IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 4, 2021, accepted January 18, 2021, date of publication January 21, 2021, date of current version January 29, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3053307

Resilient Password Manager Using Physical Unclonable Functions

MOHAMMAD MOHAMMADINODOUSHAN, (Graduate Student Member, IEEE), BERTRAND CAMBOU, (Member, IEEE), CHRISTOPHER ROBERT PHILABAUM, AND NAN DUAN

School of Informatics, Computing and Cyber Systems, Northern Arizona University, Flagstaff, AZ 86011, USA

Corresponding author: Mohammad Mohammadinodoushan (mm3845@nau.edu)

ABSTRACT The offline dictionary attacks on the database of passwords (PW) or even hashed PW are damaging as a single server break-in leads to many compromised PWs. In this regard, using Physical Unclonable Functions (PUFs) to increase the security of PW manager systems has been recently proposed. Using PUFs allows replacing the hashed PW with PUF responses, which provide an additional hardware layer of security. In this way, even with accessing the database, an adversary should have physical control of the PUF to find the PWs. However, such a scheme cannot operate without a backup in case of catastrophic failure of the PUFs. The likelihood of a failure is low unless the opponent finds a way to destroy the PUF. The scheme used in this article includes a mechanism to make the system works consistently if the PUF fails, with redundant elements. In this method, two PUF outputs are saved in the database to register a user. In authentication, the first PUF output in the database is just checked. The second PUF output in the database is only checked in the exceptional cases when the first PUF does not work correctly; therefore, both false reject rates and latencies are not degraded. A PW manager node is implemented using a low-cost microcontroller, SRAM PUF, and nonvolatile SRAM. The nonvolatile SRAM is embedded in the PWM node circuit as a local database. Statistical tests on the applied commercial SRAM in this article show better PUF quality than those used in previous research. Also, to handle the error in PUF responses, only the stable SRAM cells are used. This article presents the first prototype of a resilient PW manager node with an embedded local database to the best of our knowledge.

INDEX TERMS Database, hardware implementation, physical unclonable function, resilient password manager node, SRAM.

I. INTRODUCTION

Password (PW) authentication is the most common method to access users to websites, devices, applications, or other services. PWs are usually compared with a golden copy stored in a database (DB) or lookup tables to authenticate an entity. The simplest way being used by PW managers (PWM) like [1], [2] is keeping the IDs and PWs in plaintext format in the central DB of the server.

Disclosing users' PWs is a common issue that poses substantial financial damages [3]–[6]. The main reason for this problem is that the most common methods used by current PWM systems, such as hashing, and salting are based on known and public algorithms. Also, it has been shown that users prefer common and weak PWs [7], [8]. Therefore, hackers can disclose the PWs when they break into a single server and compare DB information with common PW hashes. Moreover, compromising a user account at one service may compromise the users' accounts at other services since users often reuse the same PWs for other applications [9].

Another approach applied in previous studies is saving encrypted PW in DB using a cryptographic key. However, when disclosing the key, the PWs are disclosed as well. In [10], the encrypted PW was saved in the central DB, and the PW hash was kept in the PWM header. To authenticate the user, first, the PW was decrypted from the DB. Then, the decrypted PWs hash was compared with the value saved in the header. Using this method improves security slightly. Nevertheless, the technique utilized in [10] work is not usable in case of losing the encryption key.

Physical Unclonable Functions (PUF) creates fingerprints from the hardware component. The idea of utilizing PUFs to add a secure HW layer to the current PWM systems has been

The associate editor coordinating the review of this manuscript and approving it for publication was Ahmed Farouk.

recently proposed in [11]. The idea proposed in [11] is implemented HW [12] using commercial Static Random-Access Memory (SRAM). This method does not require saving PWs or PW hashes. In [12], the PUF responses are replaced by hashed PW in the central DB. Therefore, hackers need physical control of the PUF to find PWs in plaintext format even if they access the central DB, which is more complicated.

Despite PUF advantages, research has shown that many current PUFs are vulnerable to modeling attacks, such as approximation attacks [13] or Covariance Matrix Adaptation Evolution Strategy attacks [14]. In [13], two advanced approximation attacks could effectively model some multiplexer based and XOR Arbiter PUFs. In [14], a new machine learning attack using a divide-and-conquer approach is used to attack the XOR PUFs. Despite continuing efforts to improve the resilience of PUFs to advanced modeling attacks, other modeling attack methods could compromise the security of the latest proposed PUFs. For example, recently proposed machine learning attacks in [13], named logical approximation and global approximation, could successfully attack the recent multiplexer PUFs proposed in [15].

As a result, with access to the central DB and PUF model, adversaries can find the PWs of many users with effort. Also, a major problem with using PUF where the PUF stops work has not been considered in previous work [16]. Therefore, if the PUF breaks, the system, or a significant part of it will fail as well.

In this article, a backup plan for the event of PUF failure is considered. The central database used in [12] is replaced by a local DB (LDB) embedded in the PWM node circuit. SRAM PUF used in this article is of higher quality than the similar SRAMs used in previous works [12]. Also, the microcontroller (MCU) cost in this article is lower than the one used in [12]. Moreover, to decrease the error in PUF responses significantly, ternary PUF [17] is used in this study.

The remainder of this article is organized as follows: Section II reviews the PUF and PWM systems. Section III presents the proposed architecture, new methods, and SRAM PUF characterization. The prototype implementation and the results are given in section IV. Finally, section V provides the conclusion and future work.

II. BACKGROUND

This section describes relevant background information and discusses the SRAM PUF and PWM technologies.

A. SRAM PUF

Many forms of PUFs have been designed in the literature [18], [19]. Memory-based PUFs such as SRAM PUFs [20], [21], Dynamic RAM (DRAM) PUFs [22], Memristor PUFs [23], [24], and Magnetic RAM (MRAM) PUFs [25] are practical because it is possible to make them from memories already existing in many systems. Also, the required size for the PUF is insignificant compared to the whole memory size. So, the location of the PUF in the memory can be another secret information that increases security when a hacker gets access to the memory. SRAM PUFs were discovered based on the original SRAMs independently and concurrently by Holcomb *et al.* [20] and Guajardo *et al.* [26]. Generating PUFs from SRAM arrays is performed by subjecting the device to power-off power-on cycles. A significant number of the cells in the SRAM PUF are always either '0' or '1'. However, few numbers of SRAM cells have a weak preference or no particular preference at all [27]. These cells are called fuzzy cells in this article. PUFs using SRAM technology have already been tested and commercialized by several companies [28].

B. PWM

In this section, the idea of adding PUF to the current PWM systems is discussed.

1) CURRENT PWM SYSTEMS

The most reliable method used in current PWM systems is hashing, salting, and encryption of the PWs and saving them in the central DB. These methods are based on known and public algorithms. As previously mentioned, users use weak PWs [7], [8]. Therefore, the PW in plain text format can be easily disclosed by hackers since they compare their different PW hashes in the dictionary with DB content. By hacking the central DB of one single server, hackers can often cause substantial damages to the users; government organizations [29], [30]; commercial vendors such as Gmail [31], GitHub [2], Twitter [32], and others [33]. Several techniques are employed to improve PWs from the user side or client-side [34], [35]. However, they have not been a solution since users' response to restrictive PW selection rules is predictable due to their memory limitations [8], [36].

