A Secure Low-Cost Edge Device Authentication Scheme for the Internet of Things

Ujjwal Guin*, Adit Singh*, Mahabubul Alam*, Janice Cañedo†, and Anthony Skjellum†
*Department of Electrical and Computer Engineering
†Department of Computer Science and Software Engineering
Auburn University, AL, USA
{ujjwal.guin, singhad, mahabubul.alam, canedo, skjellum}@auburn.edu

***Abstract*—Because of the enhanced capability of adversaries, edge devices of Internet of Things (IoT) infrastructure are now increasingly vulnerable to counterfeiting and piracy. Ensuring the authenticity of such devices is of great concern since an adversary can create a backdoor either to bypass the security, and/or to leak secret information over an unsecured communication channel. The reliability of such devices could also be called into question because they might be counterfeit, defective and/or of inferior quality. It is of prime importance to design and develop solutions for authenticating such edge devices. In this paper, we present a novel low-cost solution for authenticating edge devices. We use SRAM based PUF to generate unique "digital fingerprints" for every device, which can be used to generate a unique device ID. We propose a novel ID matching scheme to verify the identity of an edge device even though the PUF is extremely unreliable. We show that the probability of impersonating an ID by an adversary is extremely low. In addition, our proposed solution is resistant to various known attacks.**

***Index Terms*—Internet of Things (IoT), Physically Unclonable Functions (PUF), Edge Device, Authentication.**

I. Introduction

Being identified as the biggest business prospect for the next decade by International Technology Roadmap for Semiconductors (ITRS), the Internet of Things (IoT) offers huge potential for business and scientific development in the field of semiconductor design in the post Moore era [1]. IoT is an infrastructure in which billions of devices ("things") are connected to the Internet to enable direct interactions between the physical world and computer-based systems. We denote these things as *edge devices*, which are constituted from a wide variety of electronic and electromechanical devices such as smart thermostats, lights, watches, mobile phones, sensors, actuators, and many others. Recently, different reports indicate that the number of connected devices will well exceed several billions by 2020 [2], [3]. As ubiquitous sensing is the backbone of any IoT application, edge devices are distributed among large geographical areas and generate real time data for further analysis and decision making. The pervasive nature of these applications has resulted in severe resource constraints in these edge devices, such as, low power budget, low die area allocation, and low processing power.

Quite a few architectures have been proposed for IoT applications and almost all of them have similar communication scenario - edge devices connecting with server/applications through gateway devices [4]–[6]. Figure 1 shows a sample IoT system architecture. According to Cisco IoT framework, the

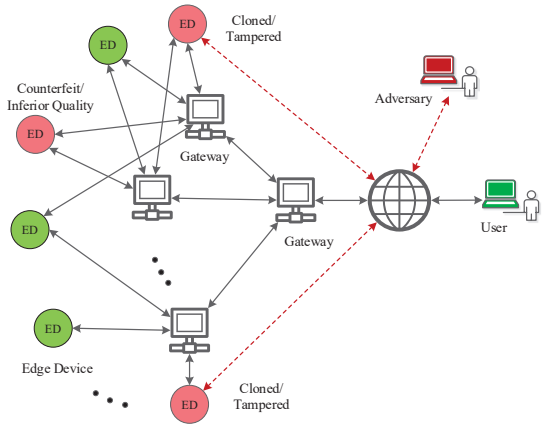

Figure 1: A standard IoT model with hardware vulnerabilities.

edge devices in IoT should have the capability of generating data, converting analog data to digital format and they should also have the ability of being queried/controlled over the network [4]. To accommodate these functions, the edge devices should be equipped with sensors, analog to digital converters, communication modules, memory and embedded processors. Importantly, edge devices in IoT applications are expected to be low-cost devices in order to facilitate widespread adoption [7]. Remote applications also limit the energy resource. Equipping edge devices with large batteries is not feasible for most applications because of the cost constraints [8]. So, minimizing power consumption in every possible ways has become a prime design concern. Energy constraint prohibits IoT edge devices to use standard cryptographic schemes [8] which has resulted in the development of several light-weight cryptographic schemes [9]. Still, a majority of the IoT devices do not use encryption due to power constraints. In a recent study, HP has found almost 70% of the tested IoT devices did not use encryption during communication with the server over the Internet [10].

