Remote Attestation of IoT Devices using Physically Unclonable Functions: Recent Advancements and Open Research Challenges

Niccolò Marastoni
niccolo.marastoni@univr.it
University of Verona
Verona, Italy

Mariano Ceccato
mariano.ceccato@univr.it
University of Verona
Verona, Italy

ABSTRACT

In the past few years, the diffusion of IoT devices used in everyday life has skyrocketed. From wearable devices to smart home appliances, these gadgets are increasingly exposed to the Internet or to open networks. This means that it is necessary to find security solutions that can guarantee the safety of these devices, while at the same time saving on energy consumption and implementation space. In this paper we explore recent works that use remote attestation as a possible solution to the security of IoT devices while also focusing on the use of Physically Unclonable Functions (PUFs). We provide a thorough analysis of the selected papers, providing insights on possible future research directions.

CCS CONCEPTS

• **Security and privacy → Embedded systems security**.

KEYWORDS

IoT; security; remote attestation; PUF; SOK

ACM Reference Format:
Niccolò Marastoni and Mariano Ceccato. 2023. Remote Attestation of IoT Devices using Physically Unclonable Functions: Recent Advancements and Open Research Challenges . In *Proceedings of the 5th Workshop on CPSIoT Security and Privacy (CPSIoTSec '23), November 26, 2023, Copenhagen, Denmark.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3605758.3623502

1 INTRODUCTION

The rapid rise of the Internet of Things (IoT) has introduced numerous smart embedded devices into various domains, ranging from industrial control systems to everyday consumer products. These devices are often plagued by security and privacy challenges that are mainly due to their extensive resource constraints and the rush towards commercialization without adequate consideration for security implications [22]. This, combined with the constant connectivity of many modern IoT devices, and their tendency to inherit vulnerabilities from the previous generation of embedded systems [23], makes them an attractive target for malicious attackers.

One way to protect against these attackers is through remote attestation, which provides a mechanism to verify the identity and integrity status of remote computing devices without the need to rely on secure hardware, like Trusted Platform Modules (TPMs), which are often not well-suited for resource-constrained IoT devices [3]. This way, IoT systems can detect any unintended or malicious modifications to the firmware, software, or hardware running on the devices.

Attestation protocols still rely on some part of the system, called the root of trust, that is proven to be not compromised and from which the protocol can base its operations. Without being able to use the previously mentioned secure hardware modules (i.e. TPMs) as a root of trust, many modern protocols have turned to Physical Unclonable Functions (PUFs). These are relatively cheap hardware modules that leverage physical characteristics of the circuit to generate a unique fingerprint that is only bound to the device and cannot be cloned by design. The inherent simplicity of PUFs is one of their main draws, but it also offers a unique set of challenges when using them as root of trust. The rapid proliferation of IoT devices in recent times and the similar rise in the deployment of PUFs in such devices makes the particular subject of this paper very modern and requiring up-to-date information. This is the main reason why we limit our scope to the past 5 years.

In this survey, we explore some recent works in the field of attestation protocols for IoT (and adjacent) devices using PUFs, focusing on conference and journal papers published between 2018 and the first half of 2023.

The paper is structured as follows: Section 2 presents some related and recent surveys in the field and justifies the need for this paper. In Section 3 we detail the methodology used to select the 14 papers presented in this survey, while Section 4 introduces more in detail the main subjects treated in the following sections, such as attestation, PUFs etc. Section 5 represents the bulk of this work, summarizing each paper, identifying common approaches, attacks and typical evaluation methods, and Section 6 closes the paper with some more insight and the identified open research challenges.

2 RELATED WORK

In this section we discuss other recent surveys that explore the research space related to the security of IoT. While there are no other works that focus specifically on attestation and PUFs for IoT, we include some literature reviews that satisfy the research methodology introduced in Section 3.

Remote attestation is identified as a way to secure RISC-V devices in [9], but the few papers surveyed on the subject do not employ a PUF. Similarly, in [34] the PUFs are only used for authentication, without any attestation protocols involved. Another recent work that mentions both PUFs and attestation without considering their synergy is [29]. The work that most closely resembles ours is *A*

CPSIoTSec '23, November 26, 2023, Copenhagen, Denmark

ACM ISBN 979-8-4007-0254-9/23/11…$15.00
https://doi.org/10.1145/3605758.3623502

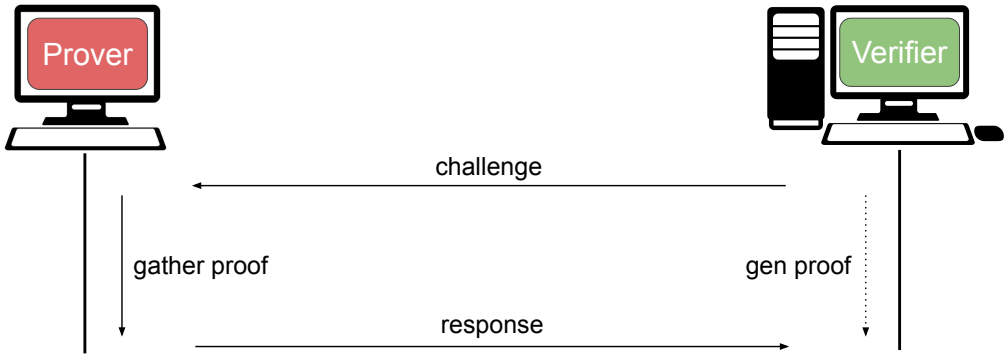

Figure 1: General workflow of an attestation protocol

survey of remote attestation in Internet of Things: Attacks, countermeasures, and prospects by Kuang et al. [24]. Their focus is not specifically on PUFs, nor on recent works so they only review 3 of the papers included in our survey. None of the mentioned surveys offer the in-depth categorization of attacks and evaluation techniques presented in this work.

3 SEARCH METHODOLOGY

Our research goal is to survey papers on attestation and PUFs in the IoT field published in the past 5 years (2018-2023), so we employed Google Scholar with various combinations of the terms "PUF", "remote attestation", "IoT" and "attestation" filtered by the publication year. The resulting 924 were then filtered to around 50 by removing all papers that: 1. did not appear in proper conferences or journals, mainly by excluding arxiv and various theses, 2. were not in English, 3. did not actually deal with either PUF, attestation or IoT devices, and 4. were duplicates. The final 14 papers selected in this survey have been selected by: 1. only keeping papers published in conferences with B rating or higher and journals with Q2 rating or higher, 2. a more thorough inspection of the paper contents to remove out of scope papers, and 3. a more lax approach at considering "IoT" as a criteria, which brought back a couple of papers that were initially removed that we then considered too interesting and related to leave out (e.g. [17]).

4 BACKGROUND

In this section we delve further into the concepts behind Physically Unclonable Functions (PUFs) and remote attestation.

4.1 PUFs

Physically Unclonable Functions (PUFs) are hardware devices that, given a challenge (or a set of challenges), can answer with a response (or a set of responses) that acts as a fingerprint. Due to small differences in the fabrication process, these fingerprints are assumed to uniquely identify the specific device that produces them. This hardware fingerprint also guarantees that PUFs are tamper-proof or at least tamper-evident, since most physical modifications of the device would incur in a change of the fingerprint.

PUFs can be classified into *weak* and *strong* PUFs, the former only providing a very small set of challenge and response pairs (CRPs) and the latter usually providing a very large set of CRPs.

4.2 Remote Attestation

Remote attestation protocols are well equipped to deal with the problem of having a remote resource (e.g. an IoT device) that needs to be secure but cannot be reliably kept secure by conventional means. This often happens because of the limited computational power of the device, its limited space (i.e. this would prevent the installation of a trusted secure hardware module) or its age and the inability of updating its obsolete hardware components.

