Physically Restricted Authentication and Encryption for Cyber-physical Systems

Michael Kirkpatrick
Department of Computer Science
Purdue University
mkirkpat@cs.purdue.edu

Elisa Bertino
CERIAS
Purdue University
bertino@cerias.purdue.edu

Frederick T. Sheldon
Cyberspace Sciences & Information Intelligence Research
Oak Ridge national Laboratory
sheldonft@ornl.gov

Abstract

Cyber-physical systems (CPS) are characterized by the close linkage of computational resources and physical devices. These systems can be deployed in a number of critical infrastructure settings. As a result, the security requirements of CPS are different than traditional computing architectures. For example, critical functions must be identified and isolated from interference by other functions. Similarly, lightweight schemes may be required, as CPS can include devices with limited computing power.

One approach that offers promise for CPS security is the use of lightweight, hardware-based authentication. Specifically, we consider the use of Physical Unclonable Functions (PUFs) to bind an access request to specific hardware with device-specific keys. PUFs are implemented in hardware, such as SRAM, and can be used to uniquely identify the device. This technology could be used in CPS to ensure location-based access control and encryption, both of which would be desirable for CPS implementations.

The submitted manuscript has been authored by a contractor of the U.S. Government under contract DE-AC05-00OR22725. Accordingly, the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.

1 Introduction

Cyber-physical systems (CPS) [16, 11], which consist of the integration of networked sensors, computational resources and physical devices, pose a number of security challenges that differ from traditional computing architectures [12, 2]. CPS include critical infrastructure, for example Supervisory Control and Data Acquisition (SCADA) and Digital Control Systems (DCS). Given the vital nature of these systems, it is crucial that CPS provide a greater level of data integrity than traditional distributed systems.

A major challenge for data integrity in CPS is that of provenance, which refers to the origin of the data. As CPS can be one aspect of a complex system, adequate protections must be in place to ensure the claimed source of the data is accurate and tamper-proof. Additionally, the data may be aggregated with other data, yet the origin must be kept intact. The next challenge is to ensure proper access control for the system. The security mechanisms must ensure that only authorized actors insert data into the supply chain, and the integrity of the received data must be guaranteed. In short, CPS requires that the origin of data be secure and traceable, and that the data is delivered without unauthorized modification. To address these issues, we look to advanced techniques in authentication, encryption, and physical device identification.

Traditional approaches to authentication and access control are identity-based. In a typical scenario, a user (or a machine with authority delegated by the user) at-

tempting to make an access request presents a set of credentials that make a claim about the user's identity along with a proof that the claim is correct. The proof may take the form of a password or a digital certificate that includes the user's public encryption key. This form of authentication may be of insufficient strength for critical systems, as users have been shown to have poor security practices, such as revealing passwords in exchange for chocolate [1]. Furthermore, access requests in CPS may originate from physical processes independent of any human user; thus, the traditional notion of identity may be inappropriate.

Instead of relying on identity-based techniques, CPS need stronger forms of assurance to prevent untrusted devices from generating or modifying data. As such, we are specifically interested in techniques of binding an access request to specific hardware. Attestation techniques have been proposed [13, 14] that partially accomplish this goal. In these schemes, before a server grants access to certain data or services, the server requires the client perform a number of verification tests. These tests can report on properties of the remote system configuration, such as the operating system version, software patch levels, etc. The server can then take this information into consideration with regard to the request.

We see three drawbacks to these approaches in the context of CPS. First, these schemes generally assume the presence of a Trusted Platform Module (TPM) [17] that is capable of a number of operations, including various forms of cryptography. This assumption may be too strong for certain forms of CPS. That is, for small devices with tight power constraints (e.g., some embedded systems, sensors), incorporation of a TPM may simply be infeasible. Second, attestation schemes focus on ensuring the machine is configured in an acceptable manner, rather than identifying the hardware itself. As such, attestation cannot necessarily distinguish two machines that are configured in the same manner. Third, someone with physical access to the machine could disable or reset the TPM, thus denying the requisite assurances.

In this paper, we propose the exploration of other hardware-based authentication techniques for CPS, specifically focusing on Physical Unclonable Functions (PUFs). The inherent physical limitations of manufacturing devices introduce minor differences between multiple copies of the same hardware design. PUFs quantify these variations to produce a value that is guaranteed to be unique for each hardware instance. However, PUFs are deterministic, as repeating the PUF evaluation on the same hardware device will always produce the same value. Thus, PUFs can be used to confirm the unique identity of a hardware device.

Once the hardware instance has been identified, CPS can then enforce a number of additional access constraints. For instance, if the location of the hardware is known, spatial constraints can be applied [4]. Other work has focused on the use of contextual factors [9] for pervasive devices. The hardware identity could be considered as one of these factors.