Moreover, most of these devices are manufactured in environments of limited trust that in particular lack relevant government or other appropriate oversight, and then travel across the globe through intermediaries in the supply chain before being deployed. These factors make it virtually impossible to gauge the origin of these systems and their components, and to track their route in the supply chain. As a result, it is becoming increasingly difficult to ensure the security, integrity, and authenticity of these edge devices. Numerous

incidents have highlighted the far-reaching penetration of such counterfeit devices into the electronics supply chain [11]–[13], including cloned systems into the United States defense supply chain [14]–[18]. Figure 1 shows an abstract view of an IoT System with various attacks originating from untrusted hardware. Hardware attacks are generally initiated at the physical location where the system is located. For instance, a rogue employee can hypothetically replace an authentic device with a counterfeit or cloned device in order to gain access to the system improperly. A user can also unknowingly add a cloned device to the system since it is nearly impossible to track the origin of a given device in today's convoluted and generally untrustworthy supply chains. In addition, various software attacks can be performed through networks such as Phishing, Denial of Service (DoS), and data spoofing [19], [20]. The scope of this paper is to address the hardware attacks by ensuring the authenticity of the edge devices.

In this paper, we have presented a lightweight communication protocol to verify the authenticity of a low-cost edge device by verifying an unclonable device ID. The communication protocol is designed to be cost effective as it can use on-board resources available in even a minimal IoT edge device, such as, a processor and SRAM memories. Furthermore, the unclonable device ID can be generated from an on-board SRAM memory (SRAM PUF [21]) to avoid the cost of programmable non-volatile memory in low cost edge devices. Recently, physically unclonable functions (PUFs) have received much attention in the hardware security community because they can generate unique and unclonable bits for the identification and authentication of ICs. PUFs use inherently uncontrollable and unpredictable variations from a manufacturing process to produce random and unclonable bits. Several PUF architectures have been proposed over the years that include arbiter PUF [22], ring oscillator PUF [23], SRAM PUF [21], among others. As the IoT edge devices have SRAM based memory and embedded processor, SRAM PUFs offer a better solution to produce device IDs with no additional cost. However, SRAM PUFs are often found to be unreliable which makes authentication through exact ID matching a difficult task [24]. So, in this paper, we have further proposed a novel repeated ID matching scheme which can be used to authenticate edge devices with unreliable IDs. The contributions of our paper are as follows:

- *Secure communication protocol:* We propose a new lightweight communication protocol that uses existing hardware resources of an IoT edge device to verify its identify. Our proposed solution uses a secure hash function [25], which can be implemented using the embedded processor and memory of the edge devices [26], [27]. Care needs to be taken such that the device ID never leaves the system without encryption. Our heuristic security evaluation shows that this protocol is resistant to various known attacks (see Section IV for details).
- *Novel device ID verification technique:* Due to the unreliable nature of SRAM PUF responses, we propose a repeated ID matching scheme (see details in Section III), which does not require expensive helper data and algorithms for error correction, currently in practice for SRAM PUFs. Our proposed repeated ID matching scheme extracts the most reliable bits from the responses of an unreliable PUF. We show that it is extremely unlikely (see details in Table I) to impersonate an edge device by an adversary. With random trials, an adversary can pass simple ID matching scheme (see Section III-A) when a PUF is unreliable. However, this is unfeasible to impersonate an ID two times by an adversary using random guesses. Note that one can also implement an authentication scheme that verifies ID more than two times to further increase the difficulty of impersonating an authentic edge device.

The rest of the paper is organized as follows. Section II describes our proposed lightweight communication protocol for authenticating an edge device. We present our proposed ID matching scheme by repeated authentication in Section III. We perform a security evaluation in Section IV. We conclude the paper in Section V.

II. Proposed Authentication Scheme

Limited device resource has become one of the major constraints for implementing standard secure protocols for authenticating an edge device. Low die area, low power, and limited memory requirements severely limit the performance of the majority of edge devices. Trappe et al. showed that these constraints prohibit the low end IoT devices from using any standard secure protocols like TLS or IPsec [8]. It has resulted in the development of IoT devices with minimum security considering the generated information is of little value to the attackers. Recent studies conducted by HP, Symantec have found major portion of the IoT devices not to use any kind of encryption during communication over the Internet at all [10], [28]. But these seemingly harmless information can be used to get into any complex systems which has been showed in recent studies [28], [29]. Counterfeit or cloned devices in an IoT environment can cause significant damage to the security of the system which raises the need for edge device authentication. In this paper, we address the verification of an edge device by correctly identifying its origin. However, we agree that the data encryption, which is only practical for the gateways as they have higher resources, is required in order to prevent various run time attacks. Our paper does not address the identity verification of gateways, and treats it as authentic.

