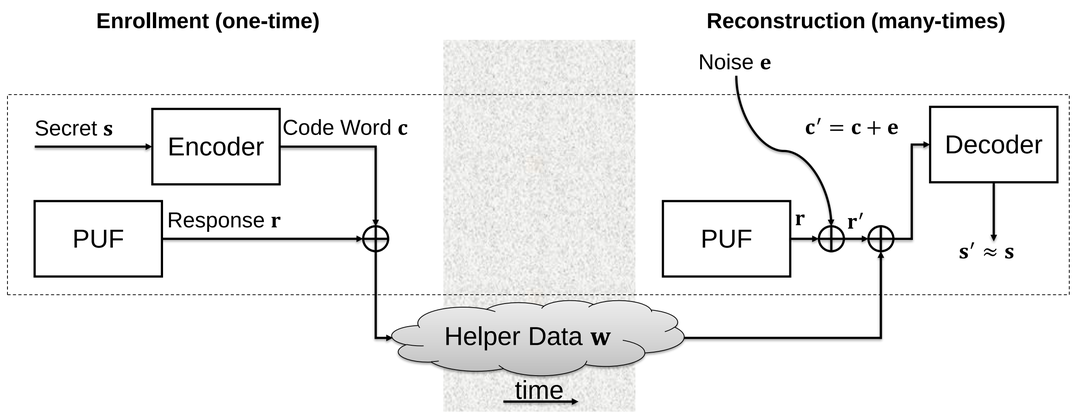
Machine Learning of Physical Unclonable Functions Using Helper Data

Revealing a Pitfall in the Fuzzy Commitment Scheme

Emanuele Strieder*, Christoph Frisch**, Michael Pehl**
*Fraunhofer Institute for Applied and Integrated Security, Munich; **Chair of Security in Information Technology, Technical University of Munich

Background and Contribution

Key Storage with PUFs and Fuzzy Commitment



- A secret is embedded in a device during *enrollment*. The actual secret $\mathbf{s}$ is encoded to a codeword and XORed (masked) with a PUF response resulting in helper data $\mathbf{w}$. $\mathbf{w}$ does not leak about $\mathbf{s}$ and is publicly stored.
- During *reconstruction*, $\mathbf{w}$ is XORed with the – expectedly same – PUF response. This response differs from the enrollment, e.g., due to measurement noise, environmental changes, and aging. Therefore the XOR results in a noisy codeword. However, the secret from enrollment can be reconstructed if the decoder was properly selected.

Machine Learning on PUFs

Machine Learning has been applied successfully to PUFs with challenge response behavior (Multi Challenge PUFs). For this purpose normally:

- Challenge response pairs of the PUF are collected.
- A model is trained based on the collected data
- A challenge-response protocol is attacked by using the mathematical clone, predicting responses for yet unseen challenges.

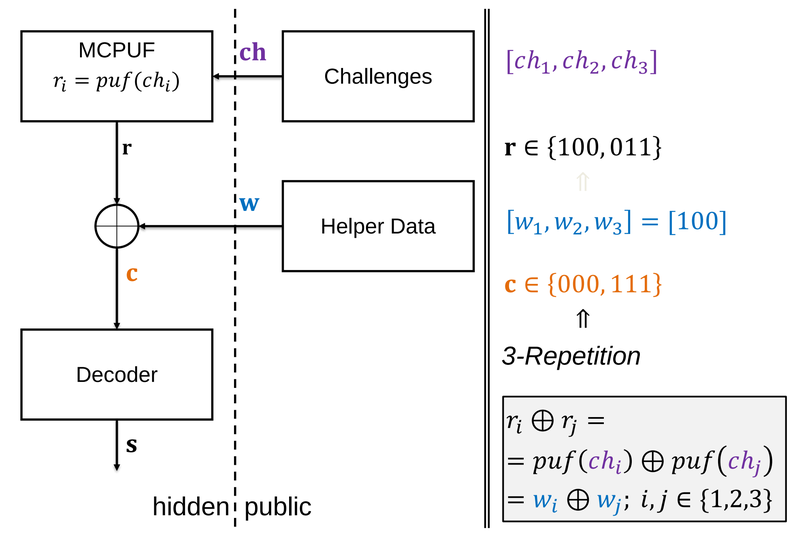
In the Fuzzy Commitment Scheme no challenge-response pairs are available. So how to exploit the potential vulnerability?

