electronics

MDPI

Article

A Novel Key Generation Method for Group-Based Physically Unclonable Function Designs

Saeed Abdolinezhad [1,*], Lukas Zimmermann [2] and Axel Sikora [1,2]

[1] Institute of Reliable Embedded Systems and Communication Electronics (ivESK), Offenburg University of Applied Sciences, 77652 Offenburg, Germany; axel.sikora@hs-offenburg.de
[2] Hahn-Schickard-Gesellschaft für Angewandte Forschung e.V., 78052 Villingen-Schwenningen, Germany; lukas.zimmermann@hahn-schickard.de
* Correspondence: saeed.abdolinezhad@hs-offenburg.de; Tel.: +49-781-205-4960

Abstract: In recent years, physically unclonable functions (PUFs) have gained significant attraction in IoT security applications, such as cryptographic key generation and entity authentication. PUFs extract the uncontrollable production characteristics of different devices to generate unique fingerprints for security applications. When generating PUF-based secret keys, the reliability and entropy of the keys are vital factors. This study proposes a novel method for generating PUF-based keys from a set of measurements. Firstly, it formulates the group-based key generation problem as an optimization problem and solves it using integer linear programming (ILP), which guarantees finding the optimum solution. Then, a novel scheme for the extraction of keys from groups is proposed, which we call positioning syndrome coding (PSC). The use of ILP as well as the introduction of PSC facilitates the generation of high-entropy keys with low error correction costs. These new methods have been tested by applying them on the output of a capacitor network PUF. The results confirm the application of ILP and PSC in generating high-quality keys.

Keywords: physically unclonable function (PUF); PUF key generation; integer linear programming; syndrome coding

check for updates

Citation: Abdolinezhad, S.; Zimmermann, L.; Sikora, A. A Novel Key Generation Method for Group-Based Physically Unclonable Function Designs. *Electronics* **2021**, *10*, 2597. https://doi.org/10.3390/electronics10212597

Academic Editor: Taeshik Shon

Received: 8 October 2021
Accepted: 21 October 2021
Published: 24 October 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The internet of things (IoT) has already changed our way of living and the way we interact with a range of devices, such as cars, home appliances, and wearables [1]. We become more dependent on IoT devices every day. Connected devices that are available everywhere means that a security issue can lead to irrecoverable consequences. One important security aspect of IoT devices is the security of data, such as microprocessor program data, sensor data, or communication data [2]. Keeping in mind that IoT devices usually need to be lightweight and low cost, there is a need for simple, cheap, and strong security mechanisms to help with the provisioning of data security. Physically unclonable functions (PUF) have proven to be a potentially robust solution for such IoT security needs [3]. They leverage the inherent features of hardware to generate unique fingerprints for security applications.

Many studies have been conducted to improve the reliability, uniqueness, and entropy of PUFs. In this way, different materials and devices have been evaluated as PUF candidates. Most PUFs proposed in the literature are silicon-based, including arbiter [4,5], ring oscillator (RO) [6], static RAM (SRAM) and butterfly [7], latch [8], and flip-flop [9] PUFs. Moreover, some designs based on other materials and technologies have also been presented, such as printed electronics (PE) [10], touchscreens [11], solar cells [12], photodiodes [13], and the electric signals of personal computers [14].

This work aims to maximize the length of cryptographic keys derived from PUF responses while keeping the cost of error correction low. More specifically, we propose an optimum key generation method for the cases where the raw output of the PUF is a

set of analog measurements. The so-called grouping algorithm has been proposed in the literature as a means of deriving longer keys from a set of PUF measurements. However, no evidence is provided to show that the grouping algorithm finds the optimum solution to maximize the length of the generated key. In this study, the problem of maximizing PUF key length is formulated as an integer linear programming (ILP) optimization problem. It will be shown that ILP is able to find the optimum solution for the problem and maximize the key length. Moreover, a new syndrome coding method is proposed to improve the entropy of the generated key while lowering the number of error bits.

This paper is organized as follows. Section 2 covers the background and the state of the current literature. Section 3 explains the proposed solution for group-based key generation as well as positioning syndrome coding. Finally, in Section 4, the results of applying the proposed methods on real PUF datasets are presented.

2. Background and Previous Works

PUF studies fall into several categories. Some of them examine different materials and propose novel PUF designs. Several works look for new applications for PUFs. There are also studies which try to evaluate PUF security levels by performing attacks, while some researchers propose methods to make PUFs more resilient against such attacks. In this work, the focus is on the studies that propose methods to derive secure keys from PUF measurements, especially group-based key generation methods.

