RESEARCH PAPERS

The following are a broad sample of the work of the Institute.

Authenticating Privately over Public Wi-Fi Hotspots

By Northeastern University's Guevara Noubir with Aldo Cassola and Erik-Oliver Blass

Abstract

Wi-Fi connectivity using open hotspots hosted on untrusted Access Points (APs) has been a staple of mobile network deployments for many years as mobile providers seek to offload smartphone traffic to Wi-Fi. Currently, the available hotspot solutions allow for mobility patterns and client identities to be monitored by the parties hosting the APs as well as by the underlying service provider. We propose a protocol and system that allows a service provider to authenticate its clients and hides the client identity from both AP and service provider at the time of authentication. Particularly, the client is guaranteed that either the provider cannot do better than to guess their identity randomly or they obtain proof that the provider is trying to reveal their identity by using different keys. Our protocol is based on Private Information Retrieval (PIR) with an augmented cheating detection mechanism based on our extensions to the NTRU encryption scheme. The somewhat-homomorphic encryption makes auditing of multiple rows in a single query possible and optimizes PIR for highly parallel GPU computations with the use of the Fast Fourier Transform (FFT).

In this work we lay out the operation of our protocol in detail, its security analysis, and propose an implementation compatible with the Wi-Fi Extensible Authentication Protocol (EAP) along with optimizations for deployments of over 10 million clients. We evaluate the performance of its mobile and provider components and show that a client can be authenticated in 43.9 milliseconds on a GPU platform, giving an end-to-end authentication of 1.12 seconds.

View online

Cascading Denial of Service Attacks on Wi-Fi Networks

By Northeastern University's Guevara Noubir with Liangxiao Xin and David Starobinski

Abstract

We unveil the existence of a vulnerability in Wi-Fi, which allows an adversary to remotely launch a Denial-of-Service (DoS) attack that propagates both in time and space. This vulnerability stems from a coupling effect induced by hidden nodes. Cascading DoS attacks can congest an entire network and do not require the adversary to violate any protocol. We demonstrate the feasibility of such attacks through experiments with real Wi-Fi cards, extensive ns-3 simulations, and theoretical analysis. The simulations show that the attack is effective both in networks operating under fixed and varying bit rates, as well as ad hoc and infrastructure modes. To gain insight into the root causes of the attack, we model the network as a dynamical system and analyze its limiting behavior. The model predicts that a phase transition (and hence a cascading attack) is possible when the retry limit parameter of Wi-Fi is greater or equal to 7 and characterizes the phase transition region in terms of the system parameters.

View online

CRLite: A Scalable System for Pushing All TLS Revocations to All Browsers

By Northeastern University's James Larisch with David Choffnes, Christo Wilson, Alan Mislove, Dave Levin and Bruce M. Maggs

Abstract

Currently, no major browser fully checks for TLS/SSL certificate revocations. This is largely due to the fact that the deployed mechanisms for disseminating revocations (CRLs, OCSP, OCSP Stapling, CRLSet, and OneCRL) are each either incomplete, insecure, inefficient, slow to update, not private, or some combination thereof. In this paper, we present CRLite, an efficient and easily-deployable system for proactively pushing all TLS certificate revocations to browsers. CRLite servers aggregate revocation information for all known, valid TLS certificates on the web, and store them in a space-efficient filter cascade data structure. Browsers periodically download and use this data to check for revocations of observed certificates in realtime. CRLite does not require any additional trust beyond the existing PKI, and it allows clients to adopt a fail-closed security posture even in the face of network errors or attacks that make revocation information temporarily unavailable.

We present a prototype of CRLite that processes TLS certificates gathered by Rapid7, the University of Michigan, and Google's Certificate Transparency on the server-side, with a Firefox extension on the client-side. Comparing CRLite to an idealized browser that performs correct CRL/OCSP checking, we show that CRLite reduces latency and eliminates privacy concerns. Moreover, CRLite has low bandwidth costs: it can represent all certificates with an initial download of 10MB (less than 1 byte per revocation) followed by daily updates of 580KB on average. Taken together, our results demonstrate that complete TLS/SSL revocation checking is within reach for all clients.

