A Highly Reliable Arbiter PUF With Improved Uniqueness in FPGA Implementation Using Bit-Self-Test

ZHANGQING HE, WANBO CHEN, LINGCHAO ZHANG, GAOJUN CHI, QI GAO, AND LEIN HARN

1Hubei Key Laboratory for High-Efficiency Utilization of Solar Energy and Operation Control of Energy Storage System, Hubei University of Technology, Wuhan 430068, China

2School of Electronics Engineering and Automation, Guilin University of Electronics Technology, Guilin 541004, China

3Department of Computer Science and Electrical Engineering, University of Missouri-Kansas City, Kansas City, MO 64110, USA

Corresponding author: Zhangqing He (ivan_hee@126.com)

This work was supported in part by the Hubei Provincial Natural Science Foundation of China under Grant 2020CFB814.

ABSTRACT Physically unclonable functions (PUFs) promise to be a critical hardware primitive for billions of Internet of Things (IoT) devices. The arbiter PUF (A-PUF) is one of the most well-known PUF circuits. However, its FPGA implementation has a poor reliability, and error correction codes (ECCs) are usually needed to eliminate the noise in the responses, which incur additional high hardware overhead and require NVM for helper data storage. In this paper, we present a highly reliable arbiter PUF with improved uniqueness using the bit-self-test (BST) strategy. A delay detection circuit is added into a classical arbiter PUF to test the delay deviation that produces each bit of the PUF response in real-time and mark the response as reliable using a reliability flag when the delay deviation is significantly more than a predefined threshold. Then, the robust responses can be used. We implemented the BST-arbiter PUF on a Xilinx Artix 7 FPGA. The test results show that the selected responses achieve outstanding performance where the bit error rate is less than $10^{-9}$, the bias is 50.3%, and the uniqueness is 49.1%. Thus, the BST-APUF, which drastically reduced the ECC overhead, can be applied to lightweight cryptography applications.

INDEX TERMS Physically unclonable functions, bit-self-test, arbiter PUF, reliability enhancement, reliability flag.

I. INTRODUCTION

Internet of Things (IoT) systems usually utilize various cryptographic algorithms to encrypt private data and authenticate each other [1], and the security of the key is particularly essential. The cryptographic keys are usually stored in the nonvolatile memory (NVM) in most IoT devices. However, the keys stored in NVM are not secure enough because they are most likely to be detected by physical attacks [2]. A PUF is an emerging cryptographic primitive for key generation/storage and can be widely used in authentication and key generation for secure operations. A PUF can be represented by the mapping of an external input challenge $C$ to an output response $R$. The response $R$ to the challenge $C$ is random but unique for an IC that contains the same PUF circuit due to their mismatches in the electrical parameters caused by the manufacturing process variations. The challenge-response pairs (CRPs) that uniquely describe an IC are prohibitively difficult to reproduce using a physical clone of the same circuit.

Recently, researchers have proposed various PUF circuits, including the arbiter PUF [3], [4], the ring oscillator PUF [5], [6], the switched-capacitor PUF [7] and the DRAM PUF [8], [9]. The arbiter PUF (A-PUF) is one of the most well-explored PUF designs since it requires fewer hardware resources than RO-PUF and has a bigger challenge-response space than memory-based PUFs. As a well-known delay-based PUF, the A-PUF has been employed as the underlying block in more complicated PUF architectures, such as the feed-forward arbiter PUF [10], lightweight PUF [11], and XOR PUF [12]. It also has been utilized by Verayo to implement RFID ICs [13] and as a hardware root of trust for its Security Authentication [14]. However, the FPGA implementation of the classical A-PUF has poor reliability [15]...
mainly due to the metastability of the delay flip-flop (DFF) arbiter and the routing constraints of the FPGA. Postprocessing using helper data algorithms (HDAs) [16] is required to extract the reliable keys from noise responses. However, most HDAs require complex error correction codes (ECCs) such as BCH with an additional high hardware overhead and NVM for helper data storage.

Given that the ECC overheads decrease significantly if the errors in the raw responses decrease, many lightweight reliability enhancement mechanisms are proposed for APUFs to reduce the error correction cost. A metastability detection technique using the 4-DFF or SR latch-based arbiter to significantly improve the average and minimum reliabilities has been proposed in [17]. However, this technique can result in unbalanced responses and increased overhead of detecting and recoding meta-stabilization states. More arbiters can also be added to reduce the number of utilizable challenges and improve the reliability of the response at the expense of the hardware overhead [18]. In addition, the outputs of multiple arbiters are correlated, which can compromise the unpredictability of the response. A reliability-enhanced A-PUF with trinary digit (trit) quadruple responses, which uses two flip-flop arbiters to produce a bit for metastability detection, was proposed in [19]. Its challenge-response quadruple classification dramatically reduces the burden of error correction, but the method still faces similar problems with high hardware resource requirements.

