Protecting devices operating in jammed and noisy networks*

Bertrand F Cambou, Michael L Garrett; Dina Ghanai Miandoab, Julie Heynssens

Abstract— **Keeping secure communications in noisy and congested environments is a difficult endeavor for many commercial and defense applications. A commercial application may include autonomous package delivery where there is a poor signal, and a defense application may include an electromagnetic environment where jammers are operating to reduce the communication capabilities of an opponent. We propose Challenge-Response-Pair (CRP) mechanisms to drive terminal devices operating in contested zero-trust environments comprised of weak signals in the presence of obfuscating electromagnetic noise. A set of encrypted files containing the instructions needed to operate the overall system is stored in the memory unit of the device. The set of ephemeral keys needed to decrypt these files is not stored in a vehicle for security reasons; a server transmits through the open network the information enabling the recovery of the keys, and the decryption of the files. The latencies of the suggested protocols must be small for near real-time operations. The proposed mechanisms accommodate the injection of obfuscating noises to mitigate several vectors of attacks and to disturb opponents trying to perform side channel analysis of the terminal devices. The methods suggested to recover error-free ephemeral cryptographic keys in noisy networks are implemented with light computing elements and do not rely on Error Correcting Codes (ECC), fuzzy extractors, or data helpers.**

I. Background Information

The objective of this work is to protect terminal devices such as unmanned aerial vehicles (UAVs), self-driving cars and other autonomous vehicles operating in zero-trust areas subjected to poor network coverage and jamming conditions. It is an active area of research. It is risky to store and distribute the cryptographic keys needed to protect sensitive information in terminal devices. Vectors of attacks are replays, man in the middle attacks, loss of information in the network, side channel analysis and physical loss to the opponent of a terminal device [1–3]. Storing the secret keys that decrypt the cipher texts of digital files in the terminal introduces the same risk level as storing non-encrypted files. In distributed networks, the clients usually store the public–private key pairs in their terminal devices, which presents an element of risk. The obfuscation of the key with a PUF enhances security, however the management of the challenges could be complex in a noisy network. The opponents can also inject noise to disturb the wireless communication between the ground operation and terminal device, introducing difficulty in the distribution of cryptographic keys without heavy ECC, fuzzy extractors, or data helpers, all of which are leaking information [4–6]. Blockchain technology with digital signatures offers protection in exposed networks [7-10]. In [11], a Challenge Response Authentication (CRA) of physical layers is described to avoid exposing passwords. [12] shows the methods to provide verification of information in a ledger. [12] define a "cryptographic challenge nonce". In [13], document tracking schemes on a distributed ledger are presented. [14,15] present a secure exchange of signed records. Methods to authenticate data based on proof verification are shown in [16–18]. In [19] a method to validate documents with blockchains is presented.[20-21] describe contract agreement verifications. The CRPs developed for PUFs provide relevant background information for our work [22-24]. Cryptographic protocols with PUF-based CRPs are shown in [25,26]. Biometry with a CRP mechanism is shown in [27,28]. CRP mechanisms and CRA schemes are also applied to protect centralized or distributed networks [29–31].

II. Description of the protocols

The devices contain sets of encrypted instructions needed to guide them. The set of ephemeral keys required to decrypt these files are not stored in the vehicle for security reasons; the information needed to retrieve these keys is transmitted through the noisy network by the server. The erratic bits injected into the transmitted information do not need to be corrected with ECC. This work includes a novel approach based on CRP schemes to protect and authenticate each digital file individually, with the objective of mitigating several vectors of attack. The idea to inject obfuscating noise during ground-to-terminal communication is developed to limit the ability of opponents to share the same wireless network for side channel analysis.

A. CRP mechanism based on digital files

The input data of the CRP mechanism is derived from file F. Its ciphertext C is concatenated with nonce ω to generate a file C* of constant length $d=2^D$, where D is the number of digits (for example, if a desired d=1 million, then D=20). The resulting d bits are located at addresses varying from 1 to d. Changing a single bit in file F results in a totally different stream C*.

- A digital file F is encrypted with an ephemeral key Sk to generate a cipher text C. A variety of algorithms under standardization by NIST [32] can be used such as CRYSTALS-Kyber [33,34], CRYSTALS-Dilithium [35,36], NTRU [37–39], Classic McElice [40], SPHINCS [41], and Falcon [42]. In our implementation we used symmetrical encryption (AES), Sk=Pk.
- The ciphertext is hashed with Standard Hash Algorithm (SHA)-512.
- The resulting steam is XORed with 512-bit long nonce ω. The function SHA algorithm and Keccack (SHAKE) is used to extend the 512-bit long stream to d-bits forming the reference stream C* [43–45].
- The CRP mechanisms are based on the d-bit long stream C* to generate N responses from N challenges:

*Research supported by the US Department of Defense.

Author is with Northern Arizona University, Flagstaff, AZ86011, USA (phone: 1-408-203-1648; e-mail: bertrand.cambou@nau.edu).

