A Plausibly Deniable Encryption Scheme Utilizing PUF's Thermo-Sensitivity

Changting Li[1,2,3], Zongbin Liu[2,3], Lingchen Zhang[2,3(✉)], Cunqing Ma[2,3], and Liang Zheng[1,2,3]

[1] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[2] Data Assurance and Communication Security Research Center, Beijing, China
[3] State Key Laboratory of Information Security, Institute of Information Engineering, CAS, Beijing, China
{lichangting, liuzongbin, zhanglingchen, macunqing, zhengliang}@iie.ac.cn

Abstract. Deniable encryption is proposed to protect sensitive data against adversaries, even when the user has been coerced to reveal his private keys and other random parameters. However, current deniable encryption schemes or techniques either require the user to remember some tedious random parameters used in encryption or demand special designs in the file system. Any abnormality in the user's behavior or in the file system tend to arouse suspicion, thus reduce the persuasion of the decrypted data. To cheat the adversary convincingly, we innovatively utilize the thermos-sensitivity of Physically Unclonable Functions (PUFs), to propose a novel and practical deniable encryption scheme, which enables the encryption system achieve deniability in a very covert way. The proposed scheme will automatically interpret the deniable ciphertext into different plaintexts at different temperatures and does not require any special designs in the file system. Furthermore, we successfully implement our scheme on Xilinx KC705 evaluation boards to prove its feasibility.

Keywords: Deniable encryption · Bistable Ring PUF
SRAM PUF · FPGA

1 Introduction

Conventional encryption schemes seldom think about situations when one or both two sides of communication are coerced to reveal their private information, e.g. private keys, nonce and other random parameters used in encryption. However, such situations can always be found in real world scenarios. For example, a man is taking a disk with encrypted sensitive documents through the Customs, but unfortunately the customs

The work is supported by a grant from the National Key Research and Development Program of China (Grant No. Y16A01602).

S. Qing et al. (Eds.): ICICS 2017, LNCS 10631, pp. 105–117, 2018.
https://doi.org/10.1007/978-3-319-89500-0_9

officer requires checking the content of his disk. In order to cheat the officer, the man would hope to convincingly deny the existence of the genuine plaintext.

One way to achieve this goal, is to explain the encrypted document into a fake innocuous one. Given this, Canetti, Dwork, Naor, and Ostrovsky firstly proposed intriguing Deniable Encryption in 1997 [1]. The main idea is to construct a fake randomness, maybe the key or some additional parameters required in the encryption, to reinterpret the ciphertext into a plausible fake plaintext. Though varieties of schemes have been proposed since then [14, 15, 18, 19], these schemes are limited in theoretical discussion. In order to satisfy information security requirements, all the theoretical schemes are suffering from extremely long length of ciphertext or key [1, 2].

In engineering practice, engineers seek another way to obtain deniability which is so called Plausibly Deniable Encryption. Plausibly Deniable Encryption aims to deny the existence of encrypted data with the help of engineering methods, e.g. TrueCrypt [3], Rubberhose filesystem [4], Steganographic File Systems [5–7, 17]. These schemes usually hide sensitive data in a hidden volume or a random-looking free space, but such schemes require some special designs in the filesystem and are under threat of flaws in the implementation [8] and forensic tools [9]. Moreover, the existence of such special designs in the file system is detectable.

No matter in theoretical discussion or in engineering practice, the basic idea to achieve deniability is the same: Though having been forced to hand in all the parameters used in encryption, the user is still able to retain a trapdoor information which is the radical difference between him and the adversary. The adversary without this trapdoor information, in spite of all the other parameters used in encryption he has had, he cannot tell whether the decrypted plaintext is a fake or a genuine one (in theoretical deniable encryption) or distinguish between a truly random sequence and a ciphertext (in plausibly deniable encryption). Therefore, the secrecy of the trapdoor information is even more important than the encrypt key in deniable encryption scenarios. This trapdoor information should be stored as covert as possible and to convincingly cheat the adversary, both the user's behavior and the encryption system should look normal enough not to arouse the adversary's suspicion.