Preselection is an efficient preprocessing method that can identify and select the stable PUF responses. After the preselection process, the error rate will decrease, and thus the error correction becomes less complex or even unnecessary [20], [21]. A simple method to determine the stable responses is measuring them multiple times across different operating conditions [22], but this requires significant additional runtime and costs. The more efficient approaches are run test on the PUF cells and the response that has a mismatch exceeding a predefined limit is selected. For example, Bhargava and Mai [23] use a built-in self-test to determine which sense amplifier (SA) PUF bits are reliable and only B. Preselection

In [29], He et al. proposed a bit-self-test (BST) preselection strategy for a switched-capacitor (SC) strong PUF in which a self-test module was added to automatically test the capacitance difference that produces each response and generates a flag bit to indicate the reliability of this response. This reliability enhancement strategy is suitable for both weak PUFs and strong PUFs, and the selected robust responses can achieve very high stability across environmental variations.

In this paper, we present an arbiter PUF circuit with improved reliability and uniqueness based on the BST strategy. The main contributions of this paper are as follows:

1) We propose a novel bit-self-test arbiter PUF (BST-APUF), which can generate a response and a reliability flag bit that indicates the reliability of the response when inputting a challenge. This means that unlike traditional PUFs, the BST-APUF produces a large amount of challenge-response-reliability flag pairs (CRRPs). The reliability flags can be used as the helper data to extract or recover the stable keys from the original PUF responses in cryptographic applications.

2) We design a BST-APUF by adding a delay detection circuit into a classical arbiter PUF. The detection circuit automatically tests the delay deviation that produces each bit of the PUF response and marks the response as reliable using a reliability flag when the delay deviation is more significant than a certain threshold.

3) The BST-arbiter PUF is implemented on Xilinx Artix 7 FPGA. The test results show that the selected responses achieve very high reliability with improved uniqueness. The hardware resource requirement of the BST-APUF is lower than those of the state-of-the-art mechanisms, and it is suitable for resource-constrained systems.
Most PUFs produce responses by amplifying some electrical characteristics (e.g., delay and threshold voltage) from two nominally identical circuit components in the PUF core. When the electrical difference is significant, the PUF response will more robust across environmental variations [15]. Therefore, if the electrical differences causing each PUF response can be tested automatically, the PUF responses with more considerable differences can be selected for key generation, and the reliability of the PUF will be significantly improved.

For an arbiter PUF, it produces responses by detecting the delay difference between two symmetric delay paths under the control of the challenge. If the delay difference $\Delta D$ for a given applied challenge at a nominal temperature is small, the polarity of $\Delta D$ is very likely to change due to the variations in temperature and supply voltage, and the response will change, as Fig. 2(a) shows. In contrast, for a larger $\Delta D$, as Fig. 2(b) shows, its polarity is unlikely to be affected by the temperature; thus, the response will be more stable. Thus, the absolute value of the delay differences $|\Delta D|$ is a good indicator of the response reliability.

However, the delay differences $|\Delta D|$ are often too small to measure directly. Moreover, the arbiter PUF can generate a large number of responses, and it is not feasible to test the reliability of each response before delivery. Concerning this problem, this paper proposes a bit-self-test arbiter PUF (BST-APUF) by adding a delay detection circuit to a classical arbiter PUF. The detection circuit can automatically test the $|\Delta D|$ that produces each response and generates a reliability flag for each response to indicate its reliability. If $|\Delta D|$ is larger than a threshold $D_t$, the flag is set to 1, and vice-versa. Generally, when a challenge is used as an input into the BST-APUF, it can generate a response and a reliability flag simultaneously, as Fig. 3 (a) shows. If the reliability flag is 1, it represents that the corresponding response is reliable; otherwise, the response is unreliable. Therefore, unlike traditional PUFs, the BST-APUF can produce a large amount of challenge-response-reliability flag pairs (CRRPs). We can select the robust responses according to the reliability flags to generate cryptographic keys or conduct identification and authentication; thus, the reliability flags can be used as the helper data to extract or recover the keys. The extraction process of robust responses is shown in Fig. 3 (b).

It should be noted that the reliability flags might change between different evaluations with the same challenge because of the noise. In some cases, the reliability flags generated during the registration phase should be stored so that it can be used in recovery phase to recover the same stable keys from the noise responses. In other cases, such as in the computationally secure fuzzy extractor proposed in...
Cao et al. [32], the reliability flags do not need to be stored. They can be treated as confidential information to recover the key, even though they change during each key extraction.

C. EXECUTION PROCESS OF THE BST-APUF

The key to realizing the BST-APUF is to design a detection circuit to measure the delay deviation that causes each response and determine if it is greater than a threshold $D_T$. Since the value of $|\Delta D|$ is too small to measure directly, we embed an additional delay module that can generate a threshold delay $D_T$ into a classical APUF. By connecting the delay module to the upper and lower delay paths separately, the reliability flags can be produced by checking the outputs in these two situations. The execution process of BST-APUF is as follows:

First, the PUF works in a response-output mode to generate a response $R_i$.

