# Security of IoT Systems: Design Challenges and Opportunities

Teng Xu, James B. Wendt, and Miodrag Potkonjak
Computer Science Department
University of California, Los Angeles
{xuteng, jwendt, miodrag}@cs.ucla.edu

***Abstract*—Computer-aided design (CAD), in its quest to facilitate new design revolutions, is again on the brink of changing its scope. Following both historical and recent technological and application trends, one can identify several emerging research and development directions in which CAD approaches and techniques may have major impacts. Among them, due to the potential to fundamentally alter everyday life as well as how science and engineering systems are designed and operated, the Internet of Things (IoT) stands out. IoT also poses an extraordinary system replete with conceptual and technical challenges. For instance, greatly reduced quantitative bounds on acceptable area and energy metrics require qualitative breakthroughs in design and optimization techniques.**

**Most likely the most demanding of requirements for the widespread realization of many IoT visions is security. IoT security has an exceptionally wide scope in at least four dimensions. In terms of security scope it includes rarely addressed tasks such as trusted sensing, computation, communication, privacy, and digital forgetting. It also asks for new and better techniques for the protection of hardware, software, and data that considers the possibility of physical access to IoT devices. Sensors and actuators are common components of IoT devices and pose several unique security challenges including the integrity of physical signals and actuating events. Finally, during processing of collected data, one can envision many semantic attacks.**

**Our strategic objective is to provide an impetus for the development of IoT CAD security techniques. We start by presenting a brief survey of IoT challenges and opportunities with an emphasis on security issues. Next, we discuss the potential of hardware-based IoT security approaches. Finally, we conclude with several case studies that advocate the use of stable PUFs and digital PPUFs for several IoT security protocols.**

## I. Introduction

Six decades of computer-aided design (electronic design automation) have witnessed numerous changes in its research and development focus. For example, the dominant design metrics have changed from area (number of transistors) in the 70's, to delay in the 80's to energy in the 90's. These shifts correspond to technology changes where, initially, the cost of transistors is replaced with the need for speed, followed by power and energy consumption emerging as the most constraining metrics. Recently, various security metrics have attracted a great deal of attention. If we analyze at the level of abstraction, the scope has shifted from physical design to logic synthesis to register transfer level to behavioral synthesis to system design. Targeted general purpose architectures have been changing from mainframes to minicomputers, workstations, personal computers and mobile processors. Similarly, application specific computing has shifted its targets from audio to video and, more recently, to multimedia and networking devices. We finish our survey with a discussion on process variation, which has had a significant impact on CAD algorithms and techniques.

Nevertheless, several properties of CAD processes have become permanent. For instance, a great emphasis has been consistently placed on accurate modeling of relevant design metrics. The most consistent CAD variable (and of paramount importance) is the optimization of systems with a large number of strongly interacting components. This type of synthesis is intrinsic to several types of important emerging systems such as data centers and platforms for support of storing and processing big data. In particular, it is ideally suited for handling millions, if not billions, of distributed communicating devices envisioned in the Internet of Things (IoT).

IoT is a rapidly emerging paradigm in which the essential concept is that a great variety of objects are instrumented in such a way that they can be queried and operated over the Internet either directly by the users or by programs that encapsulate their behavior and objectives [1] [2]. IoT will revolutionize the ways in which individuals and organizations interact with the physical world as well among themselves. For example, the interaction with home devices, cars, customer items, industrial plants, and weaponry will be fundamentally altered. Many services, such as health, learning, and resource management, will be provided in new ways that are novel, better organized, and customer-customized. Radio-frequency identification (RFID) tags, as used for inventory management by companies such as Walmart, provide a first glimpse into a very rudimentary IoT generation.

The practical realization of IoT requires the development of a number of new versions of platforms and technologies including device and process identification and tracking, sensing and actuation, communication, computational sensing, semantic knowledge processing, coordinated and distributed control, and behavioral, traffic, and user modeling [3]. The realization of IoT subsystems will be subjected to numerous constraints that include cost, power, energy, and lifetime. However, there is a wide consensus that the most challenging of requirements will be security. It is widely acknowledged that the potential for malicious attacks can and will be greatly spread and actuated from the Internet to the physical word. Hence, security of IoT is of essential importance.

One should also consider a great diversity of IoT systems from fully organized to small individual nodes [4] [5] [6]. For example, things such as cars, airplanes, and industrial

978-1-4799-6278-5/14/$31.00 ©2014 IEEE

equipment allow for much more expensive instrumentation with much high power and energy budgets in comparison to household IoT devices, such as those for food, energy, and paper documents. Therefore, although for full impact, generic algorithms and protocols are required, different customized solutions are also mandatory. This is in particularly true for security solutions.