On account of these, we propose a practical deniable encryption scheme which takes advantage of PUFs' thermo-sensitivity to implement deniable encryption in quite a covert way. Our scheme neither requires the user to remember or store any tedious trapdoor information, nor requires any special designs in the file system or extra inputs during decryption. Generally, PUF's sensitivity to temperature is regarded as an undesirable nature that undermines PUF's stability. However, we aware that if PUF's behavior varies with temperature, it may serve as a thermosensitive "hidden trigger" which can only be triggered in specific temperature range. In the proposed scheme, the PUF-based "hidden trigger" is able to perceive temperature variation, which makes the temperature become a vital and covert trapdoor information to determine whether to decrypt faithfully or not.

Details of the scheme will be described in Sect. 3 and we successfully implemented it on Xilinx KC705 evaluation boards to examine its feasibility. According to the experiment results, ciphertexts generated at extreme temperature (e.g. −40 °C or 60 °C) will be decrypted as the prepared fake plaintext at room temperature (20 °C–30 °C).

In conclusion, our contributions are summarized as follows:

1. We take advantage of PUF's thermo-sensitivity, which is always thought to be an undesirable nature of PUF, to design a novel and practical deniable encryption scheme.
2. Our scheme enables the user to achieve deniability in a very covert way. The coerced user just needs to make sure the temperature of decryption environment is out of the "trigger range" in which the deniable ciphertext will be decrypted loyally. In addition, except one encryption key and ciphertexts, no extra input or extra operation is needed.
3. From the adversary's view, our encryption system works normally. The adversary is free to choose arbitrary text to invoke the encrypt and decrypt programs to examine our system and he will be convinced that the generated ciphertext is always decrypted loyally.

The rest of this paper is organized as follows. In Sect. 2 we describe working mechanisms and evaluations of PUFs and introduce our basic idea. Then we illustrate our scheme in Sect. 3 with performance analysis and present details of experiments on Xilinx KC705 evaluation boards in Sect. 4. Finally, we conclude in Sect. 5.

2 Preliminary

Before describing our design of PUF-based Deniable Encryption, we firstly introduce some backgrounds on PUFs and then elaborate where our inspiration comes from.

2.1 Physically Unclonable Functions

PUF as an emerging technique of physical roots of trust provides new solutions for authentication, tamper resistance, anti-counterfeiting, key generation and protection etc. [16]. Because of uncontrollable and inevitable influences of random variations during manufacturing process, no perfectly identical chips can be produced. Such subtle variations on products can be regarded as chips' physical "fingerprints" and PUFs aim to extract these "fingerprints" and translate them into unique secret sequences, the response, which can be utilized to serve cryptographic primitives.

Generally, each PUF entity can be described as a one-way function PUF: C $\rightarrow$ R, where C is an input challenge set and R is the corresponding response set. For an PUF entity puf_i, its Challenge Response Pair (CRP) c_k and $r_i(c_k)$ should be unique and unpredictable, i.e. for different entities puf_i and $puf_j (i \neq j)$ with the same input c_k, their responses are different: $r_i(c_k) \neq r_j(c_k)$; and for the same PUF entity puf_i with different inputs c_{k1} and c_{k2}, its corresponding responses are different: $r_i(c_{k1}) \neq r_j(c_{k2})$. Besides, any adversary can neither predict a response before observing it, nor reversely derive its corresponding challenge.

To evaluate a PUF's performance, unpredictability, uniqueness and reliability are commonly investigated in the literature.

Unpredictability: An ideal PUF's unobserved response should be unpredictable, even if the adversary has observed enough CRPs of it. Providing every bit in a binary response sequence r $\in \{0,1\}^n$ is independent, min-entropy calculated as formula (1) offers a lower bound of responds' randomness in the worst case.

$$H_\infty(r) = \sum_{i=1}^{n} -\log_2(\max\{P(r_i = 1), P(r_i = 0)\}). \tag{1}$$

$P(r_i = 1)$ and $P(r_i = 0)$ are probabilities for the i_{th} bit of response to equal 1 and 0.

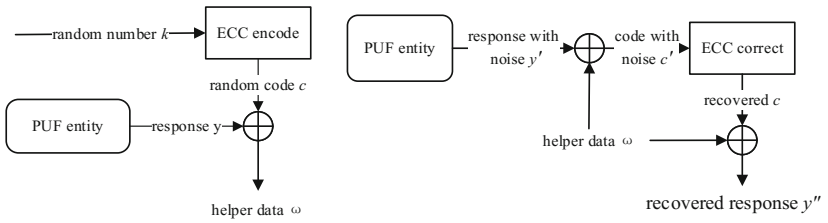
Reliability and Uniqueness: Assume we instantiate N_{puf} PUF entities, and invoke each of them with N_{chal} challenges, for each challenge we measure N_{meas} times. Thus, we obtain $N_{puf} \times N_{chal} \times N_{meas}$ response sequences. Equations (2) and (3) calculate the average intra-distance and average inter-distance respectively [13].

