PUF Security: Reviewing The Validity of Spoofing Attack Against Safe is the New Smart

Karim Lounis
karim.lounis@queensu.ca

School of Computing, Queen's University,
Kingston, ON, Canada.

***Abstract*—Due to the heterogeneity and the particular security requirements of IoT (Internet of Things), developing secure, low-cost, and lightweight authentication protocols has become a serious challenge. This has excited the research community to design and develop new authentication protocols that meet IoT requirements. An interesting hardware technology, called PUFs (Physical Unclonable Functions), has been the subject of many subsequent publications on lightweight, low-cost, and secure-by-design authentication protocols for the past six years. In 2020, a lightweight PUF-based authenticated key-exchange (AKE) scheme was proposed. The scheme claimed to provide mutual authentication and key establishment. The protocol was demonstrated to be vulnerable to a spoofing attack, where an attacker is able to compromise the authentication claims that are made during the execution of the protocol. Recently, some researchers have argued the validity of the attack due to a misunderstanding of security protocol specification principles. In this paper, we show how the authentication claim as well as the key-establishment claim of the authentication protocol can be compromised by spoofing the server and fooling the meter.**

***Index Terms*—PUF-based authentication protocol, PUF-based protocol security, PUF attacks, security protocol claims.**

I. Introduction

There has been a remarkable attraction and convergence from the research community and the industry to adopt PUFs (Physical Unclonable Functions) as a prominent physical security technology. Important industrial cores, such as NXP, Microsemi, Intel, and Xilinx, have already implemented the technology to develop secure integrated circuits. In the meantime, researchers have turned their attention to PUF technology to develop lightweight and secure-by-design authentication protocols for IoT applications [2].

In 2020, a PUF-based authenticated key-exchange scheme was proposed [3]. The scheme claimed to provide mutual authentication and key establishment. Recently, we have claimed that the protocol was vulnerable to a spoofing attack in the Ph.D. thesis [2]. This attack aims to compromise the authentication claims (authentication and key exchange) that are made during the execution of the authentication protocol.

Lately, some researchers have "claimed" the invalidity of the spoofing attack [1] that was presented in [2]. In this paper, we prove for a second time the validity of the attack by compromising the authentication claim as well as the key-establishment claim of the authentication protocol.

Identify applicable funding agency here. If none, delete this.

The remainder of the paper is organized as follows. In Section II we briefly provide a background on security protocol specification principles that are needed to better understand the attacks and prevent any misunderstanding. In Section III, we discuss the protocol and present the attack on the protocol. We conclude the paper in Section IV.

II. Background

In order to better understand the attack that we present in the next section on the authentication protocol of [3], in the next paragraph, we briefly discuss two major concepts: (1) MSC (Message Sequence Chart) and (2) Security protocol claims.

An MSC (Message Sequence Chart) is a graphical language for the description of the interaction between different components of a system [4]. This language is standardized by the ITU (International Telecommunication Union). It is widely used in describing the sequence of events, including messages, in security protocols. Every object and symbol in an MSC has a proper semantic. For example, a condition, denoted by a hexagon, is used to express that the system has entered a certain state. Therefore, MSCs are not simple flowcharts, but are standardized flowcharts.

In operational semantics of security protocols [5], a claim is an event where an agent, e.g., an authenticating party, reaches a particular state in the protocol and assumes that a certain security property, e.g., key secrecy, is satisfied. If a particular claim is compromised by an attack, then the concerned agent would assume that the claim is true, where in reality it is not the case. In such scenario, the protocol is considered compromised and vulnerable to that attack.

III. Smashing the Safe is the New Smart

In this section, we briefly present the protocol and demonstrate how an attacker can compromise the authentication claim as well as the key-establishment claim.

A. Load Modification-Resistant Smart Meter Protocol

Boyapally et al., [3] proposed a PUF-based authentication protocol for smart meters in the context of smart grid applications. The protocol allows a smart meter to be authenticated to a server using PUFs. The protocol adopts the DAPUF (Double Arbiter-PUF) augmented with a linear feedback shift register