Figure 2 shows our proposed authentication approach. A true random number generator (TRNG) is necessary in the gateway device for generating a random nonce (n). We propose to use an area-efficient cryptographically secure pseudo-random number generator (CSPRNG) depending on the implementation choice [30] or [31]. A one-time-pad (OTP) [32], [33] is used for encrypting the key with n. OTP has extremely low area footprint since it only requires a simple XOR network. We require an on-chip memory to store secret key-ID pair, $\{K_i, ID_i\}$, corresponding to the edge device (ED_i) of an IoT system. In the edge device, an SRAM PUF can be used to generate this secret unclonable device ID, ID_i (see Section III for details).

The communication protocol for authenticating an edge device is described as follows:

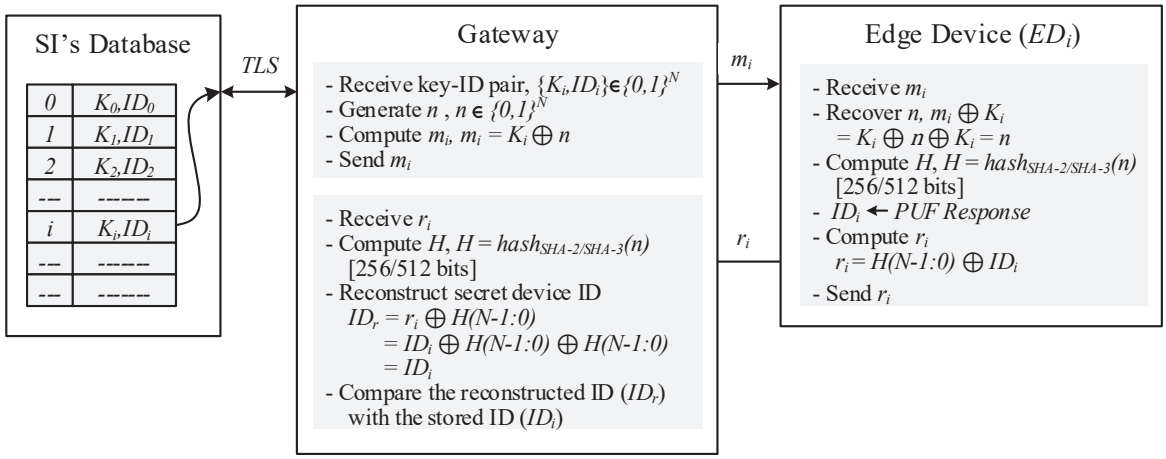

Figure 2: Proposed communication protocol for verifying the identity of an edge device.

1) The gateway uses an existing secure communication protocol (*e.g.*, TLS [34]) to receive the secret device key-ID pair $\{K_i$,$ID_i\}$, from the trusted system integrator (SI) who produced that edge device. During production, every device is registered in a secure database with a public ID, and a key-ID pair, $\{ID_i \in \{0,1\}^N,\ K_i \in \{0,1\}^N\}$. Here, N represents the size of the ID and the key. For example, it can be 128 depending on the level of security one requires. Now, the public ID is necessary to identify the edge device in the database. It is also possible to store the data in a tamper-proof NVM of the gateways. However, since the gateway is always connected to the Internet and the standard security measures in place as it does not have any resource constraints, we recommend receiving the key-ID pair from the trusted integrator rather than storing into NVM.
2) The gateway stores $\{K_i, ID_i\}$ in its on-chip (volatile or non-volatile) memory. We need to make sure that one cannot access this $\{K_i, ID_i\}$ through the input/output (I/O) pins of the gateway. This will prevent an adversary from getting access to the $\{K_i, ID_i\}$, which was generated during the registration phase of the edge device.
3) An on-chip CSPRNG generates a unique nonce (n), which is stored in the memory. This n will be used later for decrypting the secret device ID. A one-time pad (OTP) now encrypts the key (K_i) with this random nonce. The gateway then sends this encrypted key ($n \oplus K_i$) (depicted as (m_i) in the figure) to the i^{th}edge device, ED_i, to request for its identification.
4) The unique nonce (n) is recovered at the edge device by XORing the m_i with the shared secret key (K_i). A crypographically secure hash (e.g., SHA-2 or SHA-3 [25]) is computed on this nonce (n) to produce a 256/512 bits hash output (H). We recommend to use the existing hardware resources (embedded processor and memory [26], [27]) of the edge devices to compute the hash.