View online

Understanding the Role of Registrars in DNSSEC Deployment

By Northeastern University's Taejoong Chung with Christo Wilson, David Choffnes, Alan Mislove, Roland van Rijswijk-Deij, Dave Levin, and Bruce M. Maggs

Abstract

The Domain Name System (DNS) provides a scalable, flexible name resolution service. Unfortunately, its unauthenticated architecture has become the basis for many security attacks. To address this, DNS Security Extensions (DNSSEC) were introduced in 1997. DNSSEC's deployment requires support from the top-level domain (TLD) registries and registrars, as well as participation by the organization that serves as the DNS operator. Unfortunately, DNSSEC has seen poor deployment thus far: despite being proposed nearly two decades ago, only 1% of .com, .net, and .org domains are properly signed. In this paper, we investigate the underlying reasons why DNSSEC adoption has been remarkably slow. We focus on registrars, as most TLD registries already support DNSSEC and registrars often serve as DNS operators for their customers.

Our study uses large-scale, longitudinal DNS measurements to study DNSSEC adoption, coupled with experiences collected by trying to deploy DNSSEC on domains we purchased from leading domain name registrars and resellers. Overall, we find that a select few registrars are responsible for the (small) DNSSEC deployment today, and that many leading registrars do

not support DNSSEC at all or require customers to take cumbersome steps to deploy DNSSEC. Further frustrating deployment, many of the mechanisms for conveying DNSSEC information to registrars are error-prone or present security vulnerabilities. Finally, we find that using DNSSEC with third-party DNS operators such as Cloudflare requires the domain owner to take a number of steps that 40% of domain owners do not complete. Having identified several operational challenges for full DNSSEC deployment, we make recommendations to improve adoption.

View online

A Longitudinal, End-to-End View of the DNSSEC Ecosystem

By Northeastern University's Taejoong Chung, Christo Wilson, David Choffnes, Alan Mislove with Balakrishnan Chandrasekaran, Roland van Rijswijk-Deij, Dave Levin and Bruce M. Maggs

Abstract

The Domain Name System's Security Extensions (DNSSEC) allow clients and resolvers to verify that DNS responses have not been forged or modified inflight. DNSSEC uses a public key infrastructure (PKI) to achieve this integrity, without which users can be subject to a wide range of attacks. However, DNSSEC can operate only if each of the principals in its PKI properly performs its management tasks: authoritative name servers must generate and publish their keys and signatures correctly, child zones that support DNSSEC must be correctly signed with their parent's keys, and resolvers must actually validate the chain of signatures. This paper performs the first large-scale, longitudinal measurement study into how well DNSSEC's PKI is managed.

We use data from all DNSSEC-enabled subdomains under the .com, .org, and .net TLDs over a period of 21 months to analyze DNSSEC deployment and management by domains; we supplement this with active measurements of more than 59K DNS resolvers worldwide to evaluate resolver-side validation. Our investigation reveals pervasive mismanagement of the DNSSEC infrastructure. For example, we found that 31% of domains that support DNSSEC fail to publish all relevant records required for validation; 39% of the domains use insufficiently strong key-signing keys; and although 82% of resolvers in our study request DNSSEC records, only 12% of them actually attempt to validate them. These results highlight systemic problems, which motivate improved automation and auditing of DNSSEC management.

View online

Lens on the endpoint: Hunting for malicious software through endpoint data analysis

By Northeastern University's Ahmet Salih Buyukkayhan, Alina Oprea, and William Robertson

Abstract

Organizations are facing an increasing number of criminal threats ranging from opportunistic malware to more advanced targeted attacks. While various security technologies are available to protect organizations' perimeters, still many breaches lead to undesired consequences such as loss of proprietary information, financial burden, and reputation defacing. Recently, endpoint monitoring agents that inspect system-level activities on user machines started to gain traction and be deployed in the industry as an additional defense layer. Their application, though, in most cases is only for forensic investigation to determine the root cause of an incident.