IoT security encompasses several layers of abstraction and a number of dimensions. The abstraction levels range from physical layers of sensors, computation and communication, and devices to the semantic layer in which all collected information is interpreted and processed. We expect that a majority of security attacks will occur at the software level because it is currently most popular and can simultaneously cover a large number of devices and processes. From a research point of view, most novel attacks are on physical signals and, in particular, semantic attacks during data processing and decision making steps. It is important to observe that the lowest security at any level and at any dimension determines the overall security.

A significant percentage of IoT devices will operate in passive mode without batteries. Their energy will either be harvested or received using a wireless medium. Many of these systems allow for only very minimal hardware, and thus, require an ultra compact security solution with an ultra small footprint and energy budget, since many IoT devices often operate in unprotected and potentially even hostile environments.

Our main claim in this paper is that hardware-based security is ideally suited to answer IoT security requirements. However, in order to realize the full potential of hardware-based security, very significant additional research and engineering issues have to be addressed in novel and creative ways. Hardware-based security provides a natural starting point for the realization of IoT protocols and procedures due to their very low area and energy requirements. They are also naturally more resilient against side-channel and physical attacks. Also very importantly, is that they enable the creation of secure and trusted information flows [7]. Finally, they provide elegant and efficient solutions to several problems that classical cryptography has not been able to solve, such as secure location discovery.

At the same time, it is important to recognize that hardware-based security primitives and protocols have several significant limitations. Among them, three are dominant. The first is that, until the invention of the public physical unclonable function (PPUF), their application was restricted to secret key protocols. While the PPUF eliminates this restriction, the first PPUF generation induced significant time and energy overhead on at least one participating party. The second is that the key hardware-based security physical unclonable function (PUF) is rather unstable with respect to operational (e.g. supply voltage) and environmental (e.g. temperature) conditions as well due to unavoidable device aging. The third drawback is that the first generations of PUFs are analog circuitry and therefore are difficult or at least cumbersome to integrate into digital designs.

We briefly survey two recent hardware security results. The first is a very simple technique that transforms several classes of analog PUFs into stable devices with respect to operational conditions. The procedure has very small hardware overhead and no delay and energy overheads. The key idea is to use only a small subset of challenges that are stable under a wide range of conditions. Although the number of used challenges is significantly reduced from the original challenge space, their cardinality is still exponential in the number of bits used by challenges. This approach is applicable on several popular PUF structures such as delay arbiter and ring oscillator-based PUFs.

The second is the digital PUF hardware security primitive. While it is initialized using stable analog PUFs, it is a digital circuit with very small gate counts, low latency, and ultra low energy requirements. Therefore, it can be directly integrated with regular digital designs and facilitate secure and trusted information flow, elimination of side-channels, and public key security protocols with latency of only a very few number of gates. Digital PUFs can be used for the creation of new hardware security primitives such as distributed and synchronized hardware random number generators.

One of our technical objectives is to initialize the quest for conceptually new hardware security primitives. For example, we envision that physical properties of hardware and materials used for implementation can create novel security primitives that cannot be realized using classical mathematical and algorithmic techniques. Specifically, the unidirectional evolution of material properties provides the potential for detecting if there was any interaction with a particular sensing device since the legitimate user's last interaction. One physical phenomenon with such potential is device aging, however may not be fully practical due to its long term irreversibility. Most likely, a better candidate for the creation of such a hardware security primitives is the memristor, where current-voltage trajectories are such that passing through any particular voltage-current point is very difficult to repeat.

We conclude our discussion with a proposal to search for universal hardware security primitives that can be used for diverse tasks such as for malicious alternations of design, for cryptographical and trust protocols, and for ensuring secure and trusted information flow.

## II. IoT Security Desiderata

The IoT security desiderata can be grouped into two broad classes. The first class consists of required security tasks. As usual, the primary potential difficulties are related but contradictory requirements of different tasks. For instance, the strength of authentication and trust are in direct contradiction with a criterion of privacy. The second class of desiderata is related to design metrics such as cost, size, latency, and, in particular, energy requirements. As usual, the key impact of these requirements is that they greatly constrain acceptable security solutions.

The most important security requirements include authentication and tracking, data and information integrity, mutual trust, privacy, and digital forgetting. We expect that a dominant percentage of computational sensing, decision making, communication, and activity organization will be conducted in data centers. Hence, there is a need for ensuring security in data centers as well coordinating security between data centers and distributed IoT devices [8].

