Physically Related Functions: A New Paradigm for Light-weight Key-Exchange

Durba Chatterjee[1], Harishma Boyapally[1], Sikhar Patranabis[2], Urbi Chatterjee[3], Debdeep Mukhopadhyay[1], Aritra Hazra[1]
1. Indian Institute of Technology Kharagpur, 2. ETH Zurich, 3. Indian Institute of Technology Kanpur

ABSTRACT

In this paper, we propose a novel concept named *Physically Related Function* (PReF) which are devices with hardware roots of trust. It enables secure key-exchange with no pre-established/embedded secret keys. This work is motivated by the need to perform key-exchange between lightweight resource-constrained devices. We present a proof-of-concept realization of our contributions in hardware using FPGAs.

Keywords: Boolean functions, Key-exchange protocol, Physically Related Functions

1 INTRODUCTION

The most straightforward way of achieving secure key-exchange is via standard cryptographic techniques, where an initially exchanged secret key allows the devices to encrypt all ensuing communication. However, these devices are *far too resource-constrained* to either support the heavy protection mechanisms against invasive and semi-invasive attacks [12] associated with secure key-storage (as in symmetric-key primitives) or does not have provisions for renewal of certificates (as in asymmetric-key primitives). This is a scenario frequently encountered today with the widespread advent of Internet of Things (IoT) and Cyber-Physical Systems (CPS). Recently, Amazon, Uber, and UPS and other major corporations have announced plans to launch commercial autonomous drone operations [4, 7]. The widespread deployment of such technologies would potentially involve millions of devices communicating with each other. This makes it essential to design light-weight protocols for secure communication that can scale to a large number of devices.

In this paper, we introduce novel hardware primitives called *Physically Related Functions* (abbrv. *PReFs*). We present a PReF-based on-the-fly key-exchange scheme, without the need to store any secret key or the need to contact any trusted third party. We present a proof-of-concept (PoC) realization of PReFs in hardware using 84 separate Xilinx Artix 7 FPGAs.

2 PHYSICALLY RELATED FUNCTIONS

In this section, we present a brief description of PReFs and a PReF-based key-exchange scheme. A pair of PReFs $(\mathbf{D}_\mathsf{A}, \mathbf{D}_\mathsf{B})$ physically implement the functions $(f_\mathsf{A}, f_\mathsf{B})$ with input space $\mathcal{X}$ and output space $\mathcal{Y}$ such that there exists a specific subset of inputs $\mathcal{X}_{\mathsf{A,B}} \subseteq \mathcal{X}$ such that the output behaviors of the functions f_A and f_B on each input in $\mathcal{X}_{\mathsf{A,B}}$ are correlated with respect to some distance metric (eg. Hamming Distance (HD)). On the other hand, for any $x \in \mathcal{X} \setminus \mathcal{X}_{\mathsf{A,B}}$ any probabilistic poly-time bounded algorithm that only has access to an implementation of f_A and does not not have (even black-box) access to an implementation of f_B cannot distinguish $f_\mathsf{B}(x)$ from random. This is defined as the pseudorandomness property of PReFs.

$\mathbf{D}_\mathsf{A}$: Round 1

(1) Samples input $x \in \mathcal{X}_{\mathsf{A,B}}$ and key $r \xleftarrow{R} \{0,1\}^k$.
(2) Computes
$$y_\mathsf{A} = f_\mathsf{A}(x), \quad \mathbf{A} = \mathsf{E}(r) + y_\mathsf{A}.$$
(3) Sends $\langle x, \mathbf{A} \rangle$ to $\mathbf{D}_B$.

$\mathbf{D}_\mathsf{B}$: Round 1

(1) Computes
$$y_\mathsf{B} = f_\mathsf{B}(x), \quad r' = \mathsf{D}(\mathbf{A} + y_\mathsf{B}).$$

Figure 1: Basic Key Exchange

PReF-based Key Exchange Scheme

Now, we develop a key-exchange scheme for secure communication between two devices, each embodying a PReF instance. Our protocol precludes the usage of long-term secure key storage, as is usually the case with a large class of key-exchange protocols based on traditional cryptographic approaches [1, 10]. In other words, it avoids not only the need for dedicated key storage but also the associated countermeasures for preventing potential physical attacks (both invasive and non-invasive) targeting such key storage. It is also superior to the key-exchange schemes based on alternative primitives (such as Physically Unclonable Functions (PUFs)) which are asymmetric in nature requiring one (or both) device(s) to perform complex computations or require trusted third party during the protocol run. So, in a way, our scheme achieves the best of both worlds, especially in the context of lightweight resource-constrained devices.

