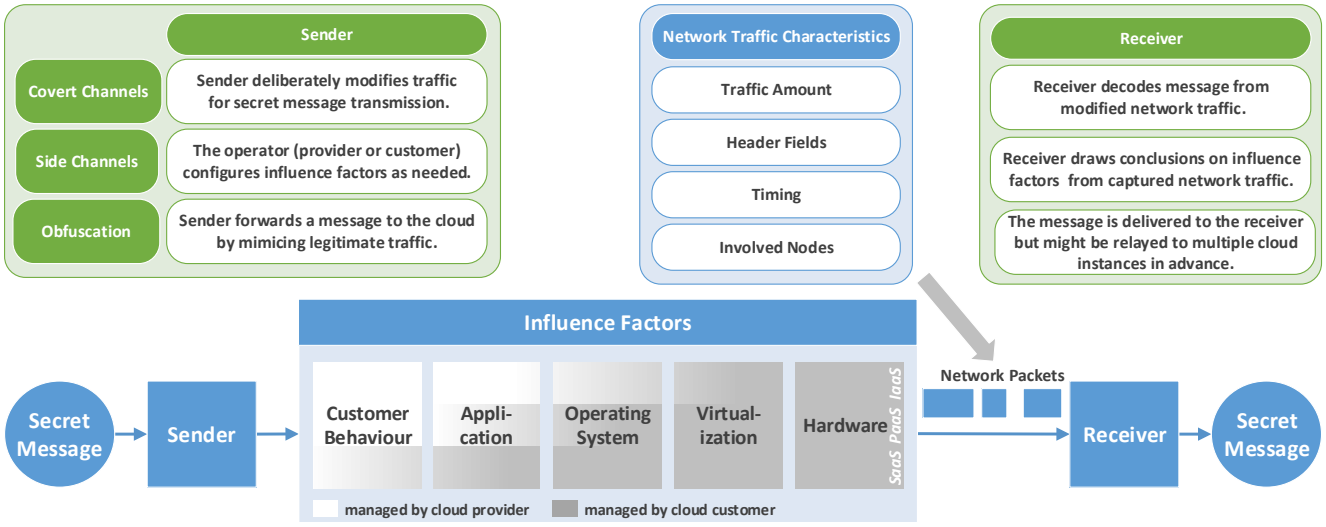


Fig. 3: Potentials for Secret Communication

provide substrate for side channels that reveal internals of the cloud structure. However, these aspects are less usable for covert channels as the underlying hardware cannot be easily modulated.

B. Virtualization: Utilization vs. Isolation

Virtualization means *"abstraction of physical hardware resources"* [12] and allows parallel execution of different operating systems on the same hardware. Virtualization uses hypervisors that control access to physical resources by intercepting requests of hosted virtual machines and mediating them to the hardware [12]. It is a key enabler of multi-tenancy in clouds, and serves a twofold goal: First, it has to partition available hardware resources among tenants efficiently to maximize hardware utilization. The better this goal is met, the higher the cloud provider's monetary revenue. Second, virtualization has to provide isolation for security [70], [71], and separate co-resident instances best possible from each other to prevent assaults. Due to resource costs, isolation is a natural antipode to maximum utilization and *"undermines the cloud's elasticity and business model"* [72].

Beyond, customers demand fair resource partitioning [73], [74]. This includes the provision of resources that customers are paying for; minor impact of neighbors' resource requests on one own's resources and the provision of minimum requirements. In contrast, cloud providers tend to resource oversubscription [75], and the sum of guaranteed resources exceeds the actually available hardware resources. This relies on the assumption that users do not request their full resource share at the same time, and is known from the power grid but may cause severe problem in case the previous assumption is intentionally falsified [76].

This leads to the following conclusion: Cloud providers have little motivation for better isolation as it would reduce monetary revenue. Quite the contrary, they over-subscribe resources to increase revenue and neighbors are likely to influence each other providing an ideal ecosystem for covert and side channels [75]. The potential is emphasized by the following two scenarios.

The hypervisor Xen [77] processes packets in a round-robin manner in regular intervals; incoming packets are prioritized over outgoing. Buffers on the path are limited by size [65]. Packets might be delayed due to a heavy networking neighbor. Heavy networking of an instance causes fluctuations in neighbors' data transmission due to delayed packets [78], or packets might even be dropped due to a full buffer. A neighbor might intentionally exploit this coherence.

Increased demand for a certain resource, may not only influence neighbors with demand for the same resource. We highlight this by an example of networking and CPU usage: Instance A and B experience network load and share the NIC equally. If instance A experiences additional CPU load that requires the full time given for the CPU, B has increased chance of networking. The reason is twofold: First, A requires obviously less of the resource (networking). Second, the resource demand might shift in time and B is provided with the resource for a longer continuous period [79]. Thus, there are also inferences among different resources that might be exploited.

C. Novel Applications in Clouds

Cloud computing reduces start-up costs as no physical equipment anticipating future needs has to be bought. At the same time, flexibility allows easy scale-up and down with immediate needs [80]. Being of comparably low risk, the cloud is an excellent substrate for start-up companies providing all types of novel applications to the market; enhanced by providers' market places where native as well as third-party applications are offered [81]. A number of now popular cloud services started in the data centers of large cloud providers, and are still there, e. g., *Spotify* and *Dropbox* in the *Amazon* cloud.

Cloud applications follow approaches that were unknown before or reuse known technologies in a different environment.

In addition, they connect users that are unrelated to each other to a single central service albeit the application's purpose does not necessarily require this radial topology. Cloud services are accessible via the network and this traffic provides potentials for secret communication. For obfuscation, the cloud may serve as a reflection point. The traffic may alternatively serve as overt channel for the covert channel [50], or reveal internals and thus be a side channel. Examples of such novelties are provided in the following paragraph.

Data deduplication associates users that are unknown to each other due to storing the same file in their personal cloud-synchronized folder in order to optimize storage capacity through the elimination of redundancy [82], [83]. *Push Notification Services* release developers from reliably delivering updates for mobile applications, and allow forwarding of messages by means of an ID making the services a reflection point with a lot of users [84]. The market place bundles services related to a certain cloud on one place. The services are not only from the cloud provider, but also from third parties. While it might *"provide the customer with peace of mind by knowing that all purchases from the vendor's marketplace will integrate [...] smoothly"* [81], the provider does not check these third party offers in detail fostering all types of misuse and slackness [85].

	Covert Channel	*Side Channel*	*Obfuscation*
Industrial Espionage	✓	✓	
Whistleblowing	✓		✓
Censorship Evasion			✓
Exchange of Illegal or Regime-Critical Content	✓		✓
Superficial Compliance to Cryptography Laws	✓		
Gain of Transmission Capacity	✓		
Compliance Checking		✓	
Reconnaissance		✓	
Data extraction from Compromized Hosts	✓		
Malware Communication	✓		✓

TABLE III: Application Scenarios wrt. Types of Secret Communication

D. Customer Behavior and Cloud Population

Clouds attract a high number of users. These users in turn maintain connections to cloud services causing massive network traffic and giving the providers significant market power. While this aspect is more societal than technological, it has particular impact on secret communication for three reasons.

- Maintained connections to these clouds are common and do not appear suspicious. A network administrator might not react to *Twitter* traffic as he might know that some users of his network are actively using this *SaaS* service. Even in case of additionally caused *Twitter* traffic, it might appear harmless and users might believe that this traffic origins from their own account.
- The pervasiveness of cloud traffic decrease the chance of being filtered or censored because users might recognize and condemn the intervention. For example, China refrained a long time from blocking large cloud providers for business reasons, and censored content could spread by moving it into the cloud [63], [64]
- Users maintain connections to the same cloud services albeit the service does not necessarily need to connect users. Thus, these services might be used as reflection point, and indirectly connect users that appear unrelated to each other in the first.

The high number of users is supported by the fact that basically everybody is able to join cloud services with almost no barriers. A lot of services are free of charge, others provide opening offers. However, an adversary is able to subscribe in the same way as an ordinary user does, and to investigate the cloud services in detail for potentials of secret communication. While providers might have a chance to check the users' identity, other users do not and have to trust the provider. Thus, there is no classical perimeter protection that separates the "trusted" inside from the "malicious" outside [70] anymore, and formerly insider attacks become outsider attacks [72].

V. Secret Communication Scenarios

The following list shows typical classes of information for which the communication partners have an incentive to hide from potential observers. Table III provides an overview on these scenarios with respect to applicable types of secret communication.

Industrial Espionage: Intellectual property is an important asset of companies and measures are taken to prevent its disclosure to the public or competitors. Due to measures preventing such content from leaving the company, an inside spy has to use covert channels to evade this barrier; see, e. g., [86]. Conversely, competitors might also spy from outside and gain information from unintended sources of the victim via side channels.

Whistleblowing: Whistleblowers disclose content "*about non-trivial illegality [...] under the control of that organization, to an external entity having potential to rectify the wrongdoing*" [87]. Whistleblowers can use covert channels to secretly transmit information or use obfuscation techniques to conceal their identity.

The scenario is comparable to inside industrial espionage, if the whistleblower reports from inside the organization with protection against leakage. The whistleblower then might use a covert channel. If the whistleblower reports from outside the organization, it suffices to stay anonymous by means of obfuscation.

Censorship Evasion: Some countries apply Internet censorship and block access to certain content [88], [89]. However, obfuscation might redirect traffic over other nodes to evade censorship. Covert channels cannot be used because direct communication (and therefore overt traffic) between the two nodes is impossible; however covert channels can be piggybacked on obfuscated traffic.

Exchange of Illegal or Regime-Critical Content: The Internet serves as a channel for illegal content, e. g., trading of drugs [90] or child pornography [91], and communication partners aim to evade detection. They might apply two approaches that protect them in different ways: Obfuscation protects from the identification of individual perpetrators, although law enforcement, e. g., might order an inquiry against person or persons unknown after seeing the content. With covert channels, even the existence of the transmission is unknown. In non-democratic states, regime-critical content may be penalized and considered *illegal* in the context of local jurisdiction. Thus, we include regime criticism here.

Superficial Compliance to Cryptography Laws: Countries may restrict or prohibit the use of encryption [92], and the use of cryptographic protocols might lead to governmental punishment. If any kind of application of cryptography is penalized, encrypted messages have to be hidden from governmental observers in a covert channel to superficially fulfill the law.

Gain of Transmission Capacity: Covert channels use channels that are unintended for communication, and thus increase the total transmission capacity. More data is transmitted without paying for, and is of interest in case of high transmission costs, e. g., Internet taxes per gigabyte [93]. Obfuscation typically causes overhead in comparison to direct communication between the end nodes, and therefore cannot be used to increase transmission capacity.

Compliance Checking: Service-level agreements are negotiated between cloud providers and customers, but are frequently standardized due to market imbalance. The provider's economic power is typically far higher than the customer's, and checking compliance of the provider by the latter is difficult [94]. Customers may use side channels to check whether measured results are plausible considering the terms of contract.

Reconnaissance: An adversary might aim to discover and get information about a (not yet compromised) victim to tailor the succeeding attack, or place his own virtual machine on the same physical server for a cross-VM attack [16]. The more information gathered, the higher the chance of a successful attack. Therefore, an adversary may exploit side channels as they transmit information that the victim does not intend to disclose.

Data Extraction from Compromised Host: An adversary may aim to leak secret information of a compromised system, e. g., a secret key, without alarming the operators. She might use a covert channel as this is the most secure way of preventing an alarm and have ongoing access to the systems as the operator might otherwise change the key.

Malware Communication: Botnets are networks of nodes that are infected by malware and coordinated by *command and control* structures. These nodes "*contact a command and control (C&C) server to receive instructions or updates*" [95]. Botnet operators aim to evade discovery, and thus conceal their traffic and are moving their infrastructure (partly) into the cloud [96]. Depending on the extent of disguise, they choose obfuscation or covert channels. A botnet that aims to gain control over as many nodes as possible, e. g., for later denial-of-service attacks, might prefer obfuscation [97] to get the best cost-benefit ratio and takes in exchange removal of malware from certain nodes into account. Alternatively, covert channels might be preferred in cases of higher demands on concealment, e. g., in targeted attacks or worms.

VI. Approaches for Secret Communication

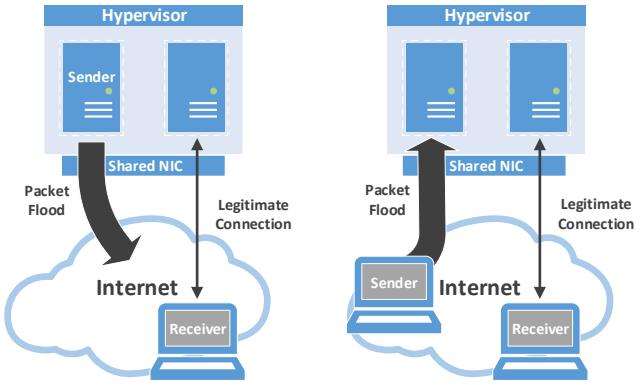
In this section, we describe and classify approaches for secret communication in cloud computing. We group known methods into (1) covert channels, (2) side channels and (3) obfuscation according to the definitions in Section III. As a number of secret communication channels rely on specific, widely unknown cloud-inherent mechanisms, we explain them directly in conjunction with the respective secret communication to ensure readability and compare different approaches with each other. Each type of secret communication is followed by a discussion that further highlights potential directions for future research. Finally, we added a unique identifier consisting of a letter – (c)overt channel, (s)ide channel and (o)bfuscation – in combination with a number to identify every means of secret communication in the following sections. Their identifiers are enclosed in brackets.

A. Approaches for Covert Channels

In the following paragraphs, we describe covert channels in clouds. In a first step, we discuss covert channel that arise from cloud-immanent physical resource sharing, in particular sharing of NICs (*Resource Sharing Covert Channels*) before highlighting channels that exploit deduplication (*Deduplication Covert Channels*), a technology for storage optimization. Finally, we discuss channels that are specific to (novel) cloud applications.

1) Resource Sharing Covert Channels: Resource sharing (in the sense of cloud computing) is cloud-immanent, and provides substrate for covert channels. Multiple virtual machines reside on the same physical server and use the same set of physical resources. This holds for central processing units (CPUs) [98], level 2 (L2) caches [99] and also for networking capabilities. Multiple virtual machines share various network interface cards (NICs) of a physical server, and the number of NICs limits the maximum amount of sent/received packets, i. e., a NIC cannot send/receive more than a single packet simultaneously. Virtual machines' network packets are scheduled and might have to wait before being forwarded to the physical network via one of the NICs due to high load resulting from given hardware limitations. As packet scheduling is a hypervisor task (or even outsourced to hardware assistance[9]),

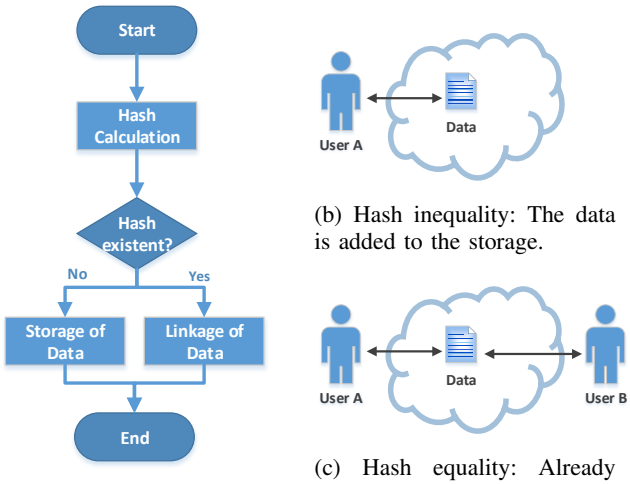
[9]http://www.intel.com/content/www/us/en/network-adapters/virtualization.html



(a) Cloud-based sender and external receiver

(b) Cloud-external sender and receiver

Fig. 4: Covert Channels using Packet Flooding



(a) Hash equality indicates that certain data is already stored.