$$\mu_{intra} = \frac{2}{N_{puf} \cdot N_{chal} \cdot N_{meas} \cdot (N_{meas} - 1)} \sum_{\substack{j_1, j_2 = 1 \\ j_1 \neq j_2}}^{N_{meas}} \sum_{i=1}^{N_{puf}} \sum_{k=1}^{N_{chal}} HD(r_i^{j_1}(c_k), r_i^{j_2}(c_k)). \tag{2}$$

$$\mu_{inter} = \frac{2}{N_{puf} \cdot (N_{puf} - 1) \cdot N_{chal} \cdot N_{meas}} \sum_{\substack{i_1, i_2 = 1 \\ i_1 \neq i_2}}^{N_{puf}} \sum_{k=1}^{N_{chal}} \sum_{j=1}^{N_{meas}} HD(r_{i_1}^{j}(c_k), r_{i_2}^{j}(c_k)). \tag{3}$$

HD (·) is a function counting the Hamming Distance (HD) between two PUF responses. Apparently, average intra-distance reflects the difference between each measurement (reliability) and average inter-distance demonstrates to what extent entities of the same PUF are different from each other (uniqueness). For a PUF design, its ideal inter-distance is 50%, while its intra-distance should be as low as possible.

Error Correcting Code (ECC): Because PUF's response is not perfectly reproductive, ECCs like Hamming code, Reed-Muller code, BCH code, repeating code etc., are widely adopted in PUF's application to guarantee that the same response is generated in every invoking. The enrollment and recovery process are shown in Fig. 1, generally the helper data can save in an unprotected NVM and the response security is guaranteed by the random number k in the enrollment process.



(a) Enrollment Process (b)Recovery Process

Fig. 1. Enroll and recover PUF response with ECC

2.2 Inspiration and Basic Idea

For almost all the electric PUFs, temperature variation is one of the principal factors that undermine PUF's reliability. However, according to our survey, we notice that PUFs' behavior does not vary with temperature irregularly.

Daniel et al. in paper [10] investigated SRAM cells which showed no obvious tendency at 293 K and found that if a neutral-skewed SRAM cell at 293 K, whose power-up tendency is '0' at 273 K, is inclined to turn into '1' at 323 K and vice versa. They also noticed that this skew shift is monotonic with temperature. Resembling phenomena are also observed by Chen et al. who firstly proposed BR PUF [11]. According to their research, an intra-distance up to 5.81% was caused while temperature changed from room temperature to 85 °C, however, at each specific temperature the BR PUF showed high stability with a maximum distance of 0.76%. Figure 2 shows temperature variation on intra-distances and inter-distances of DAC PUF [20], which we think to be a good representation of PUFs' thermo-sensitivity. According to the blue curve, if we enroll a response at 25 °C and recover it at other temperature, as the temperature difference is enlarged gradually, the Hamming distance between the enrolled and recovered response sequence increases notably. However, from the green curve we can see that if we enroll and recover a response at every temperature respectively, the intra-distance will stable at a very low level.

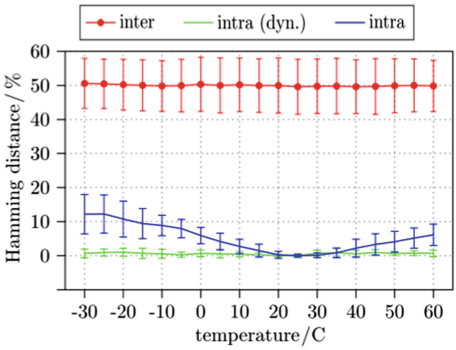


Fig. 2. Temperature variation on intra- and inter-distances of DAC PUF [20]

PUFs' such property suggests that if we choose an ECC algorithm with appropriate error correcting capability, we can control a PUF's responses only to be recovered within a temperature range, thereby utilize PUF to perceive temperature variations. If the enrolled response is recovered successfully, the ciphertext will be decrypted loyally, otherwise a prepared fake text will be output as the decrypted plaintext to cheat the adversary. This is the basic idea of the proposed deniable encryption scheme.