2.1. PUF Key Generation

In the PUF context, when using a device for secret key generation, higher entropy is desirable, which means extracting as many reliable bits from the PUF as possible. Changing environmental conditions, such as temperature and relative humidity, affect the reliability of PUFs. In general, reliable bits are independent of each other, and can be reproduced even under the impact of environmental variations. Moreover, ideally, the aging of the device should not have any effect on the reliability of the bits.

In the literature, several works focus on expanding the length of secret keys while generating reliable keys from PUF measurements under temporal and environmental variations. The authors of [15] proposed a one-out-of-eight masking scheme to generate one bit from eight ring oscillators. To increase the key length, [16] introduced a method to generate up to n-1 bits from n PUF measurements. Authors in [17] proposed the longest increasing sequence-based algorithm (LISA) to enhance secret key generation by grouping PUF measurement results. They have also introduced compact syndrome coding (CSC) to generate keys from the groups that resulted from the LISA. However, the complexity of the proposed grouping algorithm, the presence of bias in the generated keys, and the high number of error bits caused by temporal variations are some drawbacks of their work. Their work later has been improved by simplifying the grouping algorithm and combining it with Kendall syndrome coding (KSC) to generate longer keys with fewer bit errors [18]. However, this study also fails to find the optimum solution for the group-based key generation method. Our proposed key generation method, which is explained later in this paper, targets the limitations of these two works.

2.2. Group-Based Key Generation

The main idea of group-based key generation is to compare multiple PUF measurement values, rather than comparing them pairwise, in order to generate longer secret keys. The idea has been initially proposed in [17] using the LISA. Suppose that there is a group of four PUF measurement values, $g_1 = \{f_1, f_2, f_3, f_4\}$. A simple pairing algorithm can generate two pairs from these four elements, which leads to the generation of two secure bits. However, if all four elements are compared together, $4! = 24$ different orderings are possible. As shown in Figure 1, by assigning a binary code to each ordering according to CSC, five bits can be generated from these four elements. Therefore, group-based key generation is able to improve the entropy of the generated key compared to the pairwise method.

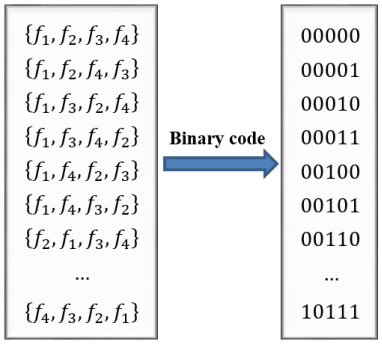


Figure 1. Mapping of different orderings to binary codes according to CSC.

The ideal case for key generation is to put all the PUF measurements in the same group. As an example, putting six PUF measurements in the same group leads to 720 different ordering possibilities, which can be encoded in 10 bits. Conversely, putting these PUF measurements in two groups of three elements generates only six bits. However, PUF measurements do not always give the same results under different environmental conditions. That means that if two PUF measurements in the same group have a very close value, after remeasuring, their order in the group might swap. Hence, a generated key from a set of PUF measurements may not be reproduced. To overcome this issue, only the PUF measurements which do not change their order under different conditions may be put together in the same group. The LISA is proposed to create a partition between PUF measurements with the aim of maximizing the key length. The LISA requires measuring every PUF value at the two extreme environmental conditions. In order to avoid bit errors, the ordering of group elements under these different conditions should give the same result for both measurements.

The definition of the problem for the group-based key generation is as follows:

Take n measured values G and put them in m subgroups to form the partition P:

$$P = \left\{ \begin{array}{c} g_1 \\ g_2 \\ \vdots \\ g_m \end{array} \right\} = \left\{ \begin{array}{c} \{f_{11}, f_{12}, \ldots\} \\ \{f_{21}, f_{22}, \ldots\} \\ \vdots \\ \{f_{m1}, f_{m2}, \ldots\} \end{array} \right\} \quad (1)$$

The following conditions apply when creating the partition:

1. $g_i \cap g_j = \varnothing$
2. $g_1 \cup g_2 \cup \ldots \cup g_m = G$
3. if $f_{ji_{min}} < f_{jk_{min}}$, then $f_{ji_{max}} < f_{jk_{max}}$ for all elements in the same subgroup
4. $\left| f_{ji_{min}} - f_{jk_{min}} \right| > f_{\text{th}}$; $\left| f_{ji_{max}} - f_{jk_{max}} \right| > f_{\text{th}}$ for all elements in the same subgroup

Where f_{ji} and f_{jk} represent the ith and kth elements of the subgroup j, respectively. Moreover, min and max subscripts represent the minimum and maximum values of the corresponding element, respectively. Conditions 1 and 2 guarantee the formation of a partition, which indicates that all f_r values have been used. Conditions 3 and 4 ensure that there is a minimum distance between PUF measurements inside a subgroup so that by changing the environment, no swap occurs by comparing elements of the subgroup. Note that a threshold value, f_{th}, is defined to ensure a minimum distance between PUF measurements in the same group. The above problem should be solved to maximize the following cost function which indicates the number of generated bits:

$$\text{Max} \sum_{i=1}^{m} \log_2(|g_i|!) \quad (2)$$

Authors in [17] proposed using a grouping algorithm to solve the above problem. However, they failed to prove that their algorithm is able to find the optimum solution, i.e., the maximum key length.