At this time, the delay module is not connected to the circuit, and the challenge $C_i$ causes the N-stage switching delay module of the arbiter PUF to form two delay paths, as shown in Fig. 4. If the total delay caused by the upper path to the input pulse is $D_1$, and the total delay caused by the lower path is $D_2$, the delay difference $\Delta D = D_1 - D_2$ is input to the arbiter. When $\Delta D > 0$, the arbiter outputs a response of 0. When $\Delta D < 0$, the arbiter outputs a response of 1, and the response $R_i$ is stored in a register.

Second, the PUF enters a reliability self-test mode to produce the reliability flag $F_i$. $F_i$ will be generated in three steps:

1) The delay module is connected to the upper delay path to generate a test output $T_{i1}$, as shown in Fig. 5. Since the delay module can generate a delay of $D_T$, the delay of the upper path is $D_1 + D_T$. The delay difference $\Delta D_1 = \Delta D + D_T$ is input to the arbiter, resulting in a test output $T_{i1}$.

2) The delay module is connected to the lower path to generate the test output $T_{i2}$. At this time, the delay difference of the two delay paths, $\Delta D_2 = \Delta D - D_T$, is input to the arbiter to generate a test output $T_{i2}$, as shown in Fig. 6.

3) Generate the reliability flag $F_i$.

In steps (1) and (2), if the output of $T_{i2}$ and $T_{i1}$ are the same, that is, the polarities of $\Delta D_1$ and $\Delta D_2$ are the same, then the following is satisfied:

\[ \Delta D_1 = \Delta D + D_T > 0 \quad \text{and} \quad \Delta D_2 = \Delta D - D_T > 0, \]

or $\Delta D_1 < 0$ and $\Delta D_2 < 0$.

Since $D_T$ is a positive value, we can get that $|\Delta D| > D_T$. That is, the absolute value of $\Delta D$ is higher than the threshold $D_T$. If $D_T$ is set to be large, the response $R_i$ generated by this $\Delta D$ will be very stable across environmental variations.

Conversely, if $T_{i2}$ is different from $T_{i1}$, it satisfies the following:

\[ \Delta D_1 > 0 \quad \text{and} \quad \Delta D_2 < 0, \quad \text{or} \quad \Delta D_1 < 0 \quad \text{and} \quad \Delta D_2 > 0. \]

This means the delay difference $|\Delta D|$ is less than $D_T$, and, thus, the output $R_i$ will be unstable.

Therefore, we can generate $F_i$ via the Exclusive-NOR (XNOR) $T_{i1}$ with $T_{i2}$, i.e., $F_i = T_{i1} \oplus T_{i2}$. When $F_i = 1$,
it means that $T_{i1}$ and $T_{i2}$ are the same, the response $R_i$ generated by the PUF circuit under the challenge $C_i$ is reliable and the response will not change as the temperature and voltage change. $F_i$ is stored in the register along with $R_i$.

By performing the above process repeatedly with different challenges $C_i$, we can get many CRRPs.

### III. IMPLEMENTATION AND TEST RESULTS

#### A. IMPLEMENTATION OF BST-APUF

The designed BST-APUF implementation structure is shown in Fig. 7, it consists of five modules: a 64-stage arbiter PUF, a self-test module, a reliability flag generator, a controller, and a UART.

In the design, the self-test module consists of a delay module, two 2-2 multiplexers $A_1$ and $A_2$, and two 2-1 multiplexers, as shown in Fig. 7. The delay module can generate a test delay $D_T$. In our design, we simply use several nongates in series. Of course, more complex circuits can be adopted if you need a more accurate $D_T$. In the following, we call each nongate a “delay gate”. The multiplexer $A_1$ is used to connect the delay gates to the upper or lower delay paths, and $A_2$ is used to ensure that the ports of the two delay paths input to the arbiter do not change. By executing the response output mode and test output mode, respectively, the arbiter and the reliability flag generator can generate the responses and the corresponding reliability flags.

The reliability flag generator includes a response register REG1, a reliability flag register REG2, a XNOR module, two 1-2 data distributors, and a 2-1 MUX, as shown in Fig. 8.

The controller generates control signals S and K to control the operation of the PUF. It works as follows:

1) Set $S = 0$ to put the PUF in the response-output mode. At this time, the signals $D_1$ and $D_2$ from the upper and lower paths are directly connected to the arbiter through MUX1 and MUX2 to generate response $R_i$. $R_i$ is stored in REG1 through DMUX1 inside the reliability flag generator.

2) Set $S = 1$ to put the PUF in self-test mode. In this mode, the control signal $K$ is set to “0” first to connect the delay module to the upper delay path, and the test output $T_{i1}$ is generated by arbiter and then stored in REG2 through DMUX2 and MUX3. Then, $K$ is set to “1”, the delay module is connected to the lower delay path, and test output $T_{i2}$ is generated and then the exclusive-NOR with $T_{i1}$ in REG2 is used to get the reliability flag $F_i$. Finally, $F_i$ is stored in REG2.