2) PWM SYSTEMS WITH PUFs

PUFs provide a physical entity, i.e., a unique 'fingerprint.' The key idea in [34] is to leverage the uniqueness and uniformity of PUF on the client-side to strengthen PWs and prevent attacks. The proposed PWM system in [34] is handled across both software and HW. Unlike hashing, which has a deterministic output for a given input, PUFs provide different outputs for the same input. In [37], the security of the central DB of PW hashes was improved using an HW security module at the authentication server to avoid PWs compromise. This method, which employs public-key encryption, is started by producing a pair of public/private keys. In the initialization step, the PW hash, the user ID, and the salt of all users are encrypted with the public key and saved. Using an HW module in [37] improves security. However, a brute-force attack can be applied by the adversary to recover the backup file information. Therefore, the PW security level in [37] would be the same as using the PW hashes.

The idea of using PUFs to improve PW protection was patented recently in [11], [38]–[40]. In the PWM system proposed in [11], the PUF is added as an HW security layer on the server-side. The scheme does not need any adjustment

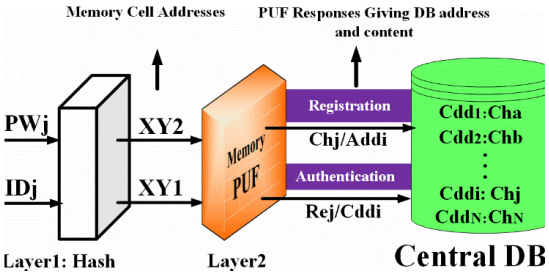

FIGURE 1. Adding the HW security layer to the PWM system using PUFs.

to the user interfaces or clients. This idea is schematically presented in FIGURE 1.

As shown in FIGURE 1, the ID and PW are supplied to known functions such as hashing to create two Message Digests (MD). The first MD is generated using the PW, and the second MD is generated using a combination ID and PW. MD1 and MD2 produce memory cell addresses (XY1 and XY2). These cell addresses are then used for extracting the PUF responses in the second hardware layer. PUF responses are used for creating both the DB address (Addi in registration and Cddi in authentication) and the content for saving in the corresponding address (Chj). In the authentication, new PUF responses create both DB address (Cddi) and content (Rej in authentication). If Cddi/Rej is matching Addi/Chj, the authentication is positive. The idea in [11] is partially implemented in [12], wherein the PUF is used to extract the DB content, not the DB address.

C. PROBLEM STATEMENT

The methods and systems outlined in [11], [12], [38], [39] are based on adding PUFs to PWM systems to enhance security. However, they suffer from two main problems. First, when the PUF fails, the whole system or a significant part of the system will fail as well. If the PUF stops working, the authentication of users will not be possible. One of the techniques proposed in this article is to find a backup plan for the event of PUF failure. The second problem emerges when the attacker accesses the central DB and the PUF. In this case, a considerable number of PWs in central DB will be disclosed.

III. METHODS AND PROPOSED ARCHITECTURE

A. CONTRIBUTIONS

This paper's main contributions (i.e., adding resiliency and using LDB) are discussed in subsections 1 and 2.

1) RESILIENCY

For solving the PUF failure problem discussed above, a recently proposed method in the filed patent [41] is implemented in this article. The scheme in [41] uses several PUFs and creates several DBs with different PUF responses. As shown in FIGURE 2, two PUF responses (iCh1j and Ch2j) are saved in DB for user registration. For authentication, we use just the first PUF response (Ch1j). The second

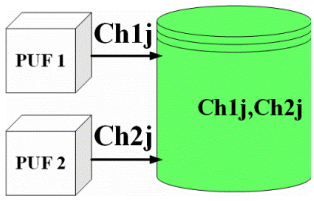

FIGURE 2. Using a database coming from two PUFs as a backup plan.

PUF response (Ch2j) is just checked in the exceptional cases when the first PUF stops working. Therefore, if one PUF fails, the system continues working using the second PUF response.

In this article, the responses generated from two parts of PUF (i.e., Ch1j and Ch2j) are saved in the DB to test the idea presented above while using one SRAM PUF HW. We did not use two SRAM PUF hardware. This is part of the planned future work.

2) LOCAL DB

As mentioned earlier, the central DB hacking attacks are highly devastating because a single server break-in can result in many compromised PWs. In this work, the centralized DB is replaced by LDB embedded in one PWM node circuit.

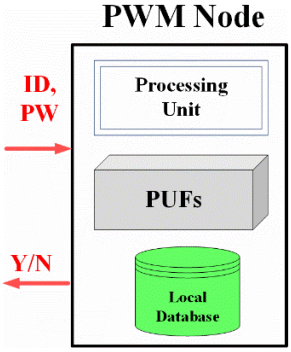

FIGURE 3. Schematic of PWM node with LDB.

B. PROPOSED ARCHITECTURE

The general schematic of each PWM node is shown in FIGURE 3. The detailed architecture of FIGURE 3, implemented in this article, is depicted in FIGURE 4. As can be seen, the whole system includes LDB, MCU, and SRAM PUF. The details of the LDB, MCU, and PUF are discussed in subsections 1, 2, and 3, respectively.

1) DB STRUCTURE

1-Mbit Serial (I2C) CYPRESS nonvolatile SRAM (NVSRAM) [42] is used to implement LDB in this article. The structure of the LDB is shown in FIGURE 5. Also, the content (PUF response) saved in the DB for each user is considered 16 bytes (16B) in this article. Therefore, 8192 users can be covered. For finding the LDB address related to each of 8192 users, a 13-bit address is needed.

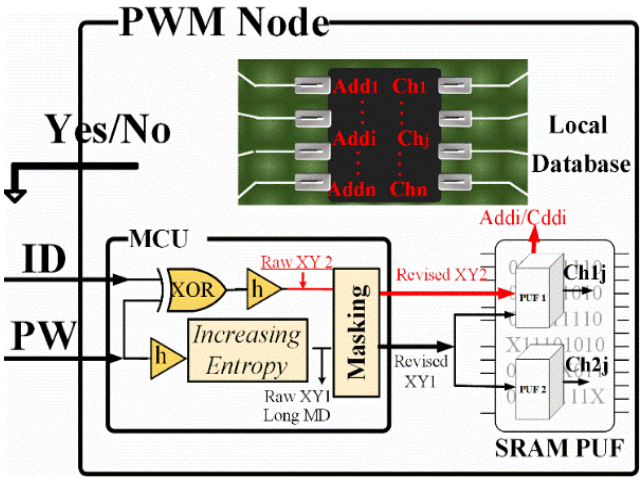

FIGURE 4. The overall architecture of the implemented PWM node.

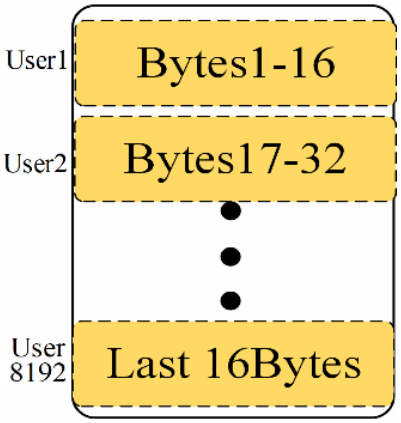

FIGURE 5. Covering 8192 users by NVSRAM.