In this paper, we demonstrate how endpoint monitoring can be proactively used for detecting and prioritizing suspicious software modules overlooked by other defenses. Compared to other environments in which host-based detection proved successful, our setting of a large enterprise introduces unique challenges, including the heterogeneous environment (users installing software of their choice), limited ground truth (small number of malicious software available for training), and coarse-grained data collection (strict requirements are imposed on agents' performance overhead). Through applications of clustering and outlier detection algorithms, we develop techniques to identify modules with known malicious behavior, as well as modules impersonating popular benign applications. We leverage a large number of static, behavioral and contextual features in our algorithms, and new feature weighting methods that are resilient against missing attributes. The large majority of our findings are confirmed as malicious by anti-virus tools and manual investigation by experienced security analysts.

View online

Analysis of SSL Certificate Reissues and Revocations in the Wake of Heartbleed

By Northeastern's Liang Zhang, David Choffnes, Alan Mislove, Christo Wilson with Dave Levin, Tudor Dumitras and Aaron Schulman

Abstract

Central to the secure operation of a public key infrastructure (PKI) is the ability to revoke certificates. While much of users' security rests on this process taking place quickly, in practice, revocation typically requires a human to decide to reissue a new certificate and revoke the old one. Thus, having a proper understanding of how often systems administrators reissue and revoke certificates is crucial to understanding the integrity of a PKI. Unfortunately, this is typically difficult to measure: while it is relatively easy to determine when a certificate is revoked, it is difficult to determine whether and when an administrator should have revoked. In this paper, we use a recent widespread security vulnerability as a natural experiment. Publicly announced in April 2014, the Heartbleed OpenSSL bug, potentially (and undetectably) revealed servers' private keys.

Administrators of servers that were susceptible to Heartbleed should have revoked their certificates and reissued new ones, ideally as soon as the vulnerability was publicly announced. Using a set of all certificates advertised by the Alexa Top 1 Million domains over a period of six months, we explore the patterns of reissuing and revoking certificates in the wake of Heartbleed. We find that over 73% of vulnerable certificates had yet to be reissued and over 87% had yet to be revoked three weeks after Heartbleed was disclosed. Moreover, our results show a drastic decline in revocations on the weekends, even immediately following the Heartbleed announcement. These results are an important step in understanding the manual processes on which users rely for secure, authenticated communication.

View online

An End-to-End Measurement of Certificate Revocation in the Web's PKI

By Northeastern's Yabing Liu, Will Tome, Liang Zhang, David Choffnes, Alan Mislove, Christo Wilson with Dave Levin, Bruce Maggs and Aaron Schulman

Abstract

Critical to the security of any public key infrastructure (PKI) is the ability to revoke previously issued certificates. While the overall SSL ecosystem is well-studied, the frequency with which certificates are revoked and the circumstances under which clients (e.g., browsers) check whether certificates are revoked are still not well-understood.

In this paper, we take a close look at certificate revocations in the Web's PKI. Using 74 full IPv4 HTTPS scans, we find that a surprisingly large fraction (8%) of the certificates served have been revoked, and that obtaining certificate revocation information can often be expensive in terms of latency and bandwidth for clients. We then study the revocation checking behavior of 30 different combinations of web browsers and operating systems; we find that browsers often do not bother to check whether certificates are revoked (including mobile browsers, which uniformly never check). We also examine the CRLSet infrastructure built into Google Chrome for disseminating revocations; we find that CRLSet only covers 0.35% of all revocations. Overall, our results paint a bleak picture of the ability to effectively revoke certificates today.

View online

Understanding the Role of Registrars in DNSSEC Deployment

By Northeastern's Taejoong Chung, David Choffnes, Alan Mislove, Christo Wilson with Roland van Rijswijk-Deij, Dave Levin and Bruce M. Maggs

Abstract

The Domain Name System (DNS) provides a scalable, flexible name resolution service. Unfortunately, its unauthenticated architecture has become the basis for many security attacks. To address this, DNS Security Extensions (DNSSEC) were introduced in 1997. DNSSEC's deployment requires support from the top-level domain (TLD) registries and registrars, as well as participation by the organization that serves as the DNS operator. Unfortunately, DNSSEC has seen poor deployment thus far: despite being proposed nearly two decades ago, only 1% of .com, .net, and .org domains are properly signed.