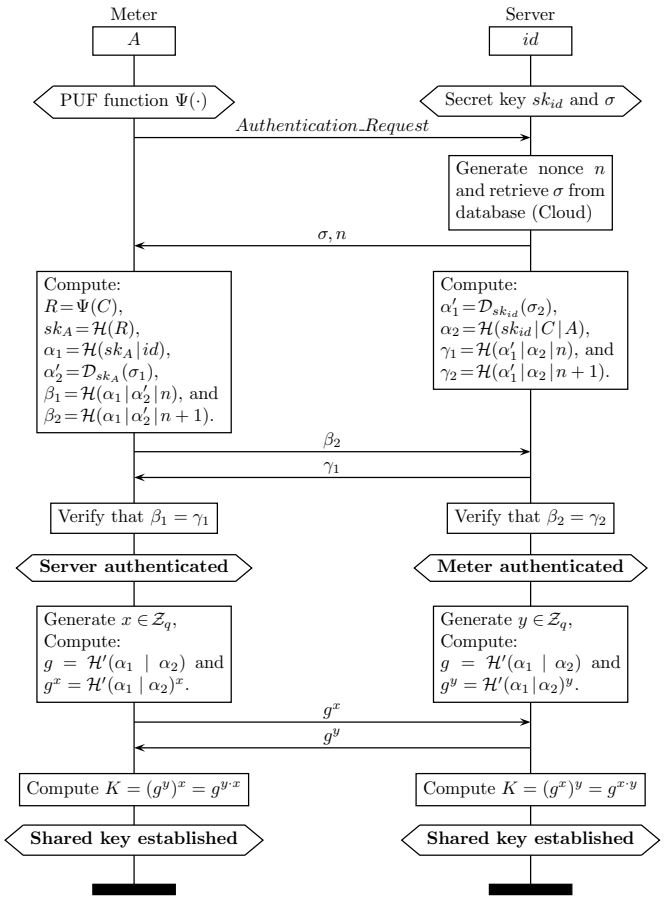

Fig. 1. Load modification-resistant smart meter authentication protocol by Boyapally et al., [3]. This authentication protocol allows a PUF-enabled smartmeter to authenticate to a server in the context of smart grid applications.

(LSFR) module to take a 64-bit challenge and outputs a 128-bit response (which makes it harder to guess). The authors showed that it is resilient to man-in-the-middle attacks as well as PUF-modeling attacks. They implemented the protocol over the Digilent Nexys Artix-7 FPGA-board in a smart meter. The protocol operates symmetric encryption (using AES) along with some asymmetric computations, such as bilinear pairings, to encrypt the response of the smart meters and store it securely on the server. Also, the responses of the smart meter are never exchanged in plaintext, which would prevent modeling attacks.

During the authentication of the smart meter to the server, the smart meter initiates the authentication by sending an authentication request. Based on the source of the request, the server retrieves from the database a cryptographic information, denoted by $\sigma = (C, \sigma_1, \sigma_2)$. This information contains the challenge C as well as two values, σ_1 and σ_2. The value of σ_1 was generated during the enrolment phase as $\sigma_1 = \mathcal{E}_{sk_A}(\alpha_2)$, where $sk_A = \mathcal{H}(\Psi(C))$ is the secret key of the smart meter, $\alpha_2 = \mathcal{H}(sk_{id}\,|\,C\,|\,A)$, A the identity of the smart meter, sk_{id} is the secret key of the server, $\Psi(\cdot)$ is the smart meter PUF function, and $\mathcal{H}(\cdot)$ is a hash function. The server sends to the meter the value of σ along with a random nonce n. The meter receives σ and n and uses the challenge C contained in σ to computes the response $R = \Psi(C)$. It also computes the values: $sk_A = H(R)$, $\alpha_1 = \mathcal{H}(sk_A \mid id)$, $\alpha'_2 = \mathcal{D}_{sk_A}(\sigma_1)$, $\beta_1 = \mathcal{H}(\alpha_1 \mid \alpha'_2 \mid n)$, and $\beta_2 = \mathcal{H}(\alpha_1 \mid \alpha'_2 \mid n+1)$. The meter then sends to the server the value β_2 to prove its identity. In the mean time, the server computes the values: $\alpha'_1 = \mathcal{D}_{sk_A}(\sigma_2)$, $\alpha_2 = \mathcal{H}(sk_A \mid C \mid A)$, $\gamma_1 = \mathcal{H}(\alpha'_1 \mid \alpha_2 \mid n)$, and $\beta_2 = \mathcal{H}(\alpha'_1 \mid \alpha_2 \mid n+1)$. Upon receiving the value of β_2 from the meter, the server sends the value of γ_1 to the meter. The meter checks whether $\beta_1 = \gamma_1$ to *claim* that the server is authenticated. Similarly, the server checks whether $\beta_2 = \gamma_2$ to *claim* that the meter is authenticated. These two authentication claims take place after an acceptation messages is sent by the meter to the server, and the latter replied back with an acknowledgement.

