PUF-Based Two-Factor Authentication Protocol for Securing the Power Grid Against Insider Threat

Kevin Hutto*, Shuva Paul*, Benjamin Newberg*, Vishnu Boyapati^,
Yathiendra Vunnam*, Santiago Grijalva*, and Vincent Mooney*^
*School of Electrical and Computer Engineering
^School of Computer Science
Georgia Institute of Technology, Atlanta, Georgia
{khutto30, spaul94, bnewberg, vboyapati6, yvunnam3}@gatech.edu,
{sgrijalva, mooney}@ece.gatech.edu

2022 IEEE Kansas Power and Energy Conference (KPEC) | 978-1-6654-6591-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/KPEC54747.2022.9814751

***Abstract*—Recent advances in smart grid technologies have enabled additional distributed control paradigms that allow more efficient and reliable operation. However, this creates new security concerns for the grid, such as attackers using spoofed grid control devices to generate false measurements. This paper introduces a two-factor authentication protocol leveraging standard public-key cryptography as one authentication factor and a hardware-based fingerprint, known as a Physical Unclonable Function, as a second authentication factor. This protocol incurs a small overhead and prevents cyber-attacks even when an adversary is able to compromise the cryptographic keys stored in the non-volatile memory of an intelligent control device.**

***Index Terms*—Physical Unclonable Function, Power Grid, Hardware Security**

I. Introduction

One of the biggest changes in the power industry recently is the move towards a smart grid [1]. Leveraging advanced communication, software and embedded computing, the smart grid enables novel types of coordination and control. The smart grid is a major enabling technology for the integration of renewable energy sources, such as wind and solar systems. A distributed control system is needed to integrate the widely variable energy sources as well as to meet scalability requirements of potentially millions of control points. The bulk power grid control also continues to evolve with modernized Supervisory Control and Data Acquisition (SCADA) systems, digital substations, Intelligent Energy Devices (IEDs) and wide-area control architectures [1].

However, with the advancement of these technologies comes new risks. The push for the smart grid has created a ubiquity of new network-connected computing devices within substations, control centers and customer facilities. This has created an entirely new attack surface, and these devices have become a primary target for individuals, organizations, and nation-states seeking to do damage to a grid. Recent examples of attacks on distribution control systems include the attack on the Ukrainian power grid in 2015, which targeted a single substation but affected a much larger area, and within the oil and gas industry the ransomware attack on Colonial Pipeline in 2021 [2] [3].

This work has been partially supported by the U.S. Department of Energy's Office of Cybersecurity, Energy Security, and Emergency Response (CESER) under Cybersecurity for Energy Delivery Systems (CEDS) Agreement Number DE-CR0000004 to the Georgia Tech Research Corporation".

II. Background

A. SCADA and Measurement Attacks

SCADA systems are critical components in the remote control of power systems [1]. Bulk and distribution-level control centers gather information from many different locations, such as measurements at substations, and make decisions quickly in a tight control loop [1]. Attacks on measurements within the power grid have been extensively studied, such as bad data injection, where false measurements are injected at the substation level [4]. IEDs such as substation relays and Phasor Measurement Units (PMUs) include sophisticated computing and networking capabilities for communication on a control network. It has been shown that a directed attack on measurement devices within a substation can result in incorrect information presented to the operator, and hence incorrect actions. The effects of false data attacks may propagate beyond the substation in which the attack occurs to other parts of the system [4][5][6].

B. RSA and TLS

One fundamental building block of cybersecurity is known as public key cryptography, which utilizes two related keys for each entity to be authenticated: a public key and a private key [7]. A public key scheme functions such that any action performed with one key can only be reversed using the other key. For example, if a message is encrypted with an entity's public key, it can only be decrypted with that entity's private key. One of the most well-known methods of public key cryptography is the Rivest-Shamir-Adleman algorithm, better known as RSA [7].

Public key cryptography algorithms can be utilized to provide secrecy, authentication, and integrity of data sent over an insecure channel [7]. However, public key cryptography algorithms tend to be very slow compared to algorithms that rely on a single key (called symmetric key cryptography). The most common symmetric key cryptography algorithm is the Advanced Encryption Standard (AES) [7]. Thus, RSA

most often is used strictly within authentication protocols to establish a key used for symmetric cryptography, often called a "session key." The security of these protocols relies on the private keys never being disclosed. If the private keys are disclosed, the protocol is insecure [7].