The protocols generally require two entities: 1. a remote device that needs to prove that it is not compromised, called *Prover*, and 2. a secure machine that verifies the truthfulness of the Prover, called Verifier. A generic workflow can be seen in Fig. 1, where the Verifier initiates the protocol by issuing a challenge to the remote device, then the Prover needs to gather enough information about its current state to send back as the response. While this is happening, the Verifier can generate the expected response (or it could fetch it from a database) in order to check its correctness. If the response sent by the Prover corresponds to what was expected by the Verifier, then the attestation protocol can be declared a success, otherwise it either fails or it restarts with a fresh challenge.

The actual protocols in literature can vary a lot in each step of the presented workflow. The challenge can be a simple nonce or it can contain different types of information. Sometimes it can contains the entire attestation program aswell [37]. What usually varies the most from protocol to protocol is the information gathered by the Prover in order to convince of its uncompromised nature, and the methods with which this information is gathered.

As an example, the memory contents of the Prover is a common piece of data that can be sent to the Verifier to guarantee the absence of malicious adversaries. It is possible to send the checksum or hash of the entire memory dump (for very small devices) [42], which can attest the whole device, or just certain memory regions, which will in turn attest a specific software running on the device.

Root of Trust. The root of trust is a fundamental concept in remote (and normal) attestation protocols, as it defines a part of the

protocol (or of the Prover itself) that is secure by design. The protocols then build upon this root of trust in order to failsafe their procedure and guarantee that the attestation results themselves have not been tampered with.

The root of trust is defined differently across protocols, but one thing that guides the choice of a root of trust is the type of attestation algorithm. These can generally be divided into three categories: hardware based, software based or hybrid.

Hardware Based. If the attestation protocol is hardware based, then there needs to be a tamper-resistant hardware component (e.g. a TPM or a PUF) used to provide a secure environment where the attestation protocol can execute. In this scenario, the secure hardware component is usually the root of trust.

Software Based. In software based attestation protocols, the root of trust is provided by some characteristic of the algorithm itself. There are three main ways to guarantee a root of trust: 1. timing measures, 2. virtualization and 3. filling the memory with random stochastic noise. In the most common scenario, attestation protocols are designed to execute under a certain time threshold. If the procedure exceeds the threshold, then the verifier assumes that the prover is performing extra operations to circumvent the attestation, making it compromised. Virtualization approaches simply execute the code in a controlled environment, while memory filling techniques use deterministic stochastic noise to saturate the memory, so that the adversary cannot hide itself in the free space.

Hybrid. Hybrid approaches use a root of trust that involves both a hardware component and a software component. These are used when the prover devices are already deployed and thus cannot be equipped with the necessary hardware to implement a full hardware-based attestation protocol but need to rely on a hardware component that is already part of the device. This can make PUFs attractive for this scenario.

5 SURVEY RESULT

In this section we explore the 14 papers that were selected for this survey, highlighting their position in the research space by noting if and how they deal with some arguably important aspects of this research field. We have thus summarized how the selected papers can be categorized according to what type of PUF is used and how it is used, the definition of the root of trust, the attacks that are mentioned or the adversary model, and various types of evaluation, from security to computational complexity.

Table 1 intuitively depicts which papers address the specific problems outlined in this review, categorizing them with the amount of detail in which they have been treated ([L]ow, [M]edium, [H]igh or not at all).

5.1 Brief Description of the Papers

Secure Boot and Remote Attestation in the Sanctum Processor. [25] This paper proposes an attested execution processor that derives its cryptographic identity from manufacturing variation measured by a PUF, eliminating the need for non-volatile memory or explicitly assigned private keys. A trapdoor computational fuzzy extractor ensures the reliability and security of PUF keys. Detailed evaluation results are provided for both secure boot and remote attestation. The main claim of this paper is providing the first implementation of a PUF that can generate a random seed inside a secure processor.

SACHa: Self-Attestation of Configurable Hardware. [38] This paper proposes a solution where an FPGA-based prover core attests the entire FPGA, including self-attestation, serving as a tamper-resistant module. This enables hardware-based attestation of a processor, safeguarding the hardware/software system from malicious code updates. From the title and the abstract this does not seem to fit in the field of *remote* attestation, but the description of the protocol in the paper makes it clear that it does. For example, the verifier (Vrf) and the prover (Prv) communicate with each other via a public channel and Vrf is an entity not limited in processing power, e.g. a laptop, while Prv is an FPGA.

PAtt: Physics-based Attestation of Control Systems. [17] PLCs often lack hardware support for techniques like remote attestation, which can verify logic code integrity. Additionally, existing remote attestation methods do not verify the integrity of the physical process controlled by the PLC. For this reason the authors introduce PAtt, a system that combines remote software attestation with control process validation. PAtt utilizes operation permutations, subtle variations in operation sequences based on integrity measurements, to generate unique sensor readings during execution without affecting the physical process. This results in a novel PUF design that is based solely on physical processes. By incorporating integrity measurements into control operations, the system enables remote verification of the control logic's integrity using the resulting sensor data.

Defining Trust in IoT Environments via Distributed Remote Attestation using Blockchain. [23] In this work, a remote attestation protocol is introduced, utilizing blockchain technology to establish trust among IoT devices. The blockchain provides a secure framework for device registration, while in turn the attestation process relies on Physical Unclonable Functions (PUFs). Combining these technologies generates a tamper-resistant scheme, offering protection against physical and proxy attacks. This proposal aims to enhance security and integrity by leveraging blockchain and PUF-based attestation for IoT device trust establishment.

SGX-FPGA: Trusted Execution Environment for CPU-FPGA Heterogeneous Architecture. [40] This work introduces SGXFPGA, which establishes a trusted hardware isolation path and enables the first FPGA TEE. Trusted execution environments (TEEs), such as Intel SGX, are widely used for their minimal trusted computing base (TCB) and reduced attack surface, but current CPU-based TEEs do not support FPGAs. This is not ideal, given the rapid deployment of FPGA-based cloud computing services that still exhibit security vulnerabilities, hence the current proposal. SGXFPGA connects SGX enclaves and FPGAs in a heterogeneous CPU-FPGA architecture.

ATT-Auth: A Hybrid Protocol for Industrial IoT Attestation With Authentication. [5] This research paper focuses on developing attestation techniques specifically for Industrial Internet of Things (IIoT) systems. The proposed attestation protocols use PUFs to ensure hardware security and avoid physical attacks. The protocols also adopt a novel approach where the verifier does not calculate the reference checksum for each prover, instead, timing information is

	[25]	[38]	[17]	[23]	[40]	[5]	[15]	[18]	[3]	[2]	[41]	[4]	[10]	[33]
PUFs	H	L	H		M		H	H		M		L		M
RoT	H			H	L	M	M		L					H
Attacks	L	M	L	M	L	H	H	L	H	H	H	H	M	
AdvModel			H	H		H	H				L	M	H	L
SecEval		L	H			H	L	H	H	H	H	H	H	
CompEval		H			M	L	M		L		M		M	M
Energy								H	H					

Table 1: A high level overview on the papers of this survey and the level of detail with which they tackle the proposed problems

utilized. The proposed protocols achieve scalability by using timing information, instead of having the verifier calculate the reference checksums for every prover. This allows for efficient attestation of multiple devices in large-scale networks, with a high probability of detecting malware and reduced computation overhead.

AAoT: Lightweight attestation and authentication of low-resource things in IoT and CPS. [15] This paper addresses the increasing security and privacy risks associated with smart embedded devices in the context of IoT and CPS. It focuses on enabling remote attestation and authentication for low-resource embedded devices with AAoT, a mechanism that provides software integrity, mutual authentication, and tamper-proof features. AAoT utilizes PUFs, random memory filling, and software attestation without requiring changes to existing micro-controller units (MCUs). Efficient implementations and optimizations for each component of AAoT are demonstrated, including PUF-based memory filling, a checksum function, a pseudorandom function, a reverse fuzzy extractor, and a random number generator.