(b) Hash inequality: The data is added to the storage.

(c) Hash equality: Already available data is linked to a second account.

Fig. 5: Data Deduplication

virtual machines are unaware of these latencies caused by other machines' traffic in general and would also not recognize extra-latencies deliberately caused by a neighbor machine. By deliberately causing high traffic a virtual machine might modulate packet timing of its neighbor in order to establish a covert channel.

Two distinct scenarios, as depicted in Figure 4, are discussed in the literature [100], [101], [69]. They have in common that at least two virtual machines are located on the same physical machine, and share their networking resources[10]. An external node maintains a legitimate connection to one of these virtual machines, and triggers continuous data transmission, e. g., by downloading over and over again a file from a HTTP server. This external node is receiver of both covert channels.

The first attack scenario, see Subfigure 4a, considers the co-resident virtual machine to be the covert channel's sender. This machine causes high traffic, and as the hypervisor cannot send the sender's and its neighbor's packets at the same time, packets are scheduled and have to wait. This behavior increases packet latency, also in the download process of the legitimate connection. The receiver measures the packet arrival rate looking for local extrema, which indicate the channel's symbols [100], [101] (c1). Recently, a stealthier version of this side channel has been proposed [102]; however, transmission capacity has decreased by 75 percent. In the alternative scenario of [69], an external sender influences the latencies of the legitimate connection by flooding the co-resident virtual machine from outside the cloud, see Subfigure 4b (c2).

Comparison: The sender's location has certain implications. (c1) requires the sender to rent a virtual machine, while in (c2) the sender might use an arbitrary co-resident neighbor that does not have to be in her control leading to an increased level of concealment. The cloud-external sender's bandwidth of (c2) might however not be sufficient to generate enough packets in order to cause observable delays. Then, the sender would have to synchronize multiple hosts for joint flooding [69]. In comparison, the sender of (c1) is not restricted by any bandwidth limitations of a physical network. The impact of packet floods on packet latency is dependent on the hypervisor's packet handling behavior. For example, Xen prioritizes incoming packets over outgoing [65], and (c2) might thus outplay (c1) as packets from the adversary are prioritized over legitimate ones.

2) Deduplication Covert Channels: A variety of covert channels in cloud storage solutions arise from data deduplication – a technique to save storage capacity and in certain cases also networking resources. Instead of storing the same data multiple times, just one actual copy is maintained in the storage. Deduplication's working principle is depicted in Figure 5a: First, a hash is calculated over the data that should be stored, and by means of this hash the availability of this data in the storage is checked. Hash inequality implies a data's non-existence in the storage, and the data is added to the storage, see Figure 5b. In case of hash equality, just a link to the already existing file is generated, see Figure 5c.

Different approaches of data deduplication are available. In the target-based approach data deduplication as described in Subfigure 5a is fully performed in the cloud storage leaving the external client unaware of any internal activities; the latter forwards the data just into the cloud. In contrast, source-based deduplication splits the process among the stakeholders: The client calculates the hash and sends it to the cloud, the client delivers the actual data in a second step in case the latter has not been stored in the cloud yet. This way networking can be reduced; the file is solely uploaded to the cloud storage if necessary. However, the client is now able to infer whether the data is already in the cloud or not. If it is already available, he does not have to deliver the data; if the data is not available, he has to deliver it. Cross-user deduplication promises more resource savings for the cloud provider by exploiting data redundancy within several user accounts, i. e., the same data is solely stored once for

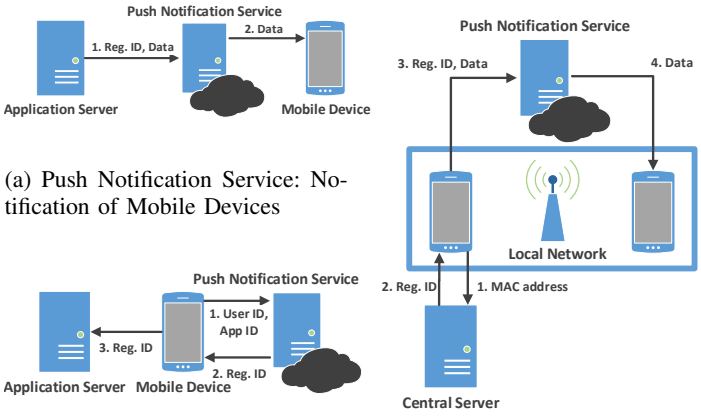
[10]We assume that a physical service has a single physical NIC for clarity of the explanations. Multiple shared NICs might reduce a virtual machine's impact on its neighbor and thus a channel's quality, but do not change a channel's basic principle.

a multitude of users. The resulting source-based cross-user deduplication then becomes a substrate for covert channels among multiple users. The first user has to upload certain data leading to the situation depicted in Subfigure 5b. Subsequent users wishing to upload the same data to their account deliver the hash in a first step. As the data is already available in the storage, their account is solely linked to the data without the need of any further data upload as depicted in Subfigure 5c. From the lack of this upload, the later user might infer the availability of this data in the storage. Data deduplication works on a file or block level. For better readability, we refer to files in the following paragraphs without loss of generality.

Based on these insights, [103] proposes two covert channels: In the first alternative, the sender and receiver have agreed on two files representing 0 and 1. The sender uploads one of these files to the storage, and the receiver is able to check which file is uploaded by attempting to upload both files. If the file representing a 1 is available, the current bit is 1; if the other file is available, the current bit is 0. The sender proceeds with transmission through file deletion and upload of the respective file for the following bit after a certain time interval (c3). The other approach is based on a pre-defined, i. e., fixed, template with a single field for variable input. The receiver is able to learn the file's content by brute-forcing all possibilities and uploading them to the cloud storage. One of the files will be already available in the cloud, and thus no upload from the receiver is necessary (c4). The latter fact allows to infer the field's value as described for example in the following scenario: We assume a cloud customer storing his bank account password in a text document in a StaaS solution. An adversary wishes to know this password, but is not allowed to directly access the document. Thus, the adversary inserts a guess for this password into a text document, and aims to store this file in the StaaS solution. If the assumed password is correct, this file is already available in the storage and the adversary is not asked to upload her file; otherwise the guess is not correct and the adversary has to go on with the next guess. This way, the adversary is finally able to gain the victim's password. Such side channel would also be applicable in guessing sensitive values in highly structured and widely known templates, e. g., medical prescriptions, tax statements, etc.

Comparison: (c3) uses two distinct symbols, but the number of symbols might be easily increased by adding additional files. Using two symbols, one could also reduce the number of files to one; the unavailability of this file would then become one symbol, its existence the other. The number of symbols of (c4) is dependent on the potential values of the field. Both covert channels assume that no other customer uploads one of these files, and it is thus necessary to choose unusual files.

In [104], data deduplication is evaluated using the example of *Dropbox*. This specific storage solution did not delete files from the storage when users did so, and just removed the link between the user account and the actual file. This implies that files remain in the storage, but are not linked to any of the user accounts. Although this prevents the above mentioned covert channels, it enables other approaches: The sender uploads a file, deletes it from its own account and provides the file's hash



(a) Push Notification Service: Notification of Mobile Devices

(b) Push Notification Service: Registration of Mobile Device

(c) Covert Channel: Messaging between two Mobile Devices

Fig. 6: Covert Channel via Push Notification Service

value (out-of-band, e. g., by e-mail) to the intended receiver. The receiver pretends to upload this file, and provides the received hash to the storage provider. Then, his account is linked to the sender's uploaded data, and he is able to access it as any normal file stored in the cloud storage (c5). Beyond, *Dropbox* allowed uploading any file to an arbitrary account solely by using the receiver's host ID. This way, the sender is able to put information directly into the receiver's storage without ever claiming it as its own data (c6).

Comparison: The advantage of (c6) is its increased anonymity in comparison to (c5) as the file is never associated with the sender's account. Even beyond, the sender does not necessarily have to be a regular customer with its own account at the *StaaS* provider. The difference between (c5) and (c6) is thus comparable to (c1) and (c2): (c5, c1) require the adversary to become a cloud customer, while (c6, c2) allow non-customers to be senders as well.

3) Novel Application Covert Channels: Spotique has been presented as a solution for communication with people in local proximity without publishing the current location to a broader audience, e. g., in social networks [84], [105]. It can also be considered as a covert channel as it introduces a communication channel that has not been intended this way. It is based on cloud push notification services like *Google Cloud Messaging*[11] or *Apple Push Notification Service*[12], whose intended functionality is the reliable notification of mobile applications, e. g., of smart phones, in case there are news at the respective application server. In such a case, an application server sends registration IDs identifying the individual mobile devices and the message to the push notification service. The service itself guarantees delivery and sends the message immediately or later in case the addressed device is offline at the moment, see Subfigure 6a. At the mobile device, the operating system forwards the message to the appropriate application.

[11] developer.android.com/google/gcm
[12] developer.apple.com/notifications

The registration IDs are generated in a registration process as depicted in Subfigure 6b. First, the mobile device reports its sender ID as well as the application ID to the push notification service, and receives a registration ID in response. In a second step, the mobile device forwards this registration ID to the application server. If the application server now wishes to notify the mobile devices, it forwards the message and the gained registration ID to the push notification service. The latter then reliably forwards the message to the intended receiver.

Spotique modifies the registration insofar as mobile devices not only forward the registration ID, but also their MAC address to a central server that replaces the application server. The actual communication of this covert channel is depicted in Subfigure 6c: Sniffing on the local network a mobile device gains the MAC address of a node in proximity. To send a message, the sender asks the central server for the registration ID of the sniffed MAC address and sends this ID in combination with its message to the push notification service. The latter reliably forwards the message to the intended receiver. Once the receiver's registration ID is known by the sender, the central server (and knowledge of the receiver's MAC address) is not needed for succeeding messages and the channel can be even used to communicate with nodes that are not in local proximity anymore. This means that a once physically close node remains reachable beyond this first encounter (c7).

Due to being a wide-spread peer-to-peer protocol, *Bittorrent* is increasingly used in *StaaS* and *SaaS* services for data synchronization among their internal nodes. The protocol uses a tracker node that enables peers to locate others. Nodes announce their files available for download at this tracker by sending the torrent's hash file, a peer ID, IP address, port number, etc. The peer ID field is random and can therefore carry covert communication. The receiver accesses the content by requesting available seeders from the tracker. The latter sends all nodes providing the respective download including the IP address and the peer ID. By means of the address, the receiver identifies the covert channel's sender and infers the covert information from the ID field (c8). There exists also a compressed tracker response format, which omits a peer ID and thus impedes this particular covert channel. In this case, the authors propose the use of another field – the alternative address. This field had originally been intended to enable connection passing through Network Address Translation (NAT) devices or proxys [106] (c9).

4) Discussion and Future Research Directions: Literature describes very different covert channel approaches. On the one hand, covert channels (c1, c2) exploit low layers of networking; on the other hand, approaches like (c5, c6) utilize the application layer. In any case, it boils down to (1) a sender that is able to modulate a certain resource (packet timing, stored files, etc.), and (2) a receiver that is able to infer this modulation. This point of view leads to the most obvious ways of covert channel mitigation – removal of (a) the shared resource in total, of (b) the sender's capability of modulation or of (c) the receiver's capability of observation. The latter two are commonly referred to as isolation (among tenants). Indeed, all alternatives are prevalent nowadays, and have their distinct drawbacks for the provider and/or the customers.

The first strategy, i. e., rigid closure of potential covert channels, is common among *IaaS* providers. For higher payment, they provide *dedicated instances*[13] that are guaranteed to be alone on a physical server, or are solely co-resident to other instances of the same customer. This appears to be practicable in utterly sensitive scenarios, but not for the majority of cases as resource pooling and sharing is one of the cornerstones enabling *computing as a utility* for reasonable prices. Dedicated instances do not only have higher hourly fees[14], but a comparably high, hourly registration fee is charged in addition. This fee is independent of the number of running instances, and discriminates low-volume customers.

The more common approach however remains isolation, i. e., tenant behavior must not impact the other virtual machines at all or only minimal. It naturally opposes optimal resource utilization as isolation itself consumes resources. The cloud provider however might favor to lease these resources out to other customers in order to gain higher revenue instead of investing into better security[15]. Finally, both strategies – dedicated instances as well as isolation – protect against other tenants, but not against channels like (c6) that can be performed by practically everbody; the sender does not have to be a regular cloud customer or user.

In consequence, we believe that other directions of mitigation appear more promising, and should be considered in future research. Nowadays strategies focus on hindering covert channels as mentioned above, but development of novel covert channels – especially with the daily introduction of new cloud applications – might be as quick and straightforward as those for video gaming [107]. Like ancient Egypt lived with the annual Nile flooding, one might adapt mitigation strategy similarly for cloud computing, i. e., living with the existence of covert channels, but protecting sensitive data against leakage via such channels. On the one hand, there are hardware-based approaches. For example, [108] protects security-relevant data like keys from unauthorized access by means of hardware-based control flow integrity, [109] proposes the implementation of additional hardware functionality that protects from a compromised hypervisor accessing its guests' memory page tables. On the other hand, there are software-based approaches; [110] partitions applications among more vulnerable public clouds running uncritical portions, and private clouds running more sensitive ones; [111] overlays public clouds with private clouds to meet higher security levels. Approaches like [108] and [110] apparently require a plan of action to distinguish critical from uncritical data, e. g., by means of risk analysis, while [109] and [111] protect all data and might be more favorable for a cloud provider that is unaware of user data details.

Cloud providers tend to camouflage their infrastructure

[13] e. g., https://aws.amazon.com/ec2/purchasing-options/dedicated-instances/

[14] e. g., https://aws.amazon.com/ec2/pricing/

[15] Obviously, a provider is interested in a basic level of security to prevent everyday attacks, but might refrain from investment into more sophisticated solutions. Beyond, providers and customers naturally have contradicting interests. For example, customers wish instances that are resilient to malware; potentially increased resource use by malware however implies more revenue for the provider.

for ostensible reasons of security; but providers consider their knowledge on infrastructure deployment also as their operational value that has to be kept secret. This behavior might increase an adversary's effort to develop covert channels, but also hinders customers from performing a detailed security analysis. This line of action conflicts with security's basic principle of disclosure [54]. Disclosure, however, would lead to substantial benefits for both customers and cloud providers. First, cloud security can benefit from analysis through external, e. g., academic, review as known from the field of cryptography. Second, disclosure might accelerate innovation in general (even beyond the field of security), and lead to the development of novel cloud applications that are in turn hosted at disclosing cloud providers (resulting in additional revenue). For example, car manufacturer *Tesla* declared its patents to be open source for exactly this reason [112], [113]. Especially cloud providers running their own hardware infrastructure do not have to fear new competitors as entry to this highly competitive market requires not only experience, but also immense upfront capital for physical infrastructure. Cloud computing requires a standardized way of disclosure that allows cloud providers to preserve their key assets while being trusted by customers at the same time.