- *Challenges*: A "challenge" is defined as the digital information needed to point at a particular position in the d-bit long stream C*. A stream of bits S* is generated by hashing and extending with eXtended output Function (XoF), creating a randomly selected seed S. The stream S* is segmented into N challenges $\{q_1, \dots, q_i, \dots, q_N\}$ that are D-bit long. The D bits of each challenges q_i are converted into number x_i, with $x_i \in \{1, d\}$, which is an address in C*, resulting in N addresses $\{x_1, \dots, x_i, \dots, x_N\}$
- *Responses:* The N addresses generate the P-bit long responses $\{r_1, \dots, r_i, \dots, r_N\}$. From each address x_i, P-bit long responses are generated from C*. The iterative method to find the P positions $\{x_{i,1}, \dots, x_{i,j}, \dots, x_{i,P}\}$, and read the P-bits is the following: The first position is: $x_{(i,j=1)} = x_i$. The other positions $x_{(i,j)}$ are given by the linear congruent random number generator, $i \in \{1, N\}$, $j \in \{1, P\}$, see equation (1), α and β are prime numbers:

$$x_{(i,j)} = \alpha\, x_{(i,j-1)} + \beta \bmod d \qquad (1)$$

- *Subset of responses*: The N-bit long ephemeral key Pk is used to filtrate the set of P-bit long responses. The subset of responses consists of the responses located at the addresses with a state of "1" in the N-bit long key. The responses located the addresses with a state of "0" in the key are discarded.

The computing power required to run the CRP mechanism is low. The security of the CRP mechanism can be enhanced with additional multi-factors of authentication.

B. *Initial set up: Encryption of the files*

The initial setting up, also called the enrollment cycle, should be carried out in a secure environment. The CRP processing, as described in Section A, enables the encryption of each file with a separate ephemeral key Pk_i. and AES. For each file F_i, we generate the two sub-keys Kc_i and Kr_i required to retrieve Pk_i, then encrypt the file F_i, see Fig.1.

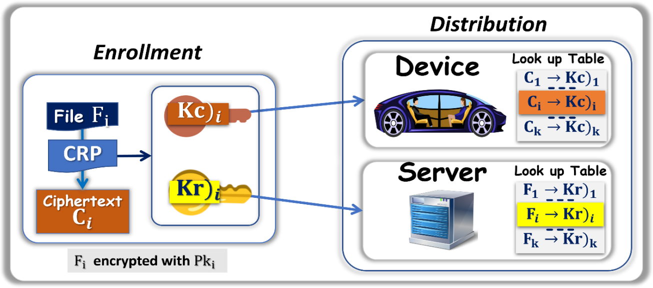


Figure 1. Enrollment cycle. From file F_i, two sub-keys are computed with the CRP mechanism: Kr_i from the responses and Kc_i from the challenges. F_i and Kr_i are kept by the server; C_i and Kc_i are kept by the terminal device.

The subkeys are defined as follows:

- Let $Kc_i = S_i$ be the seed randomly picked to generate the challenges for file F_i.
- Let Kr_i be the subset of responses computed from the full set of responses and filtrated by the ephemeral key Pk_i.

After completion of the enrollment cycles, the server keeps all files F_i and associated subkeys Kr_i, while the autonomous vehicle keeps the ciphertexts C_i and subkeys Kc_i.

C. *Normal operation in contested areas*

When the vehicle needs guidance, the operators decide that the vehicle should use file F_i; thus, they transmit Kr_i through the network (see Fig.2). The vehicle can quickly determine which file to retrieve with the following process:

- Sub-key Kc_i enables the generation of the full set of responses from ciphertext C_i, and the CRP mechanism.
- Sub-key Kr_i provides the subset of responses.
- The ephemeral key Pk_i is retrieved by comparing both sets of responses.
- File F_i is decrypted from C_i with key $Pk_{i.}$

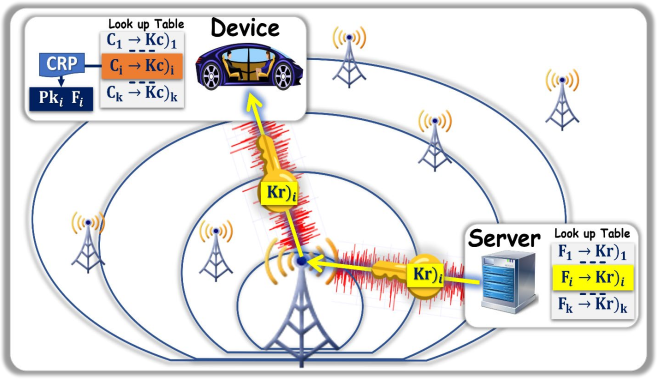


Figure 2. Recovery cycle. The server transmits Kr_i. The device retrieves the key Pk_i with the CRP mechanism, and sub-keys Kr_i and Kc_i, then decrypt F_i.

If needed, the noise can be directly injected into Kr by the server with a random number generator. The autonomous vehicle can also be equipped with a scheme emitting obfuscating noise during communication with the server, which has the potential to mitigate some side channel attacks. Having noisy responses can increase the one-wayness of the CRP mechanism by obfuscating the cryptoanalysis. The ability to manage poor signals and heavy-injected electromagnetic noises in the subkey Kr_i is discussed in the following section.