In [18], a simpler grouping algorithm has been proposed, which requires only one measurement. To achieve this, the authors also defined a threshold value, which should be the minimum distance between each pair of elements in a group. This eliminates conditions 3 and 4 from the group-based key generation problem. The new problem is defined with the following conditions:

1. $g_i \cap g_j = \varnothing$
2. $g_1 \cup g_2 \cup \ldots \cup g_m = G$
3. $\left|f_{ji} - f_{jk}\right| > f_{th}$ for all elements in the same subgroup

Authors of this work also presented an algorithm for solving this problem which is called the constructive grouping algorithm (CGA). However, they have also not provided any evidence that the CGA finds the optimum solution.

2.3. Syndrome Coding

As discussed before, in [17], CSC has been proposed to generate binary keys from a group of PUF measurements (Figure 1). Since CSC does not utilize all possible numbers that can be generated from n number of bits (in the case of five bits it only uses the range 0–23 and it never uses 24–31) the number of 0s in a key generated using CSC tends to be larger than the number of 1s, which causes bit aliasing. This makes the keys non-random and vulnerable to modeling attacks. To overcome this issue, one can shift the above range to 4–27 to make it symmetric with respect to the whole range. However, another problem with CSC is that a swap in the ordering due to temporal variations will cause several bit errors which increases the cost of error correction. For example, it can be seen in Figure 1 that a swap between f_1 and f_2 leads to the key 00110, which suffers from two bits error compared to the original key 00000. Moreover, in order to implement CSC in an IoT device, for each group length, a look-up table similar to Figure 1 should be stored. This increases the storage costs of the IoT device.

To solve the above issues with CSC, Kendall syndrome coding (KSC) has been introduced in [18]. The idea of KSC is to compare all the elements in a subgroup pairwise. Suppose subgroup g_i has k elements. The result of applying KSC to g_i is derived from Equation (3):

$$S_{g_i} = \delta(1,2)\delta(1,3)\ldots\delta(1,k)\delta(2,3)\ldots\delta(2,k)\ldots\delta(k-1,k) \quad (3)$$

where $\delta(i, j) = 0$ if $f_i < f_j$, and $\delta(i, j) = 1$ otherwise. The main advantage of KSC is that, in the case of a flip between two adjacent f_r, it limits the number of bit errors to one, because only the result of comparison between the swapped values will change. However, KSC reduces the entropy of the generated key by introducing redundancy. For example, suppose that KSC is being applied on the three elements f_1, f_2, f_3, and the ordering of the elements is $f_3 > f_1 > f_2$. When pairs (1,2) and (1,3) are compared in a triple, the result of the pair (2,3) comparison is already revealed, and the third generated bit does not add any entropy to the output key. This makes the PUF prone to security attacks.

3. Proposed Key Generation Method

3.1. Positioning Syndrome Coding

To overcome the abovementioned coding issues, we propose a novel coding scheme and name it positioning syndrome coding (PSC). In this scheme, when the elements of subgroup g_i are measured and ordered by their values, we take the elements one by one and generate a code based on the position of each element in the ordered subgroup. For example, suppose $g_1 = \{f_1, f_2, f_3, f_4\}$ and $f_2 > f_4 > f_3 > f_1$. The ordered subgroup will be $g_{1ordered} = \{f_2, f_4, f_3, f_1\}$. The position of f_1 in $g_{1ordered}$ from four possibilities is four. One can encode these four possibilities into two bits as 00, 01, 11, and 10. Therefore, 10 is

assigned to f_1, as it is shown in step 1 in Figure 2. Note that for encoding the position values, Gray code has been used. For the next element, f_2, there are three possibilities in the ordering, since position four is already determined to be f_1. According to step 2 in Figure 2, the three positioning possibilities for f_2 are encoded as 00, 01, and 11. Since it is the largest value in the remaining elements of the group $\{f_2, f_3, f_4\}$, the code 00 is assigned to f_2. Element f_3 is in the second position from the two remaining positions. These two positions are encoded with one bit as 0 and 1 (step 3 in Figure 2). Therefore, 1 is assigned to f_3. Finally, since the position of the last element, f_4, is already determined by positioning the other elements, no code can be generated from this element and it is automatically positioned as seen in step 4 in Figure 2. By concatenating the generated codes, the generated key from subgroup g_1 is key_{g_1} = 10,001.