2) PROTOCOL

Here, we discuss the protocol for registration and authentication, including error matching algorithm, Entropy Enhancement (IE) block, and masking block.

a: REGISTRATION

The steps for registration of a user are listed in TABLE 1.

b: AUTHENTICATION WITH ERROR MATCHING ALGORITHM

A typical authentication of the user is like the registration process. The previously saved PUF response (Chj) at Addi in LDB is compared with the fresh PUF responses (Cddi/Rej). If Cddi/Rej matches the reference Addi/Chj, the authentication is positive. The use of ternary PUFs can significantly help to mitigate potential mismatches between Addi and Cddi. However, just a 1-bit error occurring in Cddi can point to the wrong address and unrelated content in the LDB. This error in addressing can cause false rejection of a legitimate user. The error matching algorithm decreases the false rejection rate (FRR) when the authentication is not positive due to a 1-bit error in PUF response. The details of the error matching

TABLE 1. Algorithm of User Registration.

1	Hashing of PWA in PWM Node in MCU
2	Hash of PWA (MD) is extended to long MD to increase the entropy (in IE block)
3	Raw addresses (raw X1Y1) are calculated from long MD (in IE block)
4	Generation of Revised addresses (Revised XY1) with masking the fuzzy cells in raw XY1 (in masking block)
5	Generation of PUF response (Chj) from cells in revised XY1 (in PUF)
6	Concatenation of ID and PW (in MCU)
7	Hashing of (ID+PW) to create raw XY2 in MCU
8	Generation of Revised addresses (Revised XY2) with masking of Fuzzy cells in raw XY2 in masking
9	Generation of PUF response (Addi) from cells in revised XY2
10	Saving Chj at Addi (in LDB)

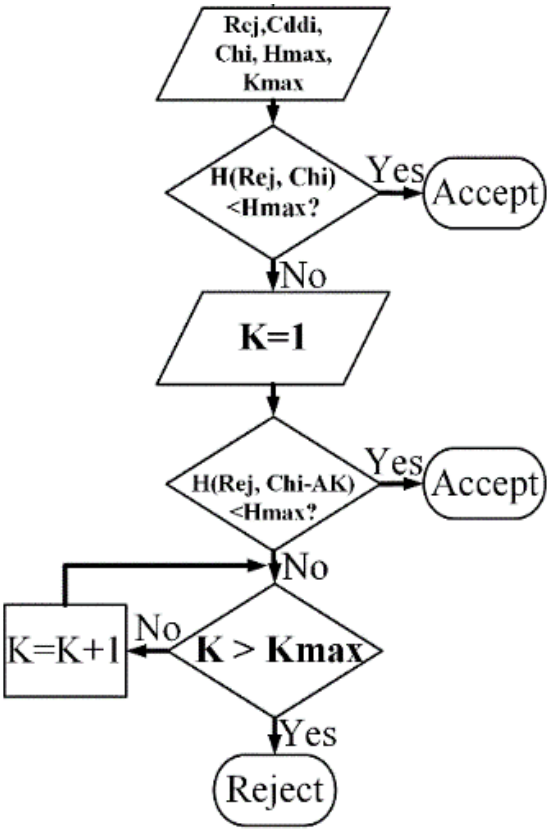

FIGURE 6. Algorithm to find the matching address in LDB [43].

algorithm are explained in [43] and are reviewed here. The flowchart of the error matching algorithm is presented in FIGURE 6. Hmax is the maximum acceptable hamming distance between Chj and Rej (H(Rej, Chj)). Hmax is considered 1 in this article to keep the authentication time in an acceptable range. Also, the probability of having more than 1-bit error in the designed PUF response is very low, as will be discussed in the SRAM PUF design section. If H(Rej, Chj) is greater than 1, all k contents (Chi-k) located in LDB at the addresses Cddi-k are examined. The addresses Cddi-k are in

the hamming distance of one with Cddi. The authentication is positive if H(Chi-k, Rej) is below Hmax. If H(Chi-k, Rej) is greater than 1, the process iterates to consider all Chi-k. Since a 128-bit response is considered for each user in this article, and the LDB size is 1Mbit, 13 bits is enough for addressing a specific LDB location. Therefore, Kmax is 13.

c: ENTROPY ENHANCEMENT

In the architecture shown in FIGURE 4, the Increasing Entropy (IE) block (similar to Block 1 in [12]) creates more extended MD out of the original MD to increase the entropy. The protocol presented is based on long MD conversion into raw addresses (Step 3 in TABLE 1). Raw addresses can point to fuzzy or non-fuzzy cells.

d: MASKING

In the masking block shown in FIGURE 4, if the raw address point to a fuzzy cell, the address is revised to the address of the next non-fuzzy cells (Step 4 in TABLE 1).

3) DESIGN OF SRAM PUF

The startup values of CY7C1021CV26, which is a 1-Mbit CYPRESS SRAM, [25], are examined for having the desired PUF quality. In this part, quality matrices of the SRAM PUF, including intra-PUF, inter-PUF, and uniformity are characterized.

a: ENROLLMENT OF TERNARY SRAM PUF

Here, the process of enrollment is defined for ternary PUFs. The enrollment objective is defining the unstable cells that should not be used due to their undesirable behavior. The enrollment process minimizes the error rate for future challenge-response pairs (CRPs) (intra-comparison) by ignoring the unstable cells. As mentioned previously, the method for designing SRAM PUF is signaling a series of power-on, power-off cycles. The responses of the PUF are not stable for few cells [27]. These unstable cells are called masked or fuzzy cells and denoted by an 'X' mark.

In the characterization (enrollment) phase, the SRAM PUF cells are read hundreds of times by repeating the power-off power-on cycles. In each reading cycle, the cells with different responses against their previous are specified as fuzzy cells. In this way, it is possible to recognize the cells that can produce stable '0' and '1' and remove those with the 'X' state. The higher number of reads is accompanied by finding more fuzzy cells, more stable responses, and lower error rates. The Mask data, which are the enrollment results, will identify the address of fuzzy or non-fuzzy cells.

The experiments are carried out on ten SRAM chips. Each chip was first read 1000 times, as designated for the enrollment. Each chip was read another 100 times, as designated for queries (i.e., the responses).

As shown in FIGURE 7, each enrollment size N analyzes all the bits that have changed their state in the previous N - 1 cycles. The percentage of fuzzy masked cells ('X') and stable ('0' + '1') cells are independently plotted against the enrollment size.

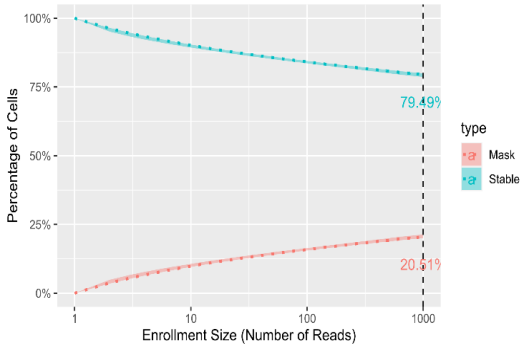

FIGURE 7. The percentage of masked and stable cells vs. the number of power cycles (enrollment size).