Protocol Description. Let $(\mathbf{D}_\mathsf{A}, \mathbf{D}_\mathsf{B})$ be a pair of PReFs as described previously, such that $\mathcal{X} = \{0,1\}^m$ and $\mathcal{Y} = \{0,1\}^n$. These devices form a PReF pair over the input subset $\mathcal{X}_{\mathsf{A,B}} \subset \mathcal{X}$ such that for any $x \in \mathcal{X}_{\mathsf{A,B}}$, $\mathsf{HD}(f(x), g(x)) \leq \delta$. Let (E, D) be the encoding and decoding algorithms of an (n, k, δ) linear error correction code (ECC).

We present a basic PReF-based key exchange protocol as described in Fig. 1, that requires no computational resources beyond evaluating PReF outputs. It enables a key exchange between two PReF devices $\mathbf{D}_\mathsf{A}$ and $\mathbf{D}_\mathsf{B}$ with the *unique* "related" input set $\mathcal{X}_{A,B}$. The protocol involves a single round of communication between the devices and is considerably light-weight given that it only uses error-correcting codes in addition to PReFs.

Theoretical Implications. This protocol also has some interesting theoretical implications about the *computational power of* PReFs. It is well-known that no computationally secure key-exchange protocol (even multi-round) can be based in a black-box manner on purely symmetric-key cryptographic primitives such as pseudo-random functions or symmetric-key encryption [8]. The fact that we can bypass this impossibility result using only PReFs and no additional cryptographic primitives/trusted parties indicates that PReFs are, in fact, more powerful than simple symmetric-key cryptoprimitives. This makes PReFs an interesting object of study from a cryptographic standpoint.

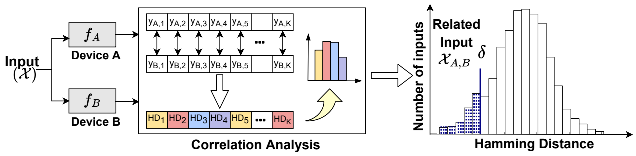


Figure 2: Overview of output-correlation estimation between two devices embodying a pair of ReFs.

3 REALIZING PREFS IN HARDWARE

In this section, we establish the feasibility of embedding "related" functions into physical devices , i.e. the feasibility of realizing PReFs in hardware. We show that the notion of correlation with respect to Hamming distance (HD) allows us to obtain a set of "related" inputs for any pair of random Boolean functions.

Correlation Analysis of Boolean Functions. In Boolean theory, the cross-correlation function [11] is used to study the cryptographic properties of Boolean functions. The cross-correlation function for two Boolean functions $f : \mathcal{X} \to \mathcal{Y}$ and $g : \mathcal{X} \to \mathcal{Y}$ where $\mathcal{X} = \{0,1\}^n$ and $\mathcal{Y} = \{0,1\}$ calculates the correlation between its outputs over the complete input set $\mathcal{X}$. It is given by:

$$\mathbf{C}_{f,g} = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} (-1)^{f(x) \oplus g(x)}$$

and its value lies in $[-1, 1]$. Now, if the functions f and g are chosen *uniformly at random* from the space of all n-bit Boolean functions, their outputs will be statistically uncorrelated. Hence, $C_{f,g} = 0$.

For this work, we have exploited the fact that even if f and g are statistically uncorrelated, there exist some inputs for which both have the same outputs. We can split $\mathcal{X}$ into two disjoint subsets $\mathcal{X}_0$ and $\mathcal{X}_1$ such that $f(x) = g(x)\ \forall x \in \mathcal{X}_0$ and $f(x) \neq g(x)\ \forall x \in \mathcal{X}_1$. Therefore, we can say that f and g are related over the subset $\mathcal{X}_0$. For a more generic analysis, consider the functions $f_A, f_B : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X} = \{0,1\}^n$ and $Y = \{0,1\}^m$. To find the input subset over which f_A and f_B are related, we split the input set into disjoint subsets $\{\mathcal{X}_0, \mathcal{X}_1, \cdots, \mathcal{X}_m\}$, such that for any input belonging to subset $\mathcal{X}_i$, the HD between the function outputs is i. Note that the HD is calculated over the m-bit response.