In this paper, we investigate the underlying reasons why DNSSEC adoption has been remarkably slow. We focus on registrars, as most TLD registries already support DNSSEC and registrars often serve as DNS operators for their customers. Our study uses large-scale, longitudinal DNS measurements to study DNSSEC adoption, coupled with experiences collected by trying to deploy DNSSEC on domains we purchased from leading domain name registrars and resellers. Overall, we find that a select few registrars are responsible for the (small) DNSSEC deployment today, and that many leading registrars do not support DNSSEC at all or require customers to take cumbersome steps to deploy DNSSEC. Further frustrating deployment, many of the mechanisms for conveying DNSSEC information to registrars are error-prone or present security vulnerabilities. Finally, we find that using DNSSEC with third-party DNS operators such as Cloudflare requires the domain owner to take a number of steps that 40% of domain owners do not complete. Having identified several operational challenges for full DNSSEC deployment, we make recommendations to improve adoption.

View online

Analysis of SSL Certificate Reissues and Revocations in the Wake of Heartbleed

By Northeastern's Liang Zhang, David Choffnes, Alan Mislove, Christo Wilson with Tudor Dumitras and Aaron Schulman

Abstract

Central to the secure operation of a public key infrastructure (PKI) is the ability to revoke certificates. While much of users' security rests on this process taking place quickly, in practice, revocation typically requires a human to decide to reissue a new certificate and revoke the old one. Thus, having a proper understanding of how often systems administrators reissue and revoke certificates is crucial to understanding the integrity of a PKI. Unfortunately, this is typically difficult to measure: while it is relatively easy to determine when a certificate is revoked, it is difficult to determine whether and when an administrator should have revoked.

In this paper, we use a recent widespread security vulnerability as a natural experiment. Publicly announced in April 2014, the Heartbleed OpenSSL bug, potentially (and undetectably) revealed servers' private keys. Administrators of servers that were susceptible to Heartbleed should have revoked their certificates and reissued new ones, ideally as soon as the vulnerability was publicly announced.

Using a set of all certificates advertised by the Alexa Top 1 Million domains over a period of six months, we explore the patterns of reissuing and revoking certificates in the wake of Heartbleed. We find that over 73% of vulnerable certificates had yet to be reissued and over 87% had yet to be revoked three weeks after Heartbleed was disclosed. Moreover, our results show a drastic decline in revocations on the weekends, even immediately following the Heartbleed announcement. These results are an important step in understanding the manual processes on which users rely for secure, authenticated communication.

View online

MOSAIC: A Platform for Monitoring and Security Analytics in Public Clouds

By Northeastern's Alina Oprea, Cristina Nita-Rotaru with Ata Turk and Orran Krieger

View online

Measurement and Analysis of Private Key Sharing in the HTTPS Ecosystem

By Northeastern's Taejoong Chung, David Choffnes, Alan Mislove, Christo Wilson with Frank Cangialosi, Dave Levin and Bruce M. Maggs

Abstract

The semantics of online authentication in the web are rather straightforward: if Alice has a certificate binding Bob's name to a public key, and if a remote entity can prove knowledge of Bob's private key, then (barring key compromise) that remote entity must be Bob. However, in reality, many websites and the majority of the most popular ones are hosted at least in part by third parties such as Content Delivery Networks (CDNs) or web hosting providers. Put simply: administrators of websites who deal with (extremely) sensitive user data are giving their private keys to third parties. Importantly, this sharing of keys is undetectable by most users, and widely unknown even among researchers.

