

NSF Report on 2nd CSR PI Meeting

October 24, 2018
Bellevue, WA



Sponsored by

National Science Foundation

NSF CSR PI Meeting Committee

Workshop Chair

- Weisong Shi, Wayne State University

Breakout Session Leaders

- Security by Systems Design
 - Dilma Da Silva, Texas A&M University
 - Dongyan Xu, Purdue University
- IoT Systems Challenges
 - Tarek Abdelzaher, University of Illinois Urbana-Champaign
 - Gurdip Singh, Syracuse University
- Women in Computing
 - Leana Golubchik, University of Southern California
 - Jennifer Weston, AAAS Science and Technology Fellow

Executive Summary

The Computing Systems Research (CSR) program of the NSF funds a wide range of systems research including (but not limited to) systems for Internet-of-Things (which makes our homes and city smarter), systems to make more affordable and energy-efficient our Supercomputers (which furthers medical research, our understanding of the universe, and warns the population of tornadoes and hurricanes, to name a few), systems to improve the usefulness, security, and reliability of our smartphones, and systems to secure and scale the cloud (which has expanded the capabilities of IT systems while dramatically decreasing the associated costs). The workshop organizes and funds a community event for the PIs of awards from the CSR program, as well as selected aspiring PIs for the CSR program. At this event, the attendees have the opportunity to meet at-large and share and discuss their research, identify systems research challenges and directions. The objective of the workshop is to provide input to NSF and Computer Systems Research Community on three key questions: (1). What is the vision of computer systems research in the next 5-10 years? (2). What are the grand challenges to achieve the vision? and (3) How to broaden the participation women in computing?

This workshop brought together 122 active PIs and aspiring PIs in the computer systems research community on October 24, 2018, at Hyatt Regency Bellevue. This report serves as a collection of such input to government funding agencies and interested parties in industry and academia.

We identified the top grand challenges and discussed potential solutions in each of the three areas, represented through the parallel breakout sessions in this one-day workshop: Security by Systems Design, Systems Challenges for Internet of Things, and Women in Computing.

Examples of grand challenge from each of these three lists include:

- Supporting both clean-slate efforts and efforts hardening existing systems: As a research community, we should continue to pursue clean-slate, ground-breaking system concepts, architectures and technologies to advance the field and define future computer system functions, capabilities and security.
- A working definition of IoT has come out as IoT systems are distributed systems with physical (sensors, robots, actuators), algorithmic/cyber, and human components, interacting to offer a service at scale. These systems are made distinct by the following three features: Open, Data Centric, and Large-scale. In the context of the definition, the group identified five challenges: (1) Data curation; (2) Machine Learning; (3) Composability and Openness/Assurance Trade-offs; (4) multi-* issues, and (5) uncertainty and failures.
- The systems (and the broader CS) community needs a culture shift where we, as researchers, spend more time on communicating about our field and the corresponding important problems. We must also clearly and openly communicate the type of culture we wish to build.

I: Overview

The Computing Systems Research (CSR) program of the NSF funds a wide range of systems research including (but not limited to) systems for Internet-of-Things (which makes our homes and city smarter), systems to make more affordable and energy-efficient our Supercomputers (which furthers medical research, our understanding of the universe, and warns the population of tornadoes and hurricanes, to name a few), systems to improve the usefulness, security, and reliability of our smartphones, and systems to secure and scale the cloud (which has expanded the capabilities of IT systems while dramatically decreasing the associated costs).

The landscape of computer systems research has changed significantly fueled by the burgeoning of Internet of Everything (IoE), e.g., wearables, lights, and vehicles, and expected 5G deployment this year. As computing technology has gradually immersed into our daily life, we have witnessed two radical changes in the past decade: rapidly growing cloud computing and pervasive mobile devices, sensors and Internet of Things. Cloud Computing, an alternative to the traditional model of owning and managing private resources by customers, provided centralized computing services and pay-as-you-use convenience to the clients. While there are many emerging issues to be solved, Cloud Computing has reaped its field from enterprises to personal end users. Meanwhile, mobile devices, such as smartphones and tablets, have become pervasive and are driving the development of many new applications, powered by the ever-improving wireless networking and mobility support. According to Cisco's conservative estimate, there will be 50 billion connected devices by 2020, forming an Internet of Things. Things across all industry domains, from transportation to healthcare to manufacturing to smart cities to smart grids, are being connected to address a growing range of needs for businesses and consumers. Enabling these future Internet of Things imposes unique challenges. For example, many devices will have limited battery power and processing capabilities, and hence cannot support computational-intensive tasks.

To undertake the increasing challenges faced by the CSR community, in this project, PI Weisong Shi has requested fund to organize the NSF CSR program 2018 PI meeting. The project organizes and funds a community event for all the PIs of awards from the CSR program and selected aspiring PIs. At this event, the researchers funded by the CSR program and aspiring PIs will have the opportunity to meet at-large and share and discuss their research, identify systems research challenges and directions.

The objective of the workshop is to provide input to NSF and Computer Systems Research Community on three key questions:

- (1) What is the vision of computer systems research in the next 5-10 years?
- (2) What are the grand challenges to achieve the vision?
- (3) How to broaden the participation women in computing?

This report serves as a collection of such input to government funding agencies and interested parties in industry and academia. We identified the top five grand challenges in each of the three areas, represented through the parallel breakout sessions in this one-day workshop: Security by Systems Design, Systems Challenges for Internet of Things, and Women in Computing.

II: Security by Systems Design

Discussions in this session started with participants identifying a computer system's layers and components with security challenges, which span the system's entire hardware and software stack. The following are layers/components with corresponding security challenges identified by the participants:

- **Hardware and architecture:** Exploitable side channels and vulnerabilities have been found in IC, FPGA, microarchitecture, and ISA, as exemplified by the recent discovery of Meltdown and Spectre vulnerabilities at the microarchitecture level. Such hardware "bugs" may be exploited by malicious programs to leak benign program execution state and memory content, posing fundamental threats to any computer that adopts the vulnerable microarchitecture.
- **Hypervisor, operating system (OS), and trusted execution environment (TEE):** These system software programs are known to contain bugs and vulnerabilities that are susceptible to malware (such as kernel rootkits) attacks. Such attacks undermine the trustworthiness of the entire software stack, for example, by violating the mutual isolation property between entities (e.g., processes and virtual machines) running at a higher layer. As a result, malicious processes or virtual machines may stealthily break out of their confined space without being detected.
- **Network, storage, and datacenter:** Going beyond individual computers, adversaries have been targeting integrated compute infrastructures such as a datacenter. For example, different types of side channels arising from various shared resources have been discovered in real-world cloud datacenters. Attackers may exploit such side channels to infer cloud tenants' private data and runtime information and inflict disruptions to their operations (e.g., DDoS).
- **Application programs and data:** There exist a wide range of vulnerabilities – at both design and implementation levels – in application programs. Till today, accurate, complete discovery of such vulnerabilities and automatic construction of corresponding exploits remains an open, difficult problem. This poses particular challenges in vetting, debugging, and patching commodity/commercial programs. Meanwhile, application data privacy has become a significant concern, as demonstrated by recent massive data leak incidents that affected millions.