The results in FIGURE 7 are taken from the median of the ten individual chips. As can be seen, at one cycle, there is no other previous cycle so that every cell is labeled as stable by default. Over half the masked (fuzzy) cells are found after 10 cycles, whereas the remaining half is found after 1000 cycles. After 1000 cycles, 20.51% of cells are fuzzy, and the other 79.49% are stable.

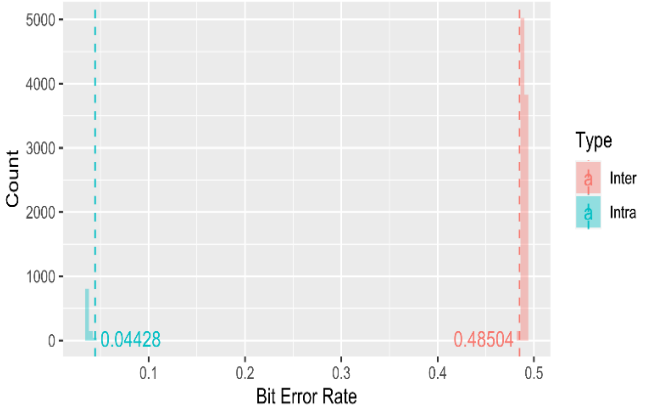

FIGURE 8. Intra- and inter-PUF without enrollment (1 cycle) – 10x 1Mbit SRAM chips.

b: INTRA- AND INTER-PUF COMPARISON

FIGURE 8 presents the results for the case that data were collected when no enrollment had occurred. The results shown in FIGURE 8 are comparable to the case of 1 cycle in FIGURE 7. This plot first compares 100 query reads of one chip against the first read of the same chip. This process comparison is done for all 10 chips and is labeled as the 'intra' in FIGURE 8. The plot also takes each chip 100 queries and compares it against every other chip single read. This comparison result is labeled as 'inter' in FIGURE 8. The results show that the median intra-PUF variation is 3.63%, while the maximum intra-PUF variation is 4.43%. The median 10 SRAMs inter-PUF variation is 48.9%, and the minimum inter-PUF variation is 48.5%. As can be seen, the device used

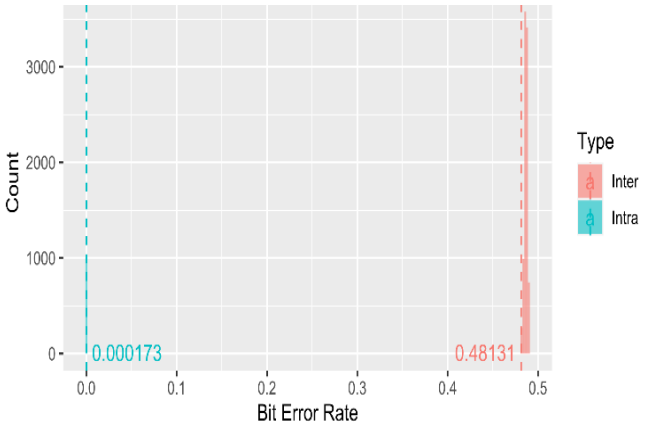

FIGURE 9. Intra- and inter-PUF with enrollment (1000 cycles) – 10x 1Mbit SRAM chips.

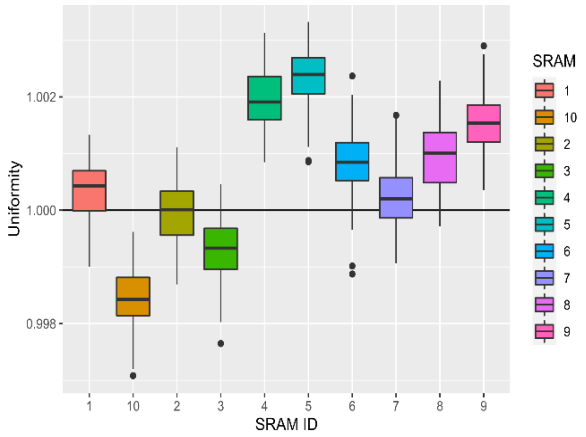

FIGURE 10. Uniformity index for 10 SRAM chips.

TABLE 2. A Uniformity of 10 SRAM PUF.

CHIP#	μ	σ	CHIP#	μ	σ
Chip1	1.0004	4.713e-04	Chip6	1.0008	5.888e-04
Chip2	1.0000	5.284e-04	Chip7	1.0002	5.743e-04
Chip3	0.9993	5.627e-04	Chip8	1.0010	5.745e-04
Chip4	1.0019	5.343e-04	Chip9	1.0015	4.998e-04
Chip5	1.0024	5.329e-04	Chip10	0.9985	4.947e-04

TABLE 3. Comparison of PUFs.

Inter(%)	*Intra(%)*	*Ref.#*	PUF TYPE
49	3.6	[44]	Crossover RO PUF
50.36	1.78	[45]	RO
50.1	5	[46]	ROPUF with a 32-bit challenge
45.83	5	[48]	MRAM
48.9	3.63	This work	SRAM (without enrollment)
48.7	0.00216	This work	SRAM (with enrollment)

as PUF has a desirable inter-PUF quality, but the intra-PUF bit error rate is high to use.

FIGURE 9 presents a similar procedure in FIGURE 8. Instead of using a single cycle for the enrollment, the intra- and inter-PUF comparisons are made against the full 1000 read enrollments. Any cell marked as the mask is not considered a part of the pool of error rates. As shown in FIGURE 9, the intra-chip error improved significantly, i.e., an average of 0.00216% with a maximum of 0.0173%. Also, the inter-PUF index got worse slightly, with an average of 48.7% and a minimum of 48.1%.

The enrollment with 1000 cycles could significantly decrease the intra-PUF error rate from 3.63% to 0.00216%. Therefore, the probability of more than 1-bit error in the ternary PUF response is 3.7×10^{-6} in Rej with 128 bits. The probability of having more than 1-bit error is less than 10^{-6} in the Cddi, which is13-bit.

c: UNIFORMITY

Uniformity is another useful metric for assessing the quality of a PUF. In this research, uniformity is defined as the ratio of '0's to '1's for each read from an SRAM. In other words, for a PUF to have an ideal uniformity, the uniformity index should be 1. The response of each of the 10 SRAM chips is analyzed when they are queried 100 times. FIGURE 10 illustrates the distribution of 100 uniformity indices for each chip as a boxplot. TABLE 2 shows the value of the mean and standard deviation of the distribution of the uniformity index. It has been observed that the uniformity mean values are very close to 1 for all the chips.

d: COMPARISON OF PUFs

The results of some of the previous publications on intra- and inter-PUF comparison of silicon PUFs are listed in Table 3. In [44], promising crossover Ring Oscillator (RO) PUF is presented. The crossover RO PUF proposed in [44] has much lower hardware overheads and better reliability than the previous RO PUFs [45], [46]. Moreover, our proposed crossover RO PUF can drastically mitigate the effect of the environment on PUF responses and generate more reliable responses. This crossover RO PUF is also resistant to side-channel attacks [44], [47].

The SRAM PUF responses, like other memory-based intrinsic PUF, can be affected by changing environment effects. Nevertheless, they can be implemented with memories available on most electronic devices. However, arbiter PUFs, as an example, need a different design and layout steps. Our results show the functional uniqueness (48.7%) between different SRAM PUF chips. Also, the intra-PUF error results show 0.00216%, which is more satisfactory than the PUFs reported in the previous studies [12], [48].