In this paper, we perform a large-scale measurement study of key sharing in today's web. We analyze the prevalence with which websites trust third-party hosting providers with their secret keys, as well as the impact that this trust has on responsible key management practices, such as revocation. Our results reveal that key sharing is extremely common, with a small handful of hosting providers having keys from the majority of the most popular websites. We also find that hosting providers often manage their customers' keys, and that they tend to react more slowly yet more thoroughly to compromised or potentially compromised keys.

View online

ReCon: Revealing and Controlling PII Leaks in Mobile Network Traffic

By Northeastern's Jingjing Ren, David Choffnes with Ashwin Rao, Martin Lindorfer and Arnaud Legout

Abstract

It is well known that apps running on mobile devices extensively track and leak users' personally identifiable information (PII); however, these users have little visibility into PII leaked through the network traffic generated by their devices, and have poor control over how, when and where that traffic is sent and handled by third parties. In this paper, we present the design, implementation, and evaluation of ReCon: a cross-platform system that reveals PII leaks and gives users control over them without requiring any special privileges or custom OSes. ReCon leverages machine learning to reveal potential PII leaks by inspecting network traffic and provides a visualization tool to empower users with the ability to control these leaks via blocking or substitution of PII. We evaluate ReCon's effectiveness with measurements from controlled experiments using leaks from the 100 most popular iOS, Android, and Windows Phone apps, and via an IRB-approved user study with 92 participants. We show that ReCon is accurate, efficient, and identifies a wider range of PII than previous approaches.

View online

SCORAM: Oblivious RAM for Secure Computation

By Northeastern's Abhi Shelat with Xiao Shaun Wang, Yan Huang, T-H. Hubert Chan and Elaine Shi

Abstract

Oblivious RAMs (ORAMs) have traditionally been measured by their bandwidth overhead and client storage. We observe that when using ORAMs to build secure computation protocols for RAM programs, the size of the ORAM circuits is more relevant to the performance.

We therefore embark on a study of the circuit-complexity of several recently proposed ORAM constructions. Our careful implementation and experiments show that asymptotic analysis is not indicative of the true performance of ORAM in secure computation protocols with practical data sizes.

We then present SCORAM, a heuristic compact ORAM design optimized for secure computation protocols. Our new design is almost 10x smaller in circuit size and also faster than all other designs we have tested for realistic settings (i.e., memory sizes between 4MB and 2GB, constrained by 2-80 failure probability). SCORAM makes it feasible to perform secure computations on gigabyte-sized data sets.

View online

UNVEIL: A Large-Scale, Automated Approach to Detecting Ransomware

By Northeastern's Amin Kharraz with Sajjad Arshad, Collin Mulliner, William Robertson and Engin Kirda

Abstract

Although the concept of ransomware is not new (i.e. such attacks date back at least as far as the 1980s), this type of malware has recently experienced a resurgence in popularity. In fact, in the last few years, a number of high-profile ransomware attacks were reported, such as the large-scale attack against Sony that prompted the company to delay the release of the film "The Interview." Ransomware typically operates by locking the desktop of the victim to render the system inaccessible to the user, or by encrypting, overwriting, or deleting the user's files. However, while many generic malware detection systems have been proposed, none of these systems have attempted to specifically address the ransomware detection problem.

In this paper, we present a novel dynamic analysis system called UNVEIL that is specifically designed to detect ransomware. The key insight of the analysis is that in order to mount a successful attack, ransomware must tamper with a user's files or desktop. UNVEIL automatically generates an artificial user environment and detects when ransomware interacts with user data. In parallel, the approach tracks changes to the system's desktop that indicate ransomware-like behavior. Our evaluation shows that UNVEIL significantly improves the state of the art and is able to identify previously unknown evasive ransomware that was not detected by the antimalware industry.

View online

Scaling ORAM for Secure Computation

By Northeastern's Jack Doerner with Abhi Shelat

Abstract

We design and implement a Distributed Oblivious Random Access Memory (DORAM) data structure that is optimized for use in twoparty secure computation protocols. We improve upon the access time of previous constructions by a factor of up to ten, their memory overhead by a factor of one hundred or more, and their initialization time by a factor of thousands. We are able to instantiate ORAMs that hold 234 bytes, and perform operations on them in seconds, which was not previously feasible with any implemented scheme.