$$H = hash_{SHA-2/SHA-3}(n)$$

5) The edge device encrypts the device ID, ID_i using N bits of computed H.

$$r_i = ID_i \oplus H(N-1:0)$$

The encrypted data $\{r_i\}$ is sent to the gateway.
6) After receiving r_i, the gateway computes the same hash (SHA-2 or SHA-3) using the stored random nonce, n. The gateway now reconstructs the secret device ID.

$$\begin{aligned} & r_i \oplus H(N-1:0) \\ & = ID_i \oplus H(N-1:0) \oplus H(N-1:0) \\ & = ID_i \end{aligned} \quad (1)$$

7) This ID is then verified with the stored ID (see Section III for details). Steps 3-6 are followed for the second stage of the authentication to increase the confidence that the ID is originated from an authentic edge device.

III. Device Authentication by ID Matching

In order to secure the semiconductor supply chain and prevent intrusion of counterfeit ICs in critical IoT applications, edge devices need to be equipped with unclonable IDs. In our proposed authentication scheme, we have considered an SRAM PUF for producing an unclonable device ID considering the IoT edge devices are expected to incorporate SRAM-based memories, which makes SRAM PUF a low cost solution for developing such IDs. However, reliability is a major concern for most PUF implementations today [21], [23], [35]. The response of a PUF must remain stable over a wide range of environmental variations such as temperature, noise generated from power supply, and electromagnetic interference, as well as unpredictable degradation that arises from aging. Therefore, it is necessary to perform error correction (e.g., fuzzy extraction [36]) to produce stable PUF responses. Helper data are often required to improve the accuracy and efficiency. However, the implementation of expensive error correction is challenging in the resource constrained edge devices.

As SRAM PUF outputs are unreliable (without error correction), authenticating an edge device by comparing IDs is challenging. SRAM PUF outputs may vary because of aging degradation, temperature variation and supply voltage

fluctuations. So, if a PUF response is taken at two different environmental conditions, the responses can vary significantly. *How can we use a PUF to verify the identity of a device, if a PUF produces an unreliable ID?* There will be a mismatch among the stored response (obtained from PUFs during the registration) and the new response while authentication from the same PUF. The basic idea is to capture multiple responses from the PUF in a very short duration under almost identical environmental conditions to ensure the responses are similar. This led us to develop a repeated authentication scheme, where the gateway interrogates an edge device more than once in short duration (see details in Section III-B).

A. Simple ID Matching

The primary reason that we can overcome PUF uncertainty is that we do not need to match completely both stored and received IDs bit-by-bit to authenticate an edge device. We propose to use Hamming distance (HD) to measure the similarity between the stored ID (ID_S, PUF response during registration) and received ID (ID_R, PUF response during authentication). The Hamming distance between two IDs is defined as the number of bits at which the two IDs differ.

The authentication can be performed as follows:

$$HD(ID_S, ID_R) \rightarrow \begin{cases} \leq HD_T, & \text{The device is authentic} \\ > HD_T, & \text{The device is counterfeit} \end{cases}$$

where, HD_T is the threshold that can be determined after characterizing the PUF. For example, one can set a threshold of 10 for a 128-bit device ID. This signifies that a device under authentication is certified as authentic when the mismatch of bits between ID_S and ID_R does not exceed 10.

Let us now determine the probability of authenticating an edge device as genuine by an adversary. We assume that the size of the stored ID (PUF response) is of N-bits and HD_T is of k-bits. Now, the probability of finding one vector with exactly $(N-k)$-bit match or k-bit mismatch is as follows:

$$p = \frac{{}^{N}C_{(N-k)}}{2^N} = \frac{\binom{N}{k}}{2^N}.$$

Thus, the probability of finding a vector with at most k-bit mismatch to pass the ID matching test becomes:

$$p = \frac{\sum_{i=0}^{k} \binom{N}{i}}{2^N}. \tag{2}$$

From Equation 2, it is clear (see Table I for details) that the attacker has a low probability of impersonating a device. However, this is not sufficient enough to prevent an adversary guessing an ID by random trials when there are many unreliable bits in an ID. This may occur when an edge device uses an SRAM PUF without any error correction. It is thus necessary to further improve simple ID matching scheme.

