DAFTAR PUSTAKA

Appel, A. W. (2015). Verification of a Cryptographic Primitive: SHA-256. ACM Transactions on Programming Languages and Systems. Princeton University.

Aumasson, J.-P., Neves, S., O'Hearn, Z., & Winnerlein, C. (2013). BLAKE2: Simpler, Smaller, Fast as MD5. International Conference on Applied Cryptography and Network Security. Canada: Springer Link.

Balakrishnan, A., & Schulze, C. (2005). Code Obfuscation Literature Survey. Computer Sciences Department University of Wisconsin, Madison.

Baldanzi, L., Crocetti, L., & Falaschi, F. (2020). Cryptographically Secure Pseudo-Random Number Generator IP-Core Based on SHA2 Algorithm. Proceedings of the Applications in Electronics Pervading Industry. Pisa: Springer LNEE.

Bavishi, N. (2011). An Executable Packer. San José State University.

Beheraa, C., & Bhaskari, L. (2015). Different Obfuscation Techniques for Code Protection. Procedia Computer Science.

Bhanot, R., & Hans, R. (2015). A Review and Comparative Analysis of Various Encryption Algorithms. International Journal of Security and Its Applications.

Davis, R. M. (1999). The Data Encryption Standard In Perspective. IEEE COMMUNICATIONS SOCIETY MAGAZINE.

Dong, S., & Li, M. (2018). Understanding Android Obfuscation Techniques: A Large-Scale Investigation in the Wild. Springer Link.

Durfina, L., & Kolar, D. (2012). C Source Code Obfuscator. Kybernetika, (hal. 494-501).

Faruki, P., Fereidooni, H., Laxmi, V., Conti, M., & Gaur, M. (2016). Android Code Protection via Obfuscation Techniques: Past, Present and Future Directions. arXiv (https://arxiv.org/pdf/1611.10231.pdf).

Frank, R. C., & Koay, P. P. (2020). Embedding Java Classes with code2vec: Improvements from. International Conference on Mining Software Repositories (MSR). Hamilton, New Zealand: IEEE.

Geethanjali, D., Ying, T. L., Melissa, C., & Balachandran, V. (2018). AEON: Android Encryption based Obfuscation. CODASPY.

Groß, T., & Müller, T. (2017). Protecting JavaScript Apps from Code Analysis. SHCIS'17.

Ivanovski, A., & Stojanovski, T. (2010). Software protection using obfuscation. Conference on Information Technologies for Young Researchers.

Jain, S. (2016). Malware Obfuscator for Malicious Executables.

Kumar, H., Kumar, S., Joseph, R., & Kumar, D. (2013). Rainbow Table to Crack Password using MD5 Hashing Algorithm. Proceedings of 2013 IEEE Conference on Information and Comminication Technologies. Vellore, India: IEEE Conference Publishing.

Lipner, S. (2014). Security and Source Code Access: Issues and Realities. IEEE Xplore. Microsoft Corporation.

May, W. E. (2015). FIPS PUB 180-4: Secure Hash Standard (SHA-256). FEDERAL INFORMATION PROCESSING STANDARDS.

Nable. (2019, September 12). Security - SHA-256 Algorithm Overview. Diambil kembali dari Solarwinds NSP: https://www.n-able.com/blog/sha-256-encryption#:~:text=SHA%2D256%20is%20a%20patented,as%20long%20as%20when%20unencrypted.

Patel, R. (2014). A Way To Protect Software Secrets From Reverse Engineering Using Code Obfuscation Techniques. ResearchGate. IEEE Symposium on Securityand Privacy (S&P'05).

Rad, B. B., & Masrom, M. (2021). Metamorphic Virus Detection in Portable Executables Using Opcodes Statistical Feature. International Scientific Conference . Proceeding of the International Conference on Advance Science, Engineering and Information Technology.

Salem, I. E., Salman, A., & Mijwil, M. (2019). A Survey: Cryptographic Hash Function for Digital Stamping. Journal of Southwest Jiaotong University.

Schneider, J., & Locher, T. (2016). Obfuscation using Encryption. arXiv:1612.03345v1.

Sebastian Schrittwieser, S. K. (2011). Code Obfuscation against Static and Dynamic Reverse Engineering. Lecture Notes in Computer Science.

Sharma, A. K., & Mittal, S. (2018). Comparative Analysis of Crypthographic Hash. Proceedings of 18th IRF International Conference. New Delhi.

Singh, A. (2009). Identifying Malicious Code Through Reverse Engineering, Advance in Information Security.

Stallings, W. (2017). Cryptography and Network Security. Pearson.

Sun, L. (2010). A Framework for Malware Packer Analysis Using Information Theory and Statistical Methods. School of Mathematical and Geospatial Sciences,College of Science, Engineering and Health,RMIT University, (hal. 22-43).

Thomsen, S. S., & Knudsen, L. R. (2009). Cryptographic Hash Functions. Technical University of Denmark.

Torgenson, W. S. (1952). Multidimensional scaling: I. Theory and method. Psychometrika, vol 17, No. 4. Social Science Research Council.

Viticchie, A., & Regano, L. (2016). Assessment of Source Code Obfuscation Techniques. Dipartimento di Automatica e Informatica,Politecnico di Torino, Torino, Italy, (hal. 1). Torino, Italy.

Vrba, Z., Halvorsen, P., & Griwodz, C. (2010). Program obfuscation by strong. International Conference on Availability, Reliability and Security.

Wang, X., & Yu, H. (2005). How to Break MD5 and Other Hash Functions. Advances in Cryptology – EUROCRYPT . SpringerVerlag, 2005.