The Internet uses a collection of protocols known as Transport Layer Security (TLS) which allows the process of authenticating using public-key cryptography and establishing a session key to be transparent to the end user [8][9]. In TLS, communication can be configured to be either one-way or two-way (mutual) authentication. One-way authentication is common for many applications (i.e., the server selling a product authenticates but the client buying the product does not), but the protocol allows for two-way authentication by enabling additional settings, which is useful for distributed environments such as the power grid. When two-way authentication is required, often the client machine is provided a private key and the server the corresponding public key prior to operation to allow the authentication. The majority of networking equipment utilizes TLS to establish a secure connection. This connection is often called a "secure channel" [8]. Finally, a typical security requirement is the ability to generate random numbers on the fly, i.e., whenever needed. A random number generated and used only once is called a "nonce."

C. PUF Technology

A relatively recent trend in hardware-based security is the use of Physical Unclonable Functions (PUFs) [10][11]. A PUF is a hardware security primitive that utilizes tiny manufacturing variations, typically in silicon, to produce a unique digital fingerprint. A PUF can function as a digital fingerprint and can be implemented on the same medium as digital circuits such as microprocessors, field programmable gate arrays (FPGAs), and other computing devices. While there are many ways PUFs can be implemented in silicon, and many different manufacturing variances within digital circuits such as gate delay, resistance, and capacitance that can be leveraged to create the digital fingerprint, the basic interface for a PUF remains the same. The PUF produces a unique output, called a response, based on a particular input, called a challenge [10].

If a PUF is to be used for authentication, a step referred to as enrollment occurs. Enrollment collects a significant number of challenge/response pairs (CRPs) and stores the CRPs in a database [10]. From this database, whenever the device containing the PUF is to be authenticated, a challenge from this database will be selected, and the response received from the PUF will be checked against the response stored in the database. PUFs are typically sensitive to environmental variations, such as temperature and voltage, and thus require error correction to obtain stable behavior [10].

PUFs are typically further divided into two classes: weak PUFs and strong PUFs [10]. The primary distinction between these two classes is that strong PUFs provide a sufficiently large number of usable challenge/response pairs such that a brute force attack is infeasible. Further, a strong PUF requires that there exists no learnable relationship among the challenge/response pairs in a way that an adversary could predict a response for a previously unseen challenge. Many proposed PUF constructions have been shown to be vulnerable to machine learning attacks, which makes them weak PUFs [12]. In fact, any PUF that does not meet the requirements for a strong PUF is classified as a weak PUF. While there are still many interesting applications of weak PUFs, some applications should only be used by a strong PUF [10]. The most common type of commercially available PUF today is known as a static random access memory (SRAM) PUF [13]. An SRAM PUF is a weak PUF that obtains a source of inter-chip randomness via the power-on state of an SRAM. An individual bit in an SRAM may settle as either a 0 bit or a 1 bit if no clear or reset signal is applied on power-up. With careful manufacturing processing and tuning, the SRAM power-on state will form a Guassian distribution for the number of 1 or 0 bits found, with the probability density peak residing at an equal number of 1 and 0 bits. The specific SRAM addresses for each of the 1 or 0 bits will vary for each physical SRAM depending on per-chip manufacturing variances. These SRAM bits may then be used as a static source for a secret such as an AES key [13].

III. Attack Scenario

In our scenario, we model a skilled lone-wolf insider with a desire to cause disasters in the grid such as a partial blackout. We assume the attacker has access to and tries to manipulate an intelligent energy device (IED) within an electrical substation. Further, we assume that the attacker is trained in electronics, microprocessors, field-programmable gate arrays, programming, cryptography, and has familiarity with the target power systems. For the purpose of this discussion, we will assume a simple scenario in which the IED provides the reading of a voltage magnitude at a given bus and sends measurements at a regular rate to the control center. The control center performs control actions based on the measurements received from the voltage meter. Damage could conceivably occur if the control center takes action based on erroneous or malicious measurements received [4].