B. Approaches for Side Channels

In this section, we describe cloud-related side channels. For better readability, we call the entity that undeliberately reveals secret information victim, and the entity that accesses this information the adversary. We arrange side channels in five categories according to the secret information that they reveal and prerequisites on placement of the adversary and the victim within the cloud.

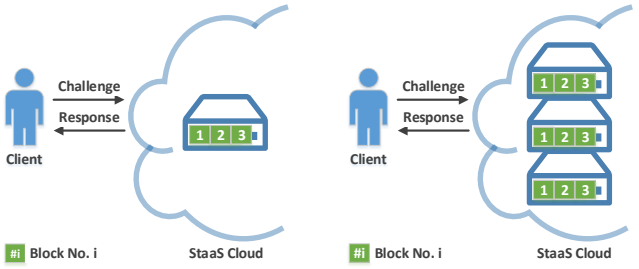
- *Internal side channels* are used by an adversary to reveal aspects about her own virtual machine or account. In this case, the cloud provider is the victim.
- By means of *placement side channels*, an adversary finds out about a victim's placement within the cloud. In the context of cloud computing, co-resident placement is of most interest, i. e., whether two virtual machines reside on the same physical server. Thus, the presented channels aim to discover (or to exclude) co-residency.
- An adversary exploits *co-residency side channels* to gather information about neighbors, i. e., the victim and the adversary are co-resident on the same physical server. Co-residency is a requirement for applying these side channels. Thus, they are typically used after discovering co-residency by means of placement side channels.
- *Co-customer side channels* allow an adversary to reveal information about a victim that resides in the same cloud, but not necessarily on the same physical server, i. e., without the prerequisite of co-residency. This implies that co-residency side channels are a subset of co-customer side channels. Exploiting co-customer side channels, the victim has to be in the same cloud as the adversary. In comparison, co-residency side channels require not only residing in the same cloud but also on the same physical server, i. e., the victim and the adversary being neighbors as defined in Subsection II-B. In consequence, co-customer side channels are applicable to all other customers also including neighbors.
  While placement side channels provide information on other customers' placement, co-residency and co-customer side channels provide information beyond placement like traffic amount, resource use, operating system versions, etc. However they require certain pre-conditions on placement in order to work properly.
- Finally, we discuss *en-route side channels* that are related to website fingerprinting. An adversary exploits network traffic characteristics to infer content that is accessed by the victim, and has therefore reside en-route, but not necessarily in the cloud, to analyze the traffic.

1) Internal Side Channels: Internal side channels represent a novel type of side channels insofar as they provide customers information that are easily accessible in traditional IT landscapes but remain hidden in clouds. By evading cloud services' non-disclosure, they are means of gathering information about the provided infrastructure or providers' compliance with service level agreements.

Being used for data backup, cloud storage providers claim to use fault-tolerance in order to guarantee that data is not lost in case hardware components fail. General best practice in cloud computing as well as in traditional computing is data duplication across different hardware as well as physical data centers, i. e., locations. In traditional environments, a customer could easily check whether multiple drivers are present by taking a look into his servers. The question arises how customers can verify a provider's guarantees in the cloud computing environment.

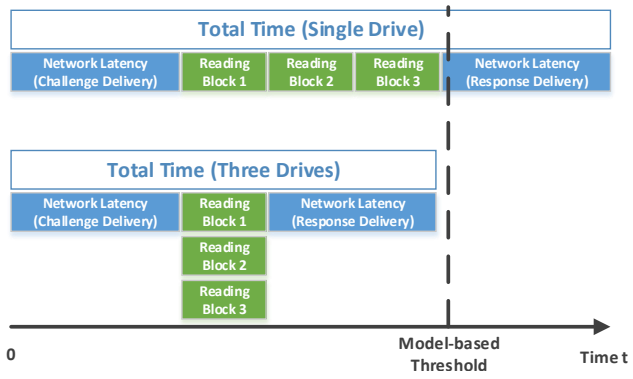
[68] presents remote assessment of fault tolerance by means of a challenge response protocol. First, the client requests the delivery of a number of data blocks from the storage solution. The latter retrieves these blocks. The time it takes to respond reveals the level of the provider's fault tolerance as it is assumed that a higher number of hard drives speeds up the response.

Figure 7 highlights this fact: The client requests three blocks in both cases. In the first scenario of Subfigure 7a, all blocks are stored on a single drive, i. e., no fault-tolerance is provided. Thus, the total time to deliver the response encompasses (1) the latency between client and server to deliver the challenge, (2) three times the period to read a block from the drive and (3) the latency to deliver the response back to the client, see also Subfigure 7c. As the total time is going to be higher than a threshold that is calculated by means of network and drive timing models, the client infers insufficient fault-tolerance. In the alternative scenario of Subfigure 7b, the blocks are available on all drives, i. e., fault-tolerance is present. In this case, the time to deliver the response consists of (1) the latency between client and server, and (3) the latency on the return path. As the three blocks can be fetched from the three drives simultaneously, the (2) period for reading a block just adds once to the total time, see Subfigure 7c. As the total time is now below the threshold, one can deduce that sufficient fault-tolerance is provided (s1).



(a) Provider without Fault-Tolerance: All blocks are stored on the same drive.

(b) Provider with Fault-Tolerance: Blocks are stored on all drives.



(c) Distributed blocks are read simultaneously, and total challenge-response time is below threshold.

Fig. 7: Remote Assessment of Fault Tolerance

[66] complements this approach by assessing whether data is replicated over a number of geographically distributed data centers by measuring response times as well. The authors developed a model that allowed to infer the geographic origin of cloud data in dependence of the response time and the client location (s2).

Comparison: Both channels (s1, s2) measure the same physical quantity, i. e., response time, but infer from the results onto different internal states of the cloud. This raises the question whether both side channels can be applied to the same cloud, and still provide results with high accuracy. Control theory uses the related term *observability* to determine if and to which extent system internals can actually be inferred from available outputs [114]. Applying these concepts to (cloud) computing could help to assess potential impact of competing co-located side channels (s1) and (s2) onto their accuracy.

Cloud providers expand their data centers gradually and in accordance with their economic necessities. For example, a data center might initially consist of hardware A; is expanded by additional physical nodes of hardware B a few years later as the hardware market advanced and faces another expansion with nodes of hardware C. Such hardware upgrades transform data centers from homogeneous to heterogeneous. Variations in hardware performance propagate in virtual machines, which then may provide more or less performance for the same money albeit being of the same instance type. [115], [116] use standard benchmarks, i. e., *UnixBench*, *RAMspeed*, *Bonnie++*, in combination with self-developed benchmarks to evaluate CPU, memory, disk and also network performance. With the gained knowledge, a customer is able to apply cost-saving approaches and choose a machine with better performance. We also refer to [117] for a more sophisticated strategy for performance optimization (s3).

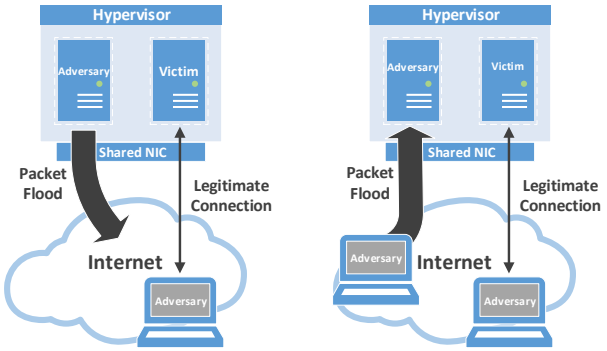
2) Placement Side Channels: In terms of cloud computing, placement refers to the physical server a virtual machine resides at and we refer to the respective side channels as placement side channels. They are of special interest due to making it possible to reveal concrete information on a victim's location in advance of an attack. A unique aspect in clouds is co-residency detection, i. e., inferring whether a certain virtual machine is placed on the same physical server as oneself. This is of interest for adversaries as a number of hypervisor exploits require previous co-residency. Exploitation typically consists of two steps: First, the adversary instantiates a number of virtual machines and checks for co-residency of the adversary's machine with the victim. Second, she executes the exploit to harm her neighbor. On the contrary, co-residency checking is also of interest for benign customers to assess risks from neighbors. The literature describes three different approaches for co-residency checking: (1) address vicinity, (2) measuring round-trip times and (3) the generation of deliberate delays in neighbor traffic.

Following the assumption that nearby nodes have close addresses, [118] developed a rapid test for non-co-residency in the *Amazon* cloud: If two node addresses are not within the same /24 IPv4 network prefix, they are not co-resident (s4). However, two addresses within the same /24 network prefix do not necessarily indicate co-residency, but there is a chance. While this side channel appears inconspicuous due to the sole need of the victim's address, it requires knowledge on the respective cloud provider's addressing strategy as behavior differs among different providers.

While adversaries are looking for other virtual machines on the same physical server in *IaaS* clouds, they look for other platforms on the same virtual machine in *PaaS* clouds. Platforms of different customers share the operating system, and communicate with cloud-external nodes by means of the same IP address. Thus, a simple address comparison reveals co-residency of two platforms [119] (s5). Similarly, co-resident virtual machines in *IaaS* clouds had the same network gateway - the Xen hypervisor's privileged virtual machine running the respective devices drivers (*Dom0*) - in the past [67], one could gain this address by looking into the virtual machine's configuration. But the gateway is hidden now, at least at *Amazon* instances [118].

Round-trip times have been reported to be shorter among neighbors than among arbitrary cloud-based nodes [67] – a fact potentially caused by short-circuiting of the hypervisor for performance reasons [120]. This implies that pinging other virtual machines, e. g, using ICMP Echo Requests and Replies, might be sufficient to identify co-residency. We denote such side channels as (s6).

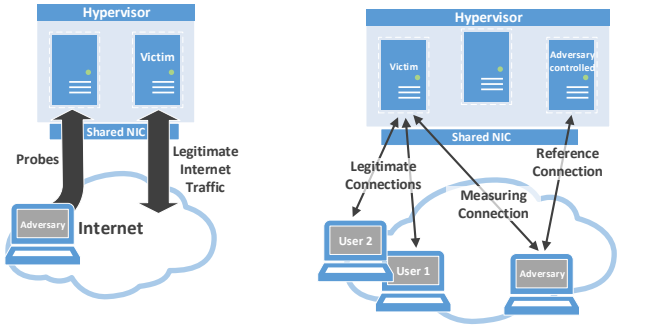
Assuming an adversary checking for co-residency with a victim based on flooding, two alternatives are available as



(a) Cloud-based flooding with two adversary-controlled nodes

(b) Cloud-external flooding with three adversary-controlled nodes

Fig. 8: Side Channels checking Co-Residency of Virtual Machines using Packet Flooding



(a) Side Channel using Low-Bandwidth, High-Frequency Probing

(b) Side Channel using Reference Connection

Fig. 9: Side Channels measuring Neighbor's Traffic

depicted in Figure 8. First, the adversary maintains a legitimate connection from an external node to the victim. At the same time, she floods the network from her cloud-based virtual instance, see Subfigure 8a. In case this flood has a negative impact on the arrival packet rate, the virtual machines are co-resident [100], [101] (s7). Alternatively, the adversary can flood her own virtual machine from another external node, see Subfigure 8b. If extra latency is introduced into the legitimate connection, the virtual machines are again co-resident [69] (s8).

Comparison: The basic principle of (s7) and (s8) is equivalent to their relative covert channels (c1) and (c2). Successful establishment of such covert channels between different nodes of the adversary implies co-residency of her virtual machine with the victim and forms a side channel. The reason lies in the fact that without co-residency these channels would not operate. Comparing the side channels, (s7) requires less nodes that are operated by the adversary in comparison to (s8). On the contrary, (s8) does not necessariliy require an adversary-controlled cloud instance and allows checking whether two virtual machines of strangers are co-resident. Then, an adversary might first attack a less secured neighbor before targeting the actual victim via the hypervisor.

3) Co-Residency Side Channels: Once co-residency is verified, an adversary is interested in further information about her neighbor. On the one hand, this additional information makes it easier to plan an attack; on the other hand, information about the neighbor is also helpful for benign customers, e. g., when deciding whether to stay, or terminate a virtual machine and instantiate a new one due to overlapping resource demands with the neighbor.

In [121], [122], timing side channels in shared event schedulers are examined. In the context of cloud computing, the appropriate scenario is depicted in Subfigure 9a. An adversary aims to infer a victim's (legitimate) networking behavior, e. g., to find usage patterns, peak loads or idle times. Therefore, she sends a low-bandwidth, high-frequency probe like, e. g., ICMP Requests, to the victim's neighbor and awaits the response. As both virtual machines share a packet scheduler the following holds: the more victim traffic, the longer the round-trip times of the probes. We denote side channels of this type with (s9). The impact on the victim's privacy depends on the applied scheduling algorithm. First-come-first-serve provides high performance, but protects privacy least. The other extreme – protecting privacy at the cost of lower performance – is time-division-multiple-access due to its rigid structure of resource assignment [122].

Comparison: The adversary could also directly probe the victim. However, there are scenarios where probing a neighbor appears more attractive: For example, a direct connection to the victim might be suspect. Further, the adversary needs a responding service of the victim. If such a service is not present, she might opt for a neighbor offering such a service. In comparison to (s7) and (s8), (s9) aims to reveal traffic volume, and not co-residency but requires previous co-residency. The exploited principle is similar to the one of covert channels (c1) and (c2). A virtual machine's traffic delays neighbors' traffic. The traffic causing the delay is deliberately caused by the adversary or sender in (c1), (c2), (s7) and (s8), but not in (s9). In (s9), the legitimate traffic delays the adversary's traffic, and thus allows the latter to make assumptions on the victim's networking behavior. As in (s8), the adversary of (s9) does not have to control a cloud-based instance and does not have to register at the cloud provider.

In [100], [101], traffic is measured in a relative way by maintaining a measuring connection to the victim as well as a reference connection to a co-resident neighbor under the adversary's control, see Subfigure 9b. A changing ratio between the connections' throughput indicates a change in the victim's traffic. Any other causes for decreased throughput are filtered: network congestion or change in a co-resident load would impact the reference as well as the measuring connection in the same manner and thus not change the ratio. As its related covert channel (c1) and co-residence detection technology (s7), the packet arrival rate is measured (s10).

A more generic approach to infer neighbor resources is provided in [79]: Using two virtual machines – one is the adversary, the other the victim – that compete for network bandwidth on the same physical node, the authors show that causing CPU bottlenecks to the victim by requesting computing-intensive dynamic webpages (instead of static

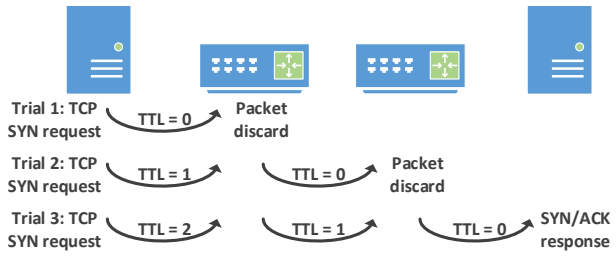


Fig. 10: Side Channel counting Intermediate Hops

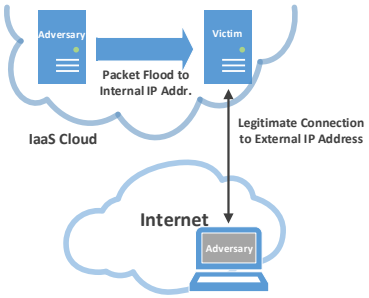


Fig. 11: Side Channel that deanonymizes a cloud-based node's internal address

ones) enables the adversary to increase its network bandwidth. When serving static web pages, the web server is limited by the network capabilities; when serving dynamic ones, it is trapped in the CPU limit and can process less network traffic. In consequence, its neighbor is able to use the residual network capacities. While this is presented as an attack mechanism called resource-freeing attack in [79], it might also serve as a side channel. If one experiences increased resource allocation for oneself while trying to trick a neighbor into such a CPU bottleneck, it is possible to infer the neighbor's resource demand in normal operation (s11).