Let $f_{\mathsf{A}}(x)[i]$ denote the i^{th} bit of the output of f_{A} for an input x and let $\mathbf{q}$ be the probability with which $f_{\mathsf{A}}(x)[i] \oplus f_{\mathsf{B}}(x)[i] = 1$ occurs, for any $i \in [1, m]$. The probability that $\mathsf{HD}(f_{\mathsf{A}}(x), f_{\mathsf{B}}(x))$ takes the value $j \in [0, m]$ can be given as:

$$\Pr\big[\mathsf{HD}(f_{\mathsf{A}}(x), f_{\mathsf{B}}(x)) = j\big] = \binom{m}{j} \mathbf{q}^j (1 - \mathbf{q})^{m-j}. \tag{1}$$

From the above equation, it is evident that the frequency distribution calculated using the HD follows a Binomial distribution. Let $\epsilon_{\mathsf{A,B}}$ denote the probability with which $\mathsf{HD}(f_{\mathsf{A}}(x), f_{\mathsf{B}}(x)) \leq \delta$ holds. We can calculate $\epsilon_{\mathsf{A,B}}$ as:

$$\epsilon_{\mathsf{A,B}} = \Pr\big[\mathsf{HD}(f_{\mathsf{A}}(x), f_{\mathsf{B}}(x)) \leq \delta\big] = \sum_{j=0}^{\delta} \binom{m}{j} \mathbf{q}^j (1 - \mathbf{q})^{m-j}. \tag{2}$$

Then the size of the input subset $\mathcal{X}_{\mathsf{A,B}} \subseteq \mathcal{X}$, over which the outputs of f_{A} and f_{B} have HD atmost δ is given as:

$$\big|\mathcal{X}_{\mathsf{A,B}}\big| = \epsilon_{\mathsf{A,B}} \big|\mathcal{X}\big|. \tag{3}$$

A pair of functions f_{A} and f_{B} are said to be related if $\mathcal{X}_{\mathsf{A,B}} \neq \emptyset$ and $\epsilon_{\mathsf{A,B}} > 0$. Thus with this notion of output-correlation, we can obtain "related" inputs for any pair of random Boolean functions.

Algorithm 1: Find $\mathcal{X}_{A,B}$ for a pair of functions (f_B, f_B)

Input: Random functions embedded in Devices: f_A, f_B Distance metric: δ

Output: Input subset: $\mathcal{X}_{A,B}$

Initialize variable $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ //Input variable which is to be generated by the solver
$\{m_A\} \leftarrow$ ML model of f_A
$y_A \leftarrow$ `getResponse`$(\mathbf{x}, m_A)$
$\{m_B\} \leftarrow$ ML model of f_B
$y_B \leftarrow$ `getResponse`$(\mathbf{x}, m_B)$
s $\leftarrow$ SMT_Solver() //Initialization of SMT Solver
Add constraint $(\mathsf{HD}(y_A, y_B) \leq \delta)$ to s
$\mathcal{X}_{A,B} \leftarrow \{\}$
while SAT(s) = SOLVED **do**
  $solution \leftarrow$ s.model()
  $\mathcal{X}_{A,B} \leftarrow \mathcal{X}_{A,B} \cup solution$
  Add constraint $(\mathbf{x} \neq solution)$ to s
end
return $\mathcal{X}_{A,B}$

Figure 3: "Related" input identification for PReF pairs

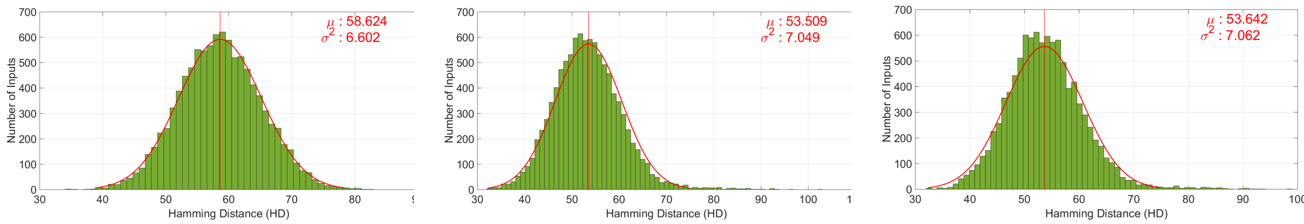
The overview of the correlation analysis for a pair of PReFs is illustrated in Fig. 2.