The attack consists of the lone-wolf insider creating a duplicate of an intelligent voltage meter by copying the

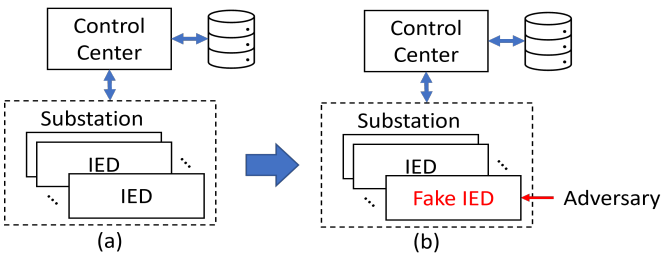

Fig. 1. Attack Scenario Overview. In (a), a substation operates normally with n IEDs sending data to a control center. In (b), an adversary has replaced one IED with a fake IED providing false data

nonvolatile memory of the device and then inserting malicious functionality into the programming of a fake device to send incorrect measurements to the control center. An overview of this is shown in Fig. 1. This duplicate IED contains the original device's private RSA keys, which would allow the fake device to pass the basic authentication steps that the control center would request from the IED during standard TLS mutual authentication. The control center would then issue a control action based on the measurements, the result of which could cause a disruption in power delivery within the grid, depending on the severity of the attack and the affected equipment [4].

In our scenario there are two substations of interest within a large power grid: Substation 1, which contains a single generator, and Substation N, which contains a complex load and an intelligent voltage meter. This is shown in Fig. 2. If the voltage meter reports a voltage below the nominal voltage, the control center would command the generator at Substation 1 to increase its reactive power output, thus increasing the voltage at Substation N. If the measurement reported at Substation N is sufficiently low, the voltage correction performed by the control center would cause an over-voltage situation at Substation N, possibly causing an operational violation and potential response of overvoltage relays [1]. This simple example can be generalized to a system with many substations, as the lone wolf with access to one substation will presumably have access to multiple substations.

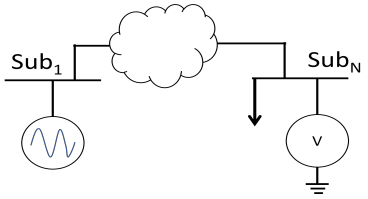

Fig. 2. Example Power System Modeled by the Prototype

This type of attack is known in the literature as a spoofing attack, so called because a fake device spoofs the behavior of a legitimate one [4]. The spoofing attack can be considered a subclass of a false data injection attack. A false data injection attack accomplishes the goal of providing a fake data stream with malicious intent, but may be performed through various methods such as interception of data packets during transmission across the network.

IV. PUF Based Authentication Protocol

In this section we develop our PUF based authentication protocol. We start by describing a archetypal authentication protocol which does not utilize a PUF. We then will show how a few extra steps can be easily inserted into the base protocol to provide additional protection based on PUF usage.

A. Base Case

The base case has no PUF, and is typical of what may be seen for typical authentication practices. During normal operation, the control center periodically authenticates all devices within the substation, including the IED (voltage meter). This authentication occurs every time the connection between the control center and the device is interrupted, and also at any other interval specified by the control center's policy. The authentication step utilizes RSA to establish a secure TLS channel (see Section II-B) between the control center and the IED to send and receive commands and measurements. The authentication protocol is shown in Fig. 3. Note that public keys are assumed to have already been distributed in an earlier secure initialization step [9]. In the case with no PUF, only the shown steps are performed:

1) Control center sends to the IED a nonce N_c encrypted with the IED's RSA public key $K_{d_{pub}}$.
2) IED decrypts the nonce received from the control center using the IED's private key $K_{d_{priv}}$.
3) The IED sends back the control center's nonce N_c and a new nonce generated by the IED, N_d, back to the control center, encrypted with the control center's public key $K_{c_{pub}}$.
4) The control center decrypts the message using the control center's private key $K_{c_{priv}}$ and verifies that the returned nonce N_c received matches what was sent.
5) Control center generates and sends an AES session key K_S, encrypted with the IED's public key $K_{d_{pub}}$, to the device along with the nonce N_d generated by the device.
6) The IED once again decrypts using $K_{d_{priv}}$ and verifies that the N_d received matches the one originally sent. If it does, a secure channel is successfully established using session key K_S received from the control center.