3 PUF-Based Deniable Encryption

The proposed scheme is a plan-ahead deniable encryption scheme, i.e. fake text is prepared before decryption. The basic idea is to let the cryptographic system vary its decryption result automatically under different temperature conditions. The scheme contains four programs.

- The **Enroll** program is responsible for recording environmental temperature in PUF's response sequence. As mentioned in Sect. 2.2, some PUFs' behavior stably varies with temperature, therefore, the enrolled response sequence can be regarded as a reflection of temperature and will serve as a "hidden trigger" which can only be successfully recovered in neighboring temperature range.
- The **Explain** program prepares alternative texts beforehand to generate deniable ciphertexts. The input of the **Explain** program are two texts m and m', where m is the genuine text and m' is the fake one which will take place of m as the decryption result to cheat the adversary.
- The **Encrypt** program also generates ciphertexts, but its ciphertexts can only be decrypted faithfully. Therefore, this program has only one input text m, the format of its ciphertext is analogous to that of the **Explain** program.
- The **Decrypt** program will selectively output genuine or fake plaintexts according to the temperature. While doing decryption, the **Decrypt** program checks the temperature condition by comparing the recovered trigger with the enrolled one. We call the recovered trigger equals the enrolled one as "the trigger is triggered". In this case, the program recovers the genuine text and output it as the final decryption result, otherwise, the program just outputs the decrypted fake text.

3.1 Overview of PUF-Based Deniable Encryption System

The hardware architecture of our deniable encryption module is shown in Fig. 3. It mainly contains two systems: The Cryptographic system and the PUF system.

The Cryptographic system is a module that achieves both encrypt function EN $(\cdot)$ and decrypt function DE $(\cdot)$ of a secure symmetric key algorithm, such as AES.

The PUF system is consisted of a PUF instances module, the ECC module and a nonvolatile memory. In the ECC module, there are two ECC algorithms with different error correcting capabilities. The weaker one ECC_{wk} only guarantees recovery of the "hidden trigger" in a narrow temperature range; while the stronger one ECC_{st} should make sure the random mask in the ciphertext can always be recovered under any condition.

3.2 Workflow

Enroll Program $Enrl: k \rightarrow (rsp_1, w_1)$. The **Enroll** program records current temperature in PUF's response sequence. It first uses the encryption key k as PUF's challenge and obtains a response sequence $rsp_1 = puf(k)$. rsp_1 will serve as the "hidden trigger" and be saved in the nonvolatile memory of the PUF module. Then the program calculates the helper data $w_1 = ECC_{wk}^{enrol}(rsp_1)$ and saves it as well.

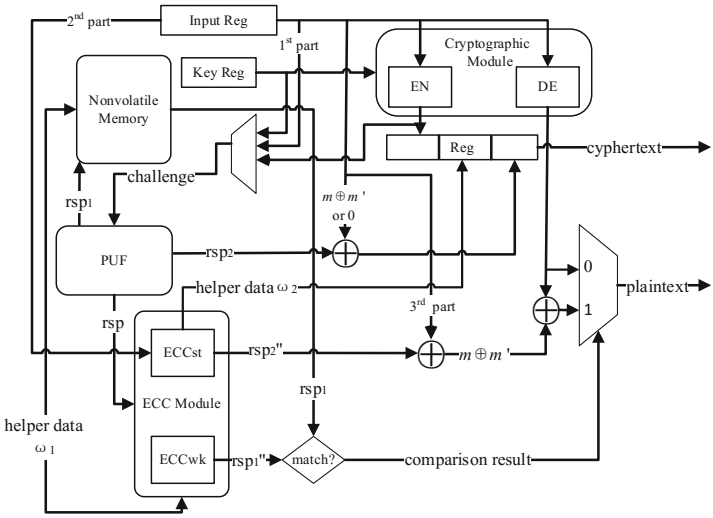


Fig. 3. The hardware architecture of the proposed deniable encryption module

Explain Program $Exp : (m, m') \rightarrow dc$. The **Explain** program generates deniable ciphertext dc with input (m, m'), m is the genuine text and m' is the fake one. First, the program encrypts the fake text normally with the symmetric key algorithm and acquires ciphertext $c' = EN\ (k, m')$. Then uses c' as PUF's challenge to get corresponding response sequence $rsp'_2 = puf\ (c')$. rsp'_2 serves as a random mask to hide the two texts' difference $m \oplus m'$. "$\oplus$" is the bit XOR operator. Finally, the helper data $w'_2 = ECC_{st}^{enrol}(rsp'_2)$ is calculated and forms the output deniable ciphertext: $dc = c' \| w'_2 \| (rsp'_2 \oplus m \oplus m')$.