III. Statistical Analysis

When the subsets of responses are too noisy, the ephemeral keys Pk retrieved from the CRP mechanisms could contain the following errors:

- Additional "1"s due to collisions between some response from the subset and responses from the full set that are too close from each other [62].
- Additional "0"s because some responses from the subset contain more errors than acceptable.

Let us assume that the responses are P-bit long, with integer *t* being the number of erratic bits injected in the subset of response. If integer *t* is lower than the acceptable threshold T, the second type of error should zero. Let us assume that the distribution of states of "1" or "0" in the responses follows a binomial distribution, with a probability to have a state of "1" of q, and a probability of a state of "0" of (1-q), the rate of collisions Ψ(P,T,q) is given by equation (2):

$$\Psi(P,T,q) = \sum_{t=0}^{T}\left[\binom{P}{t} q^{t} (1-q)^{P-t}\right] \qquad (2)$$

With this equation, the collision rates Ψ(P,T,q) for q=0.5 at various levels of response size P are computed as a function of the ratio T/P, which represents the acceptable bit error rates in the responses, see Fig. 3 and Fig. 4.

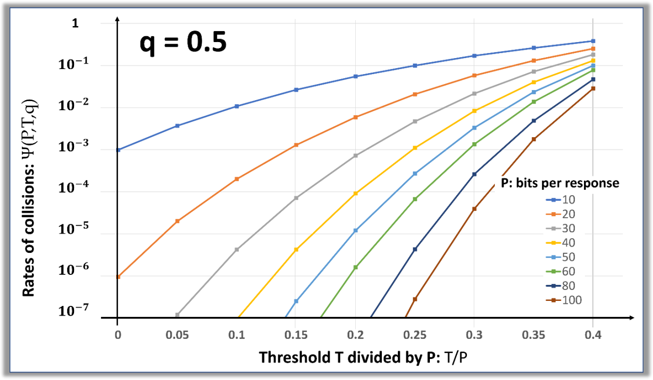


Figure 3. Rates of collisions as a function T/P with T the number of acceptable erratic bits, and P the number of bits of each response.

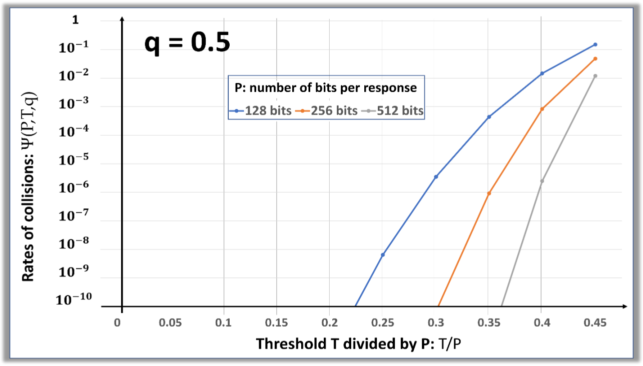


Figure 4. Rates of collisions as a function T/P with longer responses.

Rather than using error correcting schemes to correct the residual errors left in the ephemeral keys, we implemented a variation of the Response-Base Cryptography (RBC), a search engine that can find the correct key [46-47]. The RBC implemented here is able to find erratic keys with up to 15 bad bits, therefore these models anticipate that with P=32 the protocol should be able to operate with injected error rates between 25 and 30%, while with P=512, the protocol should work above 40%.

IV. Experimental Section

To validate the model, we conducted the following initial experiment [48]:

- Generate several full sets of 256 responses from a pseudo random number generator (PRNG) with the size of the responses P varying from 2 to 256.
- Generate a 256-bit long key from the PRNG and filter the responses with the key to get several subsets of 256 responses.
- Measure the rates of collisions when the bit error rate varies from 0% to 45% in the various subsets of responses.
- Repeat the experiment one thousand times, compare the results with the theoretical model.

The results are plotted in Fig.5, showing an excellent match between the experimental and theoretical curves.

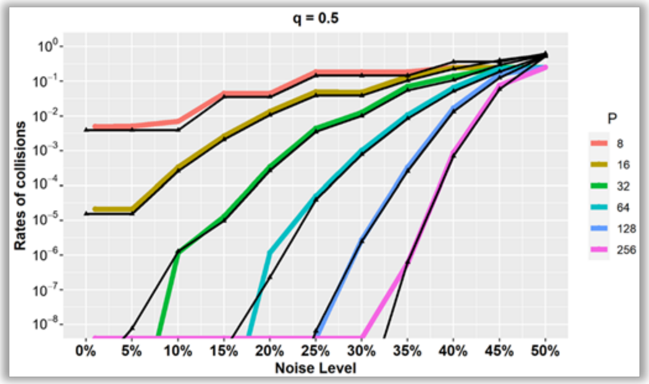


Figure 5. Rates of collisions as a function of the noise level: experiment (color) versus model (black).