Once a TLS channel is established, the voltage meter receives commands and sends measurements to the control center as usual using K_S as a symmetric key to encrypt and decrypt using, typically, AES [7]. If any step in the protocol fails, communication with the device is terminated and the control center is alerted.

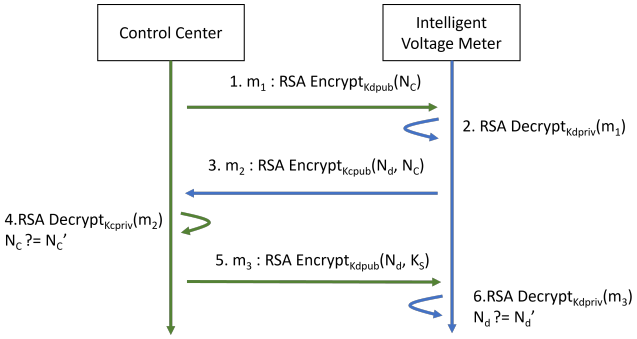

Fig. 3. Standard Mutual Authentication Protocol

B. PUF Enrollment

Before we can explain how the PUF authentication protocol works, we need to discuss the procedure used to enroll the IED's PUF into the control center's database. The protocol we use in this paper is shown in Fig.5. These steps may be performed in a secure facility prior to installation, or may

occur in a substation over TLS communications. Note that we use an SRAM PUF [13][14]. The enrollment steps are shown in Fig. 5 and work as follows:

1) The control center generates two random 128-bit numbers, one of which will act as an AES key and one which will act as a counter.
2) The IED sends a device ID to the control center, used for database management.
3) The control center sends a unique AES key and random counter value CTR to the IED.
4) The IED utilizes the SRAM PUF to store two secrets, the AES key and counter, in an encoded format. In Fig. 4 (a) this is performed with, for instance, an index '0' with the AES key as the secret value.

5a) The IED decodes the stored AES key and counter with the SRAM PUF. The decoding is shown in Fig. 4 (b). The counter is then encrypted via AES with the AES key, creating m_1, which is then transmitted to the control center.

5b) The IED increments the counter, encrypts the new counter value with the PUF and stores the encrypted counter (refer to Fig. 4 (a)), overwriting the previous encrypted counter.

6a) The control center encrypts CTR with the AES key from step 3 and verifies the result matches m_1.

6b) The control center increments the counter CTR.

7) The control center sends a message to the IED informing of successful enrollment.

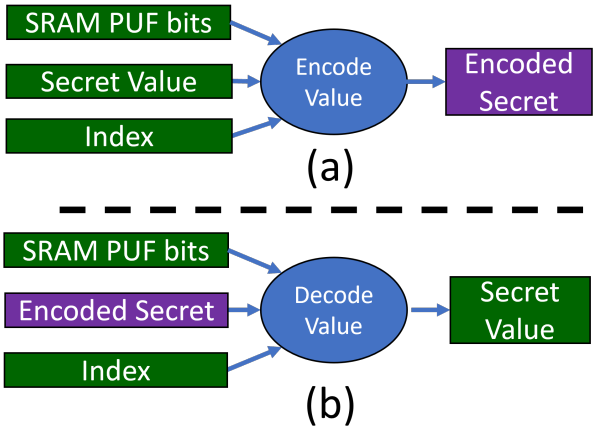

Fig. 4. (a) The PUF encodes a secret value, such as a key, producing an encoded secret value (b). The PUF decodes the encoded secret, reproducing the original secret value. Note that two chips with the same encoded secret and index will produce different secret values.

C. PUF Enhanced Protocol

With the enrollment procedure from Section IV-B completed, the PUF can be used for authentication. When the PUF device performs authentication, the six steps from Fig. 3 are performed first, and then the additional steps shown in Fig. 6 are performed. The additional steps are as follows:

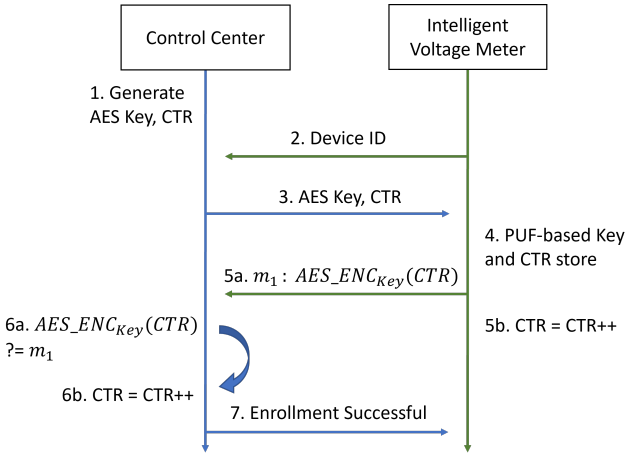

Fig. 5. Enroll Protocol for SRAM PUF

1) After the TLS channel is established, the control center sends a PUF authentication request to the device.

