Poster: Can We Trust PUF Manufacturers? Robust PUF-Based Authentication via Combiners

1st Edoardo Liberati, 2nd Pierpaolo Della Monica, 3rd Riccardo Lazzeretti, 4th Ivan Visconti

Sapienza University of Rome Rome, Italy e.liberati@diag.uniroma1.it

Sapienza University of Rome Rome, Italy dellamonica@diag.uniroma1.it

Sapienza University of Rome Rome, Italy lazzeretti@diag.uniroma1.it

Sapienza University of Rome Rome, Italy visconti@diag.uniroma1.it

***Abstract*—IoT devices have emerged as a significant attack vector, exemplified by the Mirai botnet and recent incidents involving compromised hardware. Protecting these devices is paramount, but their limited resources pose a significant challenge. One promising approach that has gained traction in recent years for device authentication is the use of Physical Unclonable Functions (PUFs), hardware components capable of efficiently generating randomness. However, the reliability of PUFs can be compromised by aging effects or malicious tampering during manufacturing, potentially undermining the protocols that depend on them. To address this issue, we propose a combiner mechanism that integrates multiple, distinct PUFs to ensure robustness against faulty or compromised units.**

***Index Terms*—physical unclonable function, combiner, IoT**

I. Motivation

In 2016, Mirai botnet has shut down a non negligible portion of Internet performing an overwhelming Distributed Denial of Service (DDoS) attack by leveraging on about 600,000 infected Internet of Things (IoT) devices including routers, printers, and cameras [1]. Also recent attacks made IoT devices perform unexpected and harmful behavior due to corrupted and malicious hardware installed on them [2]. These events taught us that the protection of IoT devices is crucial to prevent striking attacks.

Securing devices which usually have limited resources is not a trivial task. This holds particularly true considering that often standard cryptographic protocols cannot be executed on such devices. Thus, alternative solutions have been proposed to provide secure and lightweight cryptographic protocols. Among them, Physically Unclonable Functions (PUFs) [3] constitute efficient sources of randomness. These devices leverage nanoscale hardware variations introduced during the manufacturing process to generate unique and unpredictable values.

On input a challenge c, the output is a response $r = PUF(c)$. However, when supplied multiple times with the same challenge, the same PUF will output similar but potentially not identical responses. A natural post-processing step is the usage of fuzzy extractors [4]. These algorithms allow for the generation and reproduction of the same randomness starting from noisy sources, such as PUFs, as long as the PUF responses are close to each other, which again holds true for each PUF when stimulated with the same challenge. This reproduction procedure leverages public helper data crafted during the generation procedure: this additional information allows the regeneration of the generated value whilst leaking nothing about the underlying secret.

Responses' proximity can be exploited to set up an authentication protocol [5]. At the enrollment phase, the verifier sends a challenge to a prover equipped with a PUF and registers the challenge-response pair (CRP). At authentication time, the verifier sends the same challenge to the prover and compares the just received response with the stored one. If they are close to each other or if they match - in case of usage of a fuzzy extractor -, then the authentication is successful.

If, for some reason, the distance among the responses evaluated starting from the same challenges exceeds a certain threshold, i.e. they are no more close to each other, then the verifier cannot authenticate the prover. The main reasons for which a PUF may fail and provide distant responses are: (1) deterioration of the hardware itself, due to natural aging or damages; (2) malicious manufacturers delivering adversarial hardware (e.g., a stateful chip physically resembling a genuine PUF).

II. Contribution

We are developing a combiner which is able to guarantee a nominal execution of a PUF-based authentication protocol even in presence of faulty PUFs, either deteriorated or malicious.

Assume a prover is equipped with $n > 1$ PUFs and each of them is followed by a fuzzy extractor such that, on input a challenge c, the corresponding output is r. Without lack of security, we assume that all the PUFs are stimulated with the same c. Hence, the prover can compute n pairs in the form $(i, r_i(c))$, where i is indexing the PUF. Each point constitutes a finite field point in the 2D plane as depicted in Fig. 1. The n $(i, r_i(c))$ pairs uniquely identify a polynomial $p(x)$ of degree $n-1$ which can be computed through modular Lagrangian interpolation, as well as in Shamir Secret Sharing [6]. Then, the prover computes some other points lying on $p(x)$: (1) $sk = p(0)$ will be the final output of the combiner and, hence,

This work is supported by project SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU. The views are of the authors only, not of the funding entities.

of the PUF-based authentication secret; (2) $m < n$ random additional points in the form $(x, p(x))$, which represent additional information that permits to reconstruct the secret in presence of PUF failures. The verifier stores the challenge(s) c, all the m additional points, the hash of sk computed by means of a cryptographic hash function, and the helper data required by the PUFs' fuzzy extractors.

