2017-11-18

Revealing the unseen: how to expose cloud usage while protecting user privacy

Revealing the Unseen: How to Expose Cloud Usage While Protecting User Privacy

Ata Turk
Boston University
ataturk@bu.edu

Mayank Varia
Boston University
varia@bu.edu

Georgios Kellaris
Boston and Harvard Universities
kellaris@bu.edu

***Abstract*—Cloud users have little visibility into the performance characteristics and utilization of the physical machines underpinning the virtualized cloud resources they use. This uncertainty forces users and researchers to reverse engineer the inner workings of cloud systems in order to understand and optimize the conditions their applications operate. At Massachusetts Open Cloud (MOC), as a public cloud operator, we'd like to expose the utilization of our physical infrastructure to stop this wasteful effort. Mindful that such exposure can be used maliciously for gaining insight into other users workloads, in this position paper we argue for the need for an approach that balances openness of the cloud overall with privacy for each tenant inside of it. We believe that this approach can be instantiated via a novel combination of several security and privacy technologies. We discuss the potential benefits, implications of transparency for cloud systems and users, and technical challenges/possibilities.**

I. Introduction

Computation is rapidly migrating to the cloud due to its economies of scale and excellent network connectivity. Large companies operate thousands of VMs on the cloud [1]; government agencies, non-profit organizations, and research institutions adopt cloud based IT solutions [2]–[4], while advances in mobile and IoT solutions shift their computation towards the cloud [5].

Application development in the cloud has different dynamics compared to traditional practices. The freedom to choose the size and type of instances that will operate applications brings along the problem of identifying/selecting the best instances for running the applications. Here, 'best' does not mean the fastest or most powerful instance, but rather the one whose specific configuration is best-suited for the intended application at the lowest cost. Variability in cloud instance performances due to external interferences (e.g., noisy neighbors), especially in smaller sized cloud instances [6], [7], further complicate this selection process.

Cloud application developers (e.g. Netflix) spend an enormous amount of effort and money to develop and execute software for instance benchmarking, monitoring, and performance evaluation [8]. This software allows the prospective tenants to select and operate the right set of cloud resources that will optimize the performance-to-cost-ratio of their applications.

In order to aid cloud users in their endeavors to evaluate cloud instances, cloud providers (e.g. Amazon CloudWatch [9]) and various monitoring companies (e.g. Datadog [10], LogicMonitor [11]) monitor and expose application-level resource utilization and performance features. These systems can report application interactions with various services (e.g. number of EBS calls) and monitor performance based on user defined application performance metrics via instrumenting user applications.

Instance selection may be performed by using rough generic rules, by sampling and benchmarking on small scale test cases and using that information for instance performance prediction [12], or by all-out brute-force benchmarking on all possible instance types to identify the best fitting options [8]. These solutions utilize the metrics collected from applications in order to perform instance selection.

Unfortunately application-level performance metrics are indirect: Problems such as noisy neighbors (e.g. a neighbor causing many L2 cache misses), which are observed due to shared usage of underlying physical resources, manifest in the metrics after the problem impacts the performance of the application.

a) Problem statement: Current level of information provided by cloud vendors forces users to reverse engineer physical resource utilization metrics, which is a misguided and wasteful process. At Massachusetts Open Cloud, an academic public cloud, we would like to be better merchants and provide physical resource utilization measurements as a service (potentially even free of charge to differentiate our services) to our users.

The potential benefits of exposing physical resource utilization are many-fold. For example, by exposing the number of L2 cache misses on the physical host where a cloud instance is running, we can enable better performance prediction for cache sensitive applications and trigger problem prevention mechanisms to "kick-in." Similarly, exposing the "current connectivity" of physical hosts to services in the cloud (e.g. block storage solution) or to the outside world can be significantly useful during instance selection. Beyond its value to commercial application developers, publicly-accessible data on cloud utilization also provides immense value to researchers. First, for any experiment run on the cloud, utilization data can provide context for the results and facilitate replication and extension of the work. Second, utilization data can enable researchers to study the operation of a datacenter itself.

Unfortunately, the same data that can provide value for cloud tenants also have the potential to be exploited by malicious entities to gain insight into other users' workloads, reducing the

security posture of the cloud by enabling hazardous activities such as co-location and side-channel attacks. The problem at hand is to come up with mechanisms that enables us to enjoy the benefits of information release while mitigating the security implications as much as possible.

b) Contributions: In this paper we discuss the potential benefits and harms of releasing physical utilization data of public clouds in a publicly-accessible form. Next, we examine several technologies that address portions of the tension between openness and security. We propose that these technologies combine synergistically to cover each others' limitations and therefore offer a compelling solution that addresses the needs of cloud vendors and tenants alike. Finally, we pose a list of questions that we'd like to discuss with the PAIS community.