2a) The IED retrieves the stored AES key and counter value through the SRAM PUF (refer to Fig. 4 (b)). The IED then encrypts the counter via 128-bit AES with the AES key, producing m_1, which is sent to the control center.

2b) The IED increments the counter and stores the new counter value via the SRAM PUF (refer to Fig. 4 (a)).

3a) The control center encrypts its local copy of the counter with the IED's stored AES key and verifies the produced encrypted counter matches m_1 received fromt he IED. If the counter matches, authentication succeeds.

3b) The control center increments the counter stored locally.

If these steps succeed, communication continues with session key K_S. If any step fails, communication with the device is terminated and the control center is alerted.

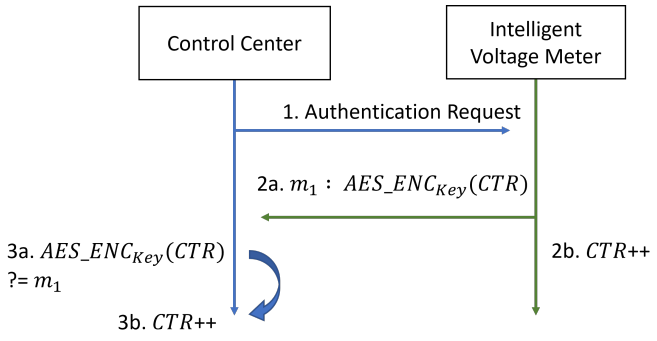

Fig. 6. PUF Based Authentication Protocol

V. Prototype System

Within a power grid, there are two interconnected pieces to consider: the power system, which consists of all of the actual electrical power flows, and the communication system, which consists of all of the computers and communication equipment between the control center and the substations. We built a prototype that models the power system shown in Fig. 2. This sys-

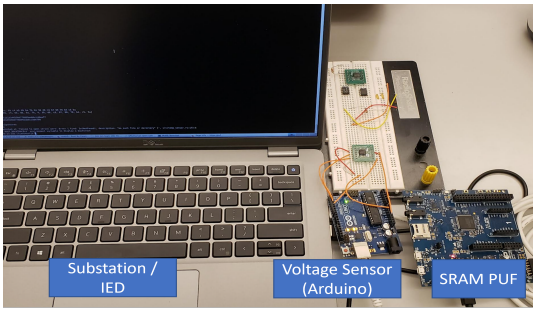

Fig. 7. Physical Configuration of the Prototype System

tem is modeled via a Python-based simulator running across two different computers, with each computer representing a single substation. Substation 1 (denoted "Sub_1") contains a single generator, and Substation N (denoted "Sub_N") contains a complex load and an intelligent voltage meter.

The computers representing Substation 1 and Substation N simulate their corresponding generator and load, respectively. The computer for Substation 1 also serves as the control center, meaning it receives measurements from the voltage meter at Substation N. The computer for Substation N simulates the voltage meter and is connected to an NXP LPC55S69 microcontroller on an LPCXpresso55S69 development board. The LPC55S69 microcontroller [14] implements an SRAM PUF [10], which we utilize to securely store an AES key and a counter value. The PUF attached to the computer at Substation N is used to demonstrate our PUF-based two-factor authentication protocol.

A. Implementation Details

We physically implemented the system as shown in Fig. 7. In the physical build we have a laptop running a Fedora Linux OS, acting as the IED and Substation N. The laptop IED is connected to an Arduino reading voltages, and the NXP LPC55S69 is on the LPCXpresso55S69 development board. The NXP LPC55S69 is used to provide PUF functionality. The microcontroller utilizes an SRAM PUF to store an AES key in an encoded format that can only be unencoded through the SRAM PUF. Once the AES key is encoded and stored on the board, the key has no external accessible interface, and can only be directly routed to the key input of a built-in AES module on the chip. We utilized the secure regions of flash memory built into the microprocessor to maintain storage of a representation of the counter needed in the protocol described in Section IV-C. We implemented the PUF protocol described above in the Rust programming language. Rust was chosen for usage due to its memory safety features and portability to various platforms [15].