Comparison: Considering (s10, s11), there are subtle differences in requirements. (s10) requires connecting to a victim via TCP and to regularly download some data whereas (s11) narrows this down to dynamic web pages. These differences are caused by the actual attack vector that is exploited by the respective side channel. (s10) utilizes the bottleneck of networking and just requires much traffic, (s11) exploits the bottleneck of computing at the CPU and thus requires computing-intensive dynamic webpages. Finally, (s11) emphasizes mutual dependence of resources. This mutual dependence appears to be widely ignored with respect to cloud security and might become a starting point for the development of future attacks.

4) Co-Customer Side Channels: In contrast to co-residency side channels, we refer to side channels that reveal information about other cloud customers without requiring co-residency as co-customer side channels.

Data deduplication in *StaaS*, as described in Section VI-A, does not only lead to covert channels, but also to related side channels: A customer can check whether a file has already been uploaded to the cloud and infer, e. g., whether somebody has placed confidential or illegal data in the cloud [103], [104]. Thereby, the algorithm that has already been used in the related covert channels (c3) and (c4), see Subfigure 5a, is followed. The adversary calculates the respective hash of the illegitimate file, and pretends to upload it to the cloud storage. If the file is already available, she is not requested to upload it and can infer its presence in the cloud storage; if she is asked for an upload, the illegitimate file is not available in the cloud. For example, [104] investigated the amount of copyright-protected material stored in *Dropbox* and found that it had been heavily used for storing *piratebay.org* torrents (s12).

IaaS assigns customers the highest level of customization, but also puts great effort into concealment of their internal infrastructure. Following the publication of [67], a number of classic network diagnosis tools were disabled, e. g., traceroute. However, [69] shows that it is still possible to count intermediate hops between two virtual machines. Its principle is highlighted in Figure 10: The Internet Protocol's TTL field is generally decreased by one at every intermediate router, and discarded in case its value equals zero. This behavior prevents packets from endless cycling. For the proposed side channel, the victim performs several trials of TCP connection attempts, and increases the TTL field from zero with every failed trial. The lowest TTL that causes a successful TCP SYN/ACK response is equivalent to the number of intermediate hops (s13). A single intermediate hop between virtual machines might indicate that they reside on the same server rack [118], and this in turn is of interest for power attacks: A simultaneous increase in power demand of machines on the same rack might in turn cause power outage by tripping the circuit breaker and results in denial-of-service [76].

Cloud-based nodes are typically reachable with an external address, but also have an internal address to be reachable for other cloud-based nodes. From an adversary's perspective, the internal address is of greater interest. First, firewalls might allow traffic from cloud-internal addresses while blocking external. Second, an adversary attacking a cloud-based victim from another cloud instance is able to utilize more bandwidth in comparison to attacking from outside as there is typically a high-speed cloud-internal network. For correlating an internal with an external address, [76] as well as [69], propose means of address deanonymization.

In [76], the authors highlight that the Domain Name System (DNS) differentiates between cloud-external and cloud-internal requests. While a request from outside of the cloud provides an external address for a certain domain, queries from another cloud-based node lead to the respective internal address for the same domain due to performance reasons. An adversary might simply correlate these addresses after two DNS requests (s14).

Alternatively, the adversary controls two nodes, at least one of them must be cloud-based to reach nodes by means of their internal address as depicted in Figure 11. The internal node sends a high amount of traffic to an internal address, while the other maintains a legitimate connection to the victim via the external address. If the latter experiences extra latencies, they are caused by the packet flood striking the same virtual

machine. This implies that the two addresses – the external and the just probed internal – belong to the same virtual machine. Otherwise the flooder continues with the next internal address, floods it and sees whether it has impact on the delays in the download process (s15).

Comparison: The setup of (s15) is analog to (c2) and (s8); however, the legitimate connection is maintained to the host that is also probed. (c2) and (s8) split these two connections among two neighbors. Comparing (s14, s15), the adversary requires different prerequisites. (s15) needs an external address, and (educated) guesses on internal addresses or vice verse. (s14) requires a domain name for resolution, and thus no previous knowledge on addresses. Further, mitigation of the latter appears rather simple and a provider could decide to include external addresses in all DNS responses; though, this might cost resources as internal traffic is detoured. Mitigation of the first appears rather lavish as it requires changes of subtle networking details (at the hypervisor or the operating systems). Beyond, the addresses gained from (s14) and (s15) might be beneficial for inferring placement as described with (s4).

IaaS clouds frequently offer a way to publish machine images from third parties on market places that are operated by the cloud providers. These images contain pre-configured operating systems that are prepared by cloud customers and can be instantiated by other customers. However, the latter customer has to trust that the publisher has not left malicious code or back doors in them, as there is no control by the cloud provider. In [123], the authors discovered a number of virtual machines using the same Secure Shell (SSH) host key revealing their origin from the same machine image (s16). This has several drawbacks: First, it is possible to identify virtual machines that are inferred from the same machine image. Second, an adversary might instantiate the image herself, and prepare an attack specific to this image. Third, virtual machines are typically offered in different performance classes, and an adversary might infer a machine's class as certain images support only a limited number of different types.

Cloud providers are strongly interested in operating systems and the exact version installed on hosted virtual machines for the purposes of penetration testing, virtual machine management and digital forensics [124], [125]. While these papers rely on memory-based technologies, traditional network-based operating system fingerprinting is also applicable. Different network stack implementations show subtle difference in the resulting network packets and their timing – despite being standardized – and allow inference on the used operating system. Here, we can distinguish between active approaches with probing respective virtual instances [126], [127] (s17) and passive ones with accessing logs or eavesdropping [128] (s18). However, as network-stack implementations are relatively robust over versions of the same operating system, they are not capable of providing the exact version.

Comparison: (s16) seems to be more advantageous for malicious purposes than (s17) and (s18). Identifying the specific machine image instead of the operating system's version allows an adversary to tailor her attack more accurately to the victim. However, cloud providers could easily mitigate (s16) through checking machine images that are offered via their market place, or by regularly checking the virtual machines that are deployed in their data centers for such striking features as equivalent host keys. In contrast, mitigation of (s17), (s18) appears unlikely as it would require to align all operating systems to behave in exactly the same manner for all details of networking, e. g., initial TTL values, timeouts, etc.

5) En-Route Side Channels: Beyond the above-mentioned categories, there are two side channels related to accessed websites. [129], [130] investigate side channels in *SaaS* providers and discover that modern *SaaS* applications make it possible to infer private information about customers. The reasons are low entropy input, e. g., a limited number of alternatives for marital status, and stateful communication of a known program (despite encryption is used). This is exacerbated by the use of Web 2.0 technologies, e. g., AJAX, where little chunks of information are transmitted separately. By analyzing tax refund or online health services, the program logic and its internal states have been modeled to allow guessing the current position of a user within the model (s19).

[131] presents a technique for inferring the accessed website from frequency distribution of packet sizes despite using privacy enhancing technologies, e. g., Tor. This way an intermediate node, e. g., the cloud provider itself, is able to know which websites were accessed by hosted instances. A similar approach based on the measurement of volume, time and the direction of traffic is presented in [132] (s20).

6) Discussion and Future Research Directions: Clouds appear to be a black box for outsiders, but also for their customers. Thus, side channels have gained momentum with the success of cloud computing. Based on our literature research, we infer that interpretation of side channel results has to be performed with care. On the one hand, different side channels measure the same physical quantity in order to extract different information. For example, (s1) measures file access times to assess a file's existence on multiple drives. (s2) measures the same to infer geographic location, i. e., the data center, of a file that is stored in the cloud. On the other hand, interpretation is often dependent on the cloud provider and side channels cannot be re-used without adaption. For example, (s4) allows statements on co-residency and claims that instances that are not within the same /24 network prefix are definitely not co-resident. This was proven to be valid for *Amazon EC2*, but might not hold true for *Rackspace, Google Compute Engine, Microsoft Azure* or any other provider. Finally, cloud providers evolve their infrastructures without noticing customers. This means that, once revealed, side channels will not necessarily work in the future. For example, *Dropbox* appears to hinder (s12) now by uploading the files in any case [133].

Side channels can also be categorized according to their application scenarios. In Section V, we identified three distinct application scenarios for side channels, namely industrial espionage, compliance checking and reconnaissance. Table IV highlights suitability of presented side channels for these scenarios. Thereby, we separated industrial espionage into two

sub-categories. On the one hand, industrial espionage might be directed against the cloud provider itself to find details about its infrastructure. On the other hand, espionage might target somebody residing in the cloud, i. e., another cloud customer. The table indicates that the application scenarios are congruent with our five categories, e. g., all internal side channels serve industrial espionage against providers, and compliance checking; all placement channels serve espionage against other customers, compliance checking and reconnaissance. The odd ones appear to be (s12, s13) as they additionally serve industrial espionage against the provider in comparison to other co-customer side channels. The reason therefore lies in the distinct nature of these channels. (s12) allows not only to infer whether another customer has uploaded a specific file to the cloud storage, but also enables to find out whether the provider utilizes deduplication per se (or not). (s13) enables not only to infer the hop distance to a potential victim (espionage against customer), but might also be used to scout the whole cloud network infrastructure by launching multiple instances and measuring the hops en-route (espionage against provider).

Further, we conclude from Table IV that most side channels predominately serve malicious purposes, i. e., espionage or reconnaissance for later attacking, and should apparently be mitigated. Internal side channels (s1-s3) represent an exception as these kinds of side channel reveal information like hardware and geographic spread or instance hardware types in a relative manner. This information supports the customer with regard to his own cloud participation, and does not reveal any major trade secrets of cloud providers. Nevertheless, we believe that utilization of side channels for benign compliance checks is just a consequence of customer's lacking insight into the cloud infrastructure. As a remedy, cloud providers might provide standardized functionality that allows their customers to check compliance. Likewise, a detailed disclosure on cloud infrastructure details as recommended in Subsection VI-A might decrease the importance of side channels for compliance checking.

Many side channels have the potential to become a covert channel, or are accompanied by an associated covert channel as emphasized in Section VII. Thus, mitigation for these covert channels can potentially mitigate side channels, as well. At the moment dedicated instances and better isolation are predominant but protect only against side channels requiring co-residency. Disclosure of internal infrastructure involves external review, which increases the likelihood of discovery and mitigation of harmful side channels. However, protection of data from illegitimate memory access as for example described in [109] does not mitigate side channels, as information that is revealed via side channels is frequently not explicitly stored in memory.

Summarizing, cloud computing seems to lack a methodical approach for security. At the moment, an arms race is taking place between research and cloud providers, and security is added in a retrospective manner. I. e., whenever a vulnerability like a side channel is detected it is patched. For example, [67] was the very first paper publishing cloud scouting by means of ordinary diagnosis tools like *ICMP Echo Requests*. In response, several cloud providers totally or partially filter *ICMP* [65]. The *Dropbox* client appears to have changed as well and is now uploading every file to the cloud in order to hinder (s12) [133]. Security however should be considered right from the beginning, i. e., in the design and development phase. The community is challenged to develop a planned approach to guarantee clouds that are secure by design. If security becomes part of the specification, the potential for side channels (but also other kind of secret communication) is likely to decrease. Such an approach might be inspired by Privacy-by-Design [134] that even becomes mandatory according to soon-to-be European legislation [135].

C. Approaches for Obfuscation

Obfuscation techniques provide tools to achieve anonymity for people when using the Internet, e. g., Tor [18]. A major drawback of some of today's anonymity tools is the limited amount of proxy or relay nodes that allow packet rerouting. On the one hand, people want to stay anonymous; but on the other hand, they prefer not to route traffic from other users who want to stay also anonymous. As a solution, [136] proposes *Dust Clouds*, which consist of short-lived virtual machines running *Tor* that are launched at cloud providers. By providing specified machine images, adequate separation between a user's home node and the virtual machine is maintained with low effort. Beyond, the use of machine images makes deployment of Tor nodes a matter of a few mouse clicks. The issue remaining is billing for these virtual machines because billing data still allows linking back to the user and prepaid solutions are not widely available.

Based on the same idea, [137] presents *Cloud-based Onion Routing (COR)* an ecosystem using *Tor* that introduces another layer of indirection between the cloud provider and the communication partners. These intermediate anonymity service providers rent virtual machines from cloud providers, and run the relay nodes. Clients with the wish to anonymously communicate may create their own relay circuit spanning multiple cloud and anonymity service providers. This way communication spans multiple administrative boundaries to overcome trusting a single provider. Payment relies on encrypted tokens (o1). The authors further claim that *Tor* nodes are easily blocked due to having publicly announced, typically static addresses, but are also actively found by the Great Firewall of China [138]. Moving relays to the cloud enables frequent address change, e. g., by terminating a virtual machine and deploying a new one from the same machine image, but censors also tend to refrain from blocking cloud providers due to societal and economic reasons.

Countries applying censorship sometimes even refrain from blocking encrypted services in case they are (economically or socially) important. Circumvention approaches over such services are ignored as they would force censors in computationally expensive traffic analysing techniques, and false positives might hamper innocuous people. *CloudTransport* as presented in [139] is such a hide-within system, and exploits public cloud storage providers as tolerated encrypted services.

The architecture, depicted in Figure 12, connects a user in a censor's jurisdiction with a bridge by means of a shared

	Internal			Placement					Co-Residency			Co-Customer							En-Route	
	s1	s2	s3	s4	s5	s6	s7	s8	s9	s10	s11	s12	s13	s14	s15	s16	s17	s18	s19	s20
Ind. Espionage agst. Provider	✓	✓	✓									✓	✓							
Ind. Espionage agst. Customer				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Compliance Checking	✓	✓	✓	✓	✓	✓	✓	✓												
Reconnaissance				✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

TABLE IV: Suitability of Side Channels wrt. Application Scenarios

ID	*Ref.*	*Exploited Channel*	*Symbol Encoding*	*Sender*	*Receiver*	*Service*	*Trans. Type*	*Bidirect.*
			Resource Sharing Covert Channels					
c1[1]	[100], [101]	NIC sharing	packet arrival rate	neigh.	ext.	IaaS	b	n
c2[1]	[69]	NIC sharing	latency of file download	ext.	ext.	IaaS	b	y
			Deduplication Covert Channels					
c3	[103]	data deduplication	availability of certain files	ext.	ext.	StaaS	u	y
c4	[103]	data deduplication	inserted values in pre-defined template	ext.	ext.	StaaS	u	y
c5	[104]	shared storage	file content	ext.	ext.	StaaS	u	y
c6	[104]	data deduplication	file content	ext.	ext.	StaaS	u	y
			Novel Applications Covert Channels					
c7	[84], [105]	push notification service	notification message content	ext.	ext.	SaaS	u	y
c8	[106]	P2P tracker mechanism	Peer ID field	int.	int.	SaaS[2]	u	y
c9	[106]	P2P tracker mechanism	alternative address field	int.	int.	SaaS[2]	u	y

Reference (Ref.)