Design, Analysis and Application of Embedded Resistive RAM Based Strong Arbiter PUF. [18] This work focuses on Resistive Random Access Memory (RRAM)-based PUF designs, enabling the production of a strong arbiter PUF which uses a 1T-1R bit cell, designed with minimal changes to conventional RRAM memory arrays. The PUF utilizes repurposed components, such as the voltage sense amplifier, address, and data lines. The proposed PUF architecture is evaluated for uniqueness, uniformity, and reliability across various stages. It demonstrates satisfactory performance in terms of intra-die Hamming Distance (HD) and inter-die HD, passing the NIST tests. The authors assess its vulnerability to machine learning attacks and showcase its application for data attestation in the Internet of Things. The proposed PUF-based data attestation offers low energy consumption and high-speed performance.

HAtt: Hybrid Remote Attestation for the Internet of Things With High Availability. [3] This work introduces a remote attestation protocol called hybrid remote attestation to address the cybersecurity concerns of IoT devices. The proposed protocol ensures high availability during the software attestation process. It utilizes a randomized approach to attest different parts of an IoT device's memory and employs PUFs to protect device secrets from physical attacks. A security analysis confirms the effectiveness of the proposed protocol in detecting roaming malware while an implementation on Raspberry Pi and AVR/ARM-based ATMEl microcontrollers, along with a comparison to existing techniques, demonstrates improved availability and reduced energy consumption.

A Lightweight Authentication and Attestation Scheme for In-Transit Vehicles in IoV Scenario. [2] This paper proposes a lightweight and secure authentication and attestation scheme for vehicles while they are on the road. The scheme includes both vehicle authentication with Road Side Units (RSUs) and attestation of ECU firmware from edge servers connected to RSUs. Security and performance analyses are conducted, comparing the proposed protocol with existing ones, demonstrating its feasibility for deployment.

Scalable Attestation Protocol Resilient to Physical Attacks for IoT Environments. [41] This article proposes a lightweight attestation protocol for IoT systems, leveraging an ideal environment with PUFs. The protocol ensures resilience against strong adversaries who physically access IoT devices, while experimental results demonstrate the scalability of the protocol and its suitability for dynamic networks. This is the first work surveyed that deals with swarm or collective attestation.

IoT-Proctor: A Secure and Lightweight Device Patching Framework for Mitigating Malware Spread in IoT Networks. [4] This work presents a secure patching framework for IoT networks to control and mitigate malware spread. Traditional schemes are ineffective due to device-to-device propagation and the scale of IoT devices. The framework uses remote attestation and virtual patching with PUFs to detect compromised devices that contain malware. It employs different network isolation levels based on the SEIR model for access control. Security and performance analyses confirm the framework's effectiveness, achieving faster malware reduction compared to existing techniques.

RPRIA: Reputation and PUF-Based Remote Identity Attestation Protocol for Massive IoT Devices. [10] This article presents a remote identity attestation protocol for IoT devices in smart cities. The protocol utilizes reputation mechanisms and PUFs to ensure efficient and secure mutual authentication and key agreement. The security of the approach is formally proven using the Burrows-Abadi-Needham (BAN) logic and Scyther tool, while empirical performance evaluation demonstrates the protocol's favorable security and efficiency compared to other protocols.

A lightweight remote attestation using PUFs and hash-based signatures for low-end IoT devices. [33] This work proposes a low-cost Root of Trust for Measuring and Reporting (RoTMR) in IoT devices to enable lightweight and secure remote attestation. The RoTMR

combines a PUF and an Attestation Read-Only Memory (A-ROM), with hash-based digital signatures used in the attestation protocol. The PUF reconstructs secret keys, ensuring they are not stored, while the A-ROM contains unalterable attestation instructions executed sequentially. The solution is quantum-resistant and robust due to the unidirectionality of a hash function. The proposal utilizes One-Time Signature (OTS) and Many-Time Signature (MTS) schemes, well-suited for resource-constrained devices. Experimental validation with the ESP32 microcontroller demonstrates the effectiveness of the proposal, with OTS schemes requiring smaller code and faster execution times compared to ECDSA. OTS schemes also offer efficient communication bandwidth due to their small signatures.

5.2 PUFs

As mentioned in Section 4, PUFs can be easily divided into weak PUFs and strong PUFs, but their implementation details can vary a lot more. It is possible to obtain a PUF with different hardware components, such as an SRAM [16], a Ring Oscillator [28], physical processes [17, 39] and various more. The way these PUFs are engineered also leads to further classifications, where the type of PUF dictates how the PUF is constructed, e.g. Arbiter PUF [27] or XOR PUF [43]. All of these offer specific advantages and drawbacks, hence why it is often important to specify which type of PUF has been used in a protocol.

5.2.1 Types. The PUFs encountered while compiling this survey have been classified with regard to their novelty and type.

novel PUF design. Four papers included in this survey provide a novel PUF design and thus have described their PUF architecture in detail.

Lebedev et al. [25] propose two PUF designs, P256 and P512, that leverage an array of Ring Oscillators and a trapdoor fuzzy extractor to safely generate the CRPs. They employ a design by Herder et al. [20] to build a trapdoor LPN PUF with an array of identical ring oscillator pairs, which is then implemented on a commercial FPGA for evaluation.

Ghaeini et al. [17] introduce a novel PUF design that uses the physical processes typical of Control Systems in order to have a reliable source of entropy able to uniquely characterize each machine. This is described in the paper as a "PUF-like" use of physical processes, as it avoids the usual sources of entropy for PUFs, such as SRAM, RRAM or Ring Oscillators. The PUF design is less detailed than others in this section mainly due to the different mechanism that generates the characteristic PUF footprint. An interesting note is that this particular type of physical PUF is less prone to aging effects (none have been observed), which are very prominent in classic PUF designs due to physical effects in the resistors [26, 31].

Govindaraj et al. [18] provide perhaps the most detailed PUF design description in this survey (alongside [25]), proposing an Arbiter PUF based on Resistive Random Access Memory (RRAM), claiming that it is a promising candidate for future implementation of a non-volatile memory unit. This work also provides a good comparison with Ring Oscillator based PUFs, another very common design choice that incurs in additional area overhead compared to RRAM. The general design of the PUF follows closely the already mentioned Arbiter PUF.

Feng et al. [15] implement an SRAM PUF and provide details about its construction along with its robustness, uniqueness and randomness. The chosen type of PUF is weak, due to its inherent resilience against Machine Learning attacks, even if their protocol can accommodate both strong and weak PUFs, as only 2 CRPs are needed.

Weak PUF. A weak PUF based on SRAM is used in [15] and [33], where the weakness of the construction is never mentioned but can be inferred by thoroughly reading some referenced papers, in particular [32].

Strong PUF. The Arbiter PUF, a fairly common PUF design that accommodates the need for many CRPs, has been used in [40] and [18] (as described above), while [2] uses a tamperproof PUF based on RRAM that was first described in [44].

Generic PUF. Two papers avoid choosing a specific PUF, based on the assumption that their approach requires an "ideal" PUF, which is described as a PUF without any weaknesses [3, 38].

In the rest of the papers there is a similar lack of proper PUF definition or description, although it is possible to infer that a strong PUF might be needed for two of these works [4, 10], while a weak PUF should be sufficient for [5, 23], since they require significantly less CRPs than their counterparts.

5.2.2 Advantages. Only three papers [5, 40, 41] explicitly state the advantages of using PUFs in their attestation protocol. According to [40], the hardware nature of the PUF reduces the overhead that is often encountered when generating and storing keys, since strong PUFs can inherently generate large sets of CRPs. On a similar note, [41] asserts that the attractive features of PUFs are their lightweight requirements while achieving high throughputs. The addition of a PUF also does not increase the overhead for the attestation protocol. In [5] the focus is on the very low cost of production of PUFs, alongside maintaining a significantly low silicon area.

Less explicitly, the authors in [23] state that they introduce PUFs to an existing protocol, SWATT, in order to mitigate its inherent weakness against physical attacks, which can be considered an advantage intrinsic to PUFs.

5.3 Attestation and PUF synergy

Attestation protocols are harder to fit into a set taxonomy, as they are usually built incrementally on previous versions with specific improvements. An interesting thing to note in the works we reviewed is how the PUF is used in the system and how (or if) it synergizes with the attestation protocol. PUFs can be used at different stages of the attestation process, but they can also be used for separate phases of a security system that includes attestation (e.g. authentication).