2 Physical Unclonable Function (PUF)

A Physical Unclonable Function (PUF) [3, 6] (also called a Physical Random Function) is a determinstic challenge-response mechanism whose behavior is determined by the physical structure of the hardware itself. That is, given a challenge C_i, the PUF for a particular device will respond with R_i with almost certain probability. To ensure determinism, certain types of PUFs require the use of a fuzzy extractor algorithm that corrects any slight variations [8]. Executing the PUF with the same input C_i on a different physical instance of the same hardware design will produce the response $R_i' \neq R_i$ due to inherent physical differences in the manufacturing process.

While a number of approaches to PUFs have been proposed, we are particularly interested in those based on SRAM [7]. When power is initially being supplied to an SRAM, an individual memory location will tend to store the same value (0 or 1) consistently, despite any writes to that location during previous operation. Thus, for SRAM PUFs, C_i consists of a block of memory locations, and R_i is the binary string that is stored in those locations during power-up. Another approach is to simulate this SRAM effect by creating an unstable cross-coupled circuit (called a Butterfly PUF) in an FPGA [10]. The output (0 or 1) depends on the delay created by unmeasurable differences in the circuit wires.

A common use of PUFs is for the secure storage of cryptographic keys [15, 8, 7]. For example, consider the storage of a private key K. When the key is

installed into PUF-enabled hardware, the PUF is presented with challenge C_i and produces response R_i. This value is then combined with the key via XOR to create a value $X = K \oplus R_i$. The value X then gets stored locally. At run-time, the private key is reconstructed by combining the stored value with the PUF response, i.e., $K = X \oplus R_i$.

Using SRAM PUFs to store keys in this manner offers a number of very attractive features. First, X has no value in and of itself, so it can be kept in plaintext on any storage device. If an attacker were to gain access to the storage device, transferring X to another machine would not leak any information about the key K. This fact derives from the observation that every bit of R_i can be a 0 or 1 with equal probability and is determined only by the PUF on that device. Thus, PUFs offer a unique ability to bind a cryptographic key to the physical hardware.

Next, SRAM is generally integrated with a processor, including ASICs, FPGAs, and micro-controllers. SRAM is also used for high-speed caches in microprocessors. As a result, evaluating the PUF can occur on the processor itself without sending the output to any other part of the machine. Thus, the reconstructed key is only present on the chip itself and is never present in main memory. As a result, an attacker could not exploit a software bug to read the key or the PUF response from memory.

Finally, SRAM is widely used in embedded systems that require low power usage. These types of systems are included in CPS. For many of these devices, a TPM or other tamper-proof hardware may be inappropriate, due to the cost of implementation, the power consumption, or other reasons. However, a PUF could then be implemented, as SRAM is already present on the device.

3 Restricted Authentication and Encryption

Given a value that is guaranteed to be unique for each instance of a hardware device, CPS can provide advanced forms of authentication and encryption. First, the PUF output could be used as a unique identifier to restrict access control to certain devices. One method for accomplishing this would be to use the PUF output in a zero-knowledge proof of identity, such as proposed by Feige, Fiat and Shamir [5]. In this scheme, a prover commits to a secret value s by revealing $s^2 \pmod n$ for some n. The protocol defines a method for proving knowledge of s without revealing any additional information about the secret. The PUF response R_i could be used as the secret value s in this protocol.

Once the proof of the machine's identity has been verified, the server can make an access control decision based on prior knowledge of the machine. For instance, if the machine's trustworthiness has been previously evaluated, the server could grant full or partial access accordingly. Additionally, if there is a strong linkage between hardware devices and users, binding authentication to physical hardware in this manner could be used to detect and track a malicious insider.

Another way to use the PUF-based hardware identifier would be to create cryptographic keys that are unique to each device. Guajardo *et al.* showed how the PUF response can be used as a key for elliptic curve cryptography. Creating keys based on PUFs can address the problem of data provenance by using the key to sign all data generated from that piece of hardware. Even if the data is aggregated at a later point, the provenance can be preserved through the use of aggregated signature techniques. Also, if the location of the requesting device is known, the PUF approach could enforce a type of location-sensitive encryption.

4 Challenges and Open Problems

There are a number of challenges and open problems with using PUFs for CPS. First, PUFs have been implemented in certain types of hardware, such as SRAM, but doing so requires extra work by the programmer. To our knowledge, there are no tools or code libraries implemented that provides an easy interface for the programmer. Generating these tools would expand the opportunities for authentication and encryption based on PUFs.

Next, protecting the challenge-response pair (C_i, R_i) is more difficult than doing so for traditional cryptographic keys. If an attacker manages to discover a cryptographic key K (far from a trivial feat), the owner of the key can simply generate a new key. However, a physical device cannot generate a new response R_i' for the challenge C_i if R_i is discovered. That is, there is a sort of delicacy to PUF-generated

keys that is not present in other keys. Thus, for any protocol built around PUFs, no information about R_i should be required or leaked. Additionally, the protocol must provide an adequate revocation process for a challenge-response pair.