B. Repeated ID Matching

The reliability issue of the PUF primarily arises from the environmental variation, such as temperature variation and device aging. The environmental conditions and device age during registration can be significantly different when an edge device is requested for authentication. This results in bit flipping, where the values of a few bits are different between registration and subsequent authentication.

$$\{ID_R[i]\}_{T_R \neq T_S, t_R > t_S} \neq \{ID_S[i]\}_{T_S, t_S}$$

where, ID_R and ID_S are the received and stored PUF responses, respectively. T_R and T_S denote the temperature during the authentication and registration. Similarly, t_R and t_S are the device age during the authentication and registration, respectively.

The primary reason for obtaining different responses is that the PUF is challenged with different conditions. We should have received the same responses if we were to apply the same challenge with similar conditions. *The basic idea of extracting the stable bits from an unreliable PUF is to provide the same challenge repeatedly in short duration such that we can extract PUF responses at similar conditions.* The repeated authentication scheme utilizes this property of a PUF to verify identity of an edge device.

The repeated authentication scheme is described below:

1) The gateway device requests an edge device (ED_i) for its device ID by sending $n_1 \oplus K_i$. The edge device returns encrypted ID_R ($H_{n_1}(N-1:0) \oplus ID_R$). The gateway first decrypts the ID (see Equation 1), and then computes the mismatch locations of the received ID ($ID_R[i] \neq ID_S[i]$). It then contracts the robust ID (RID) by discarding the mismatch bits and keeps track of the mismatch locations.
$$RID[k] = ID_R[i], \text{ if } ID_R[i] = ID_S[i]; \quad 0 < k < i < N \tag{3}$$
2) The gateway again requests ED_i for its ID by sending $n_2 \oplus K_i$ ($n_1 \neq n_2$). ED_i then returns the encrypted ID_R ($H_{n_2}(N-1:0) \oplus ID_R$). The gateway decrypts the ID by using Equation 1, and then computes the new robust ID (RID^*) by using 3.
3) The two robust IDs are compared by using Hamming distance, which is described below:
$$HD(RID, RID^*) \rightarrow \begin{cases} \leq HD_T^{\dagger}, & \text{The device is authentic} \\ > HD_T^{\dagger}, & \text{The device is counterfeit} \end{cases}$$
Note that $HD_T^{\dagger}$ is much less than HD_T because the PUF produces a similar response for similar conditions. It is important to keep in mind that one can implement an authentication scheme that uses more than two repeated IDs from the same device.

The probability of authenticating a device two times as genuine by an adversary can be determined. We assume that the size of the stored ID is of N-bits, HD_T is of k-bits and that $HD_T^{\dagger}$ is of r-bits ($r \ll k$). The probability of passing two repeated authentication becomes:

$$p = \left(\frac{\sum_{i=0}^{k} \binom{N}{i}}{2^N} \right) \times \left(\frac{\sum_{i=0}^{r} \binom{N-k}{i}}{2^{(N-k)}} \right) \tag{4}$$

Equation 4 provides an interesting observation that the probability of successful authentication is reduced significantly

Table I: Probability of matching an ID.

	Simple ID Matching		Repeated ID Matching									
			$HD_T^\dagger$ =1		$HD_T^\dagger$=2		$HD_T^\dagger$=4		$HD_T^\dagger$=8		$HD_T^\dagger$=16	
HD_T	ID =128	ID = 256	ID =128	ID = 256	ID =128	ID = 256	ID =128	ID = 256	ID =128	ID = 256	ID =128	ID = 256
1	3.8e-037	2.2e-075	2.9e-073	9.8e-150	1.8e-071	1.3e-147	2.4e-068	6.7e-144	3.2e-063	1.6e-137	2.1e-055	3.9e-127
2	2.4e-035	2.8e-073	3.6e-071	2.5e-147	2.3e-069	3.2e-145	3.0e-066	1.7e-141	3.8e-061	3.9e-135	2.4e-053	9.3e-125
4	3.2e-032	1.5e-069	1.9e-067	5.4e-143	1.2e-065	6.8e-141	1.5e-062	3.5e-137	1.8e-057	7.9e-131	9.7e-050	1.8e-120
8	4.5e-027	3.7e-063	4.1e-061	2.0e-135	2.5e-059	2.5e-133	2.9e-056	1.3e-129	3.1e-051	2.6e-123	1.2e-043	5.2e-113
16	3.2e-019	9.3e-053	6.9e-051	1.3e-122	3.9e-049	1.5e-120	4.0e-046	7.2e-117	3.2e-041	1.3e-110	7.0e-034	2.0e-100
32	6.4e-009	5.9e-037	7.9e-036	4.9e-102	3.8e-034	5.5e-100	2.8e-031	2.3e-096	1.2e-026	3.1e-090	6.7e-020	2.6e-080
64	5.4e-001	2.4e-016	1.9e-018	7.5e-072	6.0e-017	7.2e-070	2.0e-014	2.2e-066	1.5e-010	1.6e-060	2.1e-005	3.7e-051