Unlike prior ORAM constructions based on hierarchical hashing [21], permutation [21], or trees [40], our Distributed ORAM is derived from the new Function Secret Sharing scheme introduced by Boyle, Gilboa and Ishai [11, 12]. This significantly reduces the amount of secure computation required to implement an ORAM access, albeit at the cost of O(n) efficient local memory operations.

We implement our construction and find that, despite its poor O(n) asymptotic complexity, it still outperforms the fastest previously known constructions, Circuit ORAM [43] and Square-root ORAM [56], for datasets that are 32 KiB or larger, and outperforms prior work on applications such as stable matching [16] or binary search [25] by factors of two to ten.

View online

Shreds: Fine-grained Execution Units with Private Memory

By Northeastern's Yaohui Chen, Yaohui Chen, Long Lu with Sebassujeen Reymondjohnson

Abstract

Once attackers have injected code into a victim program's address space, or found a memory disclosure vulnerability, all sensitive data and code inside that address space are subject to thefts or manipulation. Unfortunately, this broad type of attack is hard to prevent, even if software developers wish to cooperate, mostly because the conventional memory protection only works at process level and previously proposed in-process memory isolation methods are not practical for wide adoption.

We propose shreds, a set of OS-backed programming primitives that addresses developers' currently unmet needs for finegrained, convenient, and efficient protection of sensitive memory content against in-process adversaries. A shred can be viewed as a flexibly defined segment of a thread execution (hence the name). Each shred is associated with a protected memory pool, which is accessible only to code running in the shred. Unlike previous works, shreds offer in-process private memory without relying on separate page tables, nested paging, or even modified hardware. Plus, shreds provide the essential data flow and control flow guarantees for running sensitive code. We have built the compiler toolchain and the OS module that together enable shreds on Linux. We demonstrated the usage of shreds and evaluated their performance using 5 non-trivial open source software, including OpenSSH and Lighttpd. The results show that shreds are fairly easy to use and incur low runtime overhead (4.67%).

View online

LAVA: Large-scale Automated Vulnerability Addition

By Northeastern's Engin Kirda, Andrea Mambretti, Wil Robertson with Brendan Dolan-Gavitt, Patrick Hulin, Tim Leek, Frederick Ulrich and Ryan Whelan

Abstract

Work on automating vulnerability discovery has long been hampered by a shortage of ground-truth corpora with which to evaluate tools and techniques. This lack of ground truth prevents authors and users of tools alike from being able to measure such fundamental quantities as miss and false alarm rates. In this paper, we present LAVA, a novel dynamic taint analysis-based technique for producing ground-truth corpora by quickly and automatically injecting large numbers of realistic bugs into program source code. Every LAVA bug is accompanied by an input that triggers it whereas normal inputs are extremely unlikely to do so. These vulnerabilities are synthetic but, we argue, still realistic, in the sense that they are embedded deep within programs and are triggered by real inputs. Using LAVA, we have injected thousands of bugs into eight real-world programs, including bash, tshark, and the GNU coreutils. In a preliminary evaluation, we found that a prominent fuzzer and a symbolic execution-based bug finder were able to locate some but not all LAVA-injected bugs, and that interesting patterns and pathologies were already apparent in their performance. Our work forms the basis of an approach for generating large ground truth vulnerability corpora on demand, enabling rigorous tool evaluation and providing a high-quality target for tool developers.

View online

Interactive Fingerprinting Codes and the Hardness of Preventing False Discovery

By Northeastern's Jonathan Ullman with Thomas Steinke

Abstract

We show an essentially tight bound on the number of adaptively chosen statistical queries that a computationally efficient algorithm can answer accurately given n samples from an unknown distribution. A statistical query asks for the expectation of a predicate over the underlying distribution, and an answer to a statistical query is accurate if it is "close" to the correct expectation over the distribution. This question was recently studied by Dwork et al. [DFH+15], who showed how to answer ˜(n2) queries efficiently, and also by Hardt and Ullman [HU14], who showed that answering ˜O(n3) queries is hard. We close the gap between the two bounds and show that, under a standard hardness assumption, there is no computationally efficient algorithm that, given n samples from an unknown distribution, can give valid answers to O(n2) adaptively chosen statistical queries. An implication of our results is that computationally efficient algorithms for answering arbitrary, adaptively chosen statistical queries may as well be differentially private.