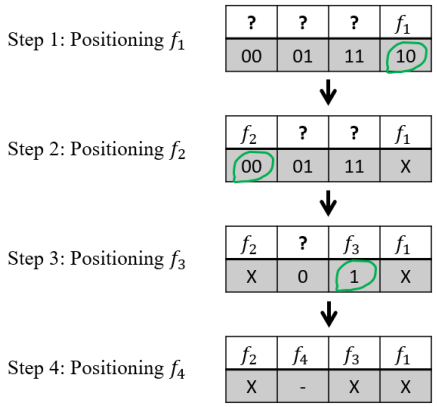


Figure 2. An example of PSC application to generate key from a subgroup with four elements.

The fact that the placement result of each element is independent of that of the previously placed elements indicates that there will be no redundancy issue in the final key. Moreover, the employment of Gray code to encode the placement result lowers the error correction cost because every flip between two adjacent elements in the ordering will cause only one bit error. In the above example, suppose that f_2 and f_4 flip in the measurement. The final key will be 10011, which only differs one bit from the original key. It is worth noting that using the conventional error correction methods, e.g., BCH codes, along with PSC leads to reproducible secret keys with low costs.

Another benefit of employing Gray code is that it reduces the bit aliasing of the generated key. The positioning of a specific number of elements produces more 0s, while another number of elements tends to produce more 1s. In the long run these tendencies cancel out each other, preventing bit aliasing. For example, positioning five elements produces more 0s because the first five Gray codes contain more 0s than 1s. On the other hand, six elements produce the same number of 1s and 0s, seven elements produce more 1s, eight elements produce the same number of 1s and 0s, and so on.

3.2. Integer Linear Programming

The length of the generated key as a result of applying PSC to a single subgroup is calculated by:

$$L_{g_i} = \sum_{j=2}^{|g_i|} [\log_2 j] \tag{4}$$

where $|g_i|$ is the number of elements of subgroup g_i. Note that in Equation (4), the length of the key is the summation of the bits generated by positioning each element of the subgroup. The length of the final key from a partition with m subgroups is calculated by:

$$L_{key} = \sum_{i=1}^{m} \sum_{j=2}^{|g_i|} [\log_2 j] \tag{5}$$

The goal of the group-based key generation is to maximize L_{key} and increase the PUF secret key length. This cost function, in addition to the previously defined constraints, form an optimization problem. Here, this problem is formulated and solved using an ILP optimization method. This method formulates and solves a problem with a linear cost function and constraints and integer decision variables. The generic form of an ILP problem [19] is:

$$\min_x C^T x$$

$$\text{Subject to:} \begin{cases} Ax \leq B \\ A_{eq}\, x = B_{eq} \\ lb \leq x \leq ub \end{cases} \tag{6}$$

where x is the decision variables vector. This generic form of the problem consists of inequality constraints, equality constraints, and the boundaries of the decision variables.

In a group-based key generation problem, n elements of group G form partition P with m subgroups. Each element can be the member of only one subgroup. The decision variables are defined by:

$$x_{ji} = \begin{cases} 1 & , f_i \in g_j \\ 0 & , f_i \notin g_j \end{cases} \tag{7}$$

Figure 3 shows the relationship between the elements, subgroups, and decision variables.

	f_1	f_2	⋯	f_n
g_1	x_{11}	x_{12}	⋯	x_{1n}
g_2	x_{21}	x_{22}	⋯	x_{2n}
⋮	⋮	⋮	⋱	⋮
g_m	x_{m1}	x_{m2}	⋯	x_{mn}

Figure 3. The relationship between the elements (f_i), subgroups (g_j), and decision variables (x_{ji}).

Problem constraints are categorized into two main categories:

1. Each measured value cannot be the member of more than one subgroup:

$$x_{1i} + x_{2i} + \ldots + x_{mi} \leq 1\,;\, 1 \leq i \leq n \tag{8}$$

2. If the distance between two PUF measurements is less than the threshold value, they cannot be members of the same subgroup:

$$\text{if } |f_i - f_k| < f_{\text{th}} \rightarrow x_{ji} + x_{jk} \leq 1 \tag{9}$$

The first category adds n inequality constraints to the problem. The number of constraints in the second category depends on the measured values of the elements. Note that, while being out of the scope of this work, care must be taken when choosing f_{th}, since it should consider the effect of environmental conditions, such as temperature and humidity, as well as the effect of aging on the PUF measurements. However, it is inevitable that in some conditions, even the elements that are carefully distanced by f_{th} swap and

cause bit errors. That makes the employment of error correction methods, such as BCH codes, necessary.