Sampling Related Inputs for PReF Devices. Equipped with this analysis, our aim is to realize PReFs using hardware devices. For this, we simply design Boolean functions as hardware circuits and rely on the internal variability of every device to introduce unpredictable yet repetitive randomness in the circuit behaviour. The next step is to identify the inputs over which the functions produce "related" outputs. Since the functionality is fixed and unpredictable, we resort to a *machine-learning and SAT [6] based reverse-engineering* approach described as follows. The idea is to generate accurate machine learning models of each PReF instance in setup phase prior to deployment, and use these models to obtain "related" challenges. The steps are described as follows:

Related Input Set Identification. We use some of the well-known learning techniques such as Logistic Regression (LR) [9], Support Vector Machine (SVM) [5], and Random Forest (RF) [2] algorithms to build, for each elementary component of PReF (hardware-embedded function or the Strong PUF embedded in hardware device), a corresponding *mathematical model* that closely approximates the input-output behaviour of the component. Since the modelling is expected to be done at setup (i.e., at the device manufacturing stage), we crucially exploit the knowledge of the circuit-internals, in particular the input-output behaviour of each *intermediate stage* of the circuit, to build an overall model with close to 100% accuracy. After building the mathematical models $\mathbf{M}_{f_A}$ and $\mathbf{M}_{f_B}$ approximating the hardware-embedded functions f_A and f_B respectively. The next step is to formulate the problem of identifying the corresponding related set of inputs $\mathcal{X}_{A,B}$ into a Boolean satisfiability problem in terms of the model parameters for pair $(\mathbf{M}_{f_A}, \mathbf{M}_{f_B})$.

Mathematical Formulation of Models. Depending on the chosen machine learning algorithm, we formulate a mathematical expression for the output of functions (corresponding to each component) in terms of the obtained model parameters and the input.

Note that the resultant mathematical expression depends not only on the implemented circuit but the instance-specific system



(a) Pair 1 (b) Pair 2 (c) Pair 3

Figure 4: Frequency distribution of HD of different pairs of PUFs implemented in FPGAs. Each PUF returns a 224-bit response.

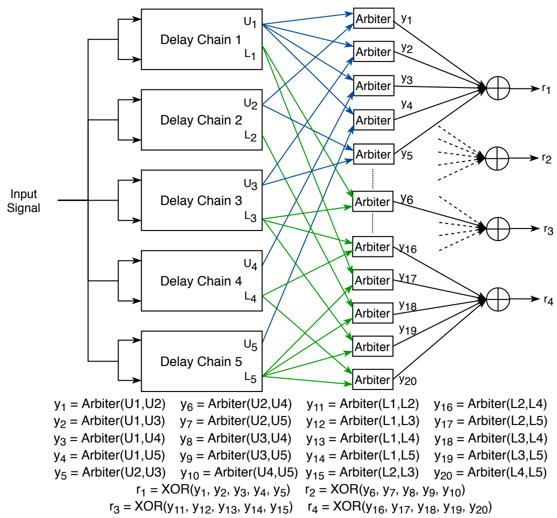


Figure 5: Schematic Representation of 5-4 DAPUF [3]

behaviour as well. Next, we represent the mathematical expressions approximating $\mathbf{M}_{f_A}$ and $\mathbf{M}_{f_B}$ in first-order logic and feed them to a standard SMT solver (for e.g., Z3 solver [6]) along with a constraint specifying the upper bound δ on the HD between two outputs. This is illustrated formally in Fig. 3. The solver converts it to a Boolean SAT formula using sophisticated simplification techniques. The SAT formulation is then sent to a underlying SAT solver that returns the inputs iteratively for which the models' outputs have at most δ HD. Thus, we have successfully retrieved a set of "related" inputs for (f_A, f_B) which are employed in the proposed key-exchange scheme.

4 EXPERIMENTAL EVALUATION

We present a proof-of-concept (PoC) realization of PReFs, using PUFs for embedding Boolean functions in hardware devices and a prototype evaluation of the PReF-based KE protocol. For the PoC realization of the PReF constructs, we deploy the 5-4 DAPUF design (refer Fig. 5) proposed in [3] in 84 Artix-7 FPGAs which takes 64-bit binary input and generates 4-bit binary output. The average bit-wise uniformity, uniqueness and bit-aliasing are 52.72%, 44.16% and 57.59% respectively. We characterise every PUF with 20K random inputs and calculate HD of the outputs for every pair. We observe that HD follows Binomial distribution, as shown in Fig. 4 which corroborates with our theoretical analysis presented in Sec. 3.