The PUF is specifically identified as the root of trust of the attestation protocol in [3, 23, 25, 40]. The same can be said of [38], since the PUF is used to generate the cryptographic key used for the MAC that will deliver the attestation payload. In [17], the PUF is based on physical signals from the actuators and it is used directly for the attestation.

A more interesting use of the PUF can be found in [5, 15, 41], where the PUF is used in each iteration of the checksum function, inherently coupling it with the attestation protocol.

The PUF is used for attestation in [4] (since they mainly leverage HAtt [3]), but also for authorization and, more interestingly, for secure patching.

The PUF in [33] is combined with an Attestation Read-Only Memory to generate the actual root of trust for attestation, while [10] uses it embedded in the identity attestation protocol, making it an implicit root of trust. Similarly, the PUF in [18] is used to encrypt data in a data attestation protocol.

The only work where the PUF is not explicitly part of the attestation algorithm is [2], where it is used mainly to generate an authenticated communication channel. It can be argued that the secure channel is then used to exchange the messages in the attestation protocol, so it is parallel to the previous cited works.

5.4 Root of Trust

Only 7 of the examined papers mention a root of trust at all and only 3 of them explain in detail how the root of trust of their system is established [15, 25, 33].

RoT details. In [25] there are multiple Roots of Trust described, the first of which refers to the first-stage bootloader, which is a trusted program responsible for loading a payload binary segment from untrusted storage into secure system memory. Then the two implemented PUFs, P256 and P512, also act as part of the Root of Trust. Interestingly, this is also the only paper that evaluates the root of trust [25], showing how the implementation of the root of trust affects the size of the code and its latency.

The PUF combined with an Attestation PUF is employed in [33] in order to establish a Root of Trust for Measuring and Reporting. The authors argue that the usual methods of establishing a Root of Trust, such as a Trusted Platform Module, are too expensive and not available to most IoT devices, while PUFs can be easily embedded in most devices.

A root of trust for low-end embedded devices is proposed in [15], where the PUF is used to generate a PUF-based Root of Trust (PUFRoT). PUFRoT is a firmware that is able to safely measure the integrity and authenticity of the device, while also delivering reports to the verifier.

PUF as RoT. The PUF itself is also explicitly identified as part of the root of trust in [5, 23, 40], without further details.

In other works the root of trust is never mentioned, but the PUF is mainly used as a quasi-reliable source of entropy for key generation, sometimes becoming an implicit root of trust.

5.5 Attacks

Attestation protocols generally protect against tampering attacks, as the prover needs to be uncompromised in order for the verifier to accept the attestation attempt. That being said, there is a host of possible attacks that can be waged against systems that employ attestation measures.

In this section we collect and present all the attacks explicitly mentioned in the papers of this survey.

Impersonation attack. In this scenario, the adversary wants to convince the verifier that it is in fact an uncompromised prover, thus impersonating an IoT device that has not been attacked. It is possible also for the attacker to pretend to be the verifier, in order to initiate an attestation loop that can be used to replay the attestation payload for a future run of the protocol. In [3, 38] the PUF is part of the main mechanism that thwarts impersonation attacks, as PUFs offer the verifier an easy way to check if they are in fact communicating with the prover, thanks to their properties of unclonability and tamper evidence. In a similar way [2] proposes a method that embeds the PUF response with the private ID of the vehicle, meaning that only a legitimate vehicle is able to verify the generated message digest, effectively foiling any impersonation attempt. The PUF in [15] is used for authentication but also inside the checksum, in order to verify that the attestation result has been in fact computed on the right machine, effectively another type of impersonation attack. In [4] the mechanism is derived from HAtt [3] and it is proven theoretically that an impersonation attack cannot happen as long as the PUF is not compromised (e.g. due to a modeling attack). Two papers, [41] and [10] (here it is referred to as "fake identity") only mention the attack as a way in which attackers can affect negatively an attestation protocol.

The Man-In-The-Middle (MITM) attack is only mentioned in [2, 5]. The attack is included here because it is not very commonly treated and, in order for a successful MITM attack to take place, the adversary has to impersonate both parties in a communication instance.

The reason most papers list the impersonation attack as one of the main ones is possibly because PUFs are often used as a source of entropy for keys used in the authentication part of the protocol, which is usually what blocks impersonation attacks.

Malware. Malware is not detected or contained in [38], rather the system is designed in a way that there will be no malicious code after an update. The system also ensures that no computing devices can connect to the FPGA and run malicious code.

The reduction of the spread of malware is one of the main goals of [4]. In this paper, a PUF-based patching mechanism is presented as a solution to unbounded malware spread, as unpatched devices tend to be the most vulnerable to malicious code attacks. If the patching mechanism fails, a fail-safe might be activated, rendering the device unable to communicate with the outside, effectively blocking the malware spread.

In [2, 5, 15, 23, 41], malware is one of the attack vectors that the attackers could realistically implement in order to inflict damage, and that the attestation protocol aims to avoid. Similar to the mechanism engineered by [4], if a device is found to contain malware in [23], it is completely dropped from the network, rendering the malware impotent.

Roving malware is a type of malicious code that can relocate itself in memory or delete itself to come back at a future time, in order to avoid detection. This is the main concern of the work presented in [3], as it is a particularly difficult malware to detect with common techniques.

Replay attack. This attack involves the adversary capturing an existing attestation payload (i.e. from a previous succesful attestation protocol run) and re-using it from a corrupted prover in order to

	[25]	[38]	[17]	[23]	[40]	[5]	[15]	[18]	[3]	[2]	[41]	[4]	[10]	[33]
Impersonation		X				X	X		X	X	X	X	X	
Malware		X		X		X	X		X	X	X	X		
Replay		X	X	X			X		X	X	X	X	X	
Physical		X		X		X			X			X		
Cloning						X			X	X	X			
Side-channel	X							X		X	X			
Tampering				X					X			X		
DOS							X				X		X	
Modeling							X	X		X				
Memory						X	X							
Specific			X	X	X		X				X	X	X	

Table 2: X indicates that the indicated paper addresses the corresponding attack

fool the verifier. It is generally possible to foil this attack by including a nonce to the attestation challenge and subsequent response, in order to make sure that the received message is fresh. This is the approach used in [2, 3, 10, 38]. In [38], the nonce is employed alongside a failsafe mechanism that also bounds the MAC computation to the order in which the verifier reads the configuration frames.

An interesting approach to replay attack detection is employed in [17], where the system can detect a replayed sensor reading by analyzing the sensor traces and comparing them to expected patterns. This can be done through either statistical analysis, signal processing techniques, or machine learning algorithms.

We believe that the proxy attack mentioned in [23] falls under the description of replay attack. It is described as an attack where the adversary has control of a proxy node with legitimate software, and thus can be used to carry out the attestation instead of the compromised device. This is an example of a replay attack where the replayed payload is not an old and reused message, but a fresh one. The attack is foiled by the protocol described in the paper by requiring the PUF to be called in the checksum function, thus rendering the proxy attestation useless.

Similarly, the collusion attack is only mentioned in [15], but its description seems to fit the replay label, since it involves using a valid attestation payload from an uncompromised device in order to escape the protocol. The solution is also similar to the above, as the PUF is bound to the checksum function, effectively blocking any external device from computing the same payload.

The papers [4, 41] only mention the replay attack as one of the attacks that could be waged against the system, without specifying how it can be dealt with.

Physical attack. Physical attacks are very common in IoT and encompass any physical modification done to a system in order to achieve an adversary goal.

The addition of a malicious hardware module to multiple parts of the system is contemplated in [38], where a partial reconfiguration and configuration memory readback is used by the architecture to avoid such cases.

PUFs are touted as a way to avoid certain physical attacks in [3], [5] and [4], as they remove the need to store keys in the device's memory, thus eliminating it as a vector for attack.

The tamper evidence property of PUFs is used in [23] to enhance the security of SWATT [36], since the PUF will be rendered useless after physical tampering.