#### B. EXPERIMENTAL ENVIRONMENT

To verify the performance of the BST-arbiter PUF, a 64-stage BST-APUF design was coded in VHDL and simulated by the
We implemented 128 BST-APUFs on 8 FPGA boards. Each FPGA contains 16 64-stage BST-APUF units and a register array. A number of 64-bit random challenges are generated by a true random number generator and input into 128 PUF units, respectively, and the responses and the reliability flags from each PUF are obtained. The data are transmitted to the computer through UART for statistical analysis.

Due to the threshold, $D_T$ dramatically affects the value of the reliability flags. The larger that $D_T$ is, the lower the probability that the reliability flag is 1, and the higher the reliability of the selected robust responses. Thus, we use different connected delay gates (from 0 to 8 nongates) in the delay module and analyze the ratio and reliability of the selected robust responses under different temperatures.

It should be noted that if the routing is performed automatically by the tool when implementing the method on the FPGA, the mismatch within the elements in A1 and A2 of the self-test module could counter the effect of the delay element. For cases when the delay module uses few delay gates, sometimes even to an extent, the path without the delay module might have a delay higher than another path. Therefore, the layout design and routing need to be performed manually to decrease the path mismatch. We did our best to constrain and guide the placement and routing software to achieve the highest degree of symmetry in the PUF layout. The estimated latency bias between the two paths within the self-test module by the ISE software is less than 0.125 ns after optimization. Since the delay of a nongate implemented by LUT in our test FPGA is approximately 0.6 ns, it is much greater than the mismatch of 0.125 ns. Thus, even for case with one nongate, the path with the delay module will have a delay more significant than another path.

**C. EXPERIMENTAL RESULTS**

1) **MISMATCH OF THE SELF TEST MODULE**

We first measured the mismatch of the self-test module to ensure that it will not have a noticeable effect on the self-test circuit. In the original APUF (as shown in Fig. 1), if the delay difference $\Delta D$ between two symmetric delay paths is greater than 0, the output response is 1; otherwise, it is 0. When the self-test mode is connected to the delay paths (as shown in Fig. 7), the mismatch of the self-test module, $D_m$, and the additional delay generated by the delay module, $D_T$, are superimposed on the original delay $\Delta D$, leading to a change of the bias (the ratio of ‘1’) in the test outputs. The larger $D_m$ or $D_T$ is, the higher the change in the bias. Since $D_m$ and $D_T$ are difficult to detect and compare directly, we can measure the changes in the bias between the test outputs and the original APUF responses instead.

If the BST-APUF uses no delay gate in the delay module, the additional delay $D_T$ is zero, and the change in the bias is entirely caused by the mismatch $D_m$. When 1 delay gate is used, the delay $D_T$ generated by 1 delay gate, together with the mismatch $D_m$, will contribute to the bias change. Therefore, we can compare the mismatch $D_m$ and the delay $D_T$ by comparing the bias changes of the PUF outputs in above cases.

We performed experiment using 64 BST-APUF instances, each measuring 65536-bit responses, to calculate the bias characteristics. We first measured the bias of the original responses for each BST-APUF instance by letting them work in the response-output mode. Then, we made them work in the self-test mode and connected the delay module to the upper and lower delay paths using no delay gate and 1 delay gate, respectively. The bias changes compared with the original PUF responses in the above four cases are calculated and the results are shown in Fig. 9, in which the X axis is the serial number of the PUF instances and the Y axis represents the value for their bias changes compared with the original responses.

It can be seen that the bias changes when connecting 1 delay gate are significantly larger than those when connecting no delay gate. This means that the delay generated by even 1 delay gate is higher than the mismatch of the self-test module. Moreover, when connecting 1 delay gate, the biases of all the PUFs are increased compared with the original responses (the bias changes are greater than 0) when the delay
module is connected to the upper path, and they are decreased when the delay module is connected to the lower path. This proves that the path with the delay module always has a delay higher than the path without the delay module. Furthermore, in order to obtain high reliability, more delay gates will be used in practical applications. Thus, the delay generated by the delay module will much higher than the mismatch of the self-test module.

2) ROBUST RESPONSES AND THEIR RELIABILITY

Fig. 10 shows the percentage of 1s in the reliability flags when using different delay gates under the normal temperature (25°C). It can be seen that as the number of test delay gates increase, the ratio of 1s in the reliability flags rapidly decreases. Then, the reliability of the selected robust responses (the responses where the reliability flags are 1) is measured. The reliability can be quantified using the bit error rate (BER), which is the average percentage of erroneous response bits obtained in different periods or operating environments. The BER of PUF instance $i$ was calculated by the formula [31], [32]:

$$\text{BER} = 1 - \text{reliability} = \frac{1}{k} \sum_{j=1}^{k} \frac{\text{HD}(R_i, R_{ij})}{n} \times 100\% \quad (1)$$

where $R_i$ is an $n$-bit response produced by PUF instance $i$ under normal operating conditions (1.0 V and 25°C) and a set of input challenges, $C$. Then, the same set of challenges is applied $k$ times to the same PUF under multiple operating temperatures ranging from $-25$ to $80°C$ to obtain the responses $R_{ij}$ for $j = 1, 2, \ldots, k$. Since $R_i$ and $R_{ij}$ are $n$-bit responses generated by $n$ challenges, their Hamming distances are divided by $n$ to calculate the average error rate per bit.