Obviously, it is not possible to formulize Equation (6) as a linear function of the decision variables. However, one can find an equivalent linear cost function which, when optimized, guarantees the maximum length of the key. It should be noted that putting as many elements in the first group as possible yields longer keys. For example, the optimum case for six elements is putting all of them in one subgroup to generate 11 bits, while all other grouping possibilities yield shorter keys (Figure 4). Therefore, the cost function can be defined in a way that the ILP solver first tries to make the first subgroup as large as possible, then from the remaining elements finds the second largest subgroup, and so on. It is formulized as a function of decision variables:

$$C^T x = c_1(x_{11}+x_{12}+\ldots+x_{1n})+c_2(x_{21}+x_{22}+\ldots+x_{2n})+\ldots+c_m(x_{m1}+x_{m2}+\ldots+x_{mn}) \quad (10)$$

$$c_j = -\frac{m-(j-1)}{m}\ ;\ 1 \leq j \leq m \quad (11)$$

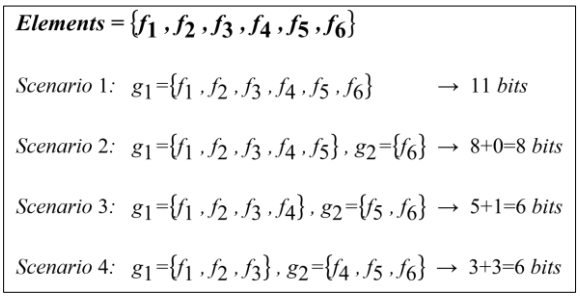


Figure 4. Different scenarios for generating key from a group of six elements using PSC.

Equations (11) and (12) force the ILP solver to first make the first subgroup as large as possible, then the second subgroup, and so on. The negative sign for the c_j coefficients is due to the fact that the solver tries to minimize the cost function. One benefit of ILP is that while its solution gives the key with the highest entropy, the key is still as robust as any shorter key produced by other group-based methods, as long as the threshold value is the robustness and reliability anchor. This is because ILP respects the threshold value throughout the problem constraints when looking for the solution, equal to other methods.

The final form of an ILP problem for group-based key generation is shown in Equation (12). When solved, ILP gives the decision variables x_{ji}. Using x_{ji} values, the elements of every group are determined according to Figure 3. When partition P is formed, PSC is applied to every subgroup and the derived bit sequences are concatenated to generate the final key. It has been proven that ILP is an NP-complete problem [19], which means that when a solution is found for the problem, it can be verified quickly that the solution is correct and "optimum". Therefore, employing ILP guarantees that the generated key is maximized with regard to the problem constraints. It should be noted that for each PUF, the ILP is applied only once at the enrollment phase to determine partition P, while PSC is performed every time the key needs to be derived in the regeneration phase.

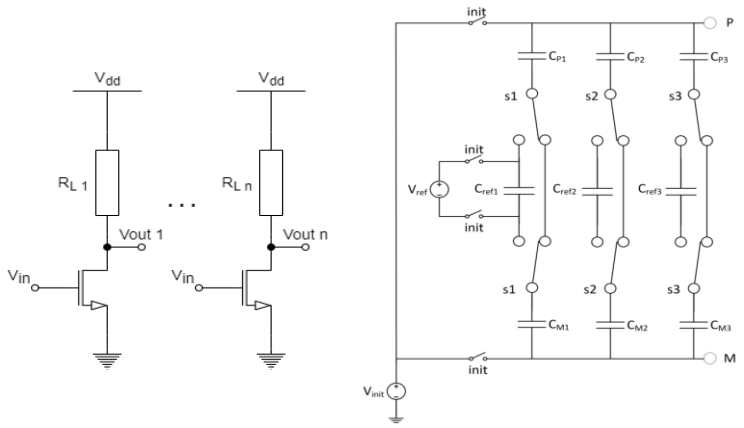
$$\min_{x} \begin{bmatrix} c_1 & | & c_2 & | & \cdots & | & c_m \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{12} \\ \vdots \\ x_{1n} \\ \hline x_{21} \\ x_{22} \\ \vdots \\ x_{2n} \\ \hline \vdots \\ \hline x_{m1} \\ x_{m2} \\ \vdots \\ x_{mn} \end{bmatrix} \quad \text{Subject to :} \begin{cases} \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_L \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{12} \\ \vdots \\ x_{1n} \\ \hline x_{21} \\ x_{22} \\ \vdots \\ x_{2n} \\ \hline \vdots \\ \hline x_{m1} \\ x_{m2} \\ \vdots \\ x_{mn} \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}; \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \leq \begin{bmatrix} x_{11} \\ x_{12} \\ \vdots \\ x_{1n} \\ \hline x_{21} \\ x_{22} \\ \vdots \\ x_{2n} \\ \hline \vdots \\ \hline x_{m1} \\ x_{m2} \\ \vdots \\ x_{mn} \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \end{cases} \tag{12}$$

4. Results and Discussions

4.1. Datasets for Testing

In order to validate the findings of this work, the proposed methods have been applied to data from two different types of PUF. These datasets were collected in previous studies on PUF at our institution. They were measured several times in different environmental conditions in order to analyze the effect of environment on the outputs.