Cloning attack. Many devices can be cloned in order to create a perfect copy that can aid in performing malicious behaviors. PUFs are used in [2, 3, 5, 41] as an obvious foil to cloning attacks, as they are unclonable by design.

Side-channel attacks. Side-channel attacks can analyze the correlation between dynamic power consumption and the CRPs to extract information about the PUF's behavior. In [18] the proposed APUF design aims to foil side channel attacks by minimizing the correlation between the responses generated by the PUF and the power consumption of the circuit. This is done by introducing variations in the path delays, making it difficult to infer information about the responses based on power consumption.

The fact that the challenges and responses generated from the PUF are never exposed to the network is given as a reason for side-channel attack resistance in [2]. The protocol described in [41] aims to address side-channel attacks but it is never explained how, or if, it does so.

Tampering. With this attack we do not consider physical tampering of the devices, as that is covered in the previous "physical attacks" paragraph. The tampering of the blockchain is theorized in [23], where the security against such attack is guaranteed by the protocol, since it is tamper-proof if the number of malicious miners fails to exceed the number of honest miners.

In [3] tampering can happen in the packets exchanged during the attestation protocol, but it is foiled by the device which can verify that the packets are intact.

Data tampering is also considered in [4], where it is proven that the only way for an adversary to actually tamper with the messages exchanged in the protocol is by having direct access to the PUF (which is unfeasible in their scenario).

Denial Of Service attacks. Denial of Service (DoS) is a class of attacks that can affect most devices that connect to a public network, as it consists of overwhelming the system resources with enough requests that its functioning either slows to a crawl or stops altogether. Detecting and protecting against DoS attacks is not trivial

as it is often hard to distinguish between legitimate heavy network traffic and an actual attack [21]. Remote attestation protocols for IoT can be affected by these attacks, as many of these protocols rely on timing to determine wether a prover is compromised.

Requiring the verifier to be authenticated with the prover prevents DoS attacks according to [15]. A similar approach is adopted in [41], where the prover is uniquely identified through an ID that then allows the verifier to call an emergency function to block the attack.

DoS attacks are only mitigated to a certain extent in [10] thanks to its reputation mechanism, since messages coming from devices with low reputation are not processed.

Machine Learning (modeling) attacks. PUFs themselves can be prone to attacks, most importantly those utilizing Machine Learning models in order to create a reliable model of the PUF with a few CRPs. These attacks are particularly easy to wage against PUFs where the challenges are highly correlated to the responses [35], which is why many PUF constructions involve the addition of non-linearities to the circuit.

Weak PUFs are inherently resistant to modeling attacks since the space of CRPs is too small, which is one of the reasons they are chosen in [15].

In order to foil a modeling attack against strong PUFs in [2], the CRPs should not be exposed to the network. This prevents an adversary from building a training set that can be used to create a model of the PUF.

In [18], resistance to modeling attacks is one of the main driving factors in the PUF design. The way modeling attacks are dealt with is by incorporating XOR operations between multiple Arbiter PUF instances. XOR is a famously hard function to deal with in machine learning and this led to the design of many XOR-based PUFs, even if the security of these is all but guaranteed [7].

Memory attacks. A memory attack can be characterized as any attack that requires the manipulation of the device's memory in order to perform a malicious activity or avoid detection. This was already explored in [3], as memory relocation is one of the common mechanisms with which roving malware avoid detection.

In [15], memory attacks are dealt with by requiring the attestation protocol to use a PUF-based seed in order to traverse the memory. Therefore, an attacker would not be able to perform memory copy or memory compression attacks without first compromising the PUF itself. A similar consideration is cited in [5], since the presented protocol is an implementation of SWATT, which inherently randomizes memory access through a secredìt random seed, rendering memory attacks impossible.

Specific attacks. Some attacks are rarely discussed and appear only in one of the works in this survey, for this reason they are shown separately. These range from attacks that might already be discussed in previous works but lack proper naming to very specific types of attacks that are only considered due to a particular slant of the work itself. This is the case for the CPU-TO-CPU, FPGA-TO-CPU and CPU-TO-FPGA attacks mentioned in [40]. These attacks only make sense in this paper, as the attestation protocol proposed identifies the CPU and the FPGA as the usual prover-verifier roles commonly found in works of this field. On a similar note, the hash approximation attack only appears in [17], since this is the only work that focuses on actuation sequences of PLC systems that are encoded into hashes.

Sanctum [25] provides protection against a specific type of software attacks that extract private information by analyzing a program's memory access patterns while additionally providing protection against subtle side channel attacks that exploit the cache state and shared page tables.

Every attestation algorithm can intuitively suffer from an attack where the adversary simply evades the protocol in order to carry out their malicious activity. However, the attestation evasion attack is only mentioned in [4], where their protocol is said to be safe against such attack due to the HAtt protocol's [3] inherent resistance to evasion.

Injecting packets in a network can be considered an attack in many scenarios, such as MITM or DoS, but it is considered an attack in and of itself only in [4].

The counterfeit attack is mentioned only in [10] and might be a mislabeled impersonation attack, but the paper lacks any description of it or any method to deal with it. On a similar note, the reflection attack is only mentioned in [41], but without a proper description of such attack it cannot be included in any of the previous labels.

Out of scope attacks. Few papers outright declare attacks to be out of scope. In [38], the PUFs adopted are assumed to be ideal, thus PUF attacks are explicitly declared out of scope. Runtime attacks, control-flow attacks and physical attacks are considered out of scope for [15]. Only non-invasive attacks are considered in [41], meaning that all invasive attacks are implicitly out of scope.

No specific attacks have been mentioned for [33], but the possibility of the adversary performing sophisticated hardware attacks (i.e. as fault injection) is ruled out, since these could allow the modification of the Program Counter.

5.6 Adversary Assumptions

Another important aspect to discuss in remote attestation works for IoT is the adversary model, as it can provide a useful framework to better identify the attacks mentioned above. The characteristic of an adversary can encompass their inherent abilities, their goals and their success criteria.

The adversary is well defined in [23], where it is said to be able to eavesdrop on communications, intercept any message and impersonate devices. The adversary can also physically compromise the IoT devices.

Another similar adversary is found in [15], with the same abilities but with an added limitation of not being able to properly clone or tamper with the PUF, since the attacker cannot access the CRPs directly.

A very similar adversary is considered in [5], with the same network message manipulation capabilities and access to the hardware to perform physical attacks. A successful adversary in this scenario is able to authenticate with the verifier and install malware in Industrial IoT devices, all with the goal of inflicting physical and economic damage.

In [17], the adversary has compromised the device (the PLC in this specific case) with the goal to affect the actuations of the

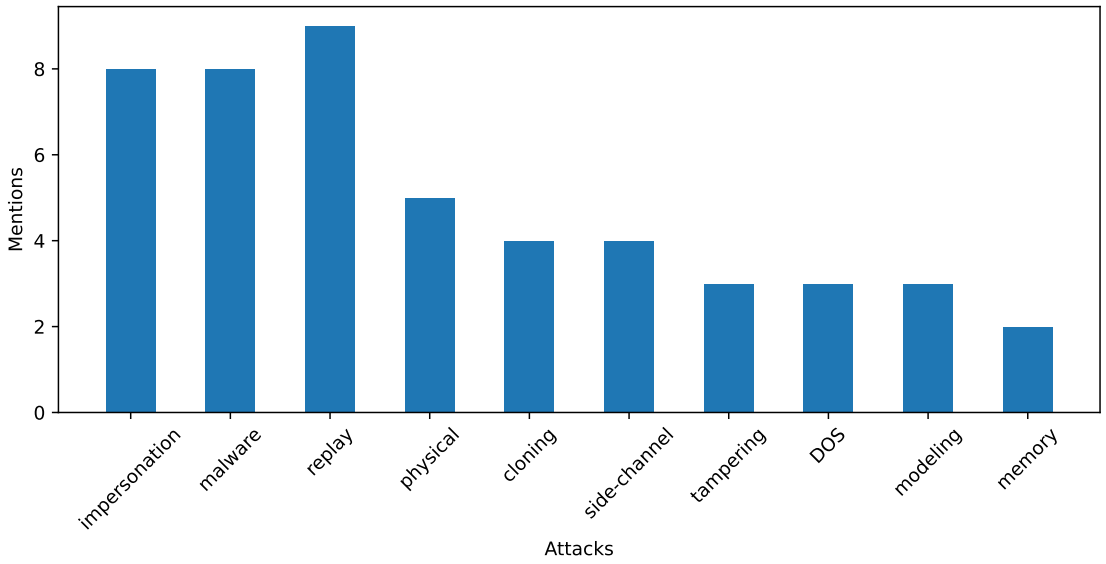

Figure 2: Number of mentions found in the papers for every attack

physical process without being detected by the attestation protocol. An attacker can only execute code on this compromised PLC, which means that they are bound to the inherent limitation of the PLC, such as limited processing power and memory.