Sender/Receiver: cloud provider (prov.), neighbor (neigh.), cloud-based node (int.), external node (ext.), node en-route (en-rou.)
Transmission Type (Trans. Type): unicast (u), multicast (m), broadcast (b)
Bidirectionality (Bidirect.): yes (y), no (n)

[1] Channel is based on co-residency.
[2] P2P protocols are used at various *SaaS* for updating internal servers in datacenters.

TABLE V: Covert Channels in Cloud Computing

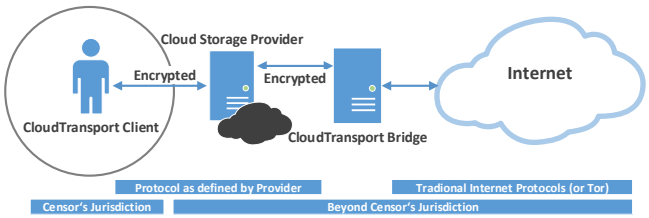


Fig. 12: CloudTransport: Architecture

storage account as a rendezvous point. The bridge is operated outside of the censored area, e. g., by volunteers. The user client wraps its network packets into a file, and uploads the latter to the account. The bridge waits for and reads the file, forwards the packets to the stated target and deletes the file afterwards. The response is delivered back the same way: The bridge writes a file into the account for the user client. Summarizing, network traffic is tunneled over a cloud storage solution.

Comparison: The advantages for the respective communication partners are manifold: First, *CloudTransport* tunnels over available cloud storage protocols and promises to be indistinguishable from ordinary storage use. Approaches that imitate certain protocols instead of hiding within these protocols are

typically more prone to detection. Second, albeit discovering bridges is easy for observers it cannot impede *CloudTransport* by filtering traffic as the clients are connected to the cloud storage and the censor does not want to stop the latter service. Third, bootstrapping is simple by means of an encrypted ticket that is written into a bridge-owned dead drop, and clients do not have to be informed of a bridge's address change (o2). A *Tor* client in contrast requires notification over a changed bridge address as well as attacking a bridge/filtering traffic towards a bridge negatively impacts data transmission.

The other aspect of obfuscation concerns botnets aiming at concealment of their *command and control* communication; otherwise, they risk detection. It has become a recent trend to install communication on social networks with the motivation that this traffic is lost amidst all the legitimate traffic users produce. [140] characterizes a *Twitter*-based communication structure (o3): The bot makes a request to a certain twitter account, and receives an RSS feed containing Base64-encoded text. Decoding the text yields one or more URLs of the URL shortener *bit.ly*. Being redirected to the original address, *zip* files are downloaded, decoded and executed. Gathered information is returned to a botnet's server. [141] even proposes a way to create cover messages for tweets to make them appear plausible to human readers, and thus to stay under the radar of detection.

[142] however use *Pastebin* (o4), a clipboard-like website for sharing text-based content without the need for registration. Users might upload data anonymously, and can share their data with others by means of an URL containing a random ID. The latest posts are also accessible via a time line. A botmaster might access the data by means of the URL that it received from an infected device, or, alternatively, fetch the timeline at a regular interval. While the providers might remove suspect messages, the adversary itself cannot be identified. The server cannot solely be used as a dropzone for stolen data, but also to disseminate commands from the master to the bots.

Comparison: SaaS based *command and control* structures' biggest advantages is their undetectability. It seems impossible to easily and accurately distinguish a botnet's behavior from legitimate one when using certain applications - especially as they have millions of users. Comparing (o3) with (o4) also reveals a difference on an application-level approach of mitigation: While *Twitter* could disable the account that had been hard-coded in (o3), the anonymous way of *Pastebin* hinders such an action.

For mobile botnets, a more sophisticated solution using push notification services is proposed in [143]. Thereby, the botmaster's commands are sent from a *command and control* server, that replaces the application server in Subfigures 6a and 6b, to infected mobile devices (bots) via the push notification service. High stealthiness in general is gained: First, the bot communicates only once directly with the *command and control* server – during registration[16]. Second, neither heartbeat traffic nor command dissemination cause high overhead or suspicious patterns in the traffic in comparison to apps, like mail clients and messengers, that are typically installed on mobile devices. The *command and control* traffic appears as a benign application's one (o5).

[16]In an enhanced architecture, the return path has also been detoured over the push notification service and there is no direct contact. Thereby, the server generates a second push notification account, subscribes itself and delivers the necessary credentials in the malicious application.

	o1	o2	o3	o4	o5
Whistleblowing	✓	✓			
Censorship Evasion	✓	✓			
Exchange of Illegal or Regime-Critical Content	✓	✓			
Malware Communication			✓	✓	✓

TABLE VI: Suitability of Obfuscation wrt. Application Scenarios

Comparison: The question, however, arises why the use of push notification services as *command and control* infrastructure is considered as obfuscation, while *Spotique* (c7) is considered a covert channel in Section VI-A. We classified (o6) as obfuscation because hiding among a large set of nodes and users is the focus, but the technology itself is used as intended – even though in bad faith. In contrast, (c7) creates a communication channel for messaging between mobile devices. The channel is used in a way that has been intended by the service operators.

Discussion and Future Research Directions: Our literature research reveals five approaches of obfuscation that utilize cloud computing. The motivation for going into the cloud however varies; in total, we identified three distinct reasons therefore. First, anonymity tools like *Tor* lack proxies and relays; there are not enough volunteers running such nodes. However, this shortage should not be considered as a shortage of computing power per se; people rather refrain from provision due to potential consequences, e. g., cyber attacks against their own servers or legal prosecution. *Dust Cloud* and *Cloud-based Onion Routing* aim to overcome this issue by means of cloud computing. Cloud computing introduces an additional layer of indistinctness between volunteers (operating and paying for the relay) and the relay itself.

The second type of obfuscation wraps (censored or illegitimate) communication into traffic that appears benign like *CloudTransport* – a well-known strategy to overcome censorship [144]. Wrapping traffic in cloud applications exploits the economic and societal power of cloud computing as popular cloud applications are not blocked – despite being known to be used for overcoming censorship. If a censor totally blocked this cloud service, it would risk severe societal or economic consequences. If it blocked certain connections, the risk of overblocking, i. e., blocking benign connections, would remain.

Third, numerous cloud applications are social apps like *Twitter* that enable people to communicate in an easy way. Entry barriers are kept low to motivate people to join, but might negatively impact users' privacy and security [145]. Such low entry barriers have at least two consequences. On the one hand, a high number of people join the service, and cause lots of application-related traffic. In consequence,

appearance of such traffic does not raise suspicion. On the other hand, entry barriers are low for botmasters and bots as well. This third kind of obfuscation exploits cloud computing due to its freedom of suspicion. Summarizing, motivation for cloud-based obfuscation is threefold: (1) Cloud computing provides an additional layer of anonymity; (2) cloud applications have a high societal or economic value and thus remain uncensored; (3) cloud applications have a high number of users, and thus related traffic appears unsuspicious.

Beyond, application scenarios appear to be a distinctive feature among obfuscation approaches. In Section V, we identified four applications scenarios for obfuscation, namely whistleblowing, censorship evasion, the exchange of illegal or regime-critical content as well as malware communication. Table VI shows the suitability of the five obfuscation approaches with respect to these four application scenarios. Approaches (o1, o2) are appropriate for the scenarios of whistleblowing, censorship evasion and the exchange of illegal or regime-critical content. The remaining approaches (o3 - o5) appear to serve malware communication, i. e., *command-and-control* communication, only. The first group appears to serve predominantly benign purposes and its implementations are of rather high technical finesse in comparison to the latter. Thus, mitigation against obfuscation in general remains a double-edged issue. Following the principles of our Western democracies, (o1 - o2) should rather be supported. Misusing social networks for bots should undoubtedly be mitigated. The development of mitigation against "benign approaches" might even play into the hands of censors.

Accordingly, we identified a number of future research directions with respect to obfuscation. These directions reflect the conflicts between mitigation and support, but also the multidisciplinarity of related application scenarios. Censorship evasion, whistleblowing and regime critics are not solely technical challenges. Thus, evaluation of obfuscation's quality should not only rely on technical analysis; but also on expertise from sociology to answer the impact on society and how users actually use certain solutions, economics to investigate the monetary impact of certain actions and political science to investigate the impact on international relations.

Cloud applications might increase their entry barriers in order to prevent bots from joining their network, e. g., by using Completely Automated Public Turing Tests to Tell Computers and Humans Apart (Captachas) [146]. This, however, increases the effort for ordinary customers as well, and they might decide to use less demanding apps as an alternative. For example, less secured mobile messengers tend to be more popular than more secure alternatives [147], [148]. On the one hand, usable security might be able to develop enhanced security mechanisms for cloud applications that are handier for humans than today's, but protect against the participation of botnets. On the other hand, machine-to-machine communication [149] as caused by botnets is likely to have different characteristics than human communication. Even traffic from botnets mimicking legitimate traffic can be detected using second order statistical metrics [150]. Thus, traffic anomaly detection is worth to be considered in future research for these distinct scenarios. Similarly, a censor could aim to detect traffic anomalies in a cloud application's traffic, and abort suspect connection attempts [144]. The *CloudTransport* approach (o2) wraps packets of HTTP requests into files that are stored at an *StaaS* service. File updates caused by such secret communication might differ from ordinary file updates, and allow inferring improper use of the respective storage service. Beyond, detecting anomalies bears the chance of identifying threats beyond secret communication and eventually mitigating them. For instance the allocation of idle cloud resources can relieve from denial-of-service attacks against individual virtual machines [151]. Integrating models of malicious activities might further improve detection quality [152].

VII. Classification

In this section, we classify covert channels, side channels and obfuscation according to their characteristics.

A. Classification of Covert Channels

We classify covert channels described in Section VI-A according to the following attributes:

Exploited Channel: This attribute names the concrete technology which is exploited in the respective covert channel.

Symbol Encoding: Symbols are used to transmit the information content in telecommunications and define how a secret message is encoded.

Sender/Receiver: The attribute identifies the respective source and sink of the secret channel, which are not necessarily the same as for the overt communication. We distinguish between *cloud providers*, *cloud-based nodes*, *(cloud-)external nodes* and *neighbors* as defined in Subsection II-B. Neighbors are cloud-based nodes but have to fulfill additional constraints with respect to co-residency. Thus, we use the more specific term of neighbor if applicable. Further, we add *nodes en-route*, i. e., nodes between the sender and receiver of an overt channel.

Service Model: We distinguish between *SaaS*, *PaaS*, *IaaS* and *StaaS* as defined in Subsection II-A.

Transmission Type: Transmission type states whether data is transmitted to a single receiver (*unicast*), a group of receivers (*multicast*) or a all receivers simultaneously (*broadcast*) – a distinctive feature also mentioned in [50]. Covert channels do not contain addresses as in classic networking protocols. Thus, we refer to channels that are accessible by everybody as broadcast. Those having a higher barrier than just measuring the respective quantity, e. g., by having some kind of token or a certain placement, are considered as unicast or multicast in dependence of the number of receivers typically fulfilling this requirement.

Bidirectionality: This column gives information on whether a channel is capable of information transfer in both directions, i. e., not only from the sender to the receiver, but also from the receiver to the sender.

The results of classifying the presented covert channels are shown in Table V.

ID	*Ref.*	*Extracted Information*	*Measured Quantity*	*Service Model*
		Internal Side Channels		
s1	[68]	hardware spread of file	file access time	StaaS
s2	[66]	geographic location of file	file access time	StaaS
s3	[117], [115], [116]	instance hardware type	network bandwidth	IaaS
		Placement Side Channels		
s4	[67], [118]	non-co-residency	IP network prefix	IaaS
s5	[119]	co-residency	IP addresses	PaaS
s6	[67]	co-residency	round-trip time	IaaS
s7	[100], [101]	co-residency	packet arrival rate	IaaS
s8	[69]	co-residency	latency in file download	IaaS
		Co-Residency Side Channels		
s9	[121], [122]	neighbor's traffic amount	round-trip time	IaaS
s10	[100], [101]	neighbor's traffic amount	TCP throughput ratio (measurement/reference connection)	IaaS
s11	[79]	neighbor's resource use	network bandwidth	IaaS
		Co-Customer Side Channels		
s12	[103], [104]	file availability	file upload	StaaS
s13	[69]	no. of intermediate hops	min. TTL of successful connection	IaaS
s14	[76]	private address	internal DNS resolving	IaaS
s15	[69]	private address	latency in file download	IaaS
s16	[123]	Amazon Machine Image (AMI) type	SSH host key	IaaS
s17	[126], [127] icw[1] [124], [125]	operating system/version	protocol response behavior (active)	IaaS
s18	[128] icw[1] [124], [125]	operating system/version	protocol behavior (passive)	IaaS
		En-Route Side Channels		
s19	[129], [130]	inserted value in web application	response data size	SaaS
s20	[131], [132]	accessed website	frequency distribution of packets	IaaS

[1] in conjunction with

TABLE VII: Side Channels in Cloud Computing

B. Classification of Side Channels

We categorize side channels described in Section VI-B according to the following attributes:

***Extracted Information**:* This column identifies the information that is gained by the application of the respective side channel.

***Measured Quantity**:* This attribute names the physical quantity that is measured by the side channel's operator to infer the extracted information.

***Service model**:* see Subsection VII-A

The results of side channel classification are shown in Table VII.

C. Potential of Side Channels Becoming Covert Channels

Side channels in general have the potential to become covert channels. Necessary conditions for covert channel existence were proposed in [153], [154]. The defined *Constraint of Communication* condition requires the existence of confidential information at respective nodes. Then, a *Potential for Communication* is identified in case communication from this source to a sink exists. However, the assessment regarding confidential information must be done for every system individually. For the purpose of this paper, we consider the exis-

ID	*Symbol Encoding*	*Sender*	*Receiver*	*Feasibility*	*Trans. Type*	*Bidirect.*	*ID*
			Internal Side Channels				
s1	number of distinct physical disks	prov.	ext.	n	-	-	-
s2	geographic locations	prov.	ext.	y	u	n	f1
s3	physical server of virtual machine	prov.	int.	y	u	n	f2
			Placement Side Channels				
s4	network prefix or physical server of virtual machine[1]	prov.	int.	y	b	n	f3[2]
s5	network address or virtual machine of platform[1]	prov.	ext.	y	b	n	f4[2]
s6	round-trip time or physical server of a virtual machine[1]	prov.	neigh.	y	m	n	f5[2]
s7	physical server of a virtual machine	prov.	ext.	y	b	n	f6[2]
s7	packet arrival rate	int.	ext.	y	b	n	c1[2]
s8	physical server of a virtual machine	prov.	ext.	y	b	n	f7[2]
s8	latency in file download	ext.	ext.	y	b	y	c2[2]
			Co-Residency Side Channels				
s9	traffic amount	neigh. or ext.	ext.	y	b	y	f8[2]
s9	physical server of a virtual machine	prov.	ext.	y	b	n	f9[2]
s10	traffic amount	neigh. or ext.	ext.	y	m	n	f10[2]
s10	physical server of a virtual machine	prov.	ext.	y	m	n	f11[2]
s11	levels of CPU use	ext.	neigh.	y	m	n	f12[2]
s11	physical server of a virtual machine	prov.	neigh.	y	m	n	f13[2]
			Co-Customer Side Channels				
s12	file availability	ext.	ext.	y	u	y	c3,c5,c6
s13	no. of intermediate nodes	prov.	int.	n	-	-	-
s14	internal addresses	prov.	int.	y	b	n	f14
s15	internal addresses	prov.	int.	y	b	n	f15
s16	host keys	int.	int. or ext.	y	b	y	f16
s17	protocol-specific response behavior	int.	int. or ext.	y	b	y	f17
s18	protocol-specific behavior	int.	en-rou.	y	m	n	f18
			En-Route Side Channels				
s19	web-service's input fields	ext.	en-rou.	y	m	n	f19
s20	access of certain websites	int.	en-rou.	y	m	n	f20

Reference (Ref.)
Sender/Receiver: cloud provider (prov.), neighbor (neigh.), cloud-based node (int.), external node (ext.), node en-route (en-rou.)
Feasibility: yes (y), no (n)
Transmission Type (Trans. Type): unicast (u), multicast (m), broadcast (b)
Bidirectionality (Bidirect.): yes (y), no (n)

[1] The provider is able to modify the one or the other; however, the receiver cannot distinguish.
[2] Channel is based on co-residency.