The first dataset is from DiffC-PUF [3], which utilizes a resistor-transistor-logic inverter array (Figure 5a). By applying the same supply voltage and measuring the output voltage of all inverters, a set of PUF measurements is produced. The manufacturing variations of transistors and resistors makes the output voltage of every inverter unique. DiffC-PUF includes 80 inverters, and the output voltage of each inverter has been measured 20 times and the average voltage is calculated to reduce the measurement error.



(**a**) (**b**)

Figure 5. The circuit of: (**a**) a switched capacitor network PUF; (**b**) a DiffC PUF.

The second PUF being used for evaluation is the switched capacitor network PUF (SCN-PUF) [20], which contains several capacitor networks in its circuit (Figure 5b). At the initial time, all reference capacitors C_{ref} are charged with the same reference voltage, V_{ref}. Then, C_{ref} capacitors are connected to the circuit one by one by enabling the corresponding switch s_i, and then discharged into the circuit. By choosing different permutations of the reference capacitors being connected to the circuit, different output voltages (V_{PM} in Figure 5b) are generated due to uncontrollable capacitance differences between seemingly similar capacitors. The employed PUF for this work consists of eight capacitor networks and produces $8! = 40320$ unique voltage values, from which we have randomly chosen 100 voltage values.

4.2. Key Generation

In order to evaluate the proposed ILP, its output has been compared with the CGA outputs for the DiffC-PUF and the SCN-PUF. After the derivation of group data using ILP and CGA, the output key is generated using PSC. Figures 6 and 7 show the number of reliable bits generated from these two methods for DiffC-PUF and SCN-PUF, respectively. The results show a roughly 10 percent improvement in the number of output bits when using ILP. Increasing the threshold value in both PUFs results in stricter constraints when putting measurement values in the groups, which in turn reduces the length of the output key. It is worth noting that, by employing the non-group-based key generation methods mentioned in Section 2.1, the best result comes from the method proposed in [16], which produces 79 bit and 99 bit keys from DiffC-PUF and SCN-PUF, respectively. It is evident that on average, our method improved the key entropy approximately three times compared to the non-group-based key generation methods.

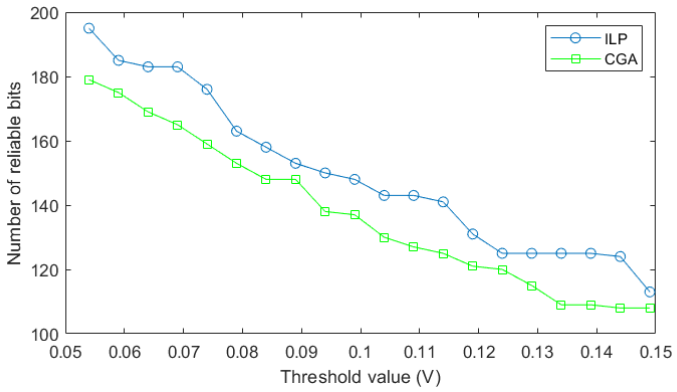


Figure 6. Reliable bits generated by ILP and CGA from DiffC-PUF.

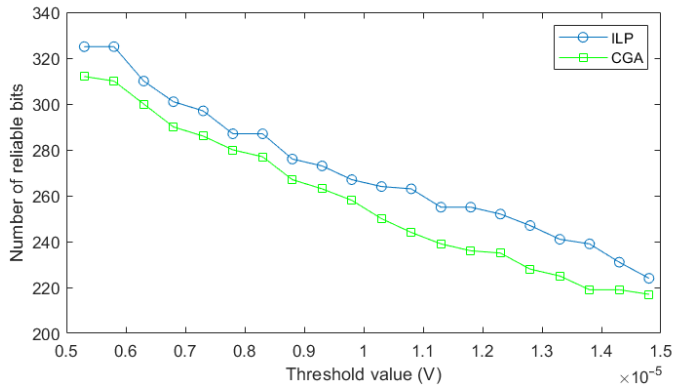


Figure 7. Reliable bits generated by ILP and CGA from SCN-PUF.

4.3. Positioning Syndrome Coding

For the assessment of PSC, after forming groups of PUF measurements by solving the group-based key generation problem with ILP, PSC and CSC have been applied to the group data to generate an output key. In Figure 8, the number of reliable bits versus different threshold values obtained by applying PSC and CSC for DiffC-PUF is depicted. They have also been applied to SCN-PUF data, and the results are shown in Figure 9. The results for both PUFs confirms the superiority of PSC over CSC for generating keys with more reliable bits. As explained above, since KSC produces low-entropy keys, and it has been eliminated from the comparisons in this section.