It is expected that billions of devices will be a part of the IoT ecosystem. Each of these nodes should have a unique identifier. In addition, at any point in time the IoT infrastructure should be able to track each item. Another level of difficulty is that many nodes will be placed in high densities and access to them may be hindered or even blocked.

It is important to ensure that all collected data is authentic. Some IoT sensors may have high bandwidth and low latency (real-time) data collection rates. Therefore appropriate data integrity techniques such as encryption and watermarking are required.

There is an essential need for ensuring that each user can be guaranteed that the data presented by an IoT device is trusted, i.e. that it is indeed collected by the stated sensor at its stated location and at its stated time. Recently, several schemes have been proposed for ensuring IoT trust. These solutions are based on hardware security primitives and should be further optimized in terms of cost and energy. Also, hardware and software attestation techniques may be used for trust related tasks. Interestingly, another important problem is operator trust, in which sensors and IoT devices can authorize and trust the instructions of IoT users.

A number of definitions of privacy have been proposed [9] [10]. They are certainly useful in a number of current scenarios, however, a completely new level of complexity is posed by IoT. Probably the most difficult privacy task is one where the attacker integrates information from different sets and modalities at the semantic level. Combining different data from different sources of information at the semantic level can result in the extraction of unexpected information.

Data revocation (i.e. digital forgetting) is the process of provably deleting all copies of a data set [11]. In addition to the tremendous amount of sensory data that IoT devices will collect, there will be huge data sets related to communication activities between various users and IoT devices. It is plausible to expect that a significant percentage of this data will contain important information and knowledge about the users and their actions and interests.

There are several data revocation techniques proposed in classical cryptography [12]. All of them are based on the simple and elegant idea that encrypted data is effectively deleted if the required decryption key is deleted. There are also several techniques that employ distributed data storage so that data is deleted due to unavoidable social and technical processes [13]. These ideas are valuable and essential to IoT systems which can collect large amounts of data that can seriously impact the privacy of many individuals and even compromise the security of economic entities and government institutions.

A large percentage of devices will depend on harvested energy. In order to reduce energy consumption, computation tasks will often be offloaded to data centers. Communication will often use technologies that require less energy than those currently widely used. Most likely, near-field communication (NFC) will greatly increase its market share. Other highly constrained metrics include cost and area. It is likely that new packaging and integration technologies will emerge. An important observation is that many deployed devices should be in operation for years if not decades. Replacing batteries can be expensive and impractical. Therefore, it is expected that regardless of technology progress, most likely energy will become the most expressed limitation.

Intel has demonstrated that passive RFID can be utilized to form WISP network in which each device collects energy from a querying device [14]. In their initial implementation, a very limited set of security and cryptographic tasks can be executed on the items themselves. However, the use of hardware-based security primitives may alter that vision.

We conclude this section with a brief summary of system software requirements for IoT systems. These requirements may serve as a check list for developers of IoT systems and CAD research. Currently, it is widely assumed that security primitives created by classical cryptography will be used. We claim that hardware security primitives can be used to create all expected protocols at a fraction of hardware and energy requirements with much higher resiliency against side-channel attacks.

IoT operating systems will have to provide a number of services and system software must be consistent with standard Internet protocols and services. Targeted security protocols include TLS, IPSec, VPN, SSH, SFTP, HTTPS, SNMP, and secure email [15]. Also, standard encryption and decryption services are required. Special attention should be placed on equipping each IoT device with a secure bootloader and automatic failback. In addition, system software should be equipped with mechanisms for detecting and reporting physical and in particular side-channel attacks and secure wireless links should be provided.

## III. PUBLIC PUF

PPUFs have extended the practicality of PUFs by enabling the creation of public key protocols. While PUFs require that their characterization and structure remain hidden and secret, the PPUF design and characterization is disclosed to the public. In this way, the design itself becomes the public key. The public nature of the PPUF makes it the premier primitive for securing IoT devices since its susceptibility to physical and side-channel attacks is eliminated. Furthermore, PPUFs have small area footprint and orders of magnitude lower energy consumption than their traditional cryptographic counterparts.

### A. XOR Network Delay PPUF

Beckmann et al. proposed the first PPUF model along with accompanying protocols for public key cryptography [16]. The public key consisted of the complete characterization of the design, including gate-level characteristics, such as leakage energy and delay. Due to the effects of process variation, inherent doping concentrations variances and line-edge roughnesses manifested as different values of effective channel lengths and threshold voltages which ultimately effect leakage energy and delay of each transistor. In this way, the public key was random, and unclonable, however, still able to be simulated, although very arduous to do so due to the design.