To test the reliability of the robust responses, 65536 random challenges are input into a BST-APUF instance, and 65536 bits responses and 65536 bits reliability flags are generated at a voltage of 1.0 V and a temperature of 25°C. The challenges and the responses with reliability flags of 1 are selected as the reference challenges and the reference robust responses. Then, the reference challenges are applied 1000 times to the same PUF instance under $-25°C$, $0°C$, $25°C$, and $80°C$, respectively. The responses generated in the above environment are compared with the reference robust responses to calculate the BER for this BST-PUF instance. Then, all the other 127 PUF instances are tested to calculate the average BER of the robust responses.

The BERs of the selected robust responses for different delay gates (from 0 to 8) are measured. Fig. 11 shows the BERs at a normal temperature (25°C). It can be seen that as the number of connected delay gates increases, the BERs of the selected robust responses rapidly decrease. When the number of connected delay gates is 0, 1 and 2, the BER of the robust responses is 0.634%, 0.129% and 0.004%, respectively. When 3 delay gates are connected, only 2 bit errors were observed among the $2^{24} \times 10^9$ bit measurements for all the 128 PUF instances; thus, the BER is approximately $8.9 \times 10^{-10}$. When the number of connected delay gates is greater than or equal to 3, no errors were found while the
tested robust responses are more than $10^{10}$ bits; thus, the bit error rate is less than $10^{-10}$.

Fig. 12 and Fig. 13 show the BERs of the selected robust responses for different delay gates at different operating temperatures. The worst-case BER was measured at a temperature of $80^\circ C$, as Fig. 12 shows. At this temperature, the BERs of the robust responses are 3.52%, 1.03%, 0.036% and $4.46 \times 10^{-9}$, respectively, when the numbers of connected delay gates are 0, 1, 2 and 3, respectively. If more than 3 delay gates are connected, no errors were found in more than $10^{10}$ bit measurement robust responses.

4) UNIQUENESS

Uniqueness represents the distinguishability of PUF pairs, and it is a measure of the average number of bit differences among the responses of different PUF instances to the same challenge. It is usually estimated by the inter Hamming distance (HD) as follows:

$$U = \frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \frac{\text{HD}(R_i, R_j)}{n}$$  \hspace{1cm} (2)

where $m$ is the number of PUF instances. $R_i$ and $R_j$ are two $n$-bit responses to the same challenge generated by two different PUF instances, $i$ and $j$, respectively.

We first tested the uniqueness of the original arbiter PUF. We randomly selected 65536 challenges, input them into 128 PUF instances, obtained 65536-bit responses from each PUF, and evaluated the Hamming distance of the 65536-bit response between each of the two PUFs. The HD distribution is shown in Fig. 16 (a), and the average inter HD is 38.5%.

Then, we measured the uniqueness of the selected robust response of the BST-APUF for different delay gates. For these measurements, it should be noted that when inputting the same challenge to different BST-APUFs, the length and position of the selected response will change; therefore, it was difficult to directly calculate the Hamming distance between them.

Since we only used the selected robust responses, the average Hamming distance between these responses represents the interchip distinction of BST-PUFs. Given that the length of the selected response for each PUF was different, we selected the shortest as the benchmark. During the test, the same 65536 challenges were input into the respectively 128 PUF circuits. If the robust responses selected from 128 PUFs were at least $N$ bits, we first collected the $N$ selected responses of all PUFs and evaluated their average Hamming
distance distribution. The average inter HD calculated by this method for different delay gates is shown in Fig. 15. It can be seen that the uniqueness of BST-PUFs greatly improved, from 38.5% to nearly 50%. The histogram of the inter HD distribution of the selected response when the number of connected delay gates was 3 is shown in Fig. 15(b), and the average inter HD was 49.1%. Therefore, the BST reliability enhancement method proposed in this paper can improve both the uniqueness and reliability.

5) RESOURCE CONSUMPTION AND PERFORMANCE
The test results show that the bit error rate, bias and uniqueness of the original 64-stage arbiter PUF were 3.52%, 58.1%, and 38.5%, respectively. The number of occupied slice LUTs and registers were 96 (20800) and 19 (41600), and the speed was 120.7 kbps. After using the BST, the number of occupied slice LUTs was 150, and the number of slice registers was 47. The resource consumption increased by approximately 70%, and the speed was reduced to one-third or 43 kbps. This increased overhead is far less than those of the existing lightweight ECC techniques [33]. Moreover, if there are many BST-APUF instances in a chip, they can share one BST circuit, and, therefore, it does not create significant overhead.