We obtain our results using a new connection between the problem of answering adaptively chosen statistical queries and a combinatorial object called an interactive fingerprinting code [FT01]. In order to optimize our hardness result, we give a new Fourier-analytic approach to analyzing fingerprinting codes that is simpler, more flexible, and yields better parameters than previous constructions.

View online

Counter-Jamming Using Mixed Mechanical and Software Interference Cancellation

By Northeastern's Triet D. Vo-Huu with Erik-Oliver Blass and Guevara Noubir

Abstract

Wireless networks are an integral part of today's cyberphysical infrastructure. Their resiliency to jamming is critical not only for military applications, but also for civilian and commercial applications. In this paper, we design, prototype, and evaluate a system for cancelling jammers that are significantly more powerful than the transmitting node. Our system combines a novel mechanical beam-forming design with a fast auto-configuration algorithm and a software radio digital interference cancellation algorithm. Our mechanical beam-forming uses a custom-designed two-elements architecture and an iterative algorithm for jammer signal identification and cancellation. We have built a fully functional prototype (using 3D printers, servos, USRP-SDR) and demonstrate a robust communication in the presence of jammers operating at five orders of magnitude stronger power than the transmitting node. Similar performance in traditional phased arrays and radar systems requires tens to hundreds of elements, high cost and size.

View online

Interleaving Jamming in Wi-Fi Networks

By Northeastern's Triet D. Vo-Huu with Tien D. Vo-Huu and Guevara Noubir

Abstract

The increasing importance of Wi-Fi in today's wireless communication systems, both as a result of Wi-Fi offloading and its integration in IoT devices, makes it an ideal target for malicious attacks. In this paper, we investigate the structure of the combined interleaver/convolutional coding scheme of IEEE 802.11a/g/n. The analysis of the first and second-round permutations of the interleaver, allows us to design deterministic jamming patterns across subcarriers that when de-interleaved results in an interference burst. We show that a short burst across carefully selected sub-carriers exceeds the error correction capability of Wi-Fi. We implemented this attack as a reactive interleaving jammer on the firmware of the low-cost HackRF SDR. Our experimental evaluation shows that this attack can completely block the Wi-Fi transmissions with jamming power less than 1% of the communication (measured at the receiver) and block 95% of the packets with less than 0.1% energy. Furthermore, it is at least 5 dB and up to 15 dB more power efficient than jamming attacks that are unaware of the Wi-Fi interleaving structure.

View online

Spooky Encryption and its Applications

By Northeastern's Daniel Wichs with Yevgeniy Dodis, Shai Halevi and Ron D. Rothblum

Abstract

Consider a setting where inputs x1,...,xn are encrypted under independent public keys. Given the ciphertexts {ci = Encpki (xi)}i, Alice outputs ciphertexts c'1,…,c'n that decrypt to y1,…,yn respectively. What relationships between the xi's and yi's can Alice induce? Motivated by applications to delegating computations, Dwork, Langberg, Naor, Nissim and Reingold [DLN+04] showed that a semantically secure scheme disallows signaling in this setting, meaning that yi cannot depend on xj for j ≠ i. On the other hand if the scheme is homomorphic then any local (component-wise) relationship is achievable, meaning that each yi can be an arbitrary function of xi. However, there are also relationships which are neither signaling nor local. Dwork et al. asked if it is possible to have encryption schemes that support such \spooky" relationships. Answering this question is the focus of our work.