The architecture of Beckmann's PPUF is a gridded network of XOR gates. Due to inherent intrinsic manufacturing variability, the physical characteristics of each XOR gate differ. Specifically, due to variations in doping concentrations and line

| | | | |
|---|---|---|---|
| $\delta_{ae}$ | 3 | $\delta_{ei}$ | 7 |
| $\delta_{be}$ | 4 | $\delta_{fi}$ | 2 |
| $\delta_{cf}$ | 5 | $\delta_{ej}$ | 5 |
| $\delta_{df}$ | 4 | $\delta_{fj}$ | 8 |
| $\delta_{ag}$ | 4 | $\delta_{gk}$ | 5 |
| $\delta_{bg}$ | 5 | $\delta_{hk}$ | 4 |
| $\delta_{ch}$ | 8 | $\delta_{gl}$ | 7 |
| $\delta_{dh}$ | 4 | $\delta_{hl}$ | 3 |

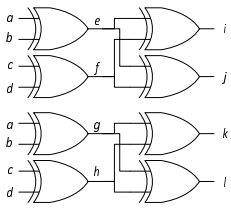


Fig. 1: Differential PPUF booster cell example. The delay of a rising edge from input $i$ to output $j$ is denoted by $\delta_{ij}$.

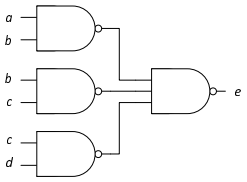


Fig. 2: Differential PPUF represser cell example.

edge roughness, differences in threshold voltages and effective channel lengths emerge. When sending an input through the gates, the rising edges will race throughout the gridded network. Each XOR gate will transition upon the arrival of a new rising or lowering edge and emit the output corresponding to its input at that particular time. These signals will propagate throughout the circuit, causing multiple transitions at each XOR gate. The input challenge is a combination of both the input vectors ($x(0)$, $x(1)$) as well as a time delay ($t$) at which to read the outputs of the network.

The design of this PPUF takes advantage of the glitching effects of multiple propagating and delayed signals throughout the XOR network. This architecture also requires ultra accurate, ultra precise, and ultra high frequency clocks in order to operate on the physical PPUF, and, for larger PPUFs, requires much longer simulation times for the communicating parties wishing to initiate authorized contact with a a PPUF owner.

### B. Differential PPUF

The differential PPUF eliminates the need for ultra accurate clock manipulation for high precision timing as well as long simulation times [17]. Like its predecessor, the unclonability of the differential PPUF relies on the inherent randomness in manufacturing variability, specifically manifesting as variances in gate delays. A key novelty of this architecture is that the challenge vector is reduced from two input vectors plus a timestamp to a single input vector. This eliminates the need for accurate clock capturing of glitch temporal characteristics because it only requires the measurement of the frontier signal.

Consider the differential PPUF booster example depicted in Figure 1. If the input switches from $0000$ to $0101$, output $i$ will switch at times 6 and 11, $j$ at times 9 and 12, $k$ at times 8 and 9, and $l$ at times 7 and 12. By placing an arbiter with inputs from $i$ and $k$ and a second arbiter with inputs from $j$ and $k$, we eliminate the need for high precision timing by only capturing the first winner of the two paths. Hence, only frontier signals are necessary. However, since one has to simulate only these frontier signals, an architecture in which one can predict which

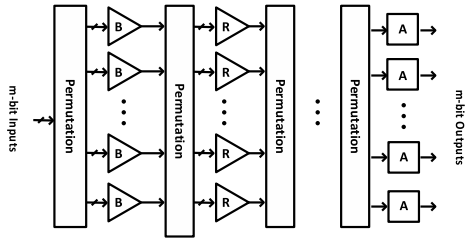


Fig. 3: Device aging-based matched PPUF architecture.

frontier signals will not cause transitions is not secure. Thus, in addition to booster cells, the differential PPUF includes represser cells consisting of a NAND gate network to terminate subsets of propagating signals in an unpredictable manner, such as the one depicted in Figure 2. Together, the alternation of booster cells followed by represser cells creates a highly non-linear system that is exponentially hard to simulate with a linear size increase.

### C. Device Aging and Matched PPUFs

All previously proposed PPUFs, including the differential PPUF, are potentially subject to long-term reverse engineering attacks. The device aging-based PPUF design eliminates the possibility of these attacks through dynamic reconfiguration. The key idea is to leverage device aging to alter the PUF's physical properties, thus changing its behavior. Specifically, device aging through techniques such as NBTI can permanently alter the threshold voltages of gates, thus increasing their delay [18].