if we perform repeated authentication. We can also prevent repeated failed authentication at the gateway by implementing a simple counter to keep track of that situation. If the count crosses some threshold, that can raise a flag.

IV. Analysis

A. Replay Attack on Our Proposed Authentication Scheme

In this attack scenario, an attacker attempts to authenticate an edge device by impersonating the ID using prior communications. We assume that an attacker does not have access to the secret key (K_i), which is stored in the edge device. Let us assume that the attacker observes two prior communications. First, he/she observes $n_1 \oplus K_i$ from the gateway and $H_{n_1}(N-1:0) \oplus ID_i$ from the edge device. From this observation the attacker can compute $K_i \oplus ID_i \oplus n_1 \oplus H_{n_1}(N-1:0)$, which is shown below:

$$(n_1 \oplus K_i) \oplus (H_{n_1}(N-1:0) \oplus ID_i) \tag{5}$$

From the second communication, the attacker observes $n_2(\neq n_1) \oplus K_i$ from the gateway and $H_{n_2}(N-1:0) \oplus ID_i$ from the edge device, and can compute $K_i \oplus ID_i \oplus n_2 \oplus H_{n_2}(N-1:0)$.

Now the attacker can perform the following operations:

$$(n_1 \oplus K_i) \oplus (n_2 \oplus K_i) = n_1 \oplus n_2 \tag{6}$$

$$\begin{aligned}(H_{n_1}(N-1:0) \oplus ID_i) \oplus (H_{n_2}(N-1:0) \oplus ID_i) \\ = H_{n_1}(N-1:0) \oplus H_{n_2}(N-1:0)\end{aligned} \tag{7}$$

From equation 7, it is obvious that an adversary successfully replays a prior communication if it becomes zero, when $H_{n_1}(N-1:0) = H_{n_2}(N-1:0)$. This contradicts the collision property of a secure hash [25]. Thus, the communication protocol becomes resistant to replay attack, when a system designer implements a secure hash function (SHA-2 or SHA-3).

B. Probability Analysis for Proper ID Matching

Hamming distance plays an important role to reduce the chances of guessing an ID in the allowable ID spaces by an adversary. It becomes harder for guessing an ID by random trials, when the hamming distance is small. However, the choice of hamming distance largely depends on the reliability of the PUF. For a reliable PUF, simple ID matching is sufficient enough to provide enough protection against cloning. However, we can achieve significant improvement from our proposed repeated ID matching scheme even for a very unreliable PUF, which can be the case for a resource constrained IoT edge device. In this section, we will analyze two different (reliable and unreliable) types of PUFs to evaluate the effectiveness of our ID matching schemes.

The Table I summarizes the probabilities of matching edge device IDs under simple ID matching scheme (see Section III-A) and proposed repeated ID matching scheme (see Section III-B). We perform our analysis considering 128-bit and 256-bit device IDs. The hamming distances (HD_Ts) chosen for the analysis are 1, 2, 4, 8, 16, 32, and 64. The PUFs are very reliable when HD_T are of 1, 2 and 4. On the other hand, we also consider very noisy PUFs where 32 or 64 bits may be flipped during authentication. As expected, the probabilities of ID matching increases with the increase of HD_T. This is intuitive as an adversary has lesser effort of matching an ID. For a reliable PUF, an attacker has an extremely low luck of matching an authentic ID. For example, the probability of finding a match becomes 4.5×10^{-27} and 3.7×10^{-63} considering HD_T of 8, when the IDs are 128 bit and 256 bit respectively. However, the probability increases significantly 5.4×10^{-1} and 2.5×10^{-16} considering HD_T of 64 for an ID of 128 and 256 bits, respectively.