When connecting 3 delay gates, the selected responses can achieve outstanding performance: The BER is less than $10^{-9}$, the bias is 50.3%, and the uniqueness is 49.1%. They can be used directly for cryptographic applications without using any error correction mechanism. At this time, the ratio of selected responses is 66.85%. This means that the BST-APUF only requires 1.5 raw response bits to generate a reliable bit. It is much better than the current ECC implementations. Therefore, the BST-APUF greatly reduces the complexity and implementation overhead of the error correction mechanism.

Table 1 gives a comparison of the BST-APUF with other newly proposed arbiter-based PUFs. Our proposed PUF has higher reliability, lower resource consumption, and similar bias and uniqueness characteristics as other state-of-the-art PUFs.

Table 2 provided the implementation overhead that produces a 1 bit response to different BERs in our experiments. Due to the improvement of robust response reliability, the number of original responses required to produce a 1-bit robust response increases, and thus the execution time and energy consumption will increase. For some cryptography applications that do not require high reliability, we can trade-off between the overhead of the implementation and the BER. We did not provide the energy consumption data in the table. This is because our design and testing are based on an FPGA platform while most of the power dissipation in an FPGA occurs from the static, clocking and I/O power consumption, making it difficult to accurately evaluate the energy consumption that produces 1 bit PUF responses.

IV. DISCUSSION
A. APPLICATION POSSIBILITIES
BST-APUF has many potential applications. Many state-of-the-art fuzzy extractors [33] and authentication protocols [34] use the response reliability confidence information to eliminate the effects of noise. For example, Herder et al. [30] propose a computationally secure fuzzy extractor that treats the reliability confidence information as a trapdoor to build a stateless key generator. The TREVERSE protocol [34] allows the prover to directly process noisy PUF responses via reliability confidence information; it ultimately discards expensive ECC logic, thus removing the necessity of the helper data that are exploitable by an adversary.
TABLE 1. Comparison of BST-APUF with other PUFS.

| Approach       | Stage | Bias or Randomness | Uniqueness | Reliability (BER) | Resource Usage |
|----------------|-------|--------------------|------------|-------------------|----------------|
| 4-DFF MA-PUF [17] | 128   | 54.1%              | 49.59%     | 0.16%             | 302 LUTs 512 FFs |
| SR latch MA-PUF [17] | 128   | Pass NIST tests    | 49.29%     | 0.17%             | 506 LUTs 256 FFs |
| TQR-APUF [19]   | 64    | Pass NIST tests    | 49.7%      | 0.577%            | 192 LUTs 132 FFs |
| RPUF[47]        | 32    | 49.6%              | 50.58%     | 6.7%              | 1536 LUTs 128 FFs |
| BST-APUF        | 64    | 50.3%              | 49.1%      | <10⁻⁶             | 150 LUTs 47 FFs |

TABLE 2. The implementation overhead to BER.

| BER    | Logic Costs | Execution time (per bit) |
|--------|-------------|--------------------------|
| 1.235% | 96 LUTs, 19 Registers | 0.01 ms |
| 0.634% | 150 LUTs, 47 Registers | 0.03 ms |
| 0.129% | 150 LUTs, 47 Registers | 0.034 ms |
| 0.004% | 150 LUTs, 47 Registers | 0.042 ms |
| <10⁻⁹  | 150 LUTs, 47 Registers | 0.045 ms |

However, for most PUF constructions, the response reliability confidence is hard to be directly measured on-chip. For instance, the APUF’s response reliability is not measurable on-chip unless expensive peripheral circuits are used. Although there are works that capture the confidence information by modeling [34]–[36], they are only suitable for resource-rich servers and not resource-constrained PUF devices. Our proposed bit-self-test strategy can measure the reliability confidence information of the APUF responses directly and easily on-chip. This makes it possible to apply APFs to the above mechanisms, which can potentially minimize the computational complexity of the device and server.

Additionally, as described in the next paragraph, BST-APUFs can be applied in many of the state-of-the-art authentication and key generation protocols if the responses and reliability flags are not exposed to the attackers.

B. SECURITY VULNERABILITY AND COUNTERMEASURES

Due to the strong linear correlation between the response and challenge, it is well known that the APUF is vulnerable to modeling attacks when using machine learning (ML) algorithms [37]. Furthermore, in our proposed BST-APUF, the reliability flags will reveal the confidential information of the responses, which will increase the modeling attack vulnerability and reduce the number of CRPs required for a successful modeling attack [38], [39]. The confidential information in [38], [39] is obtained by measuring the stability of the responses multiple times while the BST-APUF provides it directly. Therefore, the BST-APUF is more vulnerable to modeling attacks. Thus, we need to be quite careful about security when applying it to cryptography applications.