The key limitation to the original device aging-based PPUF, along with all other previously designed PPUFs, is that they employ a large time gap between execution and simulation to enable public key communication. While each PPUF design provided faster simulation time on the part of the authentication party than its predecessor, the fact remains that at least one participating party requires significant resources for communication in comparison to the participant in possession of the physical PPUF.

The matched PPUF architecture attempts to remove the need for simulation entirely by supplying both communicating parties with physical PPUFs that are globally unique post fabrication, but can be made identical through a novel matching procedure. This procedure is executed in such a way that only the two participating PPUFs become identical while it is probabilistically negligible that a third snooping adversary is able to match as well.

The architecture of the matched PPUF utilizes booster cells and represser cells, similar to those designed for the differential PPUF and depicted in Figure 1 and 2. The first matched PPUF architecture consisted of $h$ stages of $b$ booster cells followed by $r$ represser cells, and interstage networks connecting them as depicted in Figure 3. Matching is done post fabrication when two communicating parties, each with their own PPUF, enable, disable, and age their individual sets of gates until a portion of gates are matched between the two of them and their PPUFs now implement the same functionality.

An adversary snooping on the matching protocol is still only able to match 58.3% of the configuration [19]. Attempting

to match the remaining gates through simulation or special purpose hardware is not quick enough to successfully imitate the physical PUF. Furthermore, the task is made even more difficult by increasing the size of the PPUF, thereby increasing the total number of unmatched adversarial gates which has an exponential increase in simulation complexity.

The matching PPUF was later improved upon to allow for $n$-party communication and does not require device aging upon matching, but is aged immediately post-fabrication to set quanta. Multiple party matching is enabled on-the-fly through a quantized matching scheme [20].

### D. NanoPPUF

A very natural enabling technology for PUFs are emerging nanotechnologies. Recently, PPUF designs have been proposed utilizing III-V nanowires and memristors [21] [22]. Not only do the enabling components of these devices (namely, the nanotechnologies themselves) exhibit very non-linear input-output responses (this is represented by I-V curves), and thus, are able to better satisfy Shannon's confusion and diffusion principles, they also exhibit randomness in their synthesis processes. Most importantly though, they contribute to the new notion of bidirectionality which enables an entirely new security dimension.

Current IC designs rely on the most basic component of the IC, the transistor. This three terminal device has a source, a sink, and a gate, and can only be operated in a unidirectional manner. However, a memristor is a two terminal device in which either a negative or positive voltage can be placed across it, and will output differently depending upon its current state. In current IC implementations of PPUFs, input vectors are applied to input pins and flow through a circuit network ultimately producing an output at the assigned output pins. By utilizing particular nanotechnologies, the input and output pins no longer need to be necessarily statically assigned, but can be chosen at runtime. This introduces an entirely new dimension to the PPUF design space.

The NanoPPUF architecture is composed of a network of non-linear nanotechnology components gridded together as depicted in Figure 4. A challenge consists of a set of input locations and voltages. Upon applying the inputs the signals travel throughout the non-linear network components until settling at the remaining pins, yielding the output. A novel polyomino partitioning scheme enables for quick authentication while maximizing security [22].

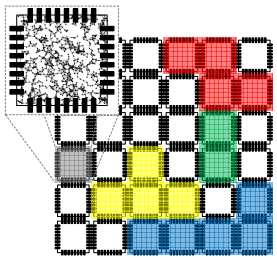

Fig. 4: Nanotechnology-based PPUF example. Each cell consists of a random network of non-linear components. The shaded regions represent example polyomino partitions.

## IV. Hardware Obfuscation

IoT devices can often be installed in insecure or unattended locations which can often be physically accessed by an attacker. Hardware logic obfuscation is a technique that protects the hardware intellectual property of these devices and secures on-chip information specifically be preventing reverse engineering attacks.

Wendt et al. have developed two techniques for hardware obfuscation using the standard delay-based PUF [23]. The first method connects pairs of wires together in such a way that a PUF's output bit determines whether the two wire values are switched or not. Since the original chip designer is the only one who knows the correct functionality of the circuit (i.e. which wires should and should not be swapped) and is the only one who knows the functionality of each PUF after fabrication, he can set the PUF inputs such that the circuit will function correctly. Furthermore, wire swappings are placed in such a way that they produce an exponential number of possible configurations. Furthermore, the specific "key" used for each IC will be different for each IC since each IC will have a unique set of PUFs.

The second technique for hardware obfuscation is the direct replacement of logic with a PUF and piece of programmable fabric. Xu et al. improve upon this design by employing the first digital PUF for direct logic obfuscation [24].