The participants also discussed security requirements and challenges in emerging or specialized systems such as cyber-physical systems, real-time systems, and IoT systems. For example,

cyber-physical systems are subject to not just cyber threats but also cyber-physical side-channel attacks (such as sensor spoofing); IoT systems are subject to physical damage, tampering, and spoofing as they are not as well protected as computers in traditional environments such as datacenters.

Following the “security by systems design” theme, the discussion then switched to identifying key system design aspects that impact a subject system’s security.

- Definition of threat model and trust model: Such models are essential to system design as they will define (and assume) the subject system’s trusted computing base (TCB). The common goal is to minimize the TCB, which makes the security mechanisms and policies on top of it more robust. Moreover, the threat model should define adversaries’ targets, goals, strategies, and capabilities. The trust model for IoT or cyber-physical systems should also define the trusted portion of system hardware and (physical) operating environments.
- Layers of abstraction: Layering has been a widely adopted methodology for system design to control complexity and decouple mechanisms and policies. It is important to define the security objectives and properties (e.g., DDoS, spoofing and/or tamper-resistance, confidentiality, etc.) of each layer and, more importantly, the mapping – both downward and upward – of security properties from one layer to the next. The interfaces between layers need to be well-defined as they are usually part of the attack surface exposed to adversaries. The infrastructure of deployment may also introduce additional vulnerabilities.
- Policies and mechanisms: For each system component or layer, it is necessary to clearly define the mechanisms it provides and the policies it enforces. Mechanisms at one layer will become programmable building blocks of higher layers to implement a wide range of security policies (e.g., access control of data, memory, and services).
- Specification, composition, and compartmentalization: It is a common practice to develop composite or federated systems (e.g., a web service) using simpler components as building blocks. Such composition needs to be specified at design time to clearly define the interface and “border-crossing” policies between system components or subsystems. This will lead to the compartmentalization of system components, which will help contain and localize the impact of an attack within one or very few “compartments”, avoiding cascading, global damage to the overall system. Compartmentalization is highly desirable in defending against advanced persistent threats (APTs), which typically involve cross-host/component reconnaissance and lateral movements. Specification practices that appropriately handle composition from the security perspective are needed. The diversity and complexity of elements in federated systems may benefit from a ‘system of systems’ approach.
- Tradeoff between performance and security: In computer systems design, it is not uncommon to observe that performance and security requirements are at odds. In other words, the optimization of one aspect may negatively affect the other. The conflict and possible tradeoff between performance and security need to be explicitly

considered during system design and implementation, under the constraint of system operation context and resource availability.

The participants also suggested that new tools be developed to help define and specify some of the aspects above, leading to a security-aware integrated development environment (IDE). The discussion then continued, with the focus on methods and methodologies that improve security by system design.

- Iterative design, implementation, and evaluation: It may not be possible to pre-identify all security requirements and vulnerabilities during the initial design phase of system development. Moreover, implementation of the system may introduce additional vulnerabilities. As a result, it is necessary to consider an iterative system development life-cycle that naturally supports system security/vulnerability assessment at all stages, which may trigger another round of design, implementation, and evaluation of system components that have been shown vulnerable.
- Complex system modeling: In addition to the traditional layering methodology for modeling a complex system (e.g., a datacenter or an autonomous vehicle), it is equally desirable to model the interaction horizontally – both explicit and implicit (e.g., via side channels) – between components operating at the same layer. System-level side channels deserve more research attention as they are often introduced for performance optimization and resource sharing/consolidation purposes, reflecting the odds between performance and security.
- Formal methods capturing both functional and non-functional semantics: Reliability and safety are widely recognized as critical properties of mission-critical systems. Meanwhile, formal methods for system function specification and verification have started to show effectiveness in real-world, large-scale systems such as OS kernels (e.g., seL4 microkernel) and UAVs. It is important to continue to support the development of advanced, practical formal methods to capture and reason about both functional and non-functional system semantics (e.g., semantics that reflects the tradeoff between efficiency and security, safety and privacy, and cost and effectiveness).
- “Hole-punching” between layers: Hole-punching has proven to be an effective method to improve cross-layer system efficiency and adaptability, and avoid redundant (or even conflicting) operations. It is equally meaningful to make low-level system information available to higher-level entities for improving reliability and security. For example, it is helpful to expose ISA semantics (including timing information) to OS developers so that the OS can possess a level of model-checking capability against the underlying ISA. As such, ISA-level vulnerabilities may be more proactively exposed and mitigated during testing and production runs. Other security-oriented “hole-punching” may be helpful the between hypervisor and (guest) OS, between OS and application, and between network and end-systems. Also, for performance-oriented “hole-punching,” it is important to demonstrate that security constraints are not violated.
- Provenance tracing and attack/incident reproduction: These methodologies would allow users or administrators to effectively investigate and learn from past attacks, failures, and successes (in detecting, preventing, or recovering from attacks). Attack investigation

and reproduction have been proven helpful in identifying and fixing design- and implementation-level flaws and vulnerabilities of computer systems. Another significant benefit of provenance tracing is to protect data privacy and confidentiality, detect unauthorized information disclosure, and identify cascading impacts of corrupted data on other data products during system operations.

- Performance and security co-design: Specialization and generalization are two possible directions in probing for the optimal balance between performance and security. Such probing needs to take into consideration the cost, usage scenarios, performance bounds, and security/privacy levels of a subject system. More quantitatively, it requires the definition of metrics and evaluation benchmarks for both security and performance, so that we can identify tradeoff opportunities and specific sweet spots. It was also suggested that side-channel modeling and categorization become part of the performance-security co-design so that exploitable side channels can be proactively discovered, confirmed, and mitigated during the early stage of system design and implementation. More generally, the modeling of possible attacks against the subject system should become a “first-class” step instead of an afterthought, so that we can build the corresponding defense into the system to minimize future patching, which may incur disruption and new, unexpected side effects.
- Context-driven approach: For real-time systems, traditional ways of handling security violations may be inappropriate in some application domains. Instead, the system’s security aspects may need to be integrated with existing support for graceful degradation. Techniques that introduce flexibility in adapting violation behavior may be necessary.
- Data-driven behavior analysis: existing techniques for enforcing system security may be enhanced by data analysis techniques offering early detection of anomalies.
- Untrusted base: novel designs for layering trusted components on top of untrusted ones may be necessary to address the practical reality of code legacy, particularly for cyber-physical applications that must work with a broad spectrum of device capabilities, protocols, and reliability.
- Security quantification: benchmarks and metrics for security or security levels may be useful when analyzing security vs. performance tradeoffs.

Finally, the participants recommended a couple of funding strategies to support systems research that promotes a strong built-in security.