VI. Experimental Results

Our prototype system is tested on two architectures: (i) a base case with no PUF, and (ii) a case that is protected by two-factor authentication with a PUF. For each case we demonstrate normal operation and an attempt at a spoofing attack. In each experiment, we will show a corresponding simulation of the effects on the voltage at Substation N.

A. Base Case, Normal Operation

An overview of our experiment is shown in Fig. 1. The first case has no PUF. During normal operation, the voltage meter at Substation N and the control center are mutually authenticated using the first six steps of the experimental protocol from the previous section, establishing a secure TLS channel with AES session key K_S. The voltage meter then sends valid measurements to the control center, and the control center takes no action.

B. Base Case, Spoofing Attempt

To perform the spoofing attack, an attacker copies the nonvolatile memory of a substation voltage meter and extracts the voltage meter's private RSA keys. The attacker loads these keys onto a malicious device which will send false measurements. When the legitimate device is replaced with the malicious device the control center will force re-authentication, using the protocol discussed in the Section IV. The malicious device will succeed at authenticating since it has the correct RSA keys. Now the attacker's malicious device sends false measurements to the control center indicating below nominal bus voltages at Substation N. The control center responds by instigating an increase of reactive power output at Substation 1, increasing the actual voltage at Substation N. The voltage at Substation N then could exceed nominal levels by a margin sufficient for negative impacts. Thus, the attacker succeeds as a partial blackout occurs.

C. PUF Protected, Normal Operation

When the PUF is included in the substation operation, the voltage meter at Substation N and the control center are mutually authenticated using the full experimental protocol described in the previous section. Once authentication succeeds, a TLS channel with AES session key K_S is established, and the voltage meter periodically sends measurements to the control center over this channel. Because there is no attack at this stage, the measurements reflect the correct nominal voltage, so the control center takes no action.

D. PUF Protected, Spoofing Attempt

Now the adversary attempts the same spoofing attack as discussed in Section VI-B. The adversary copies all of the memory contents including the RSA keys from the original voltage meter and places the memory contents on a malicious device intended to spoof the voltage meter. When the attacker disconnects the original IED and reconnects the spoofed device, the control center will force re-authentication following the full experimental protocol. The first six steps will succeed, since the spoofed device contains the original device's private key. However, because the adversary cannot clone the PUF of the device, the attack fails during Step 3a of the PUF enhanced protocol (Section IV-C). The AES key needed to encrypt the counter is inherent to the construction

of the PUF and cannot be extracted via copying nonvolatile memory or through any interface to the PUF device after initial installation. The control center is alerted of a potentially fake IED during authentication when the received encrypted counter is incorrect, allowing the control center to disregard measurements as erroneous until appropriate investigation into the device occurs.

We attempted spoofing by swapping between multiple PUFs. We enrolled one NXP microcontroller with a chosen counter and AES key derived from a true random number generator (TRNG) [16]. We then copied the encoded counter and AES key onto three other NXP microcontrollers, all of which have different inherent PUF values. The plaintext counter and AES key were retrieved from the encoded versions during normal operation via a decoding relying on the values of the SRAM PUF, leading each PUF board to retrieve a different counter and key despite running identical programs on identical memory contents. The client machine acting as Substation N was then connected to each of the four PUFs and communication with Substation 1 was attempted. In each test, the exact same program was running, with the exact same RSA keys, and the same flash memory contents on each NXP microcontroller. As such, each of the four tests should have the exact same functionality from a logical perspective. However, due to the PUF functionality, only the originally enrolled PUF correctly decoded the counter and AES key, allowing for authentication to the server. The three swapped PUFs, representing a spoofed device, failed to authenticate despite running identical programs and containing identical memory contents at runtime.

E. Results

A summary of the results from the four experiments is shown in Fig. 8. As expected, an adversary is able to spoof a device using standard authentication techniques by cloning the contents of the accessible memory contents. The added protection of the SRAM PUF, however, prevents an adversary from spoofing and successfully stops a spoofing attack.