A prototype was designed to be able to test and optimize the end-to-end protocol. The experimental set up is as follows:

1) *The server*: We used a regular PC to download the code needed to drive a rover through Bluetooth communication. A graphic user interface (GUI) was designed to allow near real-time operation.
2) *The device*: The rover, shown in Fig.6, is powered by a Raspberry Pi. We downloaded the code, and encrypted files to allow near real-time operation.

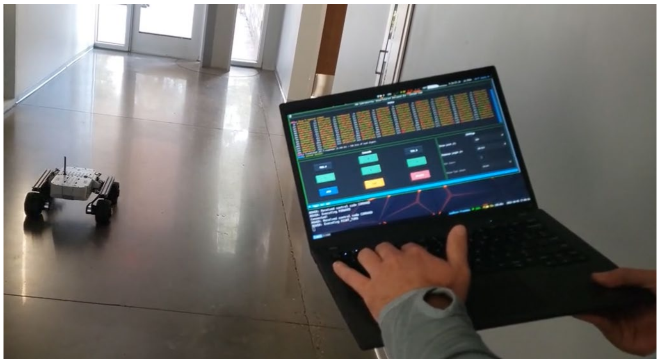


Figure 6. Experimental set up: a PC operating a rover

3) *The set of instructions*: We prepared a set of nine instructions to guide the rover in near real-time such as go left, right, up, down… The operator selects an instruction in order to transmit the corresponding sub-key Kr to the rover. The Raspberry Pi decrypt the instruction to drive the rover. The overall response times are well below one second.

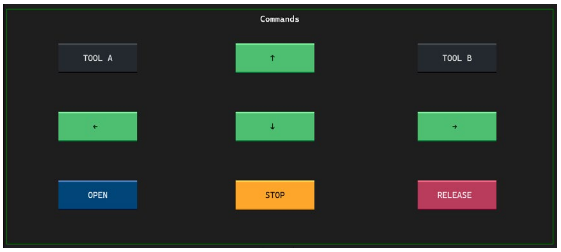
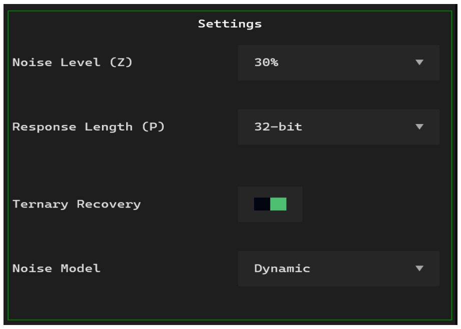


Figure 7. GUI showing 9 possible instructions to the operator.

4) *The adjustable parameters*: It is important to be able to adjust easily the parameters important to the protocol. The GUI shown in Fig. 8 allows the following:
   - Adjust the noise level (between 0% and 50%) that is injected to the sub-key Kr, the subset of responses.
   - Adjust the length of the responses, i.e. parameter P
   - Activate or not the error management engine, the RBC.
   - Pick the type of noise injected between a constant level in all responses, a random noise injected in all responses that is dynamic, and a burst mode emulating jamming.



Settings

Noise Level (Z) 30%

Response Length (P) 32-bit

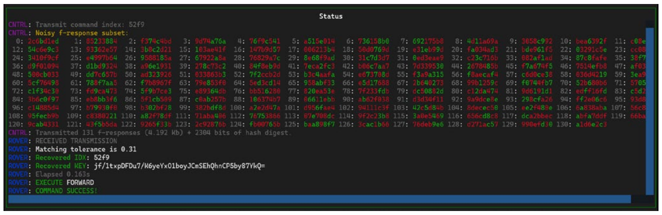
Ternary Recovery

Noise Model Dynamic

Figure 8. GUI showing the adjustable parameters

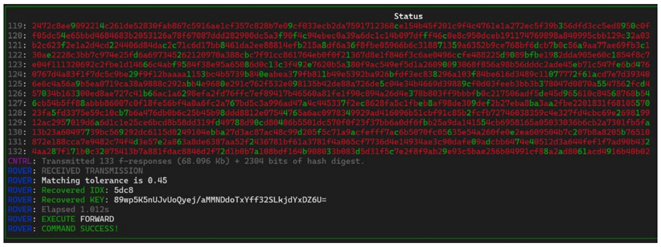
5) *Enrollment cycle*: About 1000 files were encrypted, each file containing one of the 9 instructions with a different nonce. We never use the same file twice to enhance tamper resistance. During enrollment, the encrypted files are downloaded into the rover with their corresponding sub-key Kc. The files and their corresponding sub-keys Kr are kept in the PC.
6) Operating mode: Selection of a set of instructions to guide the rover while adjusting the parameters.