The quality of PUF designed with CY62256N, CYPRESS 32 kB SRAM (used in [12]), is compared with the SRAM PUF designed in this article. The bitmap for both models is presented in FIGURE 11 and FIGURE 12. We performed 1000 measurements on 10 chips from each model. The numbers show the percentage of the times the cell read as 1. So, 100% means being 1 in 100% of the reads. FIGURE 11 shows that there are repeating patterns in every 4 rows. The results show that reading 1 in the first 4 rows is 26.9% versus

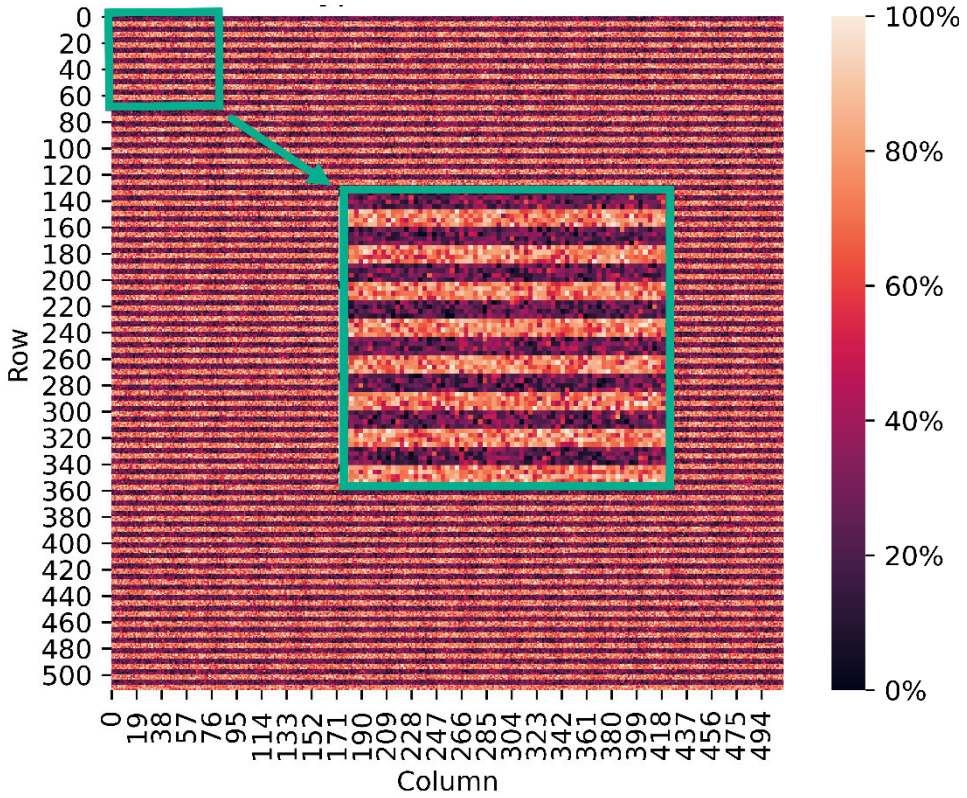

FIGURE 11. Bitmap from CY62256N, used in [12].

73.1% for the next 5 rows. This patent repeated through the whole array. The pattern in FIGURE 12 for the SRAM used in this article shows that the probability of reading 1 in the right half of the array (columns 512-1023) is higher than the left half of the array (columns 0-511). However, the measurements show that reading 1 in the left half is 56.4% versus 43.6% in the right half. Therefore, the analysis of the two models reveals that the model used in this article exhibit higher PUF quality.

IV. IMPLEMENTATION AND RESULTS

A. PROTOTYPE DESCRIPTION

The WiFire development board from Digilent, powered by microchip [40], is used to validate the protocol. MCU handles most of the tasks, including XOR, hash function, extending MD, raw address generation, masking, PUF response generation, and error matching algorithm. As shown in FIGURE 13, a custom PCB is designed to create a more compact and reliable HW package. This shield places on top of the MCU includes commercially available HWs such as SRAM PUF, NVSRAM, and other components for managing the PUF and NVSRAM Power and IO. The shield is mainly used for interface PUF and NVSRAM with the ChipKit WiFire MCU. For prototyping, a laptop is used to provide power to the MCU and shield components and handles UART communication for reading and verifying the data.

B. RESULTS

In the following, the results are explained step by step. It is of note that most of the registration and authentication steps and results are the same.

1) RECEIVING DATA

At the beginning of the program, the user selects to sign up (registration) or sign in (authentication). In both registration and authentication, the ID and PW are sent to the MCU. The MCU captures and echo every single received byte of ID to the terminal but hides the PW. In the next step, Hash (PW) and Hash (ID+PW) are calculated. The results shown in FIGURE 14 are obtained after entering 'mm3845' and 'Mohammad' as the 'Username' and 'Password.' As previously mentioned, MD is extended in the IE block. The IE block results are not shown in FIGURE 14, as similar to the one in [12].

2) GENERATING RAW ADDRESSES

FIGURE 15 shows the results of creating both LDB content and addresses (raw XY1 and raw XY2 in FIGURE 4). Raw XY1 includes 128 different addresses, and raw XY2 includes 13 different addresses.

3) GENERATING REVISED ADDRESSES BASED ON THE MASK

The raw addresses can point to both fuzzy and non-fuzzy cells. In the former case, the cell is assumed as a valid cell, and

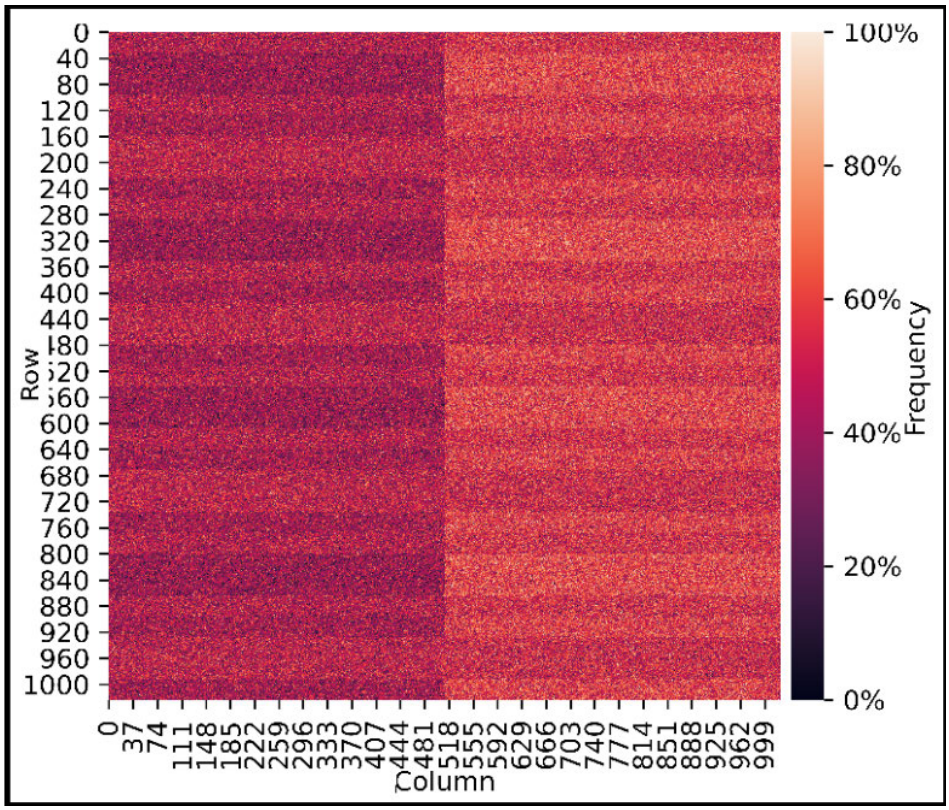

FIGURE 12. Bitmap from CY7C1021CV26, used in this paper.