- Engaging industry and convincing industry to adopt research results: We have seen companies (e.g., Intel and VMware in recent years) supporting systems research funding programs with strong security focus. They set up concrete platforms and channels to assess and potentially adopt research results from funded projects. Such heavy industry involvement encourages researchers to understand industry trends and real-world security challenges, making their research approaches and results more relevant and practical. The participants recommended more such industry-government programs.

- Supporting both clean-slate efforts and efforts hardening existing systems: As a research community, we should continue to pursue clean-slate, ground-breaking system concepts, architectures and technologies to advance the field and define future computer system functions, capabilities and security. This is exemplified by recent programs that have motivated disruptive research efforts across the community, such as software-defined networking, infrastructures, and services, with fruitful research results that address the security and privacy of such emerging, clean-slate systems. Many results have been incorporated into real-world products and infrastructures.

Meanwhile, there exist a wide range of existing systems – many are mission critical – that cannot be retired overnight and replaced by clean-slate counterparts. They need to be retrofitted with the latest security solutions and techniques with minimum modification and disruption. Some participants argued that securing legacy systems via system hardening and retrofitting is not just an engineering effort. Instead, there are significant research challenges in making legacy system support more systematic and assured, as exemplified by recent research efforts of the security community in enabling post-production commodity software transformation, with the goal of “de-bloating” and hardening the original software for higher resource efficiency and security (due to the reduced attack surface and stronger protection). The participants recommended that research efforts of similar motivation and broader systems scope be encouraged by upcoming programs.

Top 5 Challenges:

- Derive secure-by-design systems that integrate IoT devices optimized for cost or efficiency.
- Usable , extensible techniques and tools for verifying system security properties.
- Addressing the inevitability of side-channels, for example by model and categorization.
- Provenance tracing for both privacy and incident investigation.
- Systems that learn from security incidents to protect from an associated vulnerability.

III: Systems Challenges for IoT

The discussion in this session started with participants sharing their research interests as they pertain to systems challenges in IoT. The goal was to elicit the key intellectual ingredients of IoT-related challenges that catalyze community research and interest in this field. The following is a list of keywords and topics that highlight part of that discussion:

- **Data:** A key component of IoT research, reflected in participant interests, centered around data challenges such as data fusion (e.g., fusion of visual information with time-series sensor signals), efficient/on-demand video steaming/processing, and data analytics for IoT. This interest stemmed from the view that a key function of IoT systems should be to collect, analyze, and exploit data from myriads of connected “things” of different modalities, making data challenges central to IoT research.
- **Sensing:** Related to the data challenges was the topic of design of sensors and other data acquisition systems including wearables, camera systems, and IoT devices.
- **Mobile and wireless computing:** Mobility was deemed a key aspect of IoT systems, leading to a broad set of challenges pertaining to the design, development, and operation of wireless networks, wireless and mobile computing, mobile sensing, data fusion from social and urban/sensor data, and mobile multimedia, among others.
- **Machine Learning:** Given the multitudes of data collected, a key topic discussed was the need for machine learning solutions to derive value for applications. They included deep learning algorithms and explainable AI/ML models that offer intelligent IoT functions.
- **Cloud/edge computing:** Participants also referred to challenges in developing the underlying infrastructure for distributed and stream processing systems, machine intelligence, as well as service models for edge computing, storage systems, and efficient query processing.
- **System support:** Other system support called for included OS design for IoT systems, IoT-centric compilers and programming languages, compiler optimization, and performance modeling for IoT applications.
- **Security/Privacy:** Given the increased connectivity afforded by IoT systems (implying an increased attack surface and increased potential for vulnerability) and given the increased exposure of physical components to cyber threats, the topic protecting data, systems, applications, and users emerged as a key research area. **Energy:** To increase sustainability of IoT systems, and meet resource constraints, low power computing, power-efficient ML, and transient computing were mentioned as core challenges.
- **Assured systems:** An important category of IoT applications was envisioned to be mission-critical systems, where assurances of correctness become paramount. Research areas that offer such assurances in IoT environments were encouraged including formal methods, assured CPS architectures, and debugging tools.
- **Performance evaluation:** Understanding systems’ performance in extreme environments and assessing robustness/resilience was discussed as an important topic. **Hardware architecture:** Finally, multiprocessor design, CPU/GPU co-exploitation, multicore architecture, and hardware platforms for machine learning were discussed.

The session participants spent some time discussing a possible definition of IoT. The focus was on distinguishing IoT systems from other types of systems such as CPS, and traditional distributed systems. The working definition that the group arrived at is: *IoT systems are distributed systems with physical (sensors, robots, actuators), algorithmic/cyber, and human components, interacting to offer a service at scale.* These systems are made distinct by the following features:

- Open
- Data centric
- Large scale

Following a discussion during the first half of the session, the participants separated into four groups to refine, prioritize, and elaborate the challenges. The following are the challenges identified by each group:

- 1 **Data Curation Challenges:** The discussion was framed around typical IoT applications such as a collection of video cameras (and perhaps other sensing devices) monitoring an area of interest such as a bridge (that may be prone to flooding) or an elderly care facility. In the context of such applications, the fundamental question is where does “truth” reside. It was noted that in IoT systems, truth resides at the edge - closest to the physical world. The cloud, on the other hand, holds the curated (perhaps aggregated) truth. Given the large-scale and heterogeneous, fundamental questions about the “truth curation pipeline” become critical in IoT systems. For example, users in IoT applications may have different requirements on truth and need different semantic interpretations of data. A driver, for instance, may want to know whether it is safe to cross a flooded bridge whereas traffic personnel may be interested in knowing details of the location/number of vehicles on (or entering) the bridge. Furthermore, data truth may be time-varying with freshness and latency implications. Finally, data itself may be noisy and intermittent. Hence, a challenge is to develop architectures for the data curation pipeline that collect and fuse data (of different quality) from diverse sources and deliver information to consumers with varied requirements. In particular, the architectures must use “data utility” as a key concept. Research will be needed to define consistency of information with respect to how the same information is viewed by different consumers and timeliness. The architecture must account for multiple tiers (Cloud, Cloudlets, Mobile Devices, Sensors and other harvested data sources) in the pipeline. Efficiency consideration will require development of techniques such as those that filter data as much as possible at the source so as to avoid overloading shared resources down the pipeline, and techniques to exploit physical models to help de-noise and filter data. An interesting question brought up was the challenge of data de-biasing. Both humans and machine learning solutions can impose biases in data collection, labeling, and/or interpretation, resulting in inaccurate views of the world, and in biased decisions and actuation. Algorithms for understanding and neutralizing these biases become important.
- 2 **Machine Learning Issues:** Machine learning will be a critical technology in the data curation pipeline. There are several aspects that make ML different in the context of IoT. The first is the *context challenge*: Most of the machine learning models cannot be used as is because of

contextual information. They need for context-sensitive retraining, which opens up interesting questions regarding efficiency, scalability, and feasibility. Approaches that streamline or obviate retraining in different contexts (e.g., transfer learning) are needed for IoT applications. The second is the *allocation challenge*: where do you run the model? Cloud, edge, device, or combination? Consensus is to run parts of ML on all of these, which is a challenging intelligence allocation problem. IoT devices are “smart” and heterogeneous, offering a rich solution space for allocation problems, and increasing their complexity. Techniques to evaluate tradeoffs between the possible allocations will be needed. The third (training challenge) concerns how to train? Should it be done centrally or in a distributed manner (for privacy)? Finally, the actuation challenge deals the issue that decisions may have to be made on machine learning outputs, possibly without full understanding of the reason for the ML output. In such scenarios, the challenge is how to make that safe, and what post-mortem diagnostics would be needed?