Identification of related inputs: In the setup phase, as the designers have access to the internal circuitry of the device, it can model each arbiter output (y) individually and compute the final response by XORing the predicted outputs. For our implementation, we use Logistic Regression to model the individual arbiters. The average modeling accuracy of all the arbiters is observed to be almost 100% for a test set of 20K challenge-arbiter output pairs. To generate "related" input subset for a pair of PReF instances, we use the mathematical model to formulate a satisfiability expression and feed it to SMT solver (programmed using Z3Py API in Python 2.7). The solver iteratively returns related challenges.

Next, we find the true positive rate (TPR) and false positive rate (FPR) of protocol. To calculate TPR, we randomly choose 1000 inputs over which a pair of related devices can communicate. We observe that the probability that two devices establish the same key is 100%, for all the 1000 inputs. We use the same inputs to find the probability (FPR) that an illegitimate device (not related over the chosen input subset) can successfully exchange the key with a legitimate device. The FPR of the protocol is observed to range from 2^{-90} to 2^{-110} for all the 1000 inputs.

5 CONCLUSION AND FUTURE WORK

In this paper, we initiated the study of *Cryptophasia in Hardware* using a novel concept called *Physically Related Functions* (PReFs) that have not been studied before to the best of our knowledge. We demonstrated how PReFs can be used to establish a key-exchange scheme with no pre-established secure channels and no secure storage for cryptographic keys. We established the feasibility of our proposal via concrete prototype implementations and extensive experimental implementations on Artix-7 FPGAs. As a future work, we will build authenticated key-exchange schemes for a network of resource-constrained devices, ensuring secure communication between any two devices.

REFERENCES

[1] Gildas Avoine, Sébastien Canard, and Loïc Ferreira. 2020. Symmetric-Key Authenticated Key Exchange (SAKE) with Perfect Forward Secrecy. In *Topics in Cryptology – CT-RSA 2020*, Stanislaw Jarecki (Ed.). Springer International Publishing, Cham, 199–224.

[2] Leo Breiman. 2001. Random forests. *Machine learning* 45, 1 (2001), 5–32.

[3] Urbi Chatterjee, Durga Prasad Sahoo, Debdeep Mukhopadhyay, and Rajat Subhra Chakraborty. 2018. Trustworthy proofs for sensor data using FPGA based physically unclonable functions. In *2018 DATE*. IEEE, 1504–1507.

[4] Casey Coombs. 2019. Amazon's Ambitious Drone Delivery Plans Take Shape. (2019). https://www.thedailybeast.com/with-prime-air-amazon-wants-to-deliver-packages-in-30-minutes-or-less-via-drone

[5] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning* 20, 3 (1995), 273–297.

[6] Leonardo De Moura and Nikolaj Bjørner. 2008. Z3: An efficient SMT solver. In *International conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 337–340.

[7] Lauren Feiner. 2019. Amazon debuts its new delivery drone. (2019). https://www.cnbc.com/2019/06/05/amazon-debuts-its-new-delivery-drone.html

[8] Russell Impagliazzo and Steven Rudich. 1989. Limits on the Provable Consequences of One-Way Permutations. In *ACM STOC*, David S. Johnson (Ed.). ACM, 44–61.
[9] Michael H Kutner, Christopher J Nachtsheim, John Neter, William Li, et al. 2005. *Applied linear statistical models*. Vol. 5. McGraw-Hill Irwin New York.
[10] Quanrun Li, Ching-Fang Hsu, Kim-Kwang Raymond Choo, and Debiao He. 2019. A Provably Secure and Lightweight Identity-Based Two-Party Authenticated Key Agreement Protocol for Vehicular Ad Hoc Networks. *Security and Communication Networks* 2019 (12 2019), 1–13. https://doi.org/10.1155/2019/7871067
[11] Palash Sarkar and Subhamoy Maitra. 2002. Cross-correlation analysis of cryptographically useful Boolean functions and S-boxes. *Theory of Computing Systems* 35, 1 (2002), 39–57.
[12] Sergei P. Skorobogatov. 2005. Semi-invasive attacks – A new approach to hardware security analysis. (2005).