**Arrange the sentences of the following abstracts in the correct order.**

**A DNA-Based Cryptographic Key Generation Algorithm**

(5), (4), (2), (6), (1), (3)

This paper presents a detailed description of a new DNA-based cryptographic key generation algorithm that can be used to generate strong cryptographic key(s) for symmetric ciphering applications. The algorithm uses an initial private/secret key as an input to the Key-Based Random Permutation (KBRP) algorithm to generate a permutation of size n, which is half of the size of the required cryptographic key, and to derive four vectors of size n representing the DNA bases (A, C, G, and T) of the private key. The DNA vectors are mathematically processed using a linear formula to generate the cryptographic key. The generated bases are re-permuted using the same permutation vector and re-processed to determine new cryptographic keys, and this can be continued as much as new cryptographic keys are required. The performance of the new algorithm is evaluated in two different scenarios that demonstrate its high potential for providing high randomness cryptographic key(s). The results show that the generated cryptographic keys always have ≈0.7 entropy, and acceptable maximum and average run length for both 0’s and 1’s for various key-lengths and private keys.

**SHARKS: Smart Hacking Approaches for Risk Scanning in Internet-of-Things and Cyber-Physical Systems based on Machine Learning**

(6), (3), (1), (4), (7), (5), (2)

Cyber-physical systems (CPS) and Internet-of-Things (IoT) devices are increasingly being deployed across multiple functionalities, ranging from healthcare devices and wearables to critical infrastructures, e.g., nuclear power plants, autonomous vehicles, smart cities, and smart homes. These devices are inherently not secure across their comprehensive software, hardware, and network stacks, thus presenting a large attack surface that can be exploited by hackers. In this article, we present an innovative technique for detecting unknown system vulnerabilities, managing these vulnerabilities, and improving incident response when such vulnerabilities are exploited. The novelty of this approach lies in extracting intelligence from known real-world CPS/IoT attacks, representing them in the form of regular expressions, and employing machine learning (ML) techniques on this ensemble of regular expressions to generate new attack vectors and security vulnerabilities. Our results show that 10 new attack vectors and 122 new vulnerability exploits can be successfully generated that have the potential to exploit a CPS or an IoT ecosystem. The ML methodology achieves an accuracy of 97.4% and enables us to predict these attacks efficiently with an 87.2% reduction in the search space. We demonstrate the application of our method to the hacking of the in-vehicle network of a connected car. To defend against the known attacks and possible novel exploits, we discuss a defense-in-depth mechanism for various classes of attacks and the classification of data targeted by such attacks. This defense mechanism optimizes the cost of security measures based on the sensitivity of the protected resource, thus incentivizing its adoption in real-world CPS/IoT by cybersecurity practitioners.

**HookTracer: A Systemfor Automated and Accessible API Hooks Analysis**

(4), (3), (6), (5), (2), (1)

The use of memory forensics is becoming commonplace in digital investigation and incident response, as it provides critically important capabilities for detecting sophisticated malware attacks, including memory-only malware components. In this paper, we concentrate on improving analysis of API hooks, a technique commonly employed by malware to hijack the execution flow of legitimate functions. These hooks allow the malware to gain control at critical times and to exercise complete control over function arguments and return values. Existing techniques for detecting hooks, such the Volatility plugin apihooks, do a credible job, but generate numerous false positives related to non-malicious use of API hooking. Furthermore, deeper analysis to determine the nature of hooks detected by apihooks typically requires substantial skill in reverse engineering and an extensive knowledge of operating systems internals. In this paper, we present a new, highly configurable tool called hooktracer, which eliminates false positives, provides valuable insight into the operation of detected hooks, and generates portable signatures called hook traces, which can be used to rapidly investigate large numbers of machines for signs of malware infection.