- 3 ***Composability and Openness/Assurance Trade-offs***: As applications may arrive dynamically, techniques will be needed to dynamically assemble available resources and components to deploy applications. Composability will be a challenge in deploying such applications. Frameworks will be needed that allow decomposition and mapping of application-level goals to lower-level goals that can be satisfied by the available components. Furthermore, due to the dynamic and uncertain nature of IoT systems, this decomposition will need to adapt at runtime to ensure application goals. Hence, adaptation must be a first-order principle in designing IoT systems. Adaptation algorithms are needed that ensure graceful degradation and combine responsiveness with intelligent behavior. The requirements for responsiveness and intelligence are at odds in that responsiveness is best achieved by pre-wired (instinct-like) responses, whereas intelligence calls for a more in-depth run-time assessment of the situation and context. Techniques are needed to seamlessly combine the best of these solution extremes. Techniques to infer end-to-end metrics from performance metrics of the individual components will have to be developed. There is also a tension between openness in IoT systems and assurances - openness will make systems susceptible to un-anticipated threats, and providing assurances in such situations may be challenging. Frameworks will be needed that can evaluate and assess assurance (online, run-time analysis) of dynamically composed IoT systems. Assurances should be provided in the face of potential lack of complete control on parts of the system, and should account for multiple failure dimensions.
- 4 ***Multi-* Issues***: The large-scale, heterogeneous nature of IoT systems coupled with the different types of usages envisioned pose several multi-* challenges. The IoT infrastructure, for instance, may be shared among multiple applications. Traditional mission-critical systems are closed in nature; as a result, the environment is more predictable, and thus amenable to static analysis and assurance guarantees. When infrastructure may be shared between a number of applications (which may be unknown in advance), it becomes more difficult to guarantee assurances (in particular, for mission-critical applications). This problem is made more difficult by the fact that there might be multiple domains of control as resources (sensors, computing, network) owned by different entities/organizations may have to be pooled dynamically to configure an application. This creates a tension between manageability (homogeneity) and diversity. Hence, techniques to develop applications that

can provide guarantees in such a multi-mission, multi-domain IoT system is a fundamental challenge. These techniques must exploit trade-offs between (lack of complete) control and (need for) assurances.

- 5 ***Uncertainty and Failures:*** Due to large-scale nature of IoT systems as well as their operational environments (e.g., operations after a natural disaster, during blackout/brownout or high demand during pop-up events), failures and uncertainty in resource availability will pose configuration and deployment challenges. For instance, the components may have multiple modes of failures (non-binary failure). There may also be other failure dimensions - data-related failures, consistency failures. Thus, comprehensive frameworks that accommodate failures and enable graceful degradation (with respect to performance) have to be developed. Here, the challenge is that applications must operate under uncertainty but with performance assurances. Techniques to mask the failures and to communicate degraded modes of operation to the users will have to be studied. These may require applications to be self-aware and self-configuring.

While IoT is a heavily used term in industry, the discussion stressed fundamental inadequacies in the way that above challenges might be addressed if left to industrial resources alone. Specifically, industrial incentives are more aligned with vertically integrated solutions to application problems. These incentives hurt fundamental knowledge, interoperability, and broader applicability. Academia can provide open source solutions at different layers of the IoT system infrastructure that enhance interoperability, openness, and component reuse, ultimately reducing overall cost of IoT application design and development.

IV: Women in Computing

Introduction

The ideas, problems, and solutions suggested in this report were all provided by members of the research community at the Computer Systems Research PI meeting. The goal of this session was to outline some of the underlying reasons that women are underrepresented in the computer systems research community, and to draft workable action items to help improve this situation. It will take everyone involved in systems to change the culture, so we aimed for input of the full systems community to make sure that a wide spread of ideas and experiences were represented. At the session, we first broke down into small groups, discussing the problems surrounding women in computer science and computer systems research that participants had witnessed or experienced. After each group had time to discuss these issues, we came together to summarize and develop possible action items to address them -- as individuals, as institutions, and as a community.

As a systems community, our goal is for actions and solutions to be scalable, so that we can take one person's or one institution's efforts and build on them. Sharing best practices and metrics to measure and evaluate our process will help in that effort. Additionally, our solutions should provide a better space and pipeline not only for women, but for everyone -- particularly other underrepresented or underserved groups. This is the only way to change the culture of computer science and systems and to build institutional pressure to improve. If we start with the systems community, we can build out from there -- this should not be limited to the academic community alone. Note that these solutions must be undertaken by the entire community: this is *not* a one-gender solution -- leaving it to women and underrepresented minorities to 'fix' problems of diversity only increases the burden on these groups.

Culture and Communication in Computer Science

Systems Specific Issues

Systems research in particular does not always stand out in computer science. Several participants noted that systems researchers find it challenging to stand out and to communicate the applications of our research. The nature of systems is that we create artifacts that are application independent -- often extracting the parts from applications that can be generalized and used with any type of data. This is something we typically pride ourselves on. In some ways, the systems community prides itself on being "intimidating", on being "hardcore deep wizardry" -- an attitude that can be off putting to many. Intimidating introductions to

systems with little initial support, context, or idea of what skills are valuable can also be harmful to the field. The current challenges on which systems researchers are working are often not presented, leaving newcomers with little idea of what problems they might be able to work on and what applications and impact their work could have. This can be particularly off-putting to under-represented groups.

While the field of systems research has a lot of creativity, there are often no mechanisms to express this and to communicate with others. *We need a culture shift where we, as researchers, spend more time on communication.* We could, for example, bring in communication experts from outside of systems and computer science. The works of E.G. Tufte on presentations are a valuable resource for any scientist, as is the Alan Alda Center for Communicating Science.

Universities and institutions like NSF could provide more professional development funding for communication resources -- be it teaching resources or workshops on improv methods. This would enable faculty, PIs, students, and junior researchers to express and communicate their work in a more compelling way. This in turn would help spark interest in our work to a wider audience.

One additional challenge in computer systems research is that the field has a smaller number of students and faculty compared to some other sub-topics of computer science. This often means that the power structure is highly dependent on the advisor, who sets the tone for the group. **It is important for a PI to think about how to establish an appropriate culture in the work group,** since they won't always be present.

General Computer Science

K-12

The problems present in systems research are a subset of those present in computer science in general. The perception of computer science being 'not for girls' starts early, and is reinforced both at home and at school. For girls, existing K-12 system is not providing a good pipeline -- middle school in particular was called out as "A Wasteland of Opportunities" -- which is particularly dismaying as by high school the field is already male dominated.