For repeated ID matching scheme, the hamming distances ($HD_T^\dagger s$), chosen for the second stage of the authentication process, are of 1, 2, 4, 8, and 16 bits. As before, the probabilities of ID matching becomes higher with the increase of $HD_T^\dagger s$. However, it is much less compared to the simple ID matching scheme. We now have a significant improvement of not finding an ID as the probability has decreased significantly. For example, the probability of finding an ID is 1.9×10^{-18} for a PUF that is heavily impacted by aging (64 out of 128 bits are unstable) assuming the stable bits remains stable ($HD_T^\dagger$ is of 1). For a 128 bit ID, we also have a very low probability of 1.5×10^{-10} and 2.1×10^{-5} with $HD_T^\dagger$ of 8 and 16 bits, respectively. For a 256 bit ID, it is fairly impossible for an adversary to pass the repeated ID matching scheme even though PUF responses are unreliable.

C. Physical Attacks

The security of our proposed scheme largely depends on the shared secret key between the gateway and the edge devices which should be stored in tamper-proof memory and the SRAM PUF based device IDs for edge devices. These information can be stolen through sophisticated physical attacks or reverse engineering. Today's optical microscopes can produce 3D images of a microchip with superfine resolution. Scanning Electron Microscopes (SEM) and Transmission Electron Microscopes (TEM) can generate images of different inner layers of a microchip. Chipworks (now TechInsights) have successfully performed such experiments legitimately for the purpose of competitive analysis and patent research.

The physical layout of a chip can be reconstructed through destructive physical attacks as well. Data stored in a non-volatile memory (NVM) can be reconstructed through infrared backside imaging, which can be used to directly look at the memory contents. All these physical attacks can definitely be used to find the secret key or the device ID. However, an adversary can impersonate only one device through such physical attacks, which does not make any financial motivation for performing such attacks.

V. Conclusion

In this paper, we have presented a novel communication protocol to authenticate edge devices for an IoT system by using unclonable device IDs. As resource limitation prohibits standard cryptographic schemes to be followed in the edge devices, we have presented a light-weight encryption scheme that uses a secure hash function and can be implemented with existing resources of the IoT edge devices. The unclonable device ID can be created by using an on-board SRAM PUF, as it produces a unique device footprint based on the manufacturing variability. The unpredictable aging degradation and temperature variation make the output of the PUFs often unreliable. Thus, developing a verification scheme, which takes care of an ID generated from an unreliable PUF, is of prime importance. We address this reliability issues of PUFs by introducing repeated authentication. We have utilized the concept of hamming distance for our ID matching scheme. It is extremely difficult for an adversary to clone an ID to pass repeated ID matching scheme which is evident from our probability analysis. The larger device ID also provides higher protection against cloning since it poses increased difficulty for an attacker to pass ID matching test.

ACKNOWLEDGEMENT

This work was supported in part by an internal research grant from the Charles D. McCrary Institute to the first author. Additional support from the authors' respective departments and the Auburn Cyber Research Center is also acknowledged.

References

[1] International Technology Roadmap for Semiconductors 2.0, 2015 Edition.

[2] G. Research, "Gartner says 6.4 billion connected "things" will be in use in 2016, up 30 percent from 2015," 2015, http://www.gartner.com/newsroom/id/3165317.

[3] D. Evans, "The Internet of Things: How the Next Evolution of the Internet Is Changing Everything," 2011.

[4] Cisco, "The internet of things reference model," 2014.

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

[6] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future generation computer systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[7] J. Bryzek, "Roadmap for the trillion sensor universe," *Berkeley, CA, April*, vol. 2, 2013.

[8] W. Trappe, R. Howard, and R. S. Moore, "Low-energy security: Limits and opportunities in the internet of things," *IEEE Security & Privacy*, vol. 13, no. 1, pp. 14–21, 2015.

[9] M. Katagi and S. Moriai, "Lightweight cryptography for the internet of things," *Sony Corporation*, pp. 7–10, 2008.

[10] K. Rawlinson. Hp study reveals 70 percent of internet of things devices vulnerable to attack. [Online]. Available: http://www8.hp.com/us/en/hp-news/press-release.html?id=1744676.WUrrwWgrKM8

[11] M. M. Tehranipoor, U. Guin, and D. Forte, *Counterfeit Integrated Circuits: Detection and Avoidance*. Springer, 2015.

[12] U. Guin, K. Huang, D. DiMase, J. Carulli, M. Tehranipoor, and Y. Makris, "Counterfeit integrated circuits: A rising threat in the global semiconductor supply chain," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1207–1228, Aug 2014.