The use of PUFs creates an opportunity for new cryptographic protocols. We have described previously how PUFs could be used as part of a zero-knowledge proof of identity, such as the Feige-Fiat-Shamir scheme. Guajardo *et al.* described the use of PUFs with elliptic curve cryptography [7]. Some CPS devices may be incapable of performing the operations required for these protocols, but may still be able to execute a PUF. For such systems, new protocols would need to be designed specifically for use with PUFs.

Another challenge is how to address the heterogeneous nature of CPS. CPS incorporate sensors and devices of multiple types. Despite the pervasive presence of SRAM, PUFs may not be appropriate for all components of CPS. Assuming that to be the case, it is an open challenge of how to design a flexible access control and authentication mechanism that can combine multiple approaches to the physical restrictions. Furthermore, the administration of such a mechanism may become very complicated, especially for very large complex systems.

Finally, CPS may include very large, distributed networks of devices and sensors. Hence, these systems require scalable identity management of their components. While we have proposed the use of PUFs with existing identity protocols, we have not considered the issue of scalability for these systems. Incorporating a central trusted authority that binds a device to its PUF challenge-response pairs may be impractical. Devising a scalable identity management scheme for CPS built on PUFs is thus an open problem.

5 Conclusion

Physical Unclonable Functions (PUFs) create a challenge-response mechanism that is unique for each instance of a hardware device. As a result, PUFs offer an approach to uniquely identify hardware devices, as well as the generation and physical binding of cryptographic keys. This technique can applied to a number of challenges in the field of CPS. For instance, PUFs can be used to address data provenance, integrity, and device identity management. We have described how PUFs can be built on SRAM. The pervasive presence of SRAM implies that PUFs may be used where other approaches to trusted hardware, such as TPM, are inappropriate. Finally, we have identified a number of directions for future research involving the use of PUFs in CPS.

References

[1] Passwords revealed by sweet deal. http://news.bbc.co.uk/2/hi/technology/ 3639679.stm, April 2004.

[2] M. Anand, E. Cronin, M. Sherr, M. Blaze, Z. Ives, and I. Lee. Security challenges in next generation cyber physical systems. In *Beyond SCADA: Networked Embedded Control for Cyber Physical Systems*, November 2006.

[3] M. J. Atallah, E. D. Bryant, J. T. Korb, and J. R. Rice. Binding software to specific native hardware in a vm environment: The puf challenge and opportunity. In *VMSEC '08*. ACM, 2008.

[4] M. L. Damiani, E. Bertino, B. Catania, and P. Perlasca. Geo-rbac: A spatially aware rbac. In *ACM Transactions on Information Systems and Security*, 2006.

[5] U. Feige, A. Fiat, and A. Shamir. Zero knowledge proofs of identity. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing (STOC)*, pages 210–217, 1987.

[6] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas. Controlled physical random functions. In *Proceedings of the 18th Annual Computer Security Applications Conference (ACSAC)*, 2002.

[7] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls. Fpga intrinsic pufs and their use for ip protection. In *Proceedings of the 9th Cryptographic Hardware and Embedded Systems Workshop (CHES)*, pages 63–80, 2007.

[8] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls. Physical unclonable functions and public-key crypto for fpga ip protection. In *International Conference on Field Programmable Logic and Applications*, pages 189–195, 2007.

[9] D. Kulkarni and A. Tripathi. Context-aware role-based access control in pervasive computing systems. In *Proceedings of the 14th Symposium on Access Control Models and Technologies (SACMAT)*, 2008.

[10] S. S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen, and P. Tuyls. Extended abstract: The butterfly puf protecting ip on every fpga. In *IEEE International Workshop on Hardware-Oriented Security and Trust (HOST)*, pages 67–70, 2008.

[11] E. A. Lee. Cyber physical systems: Design challenges. Technical Report UCB/EECS-2008-8, EECS Department, University of California, Berkeley, January 2008.

[12] C. Neuman. Understanding trust and security in scada systems. In *Beyond SCADA: Networked Embedded Control for Cyber Physical Systems*, November 2006.

[13] R. Sailer, T. Jaeger, X. Zhang, and L. van Doorn. Attestation-based policy enforcement for remote access. In *In Proceedings of the 11th ACM Conference on Computer and Communications Security (CCS '04)*, pages 308–317. ACM Press, 2004.

[14] D. Schellekens, B. Wyseur, and B. Preneel. Remote attestation on legacy operating systems with trusted platform modules. In *Science of Computer Programming*, pages 13–22, 2008.

[15] G. E. Suh and S. Devadas. Physcal unclonable functions for device authentication and secret key generation. In *Proceedings of the 44th IEEE Design Automation Conference (DAC)*, pages 9–14. IEEE Press, 2007.

[16] P. Tabuada. Cyber-physical systems: Position paper. In *NSF Workshop on Cyber-Physical Systems*, 2006.

[17] Trusted Computing Group. Trusted Platform Module Main Specification. http://www.trustedcomputinggroup.org/, October 2003.