At home, parents may be more likely to buy computers for their sons and provide opportunities than for their daughters; similarly, more video games are marketed to boys than to girls, and at younger ages. Robotics toys often look 'boyish' and are thus not given to girls as frequently. This leads to girls getting less hands on experience and therefore less excitement about computing very early on. The media portrayal of a coder or computer scientist is male and 'geeky' -- and women who *do* code on TV or in movies are often marked as 'counterculture' or otherwise different. The language and culture of technology begins early, as does the experience of being intimidated by those who have (or are perceived to have) more expertise.

In school, examples of women computer scientists are often not provided, and teachers may reinforce biases about who is “supposed to” be good at computers. There is also the general problem of poor teaching quality of computer science at these lower levels: teachers themselves often lack exposure and experience. This lack can even include not fully understanding what a computer scientist does. One participant talked about about how they witnessed a panel of high school students discussing different professions. The students could identify the current grand challenges in Physics, Biology, and other professions, but for computing they could not: they saw it as just a tool, not as an area of research. *The common approach in teaching -- learning how to code without any focus on fundamentals, history or context of the field -- does not reflect the reality of what computer science is.* To quote another participant: “Chemistry isn’t taught as how to wash beakers”. **Our aim should be to get students excited by computer science and to see it as fun and promising career.**

This process is further complicated by the fact that school curricula have so much structure that it is often difficult to make changes. It can often be easier to get principals interested than teachers, so *the focus needs to be on teacher engagement and education* as well as on *providing tangible and practical materials and training* that are accessible to the teachers themselves. These materials should have a focus on exploration of what can be done with computers rather than a purely coding approach. Professional training could be designed so that it results in curricula that teachers can take back to the classroom. Curricula should provide exciting, prepackaged, scalable curriculum and activities for the appropriate age groups. Curricula should give an introduction to what computer scientists do, and identify projects that have human impact and broader societal impact -- showing the interesting and attractive side of computer science. **Our aim should be to facilitate a pipeline of qualified teachers with an exciting curriculum.**

To this end, **NSF might host workshops for high school teachers and PhD students on how to teach computer science.** Encouraging PhD students to participate in programs such as “Girls Who Code”, “Systers”, “Skype-a-Scientist”, or even “Dance your PhD” could go a long way to changing the perception of computer science as well as educating and mentoring the next generation of young female computer scientists. Other program ideas include the creation of virtual internships for 8th grade (or even younger), where students could shadow a computer scientist for a day or more, to give a more complete sense of what computer science is, besides computer programming. Connecting classrooms with things like Raspberry PI or Seymour is also a useful mechanism that could be employed to great benefit, albeit expensive to create and hard to travel with. For any such program, it would be important to make sure of the presence of women computer scientists or at least provide examples of women role models, to prevent reinforcing ideas of ‘gender-appropriateness’ for the career. Additionally, there should be a strong focus on creating educational materials that are not overtly gendered or “boy-focused”.

College/Graduate/Postgraduate

Our discussion surrounding higher education for young computer scientists focused on a different set of issues than K-12 education. Here, *the main topics of discussion were the isolation many women experience in the field, lack of student support, the culture in the academic system, and the pipeline for computer scientists as a whole*. We focused on what steps could be taken on the personal, institutional, and community levels. **We need a mentality of diversity with systematic ways to bring about changes.**

Several PIs mentioned *a lack of community and contact for female computer science students*. This isolation can stem both, from lack of female peers or mentors and from a lack of general social acceptance by their peer group. There are both subtle and explicit barriers. Being “the only one in the room” (or one of few) can be intimidating and off putting, whether at conferences or in a classroom. *College and graduate school come with many of their own stresses, and many of these stresses are particularly difficult for women in computer science*. Introductory courses are often designed with the attitude of “How do we weed out our students?” rather than “How do we get our students to do better?” The system does not necessarily support a variety of learning styles. There is an emphasis on pattern based learning (learn by competing, asserting and having an argument) rather than empathetic type learning (trying to understand through other person’s eyes, bias story towards other person).

One participant pointed out that, particularly during undergraduate education, they saw that an A- for women was often more upsetting than a C+ was for men, with *women taking failure more seriously*. This has different manifestations, such as imposter syndrome and lack of self-confidence. Another participant commented about their student that “she doesn’t realize how awesome she is” and posed the question of whether women are more honest about their abilities, or men are simply more overconfident.

While PIs may wish to provide support for their students, they may not even be aware of what options for funding or professional development are available. Often travel grants go to well-known schools only or to students of PIs who’s advisors have money already, as those are the PIs who are more aware of these opportunities for their students. Teaching colleges, undergraduate institutions, may simply lack resources or be unaware of what options are available. And while programs such as NSF’s ADVANCE exist, it’s not always clear what programs are available to women or other targeted underrepresented groups. Fellowships carry a lot more weight for a student than simply being funded by a PI. *We need to reach students and junior faculty at smaller colleges, even those who never thought of attending these conferences or may not even have heard of these conferences*. We need to think about how to reach students who don’t have access to faculty at their institutions. This could be through NSF incentives and Fellowships for women and Underrepresented Minorities to encourage students to pursue a PhD. ACM chapters could work with local colleges to make sure that they are aware of all opportunities and travel scholarships. **The REU model has been very successful, as have a**

number of bridge programs, so it could be worthwhile to have more development of those sorts of programs for students who are *not* at major research universities.

What can PIs, NSF, and the community do to support these students? *We want to build a small friendly environment in computer systems research that can grow into a big friendly environment in computer science as a whole.* We need to support women locally at universities as well as providing networking opportunities for them: more workshops, and places where we can expand opportunities for women in computer science. Additionally, simply building awareness of organizations, opportunities, and networks that already exist (N2 Women, Grace Hopper, etc.) will make sure that the resources that *are* available are being utilized. Setting up specific data services, office hours, pre-college programs, student mentoring programs, or hackathons aimed at women could also help build a sense of community and support. Similarly, there was the suggestion to set up faculty mirroring programs so that students can shadow a computer scientist and see what is involved in a professorship as well as see the value of other skills such as project management and balancing work and life.

There are a number of small actions that a PI can take that can help make a difference on a local level. Make sure to be explicit, encouraging, and open about office hours, to support students who may be more tentative about approaching a professor or PI. This would be a positive effect not only for women, but for students from other backgrounds (first generation students, etc.) who may not be as comfortable with asking for support. It is important to encourage students to take explicit ownership of their PhD and their work. *We often lack specific metrics for how a PhD is progressing, so it can be easy for students to underestimate where they stand* in their PhD -- **make sure to provide constant feedback to students.** This **includes positive feedback** -- while it is easy to mention where students could improve, make sure to also explicitly state what they are doing well!