Next, both authenticating parties start the key establishment phase which is based on Diffie-Hellman Key Exchange Protocol. Both parties first compute the value $g = \mathcal{H}'(\alpha_1 \mid \alpha_2)$. The meter and the server generate the value $x \in \mathcal{Z}_q$ and $y \in \mathcal{Z}_q$, respectively, and send to each other the value of $g^x = \mathcal{H}'(\alpha_1 \mid \alpha_2)^x$ and $g^y = \mathcal{H}'(\alpha_1 \mid \alpha_2)^y$. The meter receives the value of $g^y = \mathcal{H}'(\alpha_1 \mid \alpha_2)^y$ and raise it to the power of x to compute the session key $K = \mathcal{H}'(\alpha_1 \mid \alpha_2)^{y\cdot x}$. Similarly, the server receives the value $g^x = \mathcal{H}'(\alpha_1 \mid \alpha_2)^x$ and raise it to the power of y and compute the key $K = \mathcal{H}'(\alpha_1 \mid \alpha_2)^{x\cdot y}$. At this point both authenticating parties *claim* that the shared key has been established. The authentication and key establishment of the protocol are illustrated in the MSC of Fig. 1.

B. Compromising the Protocol

Although the authentication protocol seems to perform an authentication of a smart meter to a server and establish a shared key using Diffie-Hellman key establishment protocol, we have found that the authentication claim as well as the key establishment claim can be easily breached and the server can be spoofed. We present the attack in the next paragraph.

To generate the attack, upon receiving an authentication request from the meter, the attacker impersonates the server and sends the value σ and $n-1$, where σ is the fixed cryptographic information intercepted from a previous authentication and $n-1$ a nonce. The smart meter performs the computations shown in the MSC of Fig. 2 and sends the value $\beta_2 = \mathcal{H}(\alpha_1 \mid \alpha'_2 \mid (n-1)+1)$ to the attacker. The attacker receives the value of β_2, saves it in a variable w (i.e., $w = \beta_2$), then drops the authentication. Upon a second authentication attempt from the smart meter, the attacker sends to the smart meter the value σ and n. The smart meter performs the authentication computations and replies by sending the value $\beta_2 = \mathcal{H}(\alpha_1\,|\,\alpha'_2\,|\,n+1)$. The attacker sends the value of w to the smart meter. The smart meter checks the value of w with the value of $\beta_1 = \mathcal{H}(\alpha_1\,|\,\alpha'_2\,|\,n)$ and finds that they are equal, which authenticates the attacker as the legitimate server. This compromises the authentication claim at the meter (i.e., **Server authenticated**). Furthermore, the attacker can proceed through the key establishment phase. The meter sends the value of $g^x = \mathcal{H}'(\alpha_1 \mid \alpha_2)^x$ to the attacker, which sends it back to the meter. The meter foolishly computes the shared key K

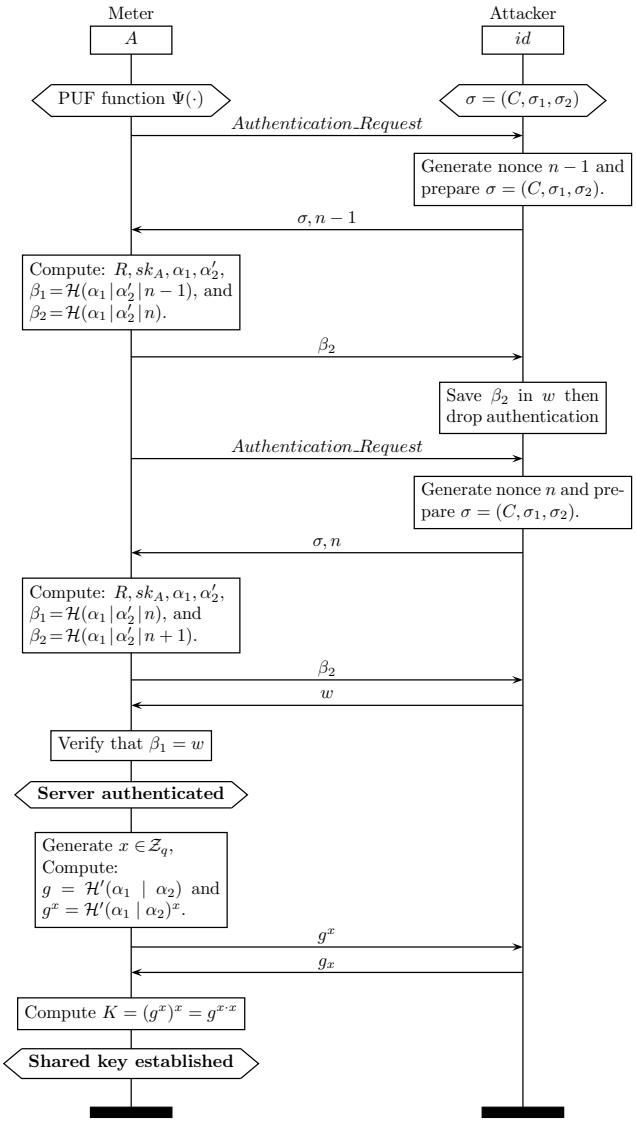

Fig. 2. A spoofing attack scenario on Boyapally et al., authentication protocol [3]. The attack allows an attacker to impersonate a server and perform a successful authentication (i.e., fool the meter into authenticating the attacker as a server). The attack also compromise the claim of key establishment at the meter (i.e., the meter is deceived to believe that that the shared key has been successfully established).