Our Contribution

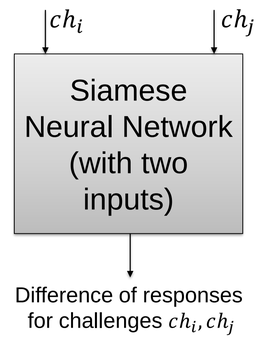
- Formalization of a new machine learning attack on multi challenge PUFs in a key storage scenario.
- Practical demonstration of the attack feasibility for different codes.
- Discussion of impact and potential countermeasures
- Introduction of Siamese Neural Networks to the PUF context

Approach

Attack Idea



- Every error correction needs redundancy. Although the codeword is masked with a PUF, and $\mathbf{w}$ does not leak about the secret, it leaks about the PUF through this redundancy.
- For binary linear block codes, normally used with PUFs, the codeword is a linear combination of information bits. The PUF response XORed to the codeword is hidden from an attacker through a unknown codeword with known structure.
- The attacker can apply transformations to eliminate unknown codeword bits. The remaining *XOR equations* depend only on XORs of PUF responses.
- Machine learning can model multi challenge PUFs given challenges and the XOR of responses. Since a difference of responses must be learned, Siamese Neural Networks are well suited for the task. Applying the model reveals further, yet unknown dependences and – together with the helper data – the key.

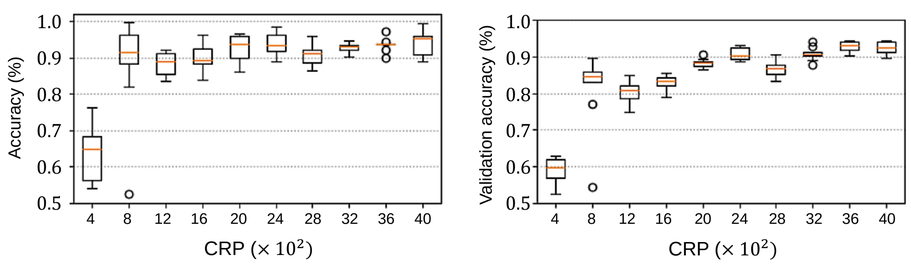


Example

Given a (3,1,1) repetition code and helper data $\mathbf{w} = [100]$, the PUF response must be $\mathbf{w} \oplus \mathbf{c} = \mathbf{r} = [r_1, r_2, r_3] \in \{[100], [011]\}$. With this knowledge, $r_i \oplus r_j = puf(ch_i) \oplus puf(ch_j) = w_i \oplus w_j$ with $i, j \in \{1,2,3\}$ is computed. A Simese Neural Network is trained with *features* ch_i, ch_j and *labels* $w_i \oplus w_j$. To exploit the model, challenges of different codewords are applied revealing the dependence between codewords and therefore the secret.

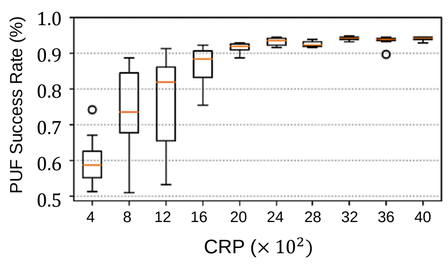
Selected Results

Attack on Arbiter PUF and (7, 1) Repetition Code

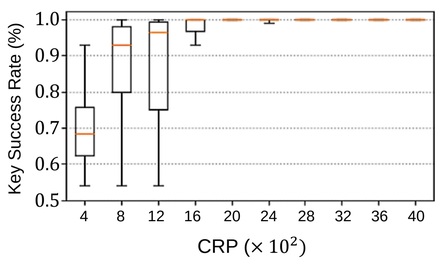


(a) Accuracy of ML model.

(b) Validation accuracy of ML model.



(c) Success rate of PUF prediction



(d) Success rate of key prediction.

Helper Data Based vs. Response Based Machine Learning

	Target	Labels	CRPs	Accuracy
Our Work	APUF	XOR of Helper Data	2k	97.51%
	APUF	PUF Response	800	99.96%
[SBC19]	APUF	PUF Response	$8k$	99.50%
	4 XOR APUF	PUF Response	$240k$	97.80%
	MPUF	PUF Response	$112k$	97.50%
	(4,4) iPUF	PUF Response	$647k$	97.68%

[SBC19] Santikellur et al. “Deep Learning based Model Building Attacks on Arbiter PUF Compositions”, IACR Cryptol. ePrint Arch., 2019:566, 2019.

Countermeasures

- Limiting number of challenge responses → fixes the problem
- Usage of complex PUFs → makes machine learning difficult
- Usage of complex codes → makes machine learning difficult

This work appeared in IACR Transactions on Cryptographic Hardware and Embedded Systems, ISSN 2569-2925, Vol. 2021, No. 2, pp. 1–36. DOI:10.46586/tches.v2021.i2.1-3. It was partly funded by the German Ministry of Education, Research and Technology through the project SecForCARs (grant number 16KIS0795).