PIs can also de-emphasize competition for resources between students and instead promote group work with students. Make sure that each student has the opportunity to lead projects -- clearly designating a leader for different projects, if necessary -- as opposed to always letting students choose among themselves, which could result in one strong personality always presenting for a group. PIs should consider sending second authors to present work, or allow students to co-present, as well as encourage students to apply for grants and jobs for practice. **Advisors should find resources for women, encouraging their students to travel and seize opportunities.** This *includes resources of support for family/life balance -- for faculty, family policies are much better than those for students or post-docs, which can be practically non-existent.*

Many people expressed the need for *institutional* support for all of these activities. On an institutional level, several participants suggested that institutions could require a computer science course, or at least make sure that a computer science course was allowable and encouraged under the existing university requirements. Finding more ways to incorporate computer science into other intro classes at the university level would also give more people a

less intimidating introduction to the subject as well as demonstrate the broader applications. Harvey Mudd's model for computer science involves having two separate tracks for students with strong computer science backgrounds and those without a lot of experience, as well as making sure to send all their women students to Grace Hopper or other women dominated computer science spaces -- they have achieved parity for women in computer science at their institution.

On a community level, workshops to combat unconscious bias are extremely useful. However, *it can be difficult to reach outside the 'usual group of people', those who are already invested in such training.* One suggestion was to **incorporate unconscious bias training within workshops on other topics.** Another suggested sending male students to predominantly female spaces such as Grace Hopper. NSF could provide support for unconscious bias training -- potentially even making setting up a local event or sending someone to this type of training a grant requirement. NSF hosting workshops on how to avoid common mistakes, on dealing with uncomfortable situations, and on how to make situations more equitable would also be helpful. There exist a lot of 'punishment' and disincentive programs for bias, but training is really important -- one can learn an unexpected amount even when one is already familiar.

In NSF grants it is not always clear where in the budget breakdown and where in the annual and final reports to include broadening participation in computing (BPC) and outreach activities, or what the impacts on the community were. There is also the feeling that to put money towards BPC in a budget means taking away money from students and other areas. While NSF now expects reports on BPC, having institutions also require a report on these activities would increase both accountability and the visibility of these programs. **NSF should also provide a clear and easy way to find collection of studies and best practices relating to unconscious bias, broadening participation, and outreach. This could act as a compendium of prepared materials so that others -- like graduate students -- could run events for their own departments.**

Some universities aren't even aware that BPC programs exist -- one PI talked about how once their Dean became aware of the issues, they were able to look for external funding. *University administrators need motivation, resources, and upper administration support, as well as social pressure from other administrators, and motivation for higher level administrators with funds.* Thus, one questions might be - *is there a way to incentivize (reward) departments for achieving greater diversity goals?*

Faculty/Academia

At the ***institutional level***, there is a lack of support for women faculty, and in particular resources, including funding. In many institutions, women are not considered to be "under-represented"; consequently, CS departments may not be allocated additional resources to help recruit, retain, and support women. It is also important to create a system (culture) where "women's work" is valued as much as men's (e.g., AAUP data still indicates that women salaries

are lower). The current system lacks built-in advocates for women, and when women “don’t ask” and men do, women do not advance at the same rate. Thus, *creating a more transparent system for advancement is important*. Similarly, universities should re-evaluate their hiring processes (including search committees structure), insuring they keep track of statistics, rotate search committee membership (to avoid the common pitfall in academia, a tendency to “self-replicate”), provide access to implicit bias material and training, including avoiding of unintentional bias in reference letters. Moreover, there is also often a university administrative structure that does not facilitate faculty involvement in broadening participation in computing, whereas faculty should be integral to this effort.

Approaches to creating institutional change could be facilitated by having someone in place (with funding and/or ability to provide time relief) who can offer guidance, mentorship, resource information, and in general support career development of women. This would include professional development opportunities, e.g., such as negotiation skills and other training offered by organizations such as COACH.

It is also critical to avoid a common pitfall of overburdening women with service, i.e., **the women tax**. Women tend to participate in more committee (and in general, support) work; this is often for a good reason and desire to improve diversity. However, time is limited, and this creates enormous pressure on women faculty, leading to substantial time commitment, given the lack of critical mass. Since it is important to have proper representation on committees, review panels, etc, (and while it is still the case that service work is not properly recognized at promotion and merit review time), one approach to balancing such need with overburdening women is to create a system for tracking and appropriately rewarding such service. An example of this would be personal or discretionary funds and teaching release. We note that there is a concern that over-use of teaching release can lead to reduced time in front of a classroom, exacerbating the problem of lack of role models. Thus, an important question is “how can we provide time and recognition for diversity efforts”. To this end, **NSF might consider providing funding for women to reduce some of their other commitments, when their time is needed for service/outreach/diversity related efforts.**

It is also important to consider the **work-life balance** in academia. On one positive side, academia allows more flexibility than industry in terms of work/life balance, with substantially more flexibility in structuring one’s time. But, it is also the case that having a small child leads to lack of productivity (on the order of a couple of years), and there is a lot of pressure to succeed during a particular time interval on the tenure track (where appropriate parental leave policies are not always in place), with substantial stigma associated with failure and a woman’s failure being particularly visible. **NSF could consider providing additional support for parental leave.**

There is also a need to consider **accountability** in diversity efforts; although there is a lot of activity, it is not clear what efforts are getting results. To this end, **NSF should require greater openness in the process and in the Broader Impact reports**, e.g., what metrics are used, and exactly what efforts were explored. Based on this, NSF would be able to provide best practices

and details of successful programs that could be adapted by other institutions. Information from institutions on their hiring practices and statistics (such as application pool, selection committee makeup and turnover, etc) could be useful as well.

Finally, there is a general sense that it is difficult for universities to make progress without considering and **addressing the pipeline** in much earlier stages. As discussed earlier, the disparities between genders in computer science starts early and K-12 schools often suffer from a lack of teachers with computer science experience. **NSF could provide resources for tenured faculty to take a semester for CS teaching activities in K-12** (i.e., such resources can be used to provide partial support of faculty salary while university can hire an adjunct, if needed, to support the department's teaching requirements). NSF could also consider efforts that can identify the biggest factors contributing to pipeline issues and develop actionable strategies to mitigate these. It may also be helpful to look at other branches of science and what methods worked best for them (although, it is recognized that each discipline has its own reasons for lack of broader participation).

Community and Culture

Work/life balance is an issue that everyone from students to faculty deals with. Visibility is critical to one's career, but with a family travel becomes difficult and one cannot go to everything. Not showing up has real consequences -- less travel means less visibility and less networking. This reinforces the cycle of having fewer women in the room. **Grants that cover childcare, or provide travel funds for an additional caregiver and child, would help immensely.** However, once children are in school, childcare at a conference is not the answer anymore. **We also need to provide more remote access to conferences and workshops** -- such as video streaming, allowing panelists or speakers to video-conference in, and providing options for remote attendance.