Our first result shows that, under the LWE assumption, there exist encryption schemes supporting a large class of "spooky" relationships, which we call additive function sharing (AFS) spooky. In particular, for any polynomial-time function f, Alice can ensure that y1,…,yn are random subject to ∑n i=1 yi = f(x1,…,xn). For this result, the public keys all depend on common public randomness. Our second result shows that, assuming sub-exponentially hard indistinguishability obfuscation (iO) (and additional more standard assumptions), we can remove the common randomness and choose the public keys completely independently. Furthermore, in the case of n = 2 inputs, we get a scheme that supports an even larger class of spooky relationships.

We discuss several implications of AFS-spooky encryption. Firstly, it gives a strong counterexample to a method proposed by Aiello et al. [ABOR00] for building arguments for NP from homomorphic encryption. Secondly, it gives a simple 2-round multi-party computation pro-

tocol where, at the end of the first round, the parties can locally compute an additive secret sharing of the output. Lastly, it immediately yields a function secret sharing (FSS) scheme for all functions.

We also define a notion of spooky-free encryption, which ensures that no spooky relationship is achievable. We show that any non-malleable encryption scheme is spooky-free. Furthermore, we can construct spooky-free homomorphic encryption schemes from SNARKs, and it remains an open problem whether it is possible to do so from falsifiable assumptions.

View online

Outsourcing Private RAM Computation

By Northeastern's Daniel Wichs with Craig Gentry, Shai Halevi and Mariana Raykova

Abstract

We construct the first schemes that allow a client to privately outsource arbitrary program executions to a remote server while ensuring that: (I) the client's work is small and essentially independent of the complexity of the computation being outsourced, and (II) the server's work is only proportional to the run-time of the computation on a random access machine (RAM), rather than its potentially much larger circuit size. Furthermore, our solutions are non-interactive and have the structure of reusable garbled RAM programs, addressing an open question of Lu and Ostrovsky (Eurocrypt 2013). We also construct schemes for an augmented variant of the above scenario, where the client can initially outsource a large private and persistent database to the server, and later outsource arbitrary program executions with read/write access to this database.

Our solutions are built from non-reusable garbled RAM in conjunction with new types of reusable garbled circuits that are more efficient than prior solutions but only satisfy weaker security. For the basic setting without a persistent database, we can instantiate the required type of reusable garbled circuits from indistinguishability obfuscation or from functional encryption for circuits as a black-box. For the more complex setting with a persistent database, we can instantiate the required type of reusable garbled circuits using stronger notions of obfuscation. It remains an open problem to instantiate these new types of reusable garbled circuits under weaker assumptions, possibly avoiding obfuscation altogether.

We also give several extensions of our results and techniques to achieve: schemes with efficiency proportional to the input-specific RAM run-time, verifiable outsourced RAM computation, functional encryption for RAMs, and a candidate obfuscator for RAMs.

View online

Simple Lattice Trapdoor Sampling from a Broad Class of Distributions

By Northeastern's Daniel Wichs with Vadim Lyubashevsky

Abstract

At the center of many lattice-based constructions is an algorithm that samples a short vector s, satisfying [A|AR - HG]s = t mod q where A, AR, H, G are public matrices and R is a trapdoor. Although the algorithm crucially relies on the knowledge of the trapdoor R to perform this sampling efficiently, the distribution it outputs should be independent of R given the public values. We present a new, simple algorithm for performing this task. The main novelty of our sampler is that the distribution of s does not need to be Gaussian, whereas all previous works crucially used the properties of the Gaussian distribution to produce such an s. The advantage of using a non-Gaussian distribution is that we are able to avoid the high-precision arithmetic that is inherent in Gaussian sampling over arbitrary lattices. So while the norm of our output vector s is on the order of √n to n - times larger (the representation length, though, is only a constant factor larger) than in the samplers of Gentry, Peikert, Vaikuntanathan (STOC 2008) and Micciancio, Peikert (EUROCRYPT 2012), the sampling itself can be done very efficiently. This provides a useful time/output trade-o_ for devices with constrained computing power. In addition, we believe that the conceptual simplicity and generality of our algorithm may lead to it finding other applications.

View online