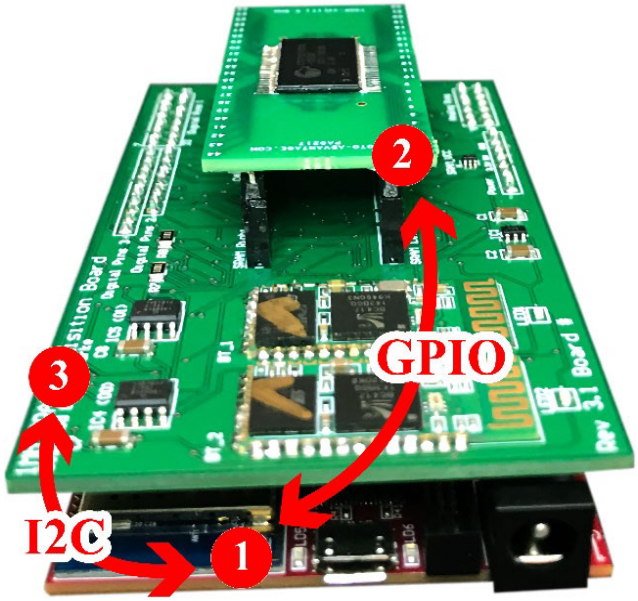

FIGURE 13. Developed PWM Node prototype (1: MCU, 2: SRAM PUF, and 3: NVSRAM as an LDB).

```
For sign in, (Authentication): press 1
For sign up, (Registeration):  press 2

You entered 2, For Sign up,
Please Enter your ID:   mm3845
Please Enter your new PW:*********
H(ID+PW):
F7 39 D0 AC AC 0D AE DE C0 62 F2 94 0B BB 90 A3
83 D9 9C 50 A5 6D 59 89 52 80 85 D6 A8 70 46 4C
MD or H(PW):
DD B2 CF AA 4A 3C 66 A6 1D 9E C1 91 A3 58 58 C6
C7 CE A8 41 3B 62 7A 74 BB 45 FE 20 F9 28 2C 05

-------OPEARATION FOR Generating LONG MESSAGE DIGEST

---Shift Done!
---8 Times Hashing Done
---Long MD is Created by Concatenation of MDs
```

FIGURE 14. Receiving data for Sign-up and generating long MD.

the PUF response bit is extracted in that address. However, if the raw address point to a fuzzy cell, that cell is ignored, and the revised address is generated. The revised address is the address of the next non-fuzzy cell after that fuzzy cell. The cells in revised addresses are used to generate the SRAM response. Cells in revised XY1 are queried to generate the content of LDB (Chj). On the other hand, the cells in revised XY2 are queried to generate the LDB address (Addi).

V. DISCUSSION ON SECURITY

Despite their advantages, SRAM PUFs are vulnerable to attacks such as photonic-emission attacks [49] or cloning

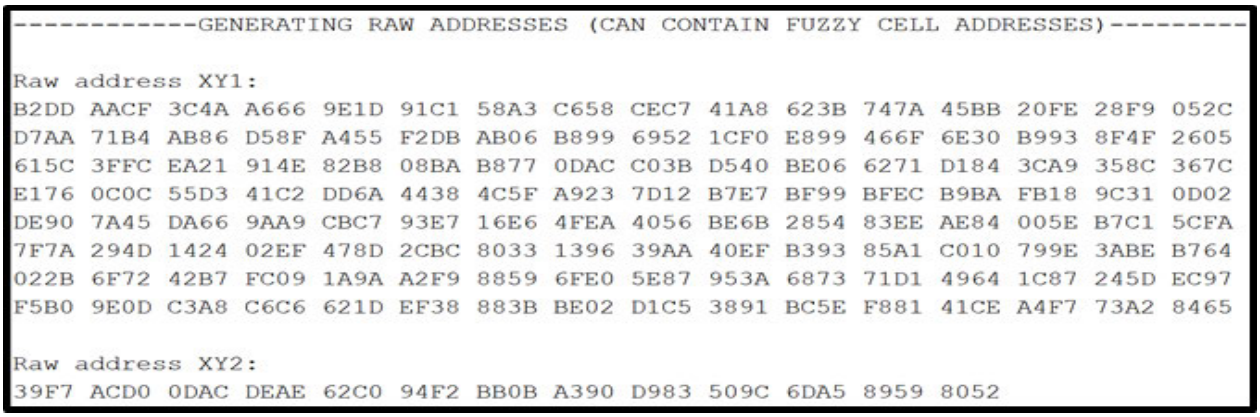

FIGURE 15. Generating raw addresses (raw XY1 for LDB content and raw XY2 for LDB address).

attacks based on remanence decay side channels [50]. Also, the hardware in this article uses generic components and therefore is not secure. Therefore, in the case of loss of the SRAM PUF hardware, the adversary can attack the SRAM PUFs. One solution that can be used to overcome the SRAM PUF security issue is taking other characteristics of SRAM PUF in addition to the pure binary response such as SRAM PUF data remanence [19].

In this article, the ternary PUF with considering three possible states for each cell is used. If the adversary access to the hardware, the ternary PUFs are a little more challenging to read and break than binary PUFs. In this article, just the first and last eight kB of the SRAM are used as a PUF. The location of the used PUF can be a piece of secret information that increases security slightly. As will be discussed in the future work section, the local database (LDB) and SRAM PUF embedded in each PWM node will cover just some users. Therefore, if the attackers access the SRAM PUFs in one node, the other nodes which use different SRAM as a PUF remain secure. However, attacking the SRAM PUF in one of the nodes can compromise the PWs of some of the users assigned to that specific node.

Replacing SRAM PUF with tamper-resistant PUFs is considered as a solution for future work. For the SRAM PUF, the first response, ''0'' or ''1,'' is read in some specific cells. The response for most of those cells is the same for the next response queries. However, in ReRAM PUF [51] or MRAM PUF [25], the cell's resistance value can be used to design a PUF. Thus, a given cell can be ''0'' when selected within a set of cells and ''1'' within a different cell set, i.e., one specific cell can be either ''0'' or ''1''. The nonobviousness of ReRAM or MRAM PUFs compared with SRAM PUFs makes them more encouraging since they are more challenging to break. In the future prototypes, ReRAMs and MRAMs will be used as a PUF.