II. Tradeoff: Transparency vs Security

We believe that increased transparency regarding cloud utilization will reduce user costs and increase the viability of using cloud services. Both of these features provide market incentives for a cloud provider to offer utilization metrics as a service. On the other hand, utilization data can expose activities of tenants on the cloud. More specifically, aggregate utilization data may be used to pinpoint a specific tenant within the cloud. An ideal solution needs to resolve the tension between transparency and user security favorably, maximizing transparency while minimizing security risks to tenants.

Current cloud offerings do not investigate the trade-off between transparency and security. They simply offer no transparency and potentially maximum security. We dispute this decision. Since the importance of both objectives are subjective, we start our discussion by listing user classes that could benefit from cloud transparency and user classes that could be harmed by it.

A. Beneficiaries of Cloud Transparency

In this section, we distinguish between three distinct types of users who can benefit from the exposition of cloud physical resource utilization information in very different ways (see Table I).

a) Current and future tenants: Physical utilization data are of immense interest to tenants of the cloud. They know the CPU, memory, disk, and network performance requirements of their application better than anybody else. Physical utilization data can enable them to navigate the performance unpredictability of instances allocated to them.

Currently, tenants spend extensive time and money on reverse engineering efforts to 'vet' several virtualized resources and gauge their relative value toward the target application. A cloud provider who offers physical utilization as a service can free prospective tenants from this burden and permit them to focus on optimizing their own application rather than the operation of the cloud overall.

b) Scientific researchers: Today, even though many research studies are conducted using cloud resources, repeatability of experiments conducted on the cloud are debatable. This variability concern when conducting experiments on the cloud forces researchers to repeat experiments many times incurring higher cost. Still the analysis made in many scenarios can be dependent on the state of the cloud at the assessment time (e.g., an experiment made on a busy working day or a working hour may not match an experiment conducted over the weekend.)

Currently it is not possible to exactly say under what conditions the experimental findings are gathered. If utilization information associated with the physical resources used in the experiments were available to researchers, these could be reported along with the experimental parameters to shed light to external effects that impact experimental observations.

c) Cloud designers and engineers: One can argue that current "black-box" design of pubic clouds prevents cloud system innovation to happen in places other than the few big cloud vendors. Utilization data open up exciting opportunities for researchers who study the design of a cloud itself. For instance, utilization data can permit engineers to examine the impacts of different load distributions on cooling systems at scale without the expense of building a realistic datacenter and cloud to 'test' theories.

B. Bearers of Potential Security Harms

Utilization data may be used to pinpoint a specific tenant within the cloud. Hence under a more transparent cloud model, current tenants have to bear the risks of identification. This is a critical risk for two reasons. First, an attacker could passively use this information to monitor the tenant's use of the cloud (e.g., for one company to examine the popularity of a competitor). Second, an attacker can actively spawn a co-located VM to invade the tenant's privacy by, e.g., determining the tenant's software [13]–[15] or cryptographic keys [16]–[18].

Below, we detail several existing methods that an attacker might be able to use to identify the tenant's physical location within the cloud.

a) Cache usage: Perhaps the most common category of side-channel attacks on the cloud involves shared caches between tenants on the same physical machine. Attacks of this type permit an attacker to check whether a victim with predictable behavior (e.g., running a known web server) appears on a specific physical machine. More specifically, several papers have demonstrated the viability of Prime+Probe and Flush+Reload attacks on the cloud, whereby the attacker manipulates the cache with her own user-space process and then observes how her memory speeds are impacted by the target tenant's process [17]–[19].

So far, many of these methods have required active effort by the attacker even to perform identification, much less to extract data from the tenant afterward. Furthermore, several countermeasures have been proposed that permit the tenant to leverage the cache side-channel for her own defensive purposes [20], [21]; at a high level, they have the tenant monitor the

Type of user	Objective	Pertinent metric
Current or future tenant	Purchase/use instance with desirable performance	Accuracy of provided data
Researcher using the cloud	Denote conditions of cloud at time of experiment	Reproducibility
Cloud researcher	Obtain aggregate cloud system statistics	Precision of analysis
Current tenant	Prevent co-location attacks	Probability of being pinpointed

TABLE I
CATEGORIES OF CLOUD CUSTOMERS. WE DISCUSS THE BENEFITS TO THE FIRST THREE CATEGORIES OF USERS IN SECTION II-A AND THE POTENTIAL SECURITY CONCERNS TO THE FOURTH CATEGORY IN SECTION II-B.

cache herself to classify when its behavior is consistent with a Prime+Probe or Flush+Reload attack.