Encrypt Program $Enc : m \rightarrow ec$. The **Encrypt** program generates ciphertext ec with one input text m. First, the program encrypts m with the encryption key k by the symmetric key algorithm, i.e. $c = EN\ (k, m)$, and uses this ciphertext as challenge to invoke the PUF and get corresponding response $rsp_2 = puf(c)$. Also, the helper data w_2 is calculated by ECC_{st} and the ciphertext $ec = c \| w_2 \| rsp_2$.

Decrypt Program $Dec : c_{in} \rightarrow m_{out}$. The **Decrypt** Program explains the input ciphertext c_{in} into certain plaintext m_{out}. First, the program divides c_{in} into three equilong parts $c_{in} = c'' \| w''_2 \| mk$ and decrypts c'' with the symmetric key algorithm to get $m_{temp} = DE(k, c'')$. Then the program invokes PUF with the encryption key k and recovers the acquired response with the saved helper data w_1 by the weaker ECC algorithm, i.e. $rsp''_1 = ECC_{wk}^{recov}(puf(k), w_1)$. If rsp''_1 dose not equal the saved trigger rsp_1, i.e. $rsp''_1 \neq rsp_1$, the program outputs m_{temp} directly; otherwise, it uses c'' to invoke the PUF and recover the obtained response with w''_2 by the stronger ECC algorithm, i.e. $rsp''_2 = ECC_{st}^{recov}(puf(c''), w''_2)$, finally outputs $m_{out} = m_{temp} \oplus rsp''_2 \oplus mk$.

3.3 Performance Analyses

Correctness: The deniable ciphertext dc can be correctly decrypted into the genuine text m by the **Decrypt** program under the enrolled temperature region, because the

change of the PUF response rsp_1 that severs as the "hidden trigger" will be within the correction capability of ECC_{wk}. As long as the recovered response equals the enrolled one, the **Decrypt** program will extract the hidden information $m \oplus m'$ masked by rsp_2 (because there is: $rsp_2'' \oplus mk = rsp_2'' \oplus (rsp_2' \oplus m \oplus m') = m \oplus m'$) to reconstruct the genuine text. If the input ciphertext is generated by the **Encrypt** program, whether the trigger is "triggered" or not, the **Decrypt** program will always decrypt faithfully. Because $ec = c \| w_2 \| rsp_2$ and rsp_2 can be regarded as a masked all-zero sequence. Any sequence doing bit XOR operation with the all-zero sequence equals itself, so the output will always be $DE(k, c)$.

Deniability: While operating under certain temperature which is out of the "trigger range", the deniable ciphertext dc, which is originally generated by the **Explain** program, will be decrypted into the prepared fake text m'. Because the change of response sequence is already out of the correction ability of ECC_{wk}, thus rsp_1 cannot be successfully recovered, i.e. the "hidden trigger" will not be "triggered", the **Decrypt** program just outputs $DE(k, c'')$ directly.

Security: As with respect to the first part of the ciphertext (in the **Explain** program is the fake text m', in the **Encrypt** program is the sole input m), the adversary has no way to derive the text protected by cryptographic algorithm. Owing to PUF's unpredictability and randomness, the random mask used in the third part makes the adversary unable to figure out the hidden difference $m \oplus m'$. As the second part, the helper data of the random mask, has nothing related to either text m or m', the security of the whole ciphertext in our scheme is guaranteed.

Practicability: The prime advantage of our scheme is that the user does not need any special manipulation to cheat the adversary. In our scheme, we hide the information $m \oplus m'$ that helps us to recover the genuine text in the ciphertext itself and utilize the temperature as the covert trapdoor information to achieve deniability. Therefore, no extra input is required during decryption and the enrolled temperature, under which the deniable ciphertexts are generated, is kept in the user's mind without a trace. The user just needs to make sure that the temperature of the environment, in which he may be compelled, is most likely to be out of the "trigger range". Furthermore, in our scheme, the **Encrypt** program and the **Decrypt** program are accessible to the adversary. The adversary can choose arbitrary plaintexts or ciphertexts to examine the loyalty of the encryption system, but as the ciphertext generated by the **Encrypt** program can only be decrypted loyally, from the view of the adversary, our deniable encryption system will always perform in a normal way.