## V. Digital PUF

The concept of the digital PUF was first proposed in [25]. There are two major components that compose the digital PUF: a stable delay-based PUF and a lookup table (LUT) network. The analog delay-based PUF is made stable by the techniques discussed below.

### A. Standard Delay-based PUF Stability

The concept of the delay-based PUF was first proposed by Pappu et al. [26]. The PUF consists of two delay paths with nominally the same propagation delay. However, due to process variation, the actual delay in the two paths differ. An arbiter placed at the end of the two paths generates the output of the PUF based on which path is quicker.

A major problem of the delay-based PUF is its instability. Because the propagation delay is extremely sensitive to the external environment (e.g. temperature and voltage) the delay of the two paths are also heavily influenced by any variations. As a result, the PUF is considered not stable.

Figure 5 depicts an example of the delay-based PUF. Each stage is controlled by a single challenge bit. A rising edge is sent through the first stage and depending upon the challenge bit will either swap the trajectories (red) or remain (blue). Path differences for different challenges are depicted in Table I. Some challenges (e.g. 1101) are able to produce larger delay differences compared to other challenges. The motivation is that if the first path is much faster than the second path, even if environmental conditions change and affect the delays of both paths, it is with high probability that the first path will still be faster than the second. We label such a challenge and

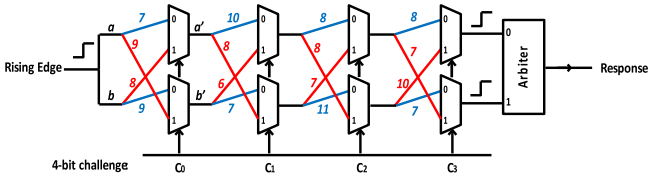


Fig. 5: An example of delay-based PUF with 4-bit challenge.

| $C$ | $\Delta d$ | $C$ | $\Delta d$ | $C$ | $\Delta d$ | $C$ | $\Delta d$ |
|---|---|---|---|---|---|---|---|
| 0000 | -1 | 0001 | 5 | 0010 | -1 | 0011 | 5 |
| 0100 | -2 | 0101 | 6 | 0110 | 0 | 0111 | 4 |
| 1000 | 0 | 1001 | 4 | 1010 | -2 | 1011 | 6 |
| 1100 | -3 | 1101 | 7 | 1110 | 1 | 1111 | 3 |

TABLE I: Delay differences ($\Delta d$) between paths for all challenges ($C$) in Figure 5.

any other challenges that are resilient to such environmental changes as stable inputs.

Xu et al. define the notion of *delay ratio* to quantify the relative delay difference between two paths [25]. The delay ratio is defined as the delay difference divided by the minimum of the two delays in question. Table II depicts the stability of inputs with a given delay ratio over varying operational conditions. Note that when the delay ratio reaches some threshold (e.g. 10%), the output of the corresponding challenges remains stable regardless of environmental conditions. Thus, these challenges are considered stable.

### B. Lookup Table Network

The lookup table network is formed with a set of randomly connected LUTs in a hierarchical structure as shown in Figure 6. Random shuffling is applied between levels. For a LUT network with $m$ inputs and $n$ outputs, the hierarchical structure provides a mapping between inputs and outputs. From an attacker's perspective, it is extremely difficult to derive the LUT connections and configurations directly from the mapping. Therefore, if the attacker wants to implement a hardware block that generates the same mapping, there is no way to reproduce the same original LUT network, instead, the attacker can only use brute force to implement the complex inputs-outputs mapping. Since the size of the mapping grows exponentially with the size of the inputs, a linear increase in its size creates an exponential increase in difficulty for an attacker.

The use of randomly connected LUT networks for security is first proposed in [27]. The outputs from a LUT network exhibit excellent statistical security properties; for example, they pass all NIST randomness tests [28]. They also satisfy the avalanche criterion. In terms of application, they enable both traditional protocols such as public key communication, as well as new protocols such as remote trust. They require orders of magnitude less energy in comparison to traditional cipher blocks, and, most importantly, the LUT network is purely digital, and thus, resilient against variations in environmental and operational conditions.

| Temp. | Delay Ratio (T=300K) | | | | | | |
|---|---|---|---|---|---|---|---|
| | 0.04 | 0.05 | 0.06 | 0.07 | 0.08 | 0.09 | 0.1 |
| 250K | 0.984 | 0.986 | 0.996 | 0.998 | 1 | 1 | 1 |
| 350K | 0.982 | 0.986 | 0.993 | 0.998 | 1 | 1 | 1 |
| 400K | 0.954 | 0.974 | 0.986 | 0.991 | 0.997 | 1 | 1 |

TABLE II: Probability that outputs of the 64-bit PUF are stable over varying temperatures for different delay ratios.