A strong adversary which is able to physically access IoT devices is described in [41]. The success criteria of the adversary is again to authenticate itself to the verifier and install malware, with the usual goal of inflicting economic and physical damage.

The adversary introduced in [4] has full control of the network, just like in above examples, and can compromise the devices in order to use them to send data to other devices in the network.

The Dolev-Yao adversary model is widely recognized as one of the most powerful in literature [11] and it is considered as the main model in [10].

The only paper to not mention any attacks, [33], similarly defines the adversary only with its inability to perform fault attacks or other sophisticated hardware attacks.

5.7 Security Evaluation

Since remote attestation protocols are employed to enhance the security of IoT, it is important to assess the actual improvements that are achieved on this front. We consider an actual "security assessment" any explanation (or description) of how the proposed protocol achieves a specific security goal, then we divide these into "formal", "informal" and "empirical".

Informal security assessment. These assessments offer a descriptive evaluation on how their respective protocols deal with the proposed attacks. This is the case of [38], where every attack is followed by a brief explanation on how the protocol would react and neutralize it. The security evaluation offered in [15] contains slightly more extensive descriptions than the previous work but it is still lacking any empirical study or formal proof.

Empirical security assessment. Common empirical assessments simulate the attacks and practically evaluate the system's ability to cope with them.

In [17], the performance evaluation is said to be "theoretical" but it uses empirical experiments that are then assessed through common metrics such as False Positive Rate (FPR), False Negative Rate (FNR), Accuracy, F1-score, Sensitivity, Precision, Specificity and Matthews Correlation Coefficient (MCC).

The security of the PUF against modeling attacks is only verified in [18], where the generated CRPs are checked for uniqueness and uniformity, both characteristic that can foil machine learning attacks. A few modeling attacks are launched against the PUF, after which a Multi Layer Perceptron (MLP) is selected in order to assess the PUF's resilience against such attacks. Side-channel attacks resilience is also tested in this work, this is done by calculating the correlation coefficient between response bits and the power drawn from the unit.

The simulation of malware attacks in an IoT network is used in [4] to assess the ability of the framework to control, contain, and mitigate malware spread. This is paired with a formal assessment which is presented in the following.

Formal security assessment. This type of assessment can include both formal proofs and tool-assisted verification. As an example, the security properties of [3] have been tested with an automated tool called ProVerif [8], which uses approximations to deliver a sound proof of such properties. This means that if the tool asserts a specific security property is met, then that is actually satisfied, while it cannot prove when certain properties fail.

Another formal verification tool, Scyther [13], is used in the security assessment of [41]. Scyther leverages unbounded model checking methods and backward symbol state searching techniques to analyse security protocols. This work also provides a formal proof of the soundness of the attestation protocol by using BAN logic,

alongside a more qualitative analysis. BAN logic is also used in a similar way in [10].

The probability of detecting malware is calculated formally both in [5] and [2]. The former proposes a thorough probabilistic calculation to derive the threshold with which an attack can be detected, then it presents the probability of detection functionally linked to multiple different probabilities of compromised nodes. A different approach is shown in the latter, where the adversary's evasion probability is calculated by showing what is their best relocation strategy. The end result for a memory of N blocks is shown to be $(1 - \frac{1}{N})^N$ for every attestation protocol run, which is the best possible scenario from a security perspective.

The informal security assessment of [4] is accompanied by many lemmas with proofs, starting from the assumption that a PUF is in fact unique and unclonable and ending with a proof that the malware is successfully contained in the proposed framework.

No security assessment. Only one paper [25] specifies that the security evaluation of the proposed attestation protocol is out of scope. The remaining papers do not explain how the security goals are met and usually just mention the attacks that they thwart without any further analysis.

5.8 Performance Evaluation

There are three types of overhead that should be considered when applying an attestation protocol to an IoT device: 1. computational complexity, meaning the time that it takes to perform the actual attestation algorithm, 2. size overhead, or how much space is occupied by the attestation software and hardware, and 3. network overhead, measuring the size of the attestation payloads sent over the network and their rate of transmission. The computational complexity of an attestation protocol is of particular importance in the IoT field, as many processes are critical and thus cannot suffer any significant slow-down due to added security protocols. Another interesting aspect that is often glossed over is the network overhead caused by the attestation protocols, rendered particularly important in those protocols that rely on timing.

Energy consumption considerations. Parallel to the importance of considering the computational complexity of attestation protocols in IoT is the focus on energy consumption. IoT systems often cannot afford to allocate a significant amount of their energy to security protocols, which means that it is fundamental to study how these protocols affect the energy consumption of the devices. Only two of the reviewed papers explicitly consider energy consumption, [18] and [3]. In the former, the ratio of energy per bit is measured when generating a key from the PUF, alongside the speed required to perform the operation. The resulting total energy for the attestation of a single block of data is measured at 9.88pJ for every 64-bits data block. In the latter, the energy consumption analysis is much more detailed, with a thorough comparison of the performances of the proposed protocol versus others in the literature, accounting for key size.

Experimental complexity assessment. The computational complexity of the protocols can be empirically assessed by measuring the duration of their runtime. This technique is employed in [38], where the duration of the execution of the entire protocol has been measured to around 28.5 seconds.

Very detailed timing data is provided in [40], where the protocol is shown to have very little added computational overhead when compared to previous work. Another very detailed analysis can be found in [15] where every component involved in the protocol is measured individually to check where the complexity resides.

The speed of the protocol in [18] is measured in kbps (120 kbps is the best result), as the duration of every attestation run depends solely on the data block size.

In [41], the computational overhead of the protocol is compared to 4 existing relevant protocols and it is shown that it is a clear improvement over 3 of them, while being very close but eventually losing in computational performance to [5]. The communication overhead (discussed below) is what sets this paper apart from the compared literature.

The main focus of [4] is patching of multiple devices, thus their evaluation for computational overhead measures the time it takes for the protocol to patch a fixed number of devices. This is then compared to [19] and it is shown to be a clear improvement on every experiment.

Each operation involved in the protocol presented in [10] is individually timed and then favorably compared to two similar works in the literature. A similar approach is taken in [33], where the process has been divided into atomic components and the execution time of each component has been recorded and compared to other approaches.

Formal complexity assessment. Interestingly, only two papers provide a formal complexity study of their protocol using Big O notation [5, 23]. The formal evaluation of [23] only states that in their approach, the attestation protocol has complexity $O(N)$ when N iterations are needed, while other approaches in literature require $O(Nm\mathbf{ln}m)$, where m is is the field size for elliptic curve cryptography. Only [6] is cited as the source of this claim.

Size overhead. The whole SACHa architecture is shown to occupy less than 9% of the entire space of the FPGA [38].

A much more detailed breakdown of the space needed to implement the approach is offered in [40], where the footprint of both the PUF and the secure monitor implementations are evaluated in term of flip-flops, lookup tables, DSPs and BRAMS. The resulting architecture is estimated to not consume more than 1% of the allotted space for the PUF and 25% for the FPGA secure monitor. With the same spirit, all components are individually evaluated for space in [15].

The size of the public key and of the generated signatures is the main concern for [33], along with the code size. This is corroborated by an analysis on the communication overhead of the protocol which we mention in the following.