At authentication time, the prover receives all the values stored by the verifier. It then evaluates the challenge(s) against the n PUFs, acquires the correspondent responses, and supplies them to the fuzzy extractor. The prover obtains the n pairs $(i, r'_i(c))$ that undergo Lagrangian modular interpolation to identify the unique polynomial $p'(x)$, which is used to compute the secret $sk' = p'(0)$. At the end, the hash of the just computed sk' is compared with the hash of sk received from the verifier. Two are the possible scenarios.

If all the n PUFs are nominally working, then we expect all the n $(i, r'_i(c))$ pairs will match the initial $(i, r_i(c))$ pairs – failure can happen with negligible probability. Thus, their interpolation will lead to $p'(x) = p(x)$, and therefore $sk' = sk$, as well as their hashes. Considering negligible the probability of collisions, authentication is successful.

On the other hand, if at least one of the n $(i, r'_i(c))$ differs from the corresponding $(i, r_i(c))$ pair, the interpolated polynomial $p'(x)$ will be different from the initial polynomial $p(x)$, leading to a different sk' and hence to invalid authentication. If this is the case, m out of n CRPs are alternatively substituted with the m additional points that, by definition, belong to $p(x)$. Each set of points constituted by m additional points and $n - m$ CRPs is consecutively interpolated to obtain again another polynomial $p''(x)$. If at least one set of points has all the faulty PUFs substituted with the m additional points, eventually $p''(x)$ will match $p(x)$, leading to a successful authentication.

To the best of our knowledge, we are the first proposing the use of a PUF combiner to guarantee correct authentication in the presence of PUF failure due to aging or wrong PUF behaviors due to malicious manufacturer.

A. Security discussion

Intuitively, the protocol is secure against PUF aging and dishonest PUF controlled by malicious manufacturers (we are currently deploying a formal demonstration). However, an attacker should also exploit the additional points to reconstruct the secret. This is possible whether it controls at least $n - m$ PUFs. In our model we assume that to guarantee the security of the IoT device, the producers are using no more than a single PUF from each manufacturer. Thus, it is sufficient that $n - m > 1$ to guarantee the security of our combiner.

B. Protocol implementation

We have already implemented the algorithm, tested it on a ESP32-C3-MINI device, and obtained some promising early results. The correctness holds true: as long as the number of faulty PUFs is less than or equal to the number of additional points m, the polynomial is always correctly interpolated. For what concerns the performances, the overhead introduced by the combiner (in addition to the evaluation of the PUFs and the usage of fuzzy extractors) goes from few milliseconds with 3 PUFs to at most ≈ 64 milliseconds with 7 PUFs.

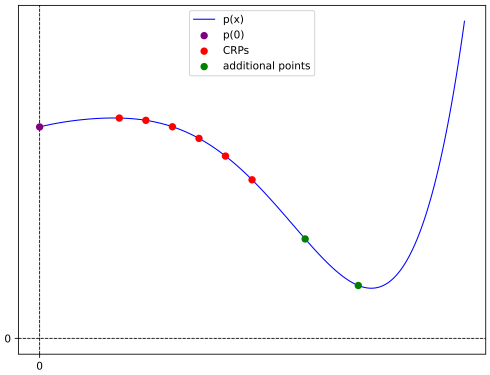


Fig. 1. Lagrangian modular interpolation of six challenge-response pairs corresponding to six PUFs to obtain a degree 5 polynomial $p(x)$. sk and two additional points are computed starting from $p(x)$.

We would like to further improve the algorithm. Instead of always substitute m additional points to m CRPs, the algorithm may start substituting one additional point, the two additional points, all the way up to m. This approach offers two advantages.

First, suppose at iteration j, exactly j points are substituted. The remaining $m - j$ additional points can be used as checksum by testing whether they belong to the polynomial. Given an additional point $(x, y = p(x))$, if $p'(x) \neq y$, then $p(x) \neq p'(x)$. The algorithm is, thus, relieved from computing hash values.

Second, by gradually increasing the number of substituting additional points, the algorithm is able to exactly identify the faulty PUFs. As soon as an interpolation turns out to be correct, the PUFs corresponding to the substituted CRPs are the faulty ones, allowing for either their replacing or the complete discard of the device itself.

REFERENCES

[1] Antonakakis, Manos, et al. "Understanding the mirai botnet." 26th USENIX security symposium (USENIX Security 17). 2017.

[2] Sheng, Chuan, et al. "Pager Explosion: Cybersecurity Insights and Afterthoughts." IEEE/CAA Journal of Automatica Sinica 11.12 (2024): 2359-2362.

[3] Pappu, Ravikanth, et al. "Physical one-way functions." Science 297.5589 (2002): 2026-2030.