TABLE VIII: Side Channels with Potential of Becoming a Covert Channel

ID	*Ref.*	*Channel*	*Sender*	*Receiver*	*Service Model*	*Trans. Type*	*Bidirect.*
o1	[136], [137]	*Tor* traffic	ext.	ext.	IaaS	u	y
o2	[139]	cloud storage	ext.	ext.	StaaS	u	y
o3	[140]	*Twitter* communication	ext. (bot master)	ext. (bot)	SaaS	b	n[1]
o4	[142]	*Pastebin* communication	ext. (bot)	ext. (bot master)	SaaS	b	y
o5	[143]	push notification service	ext. (bot master)	ext. (bot)	SaaS	u	y

Reference (Ref.)
Sender/Receiver: cloud provider (prov.), neighbor (neigh.), cloud-based node (int.), external node (ext.), node en-route (en-rou.)
Transmission Type (Trans. Type): unicast (u), multicast (m), broadcast (b)
Bidirectionality (Bidirect.): yes (y), no (n)

[1] The respective references describe a unidirectional approach, but bidirectionality would be feasible.

TABLE IX: Obfuscation in Cloud Computing

tence of constrained communication as well as the potential for communication as given.

In the following, we identify the necessities of a side channel to become a potential covert channel:

- At least two distinct **symbols** are necessary to represent 0 and 1.
- A **sender** is somebody who is able to change the symbols.
- The **receiver** is able to access these symbols after transmission to infer the secret message.

Feasibility refers to whether there is potential for a covert channel. Beyond the identification of at least two distinct symbols, sender and receiver, the change in states has to be of adequate effort. We define adequate effort as follows: (1) The change from one symbol to another one is possible within seconds. (2) The change does not require any physical modification.

Transmission Type, Bidirectionality see Subsection VII-A These fields are filled solely in case of feasibility.

The results of potential covert channels based on the presented side channels are provided in Table VIII. We identify 18 out of 20 side channels that have the potential to become a covert channel, five thereof even allow the development of two covert channels each. For every of these future covert channels, we introduce an additional identifier with the initial letter f and a number for unique identification in the remainder of the paper. Alternatively, it is referred to the respective covert channel with the initial letter c if this channel has been already described in the literature.

D. Classification of Obfuscation

We classify obfuscation techniques according to the following characteristics:

Channel: identifies the respective communication channel that is used for obfuscated traffic. In comparison to covert channels, the channel is used as intended.

Sender, Receiver Service Model, Transmission Type and Bidirectional: see Subsection VII-A

The classification results are provided in Table IX.

VIII. Findings and Discussion

In this section, we summarize and discuss our findings. We group our results into three classes: general findings, applicability to communication scenarios, and potential damage. Finally, we summarize identified directions of future research.

A. General Findings

Cloud as Obfuscation Infrastructure: The connection of external nodes via covert channels further means that the cloud serves as an intermediary, which is rather a typical characteristic of obfuscation, see Table II. We believe however that the primary classification criterion for secret communication is the adopted hiding technique. Thus, we consider (c2-c7) as well as (f8) and (f10) as covert channels. They use a resource that is unintended for communication or use this resource in an unintended way for communication. In comparison, obfuscation primarily hides their illegitimate communication in the mass of other (legitimate) communication while using traditional networking, i. e., a resource that is intended for communication. Nevertheless, it must be stated that the above mentioned covert channels include an aspect of obfuscation, which might enable even better hiding.

Interpretation of Side Channel Information: Side channels deliver various kinds of information. However, measured quantities seem ambiguous for some approaches: Table VII reveals that the access time is an indicator of the file's hardware spread (s1), but also of its geographical spread (s2). In a similar way, latency during a download is introduced in the case of probing for co-residency (s8), the neighbor's traffic amount (s9) as well as in the case of address deanonymization (s15). Thus, one has to use care at the interpretation of measurements.

Secret Communication Approaches and Cloud Delivery Models: We found approaches for secret communication in *SaaS*, *PaaS*, *IaaS* and *StaaS* clouds. This indicates that there

is potential for secret communication in all types of clouds; however, the distribution among delivery models varies. Covert channels arise primarily from *StaaS*; side channels from *IaaS* and obfuscation from *SaaS*. Approaches in *PaaS* represent a minority.

Potential for Covert Channels from Side Channels: The classification of covert channels from the literature in Table V has shown that almost all of them still connect external nodes with each other although a number of additional stakeholders are available in cloud environments, e. g., the provider or a neighbor. At the same time, considering potential future covert channels based on today's side channels revealed a wide variety of sender/receiver combinations, see Table VIII. We identified potential covert channels between cloud providers and cloud-external nodes or cloud-based nodes. This leads us to the conclusion that these novel combinations have not been considered in depth by now.

Potential of Side Channels Checking for or Demanding Co-Residency: The analysis of side channels becoming covert channels revealed a distinct group of channels. These side channels have two characteristics: First, they check for co-residency or require previous co-residency in their setup. Second, a third party (actively) performs a certain action that consequently enables to infer information from the measured quantity. These actions encompass flooding from a victim's neighbor (s7), or flooding the neighbor (s8), generating general traffic (s9, s10) or modulate resource use by means of certain network requests (s11).

These channels bear two degrees of freedom (influence factors) each that can be exploited as symbols in a potential covert channel: The cloud provider might move the virtual machine to another physical server and void co-residency indicating a symbol, and moving the machine back indicating another symbol. This leads to a series of covert channels having the provider as a sender, and placement as symbols (f6, f7, f9, f11, f13). Their difference lies in the measured quantity. Alternatively, the third party might act as a sender by performing its action or not (or at another intensity) leading to side channels among internal and external nodes in various constellations (c1, c2, f8, f10, f12). Every of the above mentioned side channels results in two covert channels each. In comparison, (s4-s6) each lead only to a single covert channel (f3-f5) because no third party is required. The provider might change a machine's placement or its network address, but the channel remains anyway the same from the receiver's point of view.

B. Applicability to Communication Scenarios

In the following paragraphs, we aim to discuss the approaches of secret communication with their relevance to the secret communication scenarios defined in Section V.

Secret Communication from Cloud Providers: With the analysis considering side channels becoming potential covert channels, we identified additional stakeholders in communication. One might ask for the adequate application scenario of covert channels with the cloud provider as sender (f1-f7, f11, f13-f15). In this context, we want to highlight their potential in insider industrial espionage or whistleblowing.

Side Channels to get Insight about Cloud Providers: In Section V, we identified three application scenarios for side channels, among them outside compliance checking. In compliance checking, side channels might be used to gain knowledge that is not directly accessible by means of the operated machine or account, and verify a providers' conformance to the service level agreement. Channels (s1) and (s2) are of interest to prove whether the provider has spread data for fault tolerance over various hardware and geographical locations, (s3) allows to choose a certain type of offered hardware. (s12) makes it possible to check whether a *StaaS* provider applies deduplication across multiple customers. (s13) allows to infer a provider's internal network structure. Beyond, side channels might be applied for protection: (s4-s8) allow to ensure that a cloud instance is alone on a physical server, or only co-resident to benign ones, e. g., of the same organization. (s12) makes it possible to check whether a confidential file has been moved into a *StaaS* solution.

Side Channels to Search for Victims: The majority of side channels seem to enable the search for victims to plan a subsequent attack: (s5, s6, s7, s8) enable checking for co-residency or at least for vicinity (s4, s13); (s9-s11) allow to infer a neighbor's traffic load; (s16-s18) reveal a node's operating system or image type; (s14, s15) its internal address. (s12) allows to identify the presence of a victim by the availability of a specific file. (s19, s20) enable to spy on a victim's communication. Industrial espionage against cloud customers might use the same channels, but without the intention of attacking.

Obfuscation Objectives: All types of obfuscation connect external groups, see Table IX. However, approaches (o3-o5) are used for *command and control* between the botmaster and its bots. (o1, o2) have been proposed *with the good in mind*, i. e., censorship evasion and the transmission of regime-critical content; both support unicast and bidirectional communication. In comparison, obfuscation for bot nets appears less sophisticated with respect to their method of concealment as well as the lack of encryption and tend to be rather broadcast (o3,o4). An exemption is the unicast and more sophisticated approach (o5). It might thus be worthwhile to think about it as a means of censorship evasion or transmission of regime-critical content.

Obfuscation for bot nets is based without exception on *SaaS* clouds. It seems that the diversity and seemingly endless supply of *SaaS* is a good substrate for C&C infrastructures of botnets, and (partly) a successor of the formerly used IRC channels.

C. Potential Damage

The potential damage that can be caused by secret communication depends on several factors.

Covert Channel Capacity: First, the question arises how important the prevention of secret communication techniques is. Considering covert channels, their impact depends on their capacity. Low capacity channels only reveal few pieces of information and need a long time for larger amounts of data to be communicated. The literature review revealed only the

capacity of the following two approaches: 4 bits per second (c1), 20 bytes (c8) or 4 byte (c9) per announcement to the tracker. Thus, capacity estimation is important for assessing the potential damage that a covert channel can cause.

Value of Side Channel Information: In the case of side channels, one has to ask oneself whether it is acceptable that the information gained from them is known. While this might be the case for internal side channels that reveal aspects about one's own instance, it is definitely not the case if other customers are measured, as there are privacy concerns. Nevertheless, we argue that compliance checks should steer along a more official course, e. g., by means of standardized tests that are offered by cloud providers.

Dependance on Obfuscation Objectives: For obfuscation, the answer also seems divided: Approaches such as cloud-based *Tor* rather deserves support, while hindering botnets from using benign infrastructure seems worthwhile. However, mitigation remains a double-edged sword and the development of mitigation might play in the hand of censors.

D. Future Research Directions

Clouds are here to stay, and secret communication provides substrate for various directions of future research. We identified the following in the course of the paper.

Resilience against Secret Communication: In order to mitigate secret communication, channels are closed or hindered nowadays. The obstacle, however, is not the latter's mere existence; it is rather their capability of leaking (potentially sensitive) data. Future approaches might tackle mitigation from an alternative point of view, and aim to protect sensitive data from unauthorized access. Gaining resilience against secret communication in such a way appears promising. On the one hand, it does not solely protect against secret communication, but also against other kind of attacks wishing to access this specific data. On the other hand, it appears more practical to protect a chunk of sensitive data than closing all (even yet unknown) means of secret communication.

Designing Secure Clouds: Currently, security is integrated into clouds in a stepwise and retrospective manner leading to a permanent interplay of researchers revealing novel means of secret communication, and the cloud providers closing these specific vulnerabilities. The community will have to develop a more planned approach that considers security right from the design and development phase as a part of the specification; this way the potential for secret communication is likely to decrease. A process towards secure clouds by design might be inspired by Privacy-by-Design approaches that are currently developed in response to a change in European legislation.

Disclosure of Infrastructure Deployment: Cloud providers should be given a strong impetus to disclose their infrastructure (in a standardized and trusted way) as it allows external review in order to identify undesired means of secret communication. We believe that revealing this information implies a boost on innovation in cloud application (and thus also in revenue for respective providers), and further allows to better understand shared responsibilities between providers and customers. Disclosure would further eliminate the application scenario of compliance checking as information is provided anyway. Removing the only application scenario for useful side channels is a strong incentive to close all existing side channels immediately after discovery.

Convergence of Cloud Computing and the Internet of Things: The Internet of Things (IoT) is the next major step of the Internet [155], [156], and integrates multitudes of physical devices from kitchen appliances to cars or factories with the Internet. There is a trend towards converging IoT with cloud computing for at least two reasons. First, devices typically suffer from low computing power – a resource provided by clouds in high quantities. Second, the idea of providing these devices as a service ("Everything as a Service") proliferates. This convergence, however, means that not only more data, but more sensitive data like, e. g., health data from fitness trackers or smart medical devices is available in the cloud. This implies that secret communication, in particular side and covert channels, are likely to become even more interesting for adversaries, and that new channels might come into existence due to newly introduced devices.

IX. Conclusion and Future Work

In this paper, we showed the potentials of secret communication in cloud environments, and described covert channels, side channels and obfuscation techniques with respect to their applicability in several application scenarios. We surveyed current approaches and classified them according to their main characteristics. In addition, we investigated the potential of side channels to become covert channels.

Our work leads to a number of conclusions: Known covert channels mainly exist between cloud-external nodes. Our analysis considering side channels with the potential to become covert channels revealed various alternatives. For example, cloud providers or co-resident virtual machines (neighbors) might become communication partners. We have also identified more potential covert channels in this analysis than already known in the literature.

Side channels deliver diverse information, but measured quantities seem ambiguous in some cases: For example, access time is claimed to reveal hardware as well as geographical spread. It remains unclear whether this ambiguity can be resolved. Obfuscation approaches seem to be largely introduced by botnets for *command and control* infrastructure, while only a minority aim at censorship evasion and the transmission of regime-critical information. The latter however appear more sophisticated from a technological perspective, e. g., a cloud-based approach for *Tor*. The official project motivates the deployment of bridges in the cloud, and even provides a pre-configured machine image.

Based on our insights, we conjecture that there exist even more secret channels than are known today. But due to their secret nature, it is unlikely that we will ever gain an all-encompassing view. A number of approaches will never be published in academia, or elsewhere. However, an in-depth understanding of the potentials that form the substrate

for secret communication is of utmost importance for the development of adequate mitigation strategies. Research has to put in its best effort so that computing as a utility does not become a nightmare due to security issues in general, and secret communication in particular.

ACKNOWLEDGEMENTS

This research was funded by the Austrian Science Fund (FWF): P 26289-N23 and COMET K1, FFG - Austrian Research Promotion Agency.

REFERENCES

[1] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging {IT} platforms: Vision, hype, and reality for delivering computing as the 5th utility," Future Generation Computer Systems, vol. 25, no. 6, pp. 599 – 616, 2009.

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," Commun. ACM, vol. 53, pp. 50–58, Apr. 2010.

[3] B. Martens, M. Walterbusch, and F. Teuteberg, "Costing of cloud computing services: A total cost of ownership approach," in 45th Hawaii International Conference on System Science (HICSS), pp. 1563–1572, Jan 2012.

[4] A. Khajeh-Hosseini, D. Greenwood, and I. Sommerville, "Cloud migration: A case study of migrating an enterprise it system to iaas," in IEEE 3rd International Conference on Cloud Computing (CLOUD), pp. 450–457, July 2010.