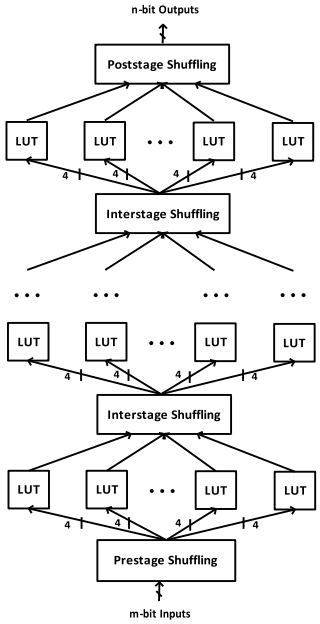


Fig. 6: Lookup table network with $m$ inputs and $n$ outputs.

### C. Lookup Table Initialization

Before operation, the LUT network must be initialized and configured. Figure 7 depicts the initialization process. The user chooses a set of stable challenges to apply to the supporting delay-based PUFs. Then the stable outputs are used to initialize the cells in the LUT network. Note that both the initialization challenges as well as their assignment in the LUT network are chosen by the user, thus preventing malicious manufacturers from subverting the system.

This integration of the stable delay-based PUF with the LUT network comprises the digital PUF. The design preserves both the unclonability of the analog PUF as well as the digital property of the LUT network. By applying only stable challenges to the delay-based PUF at initialization, the output stability is guaranteed. Together with the intrinsic digital property of the LUT network, the whole system is resilient against environmental variations and can be placed as any other component directly inside digital logic. Furthermore, because the initialization is dependent upon the delay-based PUF and the delay-based PUF can not be reproduced, the system remains unclonable. Therefore, even if an attacker steals information regarding what stable challenges are used in initialization, he cannot know the actual functionality of the LUT network without reverse engineering delay-based PUF.

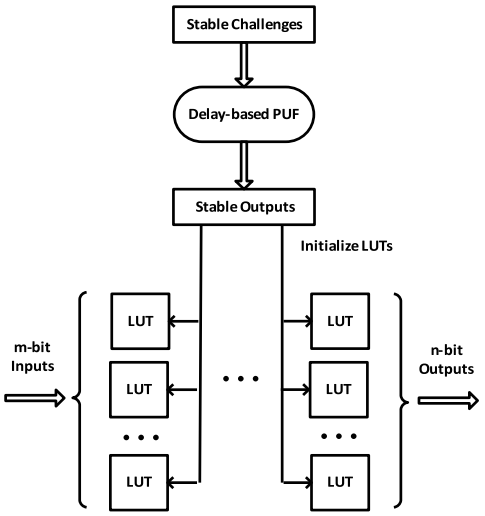

Fig. 7: LUT network initialization.

## VI. Conclusion

The Internet of Things (IoT) will connect billions of devices to the Internet and redefine how individuals, economic entities, and government organizations will interact with the physical world. The number of IoT devices will outgrow the number of personal computers and even mobile phones by several orders of magnitude. Optimization intensive CAD techniques compounded with their traditional accurate modeling are naturally suited to enable the design of highly optimized IoT devices. Two paramount constraints for IoT devices are energy and security. Both constraints can be addressed well using CAD techniques and we analyze several recently proposed hardware security primitives that enable strong and comprehensive security under very strict cost (hardware) and security constraints. We explain how stable PUFs can be created by restricting challenges to ones that are stable under a great variety of operational conditions. We also briefly surveyed recently a proposed digital PUF that enables the direct use of this hardware security primitive inside an arbitrary digital logic to create secure information flow and public key protocols that require only one clock cycle. Our strategic goal in this paper is to provide a starting points for creating CAD techniques that answer IoT design requirements.

## Acknowledgements

This work was supported in part by the NSF under award CNS-0958369, award CNS-1059435, and award CCF-0926127, and by Samsung under award GRO-20130123.

## References

[1] N. Council, "Disruptive civil technologies: Six technologies with potential impacts on us interests out to 2025," in *Conference Report CR*, 2008.

[2] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[3] A. Juels, "RFID security and privacy: A research survey," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 381–394, 2006.

[4] J.-P. Vasseur and A. Dunkels, *Interconnecting smart objects with IP: The next internet*. Morgan Kaufmann, 2010.