4 Experiment and Result

4.1 Parameter Determination

The PUF in our proposed scheme is used for two main purposes: the "hidden trigger" and random mask generator. The "hidden trigger" is supposed to possess sufficient thermos-sensitivity, as well as relatively high reliability, while the random mask should possess adequate randomness to guarantee the security of the ciphertext.

The most important thing in the real design is to determine the weaker and the stronger ECC algorithms and their correction capabilities according to PUF's actual properties. We must investigate how much influence do temperature variations pose on the PUF's reliability, because if the ECC algorithm is too strong, the trigger would be unresponsive to temperature variation, then the deniable ciphertext will be decrypted faithfully in a large temperature range; if the ECC algorithm is too weak, the correctness of our scheme cannot be ensured.

We deploy1024 Bistable Ring PUFs (BR PUF) [11] on two KC705 boards respectively. To investigate the properties of this BR PUF, we exhaust all the challenges and measure every challenge for 32 times under 5 different temperature conditions (−40 °C, −20 °C, 25 °C, 40 °C and 60 °C). For each measurement, we can obtain 1024 response bits from each board, thus we totally acquire about 1 giga-bit data. According to formulas (2) and (3), we yield the PUF's average intra-distance and inter-distance are 5.00% and 44.34% respectively. This result suggests that this kind of BR PUF is able to generate a relatively stable trigger sequence and sufficiently different random masks with different configurations.

We further calculate the average intra-distances of responses generated under different temperatures and compare this temperature-influenced distribution with the original intra-distance distribution in Fig. 4. From the figure we can see, the whole distribution shifts rightwards, and the average intra-distance increases to 8.89%. As the weaker ECC algorithm must make sure the trigger sequence to be successfully recovered in a temperature range as narrow as possible, according to the original distribution, ECC that corrects sequences with 10% error bits is desirable. While the stronger one should be able to handle at least 18% error bit rate to recover the random mask at any temperature. Therefore, we chose (15, 11) Hamming Code (can correct 1-bit error in every 11 bits) as the weaker ECC and (1, 5) Reed-Muller Code [12] (can correct 7-bit error in every 32 bits) as the stronger one.

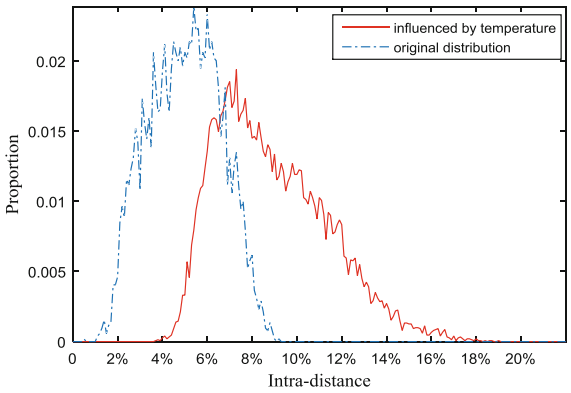


Fig. 4. The influence of temperature variation on intra-distance distribution

4.2 Implementation Details

The architecture of our evaluation system is shown in Fig. 5. We choose 128-bit AES as the symmetric key algorithm. To make the experiment more efficient, we output the generated "hidden trigger" and its corresponding helper data to the upper computer, rather than save them in a nonvolatile memory. Thus, at any specific temperature we can do enrollment and decrypt ciphertexts generated at other temperatures at the same time. The Microblaze, a soft microprocessor core designed for Xilinx FPGAs, is responsible for delivering commands and data between the upper computer and the hardware modules.