[5] M. Chauhan and M. Babar, "Migrating service-oriented system to cloud computing: An experience report," in IEEE International Conference on Cloud Computing (CLOUD), pp. 404–411, July 2011.

[6] A. Bergmayr, H. Bruneliere, J. Canovas Izquierdo, J. Gorronogoitia, G. Kousiouris, D. Kyriazis, P. Langer, A. Menychtas, L. Orue-Echevarria, C. Pezuela, and M. Wimmer, "Migrating legacy software to the cloud with artist," in 17th European Conference on Software Maintenance and Reengineering (CSMR), pp. 465–468, March 2013.

[7] A. Thakar and A. Szalay, "Migrating a (large) science database to the cloud," in 19th ACM International Symposium on High Performance Distributed Computing, pp. 430–434, 2010.

[8] S. Abolfazli, Z. Sanaei, E. Ahmed, A. Gani, and R. Buyya, "Cloud-based augmentation for mobile devices: Motivation, taxonomies, and open challenges," IEEE Communications Surveys & Tutorials, vol. 16, no. 1, pp. 337–368, 2014.

[9] A. u. R. Khan, M. Othman, S. A. Madani, and S. U. Khan, "A survey of mobile cloud computing application models," IEEE Communications Surveys & Tutorials, vol. 16, no. 1, pp. 393–413, 2014.

[10] Z. Sanaei, S. Abolfazli, A. Gani, and R. Buyya, "Heterogeneity in mobile cloud computing: Taxonomy and open challenges," Communications Surveys & Tutorials, IEEE, vol. 16, pp. 369–392, First 2014.

[11] P. Dorey and A. Leite, "Commentary : Cloud computing a security problem or solution?," Information Security Technical Report, vol. 16, no. 34, pp. 89 – 96, 2011.

[12] G. Pék, L. Buttyán, and B. Bencsáth, "A survey of security issues in hardware virtualization," ACM Comput. Surv., vol. 45, pp. 40:1–40:34, July 2013.

[13] M. D. Ryan, "Cloud computing security: The scientific challenge, and a survey of solutions," Journal of Systems and Software, vol. 86, no. 9, pp. 2263 – 2268, 2013.

[14] A. N. Toosi, R. N. Calheiros, and R. Buyya, "Interconnected cloud computing environments: Challenges, taxonomy, and survey," ACM Comput. Surv., vol. 47, pp. 7:1–7:47, May 2014.

[15] C. Rong, S. T. Nguyen, and M. G. Jaatun, "Beyond lightning: A survey on security challenges in cloud computing," Computers and Electrical Engineering, vol. 39, no. 1, pp. 47 – 54, 2013.

[16] Z. Xiao and Y. Xiao, "Security and privacy in cloud computing," Communications Surveys & Tutorials, IEEE, vol. 15, pp. 843–859, Second 2013.

[17] Y. Chen, V. Paxson, and R. H. Katz, "Whats new about cloud computing security," University of California, Berkeley Report No. UCB/EECS-2010-5, vol. 20, 2010.

[18] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in 13th USENIX Security Symposium, 2004.

[19] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," Journal of Internet Services and Applications, vol. 1, no. 1, pp. 7–18, 2010.

[20] J. Geelan, "Twenty-one experts define cloud computing." http://cloudcomputing.sys-con.com/node/612375/. Accessed: 2015-02-10.

[21] P. Mell and T. Grance, The NIST definition of cloud computing. 2011.

[22] T. Mather, S. Kumaraswamy, and S. Latif, Cloud security and privacy: an enterprise perspective on risks and compliance. O'Reilly Media, 2009.

[23] L. Columbus, "Roundup of cloud computing forecasts and market estimates, 2015." http://www.forbes.com/sites/louiscolumbus/2015/01/24/roundup-of-cloud-computing-forecasts-and-market-estimates-2015/. Accessed: 2015-08-07.

[24] J. D'Onfro, "Amazon's cloud revenue is bigger than its four closest competitors combined." http://uk.businessinsider.com/aws-revenue-is-bigger-than-its-four-closest-competitors-combined-2015-4. Accessed: 2015-07-02.

[25] L. Columbus, "Cloud computing adoption continues accelerating in the enterprise." http://www.forbes.com/sites/louiscolumbus/2014/11/22/cloud-computing-adoption-continues-accelerating-in-the-enterprise/. Accessed: 2015-08-07.

[26] S. Marston, Z. Li, S. Bandyopadhyay, J. Zhang, and A. Ghalsasi, "Cloud computing the business perspective," Decision Support Systems, vol. 51, no. 1, pp. 176 – 189, 2011.

[27] G. J. Popek and R. P. Goldberg, "Formal requirements for virtualizable third generation architectures," Commun. ACM, vol. 17, pp. 412–421, July 1974.

[28] G. Malkin, "Internet Users' Glossary." RFC 1983, Aug. 1996.

[29] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A break in the clouds: Towards a cloud definition," SIGCOMM Comput. Commun. Rev., vol. 39, pp. 50–55, Dec. 2008.

[30] S. Leimeister, M. Böhm, C. Riedl, and H. Krcmar, "The business perspective of cloud computing: Actors, roles and value networks.," in European Conference on Information Systems (ECIS), 2010.

[31] L. M. Kaufman, "Data security in the world of cloud computing," IEEE Security & Privacy, vol. 7, no. 4, pp. 61–64, 2009.

[32] M. Jensen, J. Schwenk, N. Gruschka, and L. L. Iacono, "On technical security issues in cloud computing," in Cloud Computing, 2009. CLOUD '09. IEEE International Conference on, pp. 109–116, 2009.

[33] R. L. Krutz and R. D. Vines, Cloud Security: A Comprehensive Guide to Secure Cloud Computing. Indianapolis, IN: Wiley. John Wiley & Sons, 2010.

[34] K. Popovi and . Hocenski, "Cloud computing security issues and challenges," in MIPRO, 2010 Proceedings of the 33rd International Convention, pp. 344–349, 2010.

[35] H. Takabi, J. B. Joshi, and G.-J. Ahn, "Security and privacy challenges in cloud computing environments," IEEE Security & Privacy, vol. 8, no. 6, pp. 24–31, 2010.

[36] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," Journal of Network and Computer Applications, vol. 34, no. 1, pp. 1 – 11, 2011.

[37] D. Zissis and D. Lekkas, "Addressing cloud computing security issues," Future Generation Computer Systems, vol. 28, no. 3, pp. 583 – 592, 2012.

[38] G. Aceto, A. Botta, W. de Donato, and A. Pescap, "Cloud monitoring: A survey," Computer Networks, vol. 57, no. 9, pp. 2093 – 2115, 2013.

[39] V. D. Piccolo, A. Amamou, K. Haddadou, and G. Pujolle, "A survey of network isolation solutions for multi-tenant data centers.," IEEE Communications Surveys & Tutorials, 2016.

[40] I. Ahmad, S. Namal, M. Ylianttila, and A. Gurtov, "Security in software defined networks: A survey," IEEE Communications Surveys & Tutorials, vol. 17, pp. 2317–2346, Fourthquarter 2015.

[41] D. B. Rawat and S. R. Reddy, "Software defined networking architecture, security and energy efficiency: A survey," IEEE Communications Surveys & Tutorials, 2016.

[42] S. Khan, A. Gani, A. A. Wahab, M. Guizani, and M. K. Khan, "Topology discovery in software defined networks: Threats, taxonomy, and state-of-the-art," IEEE Communications Surveys & Tutorials, 2016.

[43] S. Scott-Hayward, S. Natarajan, and S. Sezer, "A survey of security in software defined networks," IEEE Communications Surveys & Tutorials, vol. 18, pp. 623–654, Firstquarter 2016.

[44] Q. Yan, F. R. Yu, Q. Gong, and J. Li, "Software-defined networking (sdn) and distributed denial of service (ddos) attacks in cloud computing environments: A survey, some research issues, and challenges," IEEE Communications Surveys & Tutorials, vol. 18, pp. 602–622, Firstquarter 2016.

[45] C. Colman-Meixner, C. Develder, M. Tornatore, and B. Mukherjee, "A survey on resiliency techniques in cloud computing infrastructures and applications," IEEE Communications Surveys & Tutorials, pp. 1–1, 2016.

[46] S. Pearson and M. Shen, Yunand Mowbray, Cloud Computing: First International Conference, CloudCom 2009, Beijing, China, December 1-4, 2009. Proceedings, ch. A Privacy Manager for Cloud Computing, pp. 90–106. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.

[47] D. C. Latham, Trusted Computer System Evaluation Criteria. 1986.

[48] S. Cabuk, C. E. Brodley, and C. Shields, "Ip covert channel detection," ACM Trans. Inf. Syst. Secur., vol. 12, pp. 22:1–22:29, Apr. 2009.

[49] I. Moskowitz and M. Kang, "Covert channels-here to stay?," in 9th Annual Conference on Safety, Reliability, Fault Tolerance, Concurrency and Real Time, Security, pp. 235–243, Jun 1994.

[50] S. Zander, G. Armitage, and P. Branch, "A survey of covert channels and countermeasures in computer network protocols," Communications Surveys & Tutorials, IEEE, vol. 9, pp. 44–57, Third 2007.

[51] S. Cabuk, C. E. Brodley, and C. Shields, "Ip covert timing channels: Design and detection," in 11th ACM Conference on Computer and Communications Security, pp. 178–187, 2004.

[52] W. Mazurczyk, S. Wendzel, S. Zander, A. Houmansadr, and K. Szczypiorski, Information Hiding in Communication Networks: Fundamentals, Mechanisms, and Applications. Wiley-IEEE Press, 2016.

[53] S. Wendzel, S. Zander, B. Fechner, and C. Herdin, "Pattern-based survey and categorization of network covert channel techniques," ACM Comput. Surv., vol. 47, pp. 50:1–50:26, Apr. 2015.

[54] A. Kerckhoffs, "La cryptographie militaire," Journal des sciences militaires, vol. 9, pp. 5–38, 1883.

[55] B. Schneier, Applied Cryptography: protocols, algorithms and source code in C. John Wiley & Sons, 1996.

[56] W. Diffie and M. Hellman, "New directions in cryptography," IEEE Transactions on Information Theory, vol. 22, pp. 644–654, Nov 1976.

[57] R. M. Needham and M. D. Schroeder, "Using encryption for authentication in large networks of computers," Commun. ACM, vol. 21, pp. 993–999, Dec. 1978.

[58] T. Caddy, "Side-channel attacks," in Encyclopedia of Cryptography and Security, pp. 1204–1204, Springer US, 2011.

[59] D. Genkin, A. Shamir, and E. Tromer, "RSA key extraction via low-bandwidth acoustic cryptanalysis," in CRYPTO 2014, pp. 444–461, 2014.

[60] J. Postel, "Internet Protocol." RFC 791 (INTERNET STANDARD), Sept. 1981. Updated by RFCs 1349, 2474, 6864.

[61] nmap, "Tcp/ip fingerprinting methods supported by nmap." http://nmap.org/book/osdetect-methods.html. Accessed: 2015-06-25.

[62] Y. Zhang, A. Juels, A. Oprea, and M. Reiter, "Homealone: Co-residency detection in the cloud via side-channel analysis," in Security and Privacy (SP), 2011 IEEE Symposium on, pp. 313–328, May 2011.

[63] J. South, "Punching a hole in the great firewall." http://www.chinafile.com/Punching-Hole-Great-Firewall. Accessed: 2015-06-26.

[64] E. Dou and A. Barr, "U.s. cloud providers face backlash from china's censors." http://www.wsj.com/articles/u-s-cloud-providers-face-backlash-from-chinas-censors-1426541126. Accessed: 2015-06-26.

[65] J. Ullrich and E. Weippl, "Protection through isolation: Virtues and pitfalls," in The Cloud Security Ecosystem - Technical, Legal, Business and Management Issues (R. Ko and R. Choo, eds.), Elsevier/Syngress, 2015.

[66] K. Benson, R. Dowsley, and H. Shacham, "Do you know where your cloud files are?," in 3rd ACM Cloud Computing Security Workshop, pp. 73–82, 2011.

[67] 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 16th ACM Conference on Computer and Communications Security, pp. 199–212, 2009.

[68] K. D. Bowers, M. van Dijk, A. Juels, A. Oprea, and R. L. Rivest, "How to tell if your cloud files are vulnerable to drive crashes," in 18th ACM Conference on Computer and Communications Security, pp. 501–514, 2011.

[69] A. Herzberg, H. Shulman, J. Ullrich, and E. Weippl, "Cloudoscopy: Services discovery and topology mapping," in ACM Cloud Computing Security Workshop, pp. 113–122, 2013.

[70] L. M. Kaufman, "Can public-cloud security meet its unique challenges?," IEEE Security & Privacy, vol. 8, no. 4, pp. 55–57, 2010.

[71] A. Roy, S. Sarkar, R. Ganesan, and G. Goel, "Secure the cloud: From the perspective of a service-oriented organization," ACM Comput. Surv., vol. 47, pp. 41:1–41:30, Feb. 2015.

[72] A. Aviram, S. Hu, B. Ford, and R. Gummadi, "Determinating timing channels in compute clouds," in Proceedings of the 2010 ACM Workshop on Cloud Computing Security Workshop, CCSW '10, pp. 103–108, 2010.

[73] A. Ghodsi, M. Zaharia, B. Hindman, A. Konwinski, S. Shenker, and I. Stoica, "Dominant resource fairness: Fair allocation of multiple resource types," in USENIX Symposium on Networked Systems Design and Implementation (NSDI), 2011.

[74] D. Shue, M. J. Freedman, and A. Shaikh, "Performance isolation and fairness for multi-tenant cloud storage," in Presented as part of the 10th USENIX Symposium on Operating Systems Design and Implementation (OSDI 12), (Hollywood, CA), pp. 349–362, USENIX, 2012.

[75] J. Szefer, E. Keller, R. B. Lee, and J. Rexford, "Eliminating the hypervisor attack surface for a more secure cloud," in Proceedings of the 18th ACM Conference on Computer and Communications Security, CCS '11, pp. 401–412, 2011.

[76] Z. Xu, H. Wang, Z. Xu, and X. Wang, "Power Attack: An INcreasing Threat to Data Centers," in Network and Distributed System Security (NDSS) Symposium, NDSS 14, February 2014.

[77] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," SIGOPS Oper. Syst. Rev., vol. 37, pp. 164–177, Oct. 2003.

[78] B. Adamczyk and A. Chydzinski, "On the performance isolation across virtual network adapters in xen," in 2nd International Conference on Cloud Computing, GRIDs, and Virtualization (CLOUD COMPUTING), pp. 222–227, 2011.

[79] V. Varadarajan, T. Kooburat, B. Farley, T. Ristenpart, and M. M. Swift, "Resource-freeing attacks: Improve your cloud performance (at your neighbor's expense)," in ACM Conference on Computer and Communications Security, pp. 281–292, 2012.

[80] B. Iyer, "Why cloud technology is the smart move right from start up." http://www.entrepreneur.com/article/241914. Accessed: 2015-07-02.

[81] M. Rouse, "Cloud marketplace." http://searchcloudprovider.techtarget.com/definition/cloud-marketplace. Accessed: 2015-07-02.

[82] N. Mandagere, P. Zhou, M. A. Smith, and S. Uttamchandani, "Demystifying data deduplication," in ACM/IFIP/USENIX Middleware '08, pp. 12–17, 2008.