[13] U. Guin, D. DiMase, and M. Tehranipoor, "Counterfeit integrated circuits: Detection, avoidance, and the challenges ahead," *Journal of Electronic Testing*, vol. 30, no. 1, pp. 9–23, 2014.

[14] "Texas brothers plead guilty to selling counterfeit 'Cisco' computer products," News Releases, U.S. Immigration and Customs Enforcement, August 2009.

[15] The Federal Bureau of Investigation, "Departments of Justice and Homeland Security Announce 30 Convictions, More Than $143 Million in Seizures from Initiative Targeting Traffickers in Counterfeit Network Hardware ," May 2010.

[16] U. Guin, S. Bhunia, D. Forte, and M. Tehranipoor, "Sma: A system-level mutual authentication for protecting electronic hardware and firmware," *IEEE Transactions on Dependable and Secure Computing*, 2016.

[17] U. Guin, Q. Shi, D. Forte, and M. Tehranipoor, "FORTIS: A Comprehensive Solution for Establishing Forward Trust for Protecting IPs and ICs," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 2016.

[18] M. M. Tehranipoor, U. Guin, and S. Bhunia, "Invasion of the hardware snatchers," *IEEE Spectrum*, vol. 54, no. 5, pp. 36–41, 2017.

[19] T. Borgohain, U. Kumar, and S. Sanyal, "Survey of security and privacy issues of internet of things," *CoRR*, vol. abs/1501.02211, 2015. [Online]. Available: http://arxiv.org/abs/1501.02211

[20] H. He, C. Maple, T. Watson, A. Tiwari, J. Mehnen, Y. Jin, and B. Gabrys, "The security challenges in the iot enabled cyber-physical systems and opportunities for evolutionary computing other computational intelligence," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, July 2016, pp. 1015–1021.

[21] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, "Fpga intrinsic pufs and their use for ip protection," in *International workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2007, pp. 63–80.

[22] B. Gassend, D. Clarke, M. Van Dijk, and S. Devadas, "Silicon physical random functions," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*. ACM, 2002, pp. 148–160.

[23] G. Suh and S. Devadas, "Physical Unclonable Functions for device authentication and secret key generation," in *Proc. of ACM/IEEE on Design Automation Conference*, 2007, pp. 9–14.

[24] K. Xiao, M. T. Rahman, D. Forte, Y. Huang, M. Su, and M. Tehranipoor, "Bit selection algorithm suitable for high-volume production of sram-puf," in *Hardware-Oriented Security and Trust (HOST), 2014 IEEE International Symposium on*. IEEE, 2014, pp. 101–106.

[25] NIST, "FIPS PUB 180-4: Secure Hash Standard (SHS)," August 2015.

[26] T. Eisenbarth, S. Heyse, I. von Maurich, T. Poeppelmann, J. Rave, C. Reuber, and A. Wild, "Evaluation of sha-3 candidates for 8-bit embedded processors," in *The Second SHA-3 Candidate Conference*, 2010.

[27] "Atmel AVR232: Authentication Using SHA-256," Tech. Rep. [Online]. Available: http://www.atmel.com/Images/doc8184.pdf

[28] M. B. Barcena and C. Wueest, "Insecurity in the internet of things."

[29] R. M. Ishtiaq Roufa, H. Mustafaa, S. O. Travis Taylora, W. Xua, M. Gruteserb, W. Trappeb, and I. Seskarb, "Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study," in *19th USENIX Security Symposium, Washington DC*, 2010, pp. 11–13.

[30] D. E. Holcomb, W. P. Burleson, and K. Fu, "Initial sram state as a fingerprint and source of true random numbers for rfid tags," in *In Proceedings of the Conference on RFID Security*, 2007.

[31] B. Sunar, W. Martin, and D. Stinson, "A provably secure true random number generator with built-in tolerance to active attacks," *Computers, IEEE Transactions on*, vol. 56, no. 1, pp. 109–119, Jan 2007.

[32] G. S. Vernam, "Secret signaling system," 1919, US Patent 1,310,719.

[33] J. Katz and Y. Lindell, *Introduction to modern cryptography*. CRC press, 2014.

[34] T. Dierks, "The transport layer security (tls) protocol version 1.2," 2008.

[35] A. Maiti and P. Schaumont, "The impact of aging on a physical unclonable function," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 9, pp. 1854–1864, 2014.

[36] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *SIAM journal on computing*, vol. 38, no. 1, pp. 97–139, 2008.