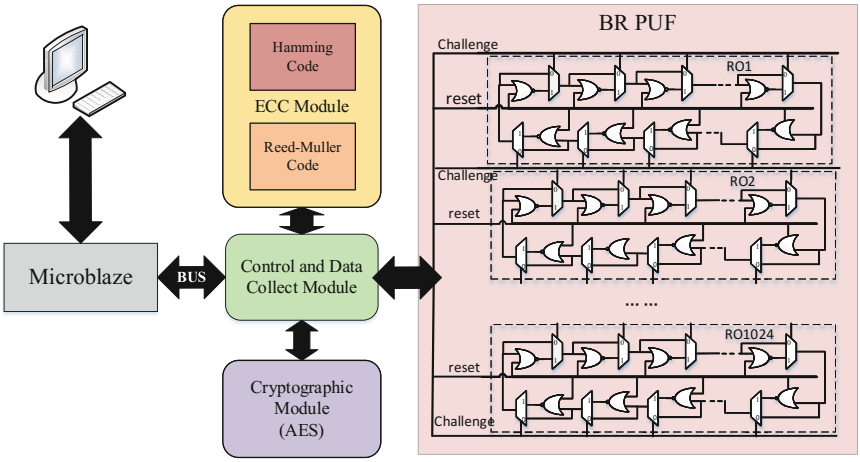


Fig. 5. The experimental evaluation system

4.3 Experiment Result

We generated 5000 128-bit random masks, substitute the result into formula (1) and acquire the random mask's min-entropy is 123.32 bits, i.e. averagely 0.96-bit entropy for each bit in the mask sequence, which demonstrates that the generated masks possess adequate randomness. Also, we hope the random masks are sufficiently different from each other. Therefore, we investigate the Hamming distance between every two masks and draw the distribution in Fig. 6. The average distance of the mask is 50.01%, which is quite desirable.

Whether the ciphertext will be decrypted loyally or not is decided by the recovery result of the trigger. Therefore, we randomly generate 100 encryption keys. Enroll them at −40 °C, 10 °C and 60 °C respectively, and then try to recover them with the enrolled helper data at every 10 °C from −40 °C to 60 °C. The changing patterns of recovery probability are drawn in Fig. 7, the x-coordinate is temperature and the y-coordinate represents recovery probability. For comparison, results on different boards are displayed separately.

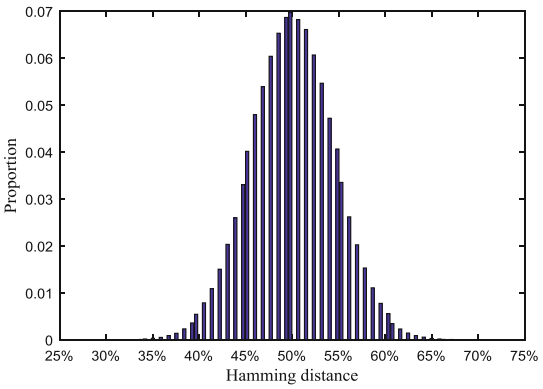


Fig. 6. The distributions of random masks' Hamming distances

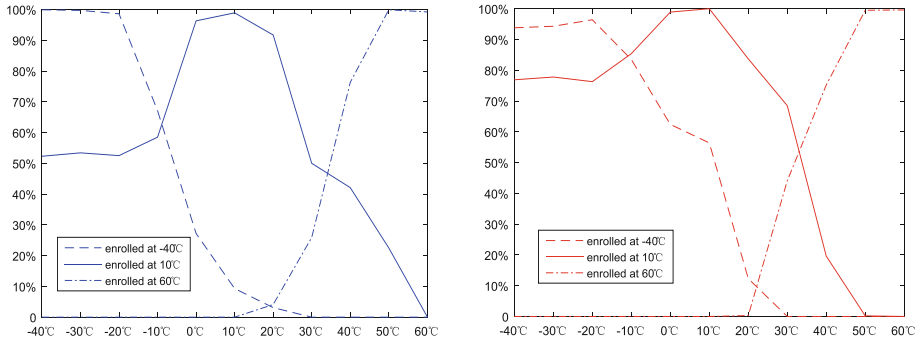


Fig. 7. Recovery probability under different circumstances

Comparing these two graphs, the changing trends of triggers' recovery probability on these two boards are the same on the whole. As the temperature difference is enlarging, the recovery probability declines obviously. With respect to changing patterns in the higher temperature region (10 °C–60 °C), we can see that triggers enrolled at 10 °C and 60 °C cannot be recovered when temperature difference reaches 50 °C. However, as lines tend to stay stable when temperature falls below −20 °C, triggers enrolled at 10 °C can still be recovered with a relatively high probability at −40 °C. Though the changing patterns at low temperatures are gentler, triggers enrolled at −40 °C are not likely to be recovered at room temperature (20 °C–30 °C). Considering heats emitted by electronic devices during working process, generating deniable ciphertexts at extreme low temperature could be a better choice.