To resist modeling attacks, many defenses for traditional APUFs are proposed and can be roughly classed into structural nonlinearization and CRP obfuscation [40]. Structural nonlinearization implements nonlinear PUF structures to obstruct ML-based modeling. CRP obfuscation [35] can hide the mapping of CRPs to prevent attackers from collecting valid CRPs to model strong PUFs. Many of them, such as using nonlinear structures such as forward feedback PUF (FF PUFs) [41] and multiplexer PUFs (MPUFs) [42], can be applied to BST-APUFs to increase their modeling attack resistance. However, most of the nonlinearization structure and CRP obfuscation methods are vulnerable to advanced ML attacks such as approximation attacks [43] and CMA-ES [38], [39]. Another approach is to reduce the exposure of responses and helper data in the protocol. For example, Yu et al. [44] proposed a lockdown technology that restricts the maximum number of APUF CRPs that can be acquired by an adversary, increasing the machine learning difficulty. However, it limited the number of CRPs for authentication.

In fact, since the helper data of the widely used fuzzy extractor also may leak some information about the derived key, many provable security PUF-based authentication protocols [45], [46] proposed recently encrypt the responses and helper data before sending them over the network, making it impossible for attackers to obtain the CRPs and the helper data. By using lightweight encryption such as XOR, these protocols provide a good balance between the security and a low overhead. In future work, we will demonstrate that our proposed BST-APUF can be easily applied to the above protocol. By using the reliability flags instead of the helper data of the fuzzy extractor, the error correction process can be significantly simplified, and the execution overhead will be significantly reduced, making these protocols more suitable for resource-constrained IOT systems. Meanwhile, the responses and reliability flags are encrypted before transmission, preventing an attacker from acquiring the CRPs to perform modeling attacks.

V. CONCLUSION

This paper presents a bit-self-test arbiter PUF with high reliability and uniqueness. We first proposed a new concept of
BST-APUFs and then designed a BST-arbiter PUF by adding a delay detection circuit into a classical arbiter PUF. The detection circuit can automatically test the delay difference that produces each response and generate a reliability flag for this response. Thus, we can test and select the reliable responses automatically at normal temperatures and working environments with low overhead. We implemented the BST-APUF on a Xilinx Artix 7 FPGA. The test results show that the selected responses achieved very high reliability with improved uniqueness, and thus they drastically reduced the ECC overhead.

REFERENCES

[1] X. Wang, Z. Liu, Y. Gao, X. Zheng, Z. Deng, and X. Shen, “A near-optimal protocol for the grouping problem in RFID systems,” *IEEE Trans. Mobile Comput.*, early access, Dec. 24, 2020, doi: 10.1109/TMC.2019.2962125.

[2] H. Handschuh, P. Paillier, and J. Stern, “Probing attacks on tamper-resistant devices,” in *Cryptographic Hardware and Embedded Systems* (Lecture Notes in Computer Science), vol. 1717. Berlin, Germany: Springer, 1999, pp. 303–315.

[3] D. Lim, J. W. Lee, B. Gassend, G. E. Suh, M. van Dijk, and S. Devadas, “Extracting secret keys from integrated circuits,” *IEEE Trans. Very Large Scale Intege. (VLSI) Syst.*, vol. 13, no. 10, pp. 1200–1205, Oct. 2005.

[4] Z. He, W. Chen, X. Xu, L. Ham, and M. Wan, “A reliable and efficient PUF-based cryptographic key generator using bit self-tests,” *Electron. Lett.*, vol. 56, no. 16, pp. 803–806, Aug. 2020.

[5] A. Cherkaoui, L. Bossuet, and C. Marchand, “Design, evaluation, and optimization of physical unclonable functions based on transient effect ring oscillators,” *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 6, pp. 1291–1305, Jun. 2016.

[6] S. Khan, A. P. Shah, N. Gupta, S. S. Chouhan, J. G. Pandey, and S. K. Vishwakarma, “An ultra-low power, reconfigurable, aging resilient RO PUF for IoT applications,” *Microelectron. J.*, vol. 92, Oct. 2019, Art. no. 104605.

[7] M. Wan, Z. H. Han, K. Dai, and X. Zou, “An invasive-attack-resistant PUF based on switched-capacitor circuit,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 8, pp. 2024–2034, Aug. 2015.

[8] A. Schaller, W. Xiong, N. A. Anagnostopoulos, M. U. Saleem, S. Gabmeyer, B. Skoric, S. Katzenbeisser, and J. Szefer, “Decay-based DRAM PUFs in commodity devices,” *IEEE Trans. Depend. Sec. Comput.*, vol. 16, no. 3, pp. 462–475, May 2019.

[9] F. Tehranipoor, N. Karimian, W. Yan, and J. A. Chandy, “DRAM-based PUFs,” in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, pp. 523–528, Nov. 2010.