When the response length (P) is set at 32-bits, we were able to guide correctly the rover with noises up to 30%. An example of such a configuration is shown in Fig. 9. With P=512 we are able to guide the rover with noises up to 44%, see Fig. 10



```
                                                    Status
CNTRL: Transmit command index: 52f9
CNTRL: Noisy f-response subset:
  0: 2c6bd1ed    1: 85233884    2: f374c4bd    3: 9d74a76a    4: 76f9c541    5: a515e014    6: 736158b0    7: 692175b8    8: 4d11a69a    9: 3058c992   10: bea6392f   11: c08e
 12: 54c6e9c3   13: 93362e57   14: 3b8c2d21   15: 103ae41f   16: 147b9d57   17: 006213b4   18: 50d0769d   19: e31eb99d   20: fa034ad3   21: bde961f5   22: 03291c5e   23: cc08
 24: 3410f9cf   25: e4997b64   26: 9588185a   27: 67922a8a   28: 76829a7c   29: 8e68f9ad   30: 31c7d3d7   31: 0ed3eae9   32: c23c716b   33: 082af1ad   34: 87c8fafe   35: 38f7
 36: d9f01094   37: d1bd9324   38: a96e1931   39: 278c73c2   40: 84f8eb9d   41: 7eca2fc3   42: b06c7aa7   43: 7d339530   44: 2678485b   45: f7a674f5   46: 7514efb8   47: af03
 48: 500cb033   49: dd7c657b   50: ad323926   51: 033863b3   52: 7f2ccb2d   53: b3c4aafa   54: e673708d   55: f3a9a315   56: f8aecaf4   57: c6d0ce38   58: 036d4219   59: 3ea9
 60: 5cf76498   61: 788f7aa5   62: f7b8967f   63: 79e853f0   64: 5ed3cd14   65: 958abf33   66: e5d17688   67: 2b640273   68: 99b1259c   69: f0744fb7   70: 52b680b6   71: 5705
 72: c1f34c38   73: fd9ca473   74: 5f9b7ce3   75: e89364db   76: bb516280   77: 820ea53e   78: 7f233fdb   79: dc50882d   80: c12da474   81: 9d6191d1   82: edff16fd   83: c5d2
 84: 3b6c0f97   85: eb8bb3f6   86: 5f1cb509   87: c0ab257b   88: 106374b7   89: 86611ebb   90: ab62f038   91: d3d34f11   92: 9a9dce8e   93: 298cfa26   94: ff2e06c6   95: 93d8
 96: c14885d4   97: b79930f0   98: b302bf28   99: 382bdf8d  100: a2e2d47a  101: d956fae4  102: 94111c3f  103: 42c5d84b  104: 8decec50  105: ee2f480e  106: 6a838aba  107: 56c8
108: 95fecb9b  109: c8380221  110: a82f78df  111: 71aba406  112: 76753866  113: 07e708dc  114: 9f2c23b8  115: 3a0e5469  116: 656cd8c8  117: dca2bbec  118: abfa7ddf  119: 66ba
120: 9cab4331  121: 43f5b5da  122: 9265f33b  123: 2c92876b  124: fb00765b  125: baa898f7  126: 3cac1b66  127: 76deb9e6  128: d271ac57  129: 990efd30  130: a1d6e2c3
CNTRL: Transmitted 131 f-responses (4.192 Kb) + 2304 bits of hash digest.
ROVER: RECEIVED TRANSMISSION
ROVER: Matching tolerance is 0.31
ROVER: Recovered IDX: 52f9
ROVER: Recovered KEY: jf/1txpDFDu7/K6yeYxO1boyJCmSEhQhnCP5by87YkQ=
ROVER: Elapsed 0.163s
ROVER: EXECUTE FORWARD
ROVER: COMMAND SUCCESS!
```