5 Conclusion

In this paper, we present a novel and practical PUF-based deniable encryption scheme. Our key thought is to convert temperature into a covert trapdoor information, i.e. by utilizing PUF's thermo-sensitivity, we enable the decrypt program to perceive temperature variations thereby changes its output under different temperatures. In our scheme, because the trapdoor information is hidden in user's mind and as a physical factor it does not need to be invoked deliberately, the user is able to decrypt the ciphertext deniably without any abnormal manipulation, which makes the output plaintext more convincing. Based on this, we presented our architectural design and analysis its performances. In addition, we implement this scheme with BR PUFs on two Xilinx KC795 evaluation boards to prove its feasibility.

References

1. Canetti, R., Dwork, C., Naor, M., Ostrovsky, R.: Deniable encryption. In: Kaliski, B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 90–104. Springer, Heidelberg (1997). https://doi.org/10.1007/BFb0052229
2. Amit, S., Brent, W.: how to use indistinguishability obfuscation: deniable encryption, and more. In: STOC, pp. 475–484 (2014)
3. TrueCrypt.org. Free open source on-the-fly disk encryption software. Version 7.1a, July 2012. http://www.truecrypt.org/
4. Julian, A., Suelette, D., Ralf, W.: Rubberhose Cryptographically Deniable Transparent Disk Encryption System, 15 September 2010. Accessed 21 Oct. 2010
5. Anderson, R., Needham, R., Shamir, A.: The steganographic file system. In: Aucsmith, D. (ed.) IH 1998. LNCS, vol. 1525, pp. 73–82. Springer, Heidelberg (1998). https://doi.org/10.1007/3-540-49380-8_6
6. McDonald, A.D., Kuhn, M.G.: StegFS: a steganographic file system for linux. In: Pfitzmann, A. (ed.) IH 1999. LNCS, vol. 1768, pp. 463–477. Springer, Heidelberg (2000). https://doi.org/10.1007/10719724_32
7. HweeHwa, P., Kian-Lee, T., Xuan, Z.: Stegfs: a steganographic file system. In: 19th International Conference on Data Engineering, Proceedings, pp. 657–667. IEEE (2003)
8. Adal, C.: BestCrypt IV generation flaw. http://adal.chiriliuc.com/bc_iv_flaw.html
9. Robert, M.: Encrypted hard drives may not be safe. In: IDG News Service, 17 July 2010
10. Daniel, E.H., Wayne, P.B., Kevin, F.: Power-Up SRAM state as an identifying fingerprint and source of true random numbers. IEEE Trans. Comput. **58**(9), 1198–1210 (2009)
11. Chen, Q., Csaba, G., Lugli, P., Schlichtmann, U., Ruhrmair, U.: The Bistable Ring PUF: a new architecture for strong Physical Unclonable functions. In: IEEE International Symposium on Hardware Oriented Security and Trust–HOST, pp. 134–141. IEEE (2011)
12. Sebastian, R.: Reed-Muller Codes, Carleton University (2003)
13. Roel, M.: Physically Unclonable Functions: Constructions, Properties and Applications. Katholieke Universiteit Leuven, Belgium (2012)
14. Klonowski, M., Kubiak, P., Kutyłowski, M.: Practical deniable encryption. In: Geffert, V., Karhumäki, J., Bertoni, A., Preneel, B., Návrat, P., Bieliková, M. (eds.) SOFSEM 2008. LNCS, vol. 4910, pp. 599–609. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-77566-9_52

15. Dürmuth, M., Freeman, D.M.: Deniable encryption with negligible detection probability: an interactive construction. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 610–626. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20465-4_33
16. Tuyls, P., Škorić, B.: Strong authentication with physical unclonable functions. In: Petković, M., Jonker, W. (eds.) Security, Privacy, and Trust in Modern Data Management, pp. 133–148. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-69861-6_10
17. Katzenbeisser, S., Petitcolas, F.A.: Information Hiding Techniques for Steganography and Digital Watermarking. Artech House, Inc. (2000)
18. Howlader, J., Basu, S.: Sender-side public key deniable encryption scheme. In: International Conference on Advances in Recent Technologies in Communication and Computing, pp. 9–13. IEEE (2009)
19. Meng, B., Wang, J.Q.: An efficient receiver deniable encryption scheme and its applications. J. Netw. **5**(6), 683–690 (2010)
20. Herkle, A., Becker, J., Ortmanns, M.: Exploiting weak PUFs from data converter nonlinearity—E.g., a multibit CT ΔΣ modulator. IEEE Trans. Circ. Syst. I Regul. Pap. **63** (7), 994–1004 (2016)