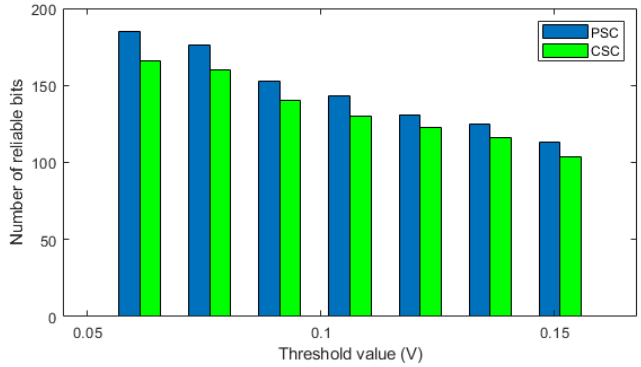


Figure 8. Comparison of generated bits using PSC and CSC for DiffC-PUF data.

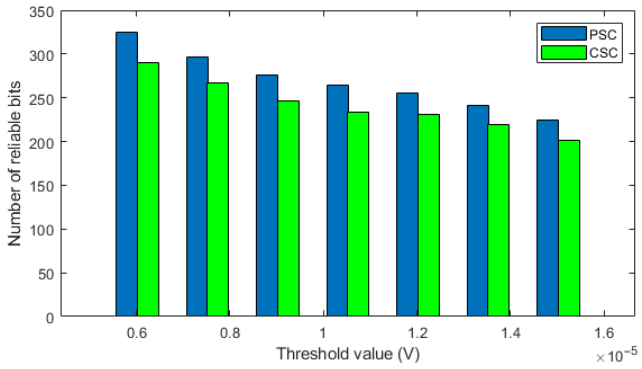


Figure 9. Comparison of generated bits using PSC and CSC for SCN-PUF data.

4.4. Reliability Analysis

As discussed before, when employing CSC, a single swap in the ordering of the members of a group can cause several error bits. On the other hand, by using PSC, each swap leads to only one bit error. In order to evaluate PSC, the reliability of the keys generated from this method have been compared with those generated by CSC. Figure 10 shows the reliability of the keys when generated using DiffC-PUF measurements by employing PSC and CSC in different threshold voltages. It is evident that the reliability has been improved by roughly 2%, reaching 98% when PSC is used. The datasets used for key generation in this work show significant variations with temperature [3,20]. However, this reliability analysis shows that even for the PUFs which are sensitive to environmental conditions, our proposed method offers close to ideal reliability results.

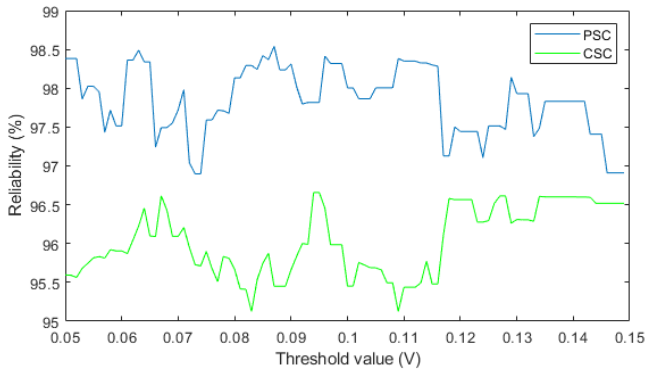


Figure 10. Comparison of the reliability of the keys generated by PSC and CSC for DiffC-PUF data.

5. Conclusions

In this work, we reformulated the group-based key generation problem using ILP which gives the optimum solution. It has been shown that it improves the output key length by around 10 percent compared to previous methods. Moreover, we have shown that with the proposed PSC method, an average reliability of 98% could be achieved, which improves the reliability by 2% compared to conventional methods. In the end, in order to confirm the findings of this work, they have been evaluated using the datasets from two different PUFs that can be found in the literature.