VI. CONCLUSION

A resilient PWM prototype using SRAM PUF, low-cost MCU, and NVSRAM is demonstrated in this study. The LDB is embedded in the PWM node circuit. A significant issue, which is a system failure in the case of PUF failure, is considered in this article. This issue is solved by using a redundant element for the event of PUF failure. By saving two PUF outputs in the embedded LDB, the PWM node circuit receives the user credentials and outputs the authentication result even if the PUF fails. Based on this paper's data analysis, the applied SRAM is of much higher PUF quality than similar SRAM used in previous work. In the protocol used in this work, the PUF is used for creating both LDB addresses and contents to make the task of an adversary more challenging. Also, the protocol is based on the upfront characterization of the SRAM PUF and masking the unstable cells to improve the error rate considerably. The PWM prototype does not need central DB and is resilient to the PUF failure. This study's results can deal with one of the most critical cybersecurity attacks in many networks' PWM systems.

VII. FUTURE WORK

A limitation of using PUFs in the PWM system is latency, which precludes applications with large numbers of users. This issue will be the focus of future work. The solution considered is implementing high throughput architecture, in which several PWM nodes handled by a router and matrix controller will be used. The router can select one of the nodes based on the first character of the user ID. The LDB embedded in each PWM node will cover part of the users. Therefore, the central DB will not be required. Despite the advantages, the implementation of the architecture in which several nodes communicate with the router simultaneously is challenging.

ACKNOWLEDGMENT

The authors would like to thank several staff and instructors at Northern Arizona University, particularly Ian Burke, and Julie B Heynssens.

REFERENCES

[1] J. Hendrickson. *Why are Companies Still Storing Passwords in Plain Text?* Accessed: 2020. [Online]. Available: https://www.howtogeek.com

[2] *GitHub Says Bug Exposed Some Plaintext Passwords*. Accessed: 2020. [Online]. Available: https://www.securitynewspaper.com
[3] S. Morgan, "Cybercrime damages will cost the world $6 trillion annually by 2021," Cybersecurity Ventures, Herjavec Group, USA, Cybercrime Rep. 2017, 2017.
[4] J. Keane. *Security Researcher Dumps 427 Million Hacked Myspace Passwords*. Accessed: 2020. [Online]. Available: https://www.digitaltrends.com
[5] *Target: Data Stolen From up to 70 Million Customers*. Accessed: 2020. [Online]. Available: https://www.usatoday.com
[6] J. Blocki, B. Harsha, and S. Zhou, "On the economics of offline password cracking," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2018, pp. 853–871.
[7] D. Florencio and C. Herley, "A large-scale study of Web password habits," in *Proc. 16th Int. Conf. World Wide Web (WWW)*, 2007, pp. 657–666.
[8] R. Wash, E. Rader, R. Berman, and Z. Wellmer, "Understanding password choices: How frequently entered passwords are re-used across websites," in *Proc. 12th Symp. Usable Privacy Secur. (SOUPS)*, 2016, pp. 175–188.
[9] A. Hanamsagar, S. Woo, C. Kanich, and J. Mirkovic. *How Users Choose and Reuse Passwords*. Accessed: 2020. [Online]. Available: https://www.isi.edu/
[10] R. Arenburg, S. Chawla, A. Mathur, and C. Skawratananond, "Method, apparatus and program storage device for providing a secure password manager," U.S. Patent 2007 0 074 038 A1, Mar. 29, 2007.
[11] B. F. Cambou, "Password management with addressable physical unclonable function generators," U.S. Patent 2019 0 354 672 A1, Nov. 21, 2019.
[12] M. Mohammadinodoushan, B. Cambou, C. Philabaum, D. Hely, and D. D. Booher, "Implementation of password management system using ternary addressable PUF generator," in *Proc. 16th Annu. IEEE Int. Conf. Sens., Commun., Netw. (SECON)*, Jun. 2019, pp. 1–8.
[13] J. Shi, Y. Lu, and J. Zhang, "Approximation attacks on strong PUFs," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 10, pp. 2138–2151, Oct. 2020.
[14] G. T. Becker, "The gap between promise and reality: On the insecurity of XOR arbiter PUFs," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.* Berlin, Germany: Springer, 2015, pp. 535–555.
[15] D. P. Sahoo, D. Mukhopadhyay, R. S. Chakraborty, and P. H. Nguyen, "A multiplexer-based arbiter PUF composition with enhanced reliability and security," *IEEE Trans. Comput.*, vol. 67, no. 3, pp. 403–417, Mar. 2018.
[16] S. Assiri, B. Cambou, D. D. Booher, and M. Mohammadinodoushan, "Software implementation of a SRAM PUF-based password manager," in *Proc. Sci. Inf. Conf.* Cham, Switzerland: Springer, 2020, pp. 361–379.
[17] B. Cambou and D. Telesca, "Ternary computing to strengthen cybersecurity," in *Proc. Sci. Inf. Conf.* Cham, Switzerland: Springer, 2018, pp. 898–919.
[18] T. McGrath, I. E. Bagci, Z. M. Wang, U. Roedig, and R. J. Young, "A PUF taxonomy," *Appl. Phys. Rev.*, vol. 6, no. 1, Mar. 2019, Art. no. 011303.
[19] Y. Gao, S. F. Al-Sarawi, and D. Abbott, "Physical unclonable functions," *Nature Electron.*, vol. 3, no. 2, pp. 81–91, Feb. 2020, doi: 10.1038/s41928-020-0372-5.
[20] D. E. Holcomb, W. P. Burleson, and K. Fu, "Power-up SRAM state as an identifying fingerprint and source of true random numbers," *IEEE Trans. Comput.*, vol. 58, no. 9, pp. 1198–1210, Sep. 2009.
[21] B. Cambou, M. Mohammadi, C. Philabaum, and D. Booher, "Statistical analysis to optimize the generation of cryptographic keys from physical unclonable functions," in *Proc. Sci. Inf. Conf.* Cham, Switzerland: Springer, 2020, pp. 302–321.
[22] F. Tehranipoor, N. Karimian, W. Yan, and J. A. Chandy, "DRAM-based intrinsic physically unclonable functions for system-level security and authentication," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 25, no. 3, pp. 1085–1097, Mar. 2017.
[23] S. Assiri, B. Cambou, D. D. Booher, D. G. Miandoab, and M. Mohammadinodoushan, "Key exchange using ternary system to enhance security," in *Proc. IEEE 9th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2019, pp. 0488–0492.
[24] M. Mohammadinodoushan, "Hardware implementation of keyless encryption scheme for Internet of Things based on image of memristors," 2020, *arXiv:2007.05596*. [Online]. Available: http://arxiv.org/abs/2007.05596
[25] A. Nejat, F. Ouattara, M. Mohammadinodoushan, B. Cambou, K. Mackay, and L. Torres, "Practical experiments to evaluate quality metrics of MRAM-based physical unclonable functions," *IEEE Access*, vol. 8, pp. 176042–176049, 2020, doi: 10.1109/ACCESS.2020.3024598.
[26] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, "FPGA intrinsic PUFs and their use for IP protection," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.* Berlin, Germany: Springer, 2007, pp. 63–80.
[27] S. Eiroa, J. Castro, M. C. Martínez-Rodríguez, E. Tena, P. Brox, and I. Baturone, "Reducing bit flipping problems in SRAM physical unclonable functions for chip identification," in *Proc. 19th IEEE Int. Conf. Electron., Circuits, Syst. (ICECS)*, Dec. 2012, pp. 392–395.
[28] *The Reliability of SRAM PUF*. Accessed: 2020. [Online]. Available: https://www.intrinsic-id.com
[29] J. H. Davis, "Hacking of government computers exposed 21.5 million people," The New York Times, 2015, vol. 9.
[30] E. Nakashima. *Hacks of OPM Databases Compromised 22.1 Million People, Federal Authorities Say*. Accessed: 2020. [Online]. Available: https://www.washingtonpost.com
[31] K. Hill. *Google Says Not to Worry About 5 Million Gmail Passwords Leaked*. Accessed: 2020. [Online]. Available: https://www.forbes.com/
[32] S. Lee. *Twitter to All Users: Change Your Password Now!* Accessed: 2020. [Online]. Available: https://www.mercurynews.com/
[33] I. Ilascu. *Hackers Sell Stolen User Data From HomeChef, ChatBooks, and Chronicle*. Accessed: 2020. [Online]. Available: https://www.bleepingcomputer.com
[34] Q. Guo, J. Ye, B. Li, Y. Hu, X. Li, Y. Lan, and G. Zhang, "PUFPass: A password management mechanism based on software/hardware codesign," *Integration*, vol. 64, pp. 173–183, Jan. 2019.
[35] R. Biddle, S. Chiasson, and P. C. van Oorschot, "Graphical passwords: Learning from the first twelve years," *ACM Comput. Surv.*, vol. 44, no. 4, p. 19, Aug. 2012.
[36] S. Komanduri, R. Shay, P. G. Kelley, M. L. Mazurek, L. Bauer, N. Christin, L. F. Cranor, and S. Egelman, "Of passwords and people: Measuring the effect of password-composition policies," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, 2011, pp. 2595–2604.
[37] M. H. Almeshekah, C. N. Gutierrez, M. J. Atallah, and E. H. Spafford, "Ersatzpasswords: Ending password cracking and detecting password leakage," in *Proc. 31st Annu. Comput. Secur. Appl. Conf.*, 2015, pp. 311–320.
[38] B. F. Cambou, "PUF-based password generation scheme," U.S. Patent 10 320 573 B2, Jun. 11, 2019.
[39] B. F. Cambou, "PUF hardware arrangement for increased throughput," U.S. Patent 10 185 820 B2, Jan. 22, 2019.
[40] M. Mohammadinodoushan, "Implementation of password manager with sram-based physical unclonable function," 2020, *arXiv:2006.02562*. [Online]. Available: http://arxiv.org/abs/2006.02562
[41] B. Cambou and M. Mohammadinodoushan, "Resilient password management system using an array of addressable physical unclonable functions," U.S. Patent 63 010 413, Apr. 15, 2020.
[42] CYPRESS Semiconductor Corporation, San Jose, CA, USA. *1-Mbit nvSRAM, 001-54050 Rev.* Accessed: 2020. [Online]. Available: https://www.cypress.com
[43] B. Cambou, "Password manager combining hashing functions and ternary PUFs," in *Proc. Comput. Conf. Intell. Comput.* Cham, Switzerland: Springer, 2019, pp. 494–513.
[44] Z. Pang, J. Zhang, Q. Zhou, S. Gong, X. Qian, and B. Tang, "Crossover ring oscillator PUF," in *Proc. 18th Int. Symp. Qual. Electron. Design (ISQED)*, Mar. 2017, pp. 237–243.
[45] D. Deng, S. Hou, Z. Wang, and Y. Guo, "Configurable ring oscillator PUF using hybrid logic gates," *IEEE Access*, vol. 8, pp. 161427–161437, 2020.
[46] Y. Cui, C. Gu, Q. Ma, Y. Fang, C. Wang, M. O'Neill, and W. Liu, "Lightweight modeling attack-resistant multiplexer-based multi-PUF (MMPUF) design on FPGA," *Electronics*, vol. 9, no. 5, p. 815, May 2020.
[47] J. Zhang and G. Qu, "Physical unclonable function-based key sharing via machine learning for IoT security," *IEEE Trans. Ind. Electron.*, vol. 67, no. 8, pp. 7025–7033, Aug. 2020.
[48] A. Kumar, S. Sahay, and M. Suri, "Switching-time dependent PUF using STT-MRAM," in *Proc. 31st Int. Conf. VLSI Design 17th Int. Conf. Embedded Syst. (VLSID)*, Jan. 2018, pp. 434–438.
[49] C. Helfmeier, C. Boit, D. Nedospasov, and J.-P. Seifert, "Cloning physically unclonable functions," in *Proc. IEEE Int. Symp. Hardware-Oriented Secur. Trust (HOST)*, Jun. 2013, pp. 1–6.
[50] S. Zeitouni, Y. Oren, C. Wachsmann, P. Koeberl, and A.-R. Sadeghi, "Remanence decay side-channel: The PUF case," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 6, pp. 1106–1116, Jun. 2016.
[51] A. Chen, "Comprehensive assessment of RRAM-based PUF for hardware security applications," in *IEDM Tech. Dig.*, Dec. 2015, pp. 10.7.1–10.7.4.