Figure 9. Successful communication with P=32 and 30% injected noise



```
                                                    Status
119: 2472c8ee9092214c261de52830fab867c5916ae1cf357c828b7e09cf033ecb2da7591712368ce154b45f201c9f4c4761e1a272ec5f39b356dfd3cc5ed8950c0f
120: f05dc54e65bbd4684683b2053126a78f67087ddd282900dc5a3f90f4c94ebec0a39a6dc1c14b097dfff46c0e8c950dceb191174769898a840995cbb129c32a03
121: b2c623f2e1a2d4cd224406d84dac2c71c6d17bb8461da2ee88814efb215a8df6a36f8fbe05966b6c318871359a6352b9ce768bf6dcb7b0c56a9aa77ae69fb3c1
122: 30ae2228c3bb7c974e25fd6a697345262120970a388cbc7f91cc861764eb0f0f21367d8e1f846f3c6ae0496ccfe488225d9089bfbe1982dda905e60c1854f8c7
123: e04f111320692c2fbe1d14666c4abf9584f38e95a65086d0c13c3f492e7620b5a380f9ac549ef5d1a26090093068f856a98b56dddc2ade45eb71c547fe6bd476
124: 0767d4a83f1f7dc5c9be29f9f12baaaa1153bc4b5739b840eabea379fb811b49e5392ba926bfdf3ec838296a103f84be616d3489c11077772f61acd7e7d39340
125: 6e6c4a56a9b5ea0719ca38a9888c292abb4e9680e291c762f532e098135b42de88a726de5c04e34b4669d39889cf0d03feeb3bb3b378047d0870a5547562fcd4
126: 57034b163300ed8ae727c41b66ac1a6298efa2fd76ffc7ef89417b46560a81fe1f90c894a26d4e378b803ff9bbbfb0c217506adf5de45d965610c84568768b54
127: 6cb54b5ff88abbb86007c0f18fe56bf4a0a6fc2a767bd5c3a996ad47a4c445337f2ec8628fa5c1fbeb8af98de309def2b27eba8ba3aa2fbe2201831f68105570
128: 23fa5fd3375e59c10cb7b6a476db0b6c25b45b98ddd8812e07544765a6ac0978349929ad416096b51cbf91c85b2fcfb72746038359c4e327fd4cbc69e2698199
129: 12ac2957019d6a6d1c1e25ce6bcd8b58dd319fd49788d90cd80406b5501dc570f0f23f37bb6a0df6fbb25a9da141554cb6958165a05033036b6cb2a7301fb5fa
130: 13b23a60497739bc569292dc6115d0249104ebba27d3ac87ac48c99d205f5c71a9acfefff7ac6b5070fc05635e54a260fe0e2ea609504b7c207b8a8205b76510
131: 872e188cca7e9482c74f4d3e57e2a863a8de6387aa52f2436781bf61a3781f4a065cf7736d4e14934ae3c90dafe09adcbb6474e40512d3a644fef1f7ad90b432
132: 4aa287f171b0c32075413b7a881fdac8846d2f72d1b0b7a108bdf164b908033b083d5d31f5c7e2f8f9ab29e93c5bae256b04991cf88a2ad8061acd4916b40b02
CNTRL: Transmitted 133 f-responses (68.096 Kb) + 2304 bits of hash digest.
ROVER: RECEIVED TRANSMISSION
ROVER: Matching tolerance is 0.45
ROVER: Recovered IDX: 5dc8
ROVER: Recovered KEY: 89wp5K5nUJvUoQyej/aMMNDdoTxYff32SLkjdYxDZ6U=
ROVER: Elapsed 1.012s
ROVER: EXECUTE FORWARD
ROVER: COMMAND SUCCESS!
```

Figure 10. Successful communication with P=512 and 44% injected noise

The hexadecimal values shown in Fig. 8 and 9 are in green when zero erratic bit is injected and are in red when at least one bad bit was injected. About 75% of the values are in red with 30% injected noise while 95% are in red with 44% noise. The total response time was always within a second, giving a near real-time feeling to the operator. In both cases the acceptable noise level went down by about 5% by disconnecting the error management engine.

V. Discussion and Conclusion

The Shannon information theory states that the highest possible randomness to be added through noise into a data steam is to flip 50% of the bits. For example, flipping 75% of the bits is equivalent to flipping 25% after a NAND operation. Therefore, when 40% noise is injected in a subset of response only 10% of the information is useful to retrieve the ephemeral keys Pk. Most error correcting schemes that we considered or tested (such as BCH, Reed-Salomon…) are limited to bit error rates of 30% with data helpers 100 longer than the transmitted information, and lengthy latencies. With P=32, we only need to transmit a stream 16 times longer than the keys we wish to retrieve, at latencies in the 100 ms range.

We are currently seeking third party expertise to identify the weak points of the protocol, and insert additional mitigations, with a focus on tamper resistance. Unlike helper data that can leak information, the subsets of responses transmitted through the unsecure network consist of totally random streams. As the generation of C* from the encrypted files is protected by passwords and additional factors of authentication, access to the encrypted files, and the sub-key Kc is unlikely to disclose useful information to an opponent having access to the device. As part of the future work, we are studying other protocols based on the subset of responses under noisy conditions, such as the transmission of handshakes generating cryptographic keys on-demand. We are also studying the implementation of CRYSTALS Kyber to encapsulate the keys that are encrypting the files.

Acknowledgment

The work presented in this paper is the result of the research team of the cybersecurity laboratory of the School of Informatics, Computing, and Cyber-Systems of Northern Arizona University.

The authors also thanks Dr. Jeff Hoffstein from Brown University for his contribution to the development of the subset of response protocol which is a key piece of the methods presented in this paper.

The work presented in this paper includes some of the research effort conducted at NAU that has been funded by the US Department of Defense and the Army Research Lab DevCom as part of the program FREEDOM (PTE federal award No: W911NF-23-2-0014). The work doesn't represent the opinions of the funding agencies, it is the sole responsibility of the author.

REFERENCES

[1] Singh, M.; Pati, D. Countermeasures to Replay Attacks: A Review. *IETE* **2020**, https://doi.org/10.1080/02564602.2019.1684851.

[2] Conti, M.; Dragoni, N.; Lesyk, V. A Survey of MITM Attacks. *IEEECommun.***2016**, https://doi.org/10.1109/COMST.2016.2548426.

[3] Uma, M.; Padmavathi, G. A Survey on Various Cyber Attacks and Their Classification. *Int. J. Netw. Secur.* **2013**, *15*, 390–396.

[4] Vanstone, S.; van Oorschot, P. *An Introduction to Error Correcting Codes with Applications*; Computer Science Book 71; Springer: Berlin/Heidelberg, Germany, 2013.

[5] Korenda, A.; Afghah, F.; Cambou, B. A Secret Key Generation Scheme for Internet of Things using Ternary-States ReRAM-based PUFs. IWCMC, Limassol, Cyprus, 25–29 June 2018.