Author Contributions: Conceptualization, S.A.; methodology, S.A.; writing—original draft preparation, S.A.; writing—review and editing, A.S. and L.Z.; supervision, A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Abdolinezhad, S.; Schappacher, M.; Sikora, A. Secure wireless architecture for communications in a parcel delivery system. In Proceedings of the 2020 IEEE 5th International Symposium on Smart and Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS), Dortmund, Germany, 17–18 September 2020; pp. 1–6.
2. Asif, R.; Ghanem, K.; Irvine, J. Proof-of-PUF enabled blockchain: Concurrent data and device security for internet-of-energy. *Sensors* **2021**, *21*, 28. [CrossRef] [PubMed]
3. Scholz, A.; Zimmermann, L.; Gengenbach, U.; Koker, L.; Chen, Z.; Hahn, H.; Sikora, A.; Tahoori, M.B.; Aghassi-Hagmann, J. Hybrid low-voltage physical unclonable function based on inkjet-printed metal-oxide transistors. *Nat. Commun.* **2020**, *11*, 1–11.
4. Lee, J.W.; Lim, D.; Gassend, B.; Suh, G.E.; van Dijk, M.; Devadas, S. A technique to build a secret key in integrated circuits for identification and authentication applications. In Proceedings of the 2004 Symposium on VLSI Circuits. Digest of Technical Papers, Honolulu, HI, USA, 17–19 June 2004; pp. 176–179.
5. Lim, D.; Lee, J.W.; Gassend, B.; Suh, G.E.; Van Dijk, M.; Devadas, S. Extracting secret keys from integrated circuits. *IEEE Trans. Very Large Scale Integr. Syst.* **2005**, *13*, 1200–1205.
6. Gassend, B. Physical Random Functions. Master's Thesis, Massachusetts Institute of Technology, Boston, MA, USA, 2003.
7. Guajardo, J.; Kumar, S.S.; Schrijen, G.-J.; Tuyls, P. FPGA Intrinsic PUFs and their use for IP protection. In Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems, Vienna, Austria, 10–13 September 2007.
8. Su, Y.; Holleman, J.; Otis, B. A 1.6 pJ/bit 96% stable chip-ID generating circuit using process variations. In Proceedings of the Digest of Technical Papers of the 2007 IEEE International Solid-State Circuits Conference (ISSCC 2007), San Francisco, CA, USA, 11–15 February 2007; pp. 406–611.
9. Maes, R.; Tuyls, P.; Verbauwhede, I. Intrinsic PUFs from flip-flops on reconfigurable devices. In Proceedings of the 3rd Benelux Workshop on Information and System Security (WISSec 2008), Eindhoven, The Netherlands, 13–14 November 2008; Volume 17, p. 2008.
10. Zimmermann, L. Printed Electronics-Based Physically Unclonable Functions for Lightweight Security in the Internet of Things. Ph.D. Thesis, Karlsruhe Institute of Technology, Karlsruhe, Germany, 2020.
11. Scheel, R.A.; Tyagi, A. Characterizing composite user-device touchscreen physical unclonable functions (PUFs) for mobile device authentication. In Proceedings of the 5th International Workshop on Trustworthy Embedded Devices, Denver, CO, USA, 16 October 2015; pp. 3–13.
12. Rosenfeld, K.; Gavas, E.; Karri, R. Sensor physical unclonable functions. In Proceedings of the 2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), Anaheim, CA, USA, 13–14 June 2010; pp. 112–117.
13. Aponte, E. A Study on Energy Harvesters for Physical Unclonable Functions and Random Number Generation. Ph.D. Thesis, Virginia Tech, Blacksburg, VA, USA, 2017.
14. Sikora, A.; Nyemkova, E.; Lakh, Y. Accuracy improvements of identification and authentication of devices by EM-measurements. In Proceedings of the IEEE 5th International Symposium on Smart and Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS), Dortmund, Germany, 17–18 September 2020.
15. Suh, G.E.; Devadas, S. Physical unclonable functions for device authentication and secret key generation. In Proceedings of the 44th ACM/IEEE Design Automation Conference, San Diego, CA, USA, 4–8 June 2007; pp. 9–14.
16. Maiti, A.; Schaumont, P. Improving the quality of a physical unclonable function using configurable ring oscillators. In Proceedings of the International Conference on Field Programmable Logic and Applications, Prague, Czech Republic, 31 August–2 September 2009; pp. 703–707.
17. Yin, C.E.; Qu, G. Lisa: Maximizing RO PUF's secret extraction. In Proceedings of the 2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), Anaheim, CA, USA, 13–14 June 2010; pp. 100–105.

18. Yin, C.E.; Qu, G.; Zhou, Q. Design and implementation of a group-based RO PUF. In Proceedings of the Conference on Design, Automation and Test in Europe, Grenoble, France, 18–22 March 2013; pp. 416–421.
19. Papadimitriou, C.H.; Steiglitz, K. *Combinatorial Optimization: Algorithms and Complexity*; Prentice-Hall, Inc.: Upper Saddle River, NJ, USA, 1982.
20. Müller, K.U.; Stanitzki, A.; Kokozinski, R. A 47 F 2/bit charge-sharing based Sequence-dependent PUF with a permutative challenge. In Proceedings of the 2020 International Conference on Omni-layer Intelligent Systems (COINS), Barcelona, Spain, 31 August–2 September 2020; pp. 1–6.