Unfortunately, our plan to have the cloud publish physical utilization (e.g. CPU, RAM) means that we have reduced the attacker's burden from an active role to a passive one: she receives the actual cache patterns that the side channel attacks attempt to extract! Furthermore, the defensive countermeasures are rendered meaningless as well. As a result, we must design a system that adequately protects tenants' sensitive cache information even while still providing the benefits described in Section II-A.

b) Network usage: The tenant's network utilization may also be used to identify her. An attacker can influence the network latency or bandwidth available to the target tenant and observe the impact. For instance, imagine that the tenant uses the cloud to run a web server. Then, the attacker can probe the web server with a specific network flow and then fingerprint the tenant based on which network trace demonstrates the same pattern [19], [22], [23]. If the attacker is already co-resident on the cloud, then she can verify the target's presence by flooding their shared link and observing a corresponding dip in the target's connectivity [24], [25].

Even with public network utilization data, the attacker still requires some sort of active posture to conduct any of the above attacks. But, this 'active posture' may be as simple as making selective queries to a public-facing website. As a result, in order to realize our vision, we must adequately obscure or hide the tenant's current network consumption so as to thwart the attacks described above.

c) Differences over time: The attacks described above share a common property: in principle, they can be executed simply by providing utilization data at a single point in time. However, our vision is even stronger than this: we wish to update the utilization data periodically.

Differencing attacks [26] allow attackers to infer the location of users based upon changes in utilization over time. They operate by observing small changes between two similar states, in order to disclose an individuals confidential data. For example, if a single service changes behavior, while the rest of the users continue to consume the same resources, an adversary can observe the location of the change, essentially locating the service. We note that these changes can occur within the same machine or across physical machines on the cloud.

III. A PATH FORWARD

In this section, we investigate methods to overcome the tradeoff between openness and security.

A robust technique should allow the accurate monitoring of cloud resources, while ensuring the security of the cloud users. Concerning the latter, a secure approach must hide the location of each user by protecting against (i) active attacks performed by adversaries that are also tenants, and (ii) passive attacks performed by adversaries observing the published statistics. In the first part of this section, we investigate four security technologies that individually provide some (but not all!) of the necessary protections.

Then, we propose a path forward via a novel combination of the four security-enhancing technologies. At its core, our resolution to the openness-security tradeoff involves the mismatch in *time*: attackers require information about the location of a victim's VM at the time of an attack, whereas honest beneficiaries of cloud utilization data are primarily concerned with the historical performance of a physical machine independent of the people who happen to be present on it at the moment. Ergo, published metrics can balance between openness and security by providing a variable notion of accuracy that provides accurate utilization in the past but whose fidelity *decays* as time moves toward the present.

A. Existing Tools

In this section, we describe four technologies that provide some protections for cloud tenants against identification attacks. For each technology, we provide a brief description of its operation, and then focus on its capabilities and weaknesses at addressing our specific security needs.

First, *Secure Cloud Scheduling* (SCS) offers the promise of thwarting active attacks on the cloud [27]. It places VMs in a manner that is unpredictable to the adversary and that reduces its probability of successfully completing a co-location attack. Although it offers high performance, it only ensures a weak form of security. In particular, its scheduling, while adversarially-controlled, is static. Hence, if an adversary ever manages to identify a target VM, she will know this information indefinitely.

Second, *Moving Target Defense* (MTD) [28] leverages continuous decision-making to benefit the defender. In the cloud setting, MTD allows tenants to migrate their VMs to new physical hosts within the cloud by means of a scheduler that remains securely outside of the attacker's view [29]. This movement increases the attacker's uncertainty and therefore increases her overall attack cost. Additionally, it mitigates the damage from prior revelations (overcoming SCS's defect), thereby reducing the attacker's window of opportunity.

However, MTD has two drawbacks. First, the migration of VMs is resource-intensive; if the cloud needs to perform

this often at the datacenter scale, the cloud's overall efficiency will deteriorate. Second, MTD is feckless against problems that only require a passive adversary to exploit. Recall from Section II-B that some previously-known attacks are susceptible to passive adversaries; more importantly, the public availability of utilization data transforms some active attacks into passive ones. In these cases, MTD allows invasive attacks for as long the user remains in the exposed location.