[4] Dodis, Yevgeniy, Leonid Reyzin, and Adam Smith. "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data." Advances In Cryptology-EUROCRYPT 2004: International Conference On The Theory And Applications Of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004. Proceedings 23. Springer Berlin Heidelberg, 2004.

[5] Suh, G. Edward, and Srinivas Devadas. "Physical unclonable functions for device authentication and secret key generation." Proceedings of the 44th annual design automation conference. 2007.

[6] Shamir, Adi. "How to share a secret." Communications of the ACM 22.11 (1979): 612-613.

Poster: Can We Trust PUF Manufacturers? Robust PUF-Based Authentication via Combiners

Edoardo Liberati
Sapienza University of Rome
liberati@diag.uniroma1.it

Pierpaolo Della Monica
Sapienza University of Rome
dellamonica@diag.uniroma1.it

Riccardo Lazzeretti
Sapienza University of Rome
lazzeretti@diag.uniroma1.it

Ivan Visconti
Sapienza University of Rome
visconti@diag.uniroma1.it

Motivation

Physical Unclonable Functions (PUFs) provide uniqueness, ease of integration, lightweightness and not storage of secrets. These features allow for identification, authentication and key generation on Internet of Things (IoT) devices. In fact, standard cryptography is hardly achieved on most of them due to the lack of resources (energy, computational power, equipment).

BUT they may fail.
Both natural aging of the hardware and manufacturers' maliciousness can compromise PUFs reliability, invalidating the protocol they are involved in.

Background

PUF

Hardware keyed hash functions
1. Input: *challenge*
2. Output: *response*
3. Key: *PUF instance itself*

Supplying the same challenge to the same PUF will produce similar, but not identical, responses.

time $t1$: $r1 = PUF(c)$
time $t2$: $r2 = PUF(c)$
$r1 \approx r2$, i.e. $distance(r1,r2) < threshold$

This makes PUFs suitable for authentication [1].

Fuzzy extractors [2] can be employed in the protocol to generate and reproduce the same value out of noisy strings by means of additional *helper data*.

time $t1$: $w1 = PUF(c)$; $r1,P = Gen(w1)$
time $t2$: $w2 = PUF(c)$; $r2 = Rep(w2,P)$
$r1 = r2$ if $distance(w1,w2) < threshold$

This allows for both authentication and key generation.

Both aging and manufacturers' maliciousness eventually increase the distance of the two responses above the threshold, invalidating the protocol, either authentication or key generation.

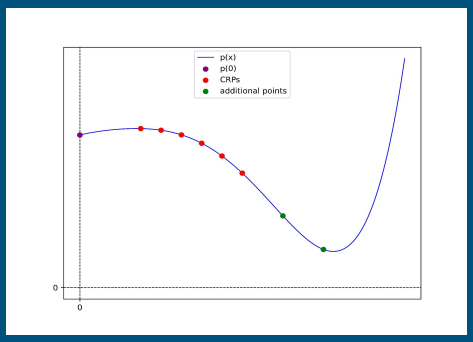
Acknowledgment:
Project "SERICS" (PE00000014) under the NRRP MUR program funded by the EU- NGEU. The views are of the authors only, not of the funding entities.

Proposal

PUF Combiner

Combine multiple PUFs such that, if a subset of them is nominally working, then also the combination is nominally working.

Idea description:
- n is the number of PUFs
- Fuzzy extractor follows each PUF.
- We can consider the pair $(i,r_i(c))$ as a point on the 2D plane (finite field), with i indexing the PUF.
- The n $(i,r_i(c))$ pairs uniquely identify a polynomial $p(x)$ of degree $n-1$.
- Given $p(x)$, we can compute $sk = p(0)$ and $m < n$ additional points belonging to $p(x)$.
- sk will constitute the secret key used by the protocol.
- The m additional points will be used as substitutes for the potentially faulty PUFs.



In case of faulty PUFs:
- We employ Shamir Secret Sharing modular Lagrangian interpolation, whose points and coefficients belong to a finite prime field.
- A random oracle, allows to check whether $p(x)$ was correctly reconstructed, i.e. the fresh sk matches the previous one.

On going works:
- formal security demonstration
- real-world implementation
- use of additional points to both validate the interpolation of $p(x)$ and identify the faulty PUFs

Bibliography

[1] Suh, G. Edward, and Srinivas Devadas. "Physical unclonable functions for device authentication and secret key generation." *Proceedings of the 44th annual design automation conference*. 2007.

[2] Dodis, Yevgeniy, Leonid Reyzin, and Adam Smith. "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data." Advances In Cryptology-EUROCRYPT 2004: International Conference On The Theory And Applications Of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004. Proceedings 23. Springer Berlin Heidelberg, 2004.