Communication overhead assessment. In [33], the size of the generated signature is the most important factor for reducing the communication overhead, as the size of the public key is less important over multiple shared messages.

The protocol introduced in [5] is argued to have less communication overhead than other related ones because it only requires the exchange of two messages.

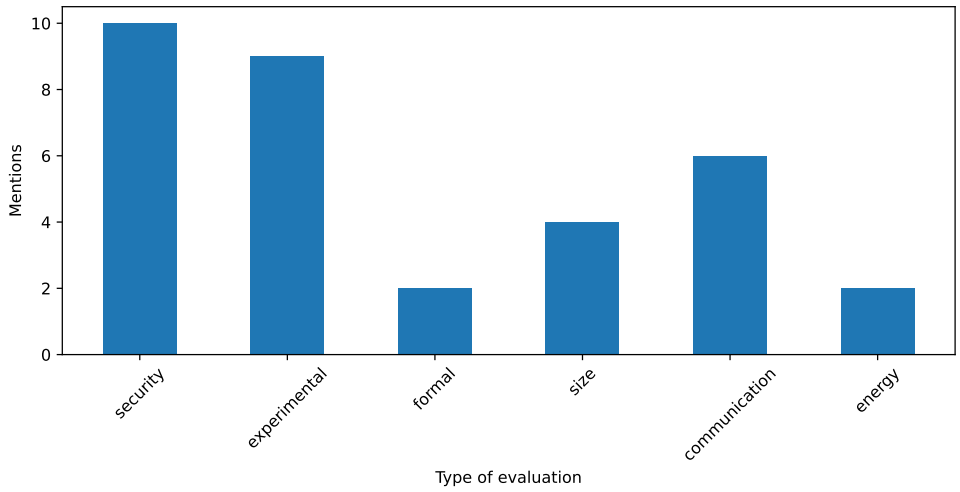

Figure 3: Types of evaluation found in this review and their frequency

The communication overhead of the system proposed in [41] is compared to 4 works in the literature and shows a notable improvement over all of them, especially while increasing the number of IoT devices to be attested. Similarly, the communication overhead of [3] is compared to other works in literature and is shown to be better than all but one, which is in turn worse in energy consumption.

The communication overhead of the central authentication server (IOC) presented in [10] is reduced when authenticating a large number of IoT devices simultaneously using the proposed protocol. This is a result of the protocol's approach to aggregate and preprocess a large number of IoT attestation request messages through the aggregator before sending them to the IOC for processing. Consequently, this approach significantly reduces the communication overhead between the IoT devices and the IOC.

The delay of the network communication is identified as the dominant characteristic of the protocol duration in [38].

An intuitive representation of the evaluation types used in the paper surveyed can be found in Fig. 3.

6 DISCUSSION AND OPEN RESEARCH CHALLENGES

In this section we summarize some insights that we gathered while compiling this survey. Some future research directions have been identified and will be proposed at the end of the section.

One of the first things that becomes evident while compiling a survey of this nature is that the terminology used across different papers that are positioned within the same research space could be more consistent. The simplest example of this is the names used for the attacks that are thwarted by the various attestation protocols, some of which appear only once and lack a formal (sometimes even informal) description (e.g. the collusion attack in [10]). This makes them especially hard to categorize within a taxonomy.

Adversarial assumptions are also underutilized, sometimes missing altogether in the surveyed papers. It is especially important to characterize the capabilities of an attacker in systems such as remote IoT devices, since the possible attack surface is particularly large.

The lack of a unified terminology and of a well-specified adversarial model are perfectly understandable problems when considering that the research space explored in this survey is an intersection of Embedded Systems, Software Security and, often, Cryptography. In order to tackle the lack of a common language, it could be helpful to develop a taxonomy of the attacks and the adversarial models that plague this specific field.

Another interesting observation is that all of the surveyed works only consider static remote attestation, meaning that the part of the system that is attested does not depend on the state of the programs being run. This means that an entire class of attacks, namely dynamic attacks, is excluded from the protection. For example, ROP attacks [30] allow the adversary to chain together the basic blocks of a non-malicious program (that would then pass static attestation) in order to generate new paths that can perform malicious activities.

Another class of attacks that is never considered is non-control data attacks [12], which typically compromise some variables that hold data used to drive the control flow of the program, resulting in the adversary gaining access to protected (albeit legitimate) paths. In order to challenge these attacks, all of the registers and data sections of the volatile memory need to be attested during the program run, which is not always an easy task.

Dynamic remote attestation is a fairly new and still not widely used approach, especially for IoT devices, but it can be introduced to deal with these attacks [1, 14].

This research direction could also provide a novel way of embedding the PUF in the attestation protocol, instead of the typical usage in authentication/encryption and in the checksum of a static attestation protocol. PUFs are lightweight devices that have low power usage when they are called so we believe they would be great candidates for a dynamic remote attestation protocol.

ACKNOWLEDGMENTS

This project has received funding from the European Union's HorizonEurope research and innovation programme under grant agreement No.101070238; additionally it was carried out within the Interconnected Nord-Est Innovation Ecosystem (iNEST) and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.5 – D.D. 1058 23/06/2022, ECS00000043). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

REFERENCES

[1] Tigist Abera, N Asokan, Lucas Davi, Jan-Erik Ekberg, Thomas Nyman, Andrew Paverd, Ahmad-Reza Sadeghi, and Gene Tsudik. 2016. C-FLAT: control-flow attestation for embedded systems software. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 743–754.

[2] Tejasvi Alladi, Sombuddha Chakravarty, Vinay Chamola, and Mohsen Guizani. 2020. A lightweight authentication and attestation scheme for in-transit vehicles in IoV scenario. *IEEE Transactions on Vehicular Technology* 69, 12 (2020), 14188–14197.

[3] Muhammad Naveed Aman, Mohamed Haroon Basheer, Siddhant Dash, Jun Wen Wong, Jia Xu, Hoon Wei Lim, and Biplab Sikdar. 2020. HAtt: Hybrid remote attestation for the Internet of Things with high availability. *IEEE Internet of Things Journal* 7, 8 (2020), 7220–7233.

[4] Muhammad Naveed Aman, Uzair Javaid, and Biplab Sikdar. 2021. IoT-proctor: a secure and lightweight device patching framework for mitigating malware spread in IoT networks. *IEEE Systems Journal* 16, 3 (2021), 3468–3479.

[5] Muhammad Naveed Aman and Biplab Sikdar. 2018. ATT-Auth: A hybrid protocol for industrial IoT attestation with authentication. *IEEE Internet of Things Journal* 5, 6 (2018), 5119–5131.

[6] Mohammad Tariq Banday. 2019. *Cryptographic Security Solutions for the Internet of Things*. IGI Global.

[7] Georg T Becker. 2015. The gap between promise and reality: On the insecurity of XOR arbiter PUFs. In *Cryptographic Hardware and Embedded Systems–CHES 2015: 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings 17*. Springer, 535–555.

[8] Bruno Blanchet. 2005. Proverif automatic cryptographic protocol verifier user manual. *CNRS, Departement dInformatique, Ecole Normale Superieure, Paris* (2005).

[9] Marco Brohet and Francesco Regazzoni. 2023. A Survey on Thwarting Memory Corruption in RISC-V. *Comput. Surveys* (2023).

[10] Jin Cao, Sheng Li, Ruhui Ma, Yuxi Han, Yueyu Zhang, and Hui Li. 2022. RPRIA: Reputation and PUF-Based Remote Identity Attestation Protocol for Massive IoT Devices. *IEEE Internet of Things Journal* 9, 19 (2022), 19174–19187.

[11] Iliano Cervesato. 2001. The Dolev-Yao intruder is the most powerful attacker. In *16th Annual Symposium on Logic in Computer Science—LICS*, Vol. 1. Citeseer, 1–2.

[12] Shuo Chen, Jun Xu, Emre Can Sezer, Prachi Gauriar, and Ravishankar K Iyer. 2005. Non-control-data attacks are realistic threats.. In *USENIX security symposium*, Vol. 5. 146.