Experiment	Result
No PUF, No Attack	Normal Operation
No PUF, Spoofing Attack	Attack Succeeds
PUF Protected, No Attack	Normal Operation
PUF Protected, Spoofing Attack	Attack Fails

Fig. 8. Experiment Summary

VII. Discussions and Conclusion

As described in the previous sections, a new two-factor authentication protocol is proposed, which successfully prevents a lone-wolf insider with the ability to clone RSA keys from being able to perform a spoofing attack with a fake IED. This protocol utilizes PUFs, which have a low cost to implement and low computational overhead for a significant increase in security [10]. The security of this protocol relies on the unique fingerprint-like functionality of PUFs, which cannot be predicted even by an adversary with direct access to the output of the PUF for an extended period of time [10][11]. This makes the protocol a natural candidate for smart grid devices, since these devices have a need for low computational overhead and an even greater need for any amount of security improvement possible.

References

[1] A. J. Wood, *Power generation, operation, and control /*, 3rd ed., 2014.

[2] K. Zetter, "Inside the cunning, unprecedented hack of ukraine's power grid," Mar 2016. [Online]. Available: https://www.wired.com/2016/03/inside-cunning-unprecedented-hack-ukraines-power-grid/

[3] W. Turton and K. Mehrotra, "Hackers breached colonial pipeline using compromised password," Jun 2021. [Online]. Available: https://www.bloomberg.com/news/articles/2021-06-04/hackers-breached-colonial-pipeline-using-compromised-password

[4] V. Chukwuka, Y.-C. Chen, S. Grijalva, and V. Mooney, "Bad data injection attack propagation in cyber-physical power delivery systems," in *2018 Clemson University Power Systems Conference (PSC)*. IEEE, 2018, pp. 1–8.

[5] G. Liang, S. R. Weller, J. Zhao, F. Luo, and Z. Y. Dong, "The 2015 Ukraine blackout: Implications for false data injection attacks," *IEEE transactions on power systems*, vol. 32, no. 4, pp. 3317–3318, 2017.

[6] Y. Liu, P. Ning, and M. Reiter, "False data injection attacks against state estimation in electric power grids," in *Proceedings of the 16th ACM conference on computer and communications security*, ser. CCS '09. ACM, 2009, pp. 21–32.

[7] J. Katz and Y. Lindell, *Introduction to modern cryptography*. CRC Press/Taylor & Francis, 2015.

[8] J. Kurose and K. Ross, *Computer networking : a top-down approach /*, seventh edition.. ed., 2016.

[9] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3," RFC 8446, Aug. 2018. [Online]. Available: https://www.rfc-editor.org/info/rfc8446

[10] R. Maes, *Physically Unclonable Functions Constructions, Properties and Applications*, 1st ed. Springer, 2013.

[11] P. Mall, R. Amin, A. K. Das, M. T. Leung, and K.-K. R. Choo, "Puf-based authentication and key agreement protocols for iot, wsns and smart grids: A comprehensive survey," *IEEE Internet of Things Journal*, pp. 1–1, 2022.

[12] U. Rührmair, J. Sölter, F. Sehnke, X. Xu, A. Mahmoud, V. Stoyanova, G. Dror, J. Schmidhuber, W. Burleson, and S. Devadas, "Puf modeling attacks on simulated and silicon data," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 11, pp. 1876–1891, 2013.

[13] "Quiddikey - Intrinsic ID: Home of PUF Technology," Feb 2022. [Online]. Available: https://www.intrinsic-id.com/products/quiddikey/

[14] NXP, "LPC55S6x datasheet rev. 2.3," August 6, 2021.

[15] N. D. Matsakis and F. S. Klock, "The rust language," in *Proceedings of the 2014 ACM SIGAda Annual Conference on High Integrity Language Technology*, ser. HILT '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 103–104. [Online]. Available: https://doi.org/10.1145/2663171.2663188

[16] M. Turan, E. Barker, J. Kelsey, K. McKay, M. Baish, and M. Boyle, "Nist special publication 800-90b: Recommendation for the entropy sources used for random bit generation," *US Department of Commerce, National Institute of Standards and Technology: Gaithersburg, MD, USA*, 2018.