[10] J. W. Lee, D. Lim, B. Gassend, G. E. Suh, M. van Dijk, and S. Devadas, “Extracting secret keys from integrated circuits,” *IEEE Trans. Very Large Scale Intege. (VLSI) Syst.*, vol. 13, no. 10, pp. 1200–1205, Oct. 2005.

[11] Z. He, W. Chen, X. Xu, L. Ham, and M. Wan, “A reliable and efficient PUF-based cryptographic key generator using bit self-tests,” *Electron. Lett.*, vol. 56, no. 16, pp. 803–806, Aug. 2020.

[12] A. Cherkaoui, L. Bossuet, and C. Marchand, “Design, evaluation, and optimization of physical unclonable functions based on transient effect ring oscillators,” *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 6, pp. 1291–1305, Jun. 2016.

[13] S. Khan, A. P. Shah, N. Gupta, S. S. Chouhan, J. G. Pandey, and S. K. Vishwakarma, “An ultra-low power, reconfigurable, aging resilient RO PUF for IoT applications,” *Microelectron. J.*, vol. 92, Oct. 2019, Art. no. 104605.

[14] M. Wan, Z. H. Han, K. Dai, and X. Zou, “An invasive-attack-resistant PUF based on switched-capacitor circuit,” *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 8, pp. 2024–2034, Aug. 2015.

[15] A. Schaller, W. Xiong, N. A. Anagnostopoulos, M. U. Saleem, S. Gabmeyer, B. Skoric, S. Katzenbeisser, and J. Szefer, “Decay-based DRAM PUFs in commodity devices,” *IEEE Trans. Depend. Sec. Comput.*, vol. 16, no. 3, pp. 462–475, May 2019.

[16] F. Tehranipoor, N. Karimian, W. Yan, and J. A. Chandy, “DRAM-based intrinsic physically uncloneable functions for system-level security and authentication,” *IEEE Trans. Very Large Scale Intege. (VLSI) Syst.*, vol. 25, no. 3, pp. 1085–1097, Mar. 2017.

[17] J. W. Lee, D. Lim, B. Gassend, G. E. Suh, M. van Dijk, and S. Devadas, “A technique to build a secure key in integrated circuits for identification and authentication applications,” in *Proc. IEEE Symp. VLSI Circuits Dig. Tech. Papers*, Jun. 2004, pp. 176–179.

[18] A. Cherkaoui, L. Bossuet, and C. Marchand, “Design, evaluation, and optimization of physical unclonable functions based on transient effect ring oscillators,” *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 6, pp. 1291–1305, Jun. 2016.

[19] S. Khan, A. P. Shah, N. Gupta, S. S. Chouhan, J. G. Pandey, and S. K. Vishwakarma, “An ultra-low power, reconfigurable, aging resilient RO PUF for IoT applications,” *Microelectron. J.*, vol. 92, Oct. 2019, Art. no. 104605.
ZHANGQING HE was born in Hubei, China, in 1980. He received the M.Sc. and Ph.D. degrees in microelectronics from the Huazhong University of Science and Technology, China, in 2008 and 2016, respectively. He currently works as a Full Associate Professor with the Hubei University of Technology. He has published over 30 journal articles and refereed international conference papers. His research interests include hardware security, silicon physical unclonable function, and analog/mixed signal VLSI circuits and systems.

WANBO CHEN was born in Hubei, China, in 1995. He received the B.S. degree in electrical engineering and automation from the Huazhong University of Science and Technology Wenhua College, Wuhan, China, in 2018. He is currently pursuing the M.S. degree with the Hubei University of Technology, Wuhan. His current research interests include electronic circuits and hardware security.

LINGCHAO ZHANG was born in Zhejiang, China, in 1995. He received the B.S. degree in electrical engineering and automation from the Huazhong University of Science and Technology at Wuchang Branch, Wuhan, China, in 2017. He is currently pursuing the M.S. degree with the Hubei University of Technology, Wuhan. His current research interests include electronic circuits.

GAOJUN CHI was born in Hubei, China, in 1997. He received the B.E. degree in electronic information engineering from the Hubei University of Technology, China, in 2019. He is currently pursuing the M.Eng. degree with the Guilin University of Electronics Technology. His current research interests include electronic circuits and polarization imaging.

QI GAO was born in Hubei, China, in 1999. He received the B.E. degree in electronics information engineering from the Hubei University of Technology, China. His current research interest includes the study of electronic circuit design and lightweight hash function.

LEIN HARN received the B.S. degree from National Taiwan University in 1977, the M.S. degree from the State University of New York at Stony Brook in 1980, and the Ph.D. degree from the University of Minnesota in 1984, all in electrical engineering. He joined the Department of Electrical and Computer Engineering, University of Missouri–Columbia, in 1984, as an Assistant Professor, and he joined the Computer Science and Telecommunication Program with the University of Missouri–Kansas City in 1986, where he was on development leave to work with the Racal Data Group in Florida for a year.

181762