[6] Darbon, j.; Sankur, B.; Maitre, H. ECC performance for watermark protection. In *Security and Watermarking of Multimedia Contents III*; SPIE 2021; https://doi.org/10.1117/12.435451 .

[7] Gamage, H.; Weerasinghe, H.; Dias, N. A Survey on Blockchain Technology. *SN* 2020, https://doi.org/10.1007/s42979-020-00123-0 .

[8] Fang, W.; Chen, W.; Zhang, W. Digital signature scheme for information non-repudiation in blockchain. *JWCN.* 2020, https://doi.org/10.1186/s13638-020-01665-w .

[9] Guggenberger, T.; Schlatt, V.; Schmid, J.; Urbach, N. A Structured Overview of Attacks on Blockchain Systems. 25th Pacific Asia Conference on Information Systems, Dubai, UAE, 12–14 July 2021.

[10] Aggarwal, S.; Kumar, S. Attacks on blockchain. In *Advances in Computers*; Elsevier. https://doi.org/10.1016/bs.adcom.2020.08.020 .

[11] Tomasin, S.; Zhang, H.; Chorti, A.; Poor, V. Challenge-Response Physical Layer Authentication Over Partially Controllable Channels; 2022, IEEE Communication; DOI:10.1109/MCOM.001.2200339.

[12] Smith, J.; Lingham, V.; Driscoll. J.; Fraser, I. Methods and Systems of Providing Verification of Information Using a Centralized or Distributed Ledger. U.S. Patent 10,558,974 B2, February 11, 2020.

[13] Chow, A.; Chan, P.; Haldenby, P.; Lee, J. Document Tracking on a Distributed Ledger. US2017/0048216 A1, February 16, 2017.

[14] Zang, X.; Liu, C.; Chai, K.; Poslad, S. Challenge-Response Assisted Authorization Scheme for Data Access in Permissioned Blockchains. *Sensors* 2020, *20*, 4681. https://doi.org/10.3390/s20174681 .

[15] Kaehler, A. Secure Exchange of Cryptographically Signed Records. U.S. Patent 11,044,101 B2, 22 June 2021.

[16] Covaci, A.; Madeo, S.; Motylinski, P.; Vincent, S. System and Method for Authenticating Off-Chain Data Based on Proof Verification. Application US2020/0322132 A1, 8 October 2020.

[17] Uhr, J.; Hong, J.; Song, J. Tampering Verification System and Method for Financial Institution Certificates, Based on Blockchain. Application US2021/0226804 A1, 22 July 2021.

[18] Sheng, X.; McGuire, T.; Hromi, J.; Chawla, R. Computationally efficient transfer processing and auditing apparatuses, methods and systems. Application US2017/0228731 A1, 10 August 2017.

[19] Manian, Z.; Krishnan, R.; Sriram, S. Hybrid Blockchain. Application US2017/0243193 A1, 24 August 2017.

[20] Watanabe, H.; Akutsu, A.; Miyazaki, Y.; Nakadaira, A.; Fujimura, S. Contract Agreement Method, Agreement Verification Method, Contract Agreement System, Agreement Verification Device. Application US2018/0205555 A1, 19 July 2018.

[21] Harvey, A. Blockchain Enterprise Data Management. Application US2019/0207750 A1, 4 July 2019.

[22] Afghah, F.; Cambou, B. Authentication Based on a Challenge and a Response. US Patent 10,469,273, Nov. 2019.

[23] Cambou, B.; Gowanlock, M.; Heynssens, J.; Jain, S.; Philabaum, C.; Booher, D.; Burke, I.; Garrard, J.; Telesca, D.; Njilla, L. Securing Additive Manufacturing with Blockchains and Distributed PUFs. *Cryptography* **2020**, *4*, 17.

[24] Cambou, B. Secure Digital Signatures Using PUF Devices with Reduced Error Rates. U.S. Patent 11,271,759, 9 March 2022.

[25] Cambou, B.; Telesca, D.; Jacinto, H. PUF-protected methods to generate session keys. *FICC 2022, Volume 2*;

[26] Cambou, B.; Jain, S. Key Recovery for Content Protection Using Ternary PUFs Designed with Pre-Formed ReRAM. *Appl. Sci.* 2022.

[27] Haasnoot, E. Presentation attack detection and biometric recognition in a challenge-response formalism Erwin. *EURASIP J. Inf. Secur.* 2022, *2022*, 5. https://doi.org/10.1186/s13635-022-00131-y .

[28] Mohamed, M.; Shrestha, P.; Saxena, N. Challenge-response behavioral mobile authentication. ACSAC'19: Dec. 2019.

[29] Blom, R. Challenge-Response User Authentication. U.S. Patent 7,194,765 B2, 20 March 2007.

[30] Song, J.; Noh, S.; Choi, J.; Yoon, H. A practical challenge-response authentication mechanism for a Programmable Logic Controller control system with one-time password in nuclear power plants. *Nucl. Eng. Technol.* 2019, https://doi.org/10.1016/j.net.2019.05.012 .