Third, *Post-Compromise Security* (PCS) [30] protects data and software in the present even if secret key material was compromised in the past. The dual to forward secrecy [31], [32], PCS is the culmination of a long body of literature into authenticated key exchange [33]–[36]. It is well-suited for combination with MTD because it can restore security guarantees against a tenant whose previous information has been compromised. For this restoration to be possible, though, the adversary must lack the information needed to compromise the target in the present. So far, all of the technologies discussed fail to provide a distinction between past and present.

Our fourth and final technology, *Differential Privacy* (DP) [37], has the potential to create a separation in time. DP protects against both active and passive attacks. It ensures that the location of any user in the data is hidden when computing statistics by perturbing them before publishing. The perturbation reduces the accuracy of the published statistics, but also obscures the presence of any user in the original data. In addition to its purpose-built use to protect individual privacy, DP has proved useful in anonymizing user location as well [38], [39].

The principal concern pertaining to the application of DP toward our setting is *accuracy*. Utilizing current DP solutions for infinite streams of cloud utilization data would greatly deteriorate the accuracy of the published statistics by protecting the user whereabouts throughout time. In the initial works that achieved DP's privacy goals in the streaming setting [40], [41], the accuracy of the output greatly deteriorates with the stream size. Fan and Xiong [42] show how to publish accurate statistics while satisfying differential privacy, i.e., obscuring the user presence in the data throughout time, assuming a finite stream. Cao et al. [43] also take into account temporal correlations among different locations. On the other hand, Kellaris et al. [44] assume an infinite scheme, but satisfies differential privacy for any time window of a predefined finite size, i.e., the user presence is solely protected within any time interval of fixed length. This final solution offers a potential path forward for us.

B. Symbiosis

We propose a combination of the four technologies described above in order to strike the appropriate balance between security, accuracy, and cost.

We begin by advocating for a datacenter-wide application of MTD. In addition to its benefits for tenants in general, its uncertainty specifically decouples location information in the past vs. present.

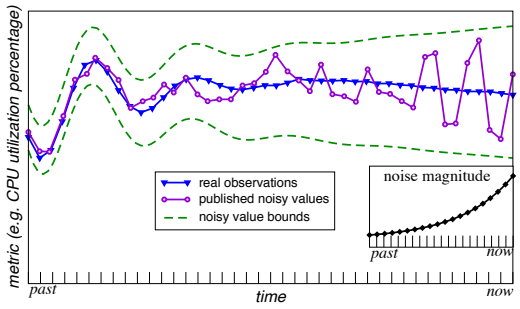

Fig. 1. A hypothetical CPU utilization graph showing both the real observations (blue) that remain private and the perturbed observations (purple) that we envision releasing publicly. Note that the noise envelope (green) that governs the perturbations varies by time, simultaneously yielding accurate past data for historical analysis and protection against location attacks in the present.

As long as the allocation of nodes within each migration step follows SCS, the attacker is also stymied when trying to execute a co-location attack based solely on side channels available in the present. Next, we advocate that cloud tenants employ PCS so as to ensure that any cryptographic material that may have been compromised in the past is now irrelevant.

Essentially, by periodically applying MTD, we may release statistics about the past to the public. On the other hand, the more recent the statistics, the higher the probability that they incorporate the current location of a user.

As such, we can employ DP in a 'decaying' manner to conceal the current whereabouts of the user, while allowing more accurate past. Figure 1 depicts the 'envelope' of uncertainty provided by DP's noise. We highlight the fact that the noise varies over time so that uncertainty rises as one approaches the present.

Combining DP and MTD introduces an inevitable but acceptable trade-off between accuracy and cost. Deploying MTD more frequently decreases the need for publishing inaccurate statistics that satisfy DP; in other words, it permits the 'envelope' of uncertainty to decay faster as one goes back through time. However, MTD is resource-intensive, and DP provides security protections for data released more recently.

Future work is needed here because none of the current DP approaches can be combined as-is with the MTD, as they do not take into account the probabilistic nature of the sensitive user location.

As such, we identify the need for a new notion of privacy, which (i) can be seamlessly combined with MTD, (ii) quantifies the offered security against co-location attacks, and (iii) returns useful monitoring statistics with bounded error. Towards this, we plan on viewing MTD as a probabilistic game (similar to [29]), and defining a new notion of privacy that offers provable guarantees similar to DP, while protecting the locations of the users between executions of the MTD.