by raising what it received by x, i.e., $(\mathcal{H}'(\alpha_1 \mid \alpha_2)^x)^x$. At this stage, the meter claims that the key has been successfully computed. This compromises the key establishment claim at the meter (i.e., **Shared key established**) as the latter believes that same key has been established on both sides. This attack is illustrated in the MSC of Fig. 2.

Now we can claim that a "fake session" was successfully initiated. The following operations, which are not supposed to occur, may happen due to this attack:

- The meter may start sending encrypted information to the attacker. Although, this may sound useless, the meter might be implemented in such a way so that it deletes any information that was sent and overwrites it with fresh one (e.g., due to storage-capacity constraint).
- Log multiple events of successful authentications and key establishments that the legitimate server cannot deny.
- Depending on the relationship between the CDH (Computational Diffie-Hellman) assumption and the DL (Discrete Logarithm) assumption, if computing the discrete logarithm (base g) in $\mathbb{Z}_q$ is easy, then the CDH problem could be solved and the secret key $g^{x \cdot x}$ may be cracked.

IV. Conclusion

In this paper, we have in fact validated the claim that the PUF-based authentication protocol proposed by Boyapally et al., in [3] is vulnerable to spoofing through a replay attack. We have demonstrated how an attacker, using a previous fake session, can impersonate the server and smash-compromise both the authentication claim and the key establishment claim.

To improve the security of the protocol, we recommend the following directions: (1) The nonces that are used by both parties are dependent (e.g., Server uses n and Meter uses $n + 1$). Indeed, there is no randomness introduced by the Meter (i.e., no actual nonce). This consequently results in the non-uniqueness of authentication sessions, and hence the possibility of generating replay attacks. (2) The authentication proofs (i.e., β_2 and γ_1) used by both authenticating parties are strongly related. This would allow an attacker to easily find a way to construct forged authentication proofs with minimum effort. For instance, in this protocol, an attacker constructs authentication proofs by interrogating the verifier (i.e., Meter).

Finally, like most protocols, if not all, this broken authentication protocol is vulnerable to race condition-based attacks that we have introduced and discussed in [6]–[9].

References

[1] H. Boyapally, P. Mathew, S. Patranabis, U. Chatterjee, and D. Mukhopadhyay. "On the Validity of Spoofing Attack Against Safe is the New Smart", Cryptology ePrint Archive, pp. 1-2, article 395, 2021.

[2] K. Lounis. "Security of Wireless Short-Range Technologies and an Authentication Protocol for IoT", Ph.D. Thesis, School of Computing, Queen's University, pp. 1-324, 2020.

[3] H. Boyapally, P. Mathew, S. Patranabis, U. Chatterjee, U. Agrawal, M. Maheshwari, S. Dey, and D. Mukhopadhyay. "Safe Is The New Smart: PUF-based Authentication for Load Modification-Resistant Smart Meters", IEEE Transactions on Dependable and Secure Computing, pp. 1-18, IEEE, 2020.

[4] S. Mauw and V. Bos. "Drawing Message Sequence Charts with LaTeX", in TUGBoat Journal, vol. 22, no. 1-2, pp. 87-92, 2001.

[5] C. Cremers and S. Mauw. "Operational Semantics and Verification of Security Protocols", Information Security and Cryptography, Springer, DOI 10.1007/978-3-540-78636-8_1, 2012.

[6] K. Lounis and M. Zulkernine. Attacks and Defenses in Short-Range Wireless Technologies for IoT. In IEEE Access, volume 8, pages 88892-88932, 2020.

[7] K. Lounis and M. Zulkernine. Exploiting Race Condition for Wi-Fi Denial of Service Attacks. In Proceedings of the 13th International Conference on Security of Information and Networks, pages 1-8, 2020.

[8] K. Lounis and M. Zulkernine. Bad-token: Denial of Service Attacks on WPA3. In volume 15 of ACM Proceedings of the 12th International Conference on Security of Information and Networks, pages 1-8, 2019.

[9] K. Lounis and M. Zulkernine. Connection Deprivation Attacks on WPA3. In volume 12026 of Proceedings of the 14th International Conference on Risks and Security of Internet and Systems, pages 164-176, 2019.