[83] J. a. Paulo and J. Pereira, "A survey and classification of storage deduplication systems," ACM Comput. Surv., vol. 47, pp. 11:1–11:30, June 2014.

[84] D. Namiot and M. Sneps-Sneppe, "Local messages for smartphones," in Conference on Future Internet Communications (CFIC), pp. 1–6, May 2013.

[85] M. Huber, M. Mulazzani, S. Schrittwieser, and E. Weippl, "Appinspect: Large-scale evaluation of social networking apps," in Proceedings of the First ACM Conference on Online Social Networks, COSN '13, (New York, NY, USA), pp. 143–154, ACM, 2013.

[86] Reuters, "Snowden says nsa engages in industrial espionage." http://www.reuters.com/article/2014/01/26/us-security-snowden-germany-idUSBREA0P0DE20140126. Accessed: 2015-02-11.

[87] P. Jubb, "Whistleblowing: A restrictive definition and interpretation," Journal of Business Ethics, vol. 21, no. 1, pp. 77–94, 1999.

[88] S. Aryan, H. Aryan, and J. A. Halderman, "Internet censorship in iran: A first look," in 3rd USENIX Workshop on Free and Open Communications on the Internet, 2013.

[89] A. Chaabane, T. Chen, M. Cunche, E. De Cristofaro, A. Friedman, and M. A. Kaafar, "Censorship in the wild: Analyzing internet filtering in syria," in Conference on Internet Measurement Conference, pp. 285–298, 2014.

[90] N. Christin, "Traveling the silk road: A measurement analysis of a large anonymous online marketplace," in 22nd International Conference on World Wide Web, pp. 213–224, 2013.

[91] BBC, "Child abuse sites on tor compromised by malware." http://www.bbc.com/news/technology-23573048. Accessed: 2015-02-11.

[92] B.-J. Koops, "Crypto law survey." http://www.cryptolaw.org/. Accessed: 2015-02-11.

[93] BBC, "Hungary internet tax cancelled after mass protests." http://www.bbc.com/news/world-europe-29846285. Accessed: 2015-02-11.

[94] A. Sunyaev and S. Schneider, "Cloud services certification," Commun. ACM, vol. 56, pp. 33–36, Feb. 2013.

[95] S. S. Silva, R. M. Silva, R. C. Pinto, and R. M. Salles, "Botnets: A survey," Computer Networks, vol. 57, no. 2, pp. 378 – 403, 2013.

[96] X. Han, N. Kheir, and D. Balzarotti, "The role of cloud services in malicious software: Trends and insights," in Detection of Intrusions and Malware, and Vulnerability Assessment, 2015.

[97] D. Brown, "Resilient botnet command and control with tor." http://conference.hitb.org/hitbsecconf2010kul/materials/D2T1%20-%20Dennis%20Brown%20-%20Botnet%20Command%20and%20Control%20with%20Tor.pdf. Accessed: 2015-10-29.

[98] K. Okamura and Y. Oyama, "Load-based covert channels between xen virtual machines," in Proceedings of the 2010 ACM Symposium on Applied Computing, SAC '10, (New York, NY, USA), pp. 173–180, ACM, 2010.

[99] 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 2011 ACM Workshop on Cloud Computing Security Workshop, CCSW '11, pp. 29–40, 2011.

[100] A. Bates, B. Mood, J. Pletcher, H. Pruse, M. Valafar, and K. Butler, "Detecting co-residency with active traffic analysis techniques," in ACM Cloud Computing Security Workshop, pp. 1–12, 2012.

[101] A. Bates, B. Mood, J. Pletcher, H. Pruse, M. Valafar, and K. Butler, "On detecting co-resident cloud instances using network flow watermarking techniques," International Journal of Information Security, vol. 13, no. 2, pp. 171–189, 2014.

[102] K. Block and G. Noubir, "Return of the covert channel, data center style," in Proceedings of the 2015 ACM Workshop on Cloud Computing Security Workshop, CCSW '15, pp. 17–28, 2015.

[103] D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Side channels in cloud services: Deduplication in cloud storage," Security & Privacy, IEEE, vol. 8, pp. 40–47, Nov 2010.

[104] M. Mulazzani, S. Schrittwieser, M. Leithner, M. Huber, and E. Weippl, "Dark clouds on the horizon: Using cloud storage as attack vector and online slack space," in USENIX Security, 8 2011.

[105] M. Sneps-Sneppe and D. Namiot, "Spotique: A new approach to local messaging," in Wired/Wireless Internet Communication, vol. 7889, pp. 192–203, 2013.

[106] J. Desimone, D. Johnson, B. Yuan, and P. Lutz, "Covert channel in the bittorrent tracker protocol," in International Conference on Security and Management (SAM'12), 2012.

[107] B. Hahn, R. Nithyanand, P. Gill, and R. Johnson, "Games without frontiers: Investigating video games as a covert channel," in 1st IEEE European Symposium on Security and Privacy, 2016.

[108] J. S. Jang, S. Kong, M. Kim, D. Kim, and B. B. Kang, "Secret: Secure channel between rich execution environment and trusted execution environment," in 22nd Annual Network and Distributed System Security Symposium (NDSS), 2015.

[109] S. Jin, J. Ahn, S. Cha, and J. Huh, "Architectural support for secure virtualization under a vulnerable hypervisor," in Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture, MICRO-44, pp. 272–283, 2011.

[110] M. Smit, M. Shtern, B. Simmons, and M. Litoiu, "Partitioning applications for hybrid and federated clouds," in Proceedings of the 2012 Conference of the Center for Advanced Studies on Collaborative Research, CASCON '12, pp. 27–41, 2012.

[111] M. Shtern, B. Simmons, M. Smit, and M. Litoiu, "An architecture for overlaying private clouds on public providers," in Proceedings of the 8th International Conference on Network and Service Management, CNSM '12, pp. 371–377, 2013.

[112] E. Musk, "All our patent are belong to you." https://www.teslamotors.com/blog/all-our-patent-are-belong-you. Accessed: 2016-01-21.

[113] Knowledge@Wharton, "Whats driving teslas open source gambit?." http://knowledge.wharton.upenn.edu/article/whats-driving-teslas-open-source-gambit/. Accessed: 2016-01-21.

[114] R. Kalman, "On the general theory of control systems," IRE Transactions on Automatic Control, vol. 4, no. 3, pp. 110–110, 1959.

[115] Z. Ou, H. Zhuang, J. K. Nurminen, A. Ylä-Jääski, and P. Hui, "Exploiting hardware heterogeneity within the same instance type of amazon ec2," in 4th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud), 2012.

[116] Z. Ou, H. Zhuang, A. Lukyanenko, J. Nurminen, P. Hui, V. Mazalov, and A. Yla-Jaaski, "Is the same instance type created equal? exploiting heterogeneity of public clouds," Cloud Computing, IEEE Transactions on, vol. 1, pp. 201–214, July 2013.

[117] B. Farley, A. Juels, V. Varadarajan, T. Ristenpart, K. D. Bowers, and M. M. Swift, "More for your money: Exploiting performance heterogeneity in public clouds," in 3rd ACM Symposium on Cloud Computing, pp. 20:1–20:14, 2012.

[118] Z. Xu, H. Wang, and Z. Wu, "A measurement study on co-residence threat inside the cloud," in 24th USENIX Security Symposium, (Washington, D.C.), pp. 929–944, USENIX Association, Aug. 2015.

[119] Y. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Cross-tenant side-channel attacks in paas clouds," in Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, CCS '14, pp. 990–1003, 2014.

[120] V. Varadarajan, Y. Zhang, T. Ristenpart, and M. Swift, "A placement vulnerability study in multi-tenant public clouds," in 24th USENIX Security Symposium (USENIX Security 15), (Washington, D.C.), pp. 913–928, USENIX Association, Aug. 2015.

[121] S. Kadloor, X. Gong, N. Kiyavash, T. Tezcan, and N. Borisov, "Low-cost side channel remote traffic analysis attack in packet networks," in IEEE International Conference on Communications (ICC), pp. 1–5, May 2010.

[122] S. Kadloor, N. Kiyavash, and P. Venkitasubramaniam, "Mitigating timing based information leakage in shared schedulers," in IEEE INFOCOM, pp. 1044–1052, 2012.

[123] S. Bugiel, S. Nürnberger, T. Pöppelmann, A.-R. Sadeghi, and T. Schneider, "Amazonia: When elasticity snaps back," in 18th ACM Conference on Computer and Communications Security, pp. 389–400, 2011.

[124] Y. Gu, Y. Fu, A. Prakash, Z. Lin, and H. Yin, "Os-sommelier: Memory-only operating system fingerprinting in the cloud," in 3rd ACM Symposium on Cloud Computing, pp. 5:1–5:13, 2012.

[125] Y. Gu, Y. Fu, A. Prakash, Z. Lin, and H. Yin, "Multi-aspect, robust, and memory exclusive guest os fingerprinting," Cloud Computing, IEEE Transactions on, vol. PP, no. 99, pp. 1–1, 2014.

[126] D. E. Comer and J. C. Lin, "Probing tcp implementations," in USENIX Summer 1994 Technical Conference, 1994.

[127] V. Paxson, "Automated packet trace analysis of tcp implementations," in Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, pp. 167–179, 1997.

[128] R. Beverly, "A robust classifier for passive tcp/ip fingerprinting," in Passive and Active Network Measurement, pp. 158–167, 2004.

[129] S. Chen, R. Wang, X. Wang, and K. Zhang, "Side-channel leaks in web applications: A reality today, a challenge tomorrow," in IEEE Symposium on Security and Privacy, pp. 191–206, May 2010.

[130] K. Zhang, Z. Li, R. Wang, X. Wang, and S. Chen, "Sidebuster: Automated detection and quantification of side-channel leaks in web application development," in 17th ACM Conference on Computer and Communications Security, pp. 595–606, 2010.

[131] D. Herrmann, R. Wendolsky, and H. Federrath, "Website fingerprinting: Attacking popular privacy enhancing technologies with the multinomial naive-bayes classifier," in ACM Cloud Computing Security Workshop, pp. 31–42, 2009.

[132] A. Panchenko, L. Niessen, A. Zinnen, and T. Engel, "Website fingerprinting in onion routing based anonymization networks," in 10th Annual ACM Workshop on Privacy in the Electronic Society, pp. 103–114, 2011.

[133] Z. Li, C. Jin, T. Xu, C. Wilson, Y. Liu, L. Cheng, Y. Liu, Y. Dai, and Z.-L. Zhang, "Towards network-level efficiency for cloud storage services," in Proceedings of the 2014 Conference on Internet Measurement Conference, IMC '14, pp. 115–128, 2014.

[134] European Union Agency for Network and Information Security, "Privacyand Data Protection by Design - from policy to Engineering," 2014.

[135] Presidency of the Council of the European Union, "General Data Protection Regulation - Analysis of the final compromise text with a view to agreement," Dec 2015.

[136] R. Mortier, A. Madhavapeddy, T. Hong, D. Murray, and M. Schwarzkopf, "Using dust clouds to enhance anonymous communication," in International Workshop on Security Protocols, 2010.

[137] N. Jones, M. Arye, J. Cesareo, and M. J. Freedman, "Hiding amongst the clouds: A proposal for cloud-based onion routing," in 1st USENIX Workshop on Free and Open Communications on the Internet (FOCI 11), 2011.

[138] R. Ensafi, D. Fifield, P. Winter, N. Feamster, N. Weaver, and V. Paxson, "Examining how the great firewall discovers hidden circumvention servers," in Proceedings of the 2015 ACM Conference on Internet Measurement Conference, IMC '15, pp. 445–458, 2015.

[139] C. Brubaker, A. Houmansadr, and V. Shmatikov, "Cloudtransport: Using cloud storage for censorship-resistant networking," in 14th International Symposium on Privacy Enhancing Technologies (PETS), pp. 73–82, 2014.

[140] E. Kartaltepe, J. Morales, S. Xu, and R. Sandhu, "Social network-based botnet command-and-control: Emerging threats and countermeasures," in Applied Cryptography and Network Security, pp. 511–528, 2010.

[141] N. Pantic and M. I. Husain, "Covert botnet command and control using twitter," in Proceedings of the 31st Annual Computer Security Applications Conference, ACSAC 2015, pp. 171–180, 2015.

[142] G. Kontaxis, I. Polakis, and S. Ioannidis, "Outsourcing malicious infrastructure to the cloud," in 1st SysSec Workshop, pp. 35–42, 2011.

[143] S. Zhao, P. P. C. Lee, J. C. S. Lui, X. Guan, X. Ma, and J. Tao, "Cloud-based push-styled mobile botnets: A case study of exploiting the cloud to device messaging service," in 28th Annual Computer Security Applications Conference (ACSAC), pp. 119–128, 2012.
[144] D. Fifield, C. Lan, R. Hynes, P. Wegmann, and V. Paxson, "Blocking-resistant communication through domain fronting," Proceedings on Privacy Enhancing Technologies, vol. 2015, no. 2, pp. 46–64, 2015.
[145] M. Fire, R. Goldschmidt, and Y. Elovici, "Online social networks: Threats and solutions," IEEE Communications Surveys & Tutorials, vol. 16, pp. 2019–2036, Fourthquarter 2014.
[146] L. von Ahn, M. Blum, and J. Langford, "Telling humans and computers apart automatically," Commun. ACM, vol. 47, pp. 56–60, Feb. 2004.
[147] S. Schrittwieser, P. Frühwirt, P. Kieseberg, M. Leithner, M. Mulanzzani, M. Huber, and E. Weippl, "Guess Who's Texting You? Evaluationg the Security of Smartphone Messaging Applications.," in Network and Distributed System Security (NDSS) Symposium, NDSS 12, February 2012.
[148] R. Mueller, S. Schrittwieser, P. Fruehwirt, P. Kieseberg, and E. Weippl, "What's new with whatsapp & co.? revisiting the security of smartphone messaging applications," in Proceedings of the 16th International Conference on Information Integration and Web-based Applications & Services, iiWAS '14, pp. 142–151, 2014.
[149] M. Weyrich, J. P. Schmidt, and C. Ebert, "Machine-to-machine communication," IEEE Software, vol. 31, pp. 19–23, July 2014.
[150] S. Yu, S. Guo, and I. Stojmenovic, "Fool me if you can: Mimicking attacks and anti-attacks in cyberspace," IEEE Transactions on Computers, vol. 64, pp. 139–151, Jan 2015.
[151] S. Yu, Y. Tian, S. Guo, and D. O. Wu, "Can we beat ddos attacks in clouds?," IEEE Transactions on Parallel and Distributed Systems, vol. 25, pp. 2245–2254, Sept 2014.
[152] S. Yu, G. Wang, and W. Zhou, "Modeling malicious activities in cyber space," IEEE Network, vol. 29, pp. 83–87, Nov 2015.
[153] J. Jaskolka and R. Khedri, "Exploring covert channels," in 44th Hawaii International Conference on System Sciences (HICSS), pp. 1–10, 2011.
[154] J. Jaskolka, R. Khedri, and Q. Zhang, "On the necessary conditions for covert channel existence: A state-of-the-art survey," Procedia Computer Science, vol. 10, pp. 458 – 465, 2012. {ANT} 2012 and MobiWIS 2012.
[155] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2347–2376, 2015.
[156] J. Granjal, E. Monteiro, and J. S. Silva, "Security for the internet of things: A survey of existing protocols and open research issues," IEEE Communications Surveys & Tutorials, vol. 17, no. 3, pp. 1294–1312, 2015.