We also need to be sure that conferences and workshops provide a good experience to everyone. ACM conferences have policies on harassment and a code of conduct -- NSF and other organizations should also codify something like this for their workshops as well. *Conferences are outside of the university system, so when something occurs there should be a clear way of reporting with and dealing with problems.* Conferences should have an identified person to report to on the organizing committee. Men often automatically expect women to fix these issues, putting the burden on women to make this work. Women are assumed to be more willing to put up with difficulties. But this needs to be everyone's priority. We need to normalize discussions of culture, diversity, and inequity at all of these events.

Unfortunately, a history of inequalities of computer science has long lasting effects. Bad experiences and history are communicated to younger generations, putting people off the field. It's commonly known that there are fewer women in CS the higher you go, that the industry is

not always a friendly or safe space for women. Senior women who have had horrible experiences in industry or academia and have retired early or left the field influence the next generation when they hear about it, coloring decisions.

It is also difficult to get into the field later in life -- alternative and non-traditional career paths are not well supported. *We need to reach out more to those who have followed non-traditional career paths, to re-engage people who have left the field and make sure that there is support for them to retrain and reintegrate themselves into computer science again.* It would be useful to have programs to train women locally so that they can build up their CV to the point that they can be hired. Similarly, **we could create post-industry post-doc positions to re-integrate people back into academia.** We need to consciously bring in women peers and success stories from industry so that there are positive role models for the next generation of computer scientists.

Conclusions

In conclusion, we summarize some of the main points of this report, while highlighting possible action items.

Summary

- The systems (and the broader CS) community needs a culture shift where we, as researchers, spend more time on communicating about our field and the corresponding important problems. We must also clearly and openly communicate the type of culture we wish to build.
- Addressing the pipeline (K-12) is critical. The language and culture of technology begin early, as does the experience of being intimidated by those who have (or are perceived to have) more expertise. The focus needs to be on teacher engagement and education as well as on providing tangible and practical materials. We also need to work to change the portrayal of computer science so that students see it not as just a tool, but as a fun and promising career.
- At the university level, many women (students and faculty) experience isolation. We need to make sure that support structures are in place for women and give them opportunities for networking and career advancement. We need a mob mentality of diversity with systematic ways to bring about changes. This includes reaching students and junior faculty in smaller colleges, particularly those who have not thought of or do not have resources to attend. We also often lack specific metrics for how a PhD is

progressing, so it can be easy for students to underestimate where they stand in their PhD.

- On a community level, workshops to combat unconscious bias are extremely useful. However, it can be difficult to reach outside the 'usual group of people', those who are already invested in such training. Incorporating unconscious bias training within workshops on other topics is a useful approach.
- To achieve change, university administrators need motivation, resources, and upper administration support, as well as social pressure from other administrators, and motivation for higher level administrators with funds.
- It is critical to avoid the women tax, i.e., overburdening women with service responsibilities without providing relief in other commitments. It is also critical to facilitate management of the work-life balance, particularly on the tenure track.
- Accountability and clear and transparent processes are also critical to moving forward as a community. For instance, conferences are outside of the university system, so when something occurs there should be a clear way of reporting with and dealing with problems. Similarly, universities should make sure that the processes of hiring and advancement are well-understood and transparent.
- As a community, we need to reach out more to those who have followed non-traditional career paths, to re-engage people who have left the field, and make sure that there is support for them to retrain and reintegrate themselves into computer science again.

Action Items

- *For PIs and community*
 - PIs should make sure to explicitly establish appropriate culture in their groups, including making sure that all students get opportunities for project leaderships.
 - PIs should be sure to be explicit, encouraging, and open about office hours, to support students who may be more tentative about approaching a professor or PI.
 - PIs should make sure to provide constant feedback to students; it is critical to this includes *positive* feedback.
 - Advisors should find resources for women, encouraging their students to travel and seize opportunities.
- *For NSF and institutions*
 - Universities and institutions like NSF should provide more professional development funding for communication.
 - Universities and institutions like NSF should facilitate a pipeline of qualified teachers with an exciting curriculum, e.g., by hosting workshops for high school

teachers and PhD students on how to teach computer science. To this end, NSF could also provide resources for tenured faculty to take a semester for CS teaching activities in K-12.

- Universities should work on incorporating computer science into non-major introductory courses to provide less intimidating and more practical introductions to the topic.
- The REU model has been very successful, as have a number of bridge programs, so it is worthwhile to consider development of such programs for students who are *not* at major research universities.
- Institutions should find resources of support for family/life balance (for faculty, family policies are much better than those for students or post-docs, which can be practically non-existent).
- NSF and institutions should hosting workshops on how to avoid common mistakes, on dealing with uncomfortable situations, and on how to make situations more equitable.
- NSF should provide a clear and easy way to find collection of studies and best practices relating to unconscious bias, broadening participation, and outreach. This could act as a compendium of prepared materials so that others (e.g., graduate students) could run events for their own departments. This can be combined with greater openness in the Broader Impact reports.
- Approaches to creating institutional change could be facilitated by institutions having someone in place (with funding and/or ability to provide time relief) who can offer guidance, mentorship, resource information, and in general support career development of women.
- NSF and institutions should consider providing funding for women to reduce some of their commitments, when their time is needed for service/outreach/diversity related efforts.
- NSF should consider providing additional support for childcare and parental leave, beyond what universities provide. Institutions should evaluate the effects of their parental leave policies.
- NSF and institutions should consider programs for supporting women (who have taken non-traditional paths) in building up their CVs so that they can re-engage with the profession, e.g., create post-industry post-doctoral positions (funds) to re-integrate people back into academia.

Resources

- Career and BPC Resources
 - BPCNet Resource Portal: <https://bpcnet.org/>
 - National Center for Women & Information Technology: <http://ncwit.org/>
 - N2 Women Networking Networking Women: <http://n2women.comsoc.org/>
 - The Annual Grace Hopper Celebration: <https://ghc.anitab.org/>
 - COACH: <https://coach.uoregon.edu/>
- Volunteering Resources & Opportunities:
 - Skype a Scientist: <https://www.skypeascientist.com/>
 - Dance your PhD: <http://gonzolabs.org/dance/>
 - Sit With Me: <https://www.sitwithme.org>
- Mentoring Resources
 - Systems program: <https://anitab.org/systems/>
 - Girls Who Code: <https://girlswhocode.com/>
- Communication
 - Alan Alda Center for Communicating Science: <https://www.aldacenter.org/>
 - On Presentations: the works of E.G. Tufte
 - “Making the Implicit Explicit: Creating Performance Expectations for the Dissertation” by Barbara Lovitts
- CISE BPC
 - To learn more about the CISE Broadening Participation in Computing (BCP) Pilot Effort, visit <https://www.nsf.gov/cise/bpc/> for additional information and resources.