[31] Rhee, K.; Kwak, J.; Kim, S.; Won, D. Challenge-Response Based RFID Authentication Protocol for Distributed Database Environment. In *Security in Pervasive Computing*; Computer Science; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3450, pp. 70–84.

[32] NIST. Status Report on the Third Round of PQC Standardization Process. 2022. https://doi.org/10.6028/NIST.IR.8413 , Sept 29, 2022.

[33] Bos, J.; Ducas, L.; Kiltz, E.; Lepoint, T.; Lyubashevsky, V.; Schanck, J.M.; Schwabe, P.; Seiler, G.; Stehle, D. CRYSTALS-Kyber: A CCA-secure module-lattice-based KEM. IEEE Euro S P, London, UK, April 2018; pp. 353–367. https://doi.org/10.1109/EuroSP.2018.00032 .

[34] Avanzi, R.; Bos, J.; Ducas, L.; Kiltz, E.; Lepoint, T.; Lyubashevsky, V.; Schanck, J.; Schwabe, P.; Seiler, G.; Stehle, D. CRYSTALS-KYBER Algorithm Specifications and Supporting Documentation. https://csrc.nist.gov/projects/post-quantumcryptography/round-3-submissions . NIST's PQC Standardization Process. Sept 2022.

[35] Ducas, L.; Kiltz, E.; Lepoint, T.; Lyubashevsky, V.; Schwabe, P.; Seiler, G.; Stehle, D. CRYSTALS-Dilithium. *IACR* 2018, https://doi.org/10.13154/tches.v2018.i1.238-268 .

[36] Bai, S.; Ducas, L.; Kiltz, E.; Lepoint, T.; Lyubashevsky, V.; Schwabe, P.; Seiler, G.; Stehle, D. CRYSTALS-Dilithium. NIST's PQC process. https://github.com/pq-crystals/dilithium/tree/round3. Feb 8, 2021.

[37] Bernstein, D.; Brumley, B.; Chen, M.; Chuengsatiansup, C.; Lange, T.; Marotzke, A.; Peng, B.; Tuveri, N.; van Vredendaal, C.; Yang, B.; NTRU Prime: Round 3, Submission to the NIST's PQC process. https://ntruprime.cr.yp.to/nist/ntruprime-20201007.pdf. 2020.

[38] Chen, C.; Hoffstein, J.; Whyte, W.; Zhang, Z. NIST PQ submission: NTRU Encrypt algorithm. https://csrc. nist. gov/Projects/Post-Quantum-Cryptography/Round-1-Submissions. NIST PQC 2017.

[39] Hoffstein, J.; Pipher, J.; Silverman, J. NTRU cryptosystem. In *Algorithmic Number Theory*.1998.

[40] Faugere, J.; Gauthier-Umana, V.; Otmani, A.; Perret, L.; Tillich, J. A distinguisher for high rate McEliece cryptosystems. IEEE Information Theory Workshop, Paraty, Brazil, 16–20 October 2011; pp. 282–286. https://doi.org/10.1109/ITW.2011.6089437 .

[41] Bernstein, D.; Hulsing, A.; Kolbl, S.; Niederhagen. ACM SIGSAC, CCS '19, New York, NY, USA, 11–15 November 2019; pp. 2129–2146. https://doi.org/10.1145/3319535.3363229 .

[42] Prest, T.; Fouque, J.; Hoffstein, J.; Kirchner, P.; Lyubashevsky, V.; Pornin, T.; Ricosset, T.; Seiler, G.; Whyte, W.; Zhang, Z. Falcon Compact Signatures over NTRU; Round 3, NIST PQC Process, 2020. https://www.di.ens.fr/~prest/Publications/falcon.pdf, Sept 29, 2022.

[43] Bertoni, G.; Daemen, J.; Peeters, M.; van Assche, G. The Keccak SHA-3. http://keccak.noekeon.org/Keccak-submission-3.pdf, NIST SHA-3 Competition (Round 3). Jan 14, 2011.

[44] National Institute of Standards and Technology. *Secure Hash Standard (SHS)*; 2015. https://doi.org/10.6028/NIST.FIPS.180-4 .

[45] National [Institute of Standards and Technology. *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*; NIST Federal Information Processing Standards Publication 202; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2015. https://doi.org/10.6028/NIST.FIPS.202 .

[46] Cambou, B. Unequally powered Cryptography with PUFs for networks of IoTs. In Proceedings of the IEEE Spring Simulation Conference, Tucson, AZ, USA, 29 April–2 May 2019.

[47] Cambou, B.; Philabaum, C.; Booher, D.; Telesca, D. Response-Based Cryptographic Methods with Ternary Physical Unclonable Functions. Proceeding of FICC 2019.

[48] Dina Ghanai Miandoab, D.; Garrett, M. L.; Alam, M.; Jain, S.; Assiri, Cambou, B.; Secure Cryptographic Key Encapsulation and Recovery Scheme in Noisy Network Conditions; Appl. Sci. 2025, 15(5), 2732; https://doi.org/10.3390/app15052732, March 2025