[5] J. Hui, D. Culler, and S. Chakrabarti, "6LoWPAN: Incorporating IEEE 802.15. 4 into the IP architecture–internet protocol for smart objects (IPSO) alliance, white paper #3," 2009.

[6] A. Dunkels and J. Vasseur, "IP for smart objects, internet protocol for smart objects (IPSO) alliance, white paper #1," 2008.

[7] M. Potkonjak, S. Meguerdichian, and J. L. Wong, "Trusted sensors and remote sensing," in *IEEE Sensors*, pp. 1104–1107, 2010.

[8] J. H. Kong, L.-M. Ang, and K. P. Seng, "Minimalist security and privacy schemes based on enhanced AES for integrated WISP sensor networks," *Journal of Communication Networks and Distributed Systems*, vol. 11, no. 2, pp. 214–232, 2013.

[9] A. M. Dunn *et al.*, "Eternal sunshine of the spotless machine: Protecting privacy with ephemeral channels.," in *Operating Systems Design and Implementation (OSDI)*, pp. 61–75, 2012.

[10] Y. Tang *et al.*, "CleanOS: Limiting mobile data exposure with idle eviction.," in *Operating Systems Design and Implementation (OSDI)*, vol. 12, pp. 77–91, 2012.

[11] Z. N. Peterson, R. C. Burns, J. Herring, A. Stubblefield, and A. D. Rubin, "Secure deletion for a versioning file system.," in *File and Storage Technologies (FAST)*, vol. 5, pp. 4–11, 2005.

[12] D. Boneh and R. J. Lipton, "A revocable backup system.," in *USENIX Security*, pp. 91–96, 1996.

[13] S. Diesburg *et al.*, "TrueErase: Per-file secure deletion for the storage data path," in *Anual Computer Security Applications Conference (ACSAC)*, pp. 439–448, 2012.

[14] J. R. Smith *et al.*, "RFID-based techniques for human-activity detection," *Communications of the ACM*, vol. 48, no. 9, pp. 39–44, 2005.

[15] K. Rowe, "Securing microcontroller RTOSes for the internet of things." http://www.embedded.com/design/operating-systems/4429868/Securing-microcontroller-RTOSes-for-the-Internet-of-Things, 2014.

[16] N. Beckmann and M. Potkonjak, "Hardware-based public-key cryptography with public physically unclonable functions," in *Information Hiding*, pp. 206–220, 2009.

[17] M. Potkonjak, S. Meguerdichian, A. Nahapetian, and S. Wei, "Differential public physically unclonable functions: architecture and applications," in *Design Automation Conference (DAC)*, pp. 242–247, 2011.

[18] M. A. Alam and S. Mahapatra, "A comprehensive model of PMOS NBTI degradation," *Microelectronics Reliability*, vol. 45, no. 1, pp. 71–81, 2005.

[19] S. Meguerdichian and M. Potkonjak, "Matched public PUF: ultra low energy security platform," in *International Symposium on Low Power Electronics and Design (ISLPED)*, pp. 45–50, 2011.

[20] S. Meguerdichian and M. Potkonjak, "Using standardized quantization for multi-party PPUF matching: Foundations and applications," in *International Conference on Computer-Aided Design (ICCAD)*, pp. 577–584, 2012.

[21] J. B. Wendt and M. Potkonjak, "Nanotechnology-based trusted remote sensing," in *IEEE Sensors*, pp. 1213–1216, 2011.

[22] J. B. Wendt and M. Potkonjak, "The bidirectional polyomino partitioned PPUF as a hardware security primitive," in *Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 257–260, 2013.

[23] J. B. Wendt and M. Potkonjak, "Hardware obfuscation using PUF-based logic," in *International Conference on Computer-Aided Design (ICCAD)*, pp. 1–8, 2014.

[24] T. Xu, J. B. Wendt, and M. Potkonjak, "Secure remote sensing and communication using digital PUFs," in *Symposium on Architectures for Networking and Communications Systems (ANCS)*, pp. 1–12, 2014.

[25] T. Xu and M. Potkonjak, "Robust and flexible FPGA-based digital PUF," in *Field Programmable Logic and Applications*, pp. 1–6, 2014.

[26] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical one-way functions," *Science*, vol. 297, no. 5589, pp. 2026–2030, 2002.

[27] T. Xu, J. B. Wendt, and M. Potkonjak, "Digital bimodal function: an ultra-low energy security primitive," in *International Symposium on Low Power Electronics and Design (ISLPED)*, pp. 292–296, 2013.

[28] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, and E. Barker, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," tech. rep., DTIC Document, 2001.