IV. Discussion Topics

While we espouse a specific vision, our intent in a workshop format is to promote a general discussion of places in which

security technologies can enable the introduction of new (non-security-related!) features that benefit cloud tenants. We believe this is generally an under-tapped area that is ripe for further discussion, which our presentation should promote. Thus, in this section, we list some of the core open questions that are integral to the continuation of this research.

Interest on cloud transparency: Is there sufficient interest in the data we propose to make public, Would people be willing to pay for more openness in cloud services, whether the benefits can provide a market differentiator if a cloud vendor adopts our vision but its competitors do not, and whether our strategy is viable to implement if the requisite security concerns are adequately addressed.

Other attack vectors: Our vision to increase the transparency of cloud infrastructures introduces a new attack surface that can be exploited in various ways. We discuss co-location attacks and potential remedies using existing security technologies but usage of released data can enable other attack vectors as well. Whether the released data can enable other types of attacks is a matter of discussion.

Additional foreseeable overheads: The particular solution we proposed requires rather substantial changes to the behavior of cloud vendors, who must adopt MTD at the datacenter scale, and tenants, who must modify applications they host on the cloud to absorb the side effects of proactive features of the system. Whether there exists other foreseeable overheads is a matter of discussion.

ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under Grants Numbered 1414119, 1347525, 1565387 and 1149232, the MassTech Collaborative Research Matching Grant Program, and the commercial partners of the Massachusetts Open Cloud, which include Brocade, Cisco, Intel, Lenovo, Red Hat and Two Sigma.

REFERENCES

[1] K. Weins, "Cloud computing trends: 2017 state of the cloud survey," http://www.rightscale.com/blog/cloud-industry-insights/cloud-computing-trends-2017-state-cloud-survey, 2017.

[2] D. Schaffhauser, "Higher ed cloud adoption on the rise," https://campustechnology.com/articles/2016/09/22/higher-ed-cloud-adoption-on-the-rise.aspx, 2016.

[3] Logicworks, "Government cloud on the rise: NSA and DOJ move to amazon web services," https://www.cloudcomputing-news.net/news/2015/jul/01/government-cloud-on-the-rise-nsa-doj-move-to-amazon-web-services/, 2015.

[4] Amazon-Web-Services, "Us federal government in the cloud," https://aws.amazon.com/federal/, 2017.

[5] A. Botta, W. de Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and internet of things," *Future Gener. Comput. Syst.*, vol. 56, no. C, pp. 684–700, Mar. 2016.

[6] C. Delimitrou and C. Kozyrakis, "Hcloud: Resource-efficient provisioning in shared cloud systems," in *ACM SIGOPS Operating Systems Review*, vol. 50, no. 2. ACM, 2016, pp. 473–488.

[7] X. Gong, N. Kiyavash, and N. Borisov, "Quantifying the noisy neighbor problem in Openstack," April 2016, presented at *OpenStack Summit*, available at https://www.openstack.org/assets/presentation-media/ZeroStack-Austin-Presentation.pdf.

[8] B. Gregg, "Performance tuning amazon ec2 instances, aws re:invent 2014," https://youtu.be/7Cyd22kOqWc, 2014.

[9] A. W. Services, "Amazon cloudwatch," https://aws.amazon.com/cloudwatch/, 2017.

[10] Datadog, "Cloud-scale monitoring," https://www.datadoghq.com/, 2017.

[11] LogicMonitor, "Saas-based performance monitoring," https://www.logicmonitor.com/, 2017.

[12] S. Venkataraman, Z. Yang, M. Franklin, B. Recht, and I. Stoica, "Ernest: efficient performance prediction for large-scale advanced analytics," in *NSDI*, 2016.

[13] G. Irazoqui, M. S. Inci, T. Eisenbarth, and B. Sunar, "Know thy neighbor: Crypto library detection in cloud," *PoPETs*, vol. 2015, no. 1, pp. 25–40, 2015.

[14] K. Suzaki, K. Iijima, T. Yagi, and C. Artho, "Memory deduplication as a threat to the guest OS," in *EUROSEC*, 2011, p. 1.

[15] Y. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Cross-tenant side-channel attacks in paas clouds," in *CCS*, 2014, pp. 990–1003.

[16] D. A. Osvik, A. Shamir, and E. Tromer, "Cache attacks and countermeasures: The case of AES," in *CT-RSA*, 2006, pp. 1–20.