MOHAMMAD MOHAMMADINODOUSHAN (Graduate Student Member, IEEE) received the M.Sc. degree in electrical engineering from Northern Arizona University, where he is currently pursuing the Ph.D. degree in informatics and computing. His past research interests include renewable energy power systems, including data mining and machine learning, intelligent control, and power electronics. His current research interests include cyber engineering for cyber security, including statistics to circuits of memory PUFs, very novel password managers utilizing PUFs, and key generation, keyless encryption, and key exchange using PUFs.

BERTRAND CAMBOU (Member, IEEE) received the Ph.D. degree from Paris-Saclay University. He worked in the smartcard/secure microcontroller industry at Gemplus (now Gemalto) and in the POS/secure payment industry at Ingenico. He spent 15 years at Motorola Semiconductor (now NXP-Freescale), where he served in multiple capacities, including CTO, and was named "Distinguished Innovator" and a Scientific Advisor of the BOD. He is currently a Professor with Northern Arizona University (NAU). He has 65 granted patents. His current research interests include cyber-security, and how to apply microelectronics to strengthen hardware security. This includes the design of novel secure elements, Physically Unclonable Functions (PUF), True Random Generators (TRNG), and the usage of nanotechnologies such as ReRAM.

CHRISTOPHER ROBERT PHILABAUM received the B.Sc. degree in computer science from Northern Arizona University, where he is currently pursuing the Ph.D. degree in informatics. His past research interests include bioinformatics, specifically on highly multiplexed barcode sequencing. His current research interests include cybersecurity and cybersystems for physical unclonable functions (PUFs), ternary systems, key exchange and generation, public key infrastructure (PKI), post-quantum cryptography, and distributed systems.

NAN DUAN received the M.Sc. degree in electrical engineering from Northern Arizona University (NAU). He is currently a temporary employee with the Cybersecurity Lab, NAU. He worked projects including characterize Physical Unclonable Functions (PUFs) which is the commercialized Static Random-Access Memory (SRAM) PUFs. And use the PUF to protect the data transmission of 3 axis sensor applied into the car to detect the road states. He was a project manager that is a cooperation project between NAU and Sawblade Adventure. His research interest includes cybersecurity, apply SRAM PUFs in different application situations.