List of Participants (122)

Abhishek Chandra	University of Minnesota
Agung Julius	RPI
Ahyoung Lee	University of South Dakota
Ali Ahmadinia	California State University San Marcos
Ali R. Butt	Virginia Tech
An Wang	Case Western Reserve University
Anshul Gandhi	Stony Brook University
Anton Burtsev	University of California, Irvine
Ardalan Amiri Sani	UC Irvine
Aruna Balasubramanian	Stony Brook University
Avani Wildani	Emory University
Bhuvan Urgaonkar	Pennsylvania State University
Bo Wu	Colorado School of Mines
Brocanelli, Marco	Wayne State University
Calton Pu	Georgia Institute of Technology
Carlos Maltzahn	UC Santa Cruz
Chandra Krintz	UC Santa Barbara
Chen Liu	Clarkson University
Cong Liu	University of Texas at Dallas
Dazhao Cheng	UNC Charlotte
Derek Doran	Wright State University
Desheng Zhang	Rutgers University
Dilma Da Silva	Texas A&M University
Divyakant Agrawal	UC Santa Barbara
Dong Dai	University of North Carolina at Charlotte
Dongyan Xu	Purdue University
Ehab Al-Shaer	Professor and Director
Fan Yao	University of Central Florida
Fengwei Zhang	Wayne State University
George Amvrosiadis	Carnegie Mellon University
George Kesidis	Pennsylvania State University
Guohong Cao	The Pennsylvania State University
Gurdip Singh	Syracuse University
Hai Zhou	Northwestern University
Haibo Zeng	Virginia Tech
Hang Liu	University of Massachusetts Lowell
Harry Xu	UCLA
Honggang Wang	University of Massachusetts Dartmouth
Hui Lu	SUNY Binghamton
Hui Zhao	University of North Texas
Ishfaq Ahmad	University of Texas at Arlington
Jacob Sorber	Clemson University
Jennifer Weston	AAAS Fellow at NSF
Jia Rao	The University of Texas at Arlington
Jingtong Hu	University of Pittsburgh
Jinyuan Stella Sun	University of Tennessee
Kenneth Calvert	NSF
Kirk Cameron	Virginia Tech

Krishna Kant	Temple University
Kyle Hale	Illinois Institute of Technology
Leana Golubchik	USC
Lina Sawalha	Western Michigan University
Linhai Song	Penn State University
Lu Peng	LSU
Mahadev Satyanarayanan	Carnegie Mellon University
Mai Zheng	Iowa State University
Matt Mutka	National Science Foundation
Meikang Qiu	Columbia University
Meng Yu	Roosevelt University
Michael M Swift	University of Wisconsin--Madison
Ming-Chun Huang	Case Western Reserve University
Mohsen Amini Salehi	University of Louisiana Lafayette
Murat Demirbas	SUNY Buffalo
Murtuza Jadliwala	University of Texas at San Antonio
Nathan Dautenhahn	Rice University
Nathan Fisher	Wayne State University
Octav Chipara	University of Iowa
Peter Dinda	Northwestern University
Prabhat Mishra	University of Florida
Qing Yang	University of North Texas
Qingyang Wang	Louisiana State University
Qun Li	William & Mary
Raja Sambasivan	Boston University
Richard Han	University of Colorado at Boulder
Robert Ricci	University of Utah
Roberto Palmieri	Lehigh University
Roger Chamberlain	Washington University in St. Louis
Rujia Wang	Illinois Institute of technology
Samee U. Khan	NSF
Sanjoy Baruah	Washington University in St. Louis
Satish Puri	Marquette University
Saurabh Bagchi	Purdue University
Shan Lu	University of Chicago
Shaolei Ren	UC Riverside
Sheng Wei	Rutgers University
Shivakant Mishra	University of Colorado at Boulder
Song Fang	University of Oklahoma
Song Jiang	UT Arlington
Songqing Chen	George Mason University
Spyros Blanas	The Ohio State University
Stacy Patterson	Rensselaer Polytechnic Institute
Tajana Rosing	UCSD
Tam Chantem	Virginia Tech
Tanu Malik	DePaul University
Tarek Abdelzaher	UIUC
Tian Guo	Worcester Polytechnic Institute
Tian Lan	George Washington University

Ulrich Kremer	Rutgers University
Umit Ogras	Arizona State University
Vaneet Aggarwal	Purdue University
Wei Gao	University of Pittsburgh
Wei Li	Georgia State University
Wei Yang	University of Texas at Dallas
Wei Zhang	Virginia Commonwealth University
Weisong Shi	Wayne State University
Xiang Chen	George Mason University
Xiaoning Ding	New Jersey Institute of Technology
Xipeng Shen	NCSU
Yan Solihin	University of Central Florida
Yanmin Gong	Oklahoma State University
Yao Liu	SUNY Binghamton
Yi Zhu	Hawaii Pacific University
Yinqian Zhang	The Ohio State University
Yiran Chen	Duke University
Yiying Zhang	Purdue University
Yuan Tian	University of Virginia
Yue Cheng	George Mason University
Yuguang Fang	University of Florida
Yuhao Zhu	University of Rochester
Yurvaj Agarwal	CMU
Yuzhe (Richard) Tang	Syracuse University
Zhiling Lan	Illinois Institute of Technology

Workshop Agenda

7:30am-8:00am	Breakfast (<i>Foyer of Auditorium/Maple</i>) and Registration (<i>Auditorium Foyer</i>)
8:00am-8:20am	Introduction to CSR program (NSF Program Directors)
8:20am-8:50am	Lightning Round
	Panel Moderator: Weisong Shi, Wayne State University
8:50am-9:50am	Panelists: Leana Golubchik, University of Southern California Shan Lu, University of Chicago Robert Ricci, University of Utah Mahadev Satyanarayanan, Carnegie Mellon University Gurdip Singh, Syracuse University Dongyan Xu, Purdue University
9:50am-10:10am	Break (<i>3rd Floor Foyer</i>)
	Breakout Session I
10:10am-12:10am	<i>Track 1: Security by Systems Design (Madrona , Chairs: Dilma Da Silva and Dongyan Xu)</i> <i>Track 2: IoT Systems Challenges (Juniper , Chairs: Tarek Abdelzaher and Gurdip Singh)</i> <i>Track 3: Women in Computing (Cottonwood, Chairs: Leana Golubchik and Jennifer Weston)</i>
12:10am-1:40pm	Lunch (<i>Regency Ballroom D, E, F, G</i>)
	Lunch talk Kenneth Calvert, Division Director, NSF/CISE/CNS <i>Title: TBD</i>
	Breakout Session II
1:40pm-3:40pm	<i>Track 1: Security by Systems Design (Madrona , Chairs: Dilma Da Silva and Dongyan Xu)</i> <i>Track 2: IoT Systems Challenges (Juniper , Chairs: Tarek Abdelzaher and Gurdip Singh)</i> <i>Track 3: Women in Computing (Cottonwood, Chairs: Leana Golubchik and Jennifer Weston)</i>
3:40pm-4:00pm	Break (<i>3rd Floor Foyer</i>)
4:00pm-4:30pm	Breakout Report

4:30pm-4:45pm

Closing Remark

4:45pm-6:00pm

Poster Session (*Laurel and Larch*)

6:00pm-8:00pm

Dinner (*Regency Ballroom D, E, F, G*)

Appendix A: Introduction slides for the workshop

Appendix B: Slides from the panelists