[13] Casimier Joseph Franciscus Cremers. 2006. Scyther: Semantics and verification of security protocols. (2006).

[14] Ghada Dessouky, Tigist Abera, Ahmad Ibrahim, and Ahmad-Reza Sadeghi. 2018. Litehax: lightweight hardware-assisted attestation of program execution. In *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 1–8.

[15] Wei Feng, Yu Qin, Shijun Zhao, and Dengguo Feng. 2018. AAoT: Lightweight attestation and authentication of low-resource things in IoT and CPS. *Computer Networks* 134 (2018), 167–182.

[16] Achiranshu Garg and Tony T Kim. 2014. Design of SRAM PUF with improved uniformity and reliability utilizing device aging effect. In *2014 IEEE international symposium on circuits and systems (ISCAS)*. IEEE, 1941–1944.

[17] Hamid Reza Ghaeini, Matthew Chan, Raad Bahmani, Ferdinand Brasser, Luis Garcia, Jianying Zhou, Ahmad-Reza Sadeghi, Nils Ole Tippenhauer, and Saman Zonouz. 2019. {PAtt}: Physics-based Attestation of Control Systems. In *22nd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2019)*. 165–180.

[18] Rekha Govindaraj, Swaroop Ghosh, and Srinivas Katkoori. 2018. Design, analysis and application of embedded resistive RAM based strong arbiter PUF. *IEEE Transactions on Dependable and Secure Computing* 17, 6 (2018), 1232–1242.

[19] Nadra Guizani and Arif Ghafoor. 2020. A network function virtualization system for detecting malware in large IoT based networks. *IEEE Journal on Selected Areas in Communications* 38, 6 (2020), 1218–1228.

[20] Charles Herder, Ling Ren, Marten Van Dijk, Meng-Day Yu, and Srinivas Devadas. 2016. Trapdoor computational fuzzy extractors and stateless cryptographically-secure physical unclonable functions. *IEEE Transactions on Dependable and Secure Computing* 14, 1 (2016), 65–82.

[21] Alefiya Hussain, John Heidemann, and Christos Papadopoulos. 2003. A framework for classifying denial of service attacks. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. 99–110.

[22] Furqan Jameel, Uzair Javaid, Wali Ullah Khan, Muhammad Naveed Aman, Haris Pervaiz, and Riku Jäntti. 2020. Reinforcement learning in blockchain-enabled IIoT networks: A survey of recent advances and open challenges. *Sustainability* 12, 12 (2020), 5161.

[23] Uzair Javaid, Muhammad Naveed Aman, and Biplab Sikdar. 2020. Defining trust in IoT environments via distributed remote attestation using blockchain. In *Proceedings of the Twenty-First International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*. 321–326.

[24] Boyu Kuang, Anmin Fu, Willy Susilo, Shui Yu, and Yansong Gao. 2022. A survey of remote attestation in Internet of Things: Attacks, countermeasures, and prospects. *Computers & Security* 112 (2022), 102498.

[25] Ilia Lebedev, Kyle Hogan, and Srinivas Devadas. 2018. Secure boot and remote attestation in the sanctum processor. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*. IEEE, 46–60.

[26] Chao Qun Liu, Yuan Cao, and Chip Hong Chang. 2017. ACRO-PUF: A low-power, reliable and aging-resilient current starved inverter-based ring oscillator physical unclonable function. *IEEE Transactions on Circuits and Systems I: Regular Papers* 64, 12 (2017), 3138–3149.

[27] Takanori Machida, Dai Yamamoto, Mitsugu Iwamoto, Kazuo Sakiyama, et al. 2015. A new arbiter PUF for enhancing unpredictability on FPGA. *The Scientific World Journal* 2015 (2015).

[28] Abhranil Maiti and Patrick Schaumont. 2011. Improved ring oscillator PUF: An FPGA-friendly secure primitive. *Journal of cryptology* 24 (2011), 375–397.

[29] Imran Makhdoom, Mehran Abolhasan, Daniel Franklin, Justin Lipman, Christian Zimmermann, Massimo Piccardi, and Negin Shariati Moghadam. 2023. Detecting Compromised IoT Devices: Existing Techniques, Challenges, and A Way Forward. *Computers & Security* (2023), 103384.

[30] Marco Prandini and Marco Ramilli. 2012. Return-oriented programming. *IEEE Security & Privacy* 10, 6 (2012), 84–87.

[31] Md Tauhidur Rahman, Fahim Rahman, Domenic Forte, and Mark Tehranipoor. 2015. An aging-resistant RO-PUF for reliable key generation. *IEEE Transactions on Emerging Topics in Computing* 4, 3 (2015), 335–348.

[32] Roberto Román, Rosario Arjona, Javier Arcenegui, and Iluminada Baturone. 2020. Hardware security for extended merkle signature scheme using SRAM-based PUFs and TRNGs. In *2020 32nd International Conference on Microelectronics (ICM)*. IEEE, 1–4.

[33] Roberto Román, Rosario Arjona, and Iluminada Baturone. 2023. A lightweight remote attestation using PUFs and hash-based signatures for low-end IoT devices. *Future Generation Computer Systems* (2023).

[34] Paul D Rosero-Montalvo, Zsolt István, and Wilmar Hernandez. 2023. A Survey of Trusted Computing Solutions Using FPGAs. *IEEE Access* 11 (2023), 31583–31593.

[35] Ulrich Rührmair, Jan Sölter, Frank Sehnke, Xiaolin Xu, Ahmed Mahmoud, Vera Stoyanova, Gideon Dror, Jürgen Schmidhuber, Wayne Burleson, and Srinivas Devadas. 2013. PUF modeling attacks on simulated and silicon data. *IEEE transactions on information forensics and security* 8, 11 (2013), 1876–1891.

[36] Arvind Seshadri, Adrian Perrig, Leendert Van Doorn, and Pradeep Khosla. 2004. SWATT: Software-based attestation for embedded devices. In *IEEE Symposium on Security and Privacy, 2004. Proceedings. 2004.* IEEE, 272–282.

[37] Mark Shaneck, Karthikeyan Mahadevan, Vishal Kher, and Yongdae Kim. 2005. Remote software-based attestation for wireless sensors. In *Security and Privacy in Ad-hoc and Sensor Networks: Second European Workshop, ESAS 2005, Visegrad, Hungary, July 13-14, 2005. Revised Selected Papers 2*. Springer, 27–41.

[38] Jo Vliegen, Md Masoom Rabbani, Mauro Conti, and Nele Mentens. 2019. SACHa: Self-attestation of configurable hardware. In *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 746–751.

[39] Oliver Willers, Christopher Huth, Jorge Guajardo, and Helmut Seidel. 2016. MEMS gyroscopes as physical unclonable functions. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 591–602.

[40] Ke Xia, Yukui Luo, Xiaolin Xu, and Sheng Wei. 2021. Sgx-fpga: Trusted execution environment for cpu-fpga heterogeneous architecture. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 301–306.

[41] Xinyin Xiang, Jin Cao, and Weiguo Fan. 2020. Scalable attestation protocol resilient to physical attacks for IoT environments. *IEEE systems journal* 15, 3 (2020), 4566–4577.

[42] Yi Yang, Xinran Wang, Sencun Zhu, and Guohong Cao. 2007. Distributed software-based attestation for node compromise detection in sensor networks. In *2007 26th IEEE International Symposium on Reliable Distributed Systems (SRDS 2007)*. IEEE, 219–230.

[43] Lei Zhang, Chenghua Wang, Weiqiang Liu, Maire O'Neill, and Fabrizio Lombardi. 2017. XOR gate based low-cost configurable RO PUF. In *2017 IEEE International symposium on circuits and systems (ISCAS)*. IEEE, 1–4.

[44] Xiaojin Zhao, Qiang Zhao, Yongpan Liu, and Feng Zhang. 2020. An ultracompact switching-voltage-based fully reconfigurable RRAM PUF with low native instability. *IEEE Transactions on Electron Devices* 67, 7 (2020), 3010–3013.