[17] M. S. Inci, B. Gülmezoglu, G. I. Apecechea, T. Eisenbarth, and B. Sunar, "Seriously, get off my cloud! cross-vm RSA key recovery in a public cloud," *IACR Cryptology ePrint Archive*, vol. 2015, p. 898, 2015.

[18] N. Benger, J. van de Pol, N. P. Smart, and Y. Yarom, ""ooh aah... just a little bit" : A small amount of side channel can go a long way," in *CHES*, 2014, pp. 75–92.

[19] 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 *CCS*, 2009, pp. 199–212.

[20] M. S. Inci, B. Gülmezoglu, T. Eisenbarth, and B. Sunar, "Co-location detection on the cloud," in *COSADE*, 2016, pp. 19–34.

[21] Y. Zhang, A. Juels, A. Oprea, and M. K. Reiter, "Homealone: Co-residency detection in the cloud via side-channel analysis," in *S&P*, 2011.

[22] X. Gong, N. Kiyavash, and N. Borisov, "Fingerprinting websites using remote traffic analysis," in *CCS*, 2010, pp. 684–686.

[23] 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.

[24] K. Block and G. Noubir, "Return of the covert channel, data center style," in *CCSW*, 2015, pp. 17–28.

[25] Y. Agarwal, V. Murale, J. Hennessey, K. Hogan, and M. Varia, "Moving in next door: Network flooding as a side channel in cloud environments," in *CANS*, 2016, pp. 755–760.

[26] J. Lucero, L. Zayatz, L. Singh, J. You, M. DePersio, and M. Freiman, "The current stage of the microdata analysis system at the us census bureau," in *ISI*, 2011.

[27] Y. Azar, S. Kamara, I. Menache, M. Raykova, and B. Shepard, "Co-location-resistant clouds," in *CCSW*, 2014, pp. 9–20.

[28] S. Jajodia, A. K. Ghosh, V. Swarup, C. Wang, and X. S. Wang, *Moving target defense: creating asymmetric uncertainty for cyber threats*. Springer Science & Business Media, 2011, vol. 54.

[29] M. H. Valizadeh, H. Maleki, W. Koch, A. Bestavros, and M. van Dijk, "Markov modeling of moving target defense games," *IACR Cryptology ePrint Archive*, vol. 2016, p. 741, 2016.

[30] K. Cohn-Gordon, C. J. F. Cremers, and L. Garratt, "On post-compromise security," in *CSF*, 2016, pp. 164–178.

[31] W. Diffie, P. C. van Oorschot, and M. J. Wiener, "Authentication and authenticated key exchanges," *Des. Codes Cryptography*, vol. 2, no. 2, pp. 107–125, 1992.

[32] N. Unger, S. Dechand, J. Bonneau, S. Fahl, H. Perl, I. Goldberg, and M. Smith, "Sok: Secure messaging," in *S&P*, 2015.

[33] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *EUROCRYPT*, 2001, pp. 453–474.

[34] B. A. LaMacchia, K. E. Lauter, and A. Mityagin, "Stronger security of authenticated key exchange," in *ProvSec*, 2007, pp. 1–16.

[35] D. Stebila and B. Ustaoglu, "Towards denial-of-service-resilient key agreement protocols," in *ACISP*, 2009, pp. 389–406.

[36] H. Krawczyk, "HMQV: A high-performance secure diffie-hellman protocol," in *CRYPTO*, 2005, pp. 546–566.

[37] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *TCC*, 2006.

[38] J. van den Hooff, D. Lazar, M. Zaharia, and N. Zeldovich, "Vuvuzela: scalable private messaging resistant to traffic analysis," in *SOSP*, 2015, pp. 137–152.

[39] D. Lazar and N. Zeldovich, "Alpenhorn: Bootstrapping secure communication without leaking metadata," in *OSDI*, 2016, pp. 571–586. [Online]. Available: https://www.usenix.org/conference/osdi16/technical-sessions/presentation/lazar
[40] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum, "Differential privacy under continual observation," in *STOC*, 2010.
[41] T. Chan, E. Shi, and D. Song, "Private and continual release of statistics," *TISSEC*, vol. 14, no. 3, p. 26, 2011.
[42] L. Fan and L. Xiong, "Real-time aggregate monitoring with differential privacy," in *CIKM*, 2012.
[43] Y. Cao, M. Yoshikawa, Y. Xiao, and L. Xiong, "Quantifying differential privacy under temporal correlations," in *ICDE*, 2017.
[44] G. Kellaris, S. Papadopoulos, X. Xiao, and D. Papadias, "Differentially private event sequences over infinite streams," in *VLDB*, 2014.